

**Disciplina:** Conectividade de Sistemas Ciberfísicos

**Professor:** Guilherme Schnirmann

**Curso:** BES

**Nome Estudante:** Diogo Bonet Sobezak e Vittorio Caprioli

## Atividade Prática / Relatório

### Socket API

#### Descrição da Atividade:

Esta atividade consiste em implementar em Python a comunicação TCP utilizando a interface socket. O intuito desta atividade é demonstrar o funcionamento do protocolo TCP, bem como as portas são mapeadas aos processos por meio da interface sockets.

#### Entrega:

Esta atividade deverá ser entregue no AVA

O estudante deverá entregar um arquivo “.pdf” contendo as respostas da atividade proposta no item especificação.

#### Especificação:

##### Exercício 1:

1. Crie o programa (servidor.py) efetuando os passos da interface API socket como visto em sala.
  - a. Fixe o IP do HOST em 127.0.0.1 ( HOST = '127.0.0.1' )
  - b. Fixe a porta (PORTA = 9999)
  - c. Inicialize o socket (IPv4, TCP)
  - d. Faça o try-except para o bind
  - e. Printe em tela uma mensagem com IP:PORTA que o servidor está aguardando conexões (após realizar a função para isso)
2. Inclua um **while True** para que o programa receba várias conexões sem encerrar.

**Dica:** Lembre-se que concretizar a conexão precisamos do comando accept() no servidor. Ele retorna o socket cliente que se conectou e o endereço (retorna 2 valores).
3. Crie o programa cliente.py e efetue a conexão com o servidor.
4. Anote o valor de porta mostrado pelo programa servidor.
5. Encerre a conexão com o cliente, e repita os testes para obter mais dois valores de porta.
6. Efetuar um printscreen dos testes realizados.

### Relatório:

Questão 1:

a) Indique os valores das três portas obtidas com o teste

```
Servidor se conectando ao ip 127.0.0.1:9999! Aguardando servidor de cliente se conectar...
Conectando no ('127.0.0.1', 50398)
Mensagem recebida (Estou enviando uma mensagem!)
Conectando no ('127.0.0.1', 50401)
Mensagem recebida (Estou enviando uma mensagem!)
Conectando no ('127.0.0.1', 50407)
Mensagem recebida (Estou enviando uma mensagem!)
```

b) Que portas são essas? O que elas significam?

R: As portas de conexão entre o servidor e o cliente.

### Exercício 2:

1. Altere o programa servidor para que a porta TCP seja passada seja passada pelo usuário:

Dica: a porta precisa ser convertida de *string* para inteiro

```
porta = int(input('Entre com a porta do servidor'))
```

2. Execute dois servidores a partir do console do Windows (prompt de comando):

```
cmd
```

```
python servidor.py
```

```
python servidor.py
```

3. Efetuar um printscreen dos testes realizados.

### Relatório:

Questão 2:

a) Indique o que acontece quando você tenta abrir dois servidores simultaneamente na mesma porta

R: O último servidor que é aberto com a mesma porta se encerra, ou seja, não é possível abrir dois servidores com a mesma porta simultaneamente.

b) Indique o que acontece quando você tenta usar uma porta menor que 1024 (exemplo: 700)

R: O servidor e cliente envia e recebe mensagens normalmente.

Servidor:

```
[SERVIDOR] Entre com a porta do servidor: 700
Servidor se conectando ao ip 127.0.0.1:700! Aguardando servidor de cliente se conectar...
Conectando no ('127.0.0.1', 50720)
Mensagem recebida (Estou enviando uma mensagem!)
```

Cliente:

```
n cliente.py
[CLIENTE] Entre com a porta do servidor: 700
Mensagem enviada b'Estou enviando uma mensagem!'
```

Exercício 3:

1. Altere o programa servidor.py para que ele imprima as mensagens recebidas do cliente.

- a. Imprima quantos bytes foram recebidos

```
[SERVIDOR] Entre com a porta do servidor: 700
Servidor se conectando ao ip 127.0.0.1:700! Aguardando servidor de cliente se conectar...
Conectando no ('127.0.0.1', 53241)
Mensagem recebida (Estou enviando uma mensagem!)
Foram recebidos 28 bytes!
```

- b. Imprima a mensagem recebida (lembre-se de utilizar o critério de parada:

if not data:

break

print(data)

**Dica:** Faça um while True. Lembre-se de utilizar o socket vindo do cliente. Receba os dados (utilize um buffer de 1024 bytes).

2. Efetuar um printscreen dos testes realizados.

Relatório:

Questão 3:

- a) Indique o que acontece quando você tenta conectar 2 clientes simultaneamente

R: Quando o segundo cliente é executado o primeiro é cancelado.

- b) Explique o que acontece quando o primeiro cliente é encerrado.

R: O servidor continua executando, independente do cliente conectado.

- c) Indique como o conteúdo é recebido quando você usa um buffer pequeno no servidor (exemplo 5).

R: O servidor recebe apenas 5 bytes de mensagem