# IFES - Serra

# Redes de Computadores Cristina Klippel Dominicini

# Matheus de Souza Pereira de Oliveira 20191BSI0301

Relatório do Trabalho #1 Utilizando API de socket em Python

### 1. Testes

Teste 1 - Primeira compra do consumidor

```
python3 servidor.py
                                                                                                                                        ● ● ● → matheus@matheus-Inspiron-5558:~/Documentos/P
 matheus in trabalho-redes/servidor on 🍃 master [!?]
) python3 servidor.py
Servidor iniciado!
                                                                                                                                      matheus in trabalho-redes/clientes/consumidor on praster [!?]
Host: localhost Port: 5000
Cliente ('127.0.0.1', 38372)
Cliente CONSUMIDOR
                                                                                                                                      Item: sapato
Quantidade em estoque: 90
Valor unitário: 150.0
#ESTOOUE#
Ttem: sapato
Quantidade em estoque: 90
Valor unitário: 150.0
                                                                                                                                      Quantidade em estoque: 97
Valor unitário: 49.99
Item: camiseta
Quantidade em estoque: 97
Valor unitário: 49.99
                                                                                                                                      Item: camiseta
Quantidade solicitada: 4
#PEDIDO DO CLIENTE CONSUMIDOR#
Item: camiseta
Quantidade solicitada: 4
                                                                                                                                      Item: sapato
Quantidade solicitada: 1
Item: sapato
Quantidade solicitada: 1
                                                                                                                                         "numero_pedido": 538570,
"vlTotal": 349.960000000000004,
"itens": [
Pedido realizado com sucesso! :D
                                                                                                                                               "item": "camiseta",
"quantidade": 4,
"valorUnitario": 49.99
#ESTOOUE#
Quantidade em estoque: 89
Valor unitário: 150.0
                                                                                                                                                "quantidade": 1,
"valorUnitario": 150.0
Quantidade em estoque: 93
Valor unitário: 49.99
```

# Teste 2 - Segunda interação de compra

```
python3 servidor.py

→ matheus@matheus-Inspiron-55

Quantidade em estoque: 93
                                                                                                            "item": "sapato",
Valor unitário: 49.99
                                                                                                            "quantidade": 1,
"valorUnitario": 150.0
Cliente ('127.0.0.1', 38380)
Cliente CONSUMIDOR
                                                                                                     matheus in trabalho-redes/clientes/consumidor
                                                                                                     > python3 cliente-consumidor.py
#ESTOOUE#
Item: sapato
Quantidade em estoque: 89
                                                                                                     #ESTOOUE#
                                                                                                     Item: sapato
Valor unitário: 150.0
                                                                                                     Quantidade em estoque: 89
                                                                                                     Valor unitário: 150.0
Item: camiseta
Quantidade em estoque: 93
Valor unitário: 49.99
                                                                                                     Item: camiseta
                                                                                                     Quantidade em estoque: 93
                                                                                                     Valor unitário: 49.99
#PEDIDO DO CLIENTE CONSUMIDOR#
                                                                                                     #PEDIDO FEITO A LOJA#
Item: sapato
Quantidade solicitada: 5
                                                                                                     Item: sapato
                                                                                                     Quantidade solicitada: 5
Pedido realizado com sucesso! :D
                                                                                                       "numero_pedido": 116279,
"vlTotal": 750.0,
"itens": [
#ESTOOUE#
Item: sapato
Quantidade em estoque: 84
                                                                                                            "item": "sapato",
Valor unitário: 150.0
                                                                                                            "quantidade": 5,
"valorUnitario": 150.0
Item: camiseta
Quantidade em estoque: 93
Valor unitário: 49.99
```

# Teste 3 - Compra aceita pela loja

```
) python3 cliente-fornecedor.py
Cliente ('127.0.0.1', 38400)
Cliente FORNECEDOR
                                                         Item: sapato
Quantidade em estoque: 84
#ESTOQUE#
                                                         Valor unitário: 150.0
Item: sapato
Quantidade em estoque: 84
Valor unitário: 150.0
                                                         Item: camiseta
                                                         Quantidade em estoque: 93
                                                         Valor unitário: 49.99
Item: camiseta
Quantidade em estoque: 93
Valor unitário: 49.99
                                                         #PEDIDO DE VENDA PARA A LOJA#
#PEDIDO DO CLIENTE FORNECEDOR#
                                                              "item": "sapato",
"qtdForn": 3,
"vlUnitario": 75.0
Item: sapato
Quantidade fornecida: 3
Valor unitario: 75.0
Item: camiseta
                                                              "item": "camiseta",
Quantidade fornecida: 2
Valor unitario: 24.995
                                                             "qtdForn": 2,
"vlUnitario": 24.995
Deseja realizar essa compra? (s/n)s
                                                            "message": "A compra foi aceita pela loja! :D"
Compra da loja!
                                                            theus in trabalho-redes/clientes/fornecedor on 🌣 master [!?] took 17s
Item: sapato
Quantidade em estoque: 87
Valor unitário: 150.0
Item: camiseta
Quantidade em estoque: 95
Valor unitário: 49.99
```

## Teste 4 - Compra rejeitada pela loja

```
matheus in trabalho-redes/clientes/fornecedor
Cliente ('127.0.0.1', 38424)
                                             > python3 cliente-fornecedor.py
Cliente FORNECEDOR
                                             #ESTOQUE#
                                             Item: sapato
#ESTOQUE#
                                             Quantidade em estoque: 87
Item: sapato
                                             Valor unitário: 150.0
Quantidade em estoque: 87
Valor unitário: 150.0
                                             Item: camiseta
                                             Quantidade em estoque: 95
                                             Valor unitário: 49.99
Item: camiseta
Quantidade em estoque: 95
Valor unitário: 49.99
                                             #PEDIDO DE VENDA PARA A LOJA#
#PEDIDO DO CLIENTE FORNECEDOR#
                                                 "item": "sapato",
Item: sapato
                                                 "qtdForn": 3,
Quantidade fornecida: 3
                                                  "vlUnitario": 75.0
Valor unitario: 75.0
Item: camiseta
                                                 "item": "camiseta",
Quantidade fornecida: 2
                                                  "qtdForn": 2,
Valor unitario: 24.995
                                                  "vlUnitario": 24.995
Deseja realizar essa compra? (s/n)n
Pedido rejeitado
                                                "message": "Pedido rejeitado! :("
#ESTOQUE#
                                             matheus in trabalho-redes/clientes/fornecedor
                                              > [
Item: sapato
Quantidade em estoque: 87
Valor unitário: 150.0
Item: camiseta
Quantidade em estoque: 95
Valor unitário: 49.99
```

### 2. Instruções de execução

- 1- Clonar repositório <a href="https://github.com/matheuss3/trabalho-redes.git">https://github.com/matheuss3/trabalho-redes.git</a> ou extrair arquivo .zip enviado
- 2- Para executar o servidor, abrir terminal na pasta <u>trabalho-redes/servidor</u> e executar o comando <u>python3 servidor.py</u>
- 3- Para executar o cliente consumidor abrir terminal na pasta trabalho-redes/clientes/consumidor e executar o comando python3 cliente-consumidor.py
- 4-Para executar cliente consumidor abrir terminal 0 na pasta trabalho-redes/clientes/fornecedor executar comando python3 е 0 cliente-fornecedor.py

#### 3. Conclusão

Com este trabalho foi possível entender na prática os assuntos estudados em aula, utilizando a api de socket do python construímos um protocolo (utilizando uma conexão do tipo TCP) específico para venda, compra e gerência de estoque dos produtos de uma loja de conteúdo esportivos, através desse protocolo toda comunicação é feita entre a loja e seus consumidores e fornecedores.

Portanto fica clara a necessidade de possuir regras bem estabelecidas pelos protocolos, pois é com eles que as comunicações entre as redes são feitas. É possível perceber também a dificuldade de fazer essa comunicação de uma forma eficaz, pois para um problema existem n soluções nessa camada onde estávamos programando, nós éramos os responsáveis por estabelecer as regras, deste modo uma liberdade maior foi obtida na hora de escrever o código, mas, em contrapartida é necessário muito tempo pra pensar em como os dados serão transmitidos, qual formato, se precisarão ser serializados, compactados, codificados etc. Além desse problema é necessário ficar muito atento ao codificar, porque ao criar um *send* ou *recieve* no servidor se faz necessário receber ou enviar dados através dos clientes.

Para a transmissão dos dados utilizei o padrão JSON e a sua própria biblioteca em python para serializar e enviar os dados através dos clientes e do servidor, sendo assim quando alterava qualquer atributo ou tipo de favor na minha estrutura praticamente todo meu código apresentava problemas e todo ele precisaria ser ajustado mostrando um forte acoplamento entre as entidades o que é ruim para uma manutenção futura.

Apesar dos problemas, o algoritmo funcionou de forma satisfatória de acordo com as figuras apresentadas anteriormente neste documento, cumprindo com seu objetivo de gerenciar o estoque da loja e receber as requisições dos clientes.