|  |  |  |  |
| --- | --- | --- | --- |
| **Curso:** | *Sistemas de Informação* | | |
| **Disciplina:** | *Comunicação de Dados* | **Turma:** | *03J* |
| **Professor:** | *Wallace Rodrigues de Santana* | **Semestre:** | *2022.2* |

|  |  |  |
| --- | --- | --- |
| **#** | **Nome do Aluno** | **RA** |
| 1 | *Jean Pazzini Domingues* | *10420319* |
| 2 | *Luis Felipe Santos do Nascimento* | *10420572* |
| 3 |  |  |
| 4 |  |  |

**ATIVIDADE 2**

**Laboratório 2 – Socket**

Após analisar o comportamento da comunicação de processos, responda às seguintes perguntas:

1. Execute o cliente TCP antes de executar o servidor TCP. O que acontece? Por quê? [1,0 ponto]

|  |
| --- |
| Resposta: Não vai funcionar, pois o servidor não foi acionado antes para esperar uma conexão, entrando em um modo de “escuta” por exemplo. Acionando primeiro o cliente, ele vai buscar se conectar ao socket do servidor que não existe ou não foi acionado ainda. |

1. Faça o mesmo procedimento para o cliente e servidor UDP. O resultado foi similar ao socket TCP? Compare os resultados e justifique. [2,0 pontos]

|  |
| --- |
| Resposta: O resultado não será similar pela diferença de natureza dos protocolos, já que pode acontecer do pacote não chegar ao destino, porque o UDP não oferece garantias de entrega, integridade ou ordem dos dados transferidos. Existe uma chance dos pacotes chegarem duplicados, fora de ordem ou até mesmo não chegarem. |

1. O que acontece se o número da porta que o cliente tentar se conectar for diferente da porta disponibilizada pelo servidor? [1,0 ponto]

|  |
| --- |
| Resposta: Caso a porta especificada pelo cliente for diferente da porta do servidor que está no modo de “escuta”, a conexão será simplesmente rejeitada, podendo indicar uma mensagem de erro para o cliente informando “Porta fechada”, ou “Firewall bloqueando a porta”, ou até mesmo “Erro de digitação”. |

**PARTE II**

1. Faça um chat entre cliente e servidor (UDP ou TCP) onde ambos os lados trocam mensagens até uma das partes enviar o comando QUIT. A porta do socket deve ser os primeiros cinco números do TIA do primeiro aluno do grupo (em ordem alfabética). [6,0 pontos]

|  |
| --- |
| Código do servidor [3,0 pontos]:  import socket  TCP\_IP = 'localhost' # ou '127.0.0.1'  TCP\_PORT = 10420  BUFFER\_SIZE = 1024  # Criação do socket TCP  server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  server.bind((TCP\_IP, TCP\_PORT))  server.listen(1)  print(f"Servidor aguardando conexões na porta {TCP\_PORT}...")  conn, addr = server.accept()  print(f'Cliente conectado: {addr}')  try:  while True:  # Recebe mensagem do cliente  data = conn.recv(BUFFER\_SIZE).decode('utf-8')  if not data or data.upper() == 'QUIT':  print("Cliente encerrou a conexão")  break  print(f"Cliente: {data}")    # Envia resposta para o cliente  message = input("Servidor: ")  if message.upper() == 'QUIT':  conn.send(message.encode('utf-8'))  print("Encerrando servidor...")  break  conn.send(message.encode('utf-8'))  finally:  conn.close()  server.close() |

|  |
| --- |
| Código do cliente [3,0 pontos]:  import socket  TCP\_IP = 'localhost' # ou '127.0.0.1'  TCP\_PORT = 10420  BUFFER\_SIZE = 1024  # Criação do socket TCP  client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  client.connect((TCP\_IP, TCP\_PORT))  print("Conectado ao servidor! Digite suas mensagens (QUIT para sair)")  try:  while True:  # Envia mensagem para o servidor  message = input("Cliente: ")  client.send(message.encode('utf-8'))    if message.upper() == 'QUIT':  print("Encerrando conexão...")  break    # Recebe resposta do servidor  data = client.recv(BUFFER\_SIZE).decode('utf-8')  if not data or data.upper() == 'QUIT':  print("Servidor encerrou a conexão")  break  print(f"Servidor: {data}")  finally:  client.close() |