

# Aplicação Cliente-Servidor com ZeroMQ

## 1. Descrição do Projeto

Este projeto implementa uma aplicação **cliente-servidor** utilizando a biblioteca **ZeroMQ** em Python, baseada em tarefas anteriores com RPC. A aplicação permite que o cliente realize operações sobre uma lista remota armazenada no servidor, incluindo:

- Mostrar a lista atual
- Adicionar elementos (**append**)
- Inserir elementos em posições específicas
- Remover elementos
- Pesquisar elementos
- Ordenar a lista (crescente ou decrescente)
- Limpar a lista

O servidor mantém a lista em memória e responde aos comandos enviados pelo cliente via protocolo **REQ-REP (request-reply)** do ZeroMQ.

## 2. Estrutura dos Arquivos

- **servidorZMQ.py** – Código do servidor, responsável por receber comandos do cliente e manipular a lista.
- **clienteZMQ.py** – Código do cliente, responsável por enviar comandos ao servidor e exibir respostas.
- **constZMQ.py** – Arquivo de configuração contendo o host e porta do servidor.

Servidor:

```
host = "0.0.0.0"  
port = 1234
```

○

Cliente:

```
host = "3.221.106.59"  
port = 1234
```

## 3. Funcionamento

### Servidor

1. O servidor cria um socket **REP** (`socket = context.socket(zmq.REP)`) e faz `bind` na porta definida.
2. Aguarda mensagens do cliente em um loop infinito.
3. Quando uma mensagem chega, ela é processada e uma resposta é enviada de volta.
4. Todas as ações na lista (`append`, `insert`, `remove`, etc.) são feitas na memória do servidor.
5. Prints de debug detalham cada passo, permitindo monitoramento em tempo real.

### Cliente

1. O cliente cria um socket **REQ** (`socket = context.socket(zmq.REQ)`) e se conecta ao servidor usando o IP público da EC2.
2. O usuário interage através de um menu de opções.
3. Cada comando do usuário é convertido em uma string e enviado ao servidor.
4. O cliente aguarda a resposta do servidor e exibe na tela.
5. Prints de debug mostram o envio e recebimento de mensagens.

## 4. Interação via AWS EC2

1. O servidor foi hospedado em uma instância EC2.
2. É necessário liberar a porta 1234 no **Security Group** do servidor para permitir conexões externas.
3. O cliente pode ser executado em outra instância EC2 ou máquina local, conectando-se ao **IP público** do servidor.
4. Fluxo típico de execução:
  - Iniciar `servidorZMQ.py` na EC2 do servidor.
  - Executar `clienteZMQ.py` na EC2 do cliente ou localmente.
  - Interagir com o menu para enviar comandos e receber respostas.

## 5. Exemplo de Execução

Servidor:

```
[DEBUG] Servidor ZeroMQ iniciado em 0.0.0.0:1234
[DEBUG] Aguardando conexões e mensagens...
[DEBUG] Mensagem recebida: append 5
[DEBUG] Enviando resposta ao cliente: Elemento '5' adicionado.
```

Cliente:

```
=====
Cliente ZeroMQ - Lista Remota
1 - Mostrar lista atual
2 - Adicionar elemento (append)
...
Escolha uma opção: 2
Digite o elemento a adicionar: 5
[DEBUG] Enviando comando ao servidor: append 5
[DEBUG] Resposta recebida do servidor: Elemento '5' adicionado.
```

## 6. Vantagens do ZeroMQ

- **Leve e rápido:** não requer servidores de mensagens externos; é apenas uma biblioteca.

- **Flexível:** suporta vários padrões de comunicação (REQ-REP, PUB-SUB, PUSH-PULL, etc.).
- **Assíncrono e escalável:** permite criar sistemas distribuídos com facilidade.
- **Portável:** funciona em diversas plataformas sem configuração complexa.

## 7. Desvantagens do ZeroMQ

- **Não possui persistência:** mensagens só existem na memória; se o servidor cair, dados podem ser perdidos.
- **Não possui controle de autenticação integrado:** é necessário implementar segurança adicional.
- **Debug e monitoramento podem ser complexos** em sistemas muito grandes, pois não há console central.

## 8. Observações

- Todos os testes foram realizados em instâncias EC2 da AWS, com a porta 1234 aberta para permitir comunicação externa.
- A aplicação demonstra como substituir uma arquitetura RPC por um modelo **client-server** com ZeroMQ, mantendo funcionalidades similares.

## Prompt servidor

```
[ec2-user@ip-172-31-86-242 ~]$ python3 servidorZMQ.py
```

```
[DEBUG] Servidor ZeroMQ iniciado em 0.0.0.0:1234
```

```
[DEBUG] Aguardando conexões e mensagens...
```

```
[DEBUG] Esperando mensagem do cliente...
```

```
[DEBUG] Mensagem recebida: mostrar
```

```
[DEBUG] Enviando resposta ao cliente: []
```

```
[DEBUG] Resposta enviada com sucesso.
```

```
[DEBUG] Esperando mensagem do cliente...
```

```
[DEBUG] Mensagem recebida: append 3
```

```
[DEBUG] Enviando resposta ao cliente: Elemento '3' adicionado.
```

```
[DEBUG] Resposta enviada com sucesso.
```

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: append 3  
[DEBUG] Enviando resposta ao cliente: Elemento '3' adicionado.  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: append 5  
[DEBUG] Enviando resposta ao cliente: Elemento '5' adicionado.  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: append 7  
[DEBUG] Enviando resposta ao cliente: Elemento '7' adicionado.  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: append 9  
[DEBUG] Enviando resposta ao cliente: Elemento '9' adicionado.  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: append 3  
[DEBUG] Enviando resposta ao cliente: Elemento '3' adicionado.  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: append 5  
[DEBUG] Enviando resposta ao cliente: Elemento '5' adicionado.  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: mostrar  
[DEBUG] Enviando resposta ao cliente: ['3', '3', '5', '7', '9', '3', '5']  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: insert 4 5  
[DEBUG] Enviando resposta ao cliente: Elemento '5' inserido na posição 4.  
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...  
[DEBUG] Mensagem recebida: remover 5  
[DEBUG] Enviando resposta ao cliente: Elemento '5' removido.  
[DEBUG] Resposta enviada com sucesso.

```
[DEBUG] Esperando mensagem do cliente...
[DEBUG] Mensagem recebida: pesquisar 6
[DEBUG] Enviando resposta ao cliente: Não encontrado
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...
[DEBUG] Mensagem recebida: ordenar_cresc
[DEBUG] Enviando resposta ao cliente: Lista ordenada crescentemente.
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...
[DEBUG] Mensagem recebida: mostrar
[DEBUG] Enviando resposta ao cliente: ['3', '3', '3', '5', '5', '7', '9']
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...
[DEBUG] Mensagem recebida: ordenar_desc
[DEBUG] Enviando resposta ao cliente: Lista ordenada decrescentemente.
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...
[DEBUG] Mensagem recebida: mostrar
[DEBUG] Enviando resposta ao cliente: ['9', '7', '5', '5', '3', '3', '3']
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...
[DEBUG] Mensagem recebida: limpar
[DEBUG] Enviando resposta ao cliente: Lista limpa.
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...
[DEBUG] Mensagem recebida: mostrar
[DEBUG] Enviando resposta ao cliente: []
[DEBUG] Resposta enviada com sucesso.

[DEBUG] Esperando mensagem do cliente...
client_loop: send disconnect: Connection reset
```

## prompt cliente

```
[ec2-user@ip-172-31-82-211 ~]$ python3 clienteZMQ.py
[DEBUG] Cliente tentando conectar ao servidor tcp://3.221.106.59:1234
```

[DEBUG] Conexão estabelecida com sucesso.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 1

[DEBUG] Opção escolhida: 1

[DEBUG] Enviando comando ao servidor: mostrar

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: []

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 2

[DEBUG] Opção escolhida: 2

Digite o elemento a adicionar: 3

[DEBUG] Enviando comando ao servidor: append 3

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Elemento '3' adicionado.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica

- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 2

[DEBUG] Opção escolhida: 2

Digite o elemento a adicionar: 3

[DEBUG] Enviando comando ao servidor: append 3

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Elemento '3' adicionado.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 2

[DEBUG] Opção escolhida: 2

Digite o elemento a adicionar: 5

[DEBUG] Enviando comando ao servidor: append 5

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Elemento '5' adicionado.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair



```
=====
Escolha uma opção: 2
[DEBUG] Opção escolhida: 2
Digite o elemento a adicionar: 7
[DEBUG] Enviando comando ao servidor: append 7
[DEBUG] Aguardando resposta do servidor...
[DEBUG] Resposta recebida do servidor: Elemento '7' adicionado.
```

```
=====
Cliente ZeroMQ - Lista Remota
1 - Mostrar lista atual
2 - Adicionar elemento (append)
3 - Inserir elemento em posição específica
4 - Remover elemento
5 - Pesquisar elemento
6 - Ordenar lista (crescente)
7 - Ordenar lista (decrescente)
8 - Limpar lista
9 - Sair
```

```
=====
Escolha uma opção: 2
[DEBUG] Opção escolhida: 2
Digite o elemento a adicionar: 9
[DEBUG] Enviando comando ao servidor: append 9
[DEBUG] Aguardando resposta do servidor...
[DEBUG] Resposta recebida do servidor: Elemento '9' adicionado.
```

```
=====
Cliente ZeroMQ - Lista Remota
1 - Mostrar lista atual
2 - Adicionar elemento (append)
3 - Inserir elemento em posição específica
4 - Remover elemento
5 - Pesquisar elemento
6 - Ordenar lista (crescente)
7 - Ordenar lista (decrescente)
8 - Limpar lista
9 - Sair
```

```
=====
Escolha uma opção: 2
[DEBUG] Opção escolhida: 2
Digite o elemento a adicionar: 3
[DEBUG] Enviando comando ao servidor: append 3
[DEBUG] Aguardando resposta do servidor...
```

[DEBUG] Resposta recebida do servidor: Elemento '3' adicionado.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 2

[DEBUG] Opção escolhida: 2

Digite o elemento a adicionar: 5

[DEBUG] Enviando comando ao servidor: append 5

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Elemento '5' adicionado.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 1

[DEBUG] Opção escolhida: 1

[DEBUG] Enviando comando ao servidor: mostrar

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: ['3', '3', '5', '7', '9', '3', '5']

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica

- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 3

[DEBUG] Opção escolhida: 3

Posição onde inserir: 4

Elemento a inserir: 5

[DEBUG] Enviando comando ao servidor: insert 4 5

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Elemento '5' inserido na posição 4.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 4

[DEBUG] Opção escolhida: 4

Elemento a remover: 5

[DEBUG] Enviando comando ao servidor: remove 5

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Elemento '5' removido.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista

9 - Sair

=====

Escolha uma opção: 5

[DEBUG] Opção escolhida: 5

Elemento a pesquisar: 6

[DEBUG] Enviando comando ao servidor: pesquisar 6

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Não encontrado

=====

Cliente ZeroMQ - Lista Remota

1 - Mostrar lista atual

2 - Adicionar elemento (append)

3 - Inserir elemento em posição específica

4 - Remover elemento

5 - Pesquisar elemento

6 - Ordenar lista (crescente)

7 - Ordenar lista (decrescente)

8 - Limpar lista

9 - Sair

=====

Escolha uma opção: 6

[DEBUG] Opção escolhida: 6

[DEBUG] Enviando comando ao servidor: ordenar\_cresc

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Lista ordenada crescentemente.

=====

Cliente ZeroMQ - Lista Remota

1 - Mostrar lista atual

2 - Adicionar elemento (append)

3 - Inserir elemento em posição específica

4 - Remover elemento

5 - Pesquisar elemento

6 - Ordenar lista (crescente)

7 - Ordenar lista (decrescente)

8 - Limpar lista

9 - Sair

=====

Escolha uma opção: 1

[DEBUG] Opção escolhida: 1

[DEBUG] Enviando comando ao servidor: mostrar

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: ['3', '3', '3', '5', '5', '7', '9']

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 7

[DEBUG] Opção escolhida: 7

[DEBUG] Enviando comando ao servidor: ordenar\_desc

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Lista ordenada decrescentemente.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 1

[DEBUG] Opção escolhida: 1

[DEBUG] Enviando comando ao servidor: mostrar

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: ['9', '7', '5', '5', '3', '3', '3']

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento

- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 8

[DEBUG] Opção escolhida: 8

[DEBUG] Enviando comando ao servidor: limpar

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: Lista limpa.

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 1

[DEBUG] Opção escolhida: 1

[DEBUG] Enviando comando ao servidor: mostrar

[DEBUG] Aguardando resposta do servidor...

[DEBUG] Resposta recebida do servidor: []

=====

Cliente ZeroMQ - Lista Remota

- 1 - Mostrar lista atual
- 2 - Adicionar elemento (append)
- 3 - Inserir elemento em posição específica
- 4 - Remover elemento
- 5 - Pesquisar elemento
- 6 - Ordenar lista (crescente)
- 7 - Ordenar lista (decrescente)
- 8 - Limpar lista
- 9 - Sair

=====

Escolha uma opção: 9

[DEBUG] Opção escolhida: 9

Saindo...

```
[ec2-user@ip-172-31-82-211 ~]$ client_loop: send disconnect: Connection reset
```