



**Instituto Politécnico Nacional**

Escuela Superior de Cómputo



Desarrollo de Sistemas Distribuidos

Prof. **Benjamín Cruz Torres**

## **Práctica No. 7**

### **Comunicación**

Grupo: 4AM1

Equipo: NetPower

- Integrantes:
1. Alcibar Zubillaga Julian
  2. De Luna Ocampo Yanina
  3. Salinas Velazquez Jacob

*Fecha: 03 de abril del 2022*

## PRÁCTICA 3: COMUNICACIÓN CON SOCKETS

**OBJETIVO DE LA PRÁCTICA:** Desarrollar un ejemplo de comunicación cliente-servidor, usando sockets.

### ESCENARIO

El primer paso para desarrollar un sistema distribuido es interconectar los equipos y permitir que haya comunicación entre ellos. Hay varios esquemas entre los que se puede dar la comunicación entre sistemas de cómputo. Una de las más comunes, y que se viene utilizando desde el inicio de las redes de computadoras, es el paso de mensajes.

### RECURSOS NECESARIO PARA REALIZAR LA PRÁCTICA

1. Computadora (Linux)
2. Intérprete de Python u otro lenguaje de programación.
3. Programas cliente y servidor de ejemplo

## INTRODUCCIÓN

### ¿Qué es un sistema distribuido?

Un sistema distribuido es un conjunto de equipos independientes que actúan de forma transparente actuando como un único equipo.

Su objetivo es descentralizar tanto el almacenamiento de la información como el procesamiento.

Esto lo vemos de cara al uso de Elasticsearch y cómo funciona.

### ¿Qué ventajas aporta?

Algunas de las ventajas que aporta un sistema distribuido son:

- Mayor eficacia.
- Mayor tolerancia a fallos: al estar distribuida la información en nodos, en caso de que se caiga un nodo, dicha información va a encontrarse replicada en otros nodos.
- Mayor velocidad y procesamiento distribuido: cuando se realiza una consulta, los procesamiento se dividen entre todos los nodos que forman el sistema distribuido, en lugar de enviarlos a un único nodo y que el mismo tenga que hacer todo el trabajo.
- Escalabilidad: si, por ejemplo, se necesita más procesamiento o añadir más disco duro, en lugar de que los equipos crezcan de forma vertical añadiendo más almacenamiento, RAM o CPU, se añaden equipos de forma horizontal al clúster o sistema distribuido.

## CONSIDERACIONES INICIALES

### SISTEMA OPERATIVO

Se recomienda realizar esta práctica en la nube (usando Microsoft Azure). Aunque, también es válida si se corre en la computadora local del usuario.

Es recomendable utilizar el sistema operativo Linux, aunque es posible correr en Windows o MacOS.

## DESARROLLO DE LA PRÁCTICA

### PAETE 1: DESCARGAR LOS ARCHIVOS NECESARIOS PARA REALIZAR LA PRÁCTICA 📁

Descarga los archivos ServSocket.py y CliSocket.py de la plataforma Teams

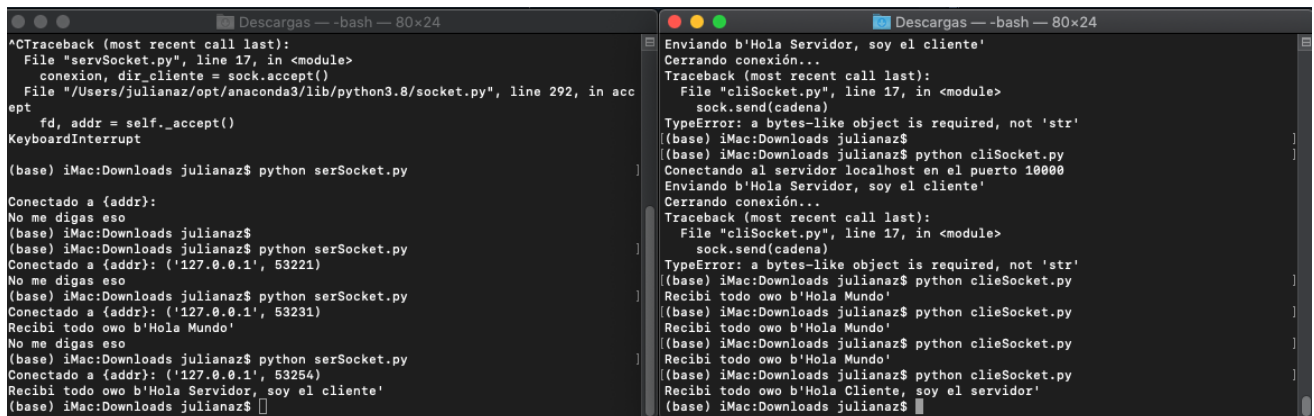
Antes de correrlos, revísalos y analízalos

### PARTE 2: EJECUCIÓN DE LOS PROGRAMAS

Ejecuta el programa ServSocket.py utilizando un intérprete de Python

Ejecuta el programa CliSocket.py utilizando un intérprete de Python

## 1. CAPTURA DE PANTALLA DE ALCIBAR ZUBILLAGA JULIÁN

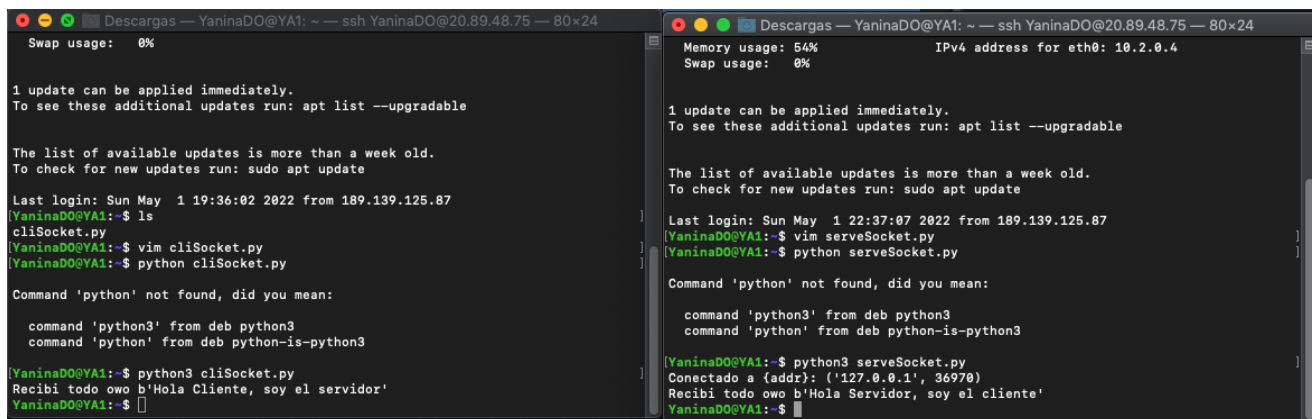


```
^CTraceback (most recent call last):
  File "servSocket.py", line 17, in <module>
    conexion, dir_cliente = sock.accept()
  File "/Users/julianaz/opt/anaconda3/lib/python3.8/socket.py", line 292, in accept
    fd, addr = self._accept()
KeyboardInterrupt

(base) iMac:Downloads julianaz$ python serSocket.py
Conectado a {addr}:
No me digas eso
(base) iMac:Downloads julianaz$
(base) iMac:Downloads julianaz$ python serSocket.py
Conectado a {addr}: ('127.0.0.1', 53221)
No me digas eso
(base) iMac:Downloads julianaz$ python serSocket.py
Conectado a {addr}: ('127.0.0.1', 53231)
Recibi todo owo b'Hola Mundo'
No me digas eso
(base) iMac:Downloads julianaz$ python serSocket.py
Conectado a {addr}: ('127.0.0.1', 53254)
Recibi todo owo b'Hola Servidor, soy el cliente'
(base) iMac:Downloads julianaz$

Enviando b'Hola Servidor, soy el cliente'
Cerrando conexión...
Traceback (most recent call last):
  File "cliSocket.py", line 17, in <module>
    sock.send(cadena)
TypeError: a bytes-like object is required, not 'str'
(base) iMac:Downloads julianaz$
(base) iMac:Downloads julianaz$ python cliSocket.py
Conectando al servidor localhost en el puerto 10000
Enviando b'Hola Servidor, soy el cliente'
Cerrando conexión...
Traceback (most recent call last):
  File "cliSocket.py", line 17, in <module>
    sock.send(cadena)
TypeError: a bytes-like object is required, not 'str'
(base) iMac:Downloads julianaz$ python cliSocket.py
Recibi todo owo b'Hola Mundo'
(base) iMac:Downloads julianaz$ python cliSocket.py
Recibi todo owo b'Hola Mundo'
(base) iMac:Downloads julianaz$ python cliSocket.py
Recibi todo owo b'Hola Mundo'
(base) iMac:Downloads julianaz$ python cliSocket.py
Recibi todo owo b'Hola Cliente, soy el servidor'
(base) iMac:Downloads julianaz$
```

## 2. CAPTURA DE PANTALLA DE DE LUNA OCAMPO YANINA



```
Swap usage: 0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun May 1 19:36:02 2022 from 189.139.125.87
YaninaDO@YA1:~$ ls
cliSocket.py
YaninaDO@YA1:~$ vim cliSocket.py
YaninaDO@YA1:~$ python cliSocket.py

Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3

YaninaDO@YA1:~$ python3 cliSocket.py
Recibi todo owo b'Hola Cliente, soy el servidor'
YaninaDO@YA1:~$

Memory usage: 54%          IPv4 address for eth0: 10.2.0.4
Swap usage: 0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

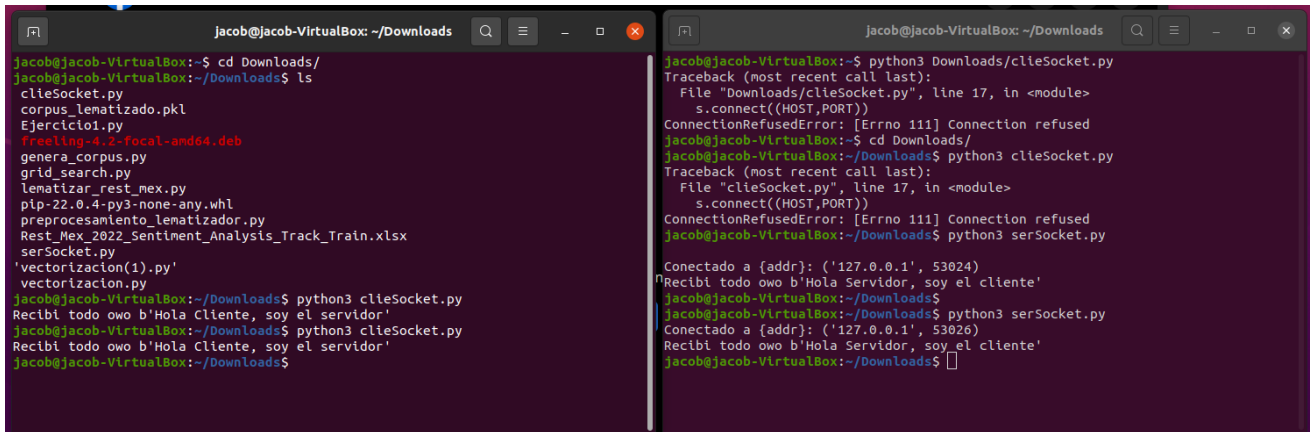
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun May 1 22:37:07 2022 from 189.139.125.87
YaninaDO@YA1:~$ vim serveSocket.py
YaninaDO@YA1:~$ python serveSocket.py

Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3

YaninaDO@YA1:~$ python3 serveSocket.py
Conectado a {addr}: ('127.0.0.1', 36970)
Recibi todo owo b'Hola Servidor, soy el cliente'
YaninaDO@YA1:~$
```

### 3. CAPTURA DE PANTALLA DE SALINAS VELÁZQUEZ JACOB



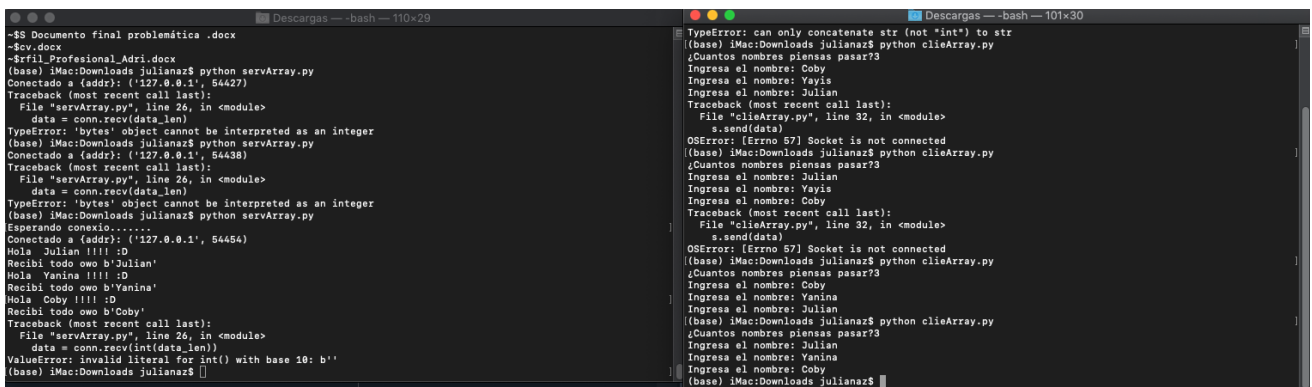
```
jacob@jacob-VirtualBox: ~/Downloads
jacob@jacob-VirtualBox:~$ cd Downloads/
jacob@jacob-VirtualBox:~/Downloads$ ls
cliSocket.py
corpus_lematizado.pkl
Ejercicio1.py
freeling-4.2-focal-amd64.deb
genera_corpus.py
grid_search.py
lematizar_rest_mex.py
plp-22.0.4-py3-none-any.whl
preprocesamiento_lematizador.py
Rest_Mex_2022_Sentiment_Analysis_Track_Train.xlsx
serSocket.py
'vectorizacion(1).py'
vectorizacion.py
jacob@jacob-VirtualBox:~/Downloads$ python3 cliSocket.py
Recibi todo owo b'Hola Cliente, soy el servidor'
jacob@jacob-VirtualBox:~/Downloads$ python3 cliSocket.py
Recibi todo owo b'Hola Cliente, soy el servidor'
jacob@jacob-VirtualBox:~/Downloads$

jacob@jacob-VirtualBox:~$ python3 Downloads/cliSocket.py
Traceback (most recent call last):
  File "Downloads/cliSocket.py", line 17, in <module>
    s.connect((HOST,PORT))
ConnectionRefusedError: [Errno 111] Connection refused
jacob@jacob-VirtualBox:~/Downloads$ python3 cliSocket.py
Traceback (most recent call last):
  File "cliSocket.py", line 17, in <module>
    s.connect((HOST,PORT))
ConnectionRefusedError: [Errno 111] Connection refused
jacob@jacob-VirtualBox:~/Downloads$ python3 serSocket.py
Conectado a {addr: ('127.0.0.1', 53024)}
Recibi todo owo b'Hola Servidor, soy el cliente'
jacob@jacob-VirtualBox:~/Downloads$ python3 serSocket.py
Conectado a {addr: ('127.0.0.1', 53026)}
Recibi todo owo b'Hola Servidor, soy el cliente'
jacob@jacob-VirtualBox:~/Downloads$
```

#### EJERCICIO ENVÍO DE CADENAS:

Modifica los programas para que el programa cliente pida al usuario varias cadenas (con los nombres completos de los integrantes del equipo), que serán enviadas al servidor. El servidor deberá leerlas y responder en consola “Hola” junto con cada uno de los nombres enviados por el cliente.

### 1. CAPTURA DE PANTALLA DE ALCIBAR ZUBILLAGA JULIÁN



```
-$ Documento final problematica .docx
-$ev.docx
-$rfil_Profesional_Adri.docx
(base) iMac:Downloads julianaz$ python servArray.py
Conectado a {addr: ('127.0.0.1', 54427)}
Traceback (most recent call last):
  File "servArray.py", line 26, in <module>
    data = conn.recv(data_len)
TypeError: 'bytes' object cannot be interpreted as an integer
(base) iMac:Downloads julianaz$ python servArray.py
Conectado a {addr: ('127.0.0.1', 54438)}
Traceback (most recent call last):
  File "servArray.py", line 26, in <module>
    data = conn.recv(data_len)
TypeError: 'bytes' object cannot be interpreted as an integer
(base) iMac:Downloads julianaz$ python servArray.py
Esperando conexio.....
Conectado a {addr: ('127.0.0.1', 54454)}
Hola Julian !!!! :D
Recibi todo owo b'Julian'
Hola Vanina !!!! :D
Recibi todo owo b'Vanina'
Hola Coby !!!! :D
Recibi todo owo b'Coby'
Traceback (most recent call last):
  File "servArray.py", line 26, in <module>
    data = conn.recv(int(data_len))
ValueError: invalid literal for int() with base 10: b''
(base) iMac:Downloads julianaz$

(base) iMac:Downloads julianaz$ python cliArray.py
¿Cuántos nombres piensas pasar?3
Ingresar el nombre: Coby
Ingresar el nombre: Yaviz
Ingresar el nombre: Julian
Traceback (most recent call last):
  File "cliArray.py", line 32, in <module>
    s.send(data)
OSError: [Errno 57] Socket is not connected
(base) iMac:Downloads julianaz$ python cliArray.py
¿Cuántos nombres piensas pasar?3
Ingresar el nombre: Julian
Ingresar el nombre: Yaviz
Ingresar el nombre: Coby
Traceback (most recent call last):
  File "cliArray.py", line 32, in <module>
    s.send(data)
OSError: [Errno 57] Socket is not connected
(base) iMac:Downloads julianaz$ python cliArray.py
¿Cuántos nombres piensas pasar?3
Ingresar el nombre: Julian
Ingresar el nombre: Vanina
Ingresar el nombre: Coby
(base) iMac:Downloads julianaz$
```

### 2. CAPTURA DE PANTALLA DE DE LUNA OCAMPO YANINA

```
YaninaDO@YA1:~$ vim servArray.py
YaninaDO@YA1:~$ python servArray.py
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
YaninaDO@YA1:~$ python3 servArray.py
Esperando conexio.....
Conectado a (addr): ('127.0.0.1', 56286)
Hola Yanina !!!! :D
Recibi todo owo b'Yanina'
Hola Juli !!!! :D
Recibi todo owo b'Juli'
Hola Coby !!!! :D
Recibi todo owo b'Coby'
Traceback (most recent call last):
  File "servArray.py", line 26, in <module>
    data = conn.recv(int(data_len))
ValueError: invalid literal for int() with base 10: b''
YaninaDO@YA1:~$

YaninaDO@YA1:~$ vim clieArray.py
YaninaDO@YA1:~$ python3 clieArray.py
¿Cuantos nombres piensas pasar?3
Ingresa el nombre: Yanina
Ingresa el nombre: Juli
Ingresa el nombre: Coby
YaninaDO@YA1:~$
```

### 3. CAPTURA DE PANTALLA DE SALINAS VELÁZQUEZ JACOB

```
jacob@jacob-VirtualBox: ~/Downloads
jacob@jacob-VirtualBox:~/Downloads$ python3 servArray.py
Esperando conexio.....
Conectado a (addr): ('127.0.0.1', 58098)
Hola Julian !!!! :D
Recibi todo owo b'Julian'
Hola Yanina !!!! :D
Recibi todo owo b'Yanina'
Hola Jacob !!!! :D
Recibi todo owo b'Jacob'
Traceback (most recent call last):
  File "servArray.py", line 26, in <module>
    data = conn.recv(int(data_len))
ValueError: invalid literal for int() with base 10: b''
jacob@jacob-VirtualBox:~/Downloads$

jacob@jacob-VirtualBox:~/Downloads$ python3 clieArray.py
¿Cuantos nombres piensas pasar?3
Ingresa el nombre: Julian
Ingresa el nombre: Yanina
Ingresa el nombre: Jacob
jacob@jacob-VirtualBox:~/Downloads$
```

#### CUESTIONARIO:

Responde las siguientes preguntas

¿QUÉ TENDRÍAS QUE MODIFICAR AL (LOS) PROGRAMA(S) PARA ENVIAR LAS CADENAS MEDIANTE UDP, CON DATAGRAMAS?

*Recoger la información correspondiente de control y de esta manera permitir que los datos se transmitan.*

YA QUE UDP NO PROVEE SEGURIDAD O CONFIABILIDAD. ¿CÓMO LAS IMPLEMENTARÍAS?

*La utilización de DTLS para tratar de minimizar la falta de seguridad a la hora de usar el UDP*

#### CONCLUSIONES

Esta práctica hizo que pudiéramos comprender de una mejor forma la implementación de un sistema distribuido para la interconexión de equipos y poder entender el la forma de comportarse de estos programas de manera completamente práctica, junto con el uso de conocimientos previos dentro de la materia.

Podemos concluir que el uso de los sistemas distribuidos tienen muchas ventajas a la hora de su uso ya que nos ayudan a desarrollar diferentes actividades de una manera más eficaz y con un mejor aprovechamiento de los recursos, su única inconveniencia que se vio dentro del desarrollo de esta práctica fue que se demostró que el uso de UDP no provee de

seguridad o de confiabilidad haciendo que en un uso más profesional existen muchos problemas haciendo que se busquen alternativas a su uso.

## BIBLIOGRAFÍA

1. Losada, S. (2020, 8 julio). *Qué es un sistema distribuido y qué ventajas aporta su funcionamiento*. OpenWebinars.net. <https://openwebinars.net/blog/que-es-un-sistema-distribuido/>
2. Atlassian. (2019, 5 junio). *¿Qué es un sistema distribuido?* <https://www.atlassian.com/es/microservices/microservices-architecture/distributed-architecture>

## CONSIDERACIONES FINALES

Descarga el documento y léelo antes de llenarlo.

Este documento se debe llenar en equipo, aunque la práctica la deben hacer TODOS los integrantes de este.

Después de llenar el documento, guárdalo como PDF y envíalo a través del tema correspondiente en la plataforma de aprendizaje virtual correspondiente.

Queda estrictamente prohibido cualquier tipo de plagio a otros equipos o grupos de este semestre o anteriores. En caso de incurrir en esta falta, se anulará la asignación correspondiente y se bajarán 2 puntos al (los) equipo (s) involucrados.