



# Universidad Nacional de Lanús

## Departamento de Ciencia y Tecnología

Carrera: Licenciatura en Sistemas

---

## Asignatura: Redes y Comunicaciones

### Trabajo Práctico: Sockets

#### Grupo W

#### Integrantes:

- *Franco Joaquín Luzardo Bermúdez*
- *Brian Muñoz*
- *Rocío Trinidad*
- *Brian Ávila*
- *Gabriel Omar Romero*

# Pruebas realizadas:

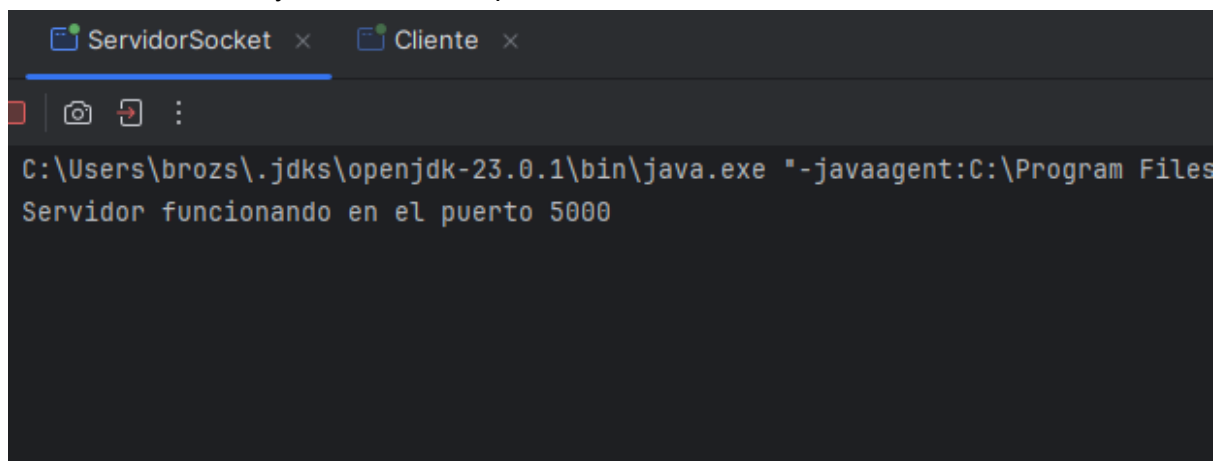
Este documento presenta las pruebas realizadas para verificar el correcto funcionamiento de la aplicación cliente-servidor desarrollada. El sistema cuenta con un servidor y dos clientes (uno en Java y otro en Python). Las funcionalidades probadas incluyen la generación de nombres de usuario, generación de correos electrónicos y validación de entradas inválidas.

Código fuente: <https://github.com/BrianMF524/ServerSocketJava>

## Conexión: servidor-cliente1 y servidor-cliente2.

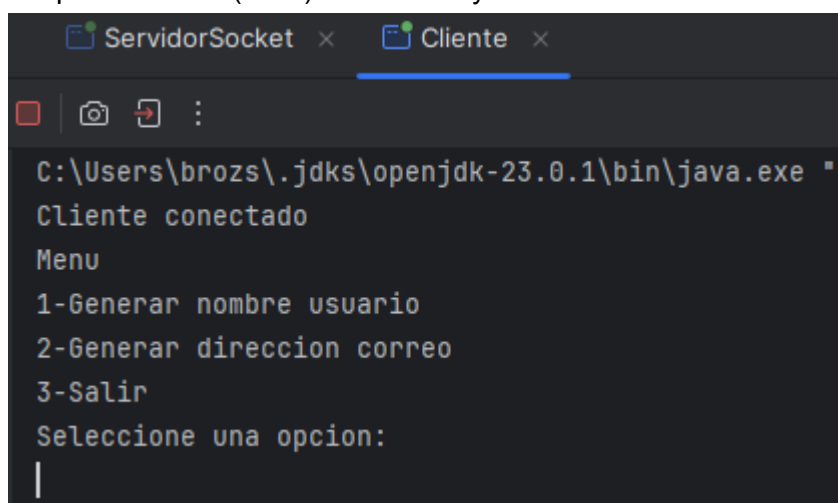
Se inicia el servidor y luego se ejecutan los clientes.

-Servidor encendido y escuchando el puerto 5000



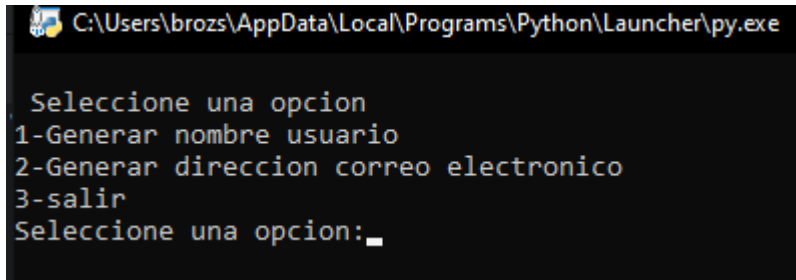
```
ServidorSocket x Cliente x
C:\Users\brozs\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files
Servidor funcionando en el puerto 5000
```

- El primer cliente(Java) se conecta y muestra el menu



```
ServidorSocket x Cliente x
C:\Users\brozs\.jdk\openjdk-23.0.1\bin\java.exe "-
Cliente conectado
Menu
1-Generar nombre usuario
2-Generar direccion correo
3-Salir
Seleccione una opcion:
|
```

-Se conecta el segundo cliente (Python)

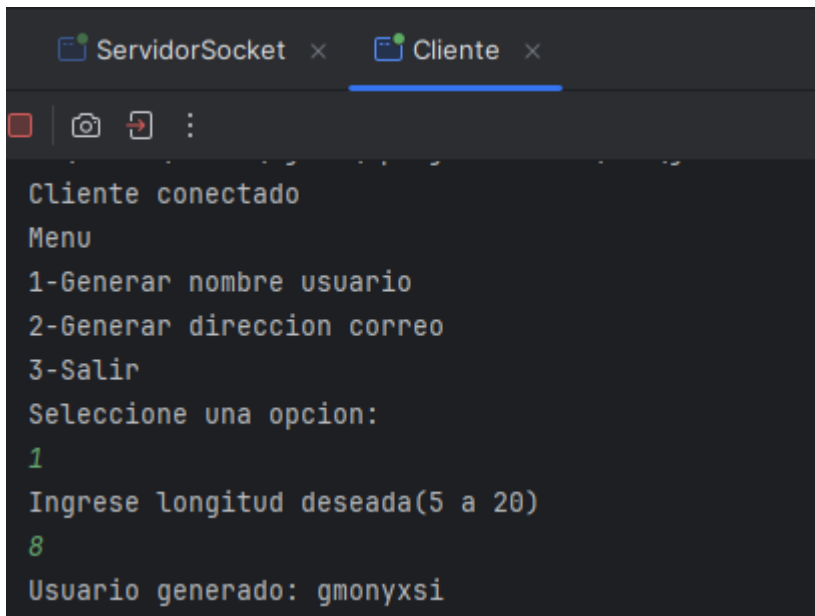


```
C:\Users\brozs\AppData\Local\Programs\Python\Launcher\py.exe

Seleccione una opcion
1-Generar nombre usuario
2-Generar direccion correo electronico
3-salir
Seleccione una opcion: _
```

## Prueba: Generación de nombre de usuario en cliente 1

El usuario selecciona la opción 1, ingresa una longitud válida (por ejemplo, 8), y el servidor responde con un nombre de usuario generado aleatoriamente.

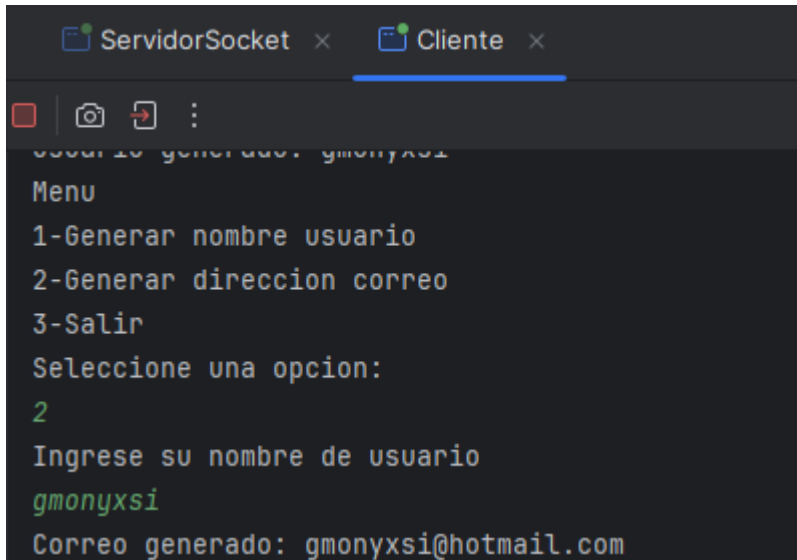


```
ServidorSocket x Cliente x

Cliente conectado
Menu
1-Generar nombre usuario
2-Generar direccion correo
3-Salir
Seleccione una opcion:
1
Ingrese longitud deseada(5 a 20)
8
Usuario generado: gmonyxsi
```

## Prueba: Generación de correo electrónico en cliente 1

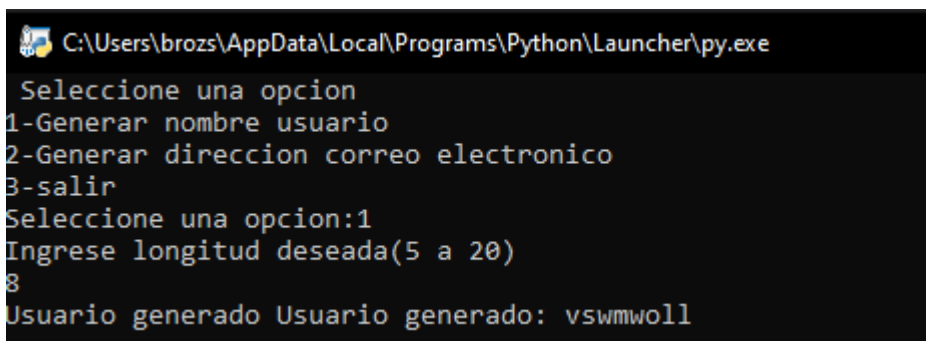
El usuario selecciona la opción 2, ingresa su nombre, y el servidor responde con un correo generado.



```
ServidorSocket x Cliente x
Menu
1-Generar nombre usuario
2-Generar direccion correo
3-Salir
Seleccione una opcion:
2
Ingrese su nombre de usuario
gmonyxsi
Correo generado: gmonyxsi@hotmail.com
```

## Prueba: Generación de nombre de usuario en cliente 2

El cliente 2 (Python) solicita la generación de un nombre de usuario. El servidor procesa la solicitud y responde correctamente.



```
C:\Users\brozs\AppData\Local\Programs\Python\Launcher\py.exe
Seleccione una opcion
1-Generar nombre usuario
2-Generar direccion correo electronico
3-salir
Seleccione una opcion:1
Ingrese longitud deseada(5 a 20)
8
Usuario generado Usuario generado: vswmwoll
```

## Prueba: Generación de correo electrónico en cliente 2

El cliente 2 solicita generar un correo ingresando su nombre de usuario. El servidor responde con el correo correspondiente.

```
Símbolo del sistema - python main.py

Seleccione una opcion
1-Generar nombre usuario
2-Generar direccion correo electronico
3-salir
Seleccione una opcion:2
Ingrese su nombre de usuario
vswmwoll
Correo generado: Correo generado: vswmwoll@hotmail.com
```

## Prueba de error: Entrada inválida

Se prueba un caso con longitud inválida para el nombre de usuario (por ejemplo, 3). El servidor responde con el mensaje de error correspondiente.

```
Run  ServidorSocket x Cliente x

Menu
↑ 1-Generar nombre usuario
↓ 2-Generar direccion correo
⇐ 3-Salir
⇌ Seleccione una opcion:
1
Ingrese longitud deseada(5 a 20)
3
Longitud invalida
```