Universidad Nacional de Lanús

Materia:	**Desarrollo de software en Sistemas Distribuidos**
Proyecto:	**Actividad Sockets**
***Responsable del Catedra*:**	**Prof. Diego Azcurra** **Prof. Marcos Amaro**
Responsables del Trabajo Práctico:	**Erika Valdez**
Fecha aprobación documento:	**1 de Septiembre de 2024**

Índice

1-Estrategia de Resolución del Trabajo Práctico

La resolución de este trabajo práctico se estructuró en torno a la implementación de un sistema cliente-servidor que permite la generación de nombres de usuario y contraseñas a través de solicitudes realizadas por los clientes al servidor. El enfoque principal fue crear una arquitectura que permita demostrar el uso de sockets para la comunicación en red, así como la implementación de lógica de generación de datos en un servidor.

Estructura del Proyecto

El proyecto se divide en dos partes principales:

1. **Servidor**: Implementado en lenguaje C, el servidor se encarga de recibir las solicitudes de los clientes y responder con un nombre de usuario o contraseña generados aleatoriamente. El servidor utiliza la biblioteca Winsock2 para manejar la comunicación por sockets, lo que le permite operar en un entorno de red TCP/IP. El servidor escucha en un puerto específico (8080) y, una vez que recibe una conexión de un cliente, procesa la solicitud y envía una respuesta adecuada.
2. **Clientes**: Se desarrollaron dos clientes, uno en lenguaje C y otro en C# .NET, para demostrar la interoperabilidad entre distintos lenguajes y plataformas. Ambos clientes se conectan al servidor, envían solicitudes que especifican el tipo de dato requerido (nombre de usuario o contraseña) y la longitud del mismo, y finalmente reciben y muestran la respuesta generada por el servidor.

2-Implementación

- **Generación de Datos**: El servidor incluye funciones específicas para generar nombres de usuario y contraseñas. Para los nombres de usuario, se combinan vocales y consonantes para crear una cadena que cumpla con la longitud solicitada. Las contraseñas se generan utilizando un conjunto de caracteres alfanuméricos.
- **Manejo de Errores**: Se verifica que las solicitudes de los clientes cumplan con ciertos criterios, como la longitud mínima y máxima de los nombres de usuario y contraseñas. En caso de recibir una solicitud incorrecta, el servidor responde con un mensaje de error.
- **Pruebas y Validación**: Se realizaron pruebas manuales para validar el correcto funcionamiento del sistema. Se ejecutaron escenarios en los que se conectaban múltiples clientes al servidor, verificando que todos recibieran las respuestas esperadas.

3-Consideraciones Técnicas

- **Interoperabilidad**: Uno de los objetivos clave fue asegurar que los clientes desarrollados en diferentes lenguajes pudieran interactuar sin problemas con el servidor. Esto se logró utilizando el protocolo TCP/IP estándar y asegurando que el formato de las solicitudes y respuestas fuera compatible entre las implementaciones.

4-Conclusión

La estructura modular y el uso de tecnologías estándar como C, .NET, y Winsock2 permiten que este sistema sea extensible y adaptable a diferentes necesidades. La implementación exitosa de este trabajo práctico no solo cumple con los requisitos establecidos, sino que también proporciona una base sólida para futuros desarrollos en sistemas distribuidos.

5-- Empezar

Estas instrucciones te permitirán obtener una copia del proyecto en tu máquina local para desarrollo y pruebas.

5.1Prerequisites

- Compilador C (como GCC)
- Visual Studio (para el cliente en C# .NET)
- Winsock2 (para la funcionalidad de red en C)
- Codeblocks

5.2 Instalacion

1. Clonar el repositorio:

git
clone https://github.com/tuusuario/proyecto.git](https://github.com/ErikaValdez/TP-ActividadSockets.git)

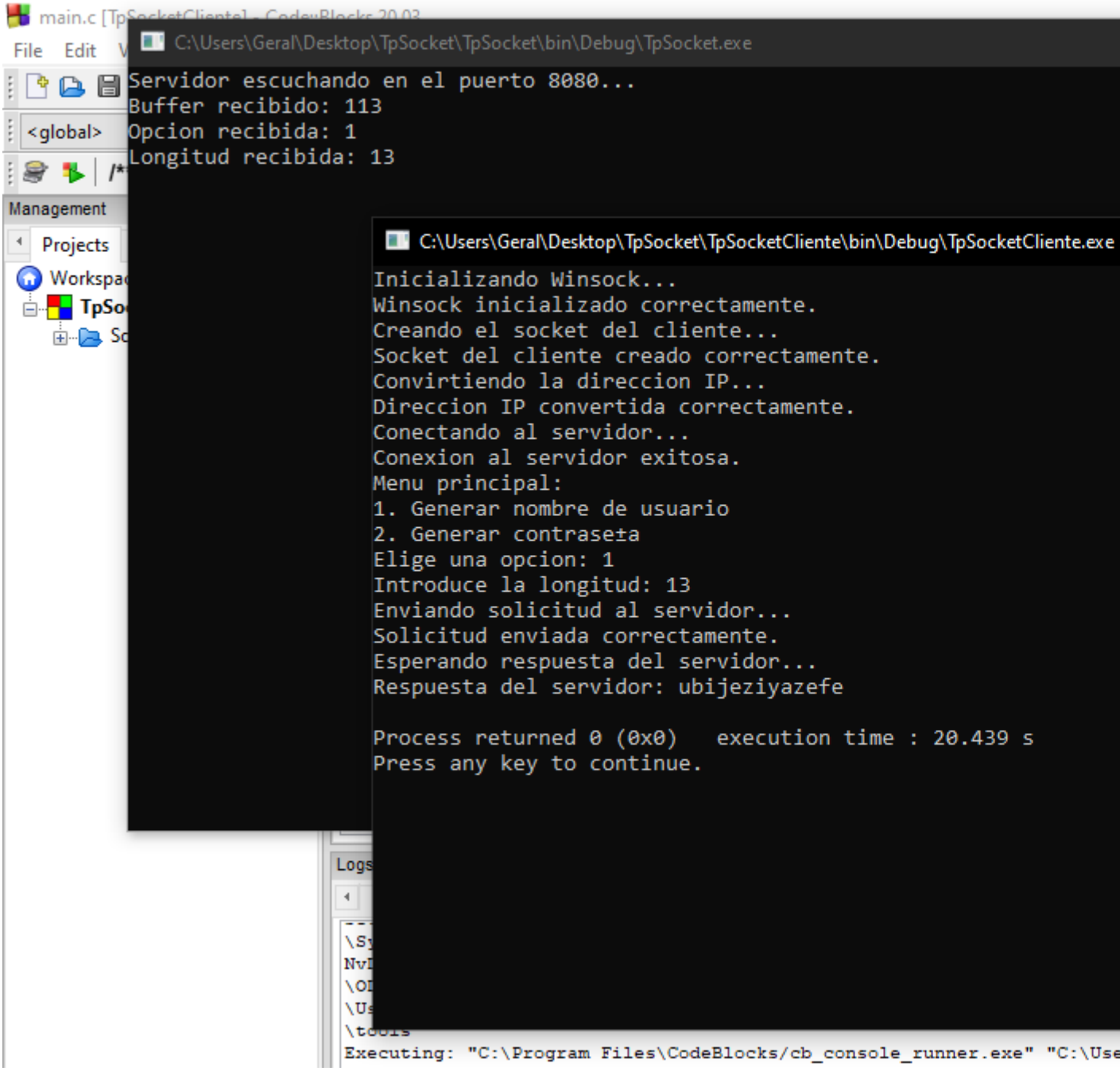
2. Compilar el Servidor

gcc servidor.c -o servidor -lws2_32

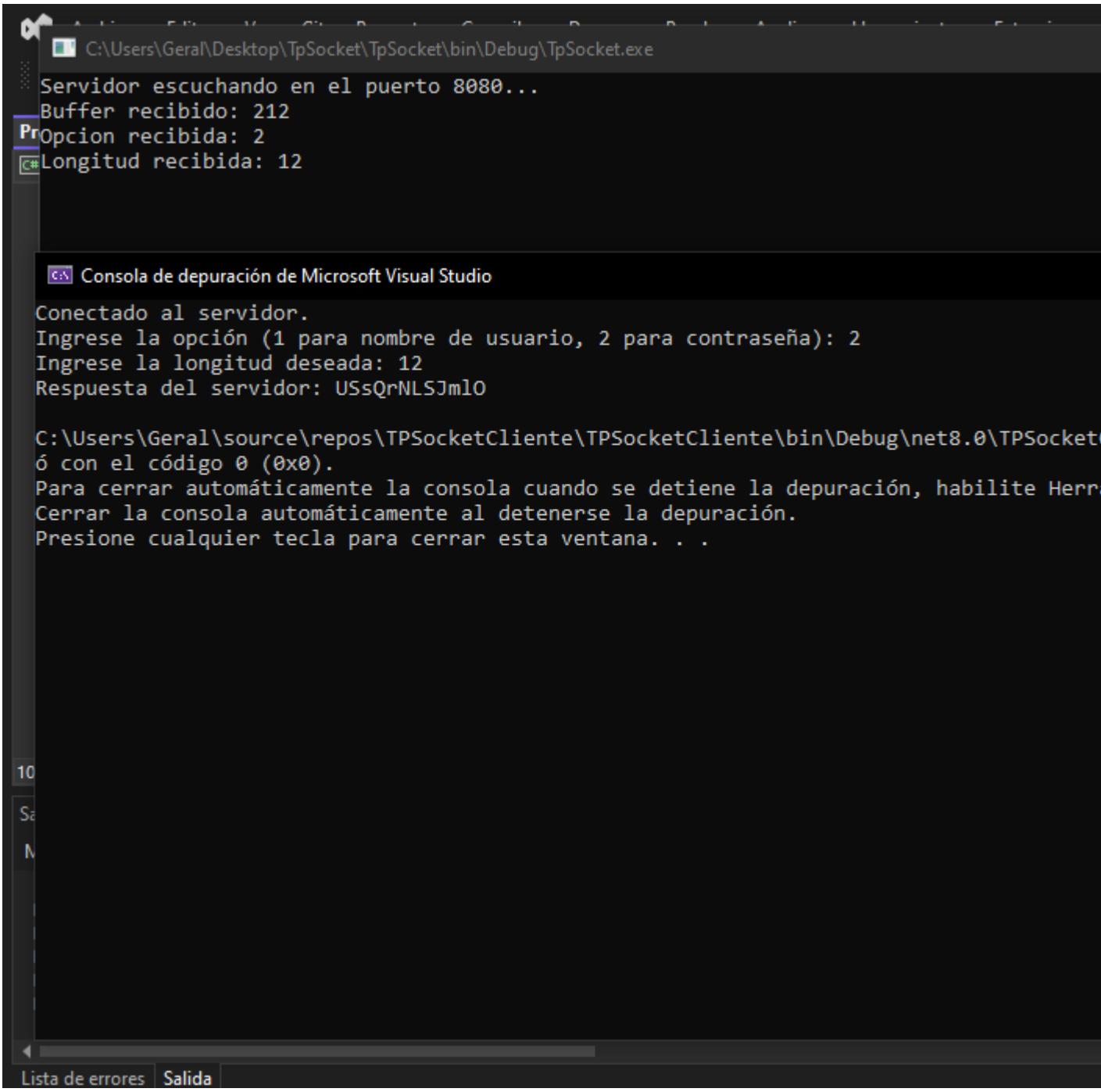
3. Configurar Visual Studio para el cliente en C# .NET.

5-Ejecucion de las pruebas

Se puede probar manualmente el sistema iniciando el servidor y conectando los clientes para verificar su
funcionalidad.



"Cliente-Servidor



6-Despliegue

Para desplegar el servidor en un entorno en vivo, asegúrate de que el puerto 8080 esté disponible y que Winsock2 esté correctamente configurado en el sistema.

7-Construido

Para el servidor y cliente. .NET - Para el cliente en C#. Winsock2 - Para la funcionalidad de red.