



Universidad Nacional de Lanús

Departamento de Desarrollo Productivo y Tecnológico

Carrera: Licenciatura en Sistemas

Materia: Desarrollo de Software en Sistemas Distribuidos (8627-1)

Trabajo Practico N°1: “Actividad - Sockets”

Docentes: Ing. Diego Andrés Azcurra / Lic. Marcos Amaro

Estudiante: Cristian Alexander Santander

DNI: 37352564

Fecha: 24/08/2024

Repositorio de Github:

<https://github.com/csantander93/socket-generador-de-usuarios-contrasenas.git>

Introducción

En el contexto de la asignatura Desarrollo de Software en Sistemas Distribuidos, se nos planteó la tarea de desarrollar un sistema basado en "sockets", compuesto por un servidor y dos clientes. El desafío consistía en implementar uno de los clientes en el mismo lenguaje de programación que el servidor, mientras que el segundo cliente debía ser desarrollado en un lenguaje diferente. Ambos clientes debían ser capaces de establecer una conexión con el servidor para generar un nombre de usuario y/o una contraseña, siguiendo ciertos criterios de longitud y validaciones específicas.

El sistema desarrollado presenta un servidor implementado en lenguaje C, que utiliza sockets para comunicarse con los dos clientes. Este servidor ofrece dos funcionalidades principales:

- Generador de nombres de usuario: Crea cadenas de texto alternando entre vocales y consonantes, con una longitud configurable.
- Generador de contraseñas: Produce cadenas alfanuméricas que incluyen mayúsculas, minúsculas y dígitos, también con una longitud configurable.

Para interactuar con el servidor, se desarrollaron dos clientes. El primer cliente fue implementado en C y permite al usuario elegir entre generar un nombre de usuario o una contraseña, especificando la longitud deseada mediante un menú de opciones. El segundo cliente fue desarrollado en JavaScript utilizando Node.js, y ofrece la misma funcionalidad, conectándose al servidor para realizar las solicitudes.

Estrategia de Resolución del Trabajo Práctico

La estrategia de resolución de este trabajo práctico se centró en el desarrollo de un sistema cliente-servidor utilizando sockets, con el objetivo de ofrecer un servicio de generación de nombres de usuario y contraseñas. A continuación, se describe cómo se estructuró la aplicación y se detallan los aspectos más significativos de la implementación.

Estructura General del Sistema

El sistema se compone de tres elementos principales:

1. Servidor en C: Es el núcleo del sistema, responsable de recibir las solicitudes de los clientes y generar las respuestas correspondientes. El servidor se encarga de gestionar las conexiones, procesar las opciones seleccionadas por el cliente, y enviar los resultados generados (nombres de usuario o contraseñas) de vuelta a los clientes.

2. Cliente en C: Este cliente se comunica directamente con el servidor, permitiendo al usuario seleccionar entre las dos funcionalidades disponibles: opción 1 (generar nombre de usuario), opción 2 (generar contraseña), y opción 0 (salir o cerrar conexión). Este cliente fue desarrollado en el mismo lenguaje que el servidor.
3. Cliente en JavaScript: Este cliente, desarrollado en Node.js, demuestra la capacidad del sistema para interactuar con clientes en un lenguaje diferente al del servidor. Al igual que el cliente en C, permite al usuario interactuar con el servidor para generar nombres de usuario y contraseñas.

Desarrollo del Servidor

El servidor fue implementado en lenguaje C y diseñado para manejar una conexión de cliente a la vez utilizando la biblioteca de sockets de Windows (winsock2.h). La lógica del servidor se estructuró en las siguientes funciones clave:

- `iniciarServidor()`: Inicializa el servidor, crea un socket de escucha, y espera conexiones entrantes de los clientes.
- `manejarConexion(SOCKET new_socket)`: Gestiona la interacción con un cliente específico. Esta función recibe la opción y la longitud solicitadas, genera el nombre de usuario o contraseña correspondiente, y envía el resultado al cliente.
- `generarNombreUsuario(int longitud, char *resultado)`: Genera un nombre de usuario alternando vocales y consonantes.
- `generarContraseña(int longitud, char *resultado)`: Genera una contraseña alfanumérica segura.

El servidor mantiene su operación en un bucle, permitiendo la conexión continua de nuevos clientes, pero solo admite una conexión activa a la vez. Una vez que un cliente se desconecta, el servidor está disponible para aceptar la conexión de un nuevo cliente.

Desarrollo del Cliente en C

El cliente en C fue diseñado para ofrecer una interfaz sencilla al usuario, permitiéndole seleccionar las opciones a través de un menú:

- `iniciarCliente()`: Establece la conexión con el servidor y gestiona la interacción con el usuario mediante un bucle que muestra el menú de opciones.
- `enviarOpcion(SOCKET sock, int opcion, int longitud)`: Envía al servidor la opción seleccionada y la longitud requerida para el nombre de usuario o la contraseña.

El cliente se mantiene en un bucle de interacción, permitiendo al usuario solicitar repetidamente nombres de usuario o contraseñas hasta que decida salir.

Desarrollo del Cliente en JavaScript

El cliente en JavaScript se desarrolló utilizando la biblioteca net de Node.js, permitiendo la comunicación mediante sockets con el servidor en C. Este cliente tiene funcionalidades equivalentes al cliente en C:

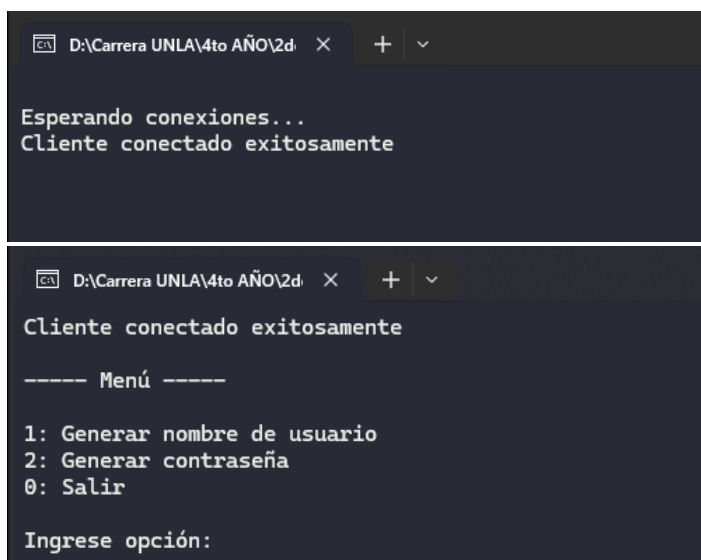
- **Conexión al Servidor:** El cliente se conecta al servidor utilizando la dirección IP y el puerto especificado.
- **Interacción con el Usuario:** Utiliza readline para recibir la entrada del usuario, permitiendo la selección de opciones y la especificación de la longitud.
- **Envío de Solicitudes y Recepción de Respuestas:** Tras enviar las opciones seleccionadas, el cliente espera y muestra la respuesta del servidor.

Casos de prueba

Para asegurar el correcto funcionamiento del sistema, se llevaron a cabo una serie de pruebas que incluyeron la verificación de la funcionalidad de cada componente, la compatibilidad entre el servidor y los clientes, y la validación de las opciones ofrecidas. Las pruebas se realizaron utilizando tanto el cliente en C como el cliente en JavaScript, verificando la capacidad del servidor para manejar las solicitudes de generación de nombres de usuario y contraseñas según los criterios especificados.

A continuación, se presentan las capturas de pantalla que ilustran los resultados de estas pruebas, mostrando la interacción entre el cliente y el servidor, así como las respuestas generadas y los mensajes de error en caso de entradas incorrectas. Estas capturas proporcionan una visión clara del comportamiento del sistema en diferentes escenarios y ayudan a validar su funcionamiento.

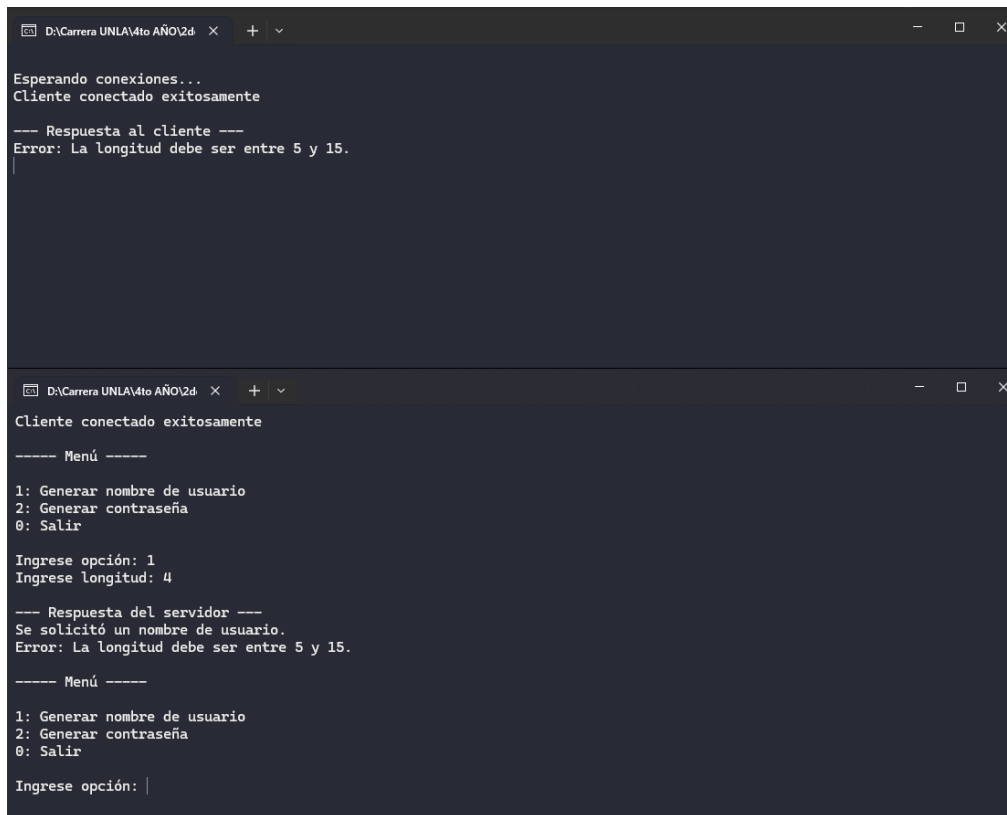
En esta captura, se ejecutan ambos programas: primero, servidor.exe y luego cliente.exe. El servidor comienza a escuchar conexiones entrantes, mientras que el cliente se conecta al servidor. La imagen muestra el momento en que el cliente y el servidor establecen la conexión inicial y están listos para intercambiar datos:



```
D:\Carrera UNLA\4to AÑO\2d. X + v
Esperando conexiones...
Cliente conectado exitosamente

D:\Carrera UNLA\4to AÑO\2d. X + v
Cliente conectado exitosamente
----- Menú -----
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción:
```

En esta captura, el cliente ha ingresado una longitud menor a 5 para la opción de generar un nombre de usuario (opción 1). El servidor, al detectar que la longitud no cumple con los requisitos mínimos, envía un mensaje de error al cliente. El mensaje de error indica que la longitud debe estar entre 5 y 15 caracteres. Tras enviar la respuesta de error, el servidor vuelve a escuchar nuevas solicitudes. El cliente recibe la respuesta del servidor y muestra nuevamente las opciones del menú, permitiendo al usuario intentar con una longitud válida o seleccionar otras opciones.



```
D:\Carrera UNLA\4to AÑO\2do >
Esperando conexiones...
Cliente conectado exitosamente
--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

D:\Carrera UNLA\4to AÑO\2do >
Cliente conectado exitosamente
----- Menú -----
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: 1
Ingrese longitud: 4
--- Respuesta del servidor ---
Se solicitó un nombre de usuario.
Error: La longitud debe ser entre 5 y 15.
----- Menú -----
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: |
```

Ingresando número mayor a 15 para la opción 1, respuesta del servidor:

```
D:\Carrera UNLA\4to AÑO\2d  X + v

Esperando conexiones...
Cliente conectado exitosamente

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

D:\Carrera UNLA\4to AÑO\2d  X + v

2: Generar contraseña
0: Salir

Ingrese opción: 1
Ingrese longitud: 4

--- Respuesta del servidor ---
Se solicitó un nombre de usuario.
Error: La longitud debe ser entre 5 y 15.

----- Menú -----
1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingrese opción: 1
Ingrese longitud: 16

--- Respuesta del servidor ---
Se solicitó un nombre de usuario.
Error: La longitud debe ser entre 5 y 15.

----- Menú -----
1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingrese opción:
```

En esta captura, el cliente ha ingresado una longitud válida para la opción de generar un nombre de usuario (opción 1). El servidor, al procesar la solicitud con la longitud proporcionada, genera el nombre de usuario "irawuqoy" y lo envía de vuelta al cliente:

```
D:\Carrera UNLA\4to AÑO\2d  X + v

Esperando conexiones...
Cliente conectado exitosamente

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.
Se ha generado un nombre de usuario.

--- Respuesta al cliente ---
irawuqoy
```

```
Ingresa opción: 1
Ingresa longitud: 8

--- Respuesta del servidor ---
Se solicitó un nombre de usuario.
irawuqoy

----- Menú -----

1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingresa opción: |
```

En esta captura, el cliente solicita la generación de una contraseña (opción 2). Primero, se ingresa una longitud menor a 8 caracteres, y el servidor responde con un mensaje de error indicando que la longitud debe ser de al menos 8 caracteres. Luego, se prueba con una longitud mayor a 50 caracteres, y el servidor responde con un mensaje de error que señala que la longitud no debe exceder los 50 caracteres.

```
D:\Carrera UNLA\4to AÑO\2d  X + v
Esperando conexiones...
Cliente conectado exitosamente

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.
Se ha generado un nombre de usuario.

--- Respuesta al cliente ---
irawuqoy

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.
|
```

```
D:\Carrera UNLA\4to AÑO\2d  X + v
2: Generar contraseña
0: Salir

Ingresa opción: 1
Ingresa longitud: 8

--- Respuesta del servidor ---
Se solicitó un nombre de usuario.
irawuqoy

----- Menú -----

1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingresa opción: 2
Ingresa longitud: 7

--- Respuesta del servidor ---
Se solicitó una contraseña.
Error: La longitud debe ser entre 8 y 50.

----- Menú -----

1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingresa opción: |
```

```
--- Respuesta al cliente ---
irawuqoy

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.
```

```
D:\Carrera UNLA\4to AÑO\2do
2: Generar contraseña
0: Salir

Ingrese opción: 2
Ingrese longitud: 7

--- Respuesta del servidor ---
Se solicitó una contraseña.
Error: La longitud debe ser entre 8 y 50.

----- Menú -----

1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingrese opción: 2
Ingrese longitud: 51

--- Respuesta del servidor ---
Se solicitó una contraseña.
Error: La longitud debe ser entre 8 y 50.

----- Menú -----

1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingrese opción: |
```

En esta captura, el cliente elige la opción 2 para generar una contraseña y proporciona una longitud válida de 50 caracteres. El servidor procesa la solicitud y responde con una contraseña generada que cumple con la longitud deseada. La respuesta del servidor se muestra al cliente, quien la recibe y puede visualizar la contraseña generada de acuerdo a la longitud deseada.

```
D:\Carrera UNLA\4to AÑO\2do
Esperando conexiones...
Cliente conectado exitosamente

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.
Se ha generado un nombre de usuario.

--- Respuesta al cliente ---
irawuqoy

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.
Se ha generado una contraseña.

--- Respuesta al cliente ---
Kb7FzP5tM1QBI6DSS92c31Apgjk9LVK7dmpdonxRwD2YvLzRhB

D:\Carrera UNLA\4to AÑO\2do
2: Generar contraseña
0: Salir

Ingrese opción: 2
Ingrese longitud: 51

--- Respuesta del servidor ---
Se solicitó una contraseña.
Error: La longitud debe ser entre 8 y 50.

----- Menú -----

1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingrese opción: 2
Ingrese longitud: 50

--- Respuesta del servidor ---
Se solicitó una contraseña.
Kb7FzP5tM1QBI6DSS92c31Apgjk9LVK7dmpdonxRwD2YvLzRhB

----- Menú -----

1: Generar nombre de usuario
2: Generar contraseña
0: Salir

Ingrese opción: |
```


Luego al salir se cierra la conexión del cliente, pero el servidor sigue funcionando:

```
D:\Carrera UNLA\4to AÑO\2d... X + v

Esperando conexiones...
Cliente conectado exitosamente

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.
Se ha generado un nombre de usuario.

--- Respuesta al cliente ---
irawuqoy

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.
Se ha generado una contraseña.

--- Respuesta al cliente ---
Kb7FzP5tM1QBI6DSS92c31Apgjk9LVK7dmpdonxRWd2YvLzRhbm
Error al recibir la opción o conexión cerrada: No error

Esperando conexiones...
```

El servidor queda a la escucha para seguir respondiendo por si hay una nueva conexión. Para este trabajo se pidió crear otro cliente, el mismo se realizó en JavaScript, como se ve a continuación el proceso fue el mismo:

```
D:\Carrera UNLA\4to AÑO\2d X + v
Cliente conectado exitosamente
--- Menú ---
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: 1
Ingrese longitud: 4

--- Respuesta del servidor ---
Error: La longitud debe ser entre 5 y 15.

--- Menú ---
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: 1
Ingrese longitud: 16

--- Respuesta del servidor ---
Error: La longitud debe ser entre 5 y 15.

--- Menú ---
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: 1
Ingrese longitud: 9

--- Respuesta del servidor ---
nasitenik

--- Menú ---
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: 2
Ingrese longitud: 7

--- Respuesta del servidor ---
Error: La longitud debe ser entre 8 y 50.

--- Menú ---
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: 2
Ingrese longitud: 19

--- Respuesta del servidor ---
sQ7sHeKcNBolLC0G8g8

--- Menú ---
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
Ingrese opción: 2
Ingrese longitud: 51

--- Respuesta del servidor ---
Error: La longitud debe ser entre 8 y 50.

--- Menú ---
1: Generar nombre de usuario
2: Generar contraseña
0: Salir
```

```
D:\Carrera UNLA\4to AÑO\2d X + v
Esperando conexiones...
Cliente conectado exitosamente

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.
Se ha generado un nombre de usuario.

--- Respuesta al cliente ---
irawuqoy

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.
Se ha generado una contraseña.

--- Respuesta al cliente ---
Kb7FzP5tM1QBI6DSS92c31Apgjk9LVK7dmpdonxRwD2YvlzRhb
Error al recibir la opción o conexión cerrada: No error

Esperando conexiones...
Cliente conectado exitosamente

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.

--- Respuesta al cliente ---
Error: La longitud debe ser entre 5 y 15.
Se ha generado un nombre de usuario.

--- Respuesta al cliente ---
nasitenik

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.
Se ha generado una contraseña.

--- Respuesta al cliente ---
sQ7sHeKcNBolLC0G8g8

--- Respuesta al cliente ---
Error: La longitud debe ser entre 8 y 50.
Error al recibir la opción o conexión cerrada: No error

Esperando conexiones...
```

Conclusión

El desarrollo del sistema de generación de nombres de usuario y contraseñas a través de sockets ha demostrado la eficacia de la comunicación entre un servidor en C y clientes en C y JavaScript. La implementación de la lógica del servidor, que incluye la generación de nombres de usuario y contraseñas con criterios específicos de longitud, y la creación de clientes en distintos lenguajes, ha sido exitosa.

El servidor, diseñado para manejar una conexión a la vez, cumple con los requisitos de validar las longitudes de los datos solicitados y proporcionar respuestas precisas. Los clientes, tanto en C como en JavaScript, permiten a los usuarios interactuar con el servidor mediante un menú intuitivo, enviando opciones y longitudes para obtener los datos deseados.

Las pruebas realizadas confirmaron el correcto funcionamiento del sistema, desde la conexión inicial hasta la generación y validación de nombres de usuario y contraseñas. Los desafíos de compatibilidad entre lenguajes y la correcta implementación de la lógica de generación se resolvieron adecuadamente, asegurando una experiencia de usuario consistente y fiable. Este proyecto demuestra una integración efectiva de tecnologías y un sólido entendimiento de la comunicación en sistemas distribuidos.