Estrategia de resolución del trabajo práctico

Alumno: Franco Astorga

DNI: 42359479

Para el trabajo primero opte por realizar las funciones solicitadas en único código, hardcodeando los valores de las variables usando Visual Studio Code. Una vez que la lógica funciono, empecé a armar tanto la aplicación cliente como servidor, haciendo una comunicación sencilla, para testear que se pueden recibir y enviar mensajes entre ambas. Para esto busque los comandos para compilar un programa C usando gcc, usando una aplicación de Windows que se parece a CMD llamada Vista previa de Terminal, la cual me da más opciones. Cuando se logra la comunicación, empecé a estructura el código anterior entre ambas aplicaciones. La lógica del trabajo se definió de la siguiente forma: los clientes solamente muestran el menú y el servidor es el que valida tanto las longitudes como los formatos de las cadenas recibidas. Tuve que solucionar problemas relacionados principalmente problemas de caracteres basura, que generaban conflictos al momento de realizar las validaciones. Esto se pudo solucionar con la implementación de funciones como getchar() y memset(), que ayudan e evitar este tipo de inconvenientes. Luego de esto pensé en que otro lenguaje podría usar para el otro cliente y por una cuestión de afinidad y eficiencia decidí optar por Python, en el cual no solo no ocurren los problemas anteriormente mencionados sino que se puede hacer lo mismo usando muchísimas menos líneas de código. Por último realice pruebas para testear las validaciones y conexiones, para asegurarme que no haya ningún fallo.

Github: https://github.com/FrancoAstorga/TP-SOCKETS

Pruebas realizadas

Para las siguientes pruebas voy a iniciar el servidor, el cual siempre va a estar escuchando por posibles clientes.

Prueba1:

Ingreso al menú cliente, y elijo una opción incorrecta

```
Inicializando Winsock...
Conectado al servidor 127.0.0.1 en el puerto 8080.

Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 5
Opcion no valida.
Presione una tecla para continuar . . .
```

Me dice **opción no válida** y tengo que presionar una tecla para que vuelva a elegir una opción.

Prueba2:

En ambos casos ingreso la opción 1 y longitudes no admitidas.

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 1
Indica la longitud deseada: 4

Respuesta del servidor: Longitud de nombre de usuario invalida.
```

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 1
Indica la longitud deseada: 16

Respuesta del servidor: Longitud de nombre de usuario invalida.
```

Prueba3:

En ambos casos ingreso la opción 1 con longitudes válidas.

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 1
Indica la longitud deseada: 10

Respuesta del servidor: lohaluxefi
```

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 1
Indica la longitud deseada: 14

Respuesta del servidor: henanewotiqogi
```

Prueba4:

En ambos casos ingreso la opción 2 y longitudes no admitidas.

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 2
Indica la longitud deseada: 7

Respuesta del servidor: Longitud de contrasena invalida.
```

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 2
Indica la longitud deseada: 50

Respuesta del servidor: Longitud de contrasena invalida.
```

Prueba5:

En ambos casos ingreso la opción 2 con longitudes válidas.

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 2
Indica la longitud deseada: 8

Respuesta del servidor: gsYh8p1J
```

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contrasenia
3. Salir
> 2
Indica la longitud deseada: 40

Respuesta del servidor: TUknp8IZSYg34QSGYxYi8D32LloYeR2z9Ig6mTjf
```

Prueba6:

Me desconecto del cliente hecho en C y me conecto con el cliente hecho en Python, para probar la opción 1 y 2.

```
Inicializando Winsock...
Esperando conexiones entrantes...
Cliente conectado.
Esperando conexiones entrantes...
```

El servidor vuelve a mostrar el mensaje **“Esperando conexiones entrantes…”.**

```
Inicializando Winsock...
Esperando conexiones entrantes...
Cliente conectado.
Esperando conexiones entrantes...
Cliente conectado.
```

Cuando inicia el menú en Python, en el servidor se muestra el mensaje **“Cliente conectado”.**

Prueba7

Una vez conectado con el cliente en Python, pruebo ingresar una longitud válida.

```
Conectado al servidor 127.0.0.1 en el puerto 8080.

Elige una opcion:
1. Generar nombre de usuario
2. Generar contraseña
3. Salir
> 1
Indica la longitud deseada: 10
Respuesta del servidor: cubemikamo
```

Prueba8

Siguiendo con el cliente en Python, pruebo ingresar una longitud no admitida.

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contraseña
3. Salir
> 1
Indica la longitud deseada: 20
Respuesta del servidor: Longitud de nombre de usuario invalida.
```

Prueba9

Por último pruebo la opción 2 tanto con una longitud válida como invalida.

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contraseña
3. Salir
> 2
Indica la longitud deseada: 10
Respuesta del servidor: paeBSrsSVq
```

```
Elige una opcion:
1. Generar nombre de usuario
2. Generar contraseña
3. Salir
> 1
Indica la longitud deseada: 60
Respuesta del servidor: Longitud de nombre de usuario invalida.
```