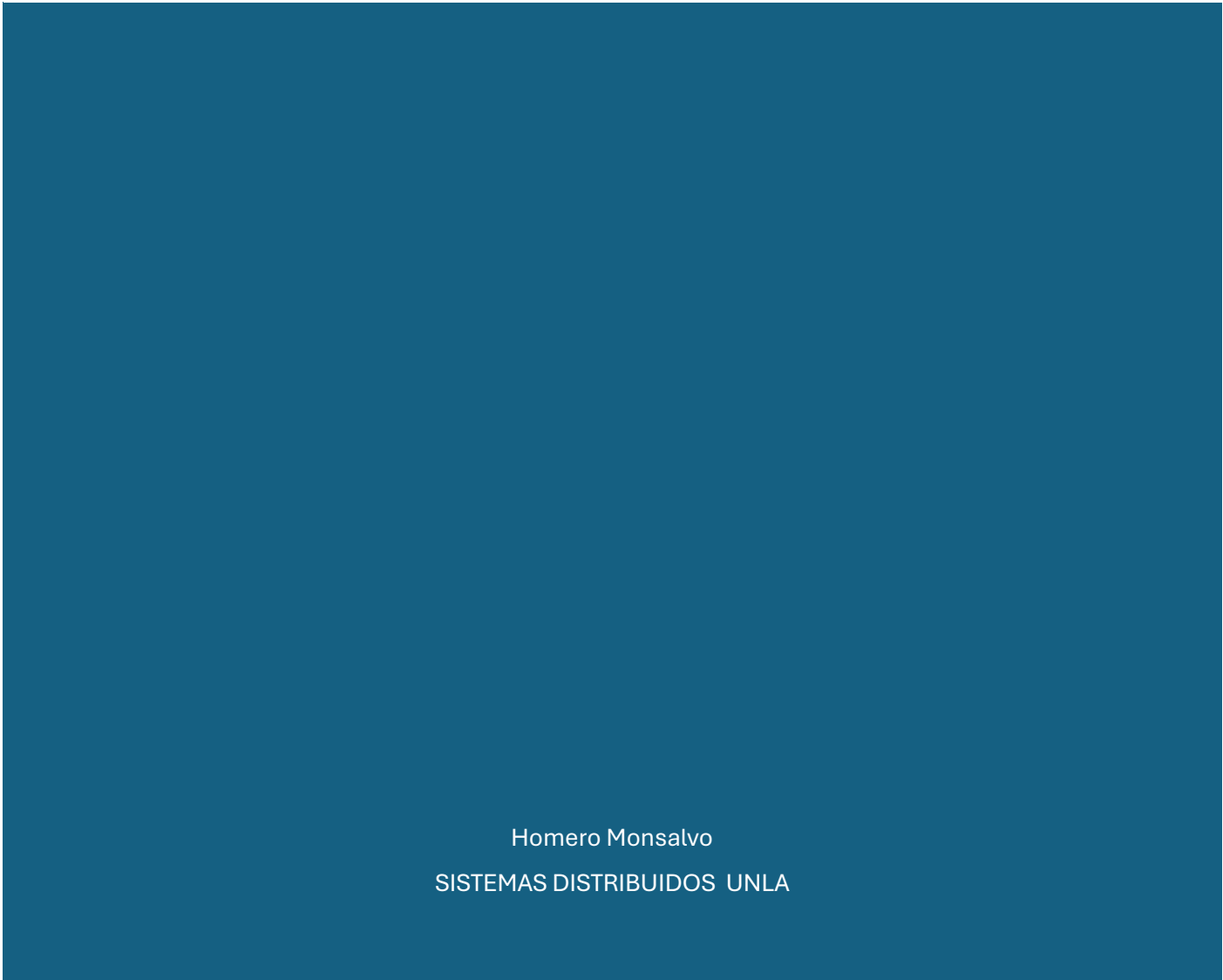




# ESTRATEGIA DE RESOLUCION DEL TRABAJO PRACTICO



Homero Monsalvo  
SISTEMAS DISTRIBUIDOS UNLA

## Contenido

1. Introducción.....	2
2. Estructura de la aplicación .....	2
3. Diseño .....	2
3.1 Diseño del Servidor.....	2
3.2 Diseño de los Clientes .....	2
4. Funcionalidad.....	3
4.1 Funcionalidad del Servidor .....	3
4.2 Funcionalidad de los Clientes .....	3
5. Pruebas .....	3
5.1 Generar usuario con ambos clientes .....	3
5.2 Generar contraseña con ambos clientes .....	4
5.3 Generar usuario con tamaño erróneo .....	7

# 1. Introducción

El objetivo del trabajo práctico es desarrollar una aplicación distribuida compuesta por un servidor y dos clientes, implementados en diferentes lenguajes de programación.

El servidor debe ser capaz de generar nombres de usuario o contraseñas según las solicitudes de los clientes, y los clientes deben poder interactuar con el servidor para obtener estos datos.

Este documento describe la estrategia utilizada para la implementación de la solución, incluyendo la estructura de la aplicación y las decisiones de diseño más relevantes.

[Enlace del repositorio en GitHub.](#)

## 2. Estructura de la aplicación

**Servidor:** Es responsable de recibir solicitudes de los clientes, generar nombres de usuario o contraseñas, y devolver los resultados.

**Clientes (C++ y Python):** Se han desarrollado dos clientes los cuales interactúan con el servidor. Ambos clientes permiten a los usuarios enviar solicitudes al servidor para generar nombres de usuario o contraseñas.

## 3. Diseño

### 3.1 Diseño del Servidor

Con respecto al servidor se utilizaron las siguientes bibliotecas:

**Winsock2:** Para la comunicación a través de sockets en sistemas Windows.

**Threads:** Para manejar la concurrencia y permitir que el servidor atienda a varios clientes al mismo tiempo.

### 3.2 Diseño de los Clientes

El cliente en C++ se implementó utilizando las mismas bibliotecas que el servidor (**Winsock2**) para la comunicación a través de sockets.

En cuanto al cliente en Python este constó de las bibliotecas **socket** y **threading**.

## 4. Funcionalidad

### 4.1 Funcionalidad del Servidor

El servidor escucha en un puerto especificado y acepta conexiones de múltiples clientes. Cada vez que un cliente se conecta, se crea un nuevo hilo para manejar esa conexión, lo que permite que el servidor continúe aceptando nuevas conexiones mientras procesa las solicitudes existentes.

El servidor incluye funciones para generar nombres de usuario y contraseñas. La longitud de los nombres de usuario y contraseñas es validada antes de generarse.

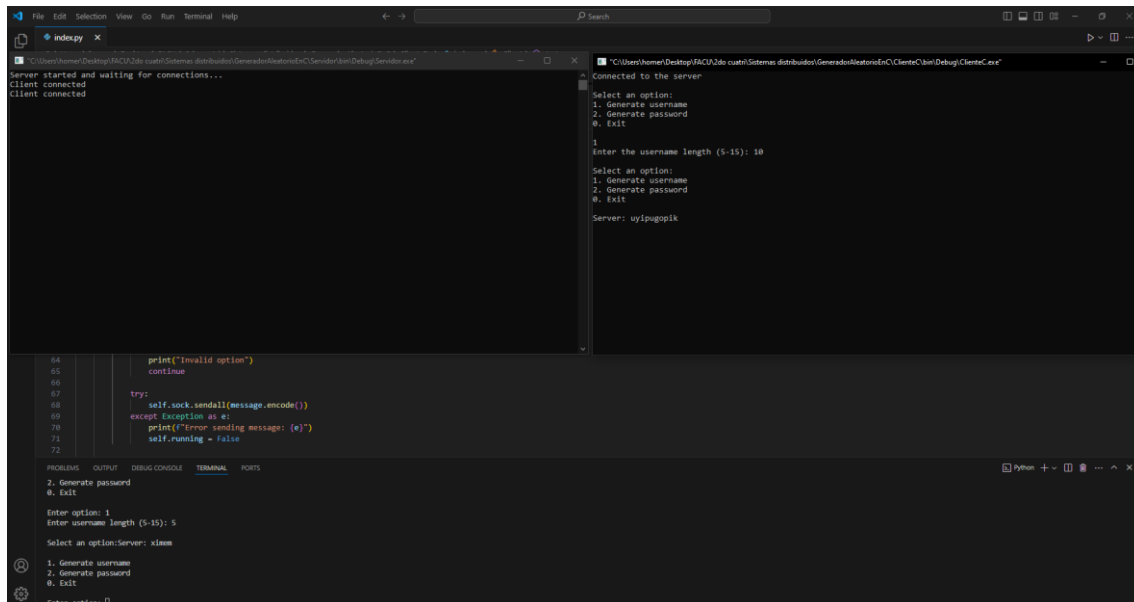
### 4.2 Funcionalidad de los Clientes

Ambos clientes cuentan con el mismo menú, donde el usuario puede seleccionar entre generar un nombre de usuario o una contraseña, especificar la longitud deseada y recibir la respuesta del servidor. Se implementó un bucle para permitir la comunicación continua del cliente hasta que el usuario decida salir.

## 5. Pruebas

### 5.1 Generar usuario con ambos clientes

#### Captura general



```
64         print("Invalid option")
65         continue
66
67     try:
68         self.sock.sendall(message.encode())
69     except Exception as e:
70         print("Error sending message: {e}")
71         self.running = False
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Server started and waiting for connections...  
Client connected  
Client connected

Connected to the server

Select an option:  
1. Generate username  
2. Generate password  
0. Exit

1

Enter the username length (5-15): 10

Select an option:  
1. Generate username  
2. Generate password  
0. Exit

Server: uyipugopik

2. Generate password  
0. Exit

Enter option: 1  
Enter username length (5-15): 5

Select an option: Server: xiam

1. Generate username  
2. Generate password  
0. Exit

Enter option: 1

#### Servidor

```
"C:\Users\homer\Desktop\FACU\2do cuatr\Sistemas distribuidos\GeneradorAleatorioEnC\Servidor\bin\Debug\Servidor.exe"
Server started and waiting for connections...
Client connected
Client connected
```

## Clientes

```
"C:\Users\homer\Desktop\FACU\2do cuatr\Sistemas distribuidos\GeneradorAleatorioEnC\ClienteC\bin\Debug\ClienteC.exe"
Connected to the server

Select an option:
1. Generate username
2. Generate password
0. Exit

1
Enter the username length (5-15): 10

Select an option:
1. Generate username
2. Generate password
0. Exit

Server: uyipugopik
```

```
PS C:\Users\homer> & C:/Users/homer/AppData/Local/Microsoft/WindowsApps/python3.11.exe
Connected to the server

Select an option:
1. Generate username
2. Generate password
0. Exit

Enter option: 1
Enter username length (5-15): 5

Select an option:Server: ximem

1. Generate username
2. Generate password
0. Exit

Enter option: █
```

## 5.2 Generar contraseña con ambos clientes

### Captura general

```
File Edit Selection View Go Run Terminal Help
index.py x
C:\Users\homer\Desktop\IACTU2do cuatr\Sistemas distribuidos\GeneradorAleatorio\Server\bin\Debug\Server.exe
Server started and waiting for connections...
Client connected
Client connected

0. Exit
Enter option: 1
Enter username length (5-15): 5
Select an option: Server: ximen
1. Generate username
2. Generate password
0. Exit
Enter option: 2
Enter password length (8-50): 8
Select an option:
Server: ANCi1912
1. Generate username
2. Generate password
0. Exit
Enter option: 0

C:\Users\homer\Desktop\IACTU2do cuatr\Sistemas distribuidos\GeneradorAleatorio\Cliente\bin\Debug\Cliente.exe
Connected to the server
Select an option:
1. Generate username
2. Generate password
0. Exit
1
Enter the username length (5-15): 10
Select an option:
1. Generate username
2. Generate password
0. Exit
Server: uyipugopik
2
Enter the password length (8-50): 8
Select an option:
1. Generate username
2. Generate password
0. Exit
Server: 443X2DjC
0
```

## Clientes

```
"C:\Users\homer\Desktop\FACU\2do cuatri\Sistemas dis
Connected to the server

Select an option:
1. Generate username
2. Generate password
0. Exit

1
Enter the username length (5-15): 10

Select an option:
1. Generate username
2. Generate password
0. Exit

Server: uyipugopik
2
Enter the password length (8-50): 8

Select an option:
1. Generate username
2. Generate password
0. Exit

Server: 44JX2DjC
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

2. Generate password
0. Exit

Enter option: 1
Enter username length (5-15): 5

Select an option:Server: ximem

1. Generate username
2. Generate password
0. Exit

Enter option: 2
Enter password length (8-50): 8

Select an option:
Server: AnCvi912
1. Generate username
2. Generate password
0. Exit

Enter option: █
```

## 5.3 Generar usuario con tamaño erróneo

### Cientes

```
1
Enter the username length (5-15): a
Invalid length
```

```
Select an option:
1. Generate username
2. Generate password
0. Exit
```

```
1
Enter the username length (5-15): 4
Invalid length
```

```
Select an option:
1. Generate username
2. Generate password
0. Exit
```

```
Select an option:
1. Generate username
2. Generate password
0. Exit
```

```
Enter option: 1
Enter username length (5-15): 4
Invalid length
```

```
Select an option:
1. Generate username
2. Generate password
0. Exit
```

```
Enter option: █
```