



## **Protocolos de Internet**

### **1° CUATRIMESTRE DE 2024**

**Trabajo Práctico N° 1C**

**Grupo: 4**

**Profesor/a: Javier Adolfo Ouret**

**Integrantes:**

<b>N o</b>	<b>Apellido y Nombre</b>	<b>Legajo</b>	<b>Email</b>
<b>1</b>	<b>Marina Mercadal</b>	<b>152150976</b>	<b>marinamercadal@uca.edu.ar</b>
<b>2</b>	<b>Martina Naiquen Ruiz</b>	<b>29181</b>	<b>martinaruiz@uca.edu.ar</b>
<b>3</b>	<b>Carolina Suarez</b>	<b>152000738</b>	<b>suarezmacar@gmail.com</b>

Para la parte C del trabajo práctico 1 hay que escribir una aplicación cliente servidor que muestre las direcciones y puertos de todos los clientes conectados del lado del servidor y devuelva a cada cliente el día y hora de conexión, y el tiempo que estuvo (o está conectado).

En primera instancia hay que hacerlo para C-S con Concurrencia Aparente (Select), el cual lo hicimos mirando los códigos en el archivo Socket\_Cliente\_Select.py y Socket\_Servidor\_Select.py.

Salida del servidor:

```
iniciando en localhost port 10000
esperando el próximo evento
  conexión desde: ('127.0.0.1', 53512)
esperando el próximo evento
  conexión desde: ('127.0.0.1', 53514)
  recibido b'Este mensaje ' desde ('127.0.0.1', 53512)
esperando el próximo evento
  recibido b'Este mensaje ' desde ('127.0.0.1', 53514)
  enviando b'Este mensaje ' a ('127.0.0.1', 53512)
esperando el próximo evento
  ('127.0.0.1', 53512) cola vacía
  enviando b'Este mensaje ' a ('127.0.0.1', 53514)
esperando el próximo evento
  ('127.0.0.1', 53514) cola vacía
esperando el próximo evento
  recibido b'es enviado ' desde ('127.0.0.1', 53512)
esperando el próximo evento
  recibido b'es enviado ' desde ('127.0.0.1', 53514)
  enviando b'es enviado ' a ('127.0.0.1', 53512)
esperando el próximo evento
```

```
  ('127.0.0.1', 53512) cola vacía
  enviando b'es enviado ' a ('127.0.0.1', 53514)
esperando el próximo evento
  ('127.0.0.1', 53514) cola vacía
esperando el próximo evento
  recibido b'en partes.' desde ('127.0.0.1', 53512)
esperando el próximo evento
  recibido b'en partes.' desde ('127.0.0.1', 53514)
  enviando b'en partes.' a ('127.0.0.1', 53512)
esperando el próximo evento
  ('127.0.0.1', 53512) cola vacía
  enviando b'en partes.' a ('127.0.0.1', 53514)
esperando el próximo evento
  ('127.0.0.1', 53514) cola vacía
esperando el próximo evento
  cerrando... ('127.0.0.1', 53514)
  cerrando... ('127.0.0.1', 53514)
esperando el próximo evento
```

Salida del cliente:

```
conectando a localhost puerto 10000
('127.0.0.1', 53512): enviando b'Este mensaje '
('127.0.0.1', 53514): enviando b'Este mensaje '
('127.0.0.1', 53512): recibido b'Este mensaje '
('127.0.0.1', 53514): recibido b'Este mensaje '
('127.0.0.1', 53512): enviando b'es enviado '
('127.0.0.1', 53514): enviando b'es enviado '
('127.0.0.1', 53512): recibido b'es enviado '
('127.0.0.1', 53514): recibido b'es enviado '
('127.0.0.1', 53512): enviando b'en partes.'
('127.0.0.1', 53514): enviando b'en partes.'
('127.0.0.1', 53512): recibido b'en partes.'
('127.0.0.1', 53514): recibido b'en partes.'
```

A esto falta agregar que devuelva a cada cliente el día y hora de conexión, y el tiempo que estuvo (o está conectado).

Ejecutado desde Pycharm:

```
conectando a localhost puerto 10000
('127.0.0.1', 53538): enviando b'Este mensaje '
('127.0.0.1', 53540): enviando b'Este mensaje '
('127.0.0.1', 53538): recibido b'Hora de conexi\xc3\xb3n: 2024-05-12 12:49:03'
('127.0.0.1', 53540): recibido b'Hora de conexi\xc3\xb3n: 2024-05-12 12:49:03Tiempo conectado: 0.00 segundos'
('127.0.0.1', 53538): enviando b'es enviado '
('127.0.0.1', 53540): enviando b'es enviado '
('127.0.0.1', 53538): recibido b'Tiempo conectado: 0.00 segundosEste mensaje Tiempo conectado: 0.00 segundoses enviado '
('127.0.0.1', 53540): recibido b'Este mensaje Tiempo conectado: 0.00 segundos'
('127.0.0.1', 53538): enviando b'en partes.'
('127.0.0.1', 53540): enviando b'en partes.'
('127.0.0.1', 53538): recibido b'Tiempo conectado: 0.00 segundos'
('127.0.0.1', 53540): recibido b'es enviado Tiempo conectado: 0.00 segundos'
```

Salida de terminal desde Visual Studio Code:

```
marina@LAPTOP-204JRJF2:~/vscode-server/.vscode/TP1C$ /bin/python3 /home/marina/.vscode-server/.vscode/TP1C/Saparente.py
Socket creado
Enlace del socket completado
Socket en modo escucha
IP: 127.0.0.1 puerto: 36446
IP:127.0.0.1, puerto:36446. Recibido!
: mensaje al servidor
cliente IP: 127.0.0.1 puerto: 36446 ha cerrado la conexión.
[]
```

```
base.py
localhost -> Puerto 8000
13 / mayo :11:21

Enviar mensaje : mensaje al servidor
recibido:13 / mayo :11:21 . tiempo luego de conexion: 0:00:07.649216

Enviar mensaje : salir
Conexion cerrada
socket cerrado
marina@LAPTOP-204JRJF2:~/vscode-server/.vscode/TP1C$
```

Salida de terminal desde Visual Studio Code:

```
/bin/python3 /home/marina/.vscode-server/.vscode/TP1C/Sconcurrente.py
marina@LAPTOP-204JRJF2:~/vscode-server/.vscode/TP1C$ /bin/python3 /home/marina/.vscode-server/.vscode/TP1C/Sconcurrente.py
Socket creado
Enlace del socket completado
Socket en modo escucha
IP: 127.0.0.1 | Puerto: 36520
Mensaje recibido de 127.0.0.1:36520: mensaje de cliente1
IP: 127.0.0.1 | Puerto: 36522
Mensaje recibido de 127.0.0.1:36522: mensaje de cliente2
```

```
marina@LAPTOP-204JRJF2:~/vscode-server/.vscode/TP1C$ python3 cliente_base.py
localhost -> Puerto 8000
13 de mayo a las 11:34 hs conexion exitosa!

Enviar mensaje : mensaje de cliente1
13 de mayo a las 11:34 hs Tiempo desde que se conecto-> 0:00:06.519050

Enviar mensaje : []
```

```
marina@LAPTOP-204JRJF2:~/vscode-server/.vscode/TP1C$ python3 cliente_base.py
localhost -> Puerto 8000
13 de mayo a las 11:34 hs conexion exitosa!

Enviar mensaje : mensaje de cliente2
13 de mayo a las 11:34 hs Tiempo desde que se conecto-> 0:00:08.738872

Enviar mensaje : []
```