

EXAMEN PARCIAL “ARQUITECTURA CLIENTE-SERVIDOR” SEMESTRE 2025-1

El examen parcial consiste en crear el par de programas Cliente-Servidor con base en el par de programas 04 visto en clase.

Índice del Tema de Sockets Internet: <http://profesores.fi-b.unam.mx/carlos/acs/Tema-03-Sockets-INTERNET-C/index.html>

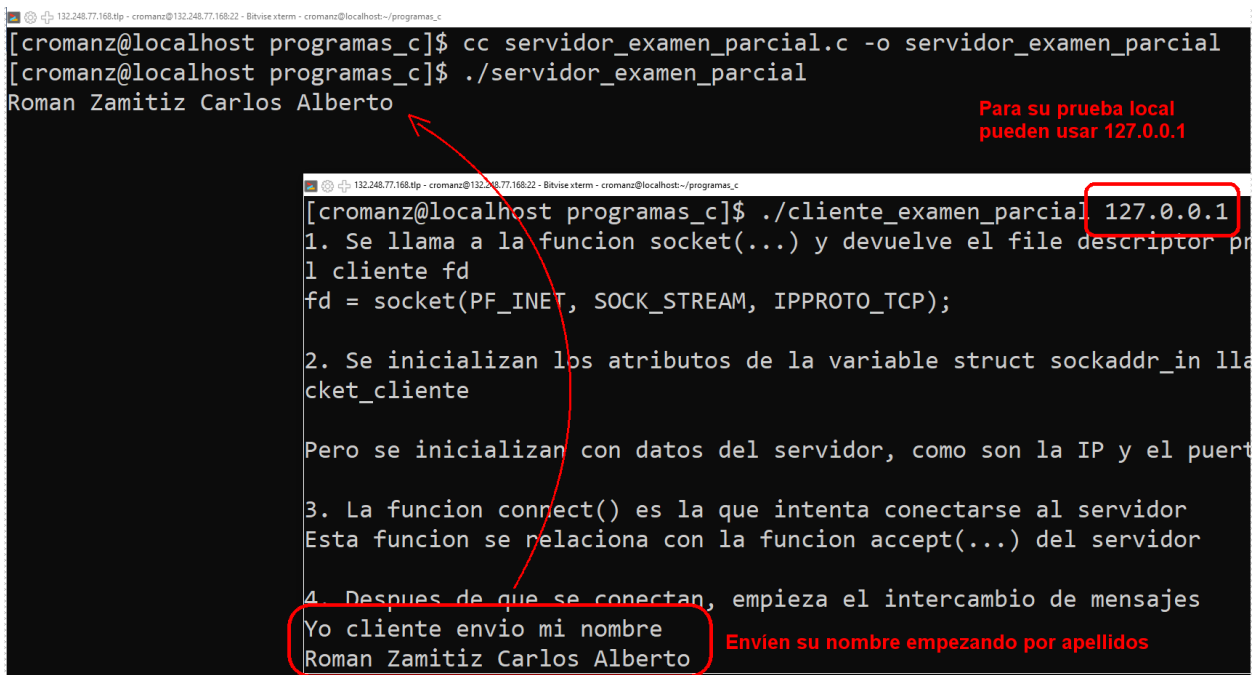
Servidor: http://profesores.fi-b.unam.mx/carlos/acs/Tema-03-Sockets-INTERNET-C/programa04_servidor_internet.c

Cliente: http://profesores.fi-b.unam.mx/carlos/acs/Tema-03-Sockets-INTERNET-C/programa04_cliente_internet.c

Prueben este par de códigos. Observen que, una vez establecida la comunicación, el Servidor envía un mensaje al Cliente y ambos terminan

ACTIVIDADES A REALIZAR:

- Modificar ese par de códigos para que, una vez establecida la comunicación, el Cliente envíe un mensaje al Servidor y ambos terminen.
- El mensaje que el Cliente debe enviar es tu nombre en formato APELLIDO_PATERNO APELLIDO_MATERNO NOMBRE(S). Para facilitarles el desarrollo del cliente, **su nombre puede estar en “código duro”**, dentro del programa.
- Recuerden que, para sus pruebas locales, si están en la misma máquina el cliente puede apuntar a la IP 127.0.0.1
- El puerto va a seguir siendo el mismo del programa 04, que es el **puerto 4898**.
- **Deben hacer captura de pantalla de ambas terminales una al lado de la otra**, una donde estén corriendo su cliente y otra donde estén corriendo el servidor y en esta terminal manden a imprimir el nombre que recibió del cliente.



```
[cromanz@localhost programas_c]$ cc servidor_examen_parcial.c -o servidor_examen_parcial
[cromanz@localhost programas_c]$ ./servidor_examen_parcial
Roman Zamitiz Carlos Alberto

[cromanz@localhost programas_c]$ ./cliente_examen_parcial 127.0.0.1
1. Se llama a la funcion socket(...) y devuelve el file descriptor pr
1 cliente fd
fd = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);

2. Se inicializan los atributos de la variable struct sockaddr_in lla
cket_cliente

Pero se inicializan con datos del servidor, como son la IP y el puert

3. La funcion connect() es la que intenta conectarse al servidor
Esta funcion se relaciona con la funcion accept(...) del servidor

4. Despues de que se conectan, empieza el intercambio de mensajes
Yo cliente envio mi nombre
Roman Zamitiz Carlos Alberto
```

- Ya que hayan terminado su prueba local, van a lanzar el cliente apuntando a la IP donde se está ejecutando el servidor en ambiente productivo. **LA IP DEL SERVIDOR ES 132.248.77.168**:

El servidor lo dejaré corriendo desde hoy y lo detendré a las 11:59 PM del lunes 14 de octubre, así que pueden conectarse y enviar su mensaje a la hora que gusten.

Después de que su cliente termine, **EN ESA MISMA TERMINAL** ejecuten un comando **date** para que quede registrada la fecha y hora en la que lanzaron el cliente al host 132.248.77.168 y así capturen la pantalla.

```
4. Despues de que se conectan, empieza e
Yo cliente envio mi nombre
Roman Zamitiz Carlos Alberto

5. Por ultimo, se cierra el file descrip
close(fd);
[cromanz@localhost programas_c]$ date
mié oct  9 18:01:00 CDT 2024
[cromanz@localhost programas_c]$
```

El entregable PDF que me enviarán será:

1. el código fuente de su cliente (el *.c) **no envíen el ejecutable porque el servidor de correo bloqueará su correo por cuestiones de seguridad.**
2. el código fuente de su servidor (el *.c) **no envíen el ejecutable porque el servidor de correo bloqueará su correo por cuestiones de seguridad.**
3. La captura de pantalla de sus terminales de su prueba local.
4. La captura de pantalla del cliente apuntando al servidor 132.248.77.168 y con la salida del comando date.

Pueden enviar el correo desde hoy y hasta la noche del lunes 14 de octubre, es decir hasta antes de la próxima clase a carlos@fib.unam.mx

Cualquier duda o pregunta envíenme un mensaje de Telegram. Pueden escribirme para decirme que ya enviaron su nombre al servidor. Yo reviso que su nombre haya llegado y les confirmo que ya quedó registrado.