

## Sockets C

Para realizar esta práctica, se han creado 3 programas independientes cuya función es conectarse entre sí mediante diferentes sockets, tanto UDP como TCP. Simularán la entrada y envío de medidas de diferentes sensores.

Las sentencias necesarias para compilar y ejecutar estos programas en UNIX son:

- Compilar: `gcc NombreArchivo.c -o nombreEjecutable`
- Ejecutar: `./nombreEjecutable`

### Sensor.c

El programa *sensor.c* será el encargado de recoger los identificadores y las medidas de los sensores. Pedirá al usuario que introduzca el identificador del sensor por teclado, especificando que la manera de introducirlo será *abcdID*, dónde:

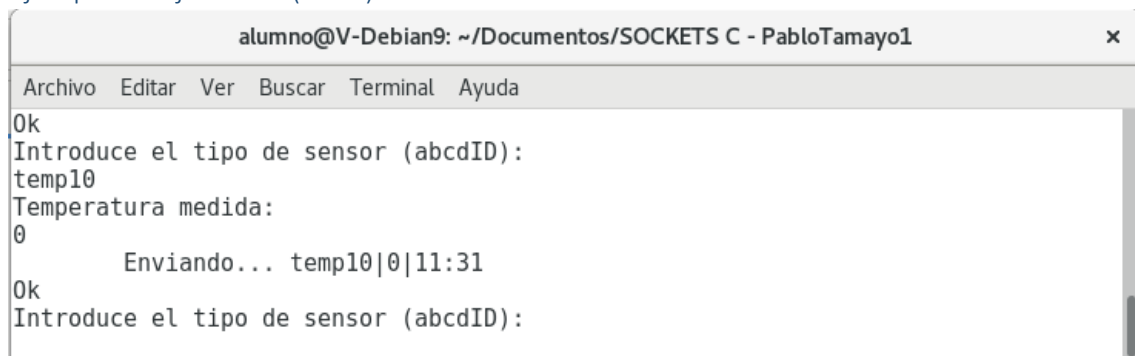
- *abcd*: 4 caracteres que describan el tipo de sensor.
- *ID*: números identificativos del sensor.

Las medidas de cada sensor se generarán mediante la función *srand(getpid())* definida al principio del código y se cargarán en una variable *medida* en cada ejecución, con un rango de 0 a 30.

También existen sentencias encargadas de recoger la hora del sistema y darle formato para enviarla en un solo mensaje junto con el tipo de sensor y la medida generada.

Una vez generados los datos, se procede a enviar un mensaje con todos ellos mediante un cliente UDP.

### Ejemplo de ejecución (UNIX)



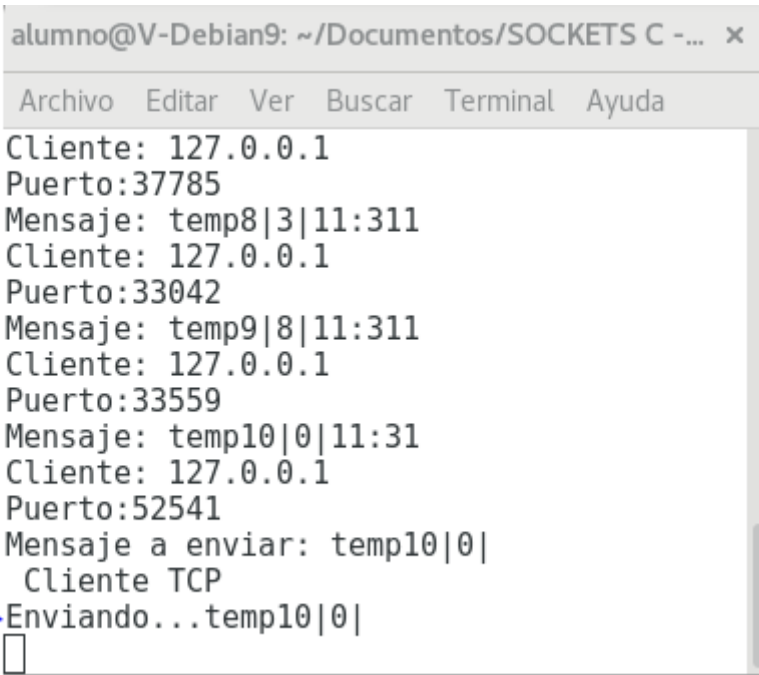
```
alumno@V-Debian9: ~/Documentos/SOCKETS C - PabloTamayo1
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Ok
Introduce el tipo de sensor (abcdID):
temp10
Temperatura medida:
0
      Enviando... temp10|0|11:31
Ok
Introduce el tipo de sensor (abcdID):
```

### ServidorCentral.c

Consiste de un servidor UDP, que recibe la información enviada desde *Sensor.c* y un cliente TCP que la reenvía al programa *PaginaWeb.c* cuándo se han obtenido 10 medidas de sensores.

Al recibir el mensaje, será necesario separar sus componentes mediante la función *strtok()*, que cargaremos en tres variables *char* preparadas a tal efecto mediante *strcpy()*. Una vez separados los diferentes campos del mensaje, los guardaremos en una estructura *Mensaje* (definida al principio del programa) y utilizaremos *strcat()* para unir los campos de la estructura en un nuevo mensaje, que enviaremos a *PaginaWeb.c*

## Ejemplo de ejecución (UNIX)



The screenshot shows a terminal window titled 'alumno@V-Debian9: ~/Documentos/SOCKETS C -...'. The output of the program is as follows:

```
Cliente: 127.0.0.1
Puerto:37785
Mensaje: temp8|3|11:311
Cliente: 127.0.0.1
Puerto:33042
Mensaje: temp9|8|11:311
Cliente: 127.0.0.1
Puerto:33559
Mensaje: temp10|0|11:31
Cliente: 127.0.0.1
Puerto:52541
Mensaje a enviar: temp10|0|
Cliente TCP
Enviando...temp10|0|
```

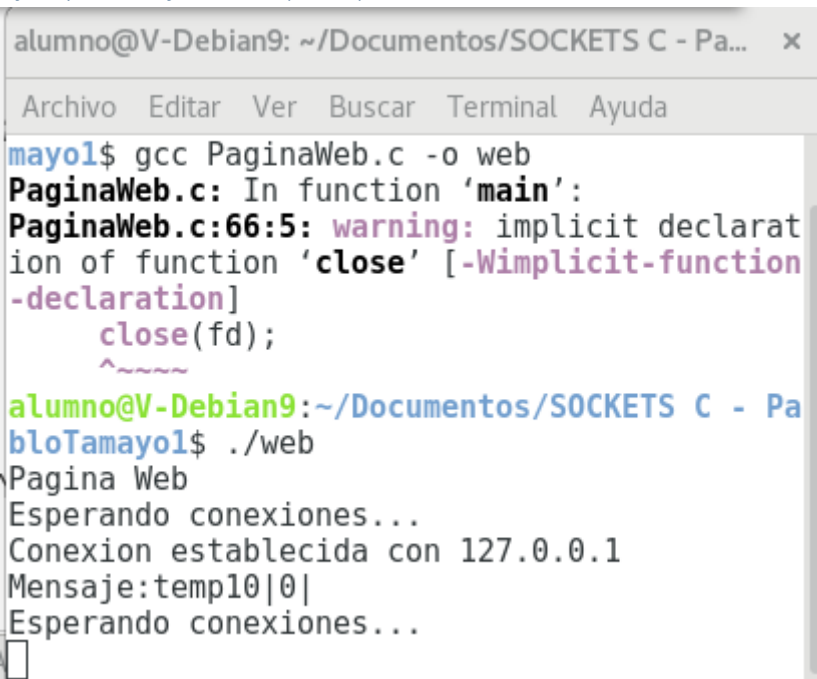
A blue arrow points to the 'Enviando...temp10|0|' line.

Se observa que siempre se envía la última medida recibida por el programa.

[PaginaWeb.c](#)

Este programa cuenta con un servidor TCP, encargado de recibir la información que nos aporta *ServidorCentral.c* y lo muestra por pantalla.

## Ejemplo de ejecución (UNIX)



The screenshot shows a terminal window titled 'alumno@V-Debian9: ~/Documentos/SOCKETS C - Pa...'. The output is as follows:

```
mayo1$ gcc PaginaWeb.c -o web
PaginaWeb.c: In function 'main':
PaginaWeb.c:66:5: warning: implicit declaration of function 'close' [-Wimplicit-function-declaration]
    close(fd);
    ^~~~~
alumno@V-Debian9:~/Documentos/SOCKETS C - Pa
bloTamayo1$ ./web
Pagina Web
Esperando conexiones...
Conexion establecida con 127.0.0.1
Mensaje:temp10|0|
Esperando conexiones...
```

## Ejemplo de ejecución conjunta

The image shows three terminal windows from a Debian 9 system, illustrating a concurrent execution of a C program and a web server.

**Top Terminal Window:** `alumno@V-Debian9: ~/Documentos/SOCKETS C - PabloTamayo1`

```
Ok
Introduce el tipo de sensor (abcdID):
temp10
Temperatura medida:
0
    Enviando... temp10|0|11:31
Ok
Introduce el tipo de sensor (abcdID):
```

**Bottom Left Terminal Window:** `alumno@V-Debian9: ~/Documentos/SOCKETS C - ...`

```
Cliente: 127.0.0.1
Puerto:37785
Mensaje: temp8|3|11:311
Cliente: 127.0.0.1
Puerto:33042
Mensaje: temp9|8|11:311
Cliente: 127.0.0.1
Puerto:33559
Mensaje: temp10|0|11:31
Cliente: 127.0.0.1
Puerto:52541
Mensaje a enviar: temp10|0|
    Cliente TCP
Enviando...temp10|0|
```

**Bottom Right Terminal Window:** `alumno@V-Debian9: ~/Documentos/SOCKETS C - Pa...`

```
mayo1$ gcc PaginaWeb.c -o web
PaginaWeb.c: In function 'main':
PaginaWeb.c:66:5: warning: implicit declaration of function 'close' [-Wimplicit-function-declaration]
    close(fd);
    ~~~~
alumno@V-Debian9:~/Documentos/SOCKETS C - Pa
bloTamayo1$ ./web
Pagina Web
Esperando conexiones...
Conexion establecida con 127.0.0.1
Mensaje:temp10|0|
Esperando conexiones...
```