

## Capítulo 16. Cliente - servidor

La programación cliente-servidor es uno de los temas más complejos en NGL.

Cuando realizamos esta programación, debemos tener en cuenta tres factores: el servidor, el cliente y la comunicación que se realiza entre ellos.

De una forma básica:

- Servidor, coordina los clientes entre ellos y mantiene la información centralizada y actualizada, comunica a los clientes conectados los cambios necesarios para que estén sincronizados.
- Cliente, procesa la información propia y envía al servidor los datos que son compartidos.

Existen varias formas de realizar esta comunicación, dependiendo del tipo de juego que se quiera realizar.

Nautilus usa un tipo de comunicación basado en “frases”, que pueden usarse de diferentes formas.

Por ejemplo, con la estructura “identidad + orden”:

“servidor activa bandera azul”

Un cliente le dice al servidor que active dicha bandera.

“cliente cinco activa alarma”

El servidor ha mandado al cliente número cinco la orden de activar la alarma.

Primero debe crearse el servidor, que es el que “escucha” y luego los clientes que son quienes “hablan” al servidor.

En el servidor debe añadirse una línea al archivo `/etc/services` similar a esta:

```
cpp_java      15557/tcp
```

`cpp_java` es el identificador usado en el ejemplo.

15557 es el puerto que usaremos.

Tcp es el protocolo de comunicación.

El identificador debe ser un nombre no usado. El puerto debe estar disponible, si se usan elementos como routers, firewalls, etc.

En el programa de ejemplo que se adjunta, el servidor envía órdenes al cliente, que pinta en pantalla una entidad.

Nautilus permite crear partidas con varios clientes conectados a un mismo servidor.

### Análisis del programa servidor

La función `ngl_net_open_net_server()` usa como parámetro el identificador indicado anteriormente.

Se espera a que los clientes conecten con la función `ngl_net_accept_client()`.

Según la tecla pulsada, copiamos a Cadena el mensaje que enviamos al cliente.

En este ejemplo, se usa la estructura: quien escribe + orden:

"server A" significa que en el servidor se ha pulsado la tecla A.

"server D" significa que en el servidor se ha pulsado la tecla D.

etc.

Cuando concluye el programa, se cierran las conexiones del servidor y del cliente con las funciones `ngl_net_close_server()` y `ngl_net_close_client()` respectivamente.

### **Análisis del programa cliente**

El cliente escucha en el host y puertos indicados con la función `ngl_net_open_network()`

Se lee el mensaje enviado por el servidor con la función `ngl_net_read()`, que usa como parámetros el identificador de la conexión, la variable en la que se almacena el mensaje (Cadena) y la longitud de dicho dato (100).

Por último, se comprueba el contenido del mensaje y se ejecutan las instrucciones correspondientes.

Funciones de servidor:

#### **`ngl_net_open_server`**

Abre un servidor en modo de red local.

#### **`ngl_net_open_net_server`**

Abre un servidor en modo Internet.

#### **`ngl_net_close_server`**

Cierra un servidor.

#### **`ngl_net_accept_client`**

Acepta la conexión de un cliente.

Funciones de cliente:

#### **`ngl_net_open_localhost`**

Abre una conexión en modo red local

#### **`ngl_net_open_network`**

Abre una conexión en modo Internet

#### **`ngl_net_close_client`**

Cierra una conexión de cliente

Funciones de datos:

**ngl\_net\_read**

Lee un dato de una conexión activa.

**ngl\_net\_write**

Escribe un dato en una conexión activa.