

(<https://poesiabinaria.net/2015/10/programas-que-me-han-salvado-la-vida-este-2015/>)

Estás aquí : Poesía Binaria (<https://poesiabinaria.net/>)

## Creando un servidor que acepte múltiples clientes simultáneos en C

6 febrero, 2011

([/#facebook](#)) ([/#twitter](#))  
([https://www.addtoany.com/add\\_to/google\\_plus?linkurl=https%3A%2F%2Fpoesiabinaria.net%2F2015/10/programas-que-me-han-salvado-la-vida-este-2015/&linkname=Programas-que-me-han-salvado-la-vida-en-2015](https://www.addtoany.com/add_to/google_plus?linkurl=https%3A%2F%2Fpoesiabinaria.net%2F2015/10/programas-que-me-han-salvado-la-vida-este-2015/&linkname=Programas-que-me-han-salvado-la-vida-en-2015))  
([/#pinterest](#)) ([/#buffer](#)) ([/#flipboard](#)) ([/#whatsapp](#))  
([/#telegram](#)) ([/#meneame](#)) ([/#linkedin](#)) ([/#pocket](#))  
(<https://www.addtoany.com/share?url=https%3A%2F%2Fpoesiabinaria.net%2F2015/10/programas-que-me-han-salvado-la-vida-este-2015/&title=Programas-que-me-han-salvado-la-vida-en-2015>)  
multiples-clientes-simultaneos-en-c%2F&title=Creando%20un%20servidor%20que%20accepte%20m%C3%BAltiples-clientes-simultaneos-en-c%2F



([https://poesiabinaria.net/wp-content/uploads/2011/02/5989707041\\_64de393f6b\\_o.jpg](https://poesiabinaria.net/wp-content/uploads/2011/02/5989707041_64de393f6b_o.jpg)) Para hacer una prueba de esto, crearemos un servidor al que nos podremos conectar por telnet y pedir cierta información a través de comandos. El ejemplo soporta los siguientes comandos (en mayúsculas):

- DATE: Pide la fecha al servidor
- TIME: Pide la hora al servidor
- HOLA: Saluda y me dice mi IP
- EXIT: Cierra el cliente actual
- CERRAR: Cierra el servidor.
- Cualquier otra cosa se manda repetida (ECHO)

```
166
167 int AtiendeCliente(int socket, struct sockaddr_in addr)
168 {
169
170     char buffer[BUFFERSIZE];
171     char aux[BUFFERSIZE];
172     int bytecount;
173     int fin=0;
174     int code=0;    /* Código de salida por defecto */
175     time_t t;
176     struct tm *tmp;
```

La clave para poder atender varios clientes simultáneos es bifurcar (fork()) el programa cuando se conecta un nuevo cliente, de esta forma, el diálogo con el cliente se lleva a cabo desde un nuevo proceso mientras que el proceso padre (el primero) sigue escuchando en el puerto esperando conexiones.

Utilizamos select() para no detener la ejecución del programa cuando estamos esperando conexiones; tal vez el programa tenga que hacer algo mientras nadie está conectándose (y es que el programa principal permanecerá aquí la mayor parte del tiempo), por ejemplo lo que hacemos aquí es controlar los hijos que se mueren y ver cómo se han muerto. select() establece un tiempo máximo de espera (500000 microsegundos en este caso, ver la variable tv); si pasado ese tiempo el descriptor socket\_host no ha cambiado, seguiremos la ejecución normal del programa, pero si durante ese tiempo cambia ejecutaremos accept()

La función select() funciona con grupos de descriptores y por eso utilizamos las macros FD\_ZERO() para limpiar el grupo y FD\_SET() para añadir un descriptor al grupo. El primer parámetro de select() debe ser el descriptor de fichero más grande a comprobar+1; como sólo tenemos un descriptor (socket\_host) debemos añadir este al grupo, y pasarle a select en su primer parámetro socket\_host+1.

Cuando llega un cliente, como dijimos antes, se abrirá un proceso hijo que lo procesará con una de estas dos funciones: AtiendeCliente() o DemasiadosClientes() dependiendo de si el número máximo de clientes ha sido alcanzado o no. Además, la salida del proceso hijo se obtendrá de la salida de estas funciones, siendo el valor de salida clave 99 el que necesita el proceso padre para cerrar/terminar el proceso servidor, es decir, si un hijo sale con el valor de retorno 99 provocará el cierre del servidor. Este valor se maneja debajo del switch(); primero con waitpid() esperamos la muerte de alguno de nuestros hijos aunque con el modificador WNOHANG provocamos que si no se ha muerto ninguno, no vamos a esperar, el programa seguirá su ejecución.

Desde AtiendeCliente() veremos un método muy rudimentario para implementar los comandos, aunque para un ejemplo y 4 comandos nos vale perfectamente.

DemasiadosClientes() devolverá únicamente un mensaje al cliente que se acaba de conectar avisando de que hay demasiados clientes conectados.

Reloj() es un pequeño reloj en modo texto que va girando, a cada iteración del bucle principal, está aquí para que nosotros veamos que el servidor está haciendo algo. Utiliza códigos ANSI, para volver a escribirse en el lugar que estaba, y fflush() para volcar el texto a pantalla antes de nada (puede ser que se acumulen varios mensajes y entonces el dibujado sería incorrecto).

Para probar este programa, debemos compilar con:











```
$ gcc -o servtcp servtcp.c
```

y ejecutarlo; y, en un terminal aparte, conectar con telnet. Si, por ejemplo lo queremos hacer todo desde el mismo ordenador; si el servidor está en un ordenador diferente y nos queremos conectar a él cambiamos «localhost» por la IP o el host del ordenador donde se esté ejecutando el cliente:

```
$ telnet localhost 7000
```

Foto principal: Seeweb (<https://www.flickr.com/photos/seeweb/5989707041>)

## También podría interesarte....

 ([/#facebook](#))  ([/#twitter](#))  
([https://www.addtoany.com/add\\_to/google\\_plus?linkurl=https%3A%2F%2Fpoesiabinaria.net%2Ftag/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c%2F&linkname=Creando%20un%20servidor%20que%20accepte%20m%C3%BAltiples-clientes-simultaneos-en-c%2F&title=Creando%20un%20servidor%20que%20accepte%20m%C3%BAltiples-clientes-simultaneos-en-c%2F](https://www.addtoany.com/add_to/google_plus?linkurl=https%3A%2F%2Fpoesiabinaria.net%2Ftag/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c%2F&linkname=Creando%20un%20servidor%20que%20accepte%20m%C3%BAltiples-clientes-simultaneos-en-c%2F&title=Creando%20un%20servidor%20que%20accepte%20m%C3%BAltiples-clientes-simultaneos-en-c%2F))  
 ([/#pinterest](#))  ([/#buffer](#))  ([/#flipboard](#))  ([/#whatsapp](#))  
 ([/#telegram](#))  ([/#meneame](#))  ([/#linkedin](#))  ([/#pocket](#))  
(<https://www.addtoany.com/share?url=https%3A%2F%2Fpoesiabinaria.net%2Ftag/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c%2F&title=Creando%20un%20servidor%20que%20accepte%20m%C3%BAltiples-clientes-simultaneos-en-c%2F>)

Posted in : C/C++ (<https://poesiabinaria.net/category/codigoc/>) , Comunicación (<https://poesiabinaria.net/category/comunicacion/>) , Linux (<https://poesiabinaria.net/category/linux/>)

Tags : accept (<https://poesiabinaria.net/tag/accept/>) , bind (<https://poesiabinaria.net/tag/bind/>) , c (<https://poesiabinaria.net/tag/c/>) , client (<https://poesiabinaria.net/tag/client/>) , clientes (<https://poesiabinaria.net/tag/clientes/>) , comandos (<https://poesiabinaria.net/tag/comandos/>) , fecha (<https://poesiabinaria.net/tag/fecha/>) , fork (<https://poesiabinaria.net/tag/fork/>) , hijo (<https://poesiabinaria.net/tag/hijo/>) , hora (<https://poesiabinaria.net/tag/hora/>) , host (<https://poesiabinaria.net/tag/host/>) , localhost (<https://poesiabinaria.net/tag/localhost/>) , multiples (<https://poesiabinaria.net/tag/multiples/>) , padre (<https://poesiabinaria.net/tag/padre/>) , proceso (<https://poesiabinaria.net/tag/proceso/>) , select (<https://poesiabinaria.net/tag/select/>) , server (<https://poesiabinaria.net/tag/server/>) , servidor (<https://poesiabinaria.net/tag/servidor/>) , simultaneos (<https://poesiabinaria.net/tag/simultaneos/>) , socket (<https://poesiabinaria.net/tag/socket/>) , waitpid (<https://poesiabinaria.net/tag/waitpid/>)

URL for this post : <https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/> (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/>)

There are 24 comments left

[Ir a comentario](#)

Pingback: Bitacoras.com (<http://bitacoras.com/anotaciones/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/16182257>) / 6 febrero, 2011 at 10:16

Pingback: BlogEsfera.com (<http://www.blogesfera.com/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/>) / 6 febrero, 2011 at 21:32

¡Atención! Este sitio utiliza cookies propias y de terceros para mejorar la experiencia de usuario. Si continúa la navegación en el sitio, asumo que estás de acuerdo con esto, aunque siempre puedes

Fancisco / 11 diciembre, 2012 at 10:50 [desactivar las cookies en tu navegador.](#) [Aceptar](#) [Leer más \(https://poesiabinaria.net/politica-de-cookies\)](#)



Usando 🦊 Mozilla Firefox 16.0 en 🐧 Ubuntu Linux

Esta muy limpio el codigo, pero al compilarlo en la consola de linux ubuntu12.04 me da el siguiente error  
265:3: aviso: el formato no es una cadena literal y no tiene argumentos de formato [-Wformat-security]

tal vez es que no se ha declarado la variable FILE,pero aun asi da el mismo error.  
Alguien tiene idea de por que pasa esto.

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=15709#respond>)



admin / Post Author 7 enero, 2013 at 18:23

Usando 🦊 Mozilla Firefox 17.0 en 🐧 Ubuntu Linux  
Gracias Francisco.

En la línea del fprintf (), la antepenúltima pon:

```
fprintf(stderr, «%s», msg);
```

Con eso no debería darte el fallo.

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=17918#respond>)



Yeison / 28 junio, 2013 at 17:27

Usando 🌐 Google Chrome 28.0.1500.52 en 🐧 Linux  
Aun hay alguien por aca?

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=22862#respond>)



admin / Post Author 12 agosto, 2013 at 0:41

Usando 🦊 Mozilla Firefox 22.0 en 🐧 Ubuntu Linux  
@Yeison

Aunque un poco tarde, pero si, me asomo por aquí de vez en cuando 😊

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=25135#respond>)



Gerardo / 13 agosto, 2013 at 4:45

Usando 🦊 Mozilla Firefox 23.0 en 🐧 Linux  
Hola.  
Gracias por el código.

Solo tengo una duda:

Osea que ésto teoricamente atendería 3 clientes al mismo tiempo? es decir, por ejemplo: si el servidor recibiera archivos, podría recibir 3 archivos 1 por cada cliente al mismo tiempo?

Saludos 😊

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=25259#respond>)



admin / Post Author 17 agosto, 2013 at 2:47

Usando 🦊 Mozilla Firefox 22.0 en 🐧 Ubuntu Linux

Exacto, de todas formas, siempre podemos ampliar el valor de MAX\_CHILDs para que se pudieran atender más peticiones al mismo tiempo creando nuevos procesos hijos que las atiendan.

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=25930#respond>)



Calos / 30 octubre, 2013 at 6:19

Usando 🌐 Google Chrome 28.0.1500.71 en 🐧 Ubuntu Linux  
Hola...

Tengo una duda... como sería el codigo del cliente, modifiqué un poco este codigo para que resuelva un dominio con gethostbyname(); pero aun no se bien como conectar un clientes sin usar telnet.

Gracias.

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=31760#respond>)



admin / Post Author 6 diciembre, 2013 at 11:07

Usando 🦊 Mozilla Firefox 24.0 en 🐧 Ubuntu Linux

@Calos

Me has dado una buena idea para un futuro post, porque veo que es algo muy largo de explicar en un comentario. Espero tener un rato para publicarlo pronto. Muchas gracias !

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replyto=37164#respond>)

Pingback: Cliente TCP no bloqueante en C, en el que podemos enviar y recibir en cualquier momento | Poesía Binaria (<https://poesiabinaria.net/2014/01/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento/>) / 13 enero, 2014 at 10:35



Andres Felipe / 9 junio, 2014 at 5:25

Usando 🌐 Google Chrome 35.0.1916.114 en 🍏 Mac OS X 10.9.0  
@Calos

Hola, el código sería algo así:

¡Atención! Este sitio utiliza cookies propias y de terceros para mejorar la experiencia de usuario. Si continúa la navegación en el sitio, asumo que estás de acuerdo con esto, aunque siempre puedes

#include

desactivar las cookies en tu navegador.

**Aceptar**

Leer más (<https://poesiabinaria.net/politica-de-cookies>)

```
#include
#include
#include
#include
#include

#define MAXLINE 512 /*Longitud máxima de mensaje*/
#define SERV_PORT 5728 /*puerto servidor*/

int
main(int argc, char **argv)
{
    int sockfd;
    struct sockaddr_in servaddr;
    char sendline[MAXLINE], recvline[MAXLINE];

    if (argc !=2) {
        perror(«No se indicó la dirección IP del Servidor»);
        exit(1);
    }

    //Creamos socket para el cliente
    //Si sockfd<0 hay un error en la creación del socket.
    if ((sockfd = socket (AF_INET, SOCK_STREAM, 0)) <0) {
        perror("Problem creando el Socket.");
        exit(2);
    }

    //Creación del socket
    memset(&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr= inet_addr(argv[1]);
    servaddr.sin_port = htons(SERV_PORT);

    //Conexión del cliente al socket del servidor
    if (connect(sockfd, (struct sockaddr *) &servaddr, sizeof(servaddr))<0) {
        perror("Problema al intentar la conexión con el Servidor");
        exit(3);
    }

    while (fgets(sendline, MAXLINE, stdin) != NULL) {

        send(sockfd, sendline, strlen(sendline), 0);

        if (recv(sockfd, recvline, MAXLINE, 0) == 0){

            perror("La conexión con el Servidor se terminó prematuramente.");
            exit(4);
        }
        printf("Cadena recibida del servidor: %s \n",recvline);

    }

    exit(0);
}
```

Lo compilas,.. gcc -g servidor -o servidor, y lo ejecutas ./servidor

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=67454#respond>)



Andres Felipe / 9 junio, 2014 at 5:26

Usando 🌐 Google Chrome 35.0.1916.114 en 🖥️ Mac OS X 10.9.0

@Andres Felipe

Perdón, le ejecutas así: ./servidor (dir. IP servidor)

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=67455#respond>)



Gaspar Fernández / Post Author 9 junio, 2014 at 7:00

Usando 🌐 Mozilla Firefox 29.0 en 🐧 Ubuntu Linux

@Andres Felipe

Gracias por el aporte!! Los #include no salieron bien, pero bueno, nos los podemos imaginar 😊

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=67458#respond>)



Andres Felipe / 10 junio, 2014 at 16:18

Usando 🌐 Google Chrome 35.0.1916.114 en 🖥️ Windows 8

@Gaspar Fernández

hola, no había notado, aquí están.

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <sys/types.h>
4 #include <sys/socket.h>
5 #include <netinet/in.h>
6 #include <string.h>
7 #include <arpa/inet.h>
```

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=67595#respond>)



Gaspar Fernández / Post Author 17 junio, 2014 at 16:15

Usando Mozilla Firefox 29.0 en Ubuntu Linux

Muchas gracias ! 😊 ¡Qué apañado!

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=68792#respond>)



Gaspar Fernández / Post Author 17 junio, 2014 at 16:23

Usando Mozilla Firefox 29.0 en Ubuntu Linux

Por cierto, hace unos meses publiqué: <https://poesiabinaria.net/2014/01/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento/> (<https://poesiabinaria.net/2014/01/cliente-tcp-no-bloqueante-en-c-en-el-que-podemos-enviar-y-recibir-en-cualquier-momento/>) un cliente tcp no bloqueante en C. Para que le echéis un ojo

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=68794#respond>)



ingrid / 3 julio, 2014 at 15:33

Usando Google Chrome 24.0.1312.52 en Linux

Buen día, excelente post, gracias por la ayuda.

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=71102#respond>)



Gaspar Fernández / Post Author 26 julio, 2014 at 1:42

Usando Mozilla Firefox 30.0 en Linux

Gracias a ti Ingrid !

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=73779#respond>)



Pablo Alvarez (<http://www.nanocode.mx>) / 31 julio, 2014 at 0:47

Usando Internet Explorer 10.0 en Windows 8

Excelente aporte, Felicidades!!

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=74129#respond>)



wsn / 29 marzo, 2016 at 20:49

Usando Debian IceWeasel 38.7.1 en Linux

Muy buen aporte, gracias!

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=82308#respond>)



pisculichi / 18 abril, 2016 at 19:51

Usando Mozilla Firefox 45.0 en Ubuntu Linux

Excelente artículo!

Que tarde encuentre este blog, desde ahora en mi cliente rss siguiéndolos.

Saludos

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=82314#respond>)



Gaspar Fernández / Post Author 23 abril, 2016 at 13:44

Usando Mozilla Firefox 45.0 en Ubuntu Linux

Muchas gracias pisculichi. Si te gustó este, tal vez te guste: <https://poesiabinaria.net/2016/02/creando-un-cliente-para-un-servicio-de-red-con-pocas-lineas-en-c/> (<https://poesiabinaria.net/2016/02/creando-un-cliente-para-un-servicio-de-red-con-pocas-lineas-en-c/>)

Es más, tengo en preparación varios posts de creación de servicios en C y C++

Un cordial saludo

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=82315#respond>)



anonimo24 / 23 noviembre, 2019 at 19:36

Usando Google Chrome 78.0.3904.97 en Windows NT

No se si alguien siga comentando pero, si quisiera que se pudiera abrir un archivo .txt y que el cliente lo lea en su terminal, como se podría hacer? URGE!!!

Reply (<https://poesiabinaria.net/2011/02/creando-un-servidor-que-accepte-multiples-clientes-simultaneos-en-c/?replytocom=83111#respond>)

¡Atención! Este sitio utiliza cookies propias y de terceros para mejorar la experiencia de usuario. Si continúa la navegación en el sitio, asumo que estás de acuerdo con esto, aunque siempre puedes

desactivar las cookies en tu navegador. [Aceptar](#) Leer más (<https://poesiabinaria.net/politica-de-cookies>)



# Leave a Reply

Comment

Name (required)

Email (required)

Website

☐ He leído y acepto la Política de privacidad (https://poesiabinaria.net/politica-de-privacidad/) \*

Post Comment

## Otras cosas

- Sobre el blog (/poesiabinaria/informacion/)
- 
- Estadísticas (/poesiabinaria/estadisticas/)
- 
- Sobre mí (/poesiabinaria/sobre-mi/)
- 
- Contacto (/poesiabinaria/contacto/)
- 
- Aviso legal (/poesiabinaria/politica-de-privacidad)
- 

## Nube de etiquetas

aplicaciones (https://poesiabinaria.net/tag/aplicaciones/) archivos (https://poesiabinaria.net/tag/archivos/) array (https://poesiabinaria.net/tag/array/) base (https://poesiabinaria.net/tag/base/)

**Bash (https://poesiabinaria.net/tag/bash/)** c (https://poesiabinaria.net/tag/c/) codigo (https://poesiabinaria.net/tag/codigo/)

comandos (https://poesiabinaria.net/tag/comandos/) Comunicación (https://poesiabinaria.net/tag/comunicacion/) concurso (https://poesiabinaria.net/tag/concurso/)

conexión (https://poesiabinaria.net/tag/conexion/) configuración (https://poesiabinaria.net/tag/configuracion/) consola (https://poesiabinaria.net/tag/consola/) correo (https://poesiabinaria.net/tag/correo/)

**datos (https://poesiabinaria.net/tag/datos/)** Emacs (https://poesiabinaria.net/tag/emacs/) facebook (https://poesiabinaria.net/tag/facebook/)

gnu (https://poesiabinaria.net/tag/gnu/) google (https://poesiabinaria.net/tag/google/) imagen (https://poesiabinaria.net/tag/imagen/) ip (https://poesiabinaria.net/tag/ip/)

libre (https://poesiabinaria.net/tag/libre/)

**Linux (https://poesiabinaria.net/tag/linux/)**

memoria (https://poesiabinaria.net/tag/memoria/) MySQL (https://poesiabinaria.net/tag/mysql/) PHP (https://poesiabinaria.net/tag/php/)

procesos (https://poesiabinaria.net/tag/procesos/)

**programacion (https://poesiabinaria.net/tag/programacion/)**

programas (https://poesiabinaria.net/tag/programas/) red (https://poesiabinaria.net/tag/red/) script (https://poesiabinaria.net/tag/script/)

seguridad (https://poesiabinaria.net/tag/seguridad/) servicios (https://poesiabinaria.net/tag/servicios/)

**servidor (https://poesiabinaria.net/tag/servidor/)** sistema (https://poesiabinaria.net/tag/sistema/)

software (https://poesiabinaria.net/tag/software/) ssh (https://poesiabinaria.net/tag/ssh/) terminal (https://poesiabinaria.net/tag/terminal/) texto (https://poesiabinaria.net/tag/texto/)

tiempo (https://poesiabinaria.net/tag/tiempo/) tuenti (https://poesiabinaria.net/tag/tuenti/) usuario (https://poesiabinaria.net/tag/usuario/)

video (https://poesiabinaria.net/tag/video/) web (https://poesiabinaria.net/tag/web/) windows (https://poesiabinaria.net/tag/windows/)

## Últimos comentarios

- anonimo24 en Cómo configurar auto escalado asociado al balanceador de carga de Amazon Web Services (https://poesiabinaria.net/2016/12/como-configurar-auto-escalado-asociado-al-balanceador-de-carga-de-amazon-web-services/#comment-83323)
- 
- anonimo24 en Foto: Chuleta de atajos de teclado de Emacs (https://poesiabinaria.net/2017/08/foto-chuleta-atajos-teclado-emacs/#comment-83322)
- 
- anonimo24 en Crear múltiples direcciones de correo desde el CLI de Plesk (https://poesiabinaria.net/2020/03/crear-multiples-direcciones-correo-desde-cli-plesk/#comment-83321)
- 
- ~~anonimo24 en Singleton en C++ y alguna nota sobre thread safety (I) (https://poesiabinaria.net/2014/04/singleton-in-cpp/#comment-83320)~~

## Lo más comentado

---

Aventuras devolviendo una licencia de Windows que venía de serie con un portátil (<https://poesiabinaria.net/2012/08/aventuras-devolviendo-una-licencia-de-windows-que-venia-de-serie-con-un-portatil/>) (47 Comments (<https://poesiabinaria.net/2012/08/aventuras-devolviendo-una-licencia-de-windows-que-venia-de-serie-con-un-portatil/#comments>))

---

Crear mapas HTML5 interactivos con RaphaelJS (<https://poesiabinaria.net/2014/10/crear-mapas-html5-interactivos-con-raphaeljs/>) (44 Comments (<https://poesiabinaria.net/2014/10/crear-mapas-html5-interactivos-con-raphaeljs/#comments>))

---

Escribiendo en muros desde nuestra aplicación de Facebook (<https://poesiabinaria.net/2012/01/escribiendo-en-muros-desde-nuestra-aplicacion-de-facebook/>) (31 Comments (<https://poesiabinaria.net/2012/01/escribiendo-en-muros-desde-nuestra-aplicacion-de-facebook/#comments>))

---

Listar archivos dentro de un directorio o carpeta en C (<https://poesiabinaria.net/2011/09/listar-archivos-dentro-de-un-directorio-o-carpeta-en-c/>) (28 Comments (<https://poesiabinaria.net/2011/09/listar-archivos-dentro-de-un-directorio-o-carpeta-en-c/#comments>))

---

Andanzas en Facebook: Publicar en el muro personal fácilmente (<https://poesiabinaria.net/2009/08/andanzas-en-facebook-publicar-en-el-muro-personal-facilmente/>) (27 Comments (<https://poesiabinaria.net/2009/08/andanzas-en-facebook-publicar-en-el-muro-personal-facilmente/#comments>))

---

© Poesía Binaria (<https://poesiabinaria.net/>) All Rights Reserved. Theme zAlive by zenoven (<http://www.zenoven.com/>).

Información (<https://poesiabinaria.net/informacion/>) Clases de programación (<https://poesiabinaria.net/clases-particulares-de-programacion/>)  
Algoritmos (<https://poesiabinaria.net/algoritmos/>) Documentos (<https://poesiabinaria.net/documentos/>)