

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)





Hosting Patrocinado por
enREDados.com


[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)



¿Todavía gestionas tu empresa con hojas de cálculo? ¿No crees que habrá un sistema mejor? Ponemos a vuestra disposición TNTConcept, nuestra herramienta de gestión interna.

 Saber más .. <http://tntconcept.sourceforge.net>

Tutorial desarrollado por: Ángel García Jerez Consultor tecnológico de desarrollo de proyectos informáticos. Puedes encontrarme en Autentia Somos expertos en Java/J2EE	Nuevo catálogo de servicios de Autentia Descargar catálogo (6,2 MB) AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.  real business solutions Catálogo de cursos
Descargar este documento en formato PDF Vi.pdf	

Trabaje desde casa - Ingresos extra desde casa.
 Tiempo completo o parcial. www.trabajaportucuenta.com

Anuncios Google

Fecha de creación del tutorial: 2007-12-21

GUÍA DE CONSULTA DE Vi.

Introducción

Vi/Vim

2.1 Moviéndote por Vi

- 2.1.1 Desplazamientos sobre líneas.
- 2.1.2 Desplazamientos por palabras.
- 2.1.3 Desplazamientos por frases.
- 2.1.4 Desplazamientos por párrafos.
- 2.1.5 Desplazamientos por pantalla.
- 2.1.6 Desplazamientos por marcas.

2.2 Manipulación de texto

- 2.2.1 Inserción de texto.

- 2.2.2 Eliminación de texto.
- 2.2.3 Sustitución de texto.
- 2.2.4 Cortar, copiar y pegar.
- 2.2.5 Búsqueda de texto.
- 2.3 General
 - 2.3.1 Gestión de ficheros.
 - 2.3.2 Activación/Desactivación de opciones.
 - 2.3.3 Abreviaturas.
 - 2.3.4 Macros.
- 2.4 Conclusión

1. Introducción

En entorno Unix/Linux, **vi/Vim** es el editor de texto por antonomasia. Todos hemos oído hablar de él y algunos de vosotros lo habréis utilizado en la universidad o en el día a día en vuestro puesto de trabajo. Hay mucha gente que lo adora pero también existe otra que lo aborrece, en mi modesta opinión porque desconoce su potencial. Este tutorial no pretende ser un manual de referencia sino una guía de consulta que permita limar asperezas entre **vi** y aquellas persona que lo desconocen.

2. Vi/Vim

Vi/Vim, como hemos dicho anteriormente, es el editor de texto más popular en entorno Linux/Unix. Consta de dos modos: comando y edición. El modo comando permite al usuario ejecutar acciones sobre el texto, donde cada acción está ligada a una combinación de teclas (como veremos más adelante) y el modo edición similar a cualquier otro editor en el que podremos insertar y eliminar texto. El modo comando está activo desde el momento en el que se accede al **vi** a través de la línea de comandos, por tanto, determinadas combinaciones de teclas ejecutarán ciertas acciones sobre el texto. El acceso al modo edición se realiza ejecutando sobre el modo comando las acciones de insertar o añadir texto y pulsando sobre la tecla **Esc** se regresará nuevamente al modo comando. El entorno de trabajo de **vi** se divide en dos áreas (véase Figura 1.). El área coloreada en rojo, visualiza el contenido del fichero y se puede denominar zona o área de trabajo y la zona coloreada en azul, la barra de estado, donde se muestra la posición (fila, columna porcentaje%) en la que se encuentra el puntero, además del modo (si se encuentra en modo edición en la parte izquierda se visualiza el texto **'-- INSERTAR --'** o **'-- INSERT --'**).

Figura 1. Pantallazo del editor Vi

```

angel@localhost: /etc/init.d - Terminal - Konsole
}
begin()
{
    gprintf "%s" "$1"
}
}
}
if [ "$system" = "suse" ]; then
    . /etc/rc.status
    fail_msg()
    {
        rc_failed 1
        rc_status -v
    }
    succ_msg()
    {
        rc_reset
        rc_status -v
    }
    begin()
    {
        gprintf "%s" "$1"
    }
}
}
KDIR="/lib/modules/`uname -r`/misc"
DEVICE=/dev/vboxdrv
MODNAME=vboxdrv
GROUPNAME=vboxusers

fail()
{
    fail_msg
    "vboxdrv" 174L, 3157C 60.3 30%
  
```

El acceso al editor **vi** se hará desde la línea de comando de un terminal tecleando la palabra **"vi"**. Existen múltiples maneras de invocación. Cada una de ellas, permiten realizar determinadas acciones sobre el fichero o ficheros a editar. A continuación, se muestra un resumen:

Comando	Descripción
vi	Invocación por defecto, se abre un editor vacío. Si se inserta texto que posteriormente se desea grabar en un fichero, el usuario debe indicar el nombre del fichero en el que se almacenará la información.
	Invocación que edita un fichero que ya existe y si no

vi fichero	existe lo crea.
vi fichero1 fichero2 ...	Invocación que edita un número determinado de ficheros. Para movernos entre los diferentes ficheros dentro del editor utilizaremos el comando "n" más el nombre del fichero al que queremos movernos.
vi +línea fichero	Invocación que edita el fichero posicionándose en una línea determinada.
vi +/texto fichero	Invocación que edita el fichero posicionándose en la primera línea en la que se encuentre el texto deseado.

Tres son los grupos en los que se ha dividido el conjunto de comandos que **vi** dispone. El primero se ha denominado "moviéndote por **vi**", que engloba los comandos de desplazamiento por el área de trabajo. El segundo, "manipulación de texto", agrupa comandos de inserción, eliminación y sustitución de caracteres y por último, "general", que aglutina el resto de comandos. Indicar que existen un gran número de comando que para ser ejecutados en el modo comando del editor previamente se tendrá que haber pulsado la tecla **:**. A lo largo del tutorial se señalará este hecho de la siguiente manera **:**.

Nota

Si ejecutamos el comando **vimtutor** en una shell tendremos la posibilidad de practicar y aprender los comandos más habituales a través de un tutorial que dispone el editor.

2.1 Moviéndote por Vi

Existen multitud de comandos para desplazarnos sobre el texto, el más básico que todo usuario debe conocer es como realizar desplazamientos de izquierda a derecha o de arriba a abajo. Estos movimientos se podrán lograr con las teclas habituales: **←**, **↑**, **→** y **↓** o sus equivalentes **h**, **k**, **l** y **j**.

	↑		k
←		→	h
	↓		j

Sugerencia

Se recomienda que el usuario se habitúe a utilizar las teclas **h**, **k**, **l** y **j** ya que en ocasiones las teclas de desplazamiento no funcionan correctamente en algunos terminales.

Ya se ha visto como desplazarse por **vi** de forma lenta pero segura, a continuación, se muestra una lista de comandos más avanzados que permiten desplazarse por **vi** mucho más rápido. Para su correcta comprensión se incluyen en la mayoría de comandos ejemplos visuales, en los cuales el carácter que se muestre de color **azul** representa la posición actual del cursor y el color **rojo** la posición final al ejecutar el comando.

2.1.1 Desplazamientos sobre líneas.

o → Permite posicionarse al principio de la línea donde se encuentre el cursor.

Realización de un **t**utorial que muestras la bondades de un editor como **vi**.

\$ → Permite posicionarse al final de la línea donde se encuentre el cursor.

Realización de un **t**utorial que muestras la bondade**s** de un editor como **vi**.

n | → Permite posicionarse en el carácter **n** de la línea donde se encuentre el cursor.

Pulsación: **6j**

Realiz**a**ción de un **t**utorial que muestras la bondades de un editor como **vi**.

Enter ó **+** → Permite posicionarse en la siguiente línea.

Realización de un **t**utorial que muestras la bondades **d**e un editor como **vi**.

- → Permite posicionarse en la anterior línea.

R → Realización de un tutorial que muestras la bondades de un editor como vi.

n G → Permite posicionarse en la línea n del fichero.

G → Permite posicionarse en la última línea del fichero.

: 0 Enter → Permite posicionarse en la primera línea del fichero.

: n Enter → Permite posicionarse en la línea n del fichero.

: \$ Enter → Permite posicionarse en la última línea del fichero.

f c → Permite posicionarse en el siguiente carácter c que se encuentre en la actual línea.

Pulsación:fm

Realización de un tutorial que muestras la bondades de un editor como vi.

F c → Permite posicionarse en el anterior carácter c que se encuentre en la actual línea.

Pulsación:Fa

Realización de un tutorial que muestras la bondades de un editor como vi.

t c → Permite posicionarse justo antes del siguiente carácter c que se encuentre en la actual línea.

Pulsación:tm

Realización de un tutorial que muestras la bondades de un editor como vi.

T c → Permite posicionarse justo después de anterior carácter c que se encuentre en la actual línea.

Pulsación:Ta

Realización de un tutorial que muestras la bondades de un editor como vi.

2.1.2 Desplazamientos por palabras.

w → (**W**ord) Permite posicionarse en la siguiente palabra teniendo en cuenta los signos de puntuación.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

b → (**B**ack) Permite posicionarse en la siguiente palabra teniendo en cuenta los signos de puntuación.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

W → (**W**ord) Permite posicionarse en la siguiente palabra teniendo únicamente en cuenta los espacios entre las palabras.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

B → (**B**ack) Permite posicionarse en la siguiente palabra teniendo únicamente en cuenta los espacios entre las palabras.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

e → (**E**nd) Permite posicionarse al final de la palabra teniendo en cuenta los signos de puntuación.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

E → (End) Permite posicionarse al final de la palabra teniendo únicamente en cuenta los espacios entre las palabras.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

2.1.3 Desplazamientos por frases.

(→ Permite posicionarse en la siguiente frase.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

) → Permite posicionarse en la anterior frase.

A las doce y un minuto de ayer, Fernando Alonso dio por concluido su sueño de plata. Aliviado, le comentó

2.1.4 Desplazamientos por párrafos.

{ → Permite posicionarse en el párrafo siguiente.

} → Permite posicionarse en el párrafo anterior.

2.1.5 Desplazamientos por pantalla.

H → (Home) Permite posicionarse al principio de la pantalla.

M → (Middle) Permite posicionarse a mitad de la pantalla.

L → (Last) Permite posicionarse al final de la pantalla.

Ctrl u → (Up) Permite desplazar el cursor media pantalla hacia atrás.

Ctrl d → (Down) Permite desplazar el cursor media pantalla hacia delante.

Ctrl f → (Forward) Permite posicionarse en la siguiente página.

Ctrl b → (Back) Permite posicionarse en la anterior página.

% → Permite posicionarse en el signo de apertura o cierre “(,), {, }, ...” contrario.

Este es un ejemplo para posicionarse en el cierre del corchete

2.1.6 Desplazamientos por marcas.

Adicionalmente, **vi**, proporciona la capacidad de establecer marcas que permiten saltar a una zona determinada del documento rápidamente, manteniéndose activas mientras el documento se encuentre abierto.

m (a-z) → Permite establecer una marca en una determinada línea del documento. La marca estará en el rango a-z.

' (a-z) → Permite posicionarse en una determinada marca.

2.2 Manipulación de texto.

Indicar que la ejecución de la mayoría de comandos de manipulación de textos que se muestran a

continuación, el editor **vi**, cambia del modo comando al modo edición. En este modo se podrá realizar las modificaciones oportunas sobre el fichero. Finalizado las acciones deseadas se retornará al modo comando pulsando **Esc**. Se ha dividido en varios apartados esta sección, agrupando los comandos por acciones comunes.

2.2.1 Inserción de texto.

i → Permite insertar texto delante del cursor.

I → Permite insertar texto al inicio de la línea donde se encuentra el cursor.

a → Permite insertar texto detrás del cursor.

A → Permite insertar texto al final de la línea donde se encuentra el cursor.

o → Crea una nueva línea debajo de la línea actual.

O → Crea una nueva línea encima de la línea actual.

2.2.2 Eliminación de texto.

x → Elimina el carácter donde se encuentra el cursor.

X → Elimina el carácter anterior al cursor.

d d → Elimina la línea donde se encuentre el cursor. Si antes de ejecutar el comando introducimos un número, éste indica el número de líneas a eliminar.

D → Elimina la línea desde el cursor hasta el final de la línea.

d CmdMov → Permite eliminar hasta el comando de desplazamiento que se introduzca, es decir, si se introduce "\$" eliminaría hasta el final de la línea. Si se introduce "w" elimina la palabra. Para más combinaciones ir al apartado de desplazamiento por el texto.

2.2.3 Sustitución de texto.

r c → Permite sustituir el carácter actual por el teclado (c).

Pulsación: rt

Realización de un **t**utorial que muestras la bondades
Realización de un **t**utorial que muestras la bondades

R → Reemplaza el texto desde la posición actual.

Pulsación: Rejercicio **Esc**

Realización de un **t**utorial que muestras la bondades
Realización de un **e**jercicio que muestras la bondades

s → Permite sustituir el carácter actual por el texto que se desee insertar.

S → Permite sustituir la línea actual.

C → Permite sustituir desde el cursor hasta el final de la línea.

c CmdMov → Este comando es idéntico al comando "d" pero en vez de eliminar permite sustituir.

J → Permite unir la línea actual con la inmediatamente siguiente.

: s / txt / txt2 → Permite sustituir la primera aparición del texto txt por txt2 en la línea actual.

: n,m s / txt / txt2 / g → Permite sustituir todas las apariciones del texto txt por txt2 entre las líneas "n" a "m".

. → Permite repetir el último comando ejecutado.

u → Permite deshacer el último comando ejecutado.

U → Permite deshacer todos los comandos aplicados en la línea actual.

2.2.4 Cortar, copiar y pegar

Otros comandos primordiales en un editor de texto son las acciones de cortar, copiar y pegar. **vi** utiliza los comandos de eliminación como comandos de corte, almacenando en el buffer del editor el texto eliminado de la última acción. Para el resto de acciones se utilizan combinaciones de teclas específicas.

y y → Permite copiar la línea actual. De igual forma que el comando "dd": si se teclea un número antes del comando se copiarán X líneas desde la posición actual.

y p → (paste) Permite pegar el buffer de un comando anterior (cortar, copiar) antes del cursor actual.

y P → (paste) Permite pegar el buffer de un comando anterior (cortar, copiar) después del cursor.

vi permite almacenar el resultado de múltiples acciones de copia y corte. Para ello, antes de ejecutar cualquiera de estos comandos se debe pulsar " seguido de un almacén (a-z) y por último el comando. Para pegar un determinado texto de un almacén, de igual manera se teclea ", el almacén del texto a pegar y el comando de pegar (p).

"(a-z)CmdCortar
"(a-z)CmdCopiar
"(a-z)CmdPegar

2.2.5 Búsqueda de texto

/ ExpReg → Permite buscar del cursor hacia delante la expresión regular ExpReg.

? ExpReg → Permite buscar del cursor hacia atrás la expresión regular ExpReg.

n → Permite visualizar la última búsqueda realizada.

N → Permite realizar la última búsqueda en sentido contrario.

2.3 General

Para finalizar este tutorial se muestran los comandos de gestión de ficheros, opciones que pueden activarse/desactivarse en el editor, uso de abreviaturas y creación de macros.

2.3.1 Gestión de ficheros

: r fichero Enter → Permite insertar el contenido del fichero debajo de la línea actual.

: w Enter → Graba las modificaciones que se hayan efectuado en el fichero.

: w fichero Enter → Graba el texto insertado en el fichero indicado siempre que éste no exista.

: w ! fichero Enter → Graba el texto actual en el fichero indicado.

: n,m **w** fichero **Enter** → Graba en fichero indicado las líneas entre n y m.

: **e** fichero **Enter** → Descarta los cambios que se hayan hecho en el fichero actual y recupera los datos del fichero indicado.

: **q** **Enter** → Permite salir del editor siempre y cuando no se haya producido ningún cambio en el fichero.

: **q !** **Enter** → Permite salir del editor sin grabar los cambios.

: **w q** **Enter** ó : **x** **Enter** ó : **Z Z** **Enter** → Permite salir del editor pero previamente se almacenan los cambios.

^ I → Permite refrescar la pantalla.

2.3.2 Activación/Desactivación de opciones.

Para activar opciones en **vi** únicamente deberemos ejecutar ":"set opcion, donde opcion, corresponde a la opción que se quiere activar. De igual modo, para desactivar una opción, solamente deberemos ejecutar ":"set noopcion. A continuación se muestran algunas opciones que pueden resultar útiles en algún momento.

: set all **Enter** → Muestra todas las opciones que dispone **Vi** con sus valores actuales.

: set ai **Enter** → Permite que cada línea se escriba con la misma indentación que la línea anterior.

: set list **Enter** → Muestra los caracteres de control.

: set nu[mber] **Enter** → Muestra la numeración de las líneas en la parte izquierda de la pantalla.

: set vb **Enter** → Sustituye el pitido por una indicación visual.

: set wrap **Enter** → Muestra partidas las líneas que no caben en la pantalla.

: set showmode **Enter** → Muestra el modo en que se encuentra el editor, con un mensaje escrito en la parte inferior de la pantalla.

2.3.3 Abreviaturas

Son utilizadas para escribir cadenas de caracteres largas usando una combinación de teclas que previamente se haya asociado. Cada vez que en el modo edición se escriba la abreviatura configurada, se sustituirá por la cadena equivalente.

: **a b** abreviatura equivalencia **Enter** → Permite definir asignar una abreviatura a una cadena de caracteres.

Ejemplo: : ab cp Comunidad de Propietarios

: **u n a** equivalencia **Enter** → Permite cancelar una abreviatura.

Ejemplo: : una Comunidad de Propietarios

2.3.4 Macros

Con la definición de macros se podrá asociar a un carácter una secuencia de órdenes que reduce las operaciones que un usuario puede realizar en su día a día.

: **m a p** tecla secuencia_ordenes **Enter** → Permite asignar al carácter "tecla" la secuencia de órdenes determinada.

Ejemplo: :map F1 "a1Y

`:unmap` tecla `Enter` → Permite cancelar la macro "tecla"

Ejemplo: `:unmap F1`

Truco

Si se quiere que tanto las abreviaturas, macros u opciones de vi se mantenga cada vez que accedemos al editor, se deberán añadir en un fichero llamado ".exrc" situado en el directorio home del usuario. La siguiente vez que se acceda al editor Vi serán ejecutadas automáticamente sin la necesidad de volverlas a insertar manualmente.

2.4 Conclusión

Como veis, manejar un editor de texto como vi no es muy complicado, solamente deberemos memorizar un número pequeño de comandos y nuestro día a día con el editor será menos difícil. Para ir memorizando poco a poco los comandos más importantes, os podéis imprimir este [pdf](#) como chuleta.

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en eConozco [haciendo clic aquí](#).



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](#)

Recuerda

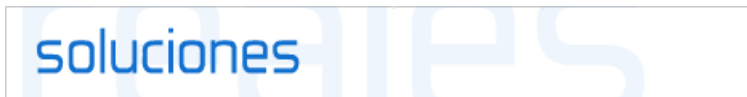
Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de suscripción a novedades:

E-mail

Otros Tutoriales Recomendados.

Nombre corto	Descripción
Gestionando nuestro tiempo, en GNU/Linux Debian	En este tutorial vamos a ver el funcionamiento de una herramienta llamada Karm que nos permitirá dar de alta varias tareas y subtareas, y medir el tiempo que dedicamos a cada una.
Editor VI en Windows	Os mostramos como conseguir e instalar el famoso editor de Unix VI para vuestra plataforma Windows. Utilizando habitualmente esta herramienta, nuestro trabajo multi-plataforma será menos traumático..
	En este tutorial, Alejandro Perez nos enseña como configurar la red Wireless WIFI de tu centrino en

Wireless Centrino en Linux Debian	Linux (Debian)
Recordar primeros pasos por Linux	En este tutorial recordamos, de un modo visual, los primeros comandos en Linux sobre una emulación Qemu.
Compilar en C y Linux con make (makefile)	Os recordamos como construir ficheros makefile en entorno Linux para compilar vuestros programas C
Backup en Linux	En este tutorial os mostramos cómo realizar copias de seguridad en un sistema Linux, utilizando sencillas herramientas del propio sistema operativo
Backup Manager: Herramienta de Backup en sistema linux	En este tutorial se explica el funcionamiento de Backup Manager, una herramienta muy completa y sencilla para la realización de copias de seguridad de sistema de ficheros locales y remotos, bases de datos, etc...
Edición de ficheros RSS con RssEditor	Os mostramos como crear y administrar los ficheros de canales RSS a través de la extensión gratuita de FireFox RssEditor
Wireless en Debian GNU/Linux con iwlmwifi para las tarjetas Intel Wireless WiFi Link 4965AGN e Intel PRO/Wireless 3945AB, y el NetworkManager	En este tutorial vamos a ver otra posibilidad de configurar la wireless para ordenadores con tarjeta 3945 (por ejemplo el Asus G1) o 4965 (por ejemplo el Asus G1S) de Intel
Virtualización en Debian GNU/Linux con KVM	En este tutorial vamos a ver como podemos sacar provecho de la virtualización por hardware con el módulo KVM
Ver todos los tutoriales	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



**¿Buscas un hospedaje de calidad
por sólo 2 € al mes?**