

**CREATIVE COMMONS**



0 - Unidad didáctica 0. Unidad introductoria.....	3
0.1 - Presentación y definición del software libre.....	3
0.1.1 - Historia y definición del software libre. ....	3
0.1.2 - Motivaciones para su creación y sostenibilidad. ....	7
0.1.3 - Comparación con el software propietario.....	7
0.1.4 - Beneficios del código abierto .....	8
0.2 El sistema operativo GNU Linux.....	8
0.2.1 - Historia de Linux .....	8
0.2.2 - Utilidades y aplicaciones .....	12
0.2.3 - Versiones existentes.....	16
0.3 - Guadalinux, la distribución andaluza .....	19
0.3.1 - Características de esta distribución. ....	19
0.3.2 - Razones para el apoyo político al software libre. ....	19
0.3.3 - Múltiples usos en nuestra comunidad .....	20
1 - Unidad didáctica 1. Internet.....	21
1.1 - Navegar por Internet .....	21
1.2 Buscar información .....	22
1.3 Marcadores.....	28
1.4 Correo electrónico .....	31
1.5 Ejercicios .....	36
2 - Unidad didáctica 2. Maquetación de páginas web (NVU) .....	38
2.1 - El entorno de trabajo .....	38
2.2 Configurar un sitio remoto.....	42
2.3 Configurar las propiedades de la página .....	45
2.4 Textos .....	47
2.5 Vínculos .....	49
2.6 Imágenes .....	50
2.7 Tablas.....	51
2.8 Formularios .....	52
2.9 Frames .....	57
2.10 Ejercicios.....	58
3 - Unidad didáctica 3. Programación en PHP y MySQL .....	60
3.1 - Sintaxis de PHP .....	60
3.2 Variables y operadores.....	61
3.3 Sentencias de control. Vectores .....	68
3.4 Las funciones include() y require() .....	75
3.5 Funciones PHP de acceso a MySQL .....	76
3.6 Creación de una base de datos en MySQL.....	84
3.7 Mostrar los datos de una consulta .....	86
3.8 Añadir, modificar y borrar registros .....	87
3.9 Ejercicios .....	89



## O - Unidad didáctica O. Unidad introductoria

### O.I - Presentación y definición del software libre

#### O.I.I - Historia y definición del software libre.

##### Un poco de historia...

Entre los años 60 y 70 del siglo XX, el software no era considerado un producto sino un añadido que los vendedores de los grandes computadores de la época (los llamados **mainframes**) aportaban a sus clientes para que éstos pudieran usarlos.



**Figura 1:** Mainframe IBM S/360 Modelo 91, desarrollado en Centro de Desarrollo de IBM (Böblingen)

En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como **DECUS** (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

##### **Definición de software libre (*free software*)**

El término **free**, traducido al castellano, significa tanto “libre” como “gratis”, por eso muchas veces suelen confundirse el freeware con el software libre aunque entre ambos existen notables diferencias, que detallamos a continuación.

**Software libre** (en inglés **free software**) es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible de forma gratuita en la red Internet o a precio del coste de la distribución a través de otros medios.

Entonces, ¿el software libre siempre es software gratuito? No es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente. Análogamente, el software gratuito (denominado usualmente **freeware**) incluye en algunas ocasiones el código fuente. Pero a diferencia del software libre, el freeware **no es libre** en el mismo sentido que el software libre, al menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Por otro lado, no debe confundirse software libre con software de **dominio público**. Éste último es aquél por el que no es necesario solicitar ninguna licencia y cuyos derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público.

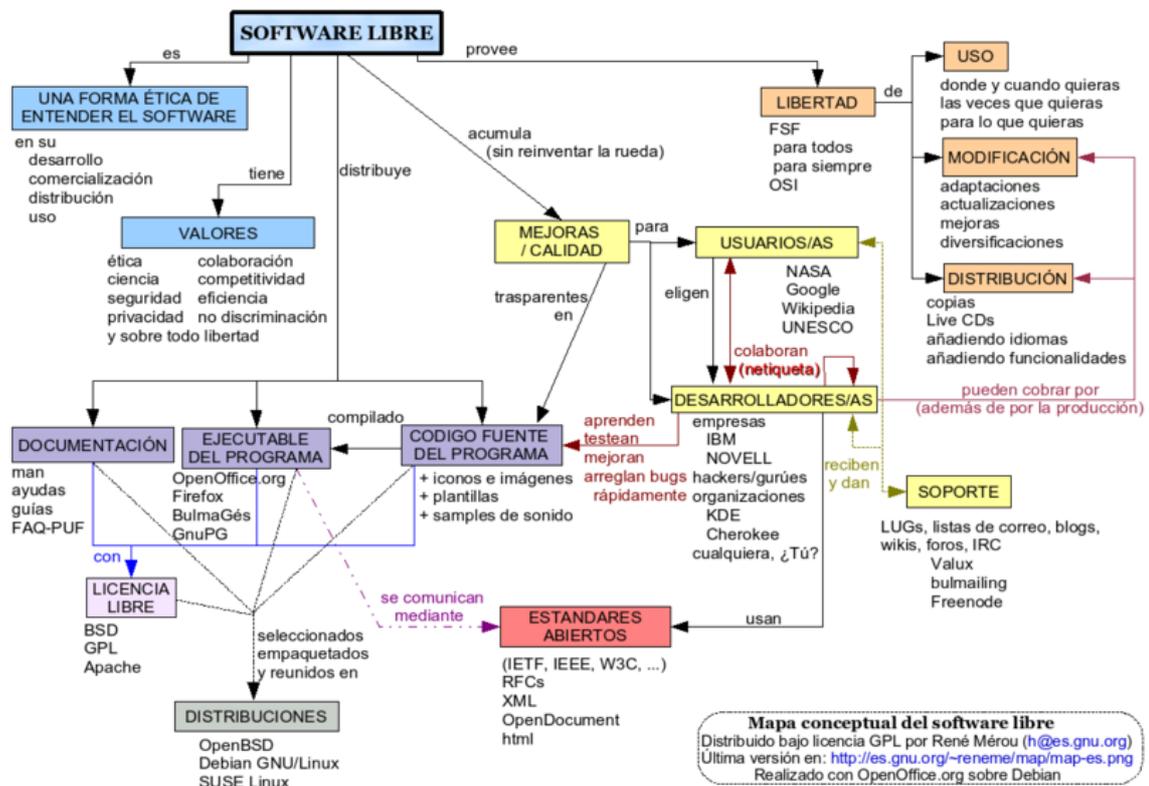


Figura 2: Mapa conceptual del software libre



## Libertades del software libre

De acuerdo con la definición antes expuesta, el software es "libre" si garantiza las siguientes libertades:

- **Libertad 0:** ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.)
- **Libertad 1:** estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente)
- **Libertad 2:** copiar el programa de manera que se pueda ayudar al vecino o a cualquiera
- **Libertad 3:** mejorar el programa, y hacer públicas las mejoras, de forma que se beneficie toda la comunidad

Es importante señalar que las libertades 1 y 3 obligan a que se tenga acceso al código fuente. La "**Libertad 2**" hace referencia a la libertad de modificar y redistribuir el software libremente licenciado bajo algún tipo de licencia de software libre que beneficie a la comunidad.

## Tipos de licencias

Una **licencia** es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatarlo. Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias:

- Las libertades definidas anteriormente están protegidas por licencias de software libre, de las cuales una de las más utilizadas es la **Licencia Pública General GNU (GPL)**. El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que no sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.
- **Licencias BSD**, llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles.

Puede argumentarse que esta licencia asegura "verdadero" software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre.

- **Licencias estilo MPL y derivadas:** esta licencia es de software libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del software libre: el navegador web Mozilla.

Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. La MPL es software libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si usas código licenciado GPL, tu desarrollo final tiene que estar licenciado GPL). Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y lamentablemente mucha gente se cierra en banda ante el uso de dicho código. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD.

Estas licencias son denominadas **de copyleft débil**. La **NPL** (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se le puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

Hay que hacer constar que el titular de los derechos de autor (copyright) de un software bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan software libre, pues esta práctica no restringe ninguno de los derechos otorgados a los usuarios de la versión copyleft.

También podría retirar todas las licencias de software libre anteriormente otorgadas, pero esto obligaría a una indemnización a los titulares de las licencias en uso. En España, toda obra derivada está tan protegida como una original, siempre que la obra derivada parta de una autorización contractual con el autor. En el caso genérico de que el autor retire las licencias "copyleft", no afectaría de ningún modo a los productos derivados anteriores a esa retirada, ya que no tiene efecto retroactivo. En términos legales, el autor no ha derecho a retirar el permiso de una licencia en vigencia. Si así sucediera, el conflicto entre las partes se resolvería en un pleito convencional



## O.I.2 - Motivaciones para su creación y sostenibilidad.

Una de las características del software libre es no solamente que el usuario tiene libertad para modificar el código -- adaptarlo a sus necesidades específicas --, sino también haber difundido masivamente un modelo de desarrollo cooperativo y comunitario del software, que se revela hoy mucho más eficiente que la lógica propietaria de las grandes empresas del sector.

El software libre es ante todo una cuestión de **libertad** y de **comunidad**. El software libre incentiva la cooperación entre usuarios y desarrolladores libremente. Esta libertad tiene su consecuencia en una mejora constante de la calidad de las distribuciones y aplicaciones desarrolladas, de los cual se benefician directamente los usuarios finales.

Además el uso extendido de aplicaciones de software libre supone una reducción de costes para las empresas, que se evitan unas fuertes inversiones en tecnología.

## O.I.3 – Comparación con el software propietario.

Una vez analizadas las características principales del software libre, vamos a ver los aspectos que determina el software no libre o propietario.

El **software no libre** (también llamado software propietario, software privativo, software privado, software con propietario o software de propiedad) se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

Para la Fundación de Software Libre (**FSF**) este concepto se aplica a cualquier software que no es libre o que sólo lo es parcialmente (semilibre), bien porque su uso, redistribución o modificación está prohibida, o bien porque requiere permiso expreso del titular del software.

En el software no libre una persona física o jurídica (compañía, corporación, fundación, etc.) posee los derechos de autor sobre un software negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito, de estudiar cómo funciona el programa y adaptarlo a las propias necesidades (donde el acceso al código fuente es una condición previa), de distribuir copias o de mejorar el programa y hacer públicas las mejoras (para esto el acceso al código fuente es un requisito previo).

De esta manera, un programa sigue siendo no libre aún si el código fuente se hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución (por ejemplo, el programa de licencias **shared source de Microsoft**).

## O.1.4 – Beneficios del código abierto

---

El código abierto (en inglés, **open source**) posee una serie de beneficios, entre los cuales destacamos:

- Ahorro considerable en inversión necesaria en aplicaciones software
- Las herramientas open source son capaces de proporcionar la misma funcionalidad que las versiones alternativas de pago.
- Siempre está sujeto a continuas revisiones y mejoras, como consecuencia del creciente nº de usuarios que desarrollan y depuran estas aplicaciones.
- Proporciona libertad e independencia tecnológica a nivel personal e institucional.



**Figura 3:** Logotipo representativo de código abierto (open source)

---

## O.2 El sistema operativo GNU Linux

---

### O.2.1 - Historia de Linux

---

#### El proyecto GNU

El **proyecto GNU** fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completo libre: el **sistema GNU**. El 27 de Septiembre de 1983 se anunció públicamente el proyecto por primera vez en el grupo de noticias net.unix-wizards. Al anuncio original, siguieron otros ensayos escritos por Richard Stallman como el "Manifiesto GNU", que establecieron sus motivaciones para realizar el proyecto GNU, entre las que destaca "volver al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras". GNU es un acrónimo recursivo que significa "**GNU no es Unix**". Stallman sugiere que se pronuncie, en inglés como "guh-noo" (se puede observar que el logo es un ñú) para evitar confusión con "new" (nuevo). En español, GNU se pronuncia fonéticamente.



**Figura 4:** Logotipo representativo del proyecto GNU



### Acerca de Unix...

UNIX es un sistema operativo no libre muy popular, porque está basado en una arquitectura que ha demostrado ser técnicamente estable. El sistema GNU fue diseñado para ser totalmente compatible con UNIX. El hecho de ser compatible con la arquitectura de UNIX implica que GNU esté compuesto de pequeñas piezas individuales de software, muchas de las cuales ya estaban disponibles, como el sistema de edición de textos TeX y el sistema gráfico X-Window, que pudieron ser adaptados y reutilizados.

Para asegurar que el software GNU permaneciera libre para que todos los usuarios pudieran "ejecutarlo, copiarlo, modificarlo y distribuirlo", el proyecto debía ser liberado bajo una licencia diseñada para garantizar esos derechos al tiempo que evitase restricciones posteriores de los mismos. La idea se conoce en inglés como **copyleft** - "izquierdo de copia" - (en clara oposición a **copyright** - "derecho de copia"), y está contenida en la Licencia General Pública de GNU (GPL).

### El sistema operativo GNU/Linux

**GNU/Linux** (GNU con Linux) es la denominación defendida por Richard Stallman y otros para el sistema operativo que utiliza el núcleo (kernel) Linux en conjunto con las aplicaciones de sistema creadas por el proyecto GNU. Comúnmente este sistema operativo es denominado como Linux, aunque según Stallman esta denominación no es correcta.



**Figura 5:** Richard Matthew Stallman. Programador y creador del proyecto GNU.

Desde 1984, Richard Stallman y voluntarios están intentando crear un sistema operativo libre con un funcionamiento similar al Unix, recreando todos los componentes necesarios para tener un sistema operativo funcional que se convertiría en el sistema operativo GNU. En el comienzo de los años 90, después de seis años, GNU tenía muchas herramientas importantes listas, como compiladores, depuradores, intérpretes de órdenes etc. excepto por el componente central: el núcleo.

Con el surgimiento del kernel Linux, esta laguna fue llenada y surgió el sistema operativo con el kernel Linux en conjunto con las herramientas GNU. De esta manera, Stallman juzga que este sistema operativo es una "versión modificada" del sistema GNU y por lo tanto debe tener la denominación GNU/Linux. Esta denominación resolvería la confusión entre el núcleo y el sistema operativo completo a que puede llevar, y de hecho ha llevado, la denominación Linux en solitario. Stallman también espera que con el aporte del nombre GNU, se dé al proyecto GNU que él encabeza el reconocimiento que merece por haber creado las aplicaciones de sistema imprescindibles para ser un sistema operativo compatible con UNIX.

Algunas distribuciones apoyan esta denominación, e incluyen los términos GNU/Linux en sus nombres, tal es el caso de Debian GNU/Linux o GNU/Linux.

Algunos sectores de la comunidad de usuarios del sistema operativo han rechazado la denominación GNU/Linux por varias razones, entre ellas que ya se había empezado a denominar Linux al sistema operativo antes de que Richard Stallman promocionase esta denominación. Otras personas se oponen a la postura ideológica de Stallman radicalmente en contra del software no libre y por ello son contrarios al uso de este nombre para evitar la promoción de las ideas del fundador del proyecto GNU. Otros sectores de la comunidad han reconocido la conveniencia de este nombre.

Hay que señalar que, al igual que es una simplificación denominar al sistema que usa el usuario final Linux, obviando las aplicaciones GNU que completan el sistema operativo, el conjunto GNU + Linux representa solamente una parte (aunque importante) del software encontrado en una distribución Linux. Existe una gran cantidad de software original producido independientemente de los proyectos GNU y Linux por otras personas u organizaciones, como por ejemplo Apache, KDE, Samba u OpenOffice.org entre otros.



## Linux

Linux es la denominación de un sistema operativo y el nombre de un núcleo. Es uno de los paradigmas del desarrollo de software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.



**Figura 6:** Tux, el logo y mascota de Linux

El término Linux estrictamente se refiere al núcleo Linux, pero es más comúnmente utilizado para describir al sistema operativo tipo Unix, que utiliza primordialmente filosofía y metodologías libres (también conocido como GNU/Linux) y que está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software libre y no libre). El núcleo no es parte oficial del proyecto GNU (el cual posee su propio núcleo en desarrollo, llamado **Hurd**), pero es distribuido bajo los términos de la licencia GNU GPL.

La expresión Linux también es utilizada para referirse a las distribuciones GNU/Linux, colecciones de software que suelen contener grandes cantidades de aplicaciones además del núcleo (entornos gráficos, suites ofimáticas, servidores web, etc.). Coloquialmente se aplica el término Linux a éstas, aunque en estricto rigor sea incorrecto.

La marca Linux pertenece a Linus Torvalds y se define como "*un sistema operativo para computadoras que facilita su uso y operación*". Actualmente Linus supervisa el uso (o abuso) de la marca a través de la organización sin fines de lucro Linux International.



**Figura 7:** Linus Torvalds, creador del núcleo de Linux.

0.2.2 – Utilidades y aplicaciones

REDES Y CONECTIVIDAD		
TIPO DE APLICACIÓN	Windows	Linux
Navegadores web	<ul style="list-style-type: none"> <li>- Internet Explorer</li> <li>- Netscape / Mozilla</li> <li>- Opera</li> <li>- Phoenix para Windows</li> </ul>	<ul style="list-style-type: none"> <li>- Netscape, Mozilla</li> <li>- Galeon</li> <li>- Konqueros</li> <li>- Opera</li> <li>- Phoenix</li> <li>- Nautilus</li> <li>- Epiphany</li> <li>- Links</li> <li>- Dillo</li> </ul>
Navegadores web en modo consola	Lynx para Windows	<ul style="list-style-type: none"> <li>- Links</li> <li>- Lynx</li> <li>- w3m</li> <li>- Xemacs</li> </ul>
Cientes de e-mail	<ul style="list-style-type: none"> <li>- Outlook Express</li> <li>- Mozilla</li> <li>- Eudora</li> <li>- Becky</li> </ul>	<ul style="list-style-type: none"> <li>- Evolution</li> <li>- Netscape / Mozilla</li> <li>- Sylpheed, Sylpheed-claws</li> <li>- Kmail</li> <li>- Gnus</li> <li>- Balsa</li> <li>- Arrow</li> <li>- Gnumail</li> <li>- Althea</li> </ul>
Lectores de noticias (news readers)	<ul style="list-style-type: none"> <li>- Xnews</li> <li>- Outlook</li> <li>- Netscape / Mozilla</li> </ul>	<ul style="list-style-type: none"> <li>- Knode</li> <li>- Pan</li> <li>- NewsReader</li> <li>- Netscape / Mozilla</li> <li>- Pine</li> <li>- Mutt</li> <li>- tin</li> <li>- Xemacs</li> </ul>
Gestor de descargas	<ul style="list-style-type: none"> <li>- FlashGet</li> <li>- Go!zilla</li> <li>- Reget</li> <li>- Getright</li> <li>- Wget para Windows</li> </ul>	<ul style="list-style-type: none"> <li>- Downloader for X</li> <li>- Caitoo</li> <li>- Prozilla</li> <li>- Wget</li> <li>- Aria</li> <li>- Axel</li> </ul>
Cientes FTP	<ul style="list-style-type: none"> <li>- FTP in far</li> <li>- SmartFTP</li> <li>- CuteFTP</li> </ul>	<ul style="list-style-type: none"> <li>- Gftp</li> <li>- Kbear</li> <li>- IglooFTP</li> <li>- Nftp</li> <li>- Wxftp</li> </ul>
Cientes IRC	<ul style="list-style-type: none"> <li>- Mirc</li> <li>- Klient</li> <li>- VIRC</li> <li>- Xircon</li> <li>- Pirch</li> </ul>	<ul style="list-style-type: none"> <li>- Xchat</li> <li>- KVirC</li> <li>- Irssi</li> <li>- BitchX</li> <li>- Epic</li> </ul>
Mensajería local con máquinas	WinPopup	



Windows		<ul style="list-style-type: none"><li>- LinPopup</li><li>- Kpopup</li></ul>
Clientes para mensajería instantánea	<ul style="list-style-type: none"><li>- ICQ</li><li>- MSN</li><li>- AIM</li><li>- Trillian ICQ</li></ul>	<ul style="list-style-type: none"><li>- Licq</li><li>- Alicq</li><li>- Gaim</li><li>- Kopete</li><li>- Everybuddy</li><li>- aMSN</li></ul>
Conferencias audio / vídeo	Netmeeting	Gnomemeeting
Comunicación por voz	Speak Freely	<ul style="list-style-type: none"><li>- Speak Freely for Unix</li><li>- TeamSpeak</li></ul>
Firewall (cortafuegos)	<ul style="list-style-type: none"><li>- BlackICE</li><li>- ATGuard</li><li>- ZoneAlarm</li><li>- Agnitum Outpost</li><li>- Winroute PRO</li></ul>	<ul style="list-style-type: none"><li>- Kmyfirewall</li><li>- Firewall builder</li><li>- Shorewall</li><li>- Guarddog</li><li>- Firestarter</li></ul>

## TRABAJO CON FICHEROS

TIPO DE APLICACIÓN	Windows	Linux
Administrador de Archivos al estilo FAR y NC	<ul style="list-style-type: none"><li>- FAR</li><li>- Norton Commander</li></ul>	<ul style="list-style-type: none"><li>- Midnight Commander</li><li>- X Northern Captain</li><li>- Deco</li><li>- Portos Commander</li><li>- Konqueror</li></ul>
Administrador de archivos al estilo Windows	Windows Explorer	<ul style="list-style-type: none"><li>- Konqueror</li><li>- Gnome-Commander</li><li>- Nautilus</li><li>- XWC</li></ul>
Inspección rápida de ficheros HTML locales	Internet Explorer	<ul style="list-style-type: none"><li>- Dillo</li><li>- Konqueror</li><li>- Nautilus</li><li>- Lynx / Links</li></ul>

**SOFTWARE DE ESCRITORIO**

TIPO DE APLICACIÓN	Windows	Linux
Editor de textos	Ntepad, Wordpad, Textpad	<ul style="list-style-type: none"> <li>- KEdit</li> <li>- GEdit</li> <li>- Gnotepad</li> <li>- Kate</li> <li>- Kwrite</li> <li>- Vim</li> <li>- Xemacs</li> </ul>
Editor de texto modo consola	FAR Editor	<ul style="list-style-type: none"> <li>- Vim</li> <li>- Emacs</li> <li>- pco</li> <li>- je</li> <li>- Jed</li> </ul>
Compresores	Winzip, Winrar	<ul style="list-style-type: none"> <li>- FileRoller</li> <li>- Gnozip</li> <li>- Linzip</li> <li>- RAR for Linux</li> </ul>
Antivirus	AVG, Dr. Web, Trendmicro	<ul style="list-style-type: none"> <li>- Dr. Web for Linux</li> <li>- RAV Antivirus</li> <li>- OpenAntivirus</li> <li>- VirusHammer</li> </ul>
Configuración del sistema	Msconfig	<ul style="list-style-type: none"> <li>- Linuxconf</li> <li>- Webmin</li> <li>- Yast, Yast2</li> <li>- RAR for Linux</li> </ul>
Software para backup	<ul style="list-style-type: none"> <li>- ntbackup</li> <li>- Legato Networker</li> </ul>	<ul style="list-style-type: none"> <li>- Lonetar</li> <li>- Disk archive</li> <li>- Bacula</li> <li>- Taper</li> </ul>
Administrador de tareas	Taskman, Taskinfo	<ul style="list-style-type: none"> <li>- Top</li> <li>- Gtop, Ktop</li> <li>- kSysGuard</li> </ul>
Reconocimiento de voz	ViaVoice	Sphinx
Recuperación de datos	R-Studio	<ul style="list-style-type: none"> <li>- e2undel</li> <li>- myrescue</li> <li>- TestDisk</li> </ul>



## MULTIMEDIA

TIPO DE APLICACIÓN	Windows	Linux
Reproductores de música (mp3, ogg)	Winamp	<ul style="list-style-type: none"><li>- XMMS</li><li>- Noatun</li><li>- Zinf</li><li>- SnackAmp</li></ul>
Grabación de CDs, DVDs	Nero, Roxio, Easy CD Creator	<ul style="list-style-type: none"><li>- K3b</li><li>- XCDRoast</li><li>- KOnCd</li><li>- GCombust</li><li>- WebCDWriter</li></ul>
Reproductores de CD	CD Player	<ul style="list-style-type: none"><li>- ksCD</li><li>- Oprheus</li><li>- Sadp</li><li>- Workman</li></ul>
Decodificadores MP3	Lame	<ul style="list-style-type: none"><li>- Lame</li><li>- Bladeenc</li><li>- NotLame</li><li>- gogo</li></ul>
Editores de audio	Soundforge, Cooledit	<ul style="list-style-type: none"><li>- Sweep</li><li>- Waveforge</li><li>- Sox</li><li>- Audacity</li></ul>
Secuenciador MIDI	CakeWalk	<ul style="list-style-type: none"><li>- RoseGarden</li><li>- Brahms</li><li>- Anthem</li><li>- Melys</li></ul>

## DISEÑO GRÁFICO – RETOQUE FOTOGRÁFICO

TIPO DE APLICACIÓN	Windows	Linux
Visualizador de archivos gráficos	ACDSee, Irfanview	<ul style="list-style-type: none"><li>- Xnview</li><li>- GQView</li><li>- Qiv</li><li>- Compupic</li></ul>
Editores simples	Paint	<ul style="list-style-type: none"><li>- Kpaint</li><li>- TuxPaint</li></ul>
Editores complejos	Adobe Photoshop	The Gimp
Secuenciador MIDI	CakeWalk	<ul style="list-style-type: none"><li>- RoseGarden</li><li>- Brahms</li><li>- Anthem</li><li>- Melys</li></ul>

### 0.2.3 – Versiones existentes

Linux es un sistema de libre distribución, por lo que podemos encontrar todos los ficheros y programas necesarios para su funcionamiento en multitud de servidores conectados a Internet.

La tarea de reunir todos los ficheros y programas necesarios, así como instalarlos en nuestro sistema y configurarlos puede resultar una tarea bastante complicada. Debido a esto nacieron las llamadas **distribuciones de Linux (Linux distros)**, es decir, empresas y organizaciones que se dedica a hacer el trabajo "sucio" para nuestro beneficio y comodidad.

Una distribución no es otra cosa, que una recopilación de programas y ficheros, organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet, o comprando los CDs de las mismas, los cuales contendrán todo lo necesario para instalar un sistema Linux bastante completo y en la mayoría de los casos un programa de instalación que nos ayudara en la tarea de una primera instalación. Casi todos los principales distribuidores de Linux, ofrecen la posibilidad de bajarse sus distribuciones, via FTP (sin cargo alguno).

Existen muchas y variadas distribuciones creadas por diferentes empresas y organizaciones a unos precios bastante asequibles.

A continuación se muestra información relativa a las distribuciones más importantes de Linux (aunque no las únicas):

#### DISTRIBUCIONES LINUX

	Debian	Distribución con muy buena calidad. El proceso de instalación es quizás un poco mas complicado, pero sin mayores problemas. Gran estabilidad antes que últimos avances.
	RedHat	Esta es una distribución que tiene muy buena calidad, contenidos y soporte a los usuarios por parte de la empresa que la distribuye. Es necesario el pago de una licencia de soporte. Enfocada a empresas.
	Fedora	Esta es una distribución patrocinada por RedHat y soportada por la comunidad. Fácil de instalar.
	Gentoo	Esta distribución es una de las únicas que últimamente han incorporado un concepto totalmente nuevo en Linux. Es un sistema inspirado en BSD-ports. Se puede compilar/optimizar el sistema completamente desde cero. No es recomendable adentrarse en esta distribución sin una buena conexión a Internet, un ordenador medianamente potente y cierta experiencia en sistemas Unix.
	SuSe	Otra de las grandes. Calidad germana. Fácil de instalar.



Existen además unas versiones de las distintas distribuciones de Linux, llamadas **LiveCD** que sirven para probar como funciona un sistema Linux, sin necesidad de instalaciones ni espacio libre en el disco duro.

Un LiveCD no es otra cosa que una distribución de Linux que funciona al 100%, sin necesidad de instalarla en el ordenador donde la probamos. Utiliza la memoria RAM del ordenador para "instalar" y arrancar la distribución en cuestión. En la memoria también se instala un "disco virtual" que emula al disco duro de un ordenador.

De esta forma sólo hace falta introducir el CD o DVD en el ordenador en cuestión y arrancarlo, al cabo de unos minutos tendremos un sistema Linux funcionando en el mismo. Este tipo de distribuciones solamente sirve para demostraciones y pruebas, ya que una vez que apagamos el ordenador, todo lo que hemos hecho desaparece.

Algunas distribuciones tipo "LiveCD" vienen también con la opción de instalación una vez que la hemos probado.

Por último se ofrece una gráfica con las distribuciones más comunes, ordenadas cronológicamente por orden de aparición en el mercado, desde el año 1991 hasta hoy.

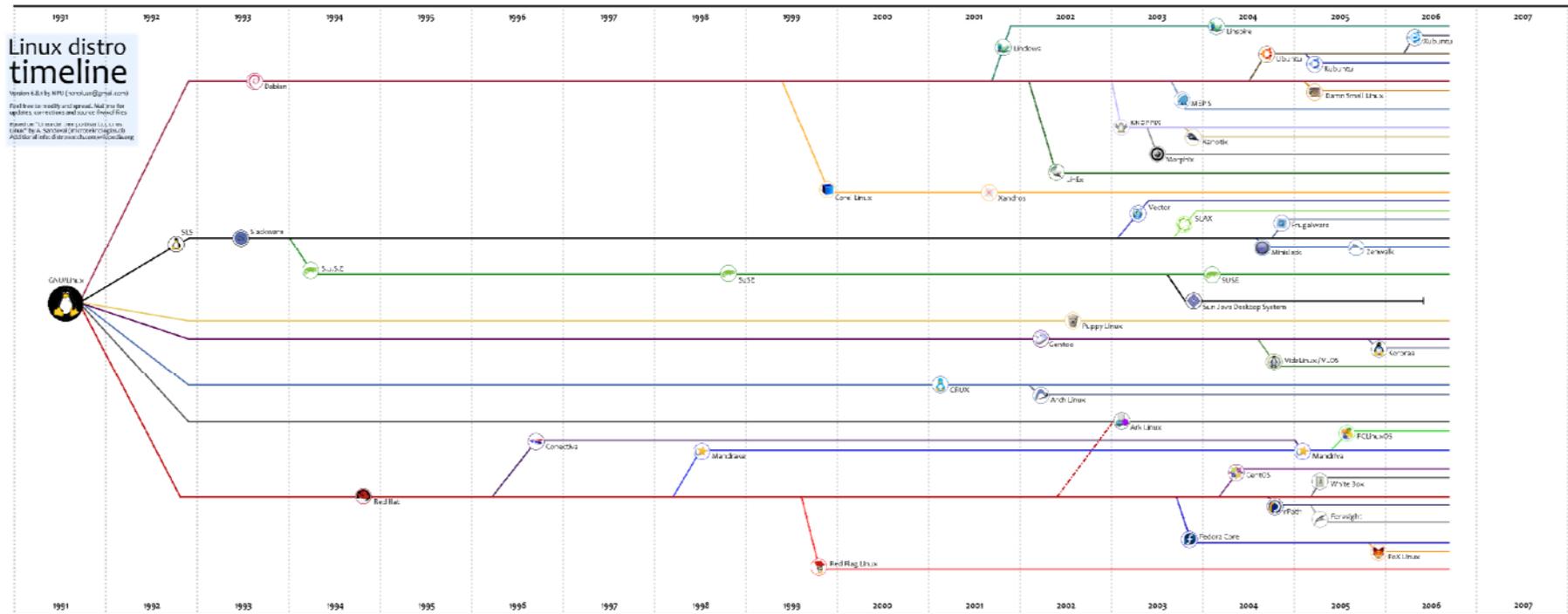


Figura 8: Cronología de distribuciones Linux (1991 – 2007)



---

## 0.3 – Guadalinux, la distribución andaluza

---

### 0.3.1 – Características de esta distribución.

---

**Guadalinux** es una distribución Linux promovida por la Junta de Andalucía para fomentar el uso del software libre dentro de Andalucía. Está inspirada en un proyecto similar de la Junta de Extremadura: GnuLinEx. Estuvo basada en la distribución Debian GNU/Linux gracias al acuerdo inicial con la Junta de Extremadura, si bien desde la versión actual (**v3**) se basa en Ubuntu.

Entre las características más relevantes de esta distribución destacamos:

- Incluye todos los programas habituales para el uso habitual del ordenador como: aplicaciones de oficina, Internet, diseño gráfico, multimedia, juegos...
- Se puede probar sin necesidad de instalación, como LiveCD.
- También es posible instalarla en el disco duro, como sistema operativo único o compartiendo disco con otros sistemas cualesquiera.
- Cuenta con el respaldo de la Junta de Andalucía y con una importante comunidad de usuarios en los foros de Guadalinux



**Figura 9:** Logotipo de distribución Guadalinux

---

### 0.3.2 – Razones para el apoyo político al software libre.

---

El propósito de la Junta es dar a conocer el software libre por las ventajas y valores que representa.

El software libre no tiene un problema de partidarios. Es sorprendente numerosa la cantidad y actividad de los grupos de usuarios de software libre a lo largo de toda nuestra geografía. Por poner una medida real, si se reúnen todos los desarrolladores de software libre de España, superarían a la mayor empresa de creación de software de este país.

Paradigma de esta situación a nivel internacional esta sourceforge.net. en este sitio web hay más de 1.050.000 'desarrolladores' registrados y casi 100.000 proyectos arrancados (un proyecto normalmente representa una aplicación). Por poner un ejemplo es seguro que Microsoft no tiene ni 30.000 programadores entre sus casi 55.000 empleados.

Otro problema que no tiene el software libre es falta de apoyo político. Al menos, en España, hay un número razonable de políticos, tanto del PSOE, como del PP, de IU, de ERC, etc que han usado el software libre como bandera y herramienta para la popularización de la sociedad de la información.

No solo seis comunidades autónomas han creado su propia distribución de software libre, sino que el resto han coqueteado en mayor o menor medida con el uso y/o promoción del software libre.

inalmente tampoco es un problema de un monopolio preexistente en el mercado. Aunque es grande, desde luego no es el obstáculo definitivo a la difusión del software libre. Como ejemplo, el lenguaje Java ha pasado a ser casi el lenguaje numero 1 del mercado desde el año 95 en que apareció en el mercado, y presenta pequeñas ventajas frente a sus competidores. El software libre presenta grandes ventajas y Linux empezó a crearse en el 91 y Richard Stallman está apoyando al software libre desde la FSF desde el año 85.

### **0.3.3 – Múltiples usos en nuestra comunidad**

---

Existen varios "sabores" de Guadalinux, según a qué público esté orientada:

- **Guadalinux Base**, de propósito general, que se publica una vez al año.
- **Guadalinux EDU**: para los centros educativos.
- **Guadalinux CDM**: para los Centros de Día de Mayores.
- **Guadalinux BIB**: para Bibliotecas (actualmente en desarrollo).
- **Guadalinux Mini**: para ordenadores antiguos.
- **Guadalinux Guadalinfo**: para los centros Guadalinfo de la Junta de Andalucía



## I - Unidad didáctica I. Internet

### I.I - Navegar por Internet

La red Internet está constituida por una red de redes a escala mundial de millones de ordenadores interconectados que se comunican por medio de una serie de protocolos de comunicación para el intercambio de información.

De cara al usuario final, Internet se traduce en un conjunto de millones de páginas web accesibles para la consulta y búsqueda de información.

Las páginas web son creadas a partir de una serie de lenguajes de marcado que permiten definir la estructura y los contenidos de una página. (ver **Figura 1**)

```
<body>
<div id="container">
<div id="navegacion">
<ul><li><a href="http://www.microsiervos.com/retro.html">Versión Texto</a></li>
<li><a href="http://www.microsiervos.com/contacto.html">Contactar</a></li>
<li> <a href
<li> <a href
</ul>
</div>
<div id="buscador">
<!-- SiteSearch Google -->
<form method="get" action="http://www.microsiervos.com/search.php" target="_top">
<input type="hidden" name="domains" value="Microsiervos.com"></input><input style
<input type="hidden" name="sitesearch" value=""></input>
<input type="hidden" name="sitesearch" value="Microsiervos.com" checked="checked">
<input type="hidden" name="client" value="pub-3494090758001309"></input>
<input type="hidden" name="forid" value="1"></input>
<input type="hidden" name="channel" value="7867421060"></input>
<input type="hidden" name="ie" value="UTF-8"></input>
<input type="hidden" name="oe" value="UTF-8"></input>
<input type="hidden" name="cof" value="GALT:#008000;GL:1;DIV:#336699;VLC:663399;AH
<input type="hidden" name="hl" value="es"></input>
</form>
```

Figura 1: Fragmento de código de una página web (define estructura y contenido)

Para navegar por Internet necesitamos de una aplicación que sea capaz de interpretar la estructura y contenidos de las páginas y mostrarlos de forma cómoda y legible al usuario. Estas aplicaciones se conocen como **navegadores web** (ver **Figura 2**)

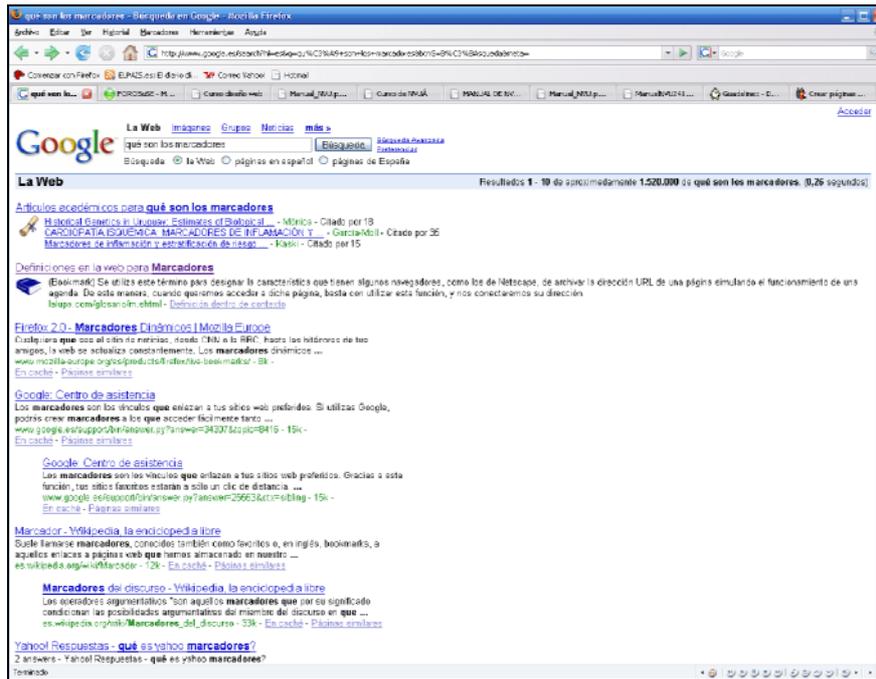


Figura: Navegador Mozilla Firefox

Además, muchas de las páginas publicadas en Internet se encuentran interconectadas unas con otras por medio de **enlaces** o **hipervínculos**, (ver **Figura 3**) que permiten movernos de unas a otras con gran facilidad. Este proceso de desplazarnos entre distintas páginas de la red Internet se denomina **navegar por Internet**.

[internet protocol](#)

Figura 3: Enlace web o hipervínculo

## 1.2 Buscar información

Una de las mayores ventajas que ofrece la red Internet es la gran cantidad de información que nos proporciona sobre cualquier materia de interés (noticias de actualidad, empleo, formación, ocio, etc.).

En muchos casos, la ventaja que supone el hecho de tener acceso a tal cantidad de información puede convertirse en algo, puesto que podemos llegar a perdernos entre tal cantidad de datos.

Es por esto que en muchos casos resulta difícil encontrar y seleccionar la información que satisfaga nuestras necesidades.

Para ayudar al usuario en esta tarea, se crearon los llamados **buscadores**. Estos no son sino aplicaciones web que permiten realizar búsquedas de información, ofreciendo los resultados al usuario en unos pocos segundos.

### Técnicas de búsqueda



La red Internet aloja a gran cantidad de buscadores. Cualquier buscador de entre los disponibles nos proporcionará información sobre las páginas que deseamos. Existen una serie de técnicas de búsqueda que nos ayudarán a encontrar información específica. Vamos a elegir uno para desarrollar estas técnicas de búsqueda. No obstante, éstas son muy similares en cualquiera de ellos.

Trabajaremos con [Google](#). Se trata de un potente buscador con una amplia base de datos. Tal vez sea, junto con **Altavista**, uno de los más capaces que se nos ofrecen hoy en la red. Además, tiene versión en castellano.

La página principal de este buscador es extremadamente sencilla y clara (ver **Figura 4**).

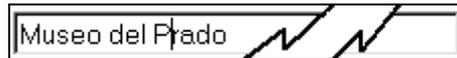


Figura 4: Página de inicio de Google

a) Búsqueda de palabras clave

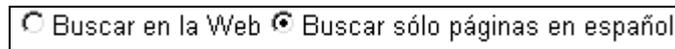
Es la que nos va a proporcionar resultados más numerosos, siempre que el objeto de la búsqueda sea concreto. Para realizar este tipo de búsqueda de forma elemental se deben seguir los siguientes pasos:

1. Introducir las palabras relacionadas con el tema deseado en el espacio en blanco colocado a tal efecto en la parte central de la ventana (**ver Figura 5**)



**Figura 5:** Campo de búsqueda

2. Elegir el ámbito de búsqueda. Google nos ofrece mostrarnos en su listado páginas en cualquier idioma (**Buscar en la web**) ó sólo páginas en castellano (**Buscar sólo páginas en español**) (**ver Figura 6**)



**Figura 6:** Selección de ámbito de búsqueda

3. Pinchar con el botón izquierdo del ratón en **Búsqueda en Google** (**ver Figura 7**)



**Figura 7:** Botón de búsqueda

El resultado de una búsqueda de este tipo será una página en la que se nos ofrece un listado de páginas web que tratan el tema reflejado en las palabras clave (**ver Figura 8**)



Figura 8: Resultados de la búsqueda

Vamos a comentar las diversas partes señaladas en rojo en la figura anterior:

- **Búsqueda Avanzada:** muestra una página que permite acotar la búsqueda si es necesario. Se suele usar en aquellos casos en que el número de resultados es excesivamente alto.
- **Preferencias:** muestra una página que permite configurar nuestras preferencias de búsqueda, incluyendo el número predeterminado de resultados por página, el idioma de la ventana, si preferimos ver las páginas encontradas en una ventana nueva, etc... Es decir, permite configurar el aspecto y funcionamiento del buscador de forma personalizada, según nuestras necesidades.
- **Barra de estadísticas:** proporciona una descripción de la búsqueda e indica el número de resultados encontrados, así como el tiempo que llevó completar su búsqueda y el número de resultados que se muestran por página. Para ver otros tantos debemos ir al hipervínculo **Siguiente**. Para cambiar el número de direcciones que se muestran por página, podemos ir al enlace **Preferencias**.
- **Título de la página:** la primera línea del resultado es el título de la página web encontrada. A veces, en vez del título aparecerá una URL, lo que significa que la página no tiene título.
- **Texto debajo del título:** este texto es un resumen de la página devuelta con los términos de búsqueda resaltados. Estos fragmentos permiten ver el

contexto en que los términos aparecen en la página, antes de hacer clic en el resultado.

- **Descripción:** si la consulta aparece en la lista del directorio web de categorías (ver el punto siguiente Búsquedas por directorio), se muestra la descripción archivada por el autor
- **Categoría:** también está relacionado con la organización de los sitios web por directorio. Si un sitio encontrado aparece en el directorio de la web, la categoría en la que aparece se muestra bajo su descripción.
- **URL del resultado:** es la dirección en la web del resultado encontrado. Haciendo un clic con el botón izquierdo del ratón sobre ella, accederemos a la página en cuestión.
- **Tamaño:** este número es el tamaño del texto de la página web encontrada. Se omite para los sitios que todavía no se han indexado.
- **En Caché:** al hacer clic en el vínculo almacenado en caché, veremos el contenido de la página web tal como aparecía en el momento en que se incorporó a la base de Google. Si por algún motivo, la página ha desaparecido, el vínculo del sitio no muestra la página actual, sino la versión almacenada en la base de Google con lo que se puede encontrar la información buscada. Los términos de la búsqueda aparecen resaltados en la versión almacenada en caché.
- **Páginas similares:** cuando selecciona el vínculo Páginas similares para un resultado en particular, Google busca automáticamente páginas en la web que estén relacionadas con este resultado.
- **Resultado jerarquizado:** cuando Google encuentra más de un resultado en el mismo sitio web, muestra primero el más relevante y las otras páginas relevantes del mismo sitio web aparecen escalonadas debajo de él.
- **Otras páginas con resultados:** nos ofrece vínculos a las páginas que nos muestran nuevos listados con resultados. En este caso, Google está configurado para que aparezcan 10 resultados en cada página.
- **Siguiente:** nos permite acceder directamente a la página siguiente.

## **b) Búsqueda por directorio**

La búsqueda por directorio se utiliza cuando el tema de la búsqueda no es demasiado concreto. El directorio nos ofrece una lista de categorías divididas en subcategorías que nos permiten ir acotando el motivo de la búsqueda.

Al final obtenemos un listado de páginas incorporadas al directorio de Google que nos ofrecen la información buscada. Pero además Google en particular nos permite hacer una búsqueda por palabras clave dentro de una categoría del directorio. Es decir, acotado parcialmente el tema de búsqueda (localizada la categoría) usamos las palabras clave para obtener las páginas que, en esa categoría, hacen referencia a lo que buscamos.

Para realizar una búsqueda por directorio, en la página principal de elegiremos la opción **Directorio** que aparece sobre el logo del buscador.



**Figura 9:** Enlace para búsquedas por directorio





Figura 11: Resultados de búsqueda por directorio

A continuación se explican las áreas marcadas en rojo en la figura anterior:

- **Buscar resultados:** nos ofrece criterios de selección de las páginas en función de cómo aparecen en ellas las palabras clave.
- **Nº de resultados:** permite especificar el número de resultados que deseamos por página.
- **Idioma:** elegimos el idioma en el que están escritas las páginas que nos devuelve como resultados
- **Formato de archivo:** permite localizar páginas que contienen archivos en formatos concretos (.doc, .pdf, etc...)
- **Fecha:** limita las páginas devueltas como resultados en función de la última fecha en que fueron actualizadas.
- **Presencia:** filtra las páginas en función de en qué parte de las mismas aparecen nuestras palabras clave (en el título, en la dirección, en los vínculos, etc...)
- **Dominios:** permite limitar las páginas obtenidas a las que poseen en su dirección un dominio concreto. Se pueden elegir varios. (por ejemplo .es, aragob.es, etc...) Si se usan varios, se separan por comas.

### I.3 Marcadores

Los marcadores (**bookmarks** en inglés) se utilizan para archivar la dirección URL de una página simulando el funcionamiento de una agenda. De esta manera, cuando queremos acceder a dicha página, basta con utilizar esta función, y nos conectaremos su dirección directamente, sin tener que introducirla de nuevo en la barra de direcciones de nuestro navegador.

Vamos a ver continuación como podemos gestionar nuestros marcadores en uno de los navegadores de mayor uso, el Mozilla Firefox

El navegador Mozilla cuenta con una barra de menús situada en la parte superior, desde la cual podemos acceder a la gestión de nuestros marcadores (**ver Figura 12**)



Figura 12: Barra de menús del navegador Mozilla Firefox

A través del menú **Marcadores** accedemos a una serie de opciones (**ver Figura 13**)

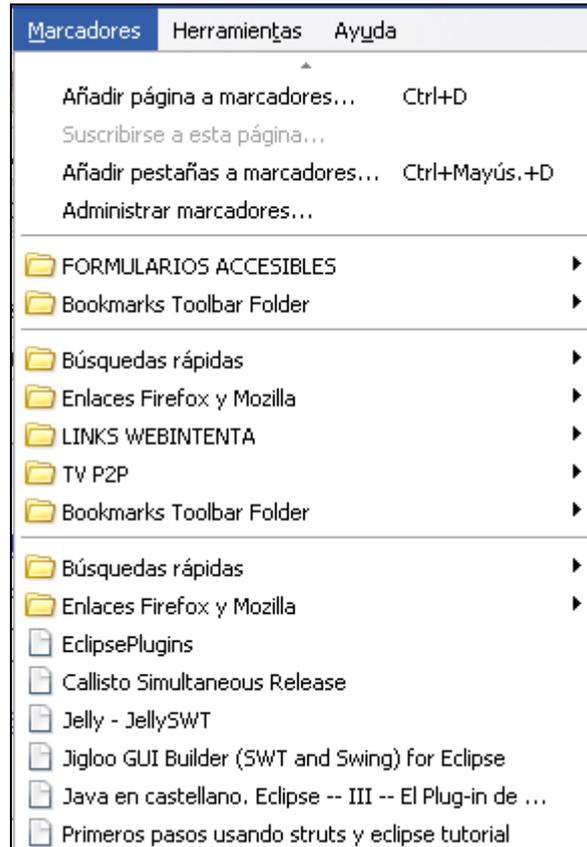


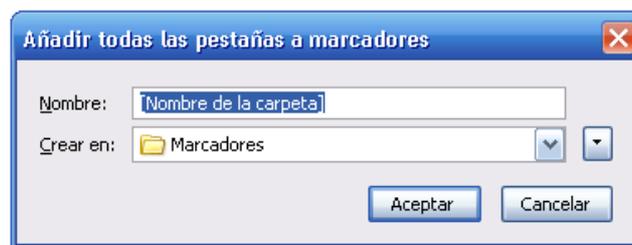
Figura 13: Opciones del menú **Marcadores**

- **Añadir página a marcadores:** nos permite agregar una nueva dirección URL a nuestra lista de marcadores, permitiendo crear carpetas dentro de la lista, para una mejor organización (**ver Figura 14**)
- **Añadir pestañas a marcadores:** Firefox es un navegador web basado en pestañas, lo que nos permite tener abiertas diversas páginas web al mismo tiempo, cada una accesible a través de una pestaña. Esta opción del menú Marcadores nos permite añadir todas las páginas que actualmente estamos visualizando en las pestañas del navegador a la lista de marcadores (**ver Figura 15**)
- **Administrar marcadores:** acceso a diversas opciones de administración de la lista de marcadores (**ver Figura 16**). Algunas de las opciones disponibles son:
  - o Nuevo marcador: permite añadir un nuevo marcador a la lista

- Nueva carpeta: permite crear una nueva carpeta para organizar los marcadores en categorías
  - Nuevo separador: permite introducir un separador en la lista de marcadores
  - Mover: permite reordenar la lista de marcadores
  - Propiedades: acceso a propiedades de los marcadores: nombre, URL, palabras clave, descripción, etc. (**ver Figura 17**)
  - Eliminar: permite borrar un marcador de la lista
- **Lista de marcadores:** lista de marcadores registrados, con posibilidad de organizarlos en carpetas



**Figura 14:** Cuadro de diálogo **Añadir marcador**



**Figura 15:** Cuadro de diálogo **Añadir todas las pestañas a marcadores**

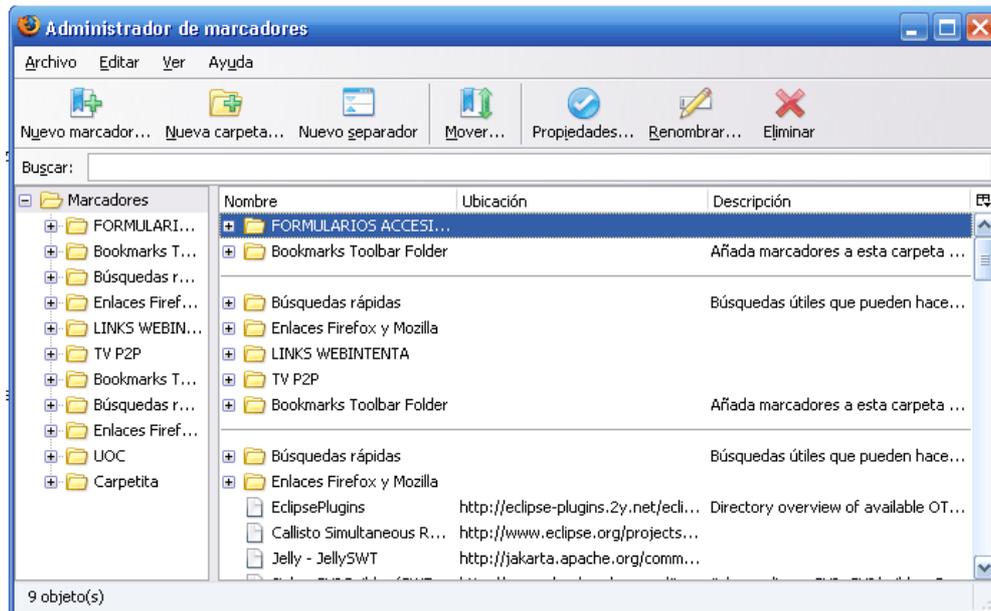


Figura 16: Cuadro de diálogo Administración de marcadores

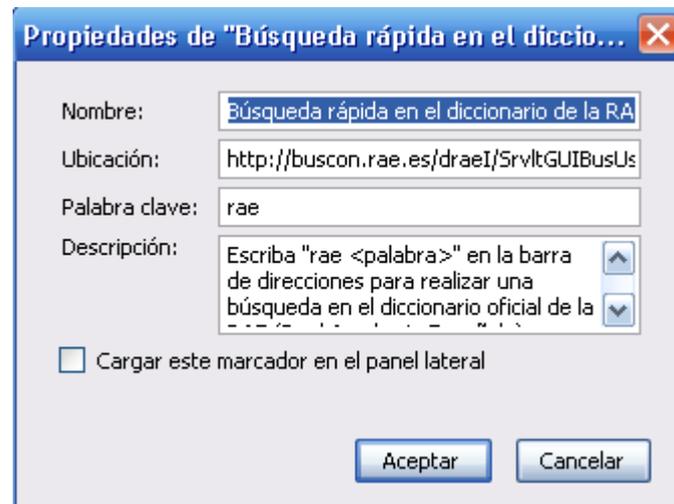


Figura 17: Cuadro de diálogo Propiedades de marcador

## 1.4 Correo electrónico

El correo electrónico (**email** en inglés), es un servicio de red que permite a los usuarios enviar y recibir mensajes. Junto con los mensajes también pueden ser enviados ficheros adjuntos (fotos, documentos, bases de datos, etc.)

Dentro de la red Internet, existen multitud de páginas que nos ofrecen servicios de correo electrónico gratuito a los usuarios. Es lo que se conoce como **correo**

**web** (al ser un servicio que se ofrece a través de una página web) A través de estos servicios el usuario puede crearse una cuenta personalizada, y enviar y recibir mensajes de correo.

A continuación vamos a ver un ejemplo a través de uno de los servicios de correo web más utilizado por los usuarios, el correo **Yahoo!**. Para acceder a dicho servicio podemos hacerlo a través del portal Yahoo! (<http://www.yahoo.es>) y una vez dentro, pulsar sobre el enlace **Mi correo** (ver Figura)

Esto nos llevará a la página de acceso al servicio de correo de **Yahoo!** (ver **Figura 18**)



**Figura 18:** Portal Yahoo España y acceso a correo Yahoo!



**YAHOO! CORREO ESPAÑA** [Yahoo!](#) [Ayuda](#)

**CORREO YAHOO! - EL FAVORITO DE TODO EL MUNDO\***

¿Ya tienes ID de Yahoo!?  
Por favor, entra para leer tu correo

Introduce tu ID y contraseña

ID de Yahoo!:

Contraseña:

Recordar mi ID en este ordenador

**Entrar**

MODO: Normal | [Seguro](#)

[¿Problemas para entrar?](#)  
[¿Has olvidado tu contraseña?](#)

**Regístrate ahora en Correo Yahoo! - Es gratis**

- Rápido y fácil de usar: te ayuda a hacer tus tareas
- Protección y seguridad: con lo último en protección antispam y antivirus
- 1 GB de almacenamiento: todo el espacio y la capacidad que necesitas para guardar tus mensajes

**Regístrate en Correo Yahoo!**

\*Comscore, Sept. 2006

Opiniones favorables sobre el Correo Yahoo! : "Yahoo! ha incorporado el diseño de Web 2.0 y ha creado un servicio de correo electrónico tipo web casi perfecto". PC plus. Haz [clic aquí](#) para conocer más detalles sobre el Correo Yahoo!

**¡Nuevo!** El Fotocorreo de Correo Yahoo! está disponible en versión Beta. [Pruébalo hoy](#)

**Correo Yahoo! para usuarios internacionales**

*Europa* : [Dinamarca](#) · [Alemania](#) · [España](#) · [Francia](#) · [Grecia](#) · [Italia](#) · [Noruega](#) · [Polonia](#) · [Suecia](#) · [Turquía](#) · [RU/Rusia](#)

*Asia Pacífico*: [Australia/NZ](#) · [China](#) · [Hong Kong](#) · [India](#) · [Indonesia](#) · [Japón](#) · [Corea](#) · [Malasia](#) · [Filipinas](#) · [Singapur](#) · [Tailandia](#) · [Taiwán](#) · [Vietnam](#)

*Américas*: [Argentina](#) · [Brasil](#) · [Canadá](#) · [México](#) · [en chino](#) · [EEUU en español](#) · [en ruso](#)

Copyright © 2005 Yahoo! Inc. All rights reserved. [Condiciones del servicio](#)  
ATENCIÓN: En este sitio se recoge información personal.  
Para saber más sobre cómo utilizamos tus datos, consulta nuestro [Centro de privacidad](#)

Figura 19: Página de acceso a correo Yahoo!

A través del enlace **Regístrate en Correo Yahoo!** Accederemos a un formulario que nos permitirá introducir los datos para crear nuestra cuenta de correo (**ver Figura**)

**YAHOO! ESPAÑA** CORREO [Yahoo! - Ayuda](#)

¿Tienes ya una ID o cuenta de correo Yahoo!? [Entrar](#).

Los campos marcados con asteriscos \* son obligatorios.

**Consigue tu ID de Yahoo!**

\* Nombre:

\* Apellido:

\* Sexo: [Seleccionar] ▼

\* ID de Yahoo:  @yahoo.es  
La ID puede contener letras a-z, números 0-9 y guiones bajos.

\* Contraseña:   
Mínimo seis caracteres (ten cuidado, pues diferenciamos entre mayúsculas y minúsculas)

\* Reescribir contraseña:

**Si olvidas tu contraseña...**

\* Pregunta de seguridad: [Elige una] ▼

\* Tu respuesta:   
Mínimo cuatro caracteres. Asegúrate de que puedas recordar tu respuesta y que nadie la pueda adivinar.

\* Cumpleaños: dd [Elige un mes] ▼ aaaa ?

\* Código postal:

\* País: España ▼

Correo electrónico alternativo:  ?

**Ofertas especiales**

Entrad en contacto conmigo de forma ocasional para informarme de ofertas, promociones, servicios y/o productos de Yahoo! y/o de terceros.

**Comprueba tu registro**

\* Escribe el código aquí:  [Más información](#) ⓘ  
Con esto ayudas a prevenir registros automáticos.

**Figura 20:** Formulario de alta en correo Yahoo!

Una vez completados todos los campos solicitados y aceptadas las condiciones del servicio, estaremos en disposición de acceder con los datos de nuestra nueva cuenta. Para ello, introduciremos nuestra identificación de usuario y contraseña en los campos habilitados para ello en la página de acceso al correo Yahoo! (**ver Figura 21**)



¿Ya tienes ID de Yahoo?  
Por favor, entra para leer tu correo

Introduce tu ID y contraseña

ID de Yahoo:

Contraseña:

Recordar mi ID en este ordenador

**Entrar**

MODO: Normal | [Seguro](#)

[¿Problemas para entrar?](#)  
[¿Has olvidado tu contraseña?](#)

Figura 21: Campos de acceso al correo Yahoo! (ID de usuario y contraseña)

En caso de olvido de nuestra contraseña, tenemos disponible un enlace que nos permitirá recordarla.

Una vez introducidos los datos de nuestra cuenta y pulsado el botón **Entrar**, accederemos a la página principal del servicio de correo Yahoo! (**ver Figura 22**)



Figura 22: Página principal al entrar en correo Yahoo!

Una vez dentro el servicio de correo nos da la bienvenida, y nos informa del nº de mensajes que tenemos disponibles en nuestra bandeja de entrada, así como del nº de correos que están aún sin leer.

Desde esta página tenemos acceso a todas las opciones del servicio de correo, organizadas por categorías:

#### **Categoría Correo:**

- **Revisar correo:** nos permite ver los últimos correos recibidos.
- **Redactar:** nos permite crear un nuevo mensaje de correo.
- **Acceso a carpetas de correo (enviados, recibidos, borrador, Papelera y personalizadas):** los mensajes de correo se ordenan en distintas carpetas, según el tipo de mensaje (enviado, recibido, etc.). La carpeta **Papelera** contiene aquellos mensajes que borramos del resto de carpetas. Además el servicio nos permite crear carpetas para una mejor organización (carpetas personalizadas).

#### **Categoría Direcciones**

Esta categoría nos da la opción de mantener nuestra la lista de direcciones de correo electrónico de nuestros contactos.

#### **Categoría Agenda**

El servicio incluye una agenda online para la anotación de eventos.

#### **Categoría Bloc de notas**

Permite crear notas textuales a modo de recordatorios, comentarios, etc.

---

## **I.5 Ejercicios**

---

### **Ejercicio 1**

Realizar la búsqueda en Internet de páginas que traten sobre los siguientes temas:

- Un portal inmobiliario
- Una periódico de noticias online
- Un portal de ofertas de empleo
- Información sobre recetas de cocina

Para realizar las búsquedas, se puede emplear cualquiera de los buscadores disponibles en la red Internet.

### **Ejercicio 2**

Este ejercicio trata sobre la gestión del espacio **Marcadores** de nuestro navegador web. Para ello, abrimos nuestros marcadores y añadimos 3 carpetas:



- a) Cursos y Formación
- b) Nuevas Tecnologías
- c) Software Libre

Realizar la búsqueda de 3 páginas web que muestren información sobre los temas de las carpetas y agregar dichas páginas a la lista de marcadores, ubicándolas en sus carpetas correspondientes.

### Ejercicio 3

Crear una cuenta de correo electrónico en algún servicio de correo gratuito disponible en la red Internet (Ya.com, Yahoo!, Gmail, ...).

## 2 - Unidad didáctica 2. Maquetación de páginas web (NVU)

---

### 2.1 - El entorno de trabajo

---

**NVU (New-view, "nueva visión")** es un programa para crear y modificar páginas Web. Tiene el aspecto de un programa simple de elaboración con todas sus funciones inherentes. Se diferencia de otros editores de páginas Web en que se trabaja en código abierto (**ver Figura 1**)



**Figura 1:** Logo de NVU

Se trata de un editor **WYSIWYG**, (lo que usted ve, es lo que usted obtiene) que da la posibilidad de crear páginas Web con una interfaz gráfica más amigable al usuario que el código HTML comúnmente usado, ofreciendo la ventaja de visualizar los modificaciones de la página antes de ser publicadas en la red Internet. No hay necesidad de saber el lenguaje HTML, algunas nociones básicas son suficientes para utilizar las diferentes aplicaciones que presenta el programa.

Algunas propiedades de esta aplicación son:

- Editar páginas WYSIWYG (lo que usted ve, es lo que usted obtiene), convirtiendo la creación Web en algo tan sencillo como transcribir una letra con su procesador de textos.
- Administrar archivos integrados vía FTP.
- Código HTML fiable, capaz de funcionar con los navegadores más populares de la actualidad.
- Intercambiar entre el modo WYSIWYG y HTML utilizando pestañas.
- Posibilidad de trabajar con múltiples páginas a la vez.
- Poderoso soporte para formularios, tablas y plantillas.
- Y muchas otras funciones que usted irá descubriendo a medida que lo utilice.

En la siguiente imagen podemos echar un vistazo al entorno de trabajo (**ver Figura 2**):

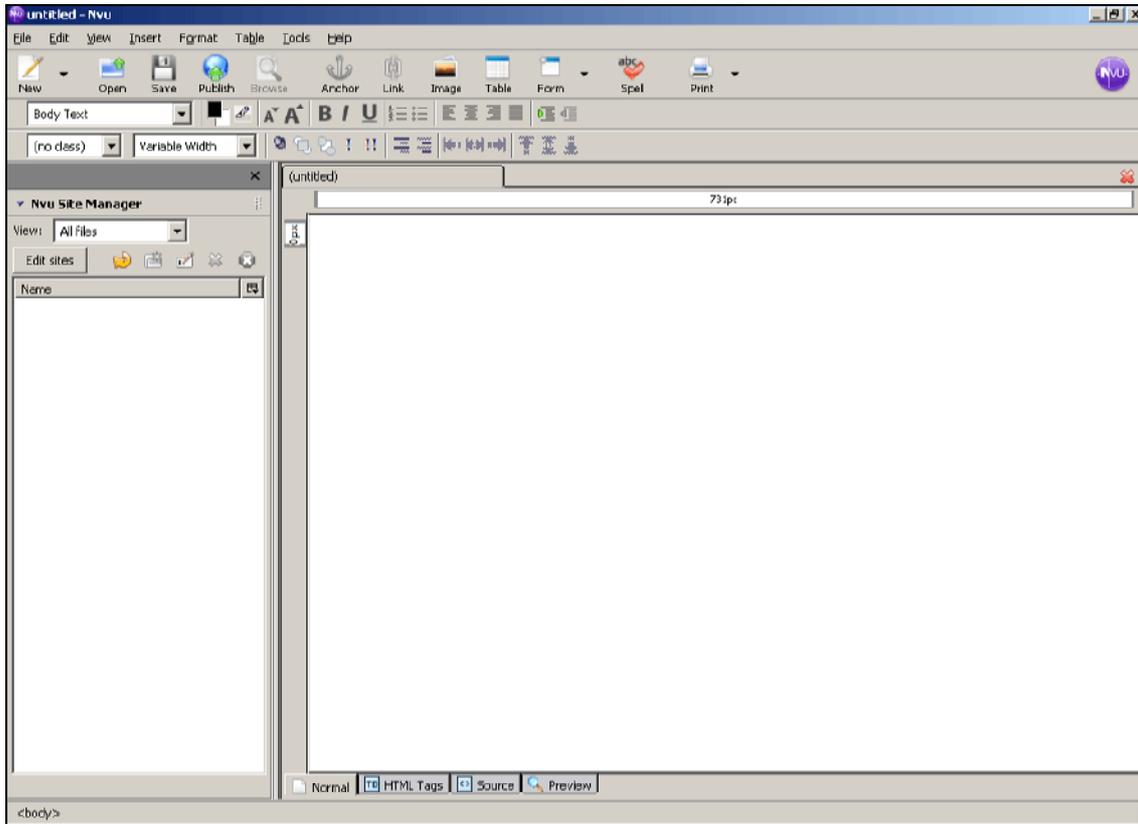


Figura 2: El entorno de trabajo de NVU

Veamos a continuación cada una de las partes que forman dicho entorno:

- **Área de trabajo:** el área de trabajo está concebida como el espacio que permite desplazarse entre las diferentes pantallas, ofreciéndonos la posibilidad de visualizar lenguaje código (HTML), observar los cambios realizados y la forma definitiva en la que se publicará la página (**ver Figura 3**)



Figura 3: Opciones de visualización del área de trabajo

Las opciones de visualización de esta área son:

- o **Normal:** escenario que permite comenzar a crear elementos propios del HTML, con estructuras prediseñadas como tablas, imágenes, texto, entre otras.
- o **HTML Tags (Etiquetas):** permite la edición de componentes al igual que en la vista normal y vista previa, sólo que se especifica sobre cuál etiqueta está el foco de trabajo. Está representada por HTML o TAGS.

- **Source (Código fuente):** permite editar y visualizar el código fuente generado
- **Preview(Vista preliminar):** previsualiza la estructura del documento
- **Menú principal:** ubicada en la parte superior de la pantalla, contiene una serie de opciones imprescindibles para realizar diversas funciones así como los íconos de acceso directo más usados (**ver Figura 4**)



**Figura 4:** Menú principal de NVU

Las opciones de dicho menú son:

- **File (Archivo):** constituye la primera opción del menú principal. En ella se contemplan opciones como crear, abrir y guardar e imprimir archivos.
- **Edit (Editar):** las opciones de edición son estándares en casi todas las herramientas. Aquí se pueden localizar las opciones para copiar, pegar, deshacer, rehacer, seleccionar, buscar y reemplazar. NVU nos brinda una opción adicional denominada "**Preferencias**". En esta opción se configuran ciertas características del documento, las cuales se aplicarán cada vez que se abra la página en construcción. Entre los atributos están: colores del texto, enlaces (visitados, activos) y campos (como la cantidad de páginas recientemente visitadas).
- **View (Ver):** En esta opción se configuran las barras de herramientas, definiendo qué elementos se desean visualizar en el área de trabajo. También constituye otra vía para desplazarse entre los diversos modos de edición (normal, etiquetas, fuente y vista preliminar), otra opción es la de cambiar el Zoom del escenario, el cual viene graduado por defecto en 100%.
- **Insert (Insertar):** Permite incorporar elementos a la escena, tales como tablas, formularios, imágenes, anclas y enlaces externos. Uno de los puntos primordiales y de gran interés es la inserción de caracteres especiales, los cuales son necesarios dentro de cualquier contenido que requiera un código especial para poder ser visualizado. Los objetos inteligentes de los cuales dispone NVU y que pueden ser insertados, no se visualizan en el área de trabajo normal, etiqueta o vista preliminar, sólo hasta que se vea en un explorador se puede tener la seguridad de qué objeto funciona correctamente.
- **Format (Formato):** Permite cambiar la apariencia de los elementos del documento como: tipografía, tamaño, color, estilo, formato del párrafo, inserción de listas, sangría, color del párrafo, letra y fondo.
- **Table (Tabla):** Las tablas permiten tener mayor orden en la estructura de contenido bien sea gráfico o textual. Aquí podemos



insertar, seleccionar y borrar una tabla, además de unir celdas seleccionadas, y definir el color de fondo de la misma.

- **Tools (Herramientas):** Se utiliza para validar el código generado en HTML, asignar y administrar las contraseñas por trabajo, asignar rutinas JavaScript desde una consola especial, así como para verificar la sintaxis generada. La opción de mayor uso es el Editor de hojas de estilos o CSS (Cascade Style Sheet), esto se trata de una especificación sobre los estilos físicos aplicables a un documento HTML, los CSS tratan de dar la separación definitiva de la lógica (estructura) y el físico (presentación) del documento.
  - **Help (Ayuda):** en esta opción está el enlace a la ayuda en línea desde el sitio oficial de NVU. Cabe mencionar que la mayoría de los documentos que se encuentran en la Web están en otros idiomas y la información en español es bastante escasa
- **Barra de herramientas:** situada en la parte superior, justo debajo del menú principal. Contiene los íconos que dan acceso a las opciones de creación de elementos (**ver Figura 5**)



**Figura 5:** Barra de herramientas de NVU

Las opciones disponibles a través de los botones de la barra de herramientas son:

- **New (Nuevo):** Permite crear un nuevo documento en blanco, generando una pestaña de acceso identificada con el nombre que se asignó al archivo al momento de ser guardado.
- **Open (Abrir):** Permite ubicar un archivo existente a partir de una dirección o nombre del archivo
- **Save (Guardar):** almacena el documento, proyectando una ventana donde se asignará el título que esta tendrá en el explorador.
- **Publish (Publicar):** representa la publicación del documento desarrollado para un Web site. En esta pantalla se deben especificar datos puntuales del servidor, (IP, contraseña, usuario) y la estructura raíz que se debe respetar al momento de publicar.
- **Browse (Navegar):** permite visualizar el documento en el explorador seleccionado.

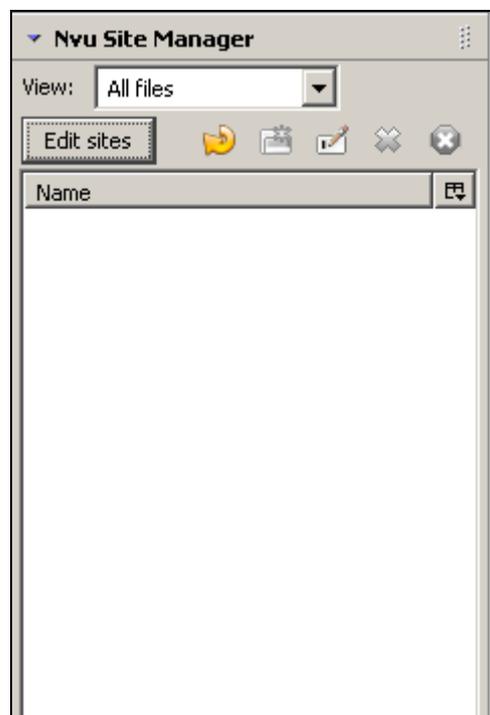
- **Anchor (Enlace interno o ancla):** permite asignarle a un determinado objeto, una dirección externa o interna que genere un llamado a un enlace o link.
- **Link (Enlace):** permite asignarle a un determinado objeto, una dirección externa o interna que genere un llamado a un enlace o link.
- **Image (Imagen):** permite insertar un archivo de imagen, tomando en cuenta las rutas donde se encuentran estas ubicadas.
- **Table (Tabla):** conjunto de celdas que permiten organizar la estructura de la información a publicar. Igualmente adecua el formato de textos y gráficos presentes en el contenido
- **Form (Formulario):** permite tener una interactividad en la Web, facilidad en planteamientos de selección múltiples, accesibilidad para la creación de cuadros de diálogo, texto, entre otros.
- **Spell (Traducción):**
- **Print (Imprimir):** permite la impresión del contenido que se visualiza en el área de trabajo normal.

---

## 2.2 Configurar un sitio remoto

---

En la parte izquierda de la interfaz de usuario de la aplicación tenemos disponible el panel **NVU Site Manager (Gestor de de sitios NVU)**, que nos va a permitir configurar un nuevo sitio web remoto (**ver Figura 6**)



**Figura 6:** Gestor de sitios remotos de NVU



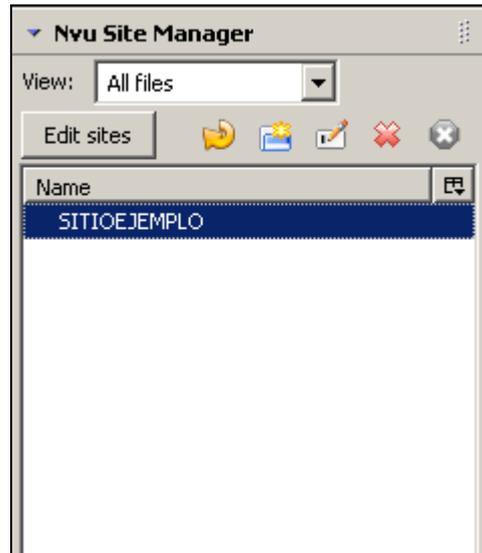
Para crear un nuevo sitio, pulsamos sobre el botón **Edit sites (Editar sitios)**, lo que nos desplegará el siguiente cuadro de diálogo (**ver Figura 7**)

**Figura 7:** Configuración de un sitio en NVU

A través de dicha ventana podemos configurar los datos de nuestro nuevo sitio web:

- **Site name (Nombre del sitio):** nombre que vamos a asignar al sitio
- **Web site information (Información del sitio web):** se corresponde con la dirección HTTP de la página inicial del sitio web.
- **Publishing server (Servidor de publicación):** ruta (carpeta) del servidor en el que se van a publicar (alojar) las páginas del sitio web. Se trata de una dirección FTP.
- **Username, password (Usuario y contraseña):** nombre de usuario y contraseña de FTP para acceder al servidor de publicación. La opción **Save password (Guardar contraseña)** permite almacenar la contraseña, de forma que no tengamos que volverla a introducir a la hora de conectar con el servidor vía FTP.

Una vez establecidos todos los parámetros, nuestro nuevo sitio se añadirá a la lista **Publishing sites (Sitios de publicación)**, así como una referencia desde el Gestor de Sitios Remotos de NVU (**ver Figura 8**)



**Figura 8:** Nuevo sitio creado en el Gestor de Sitios Remotos

Tal y como se aprecia en la figura, al seleccionar uno de los sitios disponibles en el gestor, se activan una serie de botones situados en la parte superior, que pasamos a comentar a continuación (**ver Figura 9**)



**Figura 9:** Opciones disponibles en el Gestor de Sitios Remotos

-  **Refresh (Actualizar):** permite reconectar con el sitio remoto seleccionado en la lista
-  **New folder (Nueva carpeta):** permite crear una nueva carpeta en el sitio remoto
-  **Rename file o folder (Renombrar fichero o carpeta):** permite cambiar el nombre a un fichero o carpeta del sitio remoto
-  **Delete (Eliminar):** permite eliminar un fichero o carpeta del sitio remoto
-  **Stop (Detener):** detiene la conexión al sitio remoto

Una vez creado el sitio, podemos desplegar doble clic sobre el nombre del sitio o bien pulsar sobre el botón  para conectarnos al sitio y desplegar los contenidos (ficheros y carpetas) del sitio remoto (**ver Figura 10**)

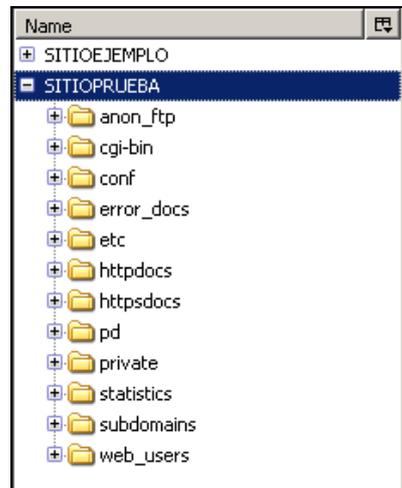


Figura 10: Contenidos de sitio remoto

## 2.3 Configurar las propiedades de la página

A la hora de trabajar con el editor de NVU para la edición de las páginas web de nuestro sitio, podemos configurar ciertas propiedades. Éstas son accesibles a través del menú **Format -> Page Title & Properties (Formato -> Título de página y Propiedades)** (ver Figura 11)

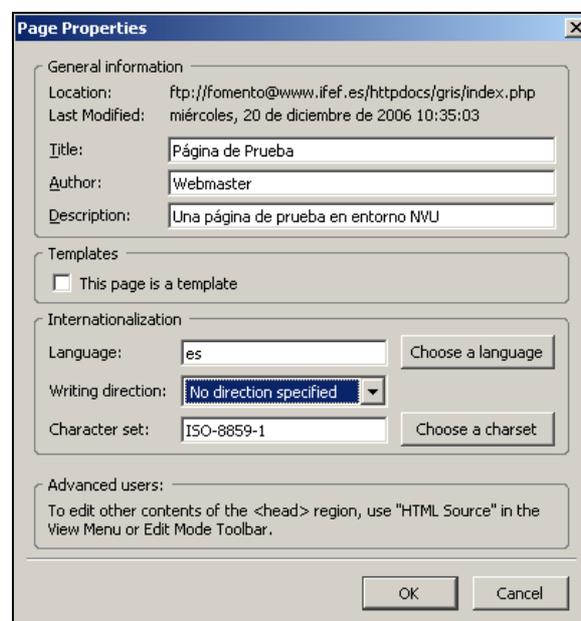
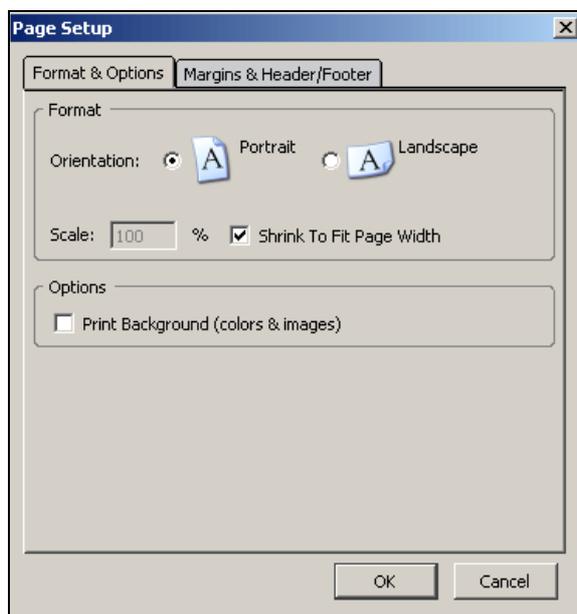


Figura 11: Propiedades de página

En esta ventana podemos configurar opciones tales como:

- Título de la página
- Autor
- Descripción
- Establecer la página como una plantilla, reutilizable para otras páginas
- Lenguaje para la página (español, inglés, ...)
- Dirección de escritura (sin indicar, de izquierda a derecha, de derecha a izquierda)
- Conjunto de caracteres para la página

Por otro lado, si queremos obtener por impresora los contenidos de una web, podemos hacerlo mediante el menú **File -> Page Setup (Archivo -> Configuración de página)** (ver Figura 12) podemos configurar las opciones a la hora de imprimir los contenidos de una web de nuestro sitio (ver Figura 12)



**Figura 12:** Configuración de página

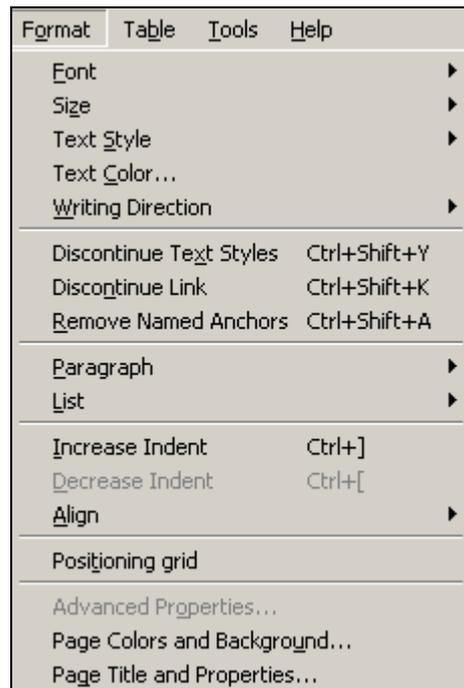
Este cuadro de diálogo tiene distribuidas sus opciones en 2 pestañas:

- **Format & Options (Formato y Opciones):** dentro de esta pestaña se encuentran las opciones que permiten establecer la orientación de la página, impresión de colores de fondo de página e imágenes, etc.
- **Margins & Header/Footer (Márgenes y Encabezado/Pie de página):** incluye las opciones para personalizar los valores de los márgenes, la cabecera y el pie de página.



## 2.4 Textos

NVU dispone de una gran variedad de opciones de configuración a la hora de introducir elementos textuales en nuestras páginas web (Ej. párrafos). Todas estas opciones están accesibles a través del menú **Format (Formato)** (ver **Figura 13**)



**Figura 13:** Opciones de texto

Vamos a ir viendo una a una cada una de estas opciones:

- **Font (Fuente):** esta opción da acceso a la lista de fuentes disponibles en la aplicación, así como a la posibilidad de establecer un ancho fijo o variable al tipo de letra elegido.
- **Size (Tamaño):** determina el tamaño de visualización para el tipo de letra, desde el más pequeño (**x-small**) hasta el más grande (**xx-large**)
- **Text Style (Estilo de texto):** ofrece una gran variedad de estilos para aplicar al texto (cursiva, negrita, subrayado, etc.) (ver **Figura 14**)

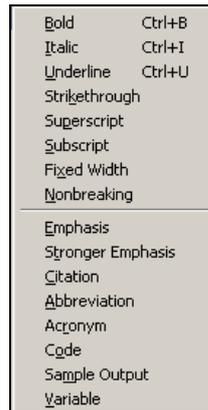


Figura 14: Opciones de estilo de texto

- **Text color (Color del texto):** permite elegir un color para el texto (ver Figura 15)

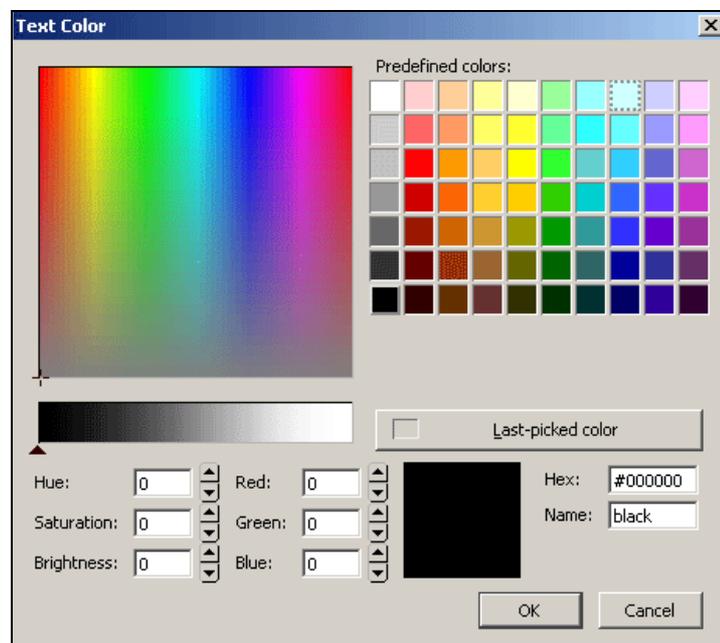


Figura 15: Elección de color del texto

- **Writing direction (Dirección de escritura):** permite establecer la dirección de escritura del texto (sin especificar, de izquierda a derecha o derecha a izquierda)
- **Discontinue text styles (Estilos de texto discontinuos):** permite establecer estilos de texto de forma discontinua
- **Discontinue link (Enlace discontinue):** de forma análoga a la opción anterior, pero para los enlaces
- **Remove Named Anchors (Eliminar Anclas Nombradas):** mediante esta opción podemos eliminar los enlaces internos (anclas) establecidos en las páginas.
- **Paragraph (Párrafo):** permite establecer distintos estilos para los párrafos (ver Figura 16)

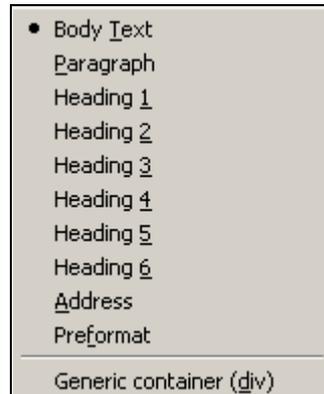


Figura 16: Estilos de párrafo

- **Lists (Listas):** desde esta opción podemos elegir el tipo de lista a emplear (sin enumerar, ordenadas, de definición de términos, etc) a la hora de establecer un listado o enumeración de elementos (**ver Figura 17**)

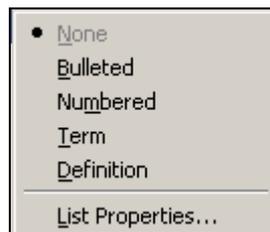


Figura 17: Tipos de listas

- **Increase indent (Incrementar indentación):** permite aumentar el valor de indentación entre elementos, a la hora de escribir el código de nuestra página web.
- **Decrease indent (Decrementar indentación):** lo contrario.
- **Align (Alineamiento):** permite establecer el alineamiento del texto (izquierda, derecha, centro, justificado)
- **Position grid (Cuadrícula de posicionamiento):** permite mostrar una cuadrícula que facilita el posicionamiento de elementos (imágenes, capas, ...) dentro de la página web.

---

## 2.5 Vínculos

---

Los **enlaces** o **vínculos** permiten enlazar una página con otra, dentro de un sitio web, o bien enlazar páginas de sitios web diferentes. Una tercera opción, conocida como **anclas o enlaces internos**, permite enlazar con distintas partes de una misma página.

Para insertar un enlace tenemos la opción **Insert -> Link (Insertar -> Enlace)** (**ver Figura 18**)

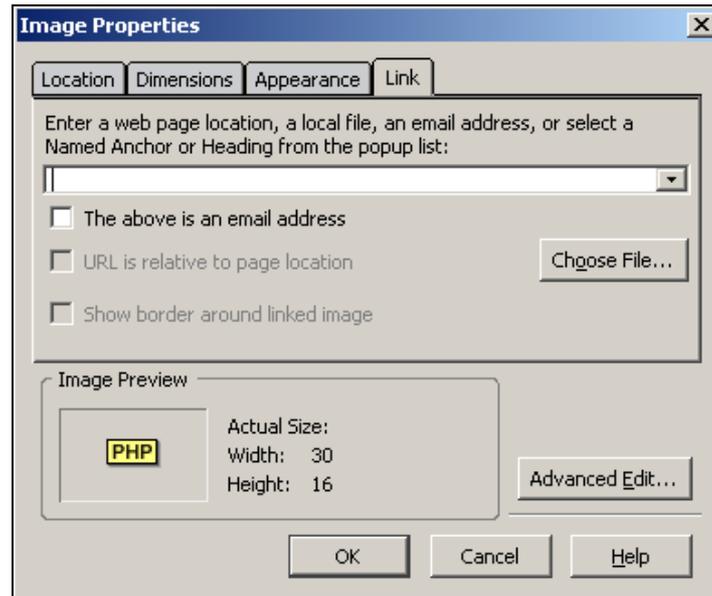


Figura 18: Inserción de enlaces

Desde la pestaña **Link (Enlace)** podemos definir el nuevo enlace. Este puede ser a una página web, a un fichero local, una dirección de correo electrónico, un enlace interno o ancla, etc.

En el caso de definir un **enlace interno o ancla**, tenemos disponible en la barra de herramientas un botón específico para ello, que nos permitirá asignarle un nombre a dicho enlace (**ver Figura 19**). Esta opción también es accesible a través de la opción **Insert -> Named Anchor (Insertar -> Ancla con nombre)**.



Figura 19: Botón Anchor (Ancla) de la barra de herramientas y cuadro de diálogo para establecer nombre de ancla

## 2.6 Imágenes

Otro elemento que podemos insertar en una página web es una imagen. Para acceder a las opciones correspondientes a este elemento tenemos disponible la opción **Insert -> Image (Insertar -> Imagen)** o bien el botón correspondiente en la barra de herramientas. (**ver Figura 20**)

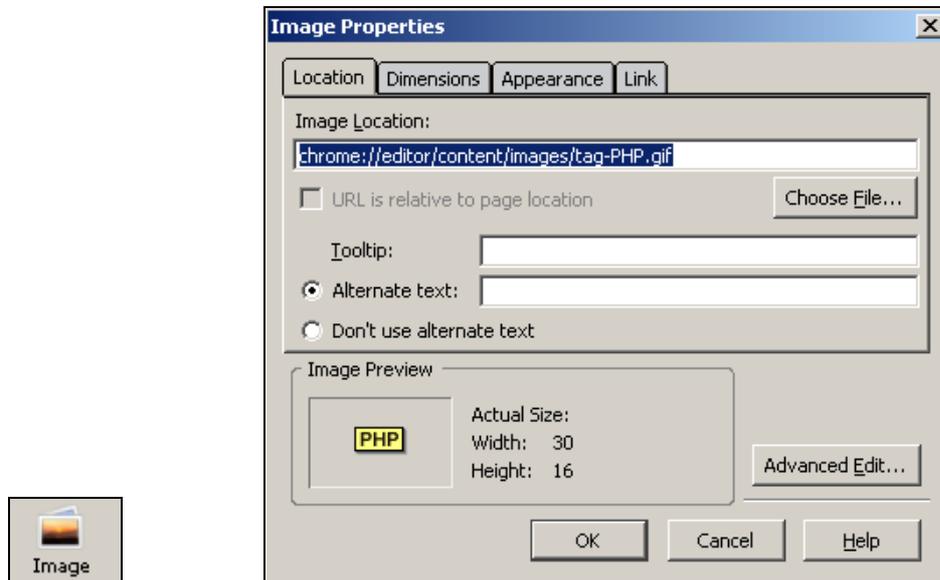


Figura 20: Propiedades de configuración de una imagen

En el cuadro de diálogo que nos aparece, disponemos de una serie de pestañas para configurar las propiedades de la imagen:

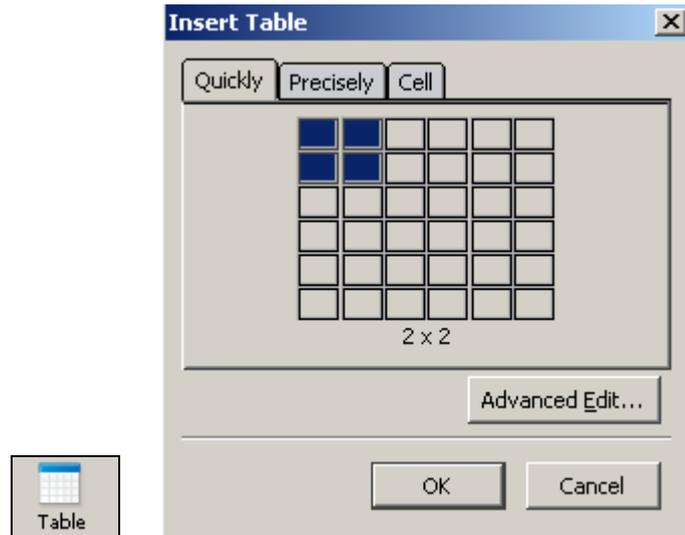
- **Location (Localización):** permite establecer la ubicación de la imagen dentro del sitio web, el mensaje a mostrar al situar el ratón sobre la imagen (**tooltip**), el texto alternativo por si la imagen no puede mostrarse en la página (**alternate text**), así como una vista previa de la imagen.
- **Dimensions (Dimensiones):** muestra las dimensiones actuales de la imagen, pudiendo modificarlas
- **Appearance (Apariencia):** permite establecer el espaciado del texto con respecto a la imagen

---

## 2.7 Tablas

---

NVU nos proporciona la posibilidad de crear e introducir tablas en nuestras páginas web de forma sencilla. Para ello disponemos de la opción de menú **Insert -> Table** o bien mediante el botón correspondiente en la barra de herramientas (**ver Figura 21**)



**Figura 21:** Botón de acceso y cuadro de diálogo para la creación de tablas

En el cuadro de diálogo que aparece tenemos 3 pestañas disponibles para establecer el nº de filas y columnas de nuestra tabla:

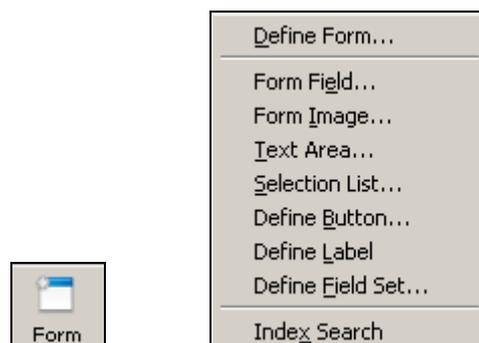
- **Quickly (Rápidamente):** nos ofrece una cuadrícula que nos permite seleccionar el nº de filas y columnas de forma rápida, pasando el ratón sobre las celdas que queremos que formen parte de nuestra tabla.
- **Precisely (De forma precisa):** permite establecer de forma numérica el nº de filas y columnas, así como el ancho de tabla y la opción de borde.
- **Cell (Celda):** son opciones de configuración sobre las celdas de la tabla (espacio entre celdas, espacio entre contenido y borde de celda, etc.)

El botón **Advanced Edit (Edición Avanzada):** posee opciones adicionales de configuración de la tabla (Ej. alineamiento de celdas)

## 2.8 Formularios

Un **formulario web** nos permite introducir información en una página web y enviarla para su procesamiento a otra página, encargada de gestionar los datos enviados por el formulario.

NVU nos permite insertar un formulario a través de la opción **Insert -> Form** o bien a través del botón correspondiente en la barra de herramientas (**ver Figura 22**).



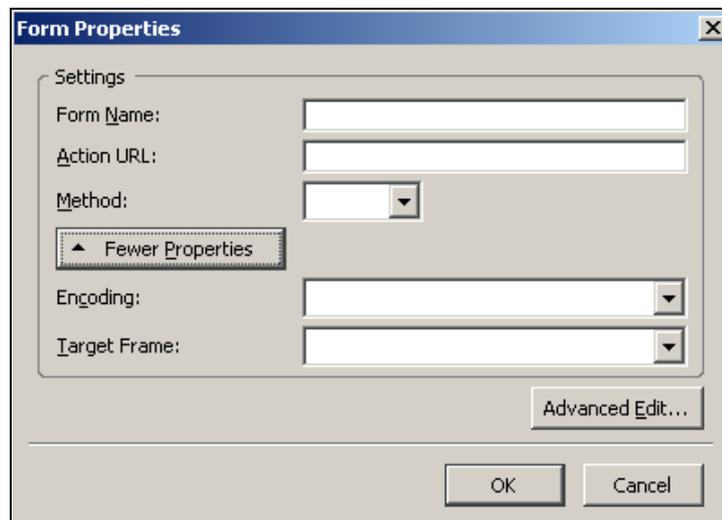


**Figura 22:** Objetos de formulario

La primera opción del menú de la figura, **Define form (Definir formulario)** nos permite crear el formulario. El resto de opciones son objetos que podemos insertar en nuestro formulario (campo de formulario, imagen, area de texto, lista de selección, botón, etiqueta, etc.)

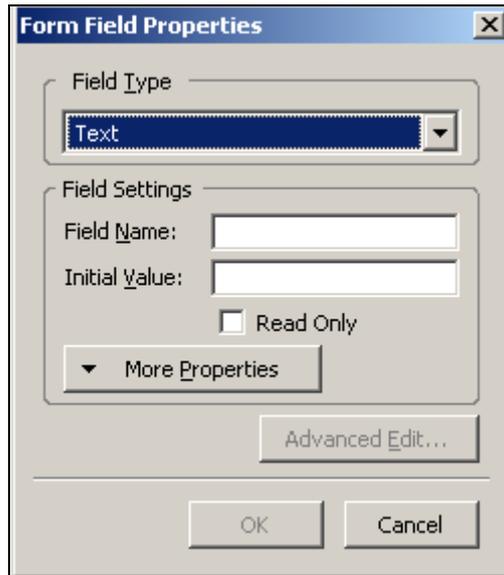
Vamos a ir viendo uno a uno dichos elementos:

- **Define form:** permite definir las propiedades del formulario (nombre, URL de la página a la que se enviarán los datos del formulario para su procesamiento, método de envío - GET o POST -, codificación y marco objetivo (si es que los datos se van a procesar en una página situada en un marco). El botón **Advanced Edit (Edición Avanzada)** permite algunas opciones adicionales de edición. (ver Figura 23)



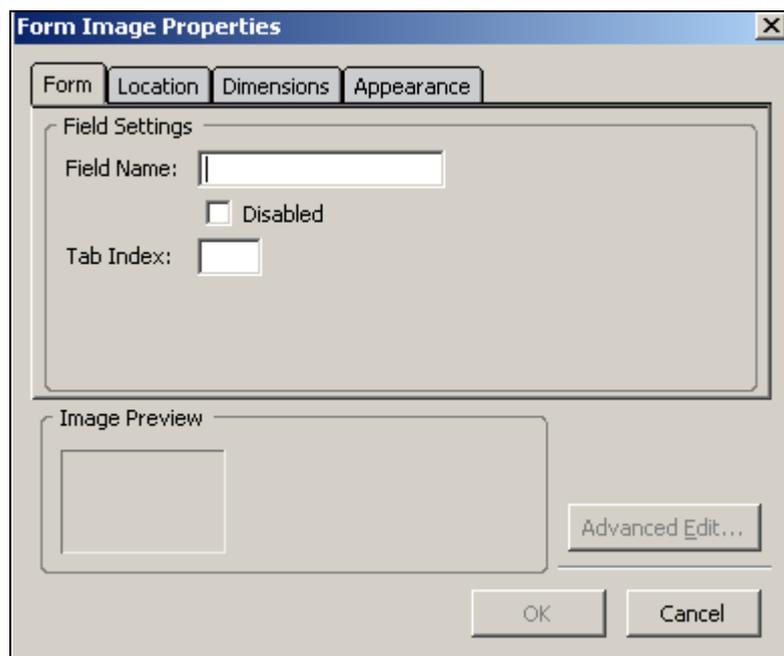
**Figura 23:** Propiedades del formulario

- **Form Field (Campo de formulario):** permite establecer las propiedades de un campo de formulario, como el tipo de campo (texto, contraseña, cuadro de chequeo, etc.), nombre del campo, valor inicial, opción de sólo lectura, etc.) (ver Figura 24)



**Figura 24:** Propiedades de campo de formulario

- **Form Image (Imagen de formulario):** permite introducir un campo para introducir una imagen a través del formulario (**ver Figura 25**)



**Figura 25:** Propiedades de imagen de formulario

Las propiedades a configurar son las mismas que las vistas para el objeto imagen, con la novedad de la pestaña **Form**, que permite establecer el nombre para el campo imagen, la opción de deshabilitarlo o no si el índice de tabulación para el campo.

- **Text Area (Área de texto):** permite introducir un área de texto, para la inclusión de grandes cantidades de texto de varias líneas (**ver Figura 26**)

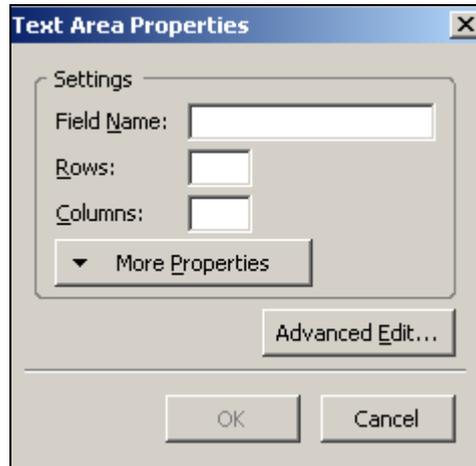


Figura 25: Propiedades de área de texto

Algunas de las propiedades que podemos configurar son el nombre del campo, el nº de filas y columnas del área, opción de sólo lectura, deshabilitación del área, tecla rápida de acceso, etc.

- **Selection list (Lista de selección):** este objeto permite escoger un objeto de una lista de opciones posibles (ver Figura 26)

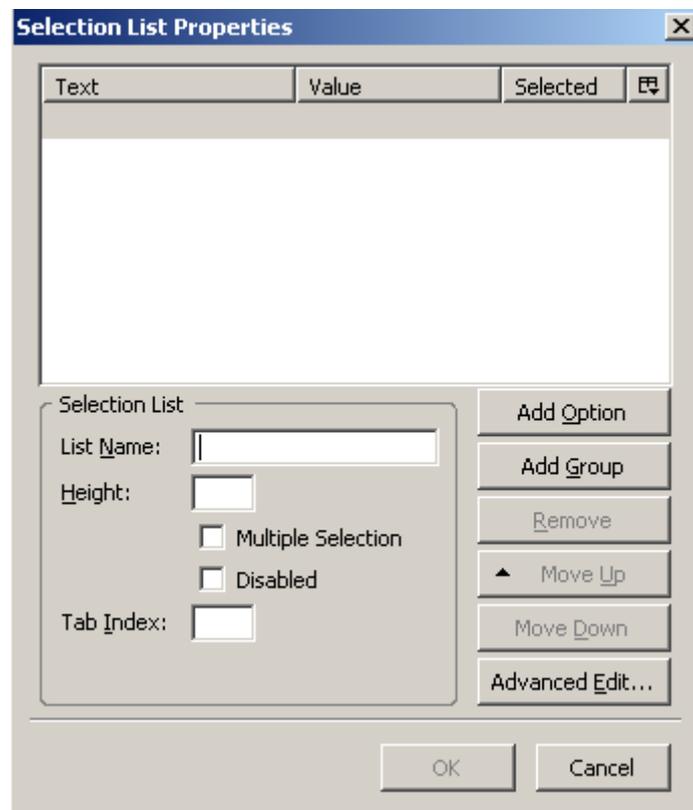
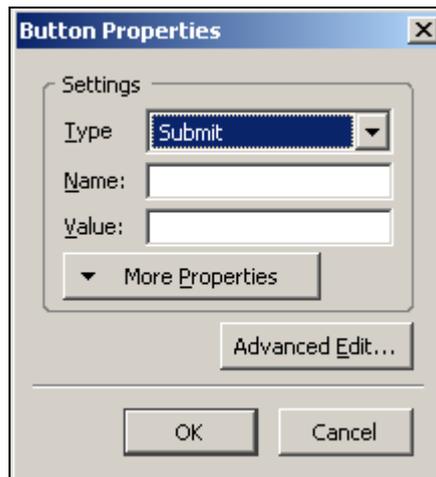


Figura 26: Propiedades de área de texto

Las opciones a configurar para este elemento incluyen el nombre para la lista, altura, adición de elementos a la lista, opción de seleccionar varios elementos de la lista a la vez, deshabilitación de la lista, creación de grupos dentro de la lista, etc.

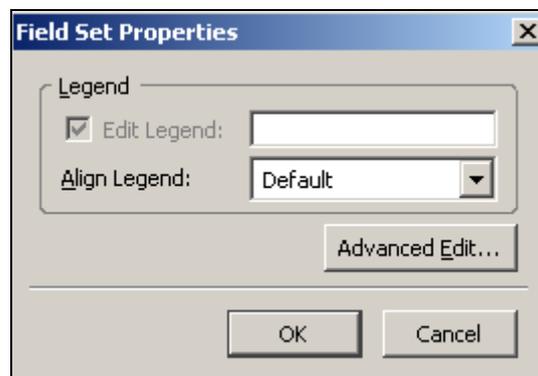
- **Define button:** permite definir un botón de acción para el formulario (ver **Figura 27**)



**Figura 27:** Propiedades de botón de acción

Entre las propiedades del botón podemos establecer el nombre, valor (texto a mostrar en el botón), tipo de botón (**Submit** – para enviar los datos del formulario, **Reset** – limpia los campos del formulario o **Button** – establece un botón de tipo genérico).

- **Define label (Definir etiqueta):** permite definir un objeto etiqueta para nombrar o describir de forma textual algún objeto de formulario
- **Define field set (Definir conjunto de campos):** permite definir un conjunto de campos, con una leyenda asociada.



**Figura 28:** Leyenda

Entre las opciones disponibles, podemos establecer el alineamiento para la leyenda del grupo de campos.



---

## 2.9 Frames

---

Los **marcos (frames en inglés)** constituyen una opción que ofrece el lenguaje HTML de dividir una página web en varias zonas, cada una de las cuales puede tener un contenido independiente de las demás de forma que cada zona es asimismo un frame.

A continuación se muestra la estructura básica de una página web dividida en frames:

```
<HTML>
<HEAD>
<title>Título página</title>
</HEAD>

<frameset cols=%columna_1,%columna_2,...,%columna_n>

<NOFRAMES>
  <!-- Código a mostrar en el caso de que el navegador web no
  soporte el uso de frames -->
</NOFRAMES>

<frame src="página_destino_1">
<frame src="página_destino_2">
<frame src="página_destino_3">
</frameset>

</HTML>
```

Donde:

- **<frameset></frameset>**: define el conjunto de marcos o frames que compondrán la página. Los argumentos **cols=%columna\_1,...,%columna\_n** definen el nº de columnas o divisiones horizontales de la página, así como su anchura en tanto por ciento.
- **<NOFRAMES></NOFRAMES>**: entre estas etiquetas se sitúa el contenido a mostrar en la página para aquellos casos en los que el navegador no soporte el uso de frames.
- **<frame src="pagina\_destino\_x">**: definición de los marcos incluidos dentro de conjunto de marcos, junto con la página destino a mostrar en dicho marco.

Comentar que aunque NVU no dispone de ninguna opción o asistente específico para crear páginas con marcos, tenemos a nuestra disposición una gran cantidad de herramientas de software libre bajo Linux para ello, como por ejemplo **Quanta +**.

---

## 2.10 Ejercicios

---

### Ejercicio 1

A través de NVU, crear una página web acerca del software libre como la que se muestra en la imagen:

#### El Software Libre

Software libre es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.



La imagen se puede obtener de la dirección: <http://es.openoffice.org/>

### Ejercicio 2

A través de NVU, crear una página web acerca de la propia herramienta como la que muestra la siguiente imagen:

#### NVU

 NVU es un editor de páginas web *multiplataforma* basado en **Mozilla Composer**, pero de ejecución independiente, y añade características nuevas como soporte integrado de CSS y mejor gestión del protocolo FTP para actualización de los ficheros. La página principal del proyecto es <http://www.nvu.com/>.

Posee las siguientes características

- Es open source.
- Disponible para Windows y Linux.
- No ocupa mucho espacio en disco.
- De fácil manejo.

### Ejercicio 3

A través de NVU, crear una página web sobre las distribuciones Linux, como la que muestra la siguiente imagen:

---



## DISTRIBUCIONES LINUX

A continuación se muestra una lista de algunas de las distribuciones Linux:

DISTRIBUCIÓN	LOGO
Red Hat	
Fedora	
Debian	

Las imágenes empleadas se pueden obtener a través de la siguiente dirección web: <http://www.linux-es.org/distribuciones>

### Ejercicio 4

A través de NVU, crear una página web que incluya un formulario como el que se muestra en la imagen:

## FORMULARIO DE CONTACTO

Para cualquier consulta o sugerencia, rellenar el siguiente formulario de contacto:

Nombre:

Apellidos:

E-mail:

Comentarios:

---

## 3 - Unidad didáctica 3. Programación en PHP y MySQL

---

### 3.1 - Sintaxis de PHP

---

**PHP** ("PHP Hypertext Pre-processor") es un lenguaje de programación interpretado de lado del servidor usado generalmente para la creación de contenido para sitios web dinámicos. Un sitio web dinámico es aquel cuyos contenidos cambian periódicamente. Dichos contenidos suelen recuperarse de una o más bases de datos.

El código PHP se escribe dentro de la propia página web, de forma conjunta con el código HTML.

De forma análoga a otros lenguajes de programación incrustados en el HTML, necesitamos especificar cuáles son las partes constitutivas del código escritas en este lenguaje. Esto se hace delimitando nuestro código PHP dentro de una serie de etiquetas especiales.

En el caso de PHP, hay varias opciones:

```
<? ..... ?>
```

```
<?php ... ?>
```

```
<% ..... %>
```

```
<script language="php" ... </script>
```

El modo de funcionamiento de una página PHP es el siguiente, a grandes rasgos, no difiere del clásico para una página dinámica de lado servidor: El servidor va a reconocer la extensión correspondiente a la página PHP (phtml, php, php4,...) y antes de enviarla al navegador va a encargarse de interpretar y ejecutar todo aquello que se encuentre entre las etiquetas correspondientes al lenguaje PHP. El resto, lo enviara sin más ya que, asumirá que se trata de código HTML absolutamente comprensible por el navegador.

Otra característica general de los scripts en PHP es la forma de separar las distintas instrucciones. Para hacerlo, hay que acabar cada instrucción con un punto y coma ";". Para la última expresión, la que va antes del cierre de etiqueta, este formalismo no es necesario.

#### Comentarios

Un comentario es una frase o palabra que nosotros incluimos en el código para comprenderlo más fácilmente al volverlo a leer un tiempo después y que, por supuesto, el ordenador tiene que ignorar ya que no va dirigido a él sino a nosotros mismos. Los comentarios tienen una gran utilidad ya que es muy fácil olvidarse del funcionamiento de un script programado un tiempo atrás y resulta



muy útil si queremos hacer rápidamente comprensible nuestro código a otra persona.

PHP cuenta con 2 modos de introducir comentarios: comentarios de una (// o bien #) o de varias líneas (/ \* ..... \*/)

```
// 2 barras verticales al texto que constituye el comentario. Suele usarse para comentarios de 1 sólo línea
```

```
# También podemos poner un comentario de una línea de esta forma
```

```
/* Esta es la otra forma de incluir comentarios, pensada para aquellos que ocupan varias líneas */
```

### Variables

Las variables en PHP se definen anteponiendo un símbolo de dólar (\$).

#### Imprimir en pantalla

La instrucción **echo** sirve para sacar en pantalla lo que hay escrito a continuación.

---

## 3.2 Variables y operadores

---

Una **variable** consiste en un elemento al cual le damos un nombre y le atribuimos un determinado tipo de información. Las variables pueden ser consideradas como la base de la programación.

De este modo podríamos escribir en un lenguaje ficticio los siguientes ejemplos de variables:

```
a="perro"  
b="muerde"
```

La variable que nosotros llamamos "**a**" posee un elemento de información de tipo texto que es "**perro**". Asimismo, la variable "**b**" contiene el valor "**muerde**".

Podríamos definir una tercera variable que fuese la suma de estas dos:

```
c=a+b
```

Si introduyésemos una petición de impresión de esta variable en nuestro lenguaje ficticio:

#### imprimir(c)

El resultado podría ser:

## perro muerde

Podríamos de la misma forma trabajar con variables que contuviesen números y construir nuestro programa:

```
a=3
b=4
c=a+b
imprimir(c)
```

El resultado de nuestro programa sería: **7**

Tal y como se comentó en el apartado anterior, las variables en PHP se crean anteniponiendo al nombre de la variable el carácter \$ (Ej. **\$contador**).

De forma genérica, el tipo de datos de una variable determina la clase o tipo de información que vamos a poder almacenar en ella (números, caracteres sueltos, cadenas de caracteres, valores de fecha y hora, etc.)

En PHP, dependiendo de la información que contenga, una variable puede ser considerada de un tipo u otro.

**Ej:**

### **Variables numéricas**

Almacenan cifras

Enteros	<code>\$entero=2002;</code>	Numeros sin decimales
Real	<code>\$real=3.14159;</code>	Numeros con o sin decimal

### **Variables alfanuméricas**

Almacenan textos compuestos de números y/o cifras

Cadenas Almacenan variables alfanuméricas `$cadena="Hola amigo";`

### **Tablas**

Almacenan series de informaciones numéricas y/o alfanuméricas

Arrays	Son las variables que guardan las tablas	<code>\$sentido[1]="ver";</code> <code>\$sentido[2]="tocar";</code> <code>\$sentido[3]="oir";</code> <code>\$sentido[4]="gusto";</code> <code>\$sentido[5]="oler";</code>
--------	--	---

### **Objetos**

Se trata de conjuntos de variables y funciones asociadas. Presentan una complejidad mayor que las variables vistas hasta ahora pero su utilidad es más que interesante.



A diferencia de otros lenguajes, PHP posee una gran flexibilidad a la hora de operar con variables. En efecto, cuando definimos una variable asignándole un valor, el ordenador le atribuye un tipo. Si por ejemplo definimos una variable entre comillas, la variable será considerada de tipo cadena

**Ej.** `$variable="5"; //esto es una cadena`

Sin embargo si pedimos en nuestro script realizar una operación matemática con esta variable, no obtendremos un mensaje de error sino que la variable cadena será asimilada a numérica:

```
<?php
$cadena="5"; //esto es una cadena
$entero=3; //esto es un entero
echo $cadena+$entero;
?>
```

Este script dará como resultado "8". La variable cadena ha sido asimilada en entero (aunque su tipo sigue siendo cadena) para poder realizar la operación matemática. Del mismo modo, podemos operar entre variables tipo entero y real. No debemos preocuparnos de nada, PHP se encarga durante la ejecución de interpretar el tipo de variable necesario para el buen funcionamiento del programa.

Sin embargo, en contraste, hay que tener cuidado en no cambiar mayúsculas por minúsculas ya que, en este sentido, PHP es sensible. Conviene por lo tanto trabajar ya sea siempre en mayúsculas o siempre en minúsculas para evitar este tipo de malentendidos a veces muy difíciles de localizar.

### Variables asignadas por referencia

En PHP también podemos asignar variables por referencia. En ese caso no se les asigna un valor, sino otra variable, de tal modo que las dos variables comparten espacio en memoria para el mismo dato.

La notación para asignar por referencia es colocar un "&" antes del nombre de la variable.

**Ej.**

```
<?php
$foo = 'Bob'; // Asigna el valor 'Bob' a $foo
$bar = &$foo; // Referencia $foo vía $bar.
$bar = "Mi nombre es $bar"; // Modifica $bar...
echo $foo; // $foo también se modifica.
echo $bar;
?>
```

Esto dará como resultado la visualización dos veces de la cadena "Mi nombre es Bob". Algo como: Mi nombre es BobMi nombre es Bob

## Variables PHP

Para terminar, vamos a ver las variables que el mismo PHP nos pone a disposición:

- **argv**: un array que incluye los parámetros que pasan al script PHP
- **argc**: como arriba, pero si el script se ejecuta a través de la línea de comandos
- **PHP\_SELF**: el nombre del script (fichero PHP) actualmente en ejecución
- **HTTP\_COOKIE\_VARS**: un array de asociación que incluye las variables que han pasado al script a través de los cookies HTTP
- **HTTP\_GET\_VARS**: un array de asociación que incluye las variables que han pasado al script a través de la solicitud de GET
- **HTTP\_POST\_VARS**: como arriba, pero referido al método POST

## Variables de sistema

Dada su naturaleza de lenguaje de lado servidor, PHP es capaz de darnos acceso a toda una serie de variables que nos informan sobre nuestro servidor y sobre el cliente. La información de estas variables es atribuida por el servidor y en ningún caso nos es posible modificar sus valores directamente mediante el script. Para hacerlo es necesario influir directamente sobre la propiedad que definen.

Existen multitud de variables de este tipo, algunas sin utilidad aparente y otras realmente interesantes y con una aplicación directa para nuestro sitio web. Aquí enumeramos algunas de estas variables y la información que aportan:

Variable	Descripción
\$HTTP_USER_AGENT	Nos informa principalmente sobre el sistema operativo y tipo y versión de navegador utilizado por el internauta. Su principal utilidad radica en que, a partir de esta información, podemos redireccionar nuestros usuarios hacia páginas optimizadas para su navegador o realizar cualquier otro tipo de acción en el contexto de un navegador determinado.
\$HTTP_ACCEPT_LANGUAGE	Nos devuelve la o las abreviaciones de la lengua considerada como principal por el navegador. Esta lengua o lenguas principales pueden ser elegidas en el menú de opciones del navegador. Esta variable resulta también extremadamente útil para enviar al internauta a las páginas escritas en su lengua, si es que existen.
\$HTTP_REFERER	Nos indica la URL desde la cual el internauta ha tenido acceso a la página. Muy interesante para generar botones de "Atrás" dinámicos o para crear nuestros propios sistemas de estadísticas de visitas.
\$PHP_SELF	Nos devuelve una cadena con la URL del script que está siendo ejecutado. Muy interesante para crear botones para recargar la página.
\$HTTP_GET_VARS	Se trata de un array que almacena los nombres y contenidos de las variables enviadas al script por URL o por formularios GET
\$HTTP_POST_VARS	Se trata de un array que almacena los nombres y contenidos de las variables enviadas al script por medio de un formulario POST
\$HTTP_COOKIES_VARS	Se trata de un array que almacena los nombres y contenidos de las cookies. Veremos qué son más adelante.



<code>\$PHP_AUTH_USER</code>	Almacena la variable usuario cuando se efectúa la entrada a páginas de acceso restringido. Combinado con <code>\$PHP_AUTH_PW</code> resulta ideal para controlar el acceso a las páginas internas del sitio.
<code>\$PHP_AUTH_PW</code>	Almacena la variable password cuando se efectúa la entrada a páginas de acceso restringido. Combinado con <code>\$PHP_AUTH_USER</code> resulta ideal para controlar el acceso a las páginas internas del sitio.
<code>\$REMOTE_ADDR</code>	Muestra la dirección IP del visitante.
<code>\$DOCUMENT_ROOT</code>	Nos devuelve el path físico en el que se encuentra alojada la página en el servidor.
<code>\$PHPSESSID</code>	Guarda el identificador de sesión del usuario. Veremos más adelante en qué consisten las sesiones.

No todas estas variables están disponibles en la totalidad de servidores o en determinadas versiones de un mismo servidor. además, algunas de ellas han de ser previamente activadas o definidas por medio de algún acontecimiento. Así, por ejemplo, la variable `$HTTP_REFERER` no estará definida a menos que el internauta acceda al script a partir de un enlace desde otra página.

### Variables superglobales

A partir de la versión **4.1.0** de PHP, se dispone de un conjunto de variables de tipo array que mantienen información del sistema, llamadas **superglobales** porque se definen automáticamente en un ámbito global.

Estas variables hacen referencia a las mismas que se accedían antes por medio de los arrays del tipo `$HTTP_*_VARS`. Éstas todavía existen, aunque a partir de PHP 5.0.0 se pueden desactivar con la directiva `register_long_arrays`.

La lista de estas variables es la siguiente:

#### **`$GLOBALS`**

Contiene una referencia a cada variable disponible en el espectro de las variables del script. Las llaves de esta matriz son los nombres de las variables globales. `$GLOBALS` existe desde PHP 3.

#### **`$_SERVER`**

Variables definidas por el servidor web ó directamente relacionadas con el entorno en don el script se esta ejecutando. Análoga a la antigua matriz `$HTTP_SERVER_VARS` (la cual está todavía disponible, aunque no se use).

#### **`$_GET`**

Variables proporcionadas al script por medio de HTTP GET. Análoga a la antigua matriz `$HTTP_GET_VARS` (la cual está todavía disponible, aunque no se use).

### **`$_POST`**

Variables proporcionadas al script por medio de HTTP POST. Análoga a la antigua matriz `$HTTP_POST_VARS` (la cual está todavía disponible, aunque no se use).

### **`$_COOKIE`**

Variables proporcionadas al script por medio de HTTP cookies. Análoga a la antigua matriz `$HTTP_COOKIE_VARS` (la cual está todavía disponible, aunque no se use).

### **`$_FILES`**

Variables proporcionadas al script por medio de la subida de ficheros via HTTP . Análoga a la antigua matriz **`$HTTP_POST_FILES`** (la cual está todavía disponible, aunque no se use). Vea también [Subiendo ficheros por método POST](#) para más información.

### **`$_ENV`**

Variables proporcionadas al script por medio del entorno. Análoga a la antigua matriz `$HTTP_ENV_VARS` (la cual está todavía disponible, aunque no se use).

### **`$_REQUEST`**

Variables proporcionadas al script por medio de cualquier mecanismo de entrada del usuario y por lo tanto no se puede confiar en ellas. La presencia y el orden en que aparecen las variables en esta matriz es definido por la directiva de configuración `variables_order`. Esta matriz no tiene un análogo en versiones anteriores a PHP 4.1.0. Vea también `import_request_variables()`.

### **`$_SESSION`**

Variables registradas en la sesión del script. Análoga a la antigua matriz `$HTTP_SESSION_VARS` (la cual está todavía disponible, aunque no se use). Vea también la sección [Funciones para el manejo de sesiones](#) para más información.



## Operadores

PHP cuenta con una serie de operadores, distribuidos en diversas categorías en función del tipo de dato con el que operan. A continuación se muestran dichos operadores:

Aritméticos: suma, resta, producto, división, resto, incremento y decremento

<code>\$a + \$b</code>	<b>Suma</b>
<code>\$a - \$b</code>	<b>Resta</b>
<code>\$a * \$b</code>	<b>Multiplicación</b>
<code>\$a / \$b</code>	<b>División</b>
<code>\$a % \$b</code>	<b>Resto de la división de \$a por \$b</b>
<code>\$a++</code>	<b>Incrementa en 1 a \$a</b>
<code>\$a--</code>	<b>Resta 1 a \$a</b>

De cadenas: operador de concatenación de cadenas ( `."` )

```
$a = "Hola";  
$b = $a . "Mundo"; // Ahora $b = "HolaMundo"
```

De comparación: mayor y menor que, igual, distinto de, mayor o igual que, menor o igual que

<code>\$a &lt; \$b</code>	<code>\$a</code> <b>menor que</b> <code>\$b</code>
<code>\$a &gt; \$b</code>	<code>\$a</code> <b>mayor que</b> <code>\$b</code>
<code>\$a &lt;= \$b</code>	<code>\$a</code> <b>menor o igual que</b> <code>\$b</code>
<code>\$a &gt;= \$b</code>	<code>\$a</code> <b>mayor o igual que</b> <code>\$b</code>
<code>\$a == \$b</code>	<code>\$a</code> <b>igual que</b> <code>\$b</code>
<code>\$a != \$b</code>	<code>\$a</code> <b>distinto que</b> <code>\$b</code>

Lógicos: operadores AND (y), OR (o), XOR (o exclusivo), ! (inversión)

<code>\$a AND \$b</code>	Verdadero si ambos son verdadero
<code>\$a &amp;&amp; \$b</code>	Verdadero si ambos son verdadero
<code>\$a OR \$b</code>	Verdadero si alguno de los dos es verdadero
<code>\$a !! \$b</code>	Verdadero si alguno de los dos es verdadero
<code>\$a XOR \$b</code>	Verdadero si sólo uno de los dos es verdadero
<code>!\$a</code>	Verdadero si <code>\$a</code> es falso, y recíprocamente

De asignación: asignación, suma y asignación, resta y asignación, producto y asignación, división y asignación, concatenación y asignación

```
$a = $b   Asigna a $a el contenido de $b
$a += $b  Le suma a $b a $a
$a -= $b  Le resta a $b a $a
$a *= $b  Multiplica $a por $b y lo asigna a $a
$a /= $b  Divide $a por $b y lo asigna a $a
$a .= $b  Añade la cadena $b a la cadena $a
```

---

### 3.3 Sentencias de control. Vectores

---

Las sentencias de control permiten ejecutar bloque de códigos dependiendo de unas condiciones. Para PHP el 0 es equivalente a Falso y cualquier otro número es Verdadero.

Existen 2 tipos de sentencias de control:

#### a) CONDICIONALES

##### **If ... else**

La sentencia **If ... else** permite ejecutar un bloque de instrucciones si la condición es **Verdadera** y otro bloque de instrucciones si ésta es **Falsa**. Es importante tener en cuenta q instrucciones si ésta es **Falsa**. Es importante tener en cuenta que la condición que evaluemos ha de estar encerrada entre **paréntesis** (esto es aplicable a todas la sentencias de control).

```
if (condición)
{
    Este bloque se ejecuta si la condición es VERDADERA
}
else
{
    Este bloque se ejecuta si la condición es FALSA
}
```

Existe una forma sencilla de usar la sentencia IF cuando no tenemos que usar el ELSE y solo tenemos que ejecutar una línea de código.

```
if ($a > 4) echo "$a es mayor que 4";
```

##### **If ... elseif ... else**

La sentencia If ... elseif ... else permite ejecuta varias condiciones en cascada. Para este caso veremos un ejemplo, en el que utilizaremos los operadores lógicos.



```
<?php
    if ($nombre == "")
    {
        echo "Tú no tienes nombre";
    }
    elseif (($nombre=="eva") OR ($nombre=="Eva"))
    {
        echo "Tu nombre es EVA";
    }
    else
    {
        echo "Tu nombre es " . $nombre;
    }
?>
```

### Switch ... case ... default

Una alternativa a If ... elseif ... else, es la sentencia Switch, la cuál evalúa y compara cada expresión de la sentencia Case con la expresión que evaluamos, si llegamos al final de la lista de Case y encuentra una condición Verdadera , ejecuta el código de bloque que haya en Default. Si encontramos una condición verdadera debemos ejecutar un Break para que la sentencia Switch no siga buscando en la lista de Case. Veamos un ejemplo.

```
<?php
switch ($dia)
{
    case "Lunes":
        echo "Hoy es Lunes";
        break;
    case "Martes":
        echo "Hoy es Martes";
        break;
    case "Miercoles":
        echo "Hoy es Miercoles";
        break;
    case "Jueves":
        echo "Hoy es Jueves";
        break;
    case "Viernes":
        echo "Hoy es Viernes";
        break;
    case "Sábado":
        echo "Hoy es Sábado";
        break;
    case "Domingo":
        echo "Hoy es Domingo";
        break;
    default:
        echo "Esa cadena no corresponde a ningún día de la semana";
}
?>
```

## b) ESTRUCTURAS REPETITIVAS

### **While**

La sentencia `While` ejecuta un bloque de código mientras se cumpla una determinada condición.

```
<?php
$num = 1;
while ($num < 5) {
    echo $num;
    $num++
}
?>
```

Podemos romper un bucle `While` utilizando la sentencia **Break**.

```
<?php
$num = 1;
while ($num < 5) {
    echo $num;
    if ($num == 3){
        echo "Aquí nos salimos \n";
        break
    }
    $num++
}
?>
```

### **Do ... while**

Esta sentencia es similar a `While`, salvo que con esta sentencia primero ejecutamos el bloque de código y después se evalúa la condición, por lo que el bloque de código se ejecuta siempre al menos una vez.

```
<?php
$num = 1;
do {
    echo $num;
    if ($num == 3){
        echo "Aquí nos salimos \n";
        break
    }
    $num++
} while ($num < 5);
?>
```

### **For**

El bucle `For` no es estrictamente necesario, cualquier bucle `For` puede ser sustituido fácilmente por otro `While`. Sin embargo, el bucle `For` resulta muy útil cuando debemos ejecutar un bloque de código a condición de que una variable se encuentre entre un valor mínimo y otro máximo. El bucle `For` también se puede romper mediante la sentencia **Break**.



```
<?php
for ($num = 1; $num <=5; $num++){
    echo $num;
    if ($num == 3){
        echo "Aquí nos salimos \n";
        break
    }
}
?>
```

En la realización de un script en PHP en múltiples ocasiones existen variables que tienen información similar y se procesan de forma semejante. Para ello PHP (y otros lenguajes) poseen un elemento denominado **array**. Un array es un conjunto de variables agrupadas bajo un único nombre. Cada variable dentro de la matriz se denomina **elemento**. Dentro de la misma matriz pueden existir variables de diferentes tipos y no es necesario que sean todas del mismo tipo.

Hay que diferenciar entre los dos tipos de matrices existentes:

- **Indexada:** aquella cuyo acceso a los elementos se realiza por la posición que ocupan dentro de la estructura (se inician siempre desde la posición 0).

Ej. \$amigos[0]

- **Asociativa:** aquella en la que los elementos están formados por pares clave-valor y el acceso se realiza proporcionando una determinada clave.

Ej. \$amigos['edad']

Para crear matrices en PHP existen dos formas:

- **De forma implícita:** consistiría en indicarle el elemento (ya sea proporcionando su posición o su clave).

Ej. \$nombres[0]='Javier';

En caso de no indicarle una posición el array tomara el valor siguiente al ultimo valor introducido.

Ej. \$nombres[]='Lucas' // tomaría como valor 1 ya que lo ultimo introducido era 0.

- **Mediante array():** en el cual le pasamos los elementos como parámetros. En caso de matriz indexada toman la posición que ocupan en la creación de la matriz, mientras que los de la matriz asociativa se les asigna su valor mediante "=>".

Ej. `$amigo=array('Nombre'=>'Jose','Direccion'=>'Neopatria 21');`

Cabe destacar que PHP no solo se limita a la existencia de matrices por sí solo sino que existen matrices de matrices, o lo que es lo mismo, matrices multidimensionales.

Ej. `$amigos[2]['Pedro']`

## Recorrido de una matriz

Disponemos de diversas herramientas para poder acceder a los elementos de una matriz. En cada momento se mantiene una referencia del elemento de la matriz al que se tiene acceso, por tanto, para recorrer una matriz bastará con modificar dicha referencia. En caso de una matriz indexada el recorrido se realizara mediante un bucle y para ello debemos saber el número de elementos totales que posee la matriz. Para ello nos basamos de la función **count(variable)** donde variable representa la variable de la que se quiere obtener el número de elementos. Si **variable** es una matriz devuelve el número de elementos que tiene, devuelve 1 si solo tiene un elemento (aunque no sea matriz) y 0 si no tiene ningún valor.

Otra función que nos permite saber el número de elementos es **sizeof(matriz)**. Para acceder a los elementos de una matriz asociativa debemos usar la función **each()** que recupera el par formado por clave y valor y además avanza una posición de puntero. Su sintaxis es `each(matriz)` y los valores que devuelve la matriz asociativa son los siguientes:

Clave	Significado
<b>0</b>	Nombre de la clave
<b>1</b>	Valor asociativo de la clave
<b>key</b>	Nombre de la clave
<b>value</b>	Valor asociado a la clave

La función que realiza el constructor **list(variable1,variable2...variableN)** es asignar los valores del elemento actual de una matriz a las variables indicadas como parámetro.

## Navegación sobre matrices

Cuando se trata de matrices indexadas la navegación es sencilla ya que tan solo basta acceder al elemento que queremos mostrar, pero al tratarse de alguna matriz asociativa no se puede aplicar el mismo tratamiento. Para ello existen un conjunto de funciones prefabricadas que nos permiten realizar multitud de acciones:

Sintaxis	Acción
<b>reset(matriz);</b>	El puntero interno vuelve a la primera posición
<b>end(matriz);</b>	El puntero interno va a la ultima posición
<b>next(matriz);</b>	El puntero va al elemento siguiente



**prev(matriz);** Accede al elemento anterior  
**current(matriz);** Devuelve el contenido del elemento actual

### Inserción de elementos

Para la inserción de elementos dentro de un array existen una serie de funciones que nos permiten añadir elementos. Entre ellas destacamos:

**array\_push(matriz,variable1,variableN);**

Añade elementos al final de la función y su longitud se incrementara tantos elementos como se hayan añadido

**array\_unshift(matriz,variable1,variableN);**

Añade elementos al principio de la función desplazando a los otros tantas posiciones como elementos haya.

**array\_pad(matriz,nuevo\_tamaño,valor\_relleno);**

Aumenta el tamaño de la matriz empleando un valor proporcionado como relleno.

### Eliminación de elementos

**array\_shift(matriz);**

Elimina el primer elemento de la matriz

**array\_pop(matriz);**

Elimina el último elemento de la matriz

**array\_splice(entrada,pos\_ini,[tamaño],[sustitutos]);**

Se usa para reemplazar o borrar el contenido de una porción de matriz, para ello debemos especificar la posición desde la cual queremos iniciar el borrado o sustitución, el tamaño o número de elementos que se verán afectados y los sustitutos (en caso que deseemos sustituirlo por algún elemento).

**array\_keys(matriz,[valor buscado]);**

Se emplea cuando deseamos eliminar un elemento cuya posición desconocemos.

**array\_values(matriz);**

Devuelve una matriz indexada con todos los valores almacenados en la matriz pasada como parámetro.

### Manipulación masiva de matrices

**array\_walk(matriz,nombre\_de\_la\_funcion,lista\_parametros);**

Se emplea para realizar el mismo proceso definido en la función en todos los elementos incluidos.

## Obtención de submatrices

### **array\_slice(matriz,posicion,tamaño);**

Permite extraer una secuencia de elementos de una matriz. Los parámetros a pasarle son la matriz en la cual queremos extraer dichos elementos, la posición desde la que se inicia la extracción y el tamaño de la extracción (posiciones que abarcamos a partir de la inicial).

## Ordenación de matrices

<b>Criterio</b>	<b>Función</b>
<b>Orden ascendente(matriz indexada)</b>	sort(matriz)
<b>Orden descendente(matriz indexada)</b>	rsort(matriz)
<b>Orden ascendente por valor(matriz asociativa)</b>	asort(matriz)
<b>Orden descendente por valor(matriz asociativa)</b>	arsort(matriz)
<b>Orden ascendente por clave(matriz asociativa)</b>	ksort(matriz)
<b>Orden descendente por clave(matriz asociativa)</b>	krsort(matriz)

## Otras funciones

En este apartado se comentaran una serie de funciones (no todas porque seria imposible) que nos pueden servir en cierto momento.

**compact()** Devuelve una matriz asociativa a partir de un numero indeterminado de parámetros

**extract()** Crea variables desde matriz asociativa

**array\_unique()** Devuelve matriz sin datos repetidos ya que algunos se eliminan

**array\_reverse()** Devuelve matriz con mismos elementos pero en orden inverso

**shuffle()** Modifica el orden de elementos de forma aleatoria

**array\_count\_values()** Devuelve una matriz asociativa que contiene frecuencias de repetición de los valores de la matriz

**in\_array()** Permite comprobar si un valor esta en la matriz

**array\_merge()** Combina elementos de dos matrices en 1.



---

### 3.4 Las funciones `include()` y `require()`

---

Las construcciones **include** y **require** son de las más conocidas en PHP. Con ellas puedes reutilizar porciones de código (script, o simple código html) cuantas veces quieras, siendo uno de sus usos más sencillos y típicos el de incluir cabeceras y pies de páginas en un sistema de plantillas.

La sentencia **include()** inserta y evalúa el archivo especificado. Puedes incluir aquí no solamente un fichero en tu servidor, sino una página web remota (indicando la URL).

Su uso típico sería `<?php include ("header.php");?>`, que llama al archivo `header.php` y lo inserta en el propio punto del script donde hacemos la llamada.

Tanto si insertamos un archivo con **include()** o **require()**, debes tener en cuenta que PHP pasa a **modo html** hasta el final del mismo, por lo que si el archivo a insertar contiene código PHP que deba ser evaluado (ejecutado), debe ser encerrado dentro de etiquetas de comienzo y fin de PHP.

Puedes también utilizar varios **include** anidados (es decir, utilizar **include** para llamar a otro archivo, dentro del archivo a incluir), con la única precaución de tener en cuenta que los archivos que se van insertando se ejecutan en el entorno del archivo primero que contiene la llamada, lo que es importante al establecer los paths.

Por ejemplo, si tienes dos archivos, `footer.php` y `notes.php` en un directorio llamado `anexos` y llamas al primero desde fuera de ese directorio, lo haríamos así:

```
include ("anexos/footer.php");
```

Y si `footer.php` contiene una llamada a otro archivo `notes.php`, debemos hacer la llamada también como:

```
include ("anexos/notes.php");
```

Puesto que aunque `footer` y `notes` están en el mismo directorio, la llamada a ambos es desde fuera, desde la misma posición que el script que hace la llamada.

Y lo mismo ocurre con las variables. El código insertado hereda el mismo alcance de variables que el código donde sea insertado.

#### Require

La diferencia documentada entre **include** y **require** consistiría en que la llamada con **include** podría hacerse condicionalmente. A diferencia de `include()`, `require()` siempre leerá el archivo referenciado, incluso si la línea en que está no se ejecuta nunca. Si se quiere incluir condicionalmente un archivo, se usa `include()`. La

sentencia condicional no afecta a `require()`, aunque si la línea en la cual aparece el `require()` no se ejecuta, tampoco se ejecutará el código del archivo referido.

Sin embargo, **a partir de la versión PHP 4.0.2 y posteriores esto ya no es así** y ambas construcciones se comportan exactamente de la misma manera, con la única diferencia de que si el archivo llamado no existe, **include** solo da una advertencia, y sigue ejecutando el código, mientras que **require** produce un error y interrumpe la ejecución.

---

### 3.5 Funciones PHP de acceso a MySQL

---

**MySQL** es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto. Se trata de un gestor de bases de datos multiusuario que gestiona bases de datos relacionales poniendo las tablas en ficheros diferenciados.

De uso extendido en plataformas Linux, puede usarse también en otras plataformas. Su uso en un servidor web es gratuito salvo en los casos que se necesite el uso de aplicaciones especiales.

A continuación vamos a ver todas las funciones que provee PHP para el manejo de bases de datos **MySQL**.

#### **mysql\_affected\_rows**

```
int mysql_affected_rows(int [link_identifier] );
```

`mysql_affected_rows` devuelve el número de filas afectado en el último `SELECT`, `UPDATE` o `DELETE` pregunta en el servidor asociado con el identificador de conexión especificado. Si no se especifica un identificador de conexión, se asume el de la última conexión abierta.

Este orden no es eficaz para las instrucciones `SELECT`, sólo en instrucciones que modifican archivos. Para recuperar el número de filas vuelto de un `SELECT`, usa `mysql_num_rows`.

#### **mysql\_close**

```
int mysql_close(int [link_identifier] );
```

Devuelve: `TRUE` si se ha cerrado correctamente, `FALSE` en caso de error.

`mysql_close` cierra la conexión a la base de datos MySQL asociada al identificador de conexión especificado. Si no se especifica un identificador de conexión, se asume el de la última conexión abierta.

Note que esta función no es normalmente necesaria en conexiones no-persistentes (abiertas con `mysql_connect`) ya que ésta se cerrará automáticamente al final de la ejecución del script o página. La función `mysql_close` no cierra una conexión persistente (abierta con `mysql_pconnect()`).



Ver tambien: `mysql_connect` y `mysql_pconnect`.

### **mysql\_connect**

```
int mysql_connect(string [hostname] , string [username] , string [password] );
```

Devuelve: un identificador de conexión, o FALSE en caso de error.

`mysql_connect` establece una conexión a un servidor de MySQL. Todos los argumentos son optativos, y si no se especifican, los valores por defecto son (' el localhost', nombre del usuario del usuario que posee el proceso del servidor, la contraseña vacía). La cadena hostname también puede incluir un número del puerto, "hostname:port".

En caso de realizar una segunda llamada a `mysql_connect` con los mismos argumentos, no se establecerá ninguna nueva conexión, sino se devolverá el identificador de conexión de la ya existente.

La conexión al servidor se cerrará en cuanto la ejecución del script acabe, a menos que la cerremos antes con la función `mysql_close`.

Ver también: `mysql_pconnect` y `mysql_close`.

### **mysql\_create\_db**

```
int mysql_create_db(string database name, int [link_identifier] );
```

La función `mysql_create_db` intenta crear una nueva base de datos en el servidor asociado con el identificado de conexión especificado.

Ver también: `mysql_drop_db`.

### **mysql\_data\_seek**

```
int mysql_data_seek(int result_identifier, int row_number);
```

Devuelve: TRUE si toda ha ido bien, y FALSE en caso de error.

La función `mysql_data_seek` mueve el puntero que indica la fila actual al número de fila de la consulta que indica el identificador. La próxima llamada al `mysql_fetch_row` o `mysql_fetch_array` devolvería esa fila.

Ver también: `mysql_data_seek`.

### **mysql\_dbname**

```
string mysql_dbname(string result, int i);
```

`mysql_dbname` devuelve el nombre de la base de datos guardado en posición `i` de los resultados del indicador de consulta devuelto por la función del `mysql_list_dbs`. La función del `mysql_num_rows` puede usarse para determinar cuántos nombres de bases de datos están disponibles.

### **mysql\_db\_query**

```
int mysql_db_query(string database, string query, int link_identifier);
```

Devuelve: un identificador de conexión, o FALSE en caso de error.

Ejecuta una consulta en una base de datos. Si el identificador no se especifica, la función intenta encontrar una conexión abierta con el servidor. Si no encuentra una conexión, intentará crear una (similar a `mysql_connect( )` sin argumentos).

Ver también: `mysql_connect`.

### **mysql\_drop\_db**

```
int mysql_drop_db(string database_name, int [link_identifier] );
```

Devuelve: TRUE si toda ha ido bien, y FALSE en caso de error.

Elimina una base de datos del servidor asociado al identificador de conexión.

Ver también: `mysql_create_db`

### **mysql\_errno**

```
int mysql_errno();
```

Devuelve el número de error asociado a la última operación realizada.

Ver también: `mysql_error`

### **mysql\_error**

```
string mysql_error();
```

Devuelve el texto asociado al error producido en la última operación realizada por la base de datos.

```
<?php
mysql_connect("marliesle");
echo mysql_errno().": ".mysql_error()."<BR>";
mysql_select_db("nonexistentdb");
echo mysql_err_db("nonexistentdb");
echo mysql_errno().": ".mysql_error()."<BR>";
$conn = mysql_query("SELECT * FROM nonexistenttable");
echo mysql_errno().": ".mysql_error()."<BR>";
?>
```

Ver también: `mysql_errno`



### mysql\_fetch\_array

```
array mysql_fetch_array(int result);
```

Devuelve un array con la información correspondiente al resultado de una consulta especificado por su identificador o 'false' si ya no hay más filas.

Es una versión extendida de `mysql_fetch_row( )`. Además de almacenar los datos a través de índices numéricos del array, también lo hace a través de índices asociativos, utilizando los nombres de los campos como claves.

Si dos o más columnas del resultado tienen el mismo nombre de campo, la última es la que tiene preferencia. Para acceder a las demás es necesario utilizar el índice numérico o construir un alias para la columna:

```
select t1.f1 as foo t2.f1 as bar from t1, t2 as foo t2.f1 as bar from  
t1, t2
```

Esta función no es más lenta que `mysql_fetch_row( )`.

### Ej.

```
<?php  
mysql_connect($host,$user,$password);  
$result = mysql_db_query("database","select * from table");  
while($row = mysql_fetch_array($result)) {  
echo $row["user_id"];  
echo $row["fullname"];  
}  
mysql_free_result($result);  
?>
```

### mysql\_fetch\_field

```
object mysql_fetch_field(int result, int [field_offset] );
```

Devuelve un objeto que contiene la información de los campos que componen un resultado de una consulta. Si no se especifica 'offset', devuelve información sobre el siguiente campo que todavía no ha sido devuelto.

Propiedades del objeto devuelto:

- `name` - nombre del campo
- `table` - nombre de la tabla a la que pertenece el campo
- `max_length` - longitud máxima que puede tomar el campo
- `not_null` - 1 si el campo no puede tomar valores nulos
- `primary_key` - 1 si el campo es una clave principal (primary key)
- `unique_key` - 1 si el campo tiene restricción de unicidad
- `multiple_key` - 1 si el campo no tiene rest
- `multiple_key` - 1 si el campo no tiene restricción de unicidad
- `numeric` - 1 si el campo es numérico

- blob - 1 si el campo es BLOB
- type - tipo de dato (del campo)
- unsigned - 1 si el campo es 'unsigned'
- zerofill - 1 si el campo es rellenado con ceros

Ver también: `mysql_field_seek`

### **mysql\_fetch\_lengths**

```
int mysql_fetch_lengths(int result);
```

Devuelve: una tabla o `FALSE` si hay error.

`mysql_fetch_lengths` almacena en una tabla la longitud de cada campo de una consulta realizada con `mysql_fetch_row` o `mysql_fetch_array`. El índice de la tabla comienza en 0.

Ver también: `mysql_fetch_row`.

### **mysql\_fetch\_object**

```
int mysql_fetch_object(int result);
```

Devuelve: un objeto o `FALSE` en caso de error.

Esta función es similar a `mysql_fetch_array`, solo que los resultados de una consulta, en lugar de una tabla, los devuelve como un objeto. En este caso, sólo se puede acceder a los datos a través de los nombres de sus campos. La velocidad de ejecución es idéntica a la de `mysql_fetch_array`. Para referenciar el valor de un campo debemos utilizar el operador típicos de los objetos (->).

Ver también: `mysql_fetch_array` y `mysql_fetch_row`.

### **mysql\_fetch\_row**

```
array mysql_fetch_row(int result);
```

Devuelve: una tabla o `FALSE` si hay error.

Devuelve un tabla con los valores de los campos de la fila actual de la consulta, la que especificar el indicador (`result`), y mueve el puntero interno que marca la fila actual a la siguiente fila, si no hay mas filas devuelve `FALSE`. El índice de la tabla comienza en 0.

Ver también: `mysql_fetch_array`, `mysql_fetch_object`, `mysql_data_seek`, `mysql_fetch_lengths`, y `mysql_result`.

### **mysql\_field\_name**

```
string mysql_field_name(string result, int i);
```

Devuelve el nombre del campo especificado por el índice.



### **mysql\_field\_seek**

```
int mysql_field_seek(int result, int field_offset);
```

Mueve el puntero del campo actual hacia adelante las posiciones actual hacia adelante las posiciones indicadas por 'offset'.

Ver también: `mysql_fetch_field`.

### **mysql\_field\_table**

```
string mysql_field_table(int result, int field_offset);
```

Devuelve el nombre de la tabla que almacena el campo especificado por el índice ( 'field\_offset' ).

### **mysql\_field\_type**

```
string mysql_field_type(string result, int field_offset);
```

Devuelve el tipo del campo del índice especificado.

### **mysql\_field\_flags**

```
string mysql_field_flags(string result, int field_offset);
```

Devuelve los especificadores (flags) del campo especificado como una cadena de texto en la que cada especificador se corresponde con una palabra, y éstas van separadas mediante un espacio simple. Se puede analizar la cadena utilizando `explode( )`

Los especificadores son: "not\_null", "primary\_key", "unique\_key", "multiple\_key", "blob", "unsigned", "zerofill", "binary", "enum", "auto\_increment", "timestamp".

### **mysql\_field\_len**

```
int mysql_field_len(string result, int field_offset);
```

Devuelve la longitud del campo especificado

### **mysql\_free\_result**

```
int mysql_free_result(int result);
```

Sólo debería ser utilizada si la cantidad de memoria utilizada para almacenar el resultado de una consulta es muy grande. Cuando se ejecuta esta función, toda la memoria asociada al resultado se libera.

### **mysql\_insert\_id**

```
int mysql_insert_id(void);
```

Esta función devuelve el ID (identificador) generado para los campos autonuméricos (AUTO\_INCREMENTED). El ID devuelto es el correspondiente al de la última operación INSERT.

### **mysql\_list\_fields**

```
int mysql_list_fields(string database, string tablename);
```

Devuelve información sobre la tabla. El valor resultante puede ser utilizado con `mysql_field_flags()`, `mysql_field_len()`, `mysql_field_name()`, y `mysql_field_type()`.

El identificador que devuelve es un entero positivo o '-1' en caso de error. El texto que describe el error se encuentra en `$phperrmsg`.

### **mysql\_list\_dbs**

```
int mysql_listdbs(void);
```

Devuelve un puntero que contiene las bases de datos disponibles para el servidor actual (mysql daemon). Este valor se utiliza con la función `mysql_dbname()`.

### **mysql\_list\_tables**

```
int mysql_list_tables(string database, , int [link_identifier]);
```

Devuelve un identificador, el cual pasaremos a la función `mysql_tablename` para extraer el nombre de las tablas de la base de datos especificada.

### **mysql\_num\_fields**

```
int mysql_num_fields(int result);
```

`mysql_num_fields` devuelve el número de campos de una consulta.

Ver también: `mysql_db_query`, `mysql_query`, `mysql_fetch_field`, `mysql_num_rows`.

### **mysql\_num\_rows**

```
int mysql_num_rows(string result);
```

Devuelve el número de filas del resultado de una consulta.

Ver también: `mysql_db_query`, `mysql_query` and, `mysql_fetch_row.query`, `mysql_query` y, `mysql_fetch_row`.

### **mysql\_pconnect**



```
int mysql_pconnect(string [hostname] , string [username] , string [password] );
```

Devuelve: A positive MySQL persistent link identifier on success, or false on error

Devuelve un identificador de conexión persistente o 'false' en caso de error. Las diferencias con respecto a `mysql_connect( )` son:

- Primero, la función intenta encontrar una conexión persistente que ya esté abierta con la misma máquina, usuario y password. Si es encontrada, devuelve el identificador de la misma, en lugar de crear una nueva conexión.
- Segundo, la conexión al servidor SQL no será cerrada cuando finalice la ejecución del script, sino que permanece abierta para un uso posterior.

La función `mysql_close` no cierra una conexión abierta con `mysql_pconnect`. Las conexiones abiertas con esta función se llaman "persistentes".

### **mysql\_query**

```
int mysql_query(string query, int [link_identifier] );
```

Ejecuta una consulta a la base de datos activa en el servidor asociado al identificador de conexión. Si no se especifica, se utiliza la última conexión abierta. Si no hay conexiones abiertas la función intenta establecer una.

Esta función devuelve TRUE o FALSE para indicar si las operaciones UPDATE, INSERT o DELETE han tenido éxito. Para la operación SELECT devuelve un nuevo identificador de resultado.

Ver también: `mysql_db_query`, `mysql_select_db`, y `mysql_connect`.

### **mysql\_result**

```
int mysql_result(int result, int row, mixed field);
```

Devuelve el contenido de la celda de un resultado. El argumento 'field' puede ser un índice o el nombre del campo correspondiente o el nombre del campo de la forma: `tabla.campo`. Si la columna tiene un alias ('select foo as bar from...') se utiliza el alias en lugar del nombre de la columna.

En lugar de esta función es preferible usar `mysql_fetch_row()`, `mysql_fetch_array()`, and `mysql_fetch_object()`, con la que obtendremos mejor rendimiento.

### **mysql\_select\_db**

```
int mysql_select_db(string database_name, int [link_identifier] );
```

Devuelve: true on success, false on error

Establece la base de datos activa en el servidor. Si no se especifica identificador de conexión se utiliza la última conexión abierta. Si no hay conexiones anejió abierta. Si no hay conexiones activas, la función intenta establecer una. A partir de la llamada a `mysql_select_db` las llamadas a `mysql_query( )` actúan sobre la nueva base de datos activa.

Ver también: `mysql_connect`, `mysql_pconnect`, and `mysql_query`

### **mysql\_tablename**

```
string mysql_tablename(int result, int i);
```

Toma como argumento un puntero devuelto por la función `mysql_list_tables( )` y devuelve el nombre de la tabla asociada al índice `i`. La función `mysql_num_rows( )` puede ser utilizada para determinar el número de tablas.

```
<?php
mysql_connect ("localhost:3306");
$result = mysql_listtables ("basededatos");
$i = 0;
while ($i < mysql_num_rows ($result)) {
    $tb_names[$i] = mysql_tablename ($result, $i);
    echo $tb_names[$i] . "<BR>";
    $i++;
}
?>
```

---

## 3.6 Creación de una base de datos en MySQL

---

Una vez instalado **MySQL**, vamos a crear una base de datos de ejemplo. Vamos a ver dos formas de crear una base de datos y sus tablas.

### **Línea de comandos**

Para crear nuestra base de datos en sistemas Linux/Unix, debemos ser el administrador de **MySQL** o tener el permiso pertinente para crear bases de datos, para ello haremos lo siguiente:

```
mysqladmin create mydb
```

Ya hemos creado una base de datos, ahora le añadiremos una tabla (Ej. **agenda**):

```
CREATE TABLE agenda (id INT NOT NULL AUTO_INCREMENT, nombre
CHAR(50),direccion CHAR(100), telefono CHAR(15), email CHAR(50), KEY
(id) )
```

A continuación vamos a insertar una serie de registros:



```
INSERT INTO agenda VALUES (0, 'Juan Pérez', 'C/ Laguna, 15.
Sevilla', '95.455.55.55', 'juan@agenda.com' )
```

```
INSERT INTO agenda VALUES (1, 'Luis García', 'C/ Betis, 22. Cádiz',
'95.655.66.33', 'luis@agenda.com' )
```

```
INSERT INTO agenda VALUES (2, 'Carlos Rodríguez', 'C/ Sevilla, 6.
Huelva', '95.113.22.77', 'carlos@agenda.com' )
```

Debemos tener en cuenta que los comandos de arriba debe escribirse cada uno en una sola línea. Se han separado para aumentar la legibilidad del código.

Ahora desde la línea de comandos ejecuta:

```
cat mydb.dump | mysql mydb
```

Cabe destacar el campo id, que no puede estar vacío, y además es autoincrementable, lo cuál deberemos tener en cuenta a la hora de actualizar y añadir registros. Si no hemos cometido ningún error, ya tenemos nuestra base de datos de ejemplo en el servidor **MySQL**.

## Script

Una segunda forma de crear las bases de datos y tablas es utilizar las funciones que para ello nos da PHP. Para crear una base de datos tenemos dos opciones, una utilizar la función `mysql_create_db()`, o bien enviando una consulta SQL con la instrucción "CREATE DATABASE mydb". Como ejemplo vamos a crear un script **crear\_my.php** que creará la anterior base de datos.

```
<html>
<body>
<?PHP
define ("CONSTANTE", "Hola Mundo");
printf (CONSTANTE);
?>
</body>
</html>
$basedatos = "mydb";
//conectamos con el servidor
$link = @mysql_connect("localhost", "root", "");

// comprobamos que hemos establecido conexión en el servidor
if (! $link)
{
echo "<h2 align='center'>ERROR: Imposible establecer conexión con el
servidor</h2>";
exit;
}
// obtenemos una lista de las bases de datos del servidor
$db = mysql_list_dbs();
```

```
// vemos cuantas BD hay
$num_bd = mysql_num_rows($db);

//comprobamos si la BD que queremos crear existe ya
$existe = "NO" ;
for ($i=0; $i<$num_bd; $i++)
{
if (mysql_dbname($db, $i) == $basedatos)
{
$existe = "SI" ;
break;
}
}

// si no existe la creamos
if ($existe == "NO")
{
/* manera 1 */
if (! mysql_create_db($basedatos, $link))
{
echo "<h2 align='center'>ERROR 1: Imposible crear base de
datos</h2>";
exit;
}
/* class="codigo" style="margin-left: 50"> /* manera 2
if (! mysql_query("CREATE DATABASE $basedatos", $link))
{
echo "<h2 align='center'>ERROR2: Imposible crear base de datos</h2>";
exit;
} */
}

// creamos la tabla
$sql = "CREATE TABLE agenda (";
$sql .= "id INT NOT NULL AUTO_INCREMENT, ";
$sql .= "nombre CHAR(50), ";
$sql .= "direccion CHAR(100), ";
$sql .= "telefono CHAR(15), ";
$sql .= "email CHAR(50), ";
$sql .= "KEY (id) ) ";

if (@mysql_db_query($basedatos, $sql, $link))
{
echo "<h2 align='center'>La tabla se ha creado con éxito</h2>";
}
else
{
echo "<h2 align='center'>No se ha podido crear la tabla</h2>";
}

?>

</body>
</html>
```

---

### 3.7 Mostrar los datos de una consulta

---



El Lenguaje de Consulta Estructurado (**SQL** o **Structured Query Language**) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.

A continuación vamos a ver una serie de consultar para recuperar y modificar los datos contenidos en una tabla.

### Consulta de selección

Para mostrar los datos de una tabla, tenemos disponible la consulta de selección, representada por la palabra reservada **SELECT**.

Esta consulta nos permite personalizar los datos devueltos, pudiendo seleccionar campos y registros concretos de la tabla mediante una serie de criterios de selección.

Veamos a continuación una posible estructura para las consultas de selección:

```
SELECT (campo_1, campo_2 , ..., campo_n | *) FROM tabla [WHERE condición] [GROUP BY campo_1, campo_2, ... campo_n] [HAVING condición] ORDER BY campo [ASC | DESC];
```

Donde:

- **campo\_1, campo\_2, ..., campo\_n**: lista de campos de la tabla que queremos mostrar en la consulta
- **\***: muestra todos los campos de la tabla
- **tabla**: nombre de la tabla (o tablas) sobre la que vamos a mostrar datos en la consulta
- **WHERE condición**: cláusula que incluye criterio de selección, se aplica sobre las columnas (campos) y filas (registros) de la tabla. Aquellos campos y registros que cumplan con el criterio de selección serán devueltos en la consulta.
- **GROUP BY campo\_1, campo\_2, ..., campo\_n**: cláusula opcional, permite agrupar los datos devueltos por uno o varios campos
- **HAVING condición**: permite establecer un criterio de selección sobre el agrupamiento.
- **ORDER BY campo [ASC | DESC]**: esta cláusula es opcional, y permite ordenar los registros devueltos por la consulta en base a uno de los campos de la tabla, con posibilidad de ordenarlo en modo ascendente o descendente

---

## 3.8 Añadir, modificar y borrar registros

---

Así como la consulta de selección, existen otra serie de consultas propias del lenguaje SQL para añadir, modificar y borrar registros de una tabla. Veamos una a una las consultas disponibles:

La estructura de la consulta para poder insertar nuevos registros (filas) en una tabla es la siguiente:

**INSERT INTO tabla VALUES (valor\_1, valor\_2, ..., valor\_n)**

Donde:

- **tabla**: tabla destino donde vamos a insertar el registro.
- **(valor\_1, valor\_2,..., valor\_n)**: valores para los distintos campos del registro. Deben coincidir en nº con el nº de campos de la tabla

### Actualización de registros

La estructura de la consulta para poder actualizar registros en una tabla es la siguiente:

**UPDATE tabla SET campo\_1=valor\_1, campo\_2=valor\_2, ..., campo\_n=valor\_n  
[WHERE condicion]**

Donde:

- **tabla**: tabla destino donde vamos a insertar el registro.
- **campo\_1, campo\_2,...,campo\_n**: son los distintos campos de la tabla a los que se les actualiza el valor
- **valor\_1, valor\_2,..., valor\_n**: nuevos valores para los distintos campos del registro.
- **WHERE condicion**: cláusula opcional que indica los registros sobre los cuales se aplicará la actualización



## Eliminación de registros

La estructura de la consulta para poder eliminar registros de una tabla es la siguiente:

**DELETE FROM tabla [WHERE condicion]**

Donde:

- **tabla:** nombre de la tabla en la que vamos a borrar el registro o registros
- **WHERE condicion:** cláusula opcional que permite indicar un criterio de selección de registros, por lo que sólo se eliminarán aquellos que cumplan la condición especificada.

---

## 3.9 Ejercicios

---

### Ejercicio 1

¿Qué mostrarían por pantalla las siguientes sentencias en PHP?

#### 1) Código 1

```
<?php
    $var1="rompe";
    $var2="-";
    $var3="cabezas";
    echo "Esto se ha convertido en un " . $var1 . $var2 . $var3;
?>
```

#### 2) Código 2

```
<?php
    $edad=18;

    if($edad >= 18)
    {
        echo "Ya eres mayor de edad";
    }
    else
    {
        echo "Aún no ha adquirido la mayoría de edad";
    }
?>
```

## Ejercicio 2

¿Cuál es el resultado de los siguientes fragmentos de código?

### 1) Código 1

```
<?php
$num = 1;
do {
    echo $num . "<br>";
    if ($num == 3)
    {
        echo "Aquí nos salimos \n";
        break
    }
    $num++
} while ($num < 5);
?>
```

### 2) Código 2

```
<?php
$opcion=5;
switch ($opcion)
{
    case 1:
        echo "OPCIÓN 1: Crear un nuevo documento";
        break;
    case 2:
        echo "OPCIÓN 2: Abrir un documento existente";
        break;
    case 3:
        echo "OPCIÓN 3: Guardar documento";
        break;
    case 4:
        echo "OPCIÓN 4: Eliminar un documento";
        break;
    case 5:
        echo "OPCIÓN 5: Proteger el documento con contraseña";
        break;
    case 6:
        echo "OPCIÓN 6: Imprimir documento";
        break;
    case 7:
        echo "OPCIÓN 7: Enviar documento por e-mail";
        break;
    default:
        echo "ERROR: la opción escogida es incorrecta";
}
?>
```

## Ejercicio 3



Crear una base de datos en MySQL que llamaremos `USUARIOS_BD`. En dicha base de datos, crear una tabla con el nombre `usuarios`, que contendrá los siguientes campos, con sus tipos de datos asociados:

- DNI (Entero)
- Nombre (Cadena)
- Apellidos (Cadena)
- Edad (Entero)
- Aficiones (Cadena)

Una vez creada la tabla, insertar en dicha tabla los siguientes datos de usuarios:

**TABLA USUARIOS**

DNI	Nombre	Apellidos	Edad	Aficiones
75267823K	Juan	Chapas Verdes	29	Me gusta el surf, los deportes de riesgo y la música clásica.
34889201J	Ana María	Torla Simón	45	Me encanta la cocina italiana, las novelas de terror y los animales en general.
56728182U	Estanislao	Girón Tormes	39	Me encantan los coches, salir a tomar cañas y el ciclismo.
87376652A	Andrés	Juaca Perlón	65	Aficionado al fútbol, experto en petanca y asiduo de las plazas de toros.

A continuación realizar las siguientes consultas sobre la base de datos:

- Obtener los datos de todos los usuarios registrados en la base de datos (`listaUsuarios.php`)
- Obtener los datos de aquellos usuarios cuya edad está comprendida entre los 35 y 50 años (`listaUsuariosEntre35_50.php`)
- Insertar los datos del usuario que se muestra a continuación (`insertarUsuario.php`):

DNI	Nombre	Apellidos	Edad	Aficiones
75222453K	Carlos	Sáinz De Bovedilla	18	Me encanta la música pop, salir con los amigos y quedar para jugar a juegos de rol.

- Actualizar la edad del usuario anteriormente insertado a 21 años (`actualizaUsuario.php`)
- Eliminar los datos del usuario con DNI **56728182U** (`eliminaUsuario.php`)

Después de llevar a cabo cada una de las consultas que implican una inserción, actualización o eliminación de datos de la tabla, se debe volver a la página que contiene la consulta que muestra todos los usuarios almacenados en la BD (`listaUsuarios.php`) para comprobar que los cambios se han realizado correctamente.