

# 10 Líneas De Comando Que Todo Diseñador Web Debe Conocer

## 10 Líneas De Comando Que Todo Diseñador Web Debe Conocer

Los [diseñadores web](#) muchas veces encuentran instrucciones como `npm install` ó `git clone`, y muchas más. Estas son [Command Line Interfaces](#) ó Interfaces de Línea de Comandos ([CLI](#)). Este tipo de comandos se usan para indicar al ordenador que debe realizar tareas específicas, generalmente **en lugar de escribir comandos específicos** en la terminal ó el símbolo del sistema.



command lines |

La Terminal y el Símbolo del sistema no pueden ser la herramienta más intuitiva de usar, especialmente para los diseñadores web, y esto es bastante comprensible, ya que los **diseñadores web** pueden estar más familiarizados con las **interfaces gráficas**. Sin embargo, herramientas como [Yeoman](#), [Bower](#) y [Google Web Starter Kit](#) operan a través de líneas de comandos. Aunque no seas un ferviente seguidor de las líneas de comando, de seguro en este artículo encontrarás el impulso para mirar el mundo de posibilidades que te brinda la Terminal ó Línea de comandos.

# Pero primero ...

Tenemos que hablar sobre que es la **Terminal** y el **Símbolo del sistema**. Ambas son aplicaciones de gran alcance que le dan acceso al núcleo del respectivo sistema operativo. Una vez que se realiza un cambio a través de estas herramientas, el cambio no es reversible por lo que cualquier cosa que hagas con estas dos aplicaciones, debes realizarse con precaución, pero no te preocupes, esto es tan solo una advertencia, y en este artículo encontraras una explicación para cada función ;)

Tampoco serás capaz de usar el ratón para mover el cursor en la terminal é en el Símbolo del sistema. Todo se hace con el teclado y, por tanto, los atajos de teclado son tus nuevos mejores amigos. Tenga en cuenta que algunas líneas de **comandos útiles** no se pueden encontrar en [Windows](#). Si eres **usuario de Windows**, te aconsejamos usar herramientas como [Cygwin](#), [UnxUtils](#) o Windows Services for UNIX versión 3.5 que trae algunas utilidades de UNIX que pueden ser usadas en Windows. Ahora si, comencemos con los dichosos comandos útiles.

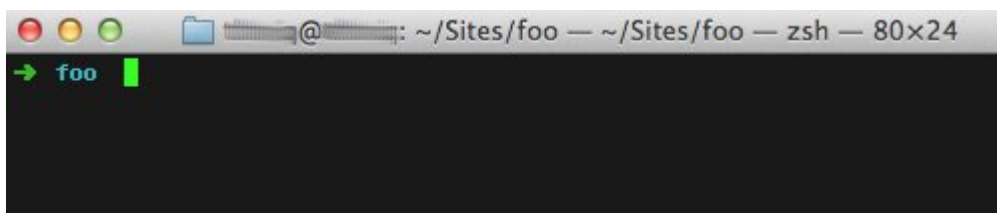
## 1. Cambiar Directorio

A menudo, tendrá que navegar a través de los directorios de archivos. La Terminal y consolas de comando, usan el mismo comando **cd** para cambiar el directorio actual al destino especificado en el comando. Digamos que usted quiere ir a una carpeta llamada foo, tendríamos que cambiar de directorio

justo de la siguiente manera:

```
[bash]cd foo[/bash]
```

Puede ver a continuación, el directorio actual se muestra en la siguiente ventana:



Puede navegar directamente al sub-directorio de foo, justo de la siguiente manera:

```
[bash]cd foo/sub-folder &&nbsp;[/bash]
```

Para regresar al directorio anterior o ir a un nivel superior del directorio actual, escriba:

```
[bash]cd .. [/bash]
```

Fácil, no? Prepárate por que tan solo estamos calentando ;)

## 2. Crear Una Nueva Carpeta

Otro comando que puede encontrar necesario para usar, es **mkdir**. En el anterior comando vemos como podemos hacer para movernos entre carpetas, con este comando creamos un nuevo directorio con el nombre especificado. El siguiente comando, por ejemplo, se creará un nuevo directorio llamado foo.

```
[bash]mkdir foo &&nbsp;[/bash]
```

También podemos crear varias carpetas a la vez. Este ejemplo de abajo creará tres directorios llamados foo, hola, y mundo, todos a la vez.

```
[bash]mkdir foo hola mundo &nbsp;&nbsp; [/bash]
```

El comando **mkdir** es compatible tanto en la Terminal y Símbolo del sistema. Este comando es útil cuando deseamos crear la estructura de archivos de nuestro proyecto, bastante importante cuando deseamos organizar el contenido como imágenes, archivos css, js, html, php, etc, etc.

## 3. Crear Un Nuevo Archivo

Bueno, ahora que ya podemos crear nuestra estructura de archivos con el anterior comando, ¿Cual sería el siguiente paso? Los archivos, sin estos de nada serviría tener una estructura de archivos. Utilice el **comando touch** para crear un archivo vacío. Por ejemplo:

```
[bash]touch nombearchivo.html &nbsp;&nbsp; [/bash]
```

Puede especificar varios nombres de archivo de la siguiente manera, para crear varios archivos a la vez.

```
[bash]touch index.html estilo.css [/bash]
```

# 4. Mover Archivos

Bastante útil crear los archivos, ¿verdad? Pero, ¿Qué pasaría si los creamos en un directorio equivocado? Para esto usamos el comando `mv` para mover un archivo a una carpeta determinada. En el siguiente ejemplo, se mueve el archivo `estilo.css` a la carpeta `css`, el `/` indica que la carpeta se encuentra dentro del mismo directorio que el del archivo, podemos usar rutas para mover el archivo.

```
[bash]mv estilo.css /css &&nbsp;&nbsp; [/bash]
```

También puede hacer uso del comando `mv` para re nombrar archivos y carpetas. En el siguiente ejemplo se cambiará el nombre de `index.html` a `about.html`.

```
[bash]mv index.html about.html &&nbsp;&nbsp; [/bash]
```

# 5. Copiar Archivos

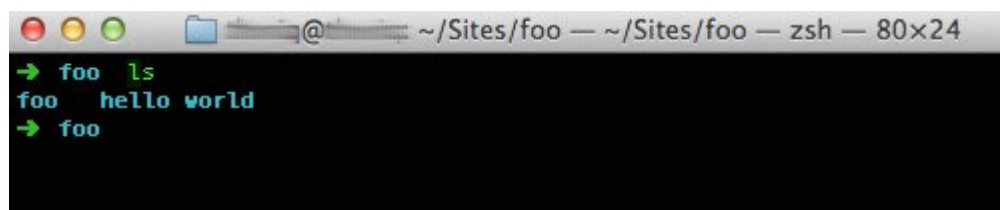
Muchas veces es necesario **copiar archivos**, ¿no crees útil este comando? ¿Acaso no guardas respaldos? Para esto usamos el comando `cp`, si desea copiar un archivo ó carpeta. A continuación se muestra un ejemplo donde copiamos `index.html` y nombre el archivo nuevo a `about.html`.

```
[bash]cp index.html about.html[/bash]
```

Si está usando la **consola en Windows**, utilice el comando `cp` directamente.

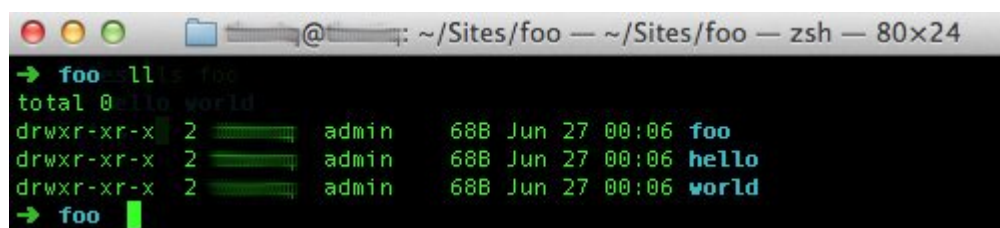
# 6. Listar El Contenido Del Directorio

Conocer los archivos o el contenido de un directorio es bastante importante en cualquier desarrollo. Es por esto, que este es uno de los comandos que se usa a menudo, con el comando `ls` podrás listar ó mostrar los archivos que se encuentren en el directorio.



```
~/Sites/foo — ~/Sites/foo — zsh — 80x24
→ foo ls
foo hello world
→ foo
```

Por otra parte, también puede obtener los detalles del contenido mostrado como la fecha de directorio (creado), el permiso de directorio, y los propietarios de directorio. Para ello, escriba `ls -l` ó simplemente `ll`.



```
~/Sites/foo — ~/Sites/foo — zsh — 80x24
→ foo ll
total 0
drwxr-xr-x  2 admin  68B Jun 27 00:06 foo
drwxr-xr-x  2 admin  68B Jun 27 00:06 hello
drwxr-xr-x  2 admin  68B Jun 27 00:06 world
→ foo
```

El comando `ls`, sin embargo, sólo funcionará en consolas UNIX. Puede ejecutar el comando `ls` en Ubuntu y Mac OS X, pero no en Windows. En Windows, escriba comando `dir` en su lugar.

```
C:\Windows\system32\cmd.exe
C:\Users\████████\foo>dir
Volume in drive C has no label.
Volume Serial Number is E0F9-04CD

Directory of C:\Users\████████\foo

06/27/2014  03:03 AM    <DIR>          .
06/27/2014  03:03 AM    <DIR>          ..
06/27/2014  03:03 AM    <DIR>          foo
06/27/2014  03:03 AM    <DIR>          hello
06/27/2014  03:03 AM    <DIR>          world
               0 File(s)              0 bytes
               5 Dir(s)  47,086,804,992 bytes free

C:\Users\████████\foo>_
```

# 7. Abrir Archivos

El comando `open`, abrirá los archivos de las carpetas con la aplicación por defecto. Por ejemplo, en nuestro caso en particular, se abrirá el escritorio en Finder (gestor de archivos en Mac OS X).

```
[bash]open ~/Desktop[/bash]
```

El siguiente comando abrirá el editor de texto por defecto del sistema operativo, mostrando el archivo `txt`.

```
[bash]open leeme.txt &&[/bash]
```

Los usuarios de Windows deben usar el comando `edit`. Dado el mismo ejemplo, puede ejecutar:

```
[bash]edit leeme.txt &&[/bash]
```

# 8. Creación De Enlaces Simbólicos

El enlace simbólico funciona como una **carpeta de acceso directo**, pero el sistema se trata como si fuese una carpeta real. Este comando es bastante útil cuando deseamos sincronizar carpetas con respaldos unidas a Dropbox. El comando se usa de la siguiente manera:

```
[bash]ln -s /origen /destino[/bash]
```

Como ejemplo práctico, supongamos que desea enlazar su carpeta de dropbox con la carpeta de respaldos ubicada en Documentos:

```
[bash]ln -s ~/Dropbox/proyecto ~/Documentos/respaldos[/bash]
```

Los usuarios de Windows pueden utilizar el comando **mklink /d**.

# 9. Uso De l



# Editor Nano

Aunque no encontremos una interfaz grafica como a la que normalmente usamos, podemos editar textos desde la consola. Digamos que necesitamos crear un nuevo **host virtual** con un nuevo nombre de dominio. Aquí es donde tendrá que editar el archivo `hosts` para registrar el nombre de dominio y la dirección IP a la cual apunta. La forma más rápida de editar el archivo `hosts` es escribiendo.

```
[bash]sudo nano /etc/hosts &nbsp;&nbsp; [/bash]
```

# 10. CLI Sublime Text

Sin duda alguna Sublime Text es un potente pero sencillo editor de texto, además podemos usarlo en modo CLI. Con el comando **subl** se puede, por ejemplo, abrir un archivo. El comando, sin embargo, aún no se reconocerá cuando se escribe `subl` en el Terminal.

Para agregar el CLI de Sublime Text, ejecute este comando primero.

```
[bash]ln -s &quot;/Applications/Sublime Text.app/Contents/SharedSupport/bin/subl&quot; ~/bin/subl[/bash]
```

Después de eso, debe tener acceso a la `subl`. Ejecutar el siguiente comando, por ejemplo, se abrirá el archivo

```
style.css.
```

```
[bash]subl style.css & & &nbsp;[/bash]
```

Incluyendo el comando `-add` abrirá el archivo o carpeta en la ventana de Sublime Text.

```
[bash]subl -add foo & & &nbsp;[/bash]
```

Para otros usos, puede escribir `subl -help`.

# Finalmente

Una vez que haya visto lo fácil que es manejar la consola de tu sistema operativo, de seguro querrás conocer mas comandos para realizar tareas que normalmente podrías hacer con una GUI. Los anteriores comando tan solo fueron un abre bocas, puedes usar funciones mas avanzadas, como por ejemplo conectarte a través de la terminal a tu servidor mediante SSH. ¿Conoces otro comando útil para cualquier diseñador?, compártelo en un comentario ;)