

<b>Miembros del grupo de estudios .....</b>	<b>1</b>
<b>Algunos criterios.....</b>	<b>1</b>
<b>Materiales de estudio .....</b>	<b>3</b>
.....	<b>3</b>
2.1.- "Los comodines" y 2.2.- "El comodín tilde" .....	3
El comando "ls" .....	5
Desafío N° 1: ¿Cómo buscar recursivamente todos los archivos de backup que van quedando por ahí, para poder eliminarlos y limpiar las carpetas? .....	7
Script N°1 (Alejandro) .....	12
Desafío N° 2 ¿Cómo reorganizar el script 1 por colores? .....	13
Script N° 2 (Fabricio).....	13
Script N° 3 (Alejandro) .....	14
Script N° 4 (Fabricio).....	16
Desafío N° 3 ¿Cómo simplificarían el script de Fabricio? .....	17
2.3.- Comodín llaves .....	17
Script N° 5 (Jorge).....	18
Script N° 6 (Rodrigo).....	19
Premio "Rex Tux Scripting" del desafío número 3.....	20
Script N° 7 (Alejandro) .....	20
Script N°8 (Fabricio).....	21
Premio "Rex Tux scripting" del desafío 4.....	23
Script N°9 (Diego) .....	24
2.4.- Los comodines extendidos.....	25
<b>AUTOEVALUACIÓN Final (primera semana) .....</b>	<b>32</b>
Último desafío: .....	34

---

## Miembros del grupo de estudios

María, Fabricio, Diego, Sergio --> **Argentina**

Jorge, Rodrigo --> **Chile**

Alejandro --> **Brasil**

---

## Algunos criterios

Hola gente!

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

- La idea es ir haciendo esto entre todos, así que, somos todos dueños del proceso. Ya le iremos "agarrando la mano".
- Propongo que nos fijemos **metas semanales**, ni muy ambiciosas ni demasiado pobres, para sostener un ritmo y para que no decaiga el interés. La semana podría comenzar todos los jueves.
- Propongo también, **que cada uno elija un color para identificarse** cuando hace un comentario. De esa forma, se va registrando también un orden temporal.
- **Formato:** para los comandos o ejemplos de scripts, recomiendo que usemos la fuente "Courier New" y la opción "Formato" --> "Sangrar párrafo"; y para el texto común "Verdana".

Ejemplo:

```
#!/bin/bash
echo "Hola Mundo"
```

- Cada intervención puede separarse por medio de una barra horizontal ("Insertar"-->"Línea Horizontal").
- Podemos usar íconos para identificar si la intervención contiene un script, si es un desafío, etc.
- Google Docs tiene un **historial de revisiones** que permite comparar las distintas versiones del documento ("Herramientas" --> "Historial de revisiones"). Digo esto, porque a veces no se percibe que hubo una edición y uno se pierde alguna información importante. De todas formas, creo que lo mejor en el caso de un código, es no editarlo y agregar la nueva versión a continuación. Ya veremos qué es más práctico.
- Esta primera semana, del 8 al 14, podríamos ver los puntos 2.1 y 2.2 del Tema 1 (El Shell Bash de Fernando López Hernández) sobre "Los comodines" y "El comodín tilde". Claro que cada uno, según sus tiempos, puede avanzar mucho más. Pero me parece que grupalmente deberíamos **acotar la cantidad de información a tratar**. En todo caso, después vamos regulando si es mucho o si es poco.
- Creo que la **mecánica de trabajo** debería ser: cada uno va leyendo y, cuando se traba en algo; surge una duda; se le ocurre una situación-problema para resolver; o cuando puede complementar con información de otro lado, etc.; lo plantea aquí, y el resto participa.
- Un día de la semana, podríamos fijar un **encuentro por mensajería instantánea** para hacer un balance de lo que se estudió y acordar cómo seguir la próxima semana.

¿Qué les parece?

Alejandro

---



## Materiales de estudio

(Si tienen alguno más para recomendar, lo agregan)

[Apuntes\\_bash\\_0.pdf / Apuntes\\_bash\\_1.pdf / bash.pdf Comandos.pdf](#)

[Carpeta con documentos vacíos para ejercitar comodines](#)

[Ejemplos de scripts en Ubuntu Forum](#)

[Un comando cada día - Ubuntu Forum](#)

[Bash scripting de supervivencia](#)

[Taller de programación shell](#)

[Shell Scripting](#)

[Google Docs](#)



### 2.1.- "Los comodines" y 2.2.- "El comodín tilde"

Comodín	Descripción
?	Uno y sólo un carácter
*	Cero o más caracteres
[conjunto]	Uno, entre los caracteres del conjunto
[!conjunto]	Un carácter que no esté dentro del conjunto
~	Se usa para referirse al directorio home de los usuarios

---

Quizás puede servir. En esta dirección encontrarán un curso muy bueno sobre [Google Docs](#)

Jorge

---

Armé una carpeta de documentos vacíos con nombres generados con combinaciones de letras, números, palabras, para hacer ejercicios con comodines. Coloco el link en la parte de materiales de estudio.

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

Sugerencia:

- Descomprimir la carpeta en Home.
- Abrir terminal.
- Hacer "cd" hasta la carpeta con los archivos:

```
cd ~/combinado
```

- Ejercitar con el comando "ls" (que es el más inofensivo, pero se podrían usar "cp"; "mv"; "rm"; etc.) distintas posibilidades de listado de esos archivos.

Ejemplos:

1. Todos los que tienen una extensión determinada.
2. Todos los que comienzan con "a" y tienen "d" en la tercera posición, pero no terminan con "t".
3. Todos los ".avi" que contienen la palabra "lejos"; etc.

¿Qué listados más o menos rebuscados se les ocurren? A ver quién presenta un desafío. Como todo tiene varias vías de solución, estaría bueno descubrir la mayor cantidad posible de variantes, para evaluar cuales son las más apropiadas o más económicas.

Alejandro

---

voy dejando las soluciones que encontré a lo que planteó (creo que Alejandro) aquí arriba.

(claro que estos comandos son ejecutados una vez que me posicioné en la carpeta que los contiene)

```
1) ls *.avi
```

```
2) ls a?d*[^t]
```

```
3) ls *lejos*.avi
```

y como 'desafío' propongo este listado: mostrar todos los archivos que contengan la palabra "casa" y que sean .pdf ó .avi (se podrá hacer con lo que vimos hasta ahora? o ya entra en juego algo de programación?)

Saludos!

Fabricio

---

Voy a dejar la solución de el problema de **Fabricio**

```
1. ls *casa*.pdf *casa*.avi
```

No se me ocurrió algo más simple, pero debe haber una solución mejor

## Diego

---

Se me ocurren estas posibilidades:

```
ls *casa*.[pa]*  
ls *casa*.[pa]??  
ls *casa*.[!x]?  
ls *casa*.[!t]??  
etc.
```

que funcionan en ese caso, pero no funcionarían en cualquier situación.

```
ls *casa*.[pa][dv][fi]
```

que es menos económica pero funcionaría en cualquier situación. Bah, tampoco, porque si la extensión fuese pdi, pvf, adi, etc., también las listaría. Así que, hasta ahora, la propuesta de **Diego** es la más segura.

Y ya que lo estamos usando

## El comando "ls"

```
ls                Lista los ficheros de un directorio concreto.  
ls -R            Lista recursivamente.  
ls -C            Lista los ficheros en columnas.  
ls -l            Lista también las propiedades y atributos.  
ls -a            Lista ficheros incluidos los ocultos de  
sistema.  
ls -la | more    Lista los ficheros de forma paginada.
```

## Alejandro

---

¡Que bueno estos ejercicios! He aprendido más que en muchas horas de estudio solo, esto de estudiar en grupo es más motivador y te va aclarando dudas. Mi solución a lo planteado por **Fabrizio**

```
ls *casa*.pdf .avi
```

Y dos dudas 1.- ¿Cómo consigo que mis comandos queden como los de todos ustedes? y 2.- Podría alguno explicarme en sencillo lo que significa "recursivamente" Gracias.

## Jorge

---

No me funcionó:

**Jorge**, no me funcionó tu comando ¿Cómo lo pensaste?

Alejandro

---

Consejos:

- Menú "Ver"-->"vista normal (sin formato)". Al menos para mí, que trabajo en una portátil y tiene la pantalla pequeña, resulta más práctico porque así puedo visualizar mejor el documento.
- Ahí descubrí una mejor forma de delimitar cada intervención o conversación: "Insertar"-->"comentario"; dividirlo, como hice aquí, en un título y la firma abajo (cortar y pegar); y después de escribir el cuerpo del mensaje, seleccionar todo y "Formato"-->"Sangrar párrafo".
- No será Google Wave, pero...

Alejandro

---

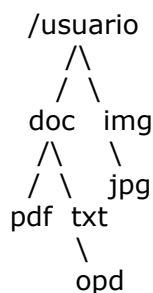
Con respecto a la pregunta de recursivo, entiendo que:

- `ls -R` muestra las carpetas y archivos contenidos en este directorio.
- lo mismo con `rm -R` (cuidado) borra recursivamente.

Diego

---

Gráficamente:



`ls /usuario *.*` listaría todo lo que hay en ese directorio: "/usuario".

`ls -R /usuario *.*` listaría todo lo que hay en ese directorio y en todos los directorios que hay debajo de él.

Alejandro

---

Ejercicios

Pego estos ejercicios para el comando ls y al final la solución que yo encontré:

1. Muestra todos los archivos del directorio actual que son imágenes jpg.
2. Muestra todos los archivos del directorio /usr/bin que empiecen por la letra j.
3. Muestra todos los archivos del directorio del usuario actual, incluyendo los archivos ocultos, con toda la información de cada uno (permisos, tamaño, etc.)
4. Muestra los archivos que empiecen por k y tengan una a en la cuarta posición, dentro del directorio /usr/bin.
5. Muestra los archivos del directorio /bin que terminen en n.
6. Muestra todos los archivos que hay en /tmp y todos los que hay dentro de cada subdirectorio, de forma recursiva.

```
1.    ls *.jpg (posicionada en mi /home)
2.    ls -l /usr/bin/j*
3.    ls -al      (en mi /home)
4.    ls -l /usr/bin/k???a*
5.    ls -l /bin/*[n]
6.    ls -R /tmp
```

Maria

---

### Desafío N° 1: ¿Cómo buscar recursivamente todos los archivos de backup que van quedando por ahí, para poder eliminarlos y limpiar las carpetas?



Los ejercicios de María, me sugirieron una idea de aplicación que puede ser útil: **¿Cómo buscar recursivamente todos los archivos de backup que van quedando por ahí, para poder eliminarlos y limpiar las carpetas?** Se las dejo como desafío. Si lo quieren probar, háganlo únicamente con "ls", no borren hasta que no estén bien seguros, por las dudas.

Alejandro

---

Gracias por las respuestas. Voy a ver si me funciona el formato.

**Alejandro:** Verifiqué mi comando y me funciona sin problemas: tienes que dejar un espacio entre .pdf y .avi

Jorval

---

Y si quiero buscar recursivamente pero con una condición, por ejemplo, que los archivos terminen en .jpg ?? mi intención es buscar los archivos .jpg que están dentro de /home/usuario de manera recursiva

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

```
ls -R ~/ *.jpg*    (nóten el espacio entre / y *) me muestra el mismo
resultado que ls -R ~/*
ls -R ~/* .jpg*    (sin espacio) muestra los .jpg de mi carpeta /home/
usuario
```

### Fabricio

---

Creo que lo mejor es ir delimitando las intervenciones con esta línea horizontal porque esto de los "comentarios" tiene sus bemoles si alguno quisiera imprimir, etc. esta página. Les copio lo que dice la ayuda de Docs al respecto: "Cuando publiques el documento como página web, lo publiques en tu blog o lo imprimas, los comentarios desaparecerán".

### Jorval

---

#### Al desafío de **Alejandro**:

Me parece que `ls -R` es un comando a nivel directorios, no de archivos. Si vos pones `*.jpg` te lo toma como directorio y no lo es.  
Si pones

```
$ ls -R Imágenes
```

Ahi si funciona

Pero no estoy segura  
(quise delimitar con las líneas pero no me salio)

### Maria

---

Para colocar las barras: "Insertar"-->"Línea horizontal"

Colocando:

```
$ ls -R Imágenes
```

Lista el contenido del directorio Imágenes y subdirectorios. Pero si quiero ver únicamente los archivos con extensión `.bmp`, ahí ya no funciona, sigue listando todo. Creo que tenés razón, `ls -R` funciona a nivel directorios. Para filtrar la búsqueda hay que usar otras herramientas que todavía no vimos, pero que en breve vamos a ver.

### Alejandro

---

```
jorval@jorval-laptop:~$ cd /tmp
jorval@jorval-laptop:/tmp$ ls -R
```

y aparecen todos los archivos del directorio

Jorval

---

Al desafío, claro solo basta como dice Jorval:

Con `ls -R`

de esa forma muestra todos los directorios, cada uno con sus archivos.

Rodrigo

---

**Jorval** y **Rodrigo**, podría ser así, también:

```
ls -R /tmp          (con un solo comando)
```

**Fabricio:**

1. Una duda, ¿cuál es el objetivo de colocar el comodín "\*" después de la extensión ".jpg"?
2. Probé las dos opciones en mi /home, con espacio y sin espacio.

Con espacio:

```
ls -R / *.jpg
```

Lista todo el contenido, como si fuera efectivamente:

```
ls -R ~/*
```

Sin espacio:

```
ls -R /*.jpg
```

Me da error:

```
ls: no se puede acceder a /home/alejandro/*.jpg: No existe el
fichero ó directorio
```

Es decir, en el primer caso, lista recursivamente, pero ignora el filtro que seleccionaría todos los jpg; y en el segundo caso, ignora la orden de buscar recursivamente, y busca únicamente en el directorio ~/ , donde no tengo ningún jpg, ya que están en un subdirectorio ~/Imágenes.

**¿Por qué será? ¿A alguien se le ocurre?**

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

Me da la impresión de que "ls" con la opción "-R" no acepta comodines, que lista todo lo que está debajo del directorio indicado, sin importar de qué tipo de archivo se trate. ¿Será así?

Alejandro

---

el comodín después de \*.jpg se me olvidó sacarlo porque en un principio estaba probando de buscar archivos que contengan la palabra CASA, y cuando me puse a probar con .jpg se me olvidó sacarlo. Igualmente ya lo edité, gracias por la corrección...

Fabricio

---

**Alejandro:**

Me parece que `ls -R` es un comando a nivel directorios, no de archivos. Si vos pones \*.jpg te lo toma como directorio y no lo es. Si pones

```
$ ls -R Imágenes
```

Ahi si funciona

Pero no estoy segura  
(quise delimitar con las lineas pero no me salio)

Maria

---

**Maria:**

arriba dice insertar -> linea  
desde ahí se puede poner la linea para delimitar ;) (Insert > Horizontal line es muy "difizil", no entiendo... XDDD .... no lo había visto, gracias!)

Quizas de esta forma podría realizarse una búsqueda de todos los archivos de un tipo en específico:

```
$ ls */*.jpg && ls */*.JPGP && ls */*.jpeg && ls  
*/*.JPEG
```

de esa forma buscaria dentro de todos los directorios, y entregaria la ubicacion de la forma si buscara en mi home ejemplo:

```
usuario/imagen.jpg
```

y bueno especifique tantos jpg, jpeg, etc. ya que suele pasar que los ponen con mayusculas, minusculas, con la 'e', sin la 'e', etc. Los archivos de un cierto tipo en caso de no existir ninguno dira que no existe el fichero o directorio.

Rodrigo

---

**[Quote=Rodrigo]**

Quizas de esta forma podría realizarse una búsqueda de todos los archivos de un tipo en específico:

```
$ ls */*.jpg && ls */*.JPGP && ls */*.jpeg &&  
ls */*.JPEG
```

**Rodrigo:** me parece, pero no puedo garantizar que sea realmente así, que el comodín no vale para reemplazar directorios, sólo nombres de archivos. **Tendríamos que investigar eso.** Hice la prueba, colocando:

```
ls */*.jpg
```

y me tira las imágenes que tengo en el directorio **siguiente** al actual (no lo que hay en el actual), y no sigue bajando en la jerarquía de directorios. Eso es extrañísimo ¿no?

Está bueno el dato de **"&&"**. También se puede usar  **";"**. Ej. --> comando1 ; comando2 ; comando3

**Respecto al desafío:**

Yo no me supe expresar bien. Vieron que cuando uno abre una aplicación, por ejemplo un documento en OpenOffice, mientras se está editando el documento, en el directorio aparece un archivo igual, pero terminado en "~". En realidad, no es un archivo temporal, es un archivo de backup (**por lo tanto no habría que borrarlos**) que se genera para que la aplicación pueda recuperar el documento. La idea era buscar en todo el equipo y listar esos archivos. Pero creo que con lo que sabemos hasta ahora no sería posible ¿o me equivoco?

**Recapitulando:**

Necesitamos saber:

- Si el comando "ls" con la opción "-R" (recursivo) acepta como parámetro un nombre de archivo, con o sin comodines. Ej. --> `ls -R *.jpg` (todo parece indicar que no, que lo que sigue a la opción "-R" va a ser interpretado como el nombre de un directorio; por eso devuelve error diciendo que no existe el directorio)
- Si es posible usar comodines para reemplazar -una parte o todo- el nombre de un directorio. Ej. --> `ls */*.jpg`

Alejandro

---

Con respecto a la pregunta de **Alejandro**, en mi caso los comandos

- `ls -R *.jpg` devuelve error y es probable que lo tome como nombre de directorio incorrecto.
- `ls */*.jpg` si funciona por lo que es posible usar este comodín, que por lo cierto es muy práctico.

Diego

---



### Script N°1 (Alejandro)

Se me ocurrió hacer un pequeño script aplicando algunas cosas que vimos. En realidad no tiene otra utilidad que practicar la escritura de script. Ahí va:

```
#!/bin/bash
##Crear directorios
mkdir ~/combinado/casa
mkdir ~/combinado/mesa
mkdir ~/combinado/patio
mkdir ~/combinado/cama
##Mover los archivos a sus directorios correspondientes
mv *casa*.* ~/combinado/casa
mv *mesa*.* ~/combinado/mesa
mv *patio*.* ~/combinado/patio
mv *cama*.* ~/combinado/cama
##Listar como quedaron los directorios reordenados
ls -R ~/combinado
```

Primero se crean 4 directorios en la carpeta "combinado" que tiene que estar en Home. Después se mueven todos los archivos que contienen la palabra X a su directorio X. Y por último se listan recursivamente todos archivos de la carpeta "combinado" que fue reordenada.

Si alguien lo quiere probar:

- Abrir gedit (editor de textos), copiar, pegar y guardar con un nombre como "script.sh" (lo importante es la extensión .sh)
- En la terminal hay que otorgarle permiso de ejecución, es decir:

```
chmod +x script.sh
```

- Y se puede ejecutar haciendo doble click, o en terminal, dos opciones:

```
./script.sh

bash script.sh
```

Siempre, claro, estando ubicados en el directorio correcto.

Y si no lo quieren probar, no se pierden gran cosa. El script en sí mismo no tiene sentido,

se pueden tipear los comandos en la consola en ese orden y el resultado es el mismo. Pero, supongamos que tenemos muchas carpetas o pc's que tenemos que organizar de la misma forma, con un script no tenemos que estar tipeando cada vez.



### Desafío N° 2 ¿Cómo reorganizar el script 1 por colores?

**Lo que podrían hacer, si tienen ganas es, a partir de ese resultado, reorganizar de otro modo, por ejemplo, por colores. Pero partiendo de la nueva organización, de como quedo a partir de este script. Quien se anima? Minuto Odol en el aire!!!**

Alejandro

---

**Alejandro:** ¿por qué empleas la ruta absoluta y no la relativa para el script? ¿no se puede o no se debe? Me imagino que en este caso uno está en su directorio /home

Jorval

---

¿La ruta absoluta no sería, en mi caso, /home/alejandro/combinado/casa, por ejemplo?  
¿Cómo lo harías **Jorval**?

Alejandro

---

Sí, claro esa es tu ruta absoluta, pero también podrías emplear la ruta relativa ya que estás situado en tu /home/alejandro

```
~$ mkdir combinado/casa
```

Esa es mi duda ¿se pueden emplear ambas o sólo la absoluta?

Jorval

---



### Script N° 2 (Fabricio)

A mí el script de **Alejandro** me funciona solamente de la siguiente manera:

```
#!/bin/bash
##Crear directorios
mkdir ~/combinado/casa
mkdir ~/combinado/mesa
mkdir ~/combinado/patio
mkdir ~/combinado/cama
```

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

```
#Mover los archivos a sus directorios correspondientes
mv ~/combinado/*casa*.* ~/combinado/casa
mv ~/combinado/*mesa*.* ~/combinado/mesa
mv ~/combinado/*patio*.* ~/combinado/patio
mv ~/combinado/*cama*.* ~/combinado/cama

## Listar como quedaron los directorios reordenados
ls -R ~/combinado
```

En **negrita** está lo que modifiqué, y con respecto a si es mejor usar rutas absolutas o relativas, para mí lo mejor son las absolutas porque estoy seguro que van a funcionar. Por ejemplo a Alejandro le funcionó su script pero a mí ya no, tuve que agregar ~/combinado/ para que funcionara

### Fabricio

---



Muy buenas observaciones, no lo había tenido en cuenta.

### Script N° 3 (Alejandro)

```
#!/bin/bash
##Ubicarse en el directorio de trabajo
cd ~/combinado
##Crear directorios
mkdir casa
mkdir mesa
mkdir patio
mkdir cama
##Mover los archivos a sus directorios correspondientes
mv *casa*.* casa
mv *mesa*.* mesa
mv *patio*.* patio
mv *cama*.* cama
##Listar cómo quedaron los directorios reordenados
ls -R
```

**¿Qué opinan ahora?** Funcionaría para cualquier caso, siempre que la carpeta "combinado" esté en la Home del usuario. En todo caso, lo único que cada uno tendría que modificar es el "cd" del principio, colocando la ruta absoluta o relativa al directorio donde tiene la carpeta "combinado".

### Alejandro



PD: Y... ¿nadie se le animó al desafío?

**"...a partir de ese resultado, reorganizar de otro modo, por ejemplo, por colores. Pero partiendo de la nueva organización, de como quedó a partir de este script. Quién se anima? Minuto Odol en el aire!!!"**

---

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

Dame unos minutos que veo que sale :D

Fabricio

---

¡Vamos, alguien que se anime!!! El primero/a, se gana una remera de Bill Gates!!!

Muy bueno lo del "Bash scripting de supervivencia" en materiales de estudio!



Alejandro

---

Ah si... lo puse yo, fijate que habla un poco más sobre estructuras de control. La lógica ya la tengo hecha para hacer el desafío, lo que me falta es agarrarle la mano a la sintáxis de Bash !

Fabricio

---

Ahí agregué dos links más de bibliografía en "Materiales de Estudio". Estaban en el mismo documento que agregé **Fabricio** y parecen interesantes.

Ah, y aclaro. Espero no llegar tarde. No vale hacer trampa, eh! El desafío es a resolver con las herramientas que vimos hasta ahora!!!!



Alejandro

---

Me funcionó el script de **Alejandro**, pero tengo una observación: El script se detiene en

```
ls -R
```

y para que funcione esta última parte tienes que darle enter, debe haber alguna forma de que ese enter sea automático en el script, pero ¿para qué nos adelantamos? sigamos el libro, de a poco.

Jorval

---

No sabría qué decirte, **Jorval**. A mí me funciona, es decir, ordena los archivos en las carpetas correspondientes y finalmente lista todo recursivamente.

Alejandro

---



## Script Nº 4 (Fabricio)

```
#!/bin/bash

# Creo los directorios
mkdir ~/combinado/verde
mkdir ~/combinado/azul
mkdir ~/combinado/rojo
mkdir ~/combinado/gris

# Recorro las 4 carpetas (casa,cama,mesa,patio) buscando
archivos que contengan la palabra verde para luego moverlos a su
directorio correspondiente (creado en el paso anterior)
mv ~/combinado/casa/*verde* ~/combinado/verde
mv ~/combinado/cama/*verde* ~/combinado/verde
mv ~/combinado/mesa/*verde* ~/combinado/verde
mv ~/combinado/patio/*verde* ~/combinado/verde

# Idem anterior
mv ~/combinado/casa/*azul* ~/combinado/azul
mv ~/combinado/cama/*azul* ~/combinado/azul
mv ~/combinado/mesa/*azul* ~/combinado/azul
mv ~/combinado/patio/*azul* ~/combinado/azul

# Idem anterior
mv ~/combinado/casa/*roja* ~/combinado/rojo
mv ~/combinado/cama/*roja* ~/combinado/rojo
mv ~/combinado/mesa/*roja* ~/combinado/rojo
mv ~/combinado/patio/*roja* ~/combinado/rojo

# Idem anterior
mv ~/combinado/casa/*gris* ~/combinado/gris
mv ~/combinado/cama/*gris* ~/combinado/gris
mv ~/combinado/mesa/*gris* ~/combinado/gris
mv ~/combinado/patio/*gris* ~/combinado/gris

# Como moví en definitiva todos los archivos que estaban dentro
de CASA, CAMA, MESA y PATIO estos directorios quedaron vacíos,
¿y para que tener directorios vacíos? mejor los borramos :D
rmdir ~/combinado/casa
rmdir ~/combinado/cama
rmdir ~/combinado/mesa
rmdir ~/combinado/patio
```

Pensaba hacer esto mismo pero dentro de algunos "FOR" para no repetir tanta veces sentencias parecidas... pero si hay que hacerlo con lo que aprendimos hasta ahora supongo que sería algo así.

Saludos!

Fabricio

---

Está bueno, **Fabricio!** Te ganaste la remera del tío Bill!!! Te la debo...

Lanzo el siguiente desafío, pero para **Jorge, María, Diego, Rodrigo y Sergio**, que no han publicado script todavía:



### **Desafío N° 3 ¿Cómo simplificarían el script de Fabricio?**

Pero sin usar bucles, rutinas y ese tipo de cosas, vamos gradualmente. El primero/a, se gana una "Ventana con Vista al mar".

Y claro que son bienvenidos los desafíos que cualquiera de ustedes quiera proponer ;)

Alejandro

---

¿Conocen **Xournal**? Es una aplicación que permite abrir un pdf y subrayar como si fuese un resaltador y hacer anotaciones al margen. No se puede editar el original. La estoy usando para remarcar lo que me parece más importante en el Libro de Bash, y hacer anotaciones sobre lo que me genera dudas, no entiendo o cosas que tendría que probar, etc., para después volcar acá en este espacio. Está en repositorios.



Alejandro

---



## **2.3.- Comodín llaves**



**Desafío N° 4** ¿Cómo generarían esta lista de palabras con un sólo comando en la terminal?

```
buscarlo buscarla buscarle buscarlos buscarlas buscarles buscarme  
buscarsela buscarselo buscarselas buscarselos buscarte buscaría buscarías  
buscarían buscarán buscarás buscaron buscaremos burlarlo burlarla burlarle  
burlarlos burlarlas burlarles burlarme burlarsela burlarselo burlarselas  
burlarselos burlarte burlaría burlarías burlarían burlarán burlarás  
burlaron burlaremos bufarlo bufarla bufarle bufarlos bufarlas bufarles
```

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

bufarme bufarsela bufarselo bufarselas bufarselos bufarte bufaría bufarias  
bufarían bufarán bufarás bufaron bufaremos bucearlo bucearla bucearle  
bucearlos bucearlas bucearles bucearme bucearsela bucearselo bucearselas  
bucearselos bucearte bucearía bucearias bucearian bucearán bucearás  
bucearon bucearemos butearlo butearla butearle butearlos butearlas  
butearles butearme butearsela butearselo butearselas butearselos butearte  
butearía butearías butearían butearán butearás butearon butearemos



Alejandro

---



Este es mi primer Script, me funcionó haciendo doble clic en el archivo, no así en la terminal, pero estoy contento, algo he avanzado

### Script Nº 5 (Jorge)

```
#!/bin/bash
##Me ubico en directorio de trabajo
cd combinado
## Creo los directorios por colores
mkdir verde azul rojo gris
##Recorro los directorios casa, cama, mesa y patio que tengan la
palabra verde, azul, rojo y gris y las envío a los directorios
verde, azul, rojo y gris
mv casa/*verde* ./verde
mv cama/*verde* ./verde
mv mesa/*verde* ./verde
mv patio/*verde* ./verde
mv casa/*azul* ./azul
mv cama/*azul* ./azul
mv mesa/*azul* ./azul
mv patio/*azul* ./azul
mv casa/*rojo* ./rojo
mv cama/*rojo* ./rojo
mv mesa/*rojo* ./rojo
mv patio/*rojo* ./rojo
mv casa/*gris* ./gris
mv cama/*gris* ./gris
mv mesa/*gris* ./gris
mv patio/*gris* ./gris
```

Jorval

---

Para el desafío Numero 4, no tiene mucha ciencia ya que el comando ls, ordena las palabras en forma ABC... de menor a mayor si se entiende

buscar, buscarla, buscarlo....

al aplicar

```
$ ls
```

devuelve

```
buscar
```

```
buscarla
```

```
buscarlo
```

Rodrigo

---

No, Rodrigo, el desafío es cómo "GENERARLA", cómo hacer que con el comando "echo", por ejemplo, y usando comodines se genere esa lista.

Por ejemplo:

```
$ echo ca{s,m}a  
casa cama
```

¿Se entiende?



Alejandro

---

aa ok entendido... xD

Para como simplificar el script de Fabricio a mi me quedo lo siguiente:



**Script N° 6 (Rodrigo)**

```
#!/bin/bash  
  
# Creo los directorios  
mkdir ~/combinado/verde  
mkdir ~/combinado/rojo  
mkdir ~/combinado/azul  
mkdir ~/combinado/gris
```

```
# Se busca en cada directorio dentro de combinado y se ordena a
las carpetas correspondientes
mv ~/combinado/*/*verde* ~/combinado/verde
mv ~/combinado/*/*rojo* ~/combinado/rojo
mv ~/combinado/*/*gris* ~/combinado/gris
mv ~/combinado/*/*azul* ~/combinado/azul

# No borro los directorios porque no se si existen mas datos xD
```

ahora a ver que puedo hacer con lo de generar la lista.

Rodrigo

---



Propongo otorgar el:

**Premio "Rex Tux Scripting" del desafío número 3**

para **Fabricio**. Si bien **Jorval** ganó la "Ventana con Vista al mar" por ser el primero.

Realmente muy buena solución. Yo había pensado algo parecido a lo de **Jorge**.

- Lo que yo le agregaría es: `cd ~/combinado` al principio, para no repetirlo después, y ahí quedaría reducido a su mínima expresión, creo.
- La creación y el borrado de directorios se puede hacer en una sola línea.
- Hay un sólo error, es "roja" y no "rojo". Ah, y no quedan más datos, así que se pueden borrar los directorios obsoletos tranquilamente.
- Y otra cosa importante, se confirma que los comodines pueden ser usados para reemplazar nombres de directorios.

Podría quedar así:



**Script N° 7 (Alejandro)**

```
#!/bin/bash

# Me ubico en el directorio
cd ~/combinado

# Creo los directorios
mkdir verde gris roja azul

# Se busca en cada directorio y se ordena en las carpetas
correspondientes
mv /*verde* verde
mv /*roja* roja
mv /*gris* gris
mv /*azul* azul
```

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

```
# Borro los directorios
rmdir cama casa patio mesa

# This is the end, my friend
```

### Alejandro

---

A la carpeta le puse de nombre **rojo** porque la clasificación era por colores... y si bien la mesa es de color **roja** su color es **rojo**... nosé si me entienden, es decir, una carpeta puede ser **amarilla** pero en realidad su color se llama **amarillo** jajaja, en fin, una boludes!!! xD

P.D.: Muy buenas las imágenes que pusieron para diferenciar scripts, desafíos, y demás!



### Fabricio

---

Es cierto **Fabricio**, la carpeta tendría que llevar como nombre "rojo", por más que su contenido diga "roja", pero, cuando muevo los archivos, ahí sí debe decir "roja", porque sino no los encuentra.

Le fui colocando imágenes y ordenando un poco el formato, acomodando el índice, etc., como para ponerle un poco de onda al documento. Ustedes no se enojen si por ahí modifico algo, no es contenido es sólo formato. Para reutilizar una imagen, es decir, si alguien agrega un script, presentar un desafío, etc., es sólo copiar y pegar una de las imágenes que ya fueron usadas. Colocando el "Estilo" adecuado, el índice se actualiza automáticamente. Para desafíos usé "Título 3" y para scripts "Título 4".

### Alejandro

---



#### Script N°8 (Fabricio)

```
#!/bin/bash
echo
bu{sc{arlo, arla, arle, arlos, arlas, arles, arme, arsela, arselo, arselas,
arselos, arte, aría, arías, arían, arán, arás, aron, aremos}, rl{arlo, arla,
arle, arlos, arlas, arles, arme, arsela, arselo, arselas, arselos, arte, aría,
arías, arían, arán, arás, aron, aremos}, fa{arlo, arla, arle, arlos, arlas,
arles, arme, arsela, arselo, arselas, arselos, arte, aría, arías, arían, arán,
arás, aron, aremos}, ce{arlo, arla, arle, arlos, arlas, arles, arme, arsela,
arselo, arselas, arselos, arte, aría, arías, arían, arán, arás, aron, aremos},
te{arlo, arla, arle, arlos, arlas, arles, arme, arsela, arselo, arselas,
arselos, arte, aría, arías, arían, arán, arás, aron, aremos}}
```

Nota: para que quede un poco más lindo a la vista agregué unos "Enter" después de cada coma, si van a probar el script eliminen esos enters de más y dejen que sea todo

seguido... sino no va a andar.

Si encuentran otro script más corto que haga lo mismo publiquenlo

Fabricio

---

Experiencia para el futuro: **Revisar todo antes de publicar!** Ayer cuando hice el Script revisé las dos primeras carpetas, la azul y la verde y me dije: **Funcionó!** y listo, pero hoy, antes de entrar a Docs, me puse a revisar y vi lo de roja con rojo e ingresé a Docs para informarlo y con alegría comprobé que ya había sido descubierto y que a todos nos había sucedido lo mismo.

Excelente el Script definitivo que hiciste Alejandro, iba a proponerte hacer algo similar para el documento final.

Jorval

---

Desafío N° 4

```
jorval@jorval-laptop:~$ echo
buscar,burlar,bufar,bucear,butear{lo,la,le,los,las,me,sela,selo,selos,
te,ía,ían,arán,arás,on,emos}
```

No pude ponerlo bien porque después de echo hay un espacio, pero funciona. No pongo el resultado para que no salga el texto del papel

Jorval

---

No me funciona tu script, **Jorval**... fijate que no da como resultado la secuencia de palabras que se pide

Fabricio

---

**Fabricio**, tienes razón, el comando repite las cuatro primeras palabras y se desarrolla sólo con la que está pegada al paréntesis. Mañana veré si lo arreglo, pero por ahí va la cosa

Jorval

---

No, mejor lo arreglo al tiro, era más fácil de lo que pensaba:

```
jorval@jorval-laptop:~$ echo
{buscar,burlar,bufar,bucear,butear}{lo,la,le,los,las,me,sela,selo,selos,te,ía,ían,
ararán,arás,on,emos}

buscarlo buscarla buscarle buscarlos buscarlas buscarme
buscarsela buscarselo buscarselos buscarte buscaría buscarían
buscararán buscararás buscaron buscaremos burlarlo burlarla
burlarle burlarlos burlarlas burlarme burlarsela burlarselo
```

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

```
burlarselos burlarte burlaría burlarían burlarararán burlararás
burlaron burlaremos bufarlo bufarla bufarle bufarlos bufarlas
bufarme bufarsela bufarselo bufarselos bufarte bufaría bufarían
bufarararán bufararás bufaron bufaremos bucearlo bucearla bucearle
bucearlos bucearlas bucearme bucearsela bucearselo bucearselos
bucearte bucearía bucearían bucearararán buceararás bucearon
bucearemos butearlo butearla butearle butearlos butearlas
butearme butearsela butearselo butearselos butearte butearía
butearían butearararán buteararás butearon butearemos
jorval@jorval-laptop:~$
```

Jorval

---

Buenísimo! y mucho más corto que el mio !

Fabricio

---



Muy bueno!!!! Propongo para **Jorge** el:

**Premio "Rex Tux scripting" del desafío 4.**

El comando con el que generé la lista era el siguiente:

```
echo
bu{sc,rl,f,ce,te}ar{l{o,a,e},l{o,a,e}s,me,se{la,lo},se{la,lo}s,te,í
{a,as,an},á{n,s},on,emos}
```

Pero el de Jorge es mucho más elegante.

Quizá, este fragmento podría utilizarse en el de **Jorge**, para abreviarlo un poco más (pero no sé si se justifica, queda lindo como está):

```
bu{sc,rl,f,ce,te}ar
```

Ah, y un detalle: colocar "án" y "ás" en lugar de "arán" y "arás", para que no dé "buscararás" y "buscararán".

Y ésta sería la idea de usar los comodines llaves anidados, llevada a sus últimas consecuencias. Como ejercicio de sintáxis, es interesante:

```
echo
bu{sc,rl,f,{c,t}e}ar{l{o,a,e},{o,a,e}s},{m,t}e,sel{a,o,{a,o}s},ía{s,n},á{n,s},on,em
```



Me gustaría encontrar una solución para evitar esta repetición del singular y el plural: {a, o, {a, o}s}, pero no se me ocurre nada.

**¿Alguien tiene una propuesta?**

Alejandro

---

¿Se nota que me puse rojo de vergüenza? Muy agradecido por el trofeo, que acepto humildemente.

Veo que avanzamos al punto "2.3 El comodín llaves" aunque **Alejandro** nada ha dicho. De ese comodín rescato dos puntos: que como en el caso del desafío último entrega el producto cartesiano de las llaves y lo segundo que funciona con dos cadenas a lo menos, con solo una no se efectúa la expansión.

Jorval

---

Hola a todos me fuí de vacaciones así que estoy muy atrasado. Este es mi pequeño monstruo **ATENCIÓN NO USAR** no se que daño puede hacer. En mi pc funciona.



**Script N°9 (Diego)**

```
#!/bin/bash
# Antiviral
# Detector de virus para pendrive
# Datos del ingenuo del autor diego

# Creo el directorio

mkdir virus

# Se busca en cada directorio y se ordena en la carpetas
correspondientes

mv */*.exe virus
mv */*.drv virus
mv */*.dll virus
mv */*.bin virus
mv */*.ovl virus

#Los virus son generalmente de estos tipos
#*.EXE, *.DRV, *.DLL, *.BIN, *.OVL, *.SYS e incluso BAT

#cortito como patada é chancho
```

Como lo mejoramos?

Diego

---



## 2.4.- Los comodines extendidos

---

**Diego:** se me ocurre esta línea para el movimiento de ficheros a los directorios:

```
mv */*.*@(exe|drv|dll|bin|ovl|bat|sys) virus
```

¿Qué les parece?

Además, incorporando al principio un:

```
cd /
```

Se garantiza que la búsqueda se realice en todo el equipo y no únicamente a partir del directorio actual.

No lo quise probar porque, como tengo Wine instalado en el equipo con algunas aplicaciones de W\$, se me puede armar un "bolonqui" bárbaro (por ahí, para probarlo se podría usar "cp" en lugar de "mv"). Pero la idea fue usar los **comodines extendidos**.



Respecto a eso, cuando leía, me surgió una **duda**. En la página 14 dice:

```
"Si lo que hubiéramos querido es encontrar todos los ficheros excepto los .gif, los .jpg y los .html podríamos haber usado el patrón !(*.gif|*.jpg|*.html). Sin embargo, en este caso no podríamos haber usado *.*!(html|gif|jpg)"
```

Pero no explica por qué, o al menos yo no encuentro el motivo. **¿Por qué ese patrón funcionaría para "@" y no para "!"?**

...



Por último, mañana termina la semana y el jueves comienza la siguiente. Creo que es importante **respetar los plazos**.

**Tenemos pendiente un encuentro virtual** por la mensajería de Gmail para definir las cuestiones de organización de aquí en más: ¿Google Docs, Wiki, Foro, etc.? ¿Cómo y cuándo hacer público el proceso o los resultados?, evaluar qué

tal nos fue hasta ahora, etc. No va a ser sencillo coordinar los horarios, creo. Cada uno de nosotros debería enviar un mensaje a todos los participantes ("Compartir"-->"Enviar mensaje a los colaboradores") con el día y horario disponible, así vemos cuál es la fecha y el horario en que más gente se puede reunir.

Propongo que hoy y mañana nos concentremos en los **comodines extendidos**, para sacarles todo el jugo posible:

```
+(pattern-list)
@(pattern-list)
!(pattern-list)
?(pattern-list)
*(pattern-list)
```

El jueves ya estaríamos en los temas **4 "Redirecciones y Pipes"** y **5 "Ejecución secuencial y concurrente de comandos"**.

Alejandro

---

### Quote alejandro:



Respecto a eso, cuando leía, me surgió una **duda**. En la página 14 dice:

*"Si lo que hubiéramos querido es encontrar todos los ficheros excepto los .gif, los .jpg y los .html podríamos haber usado el patrón `!(*.gif|*.jpg|*.html)`. Sin embargo, en este caso no podríamos haber usado `*.!(html|gif|jpg)`"*

*Pero no explica por qué, o al menos yo no encuentro el motivo. ¿Por qué ese patrón funcionaría para "@" y no para "!"?*

Supongo que en esa parte hay algún error porque esos dos comandos son iguales (o por lo menos me dan los mismos resultados). Quizás el problema tenga que ver con la versión de Bash, fijate que al principio del libro el autor dice lo siguiente:

```
" Si ahora nos logamos de nuevo con el comando login y preguntamos por la
versión de Bash:
$ echo $BASH
/sw/bin/bash
$ echo $BASH_VERSION
3.00.0(1)-release "
```

y si ahora colocamos eso mismo en la terminal sale lo siguiente (al menos en mi caso):  
**3.2.48(1)-release**

Es decir, quizás que el comentario del autor es válido para versiones antiguas de Bash...

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

Fabricio

---

Miren, prueben esto (en la carpeta "combinado" que fue reorganizada por colores):

```
cd ~/combinado

ls *.*(x|y)

ls !(*.x|*.y)
```

¿Les da el mismo resultado en los dos casos?

Alejandro

---

si, el mismo

Fabricio

---

A mí me da totalmente distinto, con el primero lista:(copio las dos primera filas)

```
a1d4.tt  a2d5.tt  a3e4.tt  b1e5.tt  b2f4.tt  b3f5.tt  c2d4.tt
c3d5.tt
a1d4.tx  a2d5.tx  a3e4.tx  b1e5.tx  b2f4.tx  b3f5.tx  c2d4.tx
c3d5.tx
a1d4.xt  a2d5.xt  a3e4.xt  b1e5.xt  b2f4.xt  b3f5.xt  c2d4.xt
c3d5.xt
a1d5.tt  a2e4.tt  a3e5.tt  b1f4.tt  b2f5.tt  c1d4.tt  c2d5.tt
c3e4.tt
```

y con el segundo me lista una infinidad de cosas: repite lo del comando anterior pero despues lista por carpetas, pongo las primera lineas de la carpeta azul

```
azul:
cama_azul_como_lejos.avi  mesa_azul_como_lejos.avi
cama_azul_como_lejos.pdf  mesa_azul_como_lejos.pdf
cama_azul_como_lejos.txt  mesa_azul_como_lejos.txt
cama_azul_como_loco.avi   mesa_azul_como_loco.avi
cama_azul_como_loco.pdf  mesa_azul_como_loco.pdf
cama_azul_como_loco.txt  mesa_azul_como_loco.txt
```

En realidad tengo que darle más vueltas a esto porque no entiendo lo que estoy haciendo

Jorval

---



Estaba probando unas cosas y me surgió esta duda: que pasa si quiero acceder a un directorio que su nombre es , por ejemplo: "Rata Blanca" es decir, que tiene un espacio...

Haciendo `cd Rata Blanca` no me toma el comando...

## Fabricio

---

**Fabricio:** sí, aparentemente la consola interpreta el espacio en blanco como una especie de interrupción, como si fueran dos cosas distintas: "Rata" por un lado y "Blanca" por el otro.

Creo que hay dos soluciones:

1. Entrecomillar `cd "Rata Blanca"`
2. Usar la barra invertida `cd Rata\ Blanca`

--- : ---

A mí me sucede lo mismo que a Jorge, el primer "ls"

```
ls *.*(x|y)
```

el que supuestamente no deberíamos usar según el libro, me da el resultado esperado, es decir: listar únicamente los ficheros que no terminan con la extensión ".x" o ".y" del directorio "combinado";

el segundo "ls"

```
ls !(*.x|*.y)
```

hace lo mismo, pero recursivamente con lo que hay en el directorio "combinado" y los subdirectorios "azul", "roja", "verde" y "gris".

**Para ponerlo a prueba y tratar de entender, hice lo siguiente** (no es un script, se puede tipear sucesivamente en la terminal):

```
# creo un directorio de prueba.

mkdir prueba

# entro al directorio.

cd prueba

# el comando touch crea archivos vacíos. Genero una lista de
archivos que comienzan con la cadena "arq", siguen con un
dígito,
# y terminan con una extensión que resulta de la combinación de
dos de las letras "t", "x" e "y".
```

```
touch arq{0..9}.{t,x,y}{t,x,y}

# creo un subdirectorio, me muevo a él y repito el
procedimiento.

mkdir subdir
cd subdir
touch arq{0..9}.{t,x,y}{t,x,y}

# Ahora tengo dos listas iguales, una dentro de la otra.

# vuelvo al directorio "prueba" y pruebo los dos comandos "ls"
con el criterio de que liste todo menos los ficheros con
# extensión formada por dos letras repetidas.

cd ..
ls !(*.xx|*.tt|*.yy)
ls *!(xx|tt|yy)
```

### Vean lo que sucede...

#### Alejandro

---

Respecto a lo que le ocurre a Fabricio con el nombre de su archivo: Rata Blanca. Sucede por el espacio que dejó en el nombre del archivo y por eso que recomiendan que los nombres de los archivos no tengan espacios en blanco y pongamos todo junto o bien separados por un guión \_ La solución es la que indica **Alejandro**.

Respecto a los comodines extendidos hice algunas pruebas.

a.- Listar todos los archivos de la carpeta azul que tengan las extensiones .pdf o .avi

```
cd combinado/azul
ls *+ (.pdf|.avi)
```

y me entregó listado con todos los archivos .pdf y .avi

b.- Listar todos los archivos de la carpeta azul que no tengan las extensiones .pdf o .avi

```
cd combinado/azul
ls *!(.pdf|.avi)  No funcionó, me entregó todos archivos de la
carpeta
ls !(.pdf|.avi)  No funcionó, me entregó todos los archivos de
la carpeta
ls !(pdf|avi)    No funcionó, me entregó todos los archivos de
la carpeta

ls !(*.pdf|*.avi)  SÍ FUNCIONÓ , me entrego listado de todos los
archivos .txt
```

Creo que el asunto es probar con cada variante. La terminal es rápida y empleando el historial de comandos no hay que teclear tanto. Esa es mi experiencia con estos comodines.

Jorval

---

## Comodines extendidos:

Está buena la propuesta metodológica experimental de Jorge. Lo que sería importante es que lleguemos a la conclusión de por qué funciona o no funciona, encontrarle la lógica, para que después tengamos claridad al momento de ponerlo en práctica en un sistema concreto y real.

Entre las opciones de prueba faltó ésta, **que también funciona** en ese caso particular:

```
ls *.*!(pdf|avi)
```

El asunto se complica cuando, por ejemplo, muevo la carpeta "verde" adentro de la "azul":

```
cd ~/combinado
mv verde azul
```

Y ahora pruebo las dos opciones de listado:

```
cd azul
ls *.*!(pdf|avi)
ls !(*.pdf|*.avi)
```

El primero lista únicamente los ".txt" del directorio "azul". El segundo lista lo mismo, pero además lista todo el contenido de "verde" que está dentro de "azul".

**Conclusión:** habría que usar la sintaxis número 1 (que es exactamente lo contrario de lo que dice el libro). Pero insisto, **creo que lo realmente importante es llegar a comprender la lógica de por qué funciona o no, y por qué funciona como funciona.**

Por ejemplo:



¿Por qué no funcionan?

```
ls *.*!(.pdf|.avi)
ls !(.pdf|.avi)
ls !(pdf|avi)
```

- El **tercero** lista todo lo que hay, recursivamente, porque busca todos los ficheros o directorios que no tengan por nombre "pdf" o "avi". Como no hay ninguno con ese nombre...
- El **segundo**, lo mismo, no existe ningún fichero o directorio que tenga por nombre ".pdf" o ".avi" (que serían archivos ocultos). Lista todo.

## Grupo de estudios: Programación Bash Script

Primera semana: del 8 al 14 de octubre de 2009

---

- Pero ¿Y el **primero**? ¿No debería interpretarse como "\*" cualquier cosa seguida por algo que no sea ".pdf" o ".avi"?

Alejandro

---

### Dilema:

Creo que entendí el primer dilema:

```
ls *!(pdf|avi)
ls !(*.pdf|*.avi)
```

En el **primer caso**, lista todos los **ficheros** con cualquier cosa "\*" seguida de un "." y una cadena **distinta** de "pdf" o "avi". Como normalmente un directorio no tiene un nombre, un punto y una extensión, no va a listar el contenido de "verde".

En el **segundo caso**, lista todo lo que no es un fichero con la siguiente estructura "\*.pdf" o "\*.avi"; lo cual incluye listar el subdirectorio "verde", que no es un fichero con esas características, y por lo tanto lista su contenido.



**Sigo sin entender por qué no funciona** `ls *!(.pdf|.avi)`

Alejandro

---

No habrá algún conflicto con `*(pattern-list)` que significaba "**cero o más ocurrencias de pattern-list**" y capáz que como usamos 2 juntos, es decir el \* y el ! quizás tenga algún problema con eso... como si al pattern-list le ponemos un \* y un ! a la vez... `*!(pattern-list)`.

Es lo primero que me vino a la mente...

Fabricao

---

Siguiendo el **método inductivo Jorvaliano**:

```
ls *?(.pdf|.avi)
```

**Cero o una ocurrencia** de los elementos del patrón: lista todos los ".avi" y ".pdf" de "azul".

```
ls *+(.pdf|.avi)
```

**Una o más ocurrencias**: hace lo mismo.

```
ls *@(.pdf|.avi)
```

**Exactamente uno de los patrones** de la lista: el mismo resultado.

```
ls *!(.pdf|.avi)
```

**Cualquier cosa excepto** los elementos del patrón: Lista todos los ficheros de "azul" y de "verde" sin importar su extensión.



O tiene razón **Fabricio** y no se pueden juntar el comodín "\*" con los comodines extendidos, o hay algo que se nos está escapando.

**Todos mis ahorros para el que resuelva esto antes del jueves!!!!!!**

Alejandro

---

Creo que el asunto es no equivocarse al escribir los archivos o directorios. Veamos:

Me puse en cd combinado/verde

```
ls ?(.avi|.txt) No funcionó. Claro, no existe el fichero o
directorio
ls ?(*.avi|*.txt) Funciona
ls *?(.avi|.txt) Funciona
ls ?(casa*|patio*) Funciona

ls *(.avi|.txt) No funciona, no existe el fichero o directorio
ls *(*.avi|*.txt) Funciona
ls **(.avi|.txt) Funciona. Lista todo, no solo la cadena
ls *(casa*) Funciona

ls @(.avi|.txt) No funciona, no existe el fichero o directorio
ls @..... todo igual al anterior comodín

Me salto el comodín + porque es lo mismo

ls !(.avi|.txt) No funcionó. Listó todo el directorio verde,
claro .avi y .txt no son...
ls !(*.avi|*.txt) Funcionó. Listó sólo los archivos .pdf
ls !(casa*|mesa*) Funcionó. Listó solo los archivos cama y patio
```

Con esto doy por terminado estos comodines extendidos

Jorval

---



## AUTOEVALUACIÓN Final (primera semana)

- Les propongo que cada uno haga una autoevaluación sobre el tema comodines.

- **No publiquen aquí los resultados** para no dar pistas de resolución a los demás, así cada uno evalúe realmente en qué situación quedó al finalizar la primera semana de estudios. En todo caso, **me mandan las soluciones a mí**, yo espero a tener las de todos (sin mirarlas) y publico las distintas soluciones. Así nadie se ve influido por los demás.
- Lo que cada uno sí puede publicar es qué item consiguió resolver y cuál no, identificándolos por el número correspondiente. Probar siempre el procedimiento en una terminal para estar absolutamente seguro de que las cosas salen como uno imagina.
- Para resolver los desafíos podemos usar cualquier tipo de comodín, tratando de encontrar la combinación más efectiva y económica. Si encuentran distintas soluciones y las quieren poner, bienvenido sea.
- El **plazo de entrega** sería el próximo miércoles cuando termine la segunda semana. Ese día, publicamos los resultados.

Comodín	Descripción
?	Uno y sólo un caracter
*	Cero o más caracteres
[conjunto]	Uno de los caracteres del conjunto
[!conjunto]	Un caracter que no pertenezca al conjunto
?(pattern-list)	Cero o una ocurrencia
*(pattern-list)	Cero o más ocurrencias
+(pattern-list)	Una o más ocurrencias
@(pattern-list)	Exactamente uno de los patrones
!(pattern-list)	Cualquier cosa menos uno de los patrones

La **lista de ficheros** sobre la que propongo trabajar es la siguiente:

```
# En la terminal

cd ~
mkdir autoevall
cd autoevall
touch {a..d}
touch {a..d}{a..d}
touch {a..d}{a..d}{a..d}
ls

a    ab   aca  adb  bab  bbc  bcd  c    cb   cca  cdb  dab  dbc
dcd
aa   aba  acb  adc  bac  bbd  bd   ca   cba  ccb  cdc  dac  dbd
dd
aaa  abb  acc  add  bad  bc   bda  caa  cbb  ccc  cdd  dad  dc
dda
```

aab abc acd b bb bca bdb cab cbc ccd d db dca  
ddb  
aac abd ad ba bba bcb bdc cac cbd cd da dba dcb  
ddc  
aad ac ada baa bbb bcc bdd cad cc cda daa dbb dcc  
ddd



### Último desafío:

Listar todos los ficheros que:

1. comienzan con una "a".
2. terminan con una "a".
3. tienen una "a" en el medio.
4. tienen al menos una "a".
5. tienen una sola "a".
6. tiene al menos dos "a".
7. tienen dos "a" en cualquier posición.
8. tienen dos "a" juntas.
9. no tienen ninguna "a".
10. están formadas por una sola letra.
11. están formadas por dos letras.
12. están formadas por tres letras.
13. están formadas por dos letras distintas.
14. están formadas por dos letras iguales.
15. están formadas por tres letras distintas.
16. están formadas por tres letras iguales.

Les aclaro que no sé si todos se pueden resolver, todavía ni lo intenté. **Si se les ocurren otros desafíos** con ese listado, los agregan al final, continuando la numeración. Si quieren proponer otro tipo de desafío, lo colocan a continuación del listado con la correspondiente consigna explicando qué hay que hacer.



**Suerte!**

Alejandro

---