

BASH - Actividades 01

[Información sobre el documento.....1](#)

[1. - Introducción.....2](#)

[2. - Actividades.....2](#)

[ej00.sh Hola mundo.....2](#)

[ej01.sh - Mensaje de información.....2](#)

[ej02.sh - Listado usuarios con cuenta.....3](#)

[ej03.sh - Usuarios conectados.....3](#)

[Comando para resolver el problema.....3](#)

[ej04.sh - Usuarios conectados.....4](#)

[Comando para resolver el problema.....4](#)

[ej05.sh - Mi Backup.....4](#)

[Comando para resolver el problema.....4](#)

[ej06.sh - Backup jpg.....5](#)

[ej07.sh - Backup png.....5](#)

[ej08.sh - Backup bmp.....5](#)

[ej09.sh - Backup jpg.....5](#)

[ej10.sh - Backup png.....5](#)

[ej11.sh - Backup bmp.....5](#)

[ej12.sh - Backup por parámetro.....6](#)

[Comando para resolver el problema.....6](#)

[Preguntas.....6](#)

[ej13.sh - Backup de documentos.....6](#)

[ej14.sh - Backup de musica.....7](#)

[ej15.sh - Backup de vídeos.....7](#)

[ej16.sh - Backup de vídeos.....7](#)

Información sobre el documento

El objetivo de este documento es enseñar, compartir conocimientos para facilitar el aprendizaje. Este documento es mejorable, y será actualizado si es preciso.

Este documento puede ser utilizado para uso personal, no comercial, como se presenta, respetando una serie de **condiciones de uso basadas en Licencia Creative Commons**, como aparece en el logotipo, que se describe a continuación



- El **documento se proporciona como está**, por tanto *no se pueden realizar modificaciones*, ni en el **formato** ni en el **contenido**, ni trabajos derivados, sin la autorización expresa del autor.
- Se debe **mencionar al autor** del mismo, por supuesto sin modificar los enlaces o imágenes introducidas por éste en el documento.
- **No se puede utilizar** este material **con fines lucrativos, comerciales o cualquier uso que pueda proporcionar, directa o indirectamente, un beneficio económico de terceros**, sin la autorización expresa del autor.
- Si se desea publicar el documento en algún sitio web, se debe hacer a través de un **enlace al documento en el sitio del autor** (www.educatica.es o cursos.educatica.es)



Si se desea publicar como un recurso dentro de un sitio web, sin utilizar un enlace al material en el sitio web del autor, se debe solicitar autorización expresa y, en cualquier caso, referenciar el sitio web del autor (www.educatica.es).

Espero sea de provecho ;)

1. - Introducción

El objetivo de este documento es practicar en la creación de shell scripts bash.

Para sacar el máximo partido al documento, se recomienda leer el apartado donde se presentan los contenidos en el material didáctico proporcionado en el sitio web de [educatica](http://educatica.es) o de terceros autores.

2. - Actividades

ej00.sh *Hola mundo*

Con el editor de textos nano en modo texto crea un nuevo fichero llamado **ej00.sh**.

El fichero deberá contener el siguiente código:

```
1 #!/bin/bash
2 #
3 # Autor: Javier Pedrero Martín - javier@educatica.es
4 # Fecha: 11/02/2019
5 #
6 echo "Hola mundo!! El clasico saludo :D"
```

Lleva a cabo el proceso para configurar como ejecutable un script y ejecútalo.

Responde, si quieres mentalmente, a las siguientes preguntas sobre el programa que has realizado.

1. ¿Cuántos comandos se han ejecutado?
2. ¿Tiene sentido hacer este script para ejecutar únicamente este comando?
3. ¿Tendría sentido en sistemas reales crear un script que ejecutar un único comando?. Pon un ejemplo.

ej01.sh - *Mensaje de información*

Crea un script llamado **ej01.sh** que muestre en pantalla la siguiente información:

```
Hola, me llamo tu nombre y estoy estudiando tus estudios actuales.
Este es un programa básico que solo muestra información.
La utilidad de este tipo de programas básicos es poder ejecutar varios comandos, uno detrás de otro.
Podemos escribir una lista de comandos o llamadas a aplicaciones en orden y el interprete de comandos o shell se encargará de ejecutarlos uno a uno.
Muchas gracias por tu atención.
Un saludo,
tu nombre.
```

Lleva a cabo el proceso para configurar como ejecutable un script y ejecútalo.

Responde, si quieres mentalmente, a las siguientes preguntas sobre el programa que has realizado.



4. ¿Cuántos comandos has ejecutado?. Aunque sea el mismo comando, se ejecutan varias veces.
5. ¿Cuánto tiempo ha tardado en ejecutarse el script?.
6. Sin utilizar un script, ¿podrías mostrar en pantalla todos esos mensajes individuales que conforman el mensaje completo que has mostrado en pantalla?
7. Si tuvieras que escribir y ejecutar cada comando de forma manual, uno tras otro, ¿cuánto tiempo crees que tardarías en hacer el script?
8. Si tuvieras que mostrar de nuevo el mismo mensaje manualmente, ¿cuanto tardarías?.
9. Si tuvieras que mostrar de nuevo el mismo mensaje ejecutando el el script, ¿cuanto tardarías?

ej02.sh - Listado usuarios con cuenta

Como sabes, en el fichero `/etc/passwd` se almacena toda la información de cuentas de usuario en el sistema. Cada línea de este fichero almacena información para un solo usuario. En cada línea, hay una serie de campos separados por el carácter delimitador ":". En total, hay 7 campos.

Si quieres consultar más información sobre este fichero de configuración puedes hacer uso del comando `man`:

```
man 5 passwd
```

Vamos a crear un script llamado **ej02.sh** que muestre en pantalla los nombres de las cuentas de usuario que utilicen como shell el interprete de comandos `/bin/bash` ordenadas alfabéticamente.

Para ello, tendremos que utilizar varios comandos enlazados por tuberías.

```
cat /etc/passwd | grep /bin/bash | cut -d":" -f1 | sort
```

Lleva a cabo el proceso para configurar como ejecutable un script y ejecútalo.

Responde, si quieres mentalmente, a las siguientes preguntas sobre el programa que has realizado.

1. ¿Cuántos comandos se han ejecutado?
2. ¿Tiene sentido hacer este script para ejecutar únicamente este comando?. ¿Por qué?

ej03.sh - Usuarios conectados

Crea un script llamado **ej03.sh** que mostrará los usuarios conectados en el sistema. El comando a utilizar es `who`, que muestra los usuarios conectados en el sistema.

Pero cuidado, si un mismo usuario ha iniciado sesión varias veces, aparecerá una entrada en pantalla para cada sesión abierta:

```
javier@educatica:~$ who
javier  :0          2019-03-17 10:25 (:0)
javier  tty3       2019-03-18 00:48
```

Queremos que nuestro script tan solo muestre el nombre de los usuarios conectados, pero sin repeticiones. Es decir, si aparece, como en este caso dos usuarios javier, tan solo mostrará un solo usuario javier. Tampoco queremos más información que el nombre de la cuenta de usuario.

Comando para resolver el problema

Debes pensar en el comando que deberías utilizar para resolver el problema. Tendrás que utilizar `who`, para obtener toda la información. Filtrar la información con `cut`, para quedarte solo con el campo que necesitamos: el nombre del usuario. Por último, queremos quitar campos o valores repetidos, así que deberíamos utilizar el comando `uniq`. Si además, queremos que los datos queden ordenados deberíamos utilizar el comando `sort`.



Si no sabes como funciona alguno de estos comandos, consulta su página de manual. Tendrás que utilizarlos conectados con tuberías para que vayan filtrando y preparando la información.

Al final el script quedará con una única línea que ejecute dicho comando.

ej04.sh - Usuarios conectados

Este ejercicio nos solicita crear un script cuyo nombre será **ej04.sh** que hará casi lo mismo que el script del caso anterior. Se desea que se muestre el nombre de los usuarios conectados en el sistema sin repeticiones. Así que podemos copiar el script anterior con el nuevo nombre ej04.sh

```
cp ej03.sh ej04.sh
```

Editamos el fichero ej04.sh y debemos modificarlo para que ahora, además de mostrar por pantalla la lista de usuarios, muestre más información y la guarde en un fichero.

Deberá mostrar la fecha y hora actuales por pantalla y guardarla en un fichero llamado usuarios.log. Si el fichero existe, añadirá la fecha y la hora al final del fichero.

Después, mostrará en pantalla la lista de usuarios conectados al sistema y, además, la añadirá en un fichero llamado usuarios.log. La lista de usuarios contendrá solo los nombres de los usuarios conectados, deberá estar ordenada y sin repeticiones.

Comando para resolver el problema

La solución es bastante parecida a la del script anterior, solo que además de mostrar información tendremos que añadirla a un fichero. Para ello tendrás que hacer uso de redirecciones.

El único comando nuevo, aparte de las redirecciones, es el comando date que nos permite consultar la fecha y hora del sistema.

ej05.sh - Mi Backup

Crea un script llamado **ej05.sh** que muestre en pantalla un mensaje saludando al usuario actual, es decir, mostrando el nombre del usuario actual.

Después, creará en el directorio personal del usuario actual un directorio llamado backup. Dentro del directorio backup, creará los siguientes directorios: documentos, imagenes, musica y videos.

Para terminar mostrará un mensaje de despedida y terminará mostrando información de permisos, propietario y grupo del directorio backup, no de su contenido.

Comando para resolver el problema

Los comandos a utilizar son echo, para mostrar mensajes. Comando mkdir, que nos permitirá crear directorios. Comando ls con las opciones adecuadas como parámetro para que muestre información de permisos de un directorio, no de su contenido. Como novedad está en como manejar la ruta del directorio personal del usuario que ejecuta el script y el nombre de usuario.

Para esto, podemos hacer uso tanto de la variable \$HOME como del símbolo ~, que contienen la ruta del directorio personal del usuario actual, para obtener la ruta de dicho directorio. Para obtener el nombre del usuario actual, podemos utilizar la variable \$USER

Como pega, si tratamos de ejecutar por segunda vez este script y los directorios ya existen, se mostrará un mensaje de error un tanto molesto.

¿Se te ocurre como acabar con ese problema?

ej06.sh - Backup jpg

Crea una script llamado **ej06.sh** que copie todos los ficheros con extensión **jpg** que estén en el directorio personal del usuario actual en el directorio backup/imagenes que está dentro de su directorio personal. Este directorio, se supone que, lo creamos con el script ej05.sh.

ej07.sh - Backup png

Crea una script llamado **ej07.sh** que copie todos los ficheros con extensión **png** que estén en el directorio personal del usuario actual en el directorio backup/imagenes que está dentro de su directorio personal. Este directorio, se supone que, lo creamos con el script ej05.sh.

ej08.sh - Backup bmp

Crea una script llamado **ej08.sh** que copie todos los ficheros con extensión **bmp** que estén en el directorio personal del usuario actual en el directorio backup/imagenes que está dentro de su directorio personal. Este directorio, se supone que, lo creamos con el script ej05.sh.

ej09.sh - Backup jpg

Crea una script llamado **ej09.sh** que copie todos los ficheros con extensión **jpg** que estén en el directorio personal del usuario actual en el directorio backup/imagenes que está dentro de su directorio personal. Este directorio, se supone que, lo creamos con el script ej05.sh.

A diferencia del anterior, se desea utilizar una variable en la que se almacenará la extensión de los ficheros a copiar. El nombre de la variable será **extension** y se le asignará el valor de la extensión de los ficheros a copiar.

Utilizando la variable, deberás realizar la copia de los ficheros con la extensión especificada en el directorio de imagenes.

ej10.sh - Backup png

Crea una script llamado **ej10.sh** que será una copia del script ej09.sh.

```
cp ej09.sh ej10.sh
```

Modifica el script lo menos posible para que **en lugar de copiar todos los ficheros con extensión jpg copie los ficheros con extensión png**, que estén en el directorio personal del usuario actual en el directorio backup/imagenes que está dentro de su directorio personal.

Responde, mentalmente, a estas preguntas:

1. ¿Qué has tenido que cambiar?

ej11.sh - Backup bmp

Crea una script llamado **ej11.sh** que será una copia del script ej09.sh.

```
cp ej09.sh ej11.sh
```

Modifica el script lo menos posible para que **en lugar de copiar todos los ficheros con extensión jpg copie los ficheros con extensión bmp**, que estén en el directorio personal del usuario actual en el directorio backup/imagenes que está dentro de su directorio personal.

Responde, mentalmente, a estas preguntas:

1. ¿Qué has tenido que cambiar?
2. Funciona y es sencillo de hacer, pero... ¿qué problema ves en esta forma de trabajar?

ej12.sh - Backup por parámetro

Crea una script llamado **ej12.sh** que será una copia del script ej09.sh.

```
cp ej09.sh ej12.sh
```

Modifica el script lo menos posible para que **en lugar de copiar todos los ficheros con extensión jpg copie los ficheros con la extensión pasada en el primer parámetro del script**, que estén en el directorio personal del usuario actual en el directorio backup/imagenes que está dentro de su directorio personal.

Comando para resolver el problema

Según hemos visto en las explicaciones teóricas, el interprete de comandos bash se encarga de asignar a unas variables especiales el valor de los parámetros que se pasan al script al ejecutarlos por el usuario.

Para acceder al parámetro 1 se utiliza la variable \$1, al parámetro 2 se utiliza la variable \$2 y así sucesivamente hasta el parámetro 9 con \$9.

Podemos acceder a toda la lista de parámetros con la variable especial \$* o \$@, que contiene una cadena de texto con todos los parámetros separados por espacios.

Como ya estamos usando una variable, lo único que tendremos que hacer, ya que no estamos realizando comprobaciones, es asignar a la variable el valor del primer parámetro.

Preguntas

Responde, mentalmente, a estas preguntas:

1. ¿Qué has tenido que cambiar?
2. Ejecuta el nuevo script con todos los parámetros anteriores de imágenes. ¿Qué sucede?
3. Ejecuta el nuevo script con un parámetro propio de extensión de documentos, por ejemplo pdf. ¿Qué sucederá?
4. Ejecuta el nuevo script con un parámetro propio de extensión de musica, por ejemplo mp3. ¿Qué sucederá?
5. ¿Para qué nos sirve pasarle parámetros a un script?
6. Ejecuta el nuevo script pero sin pasarle ningún parámetro. ¿Qué sucederá?
7. ¿Qué problemas produce el paso de parámetros a un script?

ej13.sh - Backup de documentos

Crea una script llamado **ej13.sh** que será una copia del script ej12.sh.

```
cp ej12.sh ej13.sh
```

Modifica el script lo menos posible para que **copie todos los ficheros con la extensión pasada en el primer parámetro del script al directorio ~/backup/documentos, en lugar de al directorio ~/backup/imagenes.**

Deberás crear y utilizar una nueva **variable** con nombre **destino** que contendrá la ruta del directorio de destino. En este caso, la ruta del directorio de destino será **~/backup/documentos**.

Esta nueva variable deberá ser utilizada en el comando de copia para especificar la ruta del directorio destino de la copia.

ej14.sh - Backup de musica

Crea una script llamado **ej14.sh** que será una copia del script ej13.sh.

```
cp ej13.sh ej14.sh
```

Modifica el script lo menos posible para que **copie todos los ficheros con la extensión pasada en el primer parámetro del script al directorio ~/backup/musica**, en lugar de al directorio ~/backup/documentos.

Piensa, utilizas utilizas una variable para determinar el destino de la copia. Esta variable contiene la ruta del directorio destino, ¿qué deberás hacer para cambiar el destino de la copia?

ej15.sh - Backup de vídeos

Crea una script llamado **ej15.sh** que será una copia del script ej14.sh.

```
cp ej14.sh ej15.sh
```

Modifica el script lo menos posible para que **copie todos los ficheros con la extensión pasada en el primer parámetro del script al directorio ~/backup/videos**, en lugar de al directorio ~/backup/musica.

Piensa, utilizas utilizas una variable para determinar el destino de la copia. Esta variable contiene la ruta del directorio destino, ¿qué deberás hacer para cambiar el destino de la copia?

Responde, mentalmente, a las siguientes preguntas.

1. ¿Qué tienen en común los últimos 4 scripts?
2. ¿Qué tienen diferente los últimos 4 scripts?
3. ¿Cómo se te ocurre solucionar el problema de tener tantos scripts diferentes que hacen casi casi lo mismo?

ej16.sh - Backup de vídeos

Crea una script llamado **ej16.sh** que será una copia del script ej14.sh.

```
cp ej15.sh ej16.sh
```

Modifica el script lo menos posible para que **copie todos los ficheros con la extensión pasada en el primer parámetro del script al directorio pasado como segundo parámetro dentro del directorio ~/backup**.

Es decir, el script recibe en el primer parámetro la extensión de los ficheros a copiar. En el segundo parámetro debería recibir el directorio de destino entre los creados dentro de ~/backup. documentos, imagenes, musica y videos.

Piensa, utilizas una variable para determinar el destino de la copia. Esta variable contiene la ruta del directorio destino, ¿qué deberás hacer para cambiar el destino de la copia?. **Ten en cuenta que en el segundo parámetro tan solo se indicará el nombre del directorio dentro de ~/backup, así que habrá que añadir algo delante del valor del segundo parámetro.**

Responde, mentalmente, a las siguientes preguntas.

1. Ejecuta el script, pasándole los parámetros adecuados. Es decir, en el parámetro 1 extensiones de documentos, como pdf, y en el segundo parámetro documentos.
1. Ejecuta el script, sin pasarle parámetros o pasándolo solo uno de ellos. ¿Qué sucede?
2. ¿Cómo podemos solucionar los problemas provocados por no pasar parámetro?
3. ¿Cómo podemos solucionar el problema de recibir parámetros equivocados?