

Ejercicios

José Antonio de la Rosa Cubero

Práctica 5

Ejercicio 1

```
#!/bin/bash

if ! test -d $1
then
    printf "El archivo %s no es un directorio\n" $1 >&2
    exit
fi

DIR=$1

if [ "$2" == "0" ]
then
    num=$2
elif let num=$2
then
    printf ""
else
    printf "%s no es un número\n" $2 >&2
    exit
fi

cd $DIR
for FILE in `find . -size -${ $num+1 }`
do
    echo $FILE
done
```

Ejercicio 2

```
#!/bin/bash

if ! test -d $1
then
    printf "El archivo %s no es un directorio\n" $1 >&2
    exit
fi

DIR=$1
cd $DIR
for FILE in `ls`
do
    if test -f $FILE
```

```

then
    Tipo="archivo regular"
elif test -L $FILE
then
    Tipo="enlace simbólico"
elif test -d
then
    Tipo="directorio"
else
    Tipo="de tipo desconocido"
fi

echo "$FILE es un $Tipo"
done

```

Ejercicio 3

```

#!/bin/bash

read -p "Pulsa un carácter alfanumérico: " CHAR

case "$CHAR" in
    [0-9]) echo "Fila 1";;
    [qQ] | [wW] | [eE] | [rR] | [tT] | [yY] | [uU] | [iI] | [oO] | [pP] ) echo
"Fila 2";;
    [aA] | [sS] | [dD] | [fF] | [gG] | [hH] | [jJ] | [kK] | [lL] | [ñÑ] ) echo
"Fila 3";;
    [zZ] | [xX] | [cC] | [vV] | [bB] | [nN] | [mM] ) echo "Fila 4";;
    *) echo "No es un carácter alfanumérico."
esac

```

Ejercicio 4

```

#!/bin/bash

read -p "Mes del que quiere el nombre completo: " MES

case "$MES" in
    1 | "ene") echo "enero";;
    2 | "feb") echo "febrero";;
    3 | "mar") echo "marzo";;
    4 | "abr") echo "abril";;
    5 | "may") echo "mayo";;
    6 | "jun") echo "junio";;
    7 | "jul") echo "julio";;
    8 | "ago") echo "agosto";;
    9 | "sep") echo "septiembre";;
    10 | "oct") echo "octubre";;
    11 | "nov") echo "noviembre";;
    12 | "dic") echo "diciembre";;
    *) echo "No es un mes.";;
esac

```

Ejercicio 5

```
#!/bin/bash

num=0

while [ $num -lt 1 ] || [ $num -gt 10 ]
do
    read -p "Introduce un número entre 1 y 10: " num
done

echo "Gracias"
```

Ejercicio 6

La variable `$INTERVAL` controla la velocidad a la que se cambian las líneas. Disminuyéndola se moverá más rápido, aumentándola se moverá más despacio.

La opción `-e` de `echo`, activa la interpretación de la barra invertida `\`. Así:

```
$ echo -e "\n"

$ echo "\n"
\n
```