

A continuación se muestra una salida posible (y real) del mandato *who -a* en la máquina Metropolis:

```
[juancarl@Metropolis-01 octubre]$ who -a
system boot 2017-07-04 11:15
`run-level' 3 2017-07-04 11:15
LOGIN      tty3      2017-07-04 11:21      1986 id=3
LOGIN      tty1      2017-07-04 11:21      1982 id=1
LOGIN      tty6      2017-07-04 11:21      1992 id=6
LOGIN      tty5      2017-07-04 11:21      1990 id=5
LOGIN      tty4      2017-07-04 11:21      1988 id=4
LOGIN      tty2      2017-07-04 11:21      1984 id=2
juancarl + pts/0      2017-10-21 13:18      .      9193 (76.179.221.87.dynamic.jazztel.es)
jgzapata + pts/1      2017-10-21 14:00 00:07      10040 (90.173.81.100)
jgzapata + pts/2      2017-10-21 14:22 00:04      10571 (90.173.81.100)
pts/3      2017-10-18 14:44      4986 id=ts/3 term=0 salida=0
```

Realizar un script denominado *isHere*. Tiene un parámetro, que es el nombre de un usuario del sistema

La invocación será pues, por ejemplo

```
$ ./isHere jgzapata
```

La especificación del script es la siguiente:

1. Control del número de parámetros de entrada. El programa avisa y termina si el número de parámetros es incorrecto
2. Escribirá en pantalla el nombre y apellidos del autor.
3. Generará un fichero denominado *bruto* con la salida del mandato *who -a*
4. A partir de *bruto* generará un fichero denominado *noHeader* sin las dos primeras líneas
5. A partir de *noHeader* generará un fichero denominado *noLogin* sin las líneas que comiencen por LOGIN
6. A partir de *noLogin* generará un fichero denominado *realUsers* sin las líneas que no tengan nombre de usuario
7. A partir de *realUsers* se decidirá si el usuario pasado al script como parámetro está conectado y desde qué IP:

```
$ ./isHere jgzapata
jgzapata is connected from IP (90.173.81.100)
jgzapata is connected from IP (90.173.81.100)
$ ./isHere pedrito
$
```

8. Mejore el paso 7 de modo que no repitan las líneas con el mismo usuario e IP :

```
$ ./isHere jgzapata
jgzapata is connected from IP (90.173.81.100)
```

Lo que necesita saber

El siguiente bucle lee e imprime en pantalla, una a una, todas las líneas del fichero *file*:

```
while read field1 field2 restOffline
do
    echo $field1 $field2 $restOffline
done < file
```

Recomendaciones:

El tiempo es escaso. Asegúrese de que el programa funciona paso a paso.

Tiempo: 50 minutos

```
#!/bin/bash

if [ ! $# -eq 1 ]; then
    echo Use: $(basename $0) UserName
    exit
fi

rm $1 2>/dev/null
who -a > bruto
tail -n +3 bruto > noHeader
grep -v LOGIN noHeader > noLogin
grep -v "^ " noLogin > realUsers

while read user campo2 campo3 campo4 campo5 campo6 campo7 ip
do
    if [ $user = $1 ]; then
        # Connected!!
        # Look in the register of user IPs
        grep $ip $1 1>/dev/null 2>/dev/null

        if [ ! $? -eq 0 ]; then
            # Being here means that the IP has not been registered yet
            echo $1 is Connected from $ip
            # Add the IP to the register of user IPs
            echo $ip >> $1
            #cat $1
        fi
    fi
done < realUsers

rm $1 2>/dev/null
```