

# Sistemas Operativos

[Inicio](#)[Teoría](#) ▾[Prácticas](#)[Notebooks](#)[Exámenes](#) ▾[SO – Blog](#)[Logout](#)

[Inicio](#) » [PRACTICAS](#) » [Linux y BASH](#) » Esqueleto Práctica Parte Bash

## Esqueleto Práctica Parte Bash

[Linux y BASH](#)



Os dejo un posible esqueleto de cómo podríais programar el script loteria.sh que se plantea en el [enunciado de la práctica](#). No tenéis porque seguir al pie de la letra el esqueleto, simplemente es una guía para que tengáis en cuenta cómo podéis resolver la parte bash.

```
#!/bin/bash
#-----
# Definición de Constantes
#-----

## En esta sección definiremos constantes que vamos a usar en el código, como por ejemplo
## el número máximo de jugadores, o el número máximo de sorteos, así como cualquier código de
## error que vayamos a utilizar.

#Consantes
MAX_SORTEOS=
MAX_JUGADORES=
```

```

#Errores
ERR_NUM_ARGS=
ERR_NUM_SORTEOS=
ERR_NUM_JUGADORES=

#Códigos de retorno
RET_OK=0

#-----
# Funciones
#-----

## En esta sección definiremos todas las funciones que vamos a utilizar en la sección principal
## Definir las necesarias. Algunas funciones que podéis definir serían las siguientes.

Sintaxis(){
#-----
#Esta función se encarga de mostrar la ayuda de cómo utilizar el script y termina su ejecución
#Recibe como parametro el código de terminación

exit $1
}

Inicializar(){
#-----
#Esta función se encarga de inicializar el array de premios
#Recibe como parámetro el número de jugadores que entran en los sorteos que se ejecutan

clear
for jugador in `seq 1 $1`; do
    premios[$jugador]=0
done
}

ControlArgumentos(){
#-----
# Esta función se encarga de controlar que los argumentos están en el rango permitido
# Recibe el conjunto de argumentos que se han pasado al script
# Devuelve un código de error indicando el tipo de error que se ha producido
# Si piden ayuda con -? llama a Sintaxis con el código de retorno 0

if [ ]; then
    Sintaxis $RET_OK
fi

return $ERR_
}

```

```

LimpiarFicheros() {
#-----
# Esta función elimina los ficheros de ejecuciones anteriores

}

LeerFicherosPremios() {
#-----
# Esta función se encarga de leer uno a uno los ficheros de premios
# Recibe como parámetros el numero de sorteos y el de jugadores en cada sorteo
# Rellenará el array de premios, sumando a cada jugador su importe correspondiente al numero de sorteo

}

MostrarResultados() {
#-----
# Esta función se encarga de mostrar los resultados en el informe final.
}

#-----
# PRINCIPAL - CODIGO
#-----

## Esta es la sección principal donde comienza la ejecución del código del script, aqui haremos uso
## de las funciones, constantes y errores definidos en las secciones anteriores.

#Llamamos al control de argumentos pasándole los argumentos recibidos
ControlArgumentos $*
retVal=$?
if [ $retVal -eq $ERR_ ]; then
    #Mostramos mensaje y llamamos a Sintaxis
    echo "Se ha producido el error ...."
    Sintaxis $RET_
fi

#Si superamos el control de argumentos asignamos los argumentos a las variables
numSorteos=$1
numJugadores=$2

#Inicializamos el array de premios
Inicializar $numJugadores

#Limpiamos de ejecuciones anteriores
LimpiarFicheros

#Ejecutamos los sorteos en un bucle para el número de sorteos que nos pasan.
#En cada sorteo pasamos como parámetro el número de sorteo y el número de jugadores
#sorteo es el ejecutable en C que realiza el sorteo

```

```
#Una vez realizados los sorteos leemos los premios
LeerFicherosPremios $numSorteos $numJugadores

#Y finalmente mostramos los resultados
MostrarResultados

exit 0
```

[← Entrada anterior](#)

[Entrada siguiente →](#)

## Deja un comentario

Conectado como alu28. [Edita tu perfil](#). [¿Salir?](#) Los campos obligatorios están marcados con \*

Escribe aquí...

[Publicar comentario »](#)

Copyright © 2025 Sistemas Operativos  
Escuela Politécnica Superior de Elche  
Universidad Miguel Hernández  
Miguel Onofre Martínez Rach