

EXAMEN UD01 ASO

RA 7, c.e. c)

ASO: PE 1 UD 01 Fecha: 04/12/2.023.
Nombre: _____
Observación: en esta prueba se está evaluando el RA 7.
Tiempo estimado: 1h, 30'

RA 7, c.e. c) (10 puntos, 5%)
Uno de los ficheros de configuración que nos proporciona Linux relacionado con los usuarios es el `.profile`, el cual se puede adaptar a las necesidades concretas de cada uno de los usuarios. Te muestro a continuación un ejemplo:

```
# Bloque 1
umask 022

# Bloque 2
if [ -z "$BASH_VERSION" ]; then
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# Bloque 3
if [ -d "$HOME/bin" ]; then
    PATH="$HOME/bin:$PATH"
fi

# Bloque 4
alias cls='clear'
```

Explica claramente lo que se está haciendo en cada uno de los bloques.

Bloque 1

Con el comando `umask`, se establecen como máscara, los permisos predeterminados para directorios 775 y para ficheros 644.

777

022

755

666

022

644

Bloque 2

Se comprueba si está vacía la variable de entorno `$BASH_VERSION`, después comprueba si existe el fichero `.bashrc`, expandiendo el contenido de la variable de entorno `$HOME` siendo la ruta de conexión en el home del usuario. Si existe, ejecuta el fichero `.bashrc` con `.`

Bloque 3

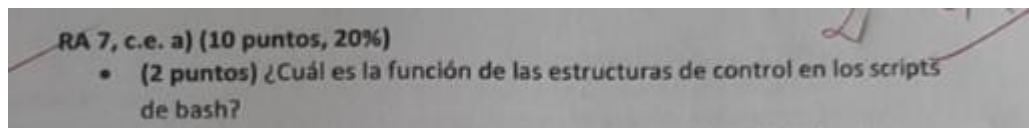
Comprueba en primer lugar si existe el directorio /home/"nombre_usuario"/bin, si existe se agrega al PATH la ruta \$HOME/bin:\$PATH. Por ejemplo:

```
If [ -d "$HOME/bin" ]; then
    PATH="/home/javier/bin:$PATH"
Fi
```

Bloque 4

En este bloque, se especifica en el fichero .bashrc un alias al comando clear llamado cls, con el que al ejecutar en la terminal el comando cls, borrará pantalla, por que internamente llamará al comando clear.

RA7, c.e.a)



La función de las estructuras de control en los scripts de bash son dirigir el flujo de ejecución del programa, según los condicionales y bucles para ejecutar código repetidamente. Principales estructuras de control en Bash

- Condicionales: if, else, elif

```
GNU nano 4.8 ejemplos_estructurasControl.sh
# Condicionales

read -p "Dime tu edad: " EDAD

if [ $EDAD -lt 18 ]; then
    echo "Eres menor de edad, no pasas"
elif [ $EDAD -gt 50 ]; then
    echo "Eres un adulto, descuento por adulto"
else
    echo "Eres un joven mayor de edad"
fi
```

- Bucles: For, While

```
# Bucles
# Bucle For
FICHERO="personas.txt"

for PERSONA in $(cat "$FICHERO"); do
    echo "Hola $PERSONA, ten un buen día"
done

# Bucle while

NUMERO=0

while [ $NUMERO -lt 10 ]; do
    echo "Este es el número $NUMERO"
    ((NUMERO++))
done
```

- Case

```
while [ "$OPCION" != "5" ]; do
    clear
    echo "Seleccione una acción a realizar:"
    echo "1.Iniciar servicio"
    echo "2.Detener servicio"
    echo "3.Reiniciar servicio"
    echo "4.Comprobar el estado del servicio"
    echo "5.Salir del programa"
    read -p "Su opción..." OPCION
    case ${OPCION} in
        1)
            read -p "Introduce el nombre del servicio que quieres iniciar: " INICIAR_SERV
            sudo systemctl start $INICIAR_SERV
            if [ $? -eq 0 ]; then
                echo "El servicio $INICIAR_SERV se ha iniciado correctamente"
            else
                echo "Se ha producido un error al iniciar el servicio $INICIAR_SERV"
            fi
            pausa
            ;;
        2)
            read -p "Introduce el nombre del servicio que quieres detener: " DETENER_SERV
            sudo systemctl stop $DETENER_SERV
            if [ $? -eq 0 ]; then
                echo "El servicio $DETENER_SERV se ha detenido correctamente"
            else
                echo "Se ha producido un error al detener el servicio $DETENER_SERV"
            fi
            pausa
            ;;
        3)
            read -p "Introduce el nombre del servicio que quieres reiniciar: " REINICIAR_SERV
            sudo systemctl restart $REINICIAR_SERV
            if [ $? -eq 0 ]; then
                echo "El servicio $REINICIAR_SERV se ha reiniciado correctamente"
            else
                echo "Se ha producido un error al reiniciar $REINICIAR_SERV"
            fi
            pausa
            ;;
        4)
            read -p "Introduce el nombre del servicio: " STATUS_SERV
            sudo systemctl status $STATUS_SERV
            pausa
            ;;
        5)
            echo "Has salido del programa, Hasta la vista..."
            ;;
        *)
            echo "No se ha encontrado tu opción, Por favor, escoje una de la lista"
            pausa
            ;;
    esac
done
```

- Funciones

```
# Función para validar el formato de un correo electrónico
validar formato correo() {
    FORMATO_CORREO='^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]{2,}$'
    if [[ $CORREO =~ $FORMATO_CORREO ]] ; then
        echo "La dirección de correo $CORREO tiene un formato válido"
        return 0
    else
        echo ";;La dirección de correo $CORREO no tiene un formato válido!!"
        return 1
    fi
}
```

```
6_4_validarCorreo.sh
#!/bin/bash

source "/usr/local/bin/marzoJunio/hoja_actividades2/6_funciones.sh"

read -p "Introduce una dirección de correo a validar: " CORREO
validar_formato_correo
```

• (6 puntos) Crea un script que solicite el nombre de un usuario y muestre en pantalla su directorio de conexión y el nombre del intérprete utilizado. Si no existe dicho usuario, le muestra un mensaje "No existe en el sistema <nombre_introducido>".

El script debe ejecutarse hasta que el usuario decida terminar.

Estructura del fichero que contiene las cuentas de usuario

```
odoo15:x:1002:1002:,,,administrador de Odoo:/home/odoo15:/bin/bash
```

Ejemplo de ejecución:

```
Introduzca login...pepe
No existe en el sistema...pepe
¿Desea continuar (s/n)? s
Introduzca login...odoo
Directorio conexión /home/odoo y shell /bin/bash
¿Desea continuar (s/n)? n
```

```
javier@javier-VirtualBox: /usr/local/bin/marcosguino
GNU nano 4.8 script1 supuestoExamen.sh
#!/bin/bash

FICHERO="/etc/passwd"

while true ; do
read -p "Introduzca login: " LOGIN

# Buscamos la línea correspondiente al usuario buscado
LINEA=$(grep ""$LOGIN:" "$FICHERO")

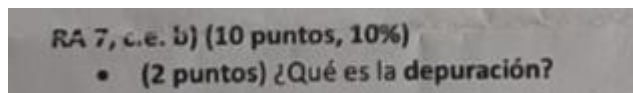
# Verificamos si el usuario existe preguntando si la línea existe en el fichero /etc/passwd
if [ -n "$LINEA" ]; then
    # Extraemos los campos del fichero /etc/passwd del directorio de conexión y la shell
    DIR_CONEXION=$(echo "$LINEA" | cut -d":" -f6)
    INTERPRETE=$(echo "$LINEA" | cut -d":" -f7)

    echo "Directorio de conexión: $DIR_CONEXION y shell $INTERPRETE"
else
    echo "El usuario no existe en el sistema"
fi

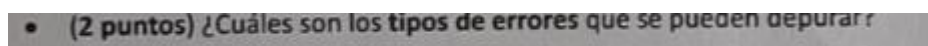
# Preguntamos al usuario si desea continuar con el programa
read -p "¿Desea continuar? (s/n): " RESPUESTA

# Si la respuesta es Sí, continuamos con la sentencia continue y el bucle volverá ejecutarse, en caso contrario si es no, salimos del bucle con la sentencia break
if [ "$RESPUESTA" = "s" ] || [ "$RESPUESTA" = "S" ]; then
    continue
else
    echo "Ok, ¡Hasta la vista!"
    break
fi
done
```

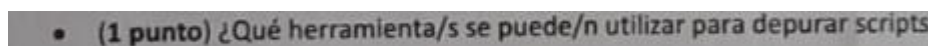
RA 7, c.e.b)



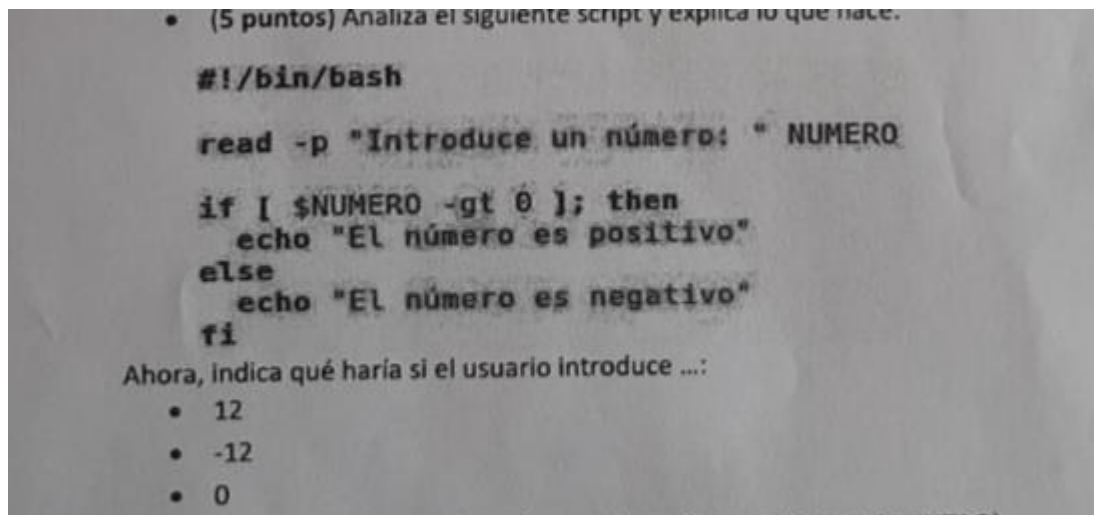
La depuración en los scripts en bash se refiere a las técnicas para analizar la traza de un script analizando cada uno de los caminos que toma el script en función a las condiciones, permitiendo así la depuración de errores.



Podemos depurar errores en la sintaxis, errores de lógica, errores de tiempo de ejecución y errores en variables no definidas.



El comando bash -x que permite ir viendo la traza del programa mientras se va ejecutando, o en su defecto, añadir en las líneas que queramos set -x para empezar a depurar por esa línea y set +x para terminar la depuración.



Este script le solicita a el usuario un número por teclado, después con el condicional if se evalúa la condición si el número introducido por el usuario es mayor que 0, entonces se muestra el mensaje que el número es positivo, en caso contrario si el número introducido es negativo por que es menor que 0, se mostraría el mensaje de que el número es negativo.

- Cuando el usuario mete un 12, se ejecuta el bloque de código del if. Es decir, el número es positivo.
- Cuando el usuario mete un -12, se ejecuta el bloque de código del else, sacando un mensaje por pantalla de que el número es negativo.
- Si el usuario introduce un 0, dirá que el número es negativo, por lo que está mal y este tipo de error es un error lógico.
- Si el usuario mete espacios en blanco dará errores, de tiempo de ejecución, por que se está comparando una cadena vacía con un número.

```
Terminal - javier@javier-Virt...  
Archivo Editar Ver Terminal Pestañas Ayuda  
javier@javier-VirtualBox: /usr/local/bin/marzoJunio  
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh  
Introduce un número: 12  
El número es positivo  
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh  
Introduce un número: -12  
El número es negativo  
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh  
Introduce un número: 0  
El número es negativo  
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh  
Introduce un número:  
./script2_supuestoExamen.sh: línea 5: [: -gt: se esperaba un operador unario  
El número es negativo  
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$
```

Esto lo podemos arreglar, cambiando el tipo de expresión numérica a expresiones de cadenas de texto, es decir en vez de `-eq ==`.

```
#!/bin/bash

read -p "Introduce un número: " NUMERO

if [[ "$NUMERO" == "0" ]]; then
    echo "El numero es 0"
elif [[ -z "$NUMERO" ]]; then
    echo "Error, no se aceptan espacios en blanco. Introduce un número positivo o negativo, Ejemplo: 12, -12"
elif [ $NUMERO -gt 0 ]; then
    echo "El número es positivo"
elif [ $NUMERO -lt 0 ]; then
    echo "El numero es negativo"
else
    echo "Número no reconocido"
fi

javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh
Introduce un número: 12
El número es positivo
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh
Introduce un número: -12
El numero es negativo
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh
Introduce un número: 0
El numero es 0
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$ ./script2_supuestoExamen.sh
Introduce un número:
Error, no se aceptan espacios en blanco. Introduce un número positivo o negativo, Ejemplo: 12, -12
javier@javier-VirtualBox: /usr/local/bin/marzoJunio$
```