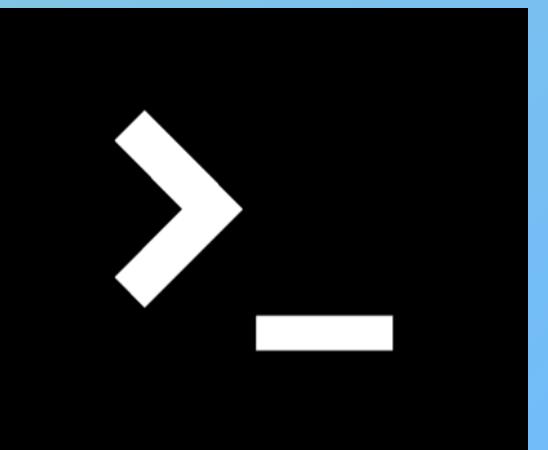


# Linux y Terminal

Clase 20



# <índice>

## Shell Script y Cron

**¿Qué es Shell Script?**

---

**Componentes de Shell Script**

---

**Cron**

---

**Ejemplos con cron**

# Shell Scripting

## ¿Qué es?

- Programación en nuestra terminal
- Recordar que la extensión de un fichero no importa
- Necesitamos permisos de ejecución para poder ejecutarlos
- `./nombredearchivo.sh` o `bash nombredearchivo.sh`
- Primera línea de nuestros documentos
  - `#!/bin/bash`

# Shell Scripting

## Nuestro primer script

bash

 Copy code

```
#!/bin/bash
echo "Hola Mundo"
```

# Shell Scripting

## Componentes

Comandos

Variables

Comentarios

Constantes

Strings

Caracteres comodines ~

input de entrada

argumentos de entrada

operaciones matemáticas

control de flujo

funciones con parámetros y

argumentos

Expansiones

# Shell Scripting

## Ejemplo 1: variables e impresión

bash

 Copy code

```
#!/bin/bash
nombre="Juan"
echo "Bienvenido, $nombre"
```

# Shell Scripting

## Ejemplo 2: condicionales

bash

 Copy code

```
#!/bin/bash
edad=18
if [ $edad -ge 18 ]; then
    echo "Eres mayor de edad"
else
    echo "Eres menor de edad"
fi
```

# Shell Scripting

## Ejemplo 3: iteraciones, for

bash

 Copy code

```
#!/bin/bash
for i in {1..5}; do
    echo "Número: $i"
done
```

# Shell Scripting

## Ejemplo 4: iteraciones, while

bash

 Copy code

```
#!/bin/bash
contador=0
while [ $contador -lt 5 ]; do
    echo "Contador: $contador"
    ((contador++))
done
```

# Shell Scripting

## Ejemplo 5: petición de valores con read

bash

 Copy code

```
#!/bin/bash
echo "¿Cuál es tu nombre?"
read nombre
echo "Hola, $nombre"
```

# Shell Scripting

## Ejemplo 6: funciones

bash

 Copy code

```
#!/bin/bash
saludar() {
    echo "Hola, $1"
}
saludar "Juan"
```

# Shell Scripting

## Ejemplo 7: procesar archivos

bash

 Copy code

```
#!/bin/bash
procesar_archivos() {
    directorio="$1"
    for archivo in "$directorio"/*; do
        # Realizar operaciones en cada archivo
        echo "Procesando $archivo"
    done
}

procesar_archivos "/ruta/a/procesar"
```

# Shell Scripting

## Ejemplo 8: procesar archivos

bash

 Copy code

```
#!/bin/bash

agregar_usuario() {
    usuario="$1"
    contraseña="$2"
    useradd "$usuario"
    echo "$usuario:$contraseña" | chpasswd
}

agregar_usuario "nuevo_usuario" "contraseña123"
```

# Shell Scripting

## Ejemplo 9: empaquetado

bash

 Copy code

```
#!/bin/bash

respaldar_archivos() {

    directorio_origen="$1"
    directorio_destino="$2"
    fecha=$(date +"%Y%m%d")
    tar -czf "$directorio_destino/respaldo_$fecha.tar.gz" "$directorio_origen"

}

respaldar_archivos "/ruta/a/respaldar" "/ruta/donde/guardar"
```

# Shell Scripting

## Ejemplo 10: sincronización y backup

```
bash                                         Copy code

#!/bin/bash

# Directorios fuente y destino
directorio_fuente="/ruta/del/directorio/fuente"
directorio_destino="/ruta/del/directorio/destino"

# Ejecutar rsync para sincronizar los directorios
rsync -av --delete "$directorio_fuente" "$directorio_destino"

# Verificar el código de salida de rsync
if [ $? -eq 0 ]; then
    echo "Sincronización completada con éxito"
else
    echo "Error durante la sincronización"
fi
```

- `rsync -av --delete "$directorio_fuente"` "\$directorio\_destino" realiza la sincronización, copiando los archivos del directorio fuente al directorio destino de manera recursiva (-a), mostrando el progreso (-v) y eliminando los archivos en el destino que no existen en el origen (--delete).
- El comando rsync devuelve un código de salida que indica si la operación fue exitosa (\$?). Se verifica y se muestra un mensaje adecuado en consecuencia.

## ¿Qué es?

- Es un servicio de programación de tareas que se ejecutan periódicamente en el sistema
- Permite a los usuarios y administradores de sistemas programar scripts y comandos para que se ejecuten en momentos específicos, ya sea una vez, de manera recurrente o en intervalos regulares.

# Crontab

## ¿Qué es?

- crontab es un archivo especial que contiene las definiciones de las tareas programadas.
- Cada usuario en el sistema puede tener su propio archivo crontab para definir tareas que se ejecutarán bajo su contexto.

# Crontab

## Comandos relacionados

- `crontab -e`: Permite editar el archivo crontab del usuario actual.
- `crontab -l`: Muestra el contenido del archivo crontab del usuario actual.
- `crontab -r`: Elimina el archivo crontab del usuario actual.
- `/etc/crontab`: El archivo crontab a nivel de sistema que define tareas para todos los usuarios.

# Crontab

## Campos de crontab

Minuto (0-59): El primer campo especifica el minuto en el que se ejecutará la tarea. Puede ser un número entre 0 y 59.

- Hora (0-23): El segundo campo especifica la hora en la que se ejecutará la tarea. Puede ser un número entre 0 y 23, donde 0 representa la medianoche.
- Día del Mes (1-31): El tercer campo especifica el día del mes en el que se ejecutará la tarea. Puede ser un número entre 1 y 31, dependiendo del mes. Tenga en cuenta que no todos los meses tienen 31 días, por lo que si especifica un valor mayor que el número de días en un mes, la tarea no se ejecutará.

# Crontab

## Campos de crontab

- Mes (1-12 o nombres abreviados de meses): El cuarto campo especifica el mes en el que se ejecutará la tarea. Puede ser un número entre 1 y 12, o el nombre abreviado del mes (por ejemplo, en inglés: Jan, Feb, Mar, etc.).
- Día de la Semana (0-7 o nombres abreviados de días de la semana): El quinto campo especifica el día de la semana en el que se ejecutará la tarea. Puede ser un número entre 0 y 7, donde 0 y 7 representan el domingo, o el nombre abreviado del día de la semana (por ejemplo, en inglés: Sun, Mon, Tue, etc.).
- Comando a Ejecutar: El último campo especifica el comando o script que se ejecutará en el momento especificado por los otros campos.

# Crontab

Ejemplo 1: cada día a las 2

bash

 Copy code

```
0 2 * * * /ruta/al/script.sh
```

# Crontab

Ejemplo 2: cada 15 minutos

bash

 Copy code

```
*/15 * * * * comando
```

# Crontab

Ejemplo 3: domingos a medianoche

bash

 Copy code

```
0 0 * * 0 /ruta/al/script.sh
```

# Crontab

Ejemplo 4: primer día de cada mes

bash

 Copy code

```
0 0 1 * * comando
```

# Crontab

Ejemplo 5: cada hora durante los días laborales

bash

 Copy code

```
0 * * * 1-5 /ruta/al/script.sh
```

# Crontab

Ejemplo 6: cada 5 minutos los días laborales

bash

 Copy code

```
*/5 * * * 1-5 comando
```

# <Despedida>

Email

**bienvenidosaez@gmail.com**

Instagram

**@bienvenidosaez**

Youtube

**youtube.com/bienvenidosaez**