

## 3. Automatización de Tareas

Copyright (c) 2025 Adrián Quiroga Linares Lectura y referencia permitidas; reutilización y plagio prohibidos

En Linux, diferenciar el "cuándo" se ejecuta una tarea es clave. Tenemos dos herramientas principales según nuestras necesidades:

1. **at / batch**: Para tareas que se ejecutan **una sola vez** en el futuro (ej: "Apaga el servidor a las 3 de la mañana").
2. **cron**: Para tareas **repetitivas** (ej: "Haz una copia de seguridad todos los viernes").

### 3.1 Ejecución Diferida (Una sola vez): at y batch

Estas herramientas ponen tareas en una cola de espera.

#### Comando at (A una hora fija)

Programa un comando para ejecutarse en un momento específico.

**Sintaxis:** `at [hora/fecha]`

**Flujo de trabajo interactivo:**

- Escribes `at 11:45`
- El sistema te da un prompt (`>at`) para que escribas los comandos que quieras
- Pulsas **CTRL+D** para guardar y salir (EOT)

Si el comando genera texto (ej: un `ls`), el sistema te enviará un **email** interno con el resultado (a menos que redirijas la salida `> archivo`). Es **persistentes**, se guarda en disco; si reinicias, el trabajo sigue pendiente (si la hora no ha pasado).

SHELL

```
$ at 11:45
warning: commands will be executed using /bin/sh
at> ls /tmp > /home/usuario/lista_tmp.txt    # Guarda el listado en un
fichero
at> env DISPLAY=:0 zenity --info --text="Hola" # Muestra ventana gráfica
(requiere variables de entorno)
at> <CTRL-D>
job 4 at Wed Nov 16 11:45:00 2025
```

#### Comando batch (Cuando el sistema descance)

Es una variante de `at`. No le dices "a qué hora", sino "hazlo cuando puedas"

- Ejecuta el trabajo solo cuando la **carga del sistema** baja de 1.5
- Ideal para tareas pesadas que no quieras que ralenticen el ordenador mientras lo usas

## Gestión de la cola (**atq** y **atrm**)

- **atq (Queue)**: Lista los trabajos pendientes.
  - El usuario normal ve los suyos. El root ve los de todos.
- **atrm [ID]**: Borra un trabajo específico (usando el número ID que da **atq**).

## 3.2 Ejecución Periódica (Recurrente): **cron**

**Cron** es el reloj del sistema. Utiliza un demonio (servicio en segundo plano) que comprueba cada minuto si hay algo que hacer.

### Gestión del fichero: **crontab**

Cada usuario tiene su propia tabla de tareas. NO se edita el fichero a mano, se usa el comando:

Comando	Acción
<b>crontab -e</b>	<b>Editar</b> tu tabla de tareas (abre nano/vim).
<b>crontab -l</b>	<b>Listar</b> tus tareas programadas.
<b>crontab -r</b>	<b>Remover</b> (borrar) toda tu tabla de tareas.

## Sintaxis de una línea de Cron

Cada línea en el fichero representan una tarea y tiene **6 campos**: 5 de tiempo y 1 de comando.

**Estructura:** `minuto hora día_mes mes día_semana comando`

Campo	Valores permitidos	Notas
<b>Minuto</b>	<b>0-59</b>	
<b>Hora</b>	<b>0-23</b>	Formato 24h.
<b>Día del mes</b>	<b>1-31</b>	
<b>Mes</b>	<b>1-12</b>	(O nombres: jan, feb...)
<b>Día Semana</b>	<b>0-7</b>	<b>0 y 7 son Domingo.</b> 1=Lunes, etc.
<b>Comando</b>	Ruta comando	Lo que se ejecutará.

## Operadores Especiales

- **\*** (Asterisco): "**Todos**". (Ej: **\*** en horas significa "todas las horas").

- **,** (Coma): **Listas.** (Ej: **1,15** = min 1 y min 15).
- **-** (Guion): **Rangos.** (Ej: **1-5** = de Lunes a Viernes).
- **/** (Barra): **Pasos/Repetición.** (Ej: **\*/15** = Cada 15 unidades).
  - **\*/2** en horas = A las 0, 2, 4, 6... (cada 2 horas).

## VARIABLES DE ENTORNO EN CRON

Al principio del fichero puedes definir variables globales:

- **MAILTO=pepe**: Si el comando falla o dice algo, envía el email a "pepe" (si está vacío **MAILTO=""**, descarta los emails).
- **SHELL=/bin/bash**: Qué intérprete usar.

## Ejemplo

```
# Borra temporales de lunes a viernes (1-5) a las 04:30 AM
30 4 * * 1-5 rm -rf /tmp/*

# Ejecuta un script cada 15 minutos (0, 15, 30, 45)
# Rango horario: de 00:00 a 08:00 y de 20:00 a 23:00
*/15 0-8,20-23 * * * /home/usuario/script_noche.sh

# Ejecuta algo el día 1 de cada mes a las 00:00
0 0 1 * * /root/backup_mensual.sh
```

SHELL

## 3.3 CRON DEL SISTEMA (DIRECTORIOS)

El administrador del sistema (root) tiene una forma alternativa de programar tareas sin pelearse con la sintaxis de los asteriscos.

Existen directorios especiales en **/etc/**. Si metes un script ejecutable dentro, el sistema lo lanzará automáticamente según el nombre de la carpeta:

- **/etc/cron.hourly**: Se ejecuta cada hora (en el minuto 17 aprox).
- **/etc/cron.daily**: Se ejecuta una vez al día (normalmente a las 6:25 AM).
- **/etc/cron.weekly**: Una vez a la semana.
- **/etc/cron.monthly**: Una vez al mes.