



Para ejecutar nuestra tarea de copia de seguridad debemos hacer uso de **cron**, el cual es un administrador regular de procesos en segundo plano que comprueba si existen tareas para ejecutar, teniendo en cuenta la hora del sistema.

Las configuraciones de las tareas a ejecutar se almacenan en el archivo **crontab** que puede ser editado con el comando `crontab -e`, si requerimos listar las tareas que tenemos configuradas ejecutamos `crontab -l`.

A continuación te muestro lo que se imprime en la pantalla al correr el comando `crontab -e`

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
```



Para establecer una tarea automatizada con **cron** se debe seguir un formato específico para definir una tarea como se muestra a continuación:

```
----- minute (0 - 59)
| .----- hour (0 - 23)
| | .----- day of month (1 - 31)
| | | .----- month (1 - 12) OR jan,feb,mar,apr ...
| | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fr
| | | | |
* * * * * user-name command to be executed
```

Lo siguiente sería definir la periodicidad de nuestro **cron**, para ello podemos hacer pruebas en el sitio <https://crontab.guru>. Nosotros queremos que nuestra copia se ejecute todos los días a las 03:15 de la mañana, pues es el momento donde menos tráfico tenemos en nuestra base de datos.

“At 03:15.”

next at 2019-08-16 03:15:00 random

15 3 * * *

minute

hour

day
(month)

month

day
(week)

Nuestro cron quedaría de la siguiente manera:

```
15 3 * * * /usr/bin/bash /home/platzi/copia.sh
```

Con esto se ejecutaría nuestro script en los horarios definidos.

Podríamos también realizar un escaneo automático de la máquina todos los días a las 05:00 PM y generar un reporte del estado de seguridad de la máquina con *lynis*.

```
* 17 * * * /usr/sbin/lynis --quick >  
/home/edison/Documentos/Platzi/lynis_files/$(date +"%F") 2>/dev/null
```

Verificamos la carpeta y encontramos la salida

| | | | |
|---|------------|---------|-------|
|  | 2019-08-07 | 26,8 kB | 7 ago |
|  | 2019-08-08 | 6,6 kB | 8 ago |
|  | 2019-08-10 | 30,2 kB | sáb |
|  | 2019-08-12 | 24,1 kB | lun |
|  | 2019-08-13 | 24,1 kB | mar |
|  | 2019-08-14 | 31,7 kB | Ayer |
|  | 2019-08-15 | 2,5 kB | 17:24 |

Esto funcionará bien siempre y cuando la máquina esté encendida, si no es así, podríamos revisar [anacron](#).