

GNU/Linux para programadores:

Automatizando tus aplicaciones con Cron

A la hora de resolver determinados requerimientos de Software es muy posible que la solución ideal implique la ejecución de ciertas tareas de forma automática y esto, solo será posible gracias *Cron*.

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)



Eugenia es **Arquitecta de Software**, **docente** instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y eXtreme Programming. Miembro de la **Free Software Foundation** e integrante del equipo de **Debian Hackers**.

Webs:

Cursos de programación a Distancia: www.cursosdeprogramacionadistancia.com
Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Cron es el administrador de procesos en segundo plano de los sistemas operativos GNU/Linux, permitiendo definir en un archivo crontab, la ejecución de comandos y/o *scripts* a intervalos regulares de tiempo.

Para **visualizar las tareas existentes**, basta con ejecutar crontab con el parámetro `-l`, como se muestra a continuación:

```
user@host:~$ crontab -l
```

El archivo crontab posee una sintaxis muy estricta en la cual, cada tarea debe especificarse en una línea. A la vez, cada línea deberá contar con la siguiente estructura:

```
# Para programar la ejecución de un script
periodicidad /ruta/al/script

# Para programar la ejecución de un comando
periodicidad comando
```

La periodicidad de ejecución puede indicarse con alguno de los intervalos preestablecidos:

```
@daily /home/eugenia/projects/myapp/bin/backup_db.sh
```

Entre los intervalos preestablecidos, podemos encontrar:

@yearly	Se ejecuta una vez al año, a las 0 hs del 1 de enero Alias: @annually
@monthly	Se ejecuta una vez al mes, a las 0 hs del primer día de cada mes
@weekly	Se ejecuta una vez a la semana, a las 0 hs del primer día de la semana (domingo)
@daily	Se ejecuta una vez al día, a las 0 hs Alias: @midnight
@hourly	Se ejecuta una vez por hora

Sin embargo, la flexibilidad de *cron* es muy amplia y nos permite ser más específicos al indicar un intervalo, utilizando la sintaxis:

```
MINUTO HORA DIA-DEL-MES MES DIA-DE-LA-SEMANA tarea
```

Donde cada valor será reemplazo por:

MINUTO:	Un entero entre 0 y 59
HORA:	Un entero entre 0 y 23
DIA-DEL-MES:	Un entero entre 1 y 31
MES:	Un entero entre 1 y 12
DIA-DE-LA-SEMANA:	Un entero entre 0 y 7 donde 0 y 7 representan al día domingo

Cuando no se desea indicar un valor específico, se deberá utilizar el asterisco:

```
0 0 5 3 * tar -czf /srv/backup/eugenia.tar.gz /home/eugenia
```

Lo anterior puede entenderse como:

Minuto:	0
Hora:	0
Día del mes:	5
Mes:	3 (marzo)
Día de la semana:	* (sin especificar)
Tarea:	tar -czf /srv/backup/eugenia.tar.gz /home/eugenia

Es decir que todos los 5 de marzo a las 0 horas, se hará una copia de respaldo de la *home* de mi usuario.

Claro que en vez de la ejecución de un comando, podría tratarse de la ejecución de un *script*. En ese caso, el *script* debe cuidar de **indicar en la primera línea, el binario con el cual debe ejecutarse**. Por ejemplo:

```
BASH:
#!/bin/bash

PERL:
#!/usr/bin/perl

PHP:
#!/usr/bin/php

PYTHON:
#!/usr/bin/python
```

Y también, debe tener **permisos de ejecución**:

```
chmod +x /ruta/al/script
```

Para **agregar tareas** al archivo crontab, se debe ejecutar el siguiente comando:

```
crontab -u usuario -e

Por ejemplo, para agregar tareas al crontab del usuario juan:
root@host:~# crontab -u juan -e
```

El parámetro `-u` será opcional si lo que se desea es agregar una tarea al crontab del usuario que está ejecutando dicho comando:

```
eugenia@hots:~$ crontab -e
Agregaré una tarea al crontab del usuario eugenia
```

Si no tienes un editor de texto por defecto, la primera vez que ejecutes `crontab -e` te pedirá que elijas el editor predeterminado.

Pero aquí, no termina todo. Cron, también permite especificar intervalos mucho más complejos y específicos y para ello, dispone de los siguientes operadores:

```
/ Salta la unidad especificada
```

```
0 */6 * * * rm -r /srv/www/myapp/temp/*  
Elimina todos los archivos del directorio /srv/www/myapp/temp/ cada 6 horas
```

```
*/10 * * * * rm -r /srv/www/myapp/temp/*  
Elimina todos los archivos del mismo directorio pero lo hace cada 10 minutos
```

, **La coma, permite especificar una lista precisa**

```
0 0 1,5,15 * * rm -r /srv/www/myapp/temp/*  
Elimina todos los archivos del mismo directorio los días 1, 5 y 15 de cada mes a las 0 horas
```

```
0 2 * 3,9 * rm -r /srv/www/myapp/temp/*  
Elimina todos los archivos del mismo directorio, todos los días a las 2:00 a.m. En los meses de marzo y septiembre
```

- **El guión medio, permite especificar un rango**

```
45 6 10-15 * * rm -r /srv/www/myapp/temp/*  
Elimina todos los archivos del mismo directorio entre los días 1 y 5 de cada mes a las 6:45 a.m.
```

```
0 0 * 9-12 * rm -r /srv/www/myapp/temp/*  
Elimina todos los archivos del mismo directorio, todos los días a las 0 horas entre los meses de septiembre y diciembre
```

Cron no permite ejecutar de forma directa, aplicaciones a través de entorno gráfico. Por ejemplo, si todos los días a las 17:00 hs. quisieras ejecutar el navegador Firefox, la siguiente instrucción, **NO funcionará**:

```
0 17 * * * firefox # No funcionará
```

Para que funcione, se debe agregar la instrucción **DISPLAY=":0"** precediendo al comando, como se muestra a continuación:

```
0 17 * * * DISPLAY=":0" firefox
```



APESOL

Asociación Peruana de Software Libre

<http://www.apesol.org.pe>