

# Como crear tareas CRON para Ruby mediante un Script

En esta guía explicamos un truco para configurar correctamente un comando Ruby para que se ejecute como tarea Cron.

Ejecutar un comando de Ruby como tarea CRON suele dar problemas, ya que el entorno del CRON no es el mismo que el de la shell. Por ejemplo, un comando concreto que ejecutas desde la línea de comandos funciona, pero si lo configuras como una tarea CRON lo más probable es que no funcione.

Y no funciona porque al cron le falta cierta información sobre el entorno correcto para que ese comando funcione. Concretamente desconoce el valor de las variables de entorno GEM\_PATH y GEM\_HOME, necesarias para ejecutar la mayoría de comandos Ruby/Rails.

## Como solucionar el problema

La forma más sencilla de pasarle esta información al cron es creando un script shell donde se definirá el entorno correcto y desde donde se ejecutará nuestro comando.

Por ejemplo, creamos el siguiente script al que llamaremos *cron.sh* y lo ubicaremos en la raíz de nuestra cuenta de hosting (es decir, en */home/USUARIO/cron.sh* ) :

```
#!/bin/sh
export GEM_PATH="/home/USUARIO/ruby/gems"
export GEM_HOME="/home/USUARIO/ruby/gems"
/home/USUARIO/ruby/bin/comando
exit 0
```

Para que este script se adapte a nuestras necesidades tenemos que modificar las líneas 2, 3 y 4. En las líneas 2 y 3 tenemos que reemplazar USUARIO por el nombre de usuario real de nuestra cuenta de hosting. En la línea 4 tenemos que escribir la ruta completa al comando que nos interesa ejecutar.

Una vez terminado el script tenemos que darle permisos de ejecución:

```
chmod 755 /home/USUARIO/cron.sh
```

Por último, crearemos una tarea CRON en la cual ejecutaremos el script *cron.sh* recién creado.