

## INTEGRACIÓN CONTINUA

## **RESOLUCIÓN DE EJERCICIOS**

A continuación encontrará el desarrollo de los ejercicios que resolvió en la semana. Contraste las respuestas entregadas por el docente con las desarrolladas por usted. En caso que no coincidan, y persistan dudas, le invitamos a repasar los contenidos y/ o consultar con su profesor.

### 1.1 construir un disparador que se automatice a cada 5 minutos

Paso 1 ingresamos a nuestro servidor de integración continua Jenkins Creamos una nueva tarea de estilo libre.





## Enter an item name

» This field cannot be empty, please enter a valid name



#### Crear un proyecto de estilo libre

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.



#### Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



#### Crear un proyecto multi-configuración

Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en multiples entornos, ejecutar sobre plataformas concretas, etc.

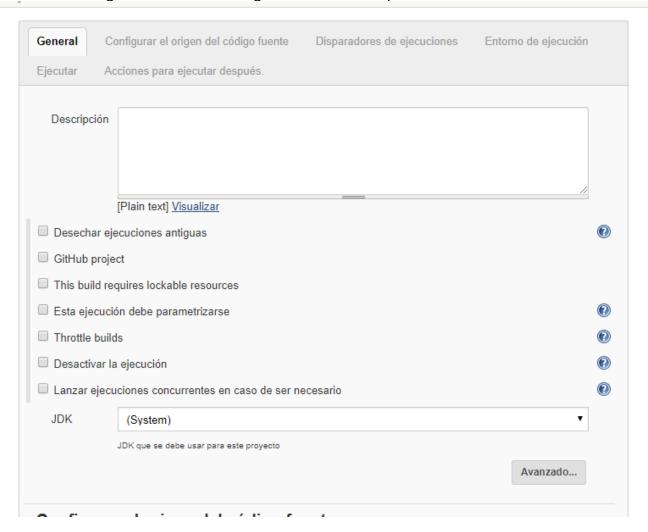


#### Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



## Paso 2: Nos dirigimos a General e ingresamos la descripción de nuestra tarea





Después presionamos la opción de disparadores de Ejecución y seleccionamos ejecución periódicamente

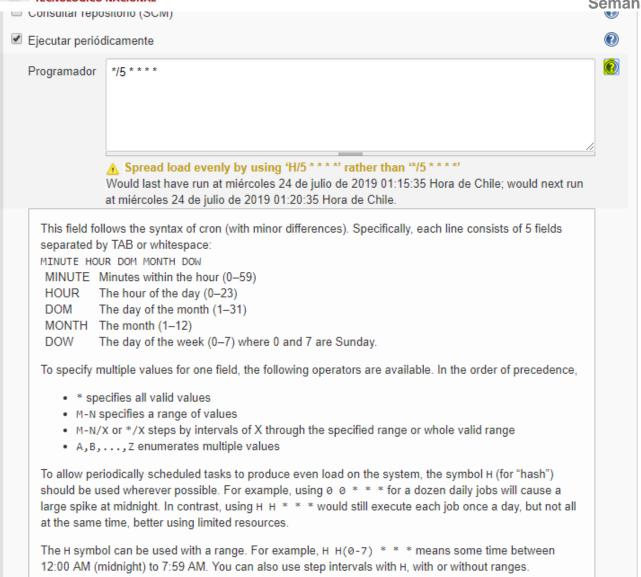




**Paso 3:** Ingresamos en le programador la secuencia de programador corn ingresando el comando \*/5 \* \* \* \* el carácter / sirve para especificar intervalos de tiempo, como en este caso nos pide que esta tarea se ejecute cada 5 minutos Jenkins también nos provee ayuda sobre como ingresar los comandos presionando el símbolo de interrogación







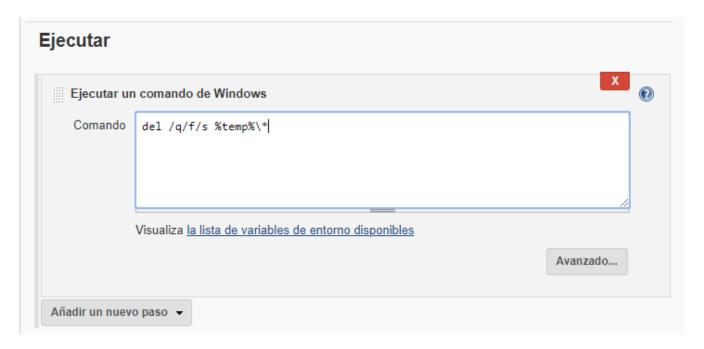
The H symbol can be thought of as a random value over a range, but it actually is a hash of the job

name, not a random function, so that the value remains stable for any given project.



**Paso 4:** luego nos dirigimos a entorno de ejecución y seleccionamos la opción de ejecución de ejecutar un comando de Windows en el cual realizara una limpieza de archivos temporales de nuestro sistema operativo, esto también se puede aplicar a la compilación de proyectos en Maven utilizando los Goas correspondientes.

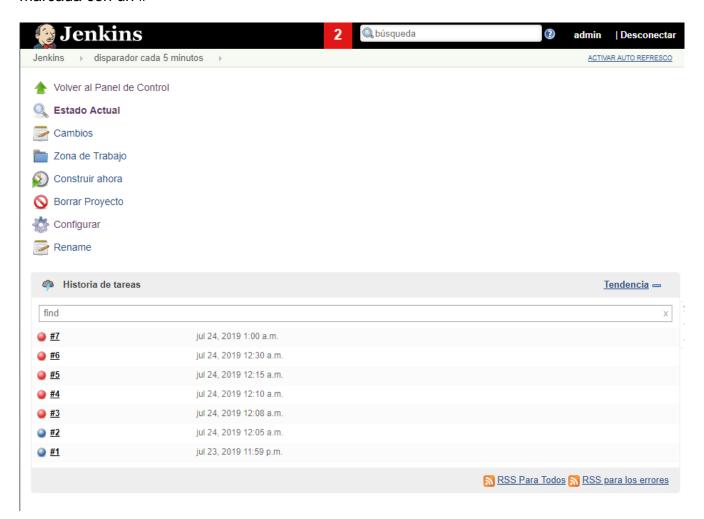






Luego presionamos en guardar, dependiendo de la hora que programemos nuestra tarea se nos ejecutara de forma automática.

Y para ver el resultado de nuestra tarea nos dirijimos a la tarea y dentro de ella nos aparecerá las veces que se ha ejecutado nuestra tarea, presionamos la ultima ejecución marcada con un #





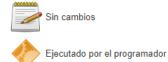
Dentro de nuestra tarea veremos el panel de contro en que nos aparece status, Cambios, Console Output etc. Presionaremos la opción de console output





Comenzó hace 29 Min Tardó <u>0,53 Seg</u>







Una vez presionado la opción de Console Output nos aparecerá el resultado de la tarea a ejecutar



# Salida de consola

Ejecutado por el programador

Running as SYSTEM

Ejecutando.en el espacio de trabajo C:\Users\Iplacex-PC\.jenkins\workspace\disparador cada 5 minutos

[disparador cada 5 minutos] \$ cmd /c call C:\Users\IPLACE~1\AppData\Local\Temp\jenkins3062725176890902555.bat

C:\Users\Iplacex-PC\.jenkins\workspace\disparador cada 5 minutos>del /q/f/s C:\Users\IPLACE~1\AppData\Local\Temp\\*

Archivo eliminado: C:\Users\IPLACE~1\AppData\Local\Temp\aria-debug-16380.log

C:\Users\IPLACE~1\AppData\Local\Temp\aria-debug-16928.log

El proceso no tiene acceso al archivo porque est siendo utilizado por otro proceso.

Archivo eliminado: C:\Users\IPLACE~1\AppData\Local\Temp\jenkins3062725176890902555.bat

 ${\tt C:\Users\IPLACE\sim1\AppData\Local\Temp\tmpzbwonl}$ 

El proceso no tiene acceso al archivo porque est siendo utilizado por otro proceso.

C:\Users\IPLACE~1\AppData\Local\Temp\winstone8033491694847124438.jar

El proceso no tiene acceso al archivo porque est siendo utilizado por otro proceso.

 ${\tt C:\Users\IPLACE$^1\AppData\Local\Temp\hsperfdata\_Iplacex-PC\18420}$ 

Acceso denegado.

C:\Users\IPLACE~1\AppData\Local\Temp\jna-332627654\jna7782352612748368467.dll

Acceso denegado.

C:\Users\IPLACE~1\AppData\Local\Temp\\_MEI178562\bz2.pyd

Acceso denegado.

C:\Users\IPLACE~1\AppData\Local\Temp\\_MEI178562\cello.pyd

Acceso denegado.

C:\Users\IPLACE~1\AppData\Local\Temp\\_MEI178562\common.time34.pyd

Acceso denegado.



## 1.2 Programar Jobs con los siguientes horarios:

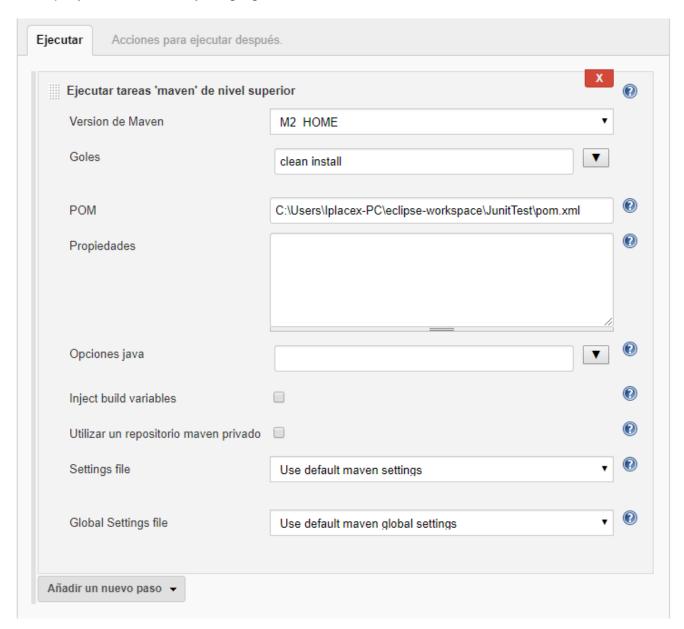
**Paso 1:** Creamos una nueva tarea en el cual se tiene que ejecutar de forma diaria repitiendo los pasos mencionados anteriormente.

Luego en el programador seleccionamos que se ejecute a las 8:00 am





Paso 2: nos dirigimos al panel de ejecución y en esta oportunidad realizaremos la instalación de un proyecto en Maven y luego guardar.





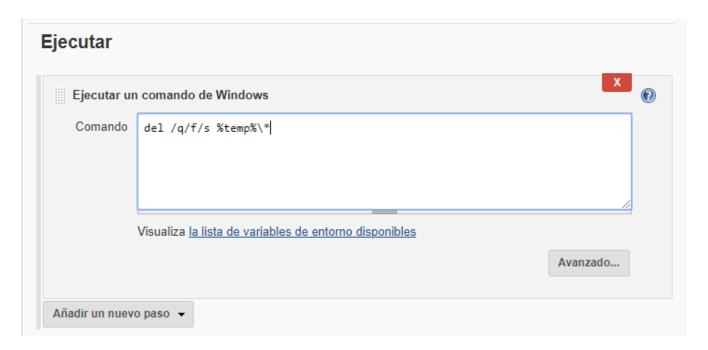
## 1.3 Ejecución de tarea cada 30 minutos

Creamos una nueva tarea en el cual realizaremos la ejecución de limpieza de archivos temporales cada 30 minutos en este paso podemos tomar como referencia el primer punto en cual automatizamos nuestra tarea que se ejecuta cada 5 minutos.





En la sección de ejecutar seleccionamos la opción de ejecutar comando Windows y luego presionamos en guardar y esperamos el tiempo correspondiente a que se ejeucten las tareas, puede utilizar su PC ejecutando Jenkins y notaran que las tareas se ejeuctan de forma automática.







Configurar Rename



Historia de tareas		<u>Tendencia</u> ⇔
find		x 1
#8	jul 24, 2019 1:30 a.m.	
<b>●</b> #7	jul 24, 2019 1:00 a.m.	
<b>◎</b> #6	jul 24, 2019 12:30 a.m.	
<b>●</b> #5	jul 24, 2019 12:15 a.m.	
<b>◎</b> #4	jul 24, 2019 12:10 a.m.	
<b>●</b> #3	jul 24, 2019 12:08 a.m.	
#2	jul 24, 2019 12:05 a.m.	
#1	jul 23, 2019 11:59 p.m.	
		RSS Para Todos RSS para los errores



#### **RESPUESTAS ESPERADAS**

1. ¿Cuál es la importancia de crear procesos automatizados para realizar trabajos en Jenkins?

La mayoría de las empresas tienen Cron Jobs para ejecutar las tareas más diversas y, a medida que aumenta el número de tareas, la capacidad de mantenimiento disminuye exponencialmente. Esto sucede porque generalmente la gente crea sus propios Cron Jobs manualmente, sin una herramienta para administrarlos. Entonces, tener muchos Cron Jobs distribuidos a través de máquinas virtuales no es tan fácil, ya aparecen problemas como: ΕI monitoreo. visibilidad. aue. documentación y ejecución de los Cron Jobs. Entonces en Jenkins, automatizamos los procesos para recordarle qué y cuándo tiene que realizar sus tareas. Cuando se trata de software, utilidades, servicios y otras tareas, necesitamos programar algunos trabajos repetitivos para activar automáticamente algunos comandos preespecificados, podemos programar cualquier tarea o trabajo para que se ejecute después de horas fijas o periódicamente durante el día. , la cantidad de veces en un mes, en días o fechas específicas, etc. Esto es lo que el CRON JOBS desempeña de manera más adecuada. Se trata de un programador de tareas basado en el tiempo que se utiliza para casi cada momento de la programación. Por ejemplo, para descargar los reposts diariamente a la 1:00 AM, para ejecutar los comandos DB y reiniciar los servidores cada vez que las máquinas se reinician, para ejecutar algunos scripts de UI o de funcionalidad durante toda la noche y así sucesivamente.