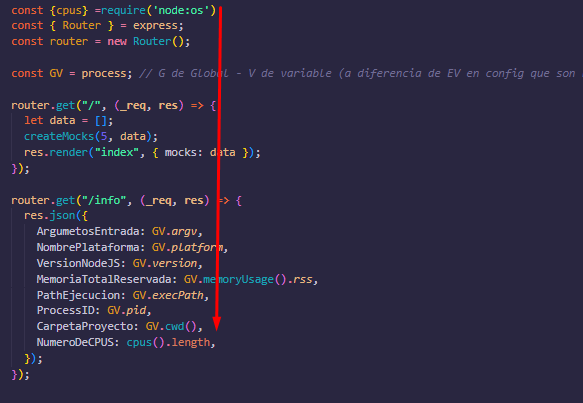
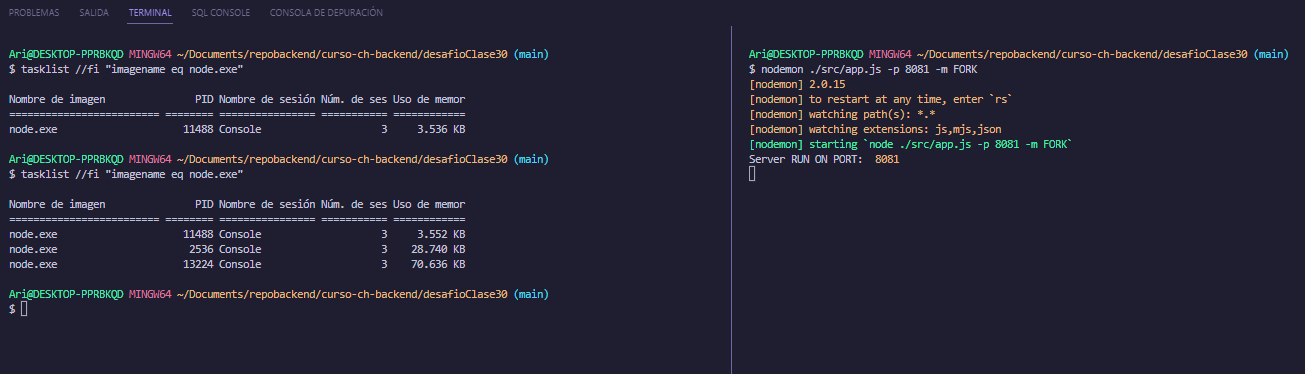
RESPUESTA DE LAS CONSIGNAS:

1. **INFORMACIÓN DE NROS DE CPU**

Se agregó en la ruta /info, con el método cpus del módulo OS, el número de procesadores presentes en la máquina que despliega la app. En mi caso son 4.



1. **NUMEROS DE PROCESOS**



**MODO FORK**

1. (CONSOLA IZQ) Se chequea cuantos procesos de node se están corriendo antes de iniciar con el comando:

tasklist //fi "imagename eq node.exe"

Esto devuelve 1 proceso

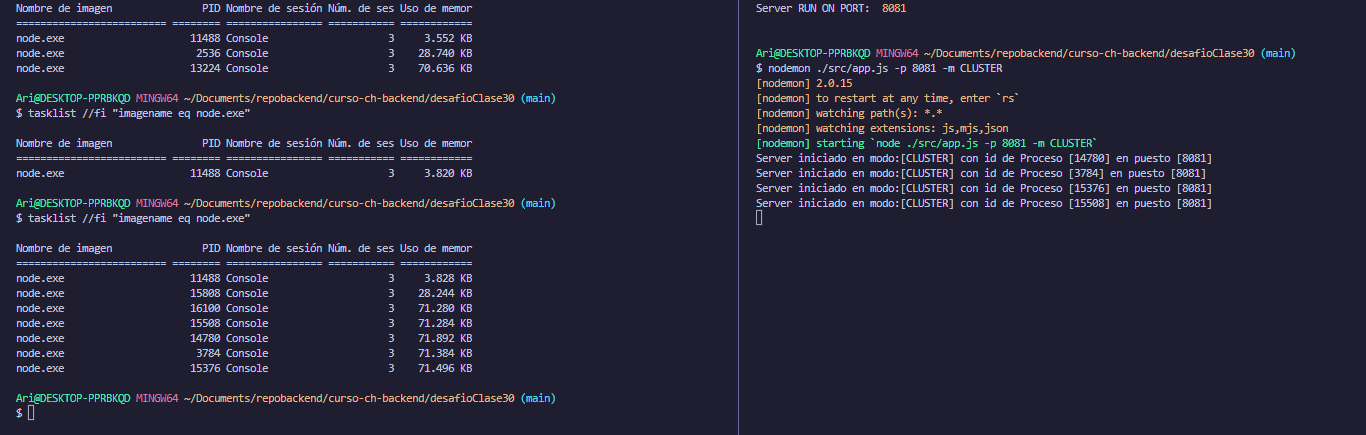
1. (CONSOLA DER) Se inicia el servidor en modo FORK con el comando:

Nodemon ./src/app.js -p 8081 -m FORK

1. (CONSOLA IZQ) Repite acción (1):

Esto devuelve 3 procesos

NOTA:Para iniciar el control como esta en gitbash, se usa la foobar //



**MODO CLUSTER**

1. (CONSOLA DER) Se cierra la consola que lleva iniciado el modo FORK con el comando:

c^ o ctrl+c

1. (CONSOLA IZQ) Se chequea cuantos procesos de node se están corriendo antes de iniciar con el comando:

tasklist //fi "imagename eq node.exe"

Esto devuelve 1 proceso

1. (CONSOLA DER) Se inicia el servidor en modo CLUSTER con el comando:

Nodemon ./src/app.js -p 8081 -m CLUSTER

1. CONSOLA IZQ) Repite acción (1):

Esto devuelve 7 procesos

NOTA:

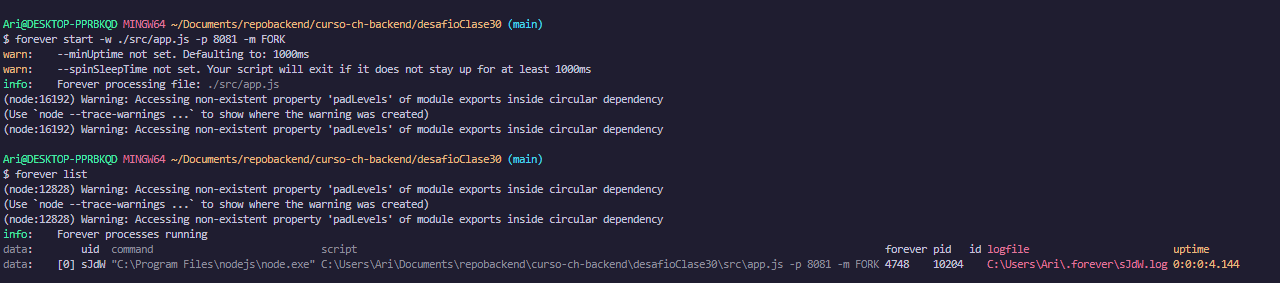
* el de FORK, ya se encuentra puesto por default con yargs
* el IF que divide a las dos MODOS, posibles: tiene adrede armado un ELSE IF para hacer un código más gráfico y entendible

1. **FOREVER**

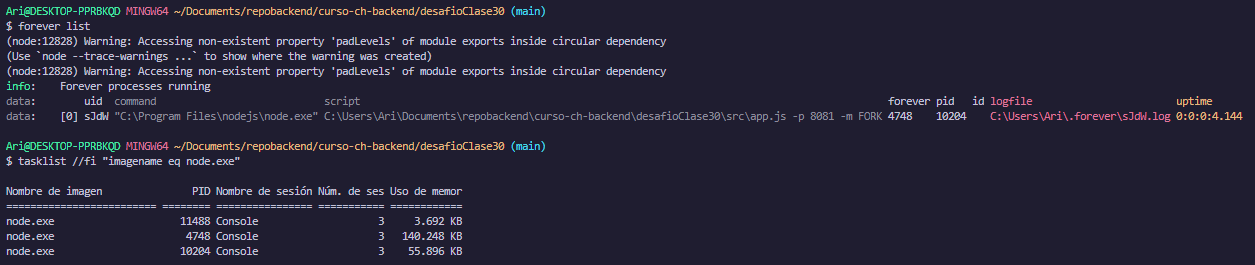
**MODO FORK**

Comando para iniciar forever start -w ./src/app.js -p 8081 -m FORK

**NOTA: se usa start -w, porque según la documentación se usa esa flag para que se active el modo escucha**



El conteo de procesos da 3



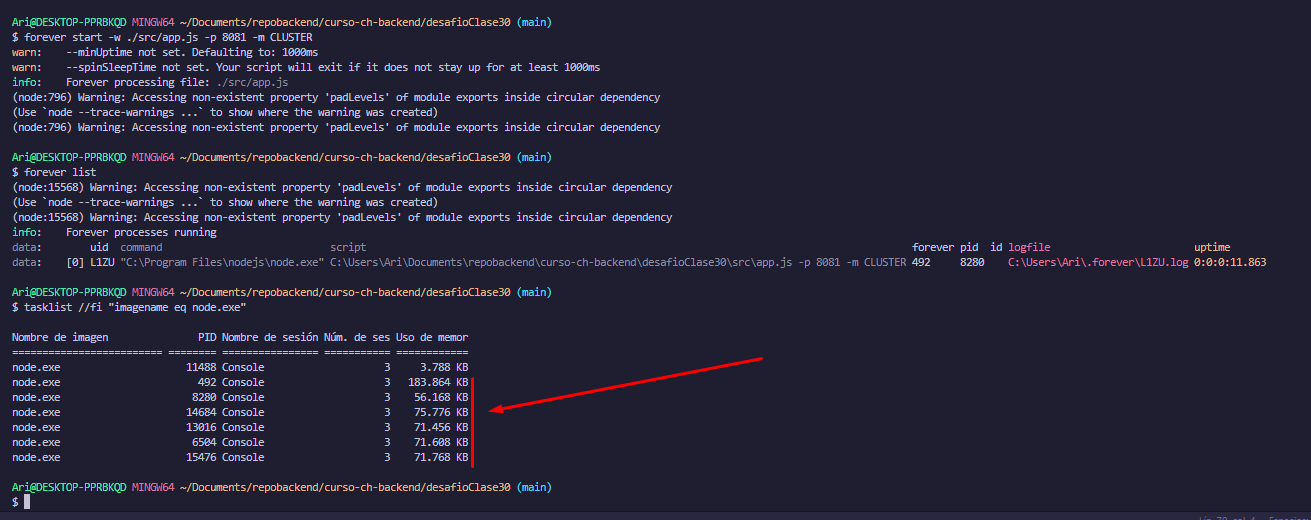
**MODO CLUSTER**

Se inicia el modo cluster con el comando: forever start -w ./src/app.js -p 8081 -m CLUSTER

Se hace solicita un comando: “forever list” para ver cuantas instancias del mismo server hay

Luego se consulta cuantos procesos hay con el comando : tasklist //fi "imagename eq node.exe"

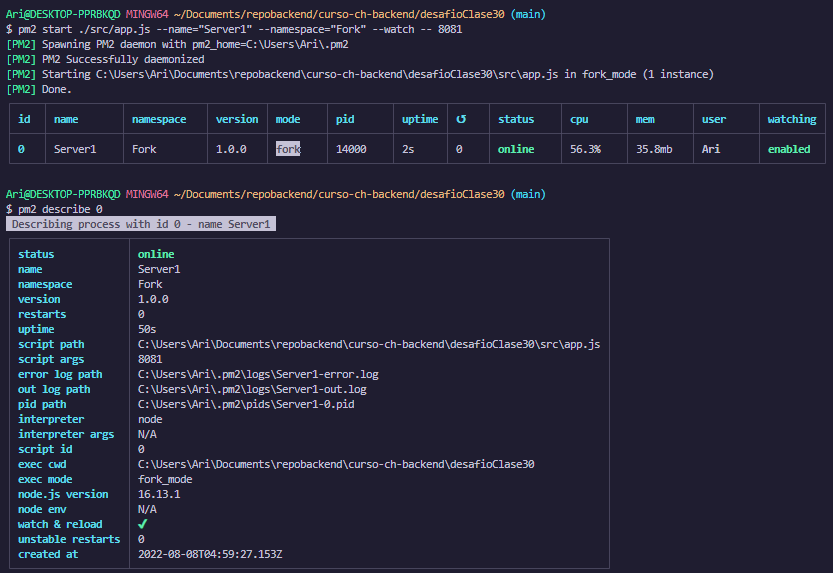
Se devuelven 7 procesos



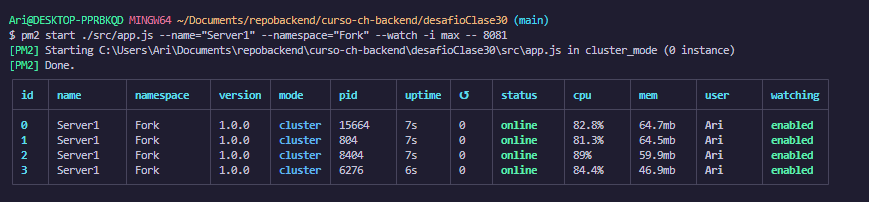
1. **PM2**

Con PM2 no es necesario ingresar un argumento por -m para entrar en modo cluster. Porque PM2, ya tiene esa opción entre sus funcionalidades,

MODO FORK: pm2 start ./src/app.js --name="Server1" --namespace="Fork" --watch -- 8081

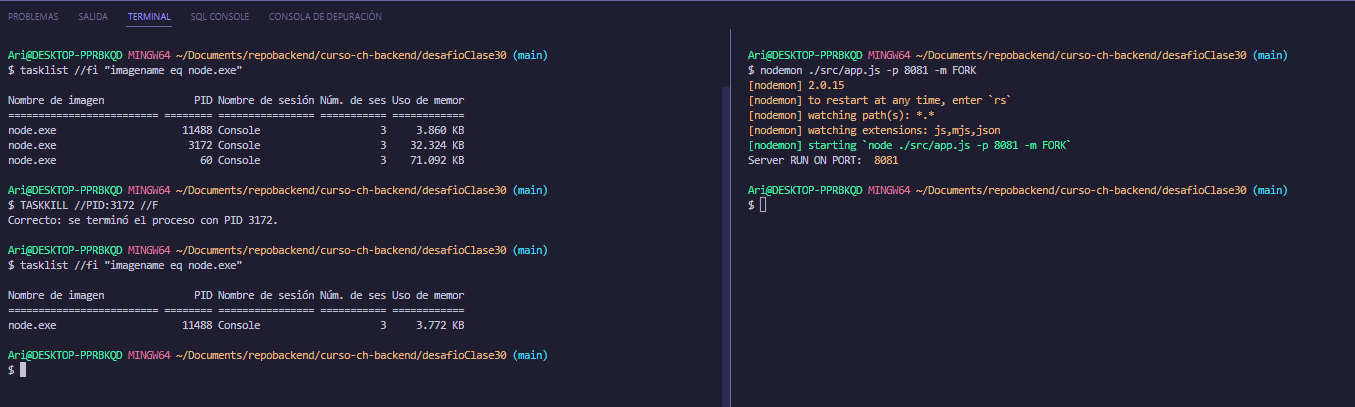


MODO CLUSTER: pm2 start ./src/app.js --name="Server2"--namespace="Cluster" --watch -i -max -- 8081

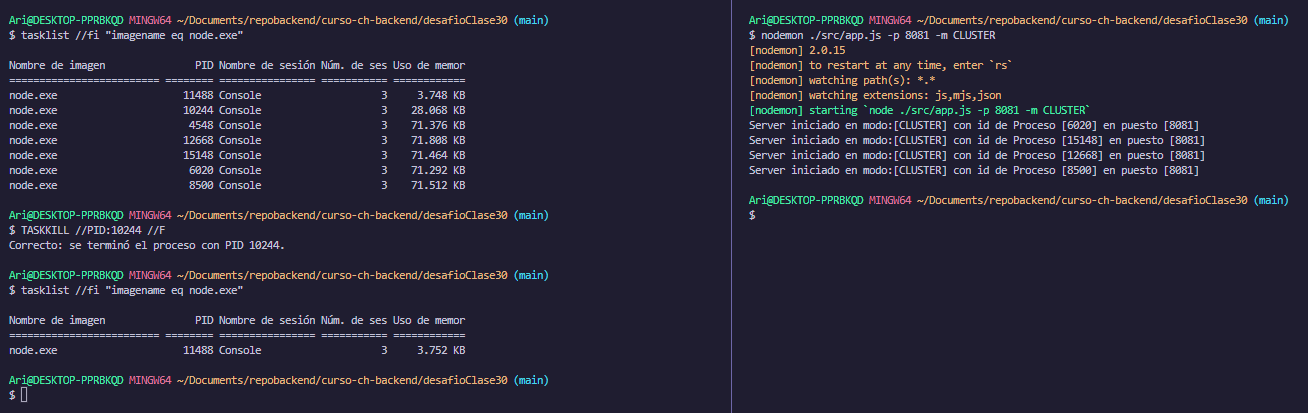


1. **Eliminar procesos:**

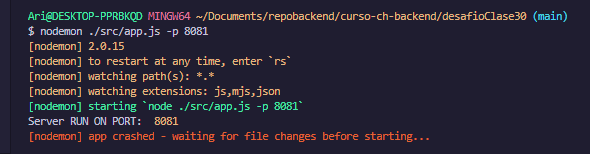
**NODEMON**



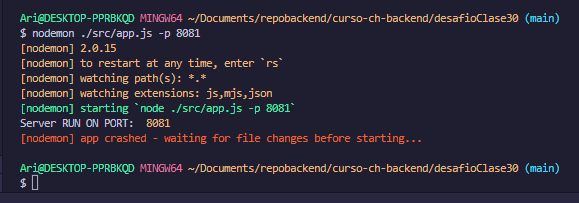
**MODO FORK**. Se inicia (en consola de Derecha) la app.js en modo FORK. Se Ejecuta (en consola de Izquierda) el tasklist y devuelve 3 procesos. Se usa el comando TASKKILL //PID:NUMERO //F (en consola de izq, se recuerda que es bash, y por eso es necesario usar la foobar).Se observa en consola derecha que se vuelve a habilitar para ingresar comandos y al consultar con tasklist (en consola izq) se encuentra con un solo proceso.



Por otro lado, en caso de no matar al primario arroja un CRASHED:



Luego si se mata al primario, habilita nuevamente los comandos:

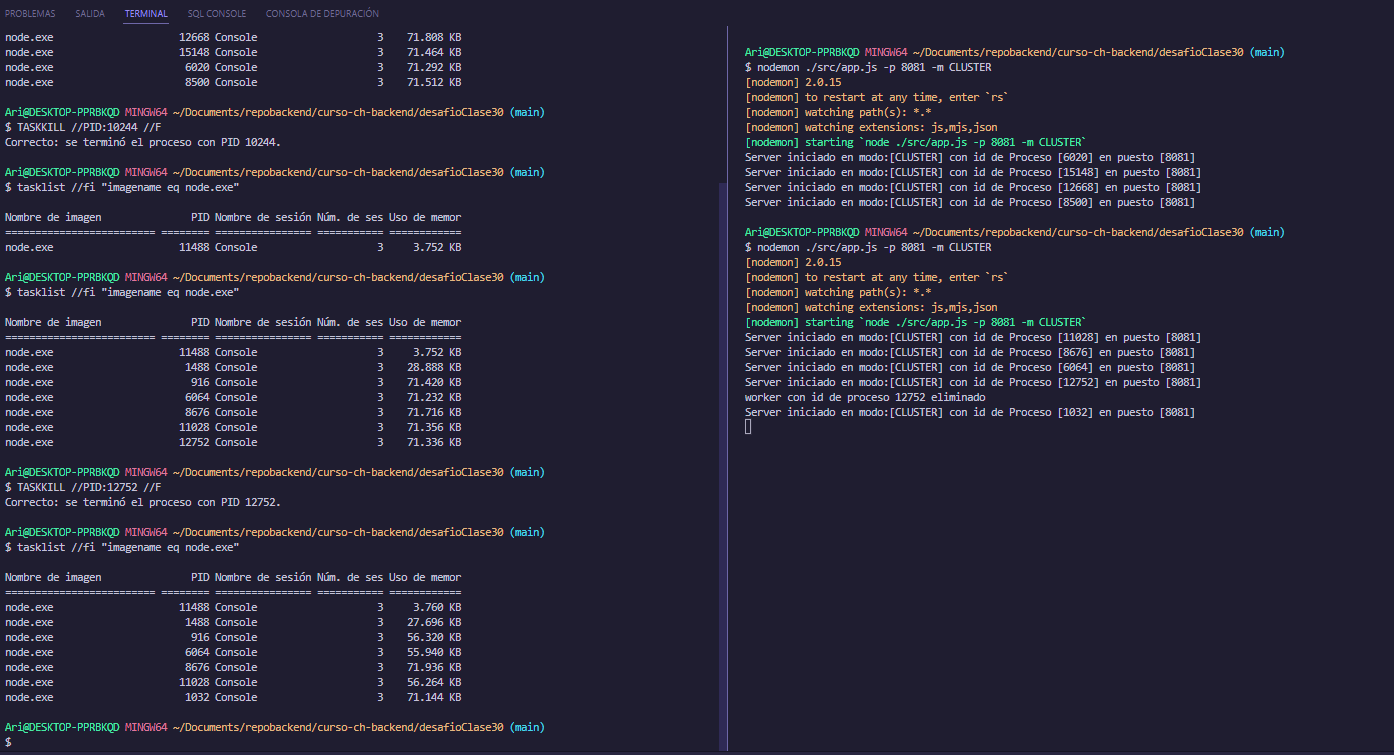


NOTA: Diferencia con forever que en estos casos al no matar al Primario reinicia al worker.

**MODO CLUSTER.**

ELIMINAR EL CLUSTER.ISPRIMARY

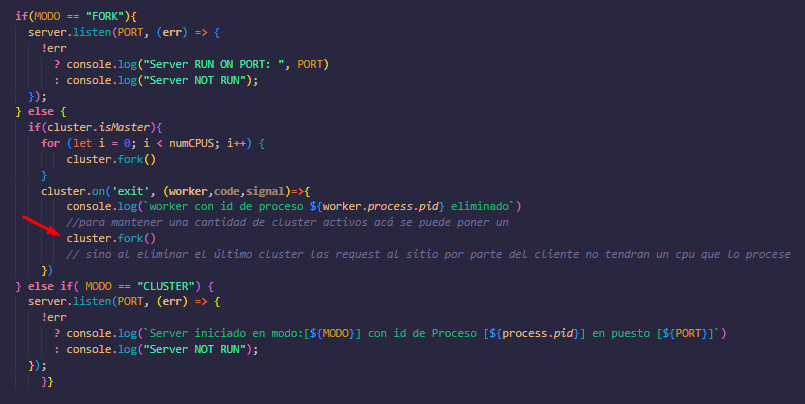
Se inicia (en consola de Derecha) la app.js en modo CLUSTER. Se Ejecuta (en consola de Izquierda) el tasklist y devuelve 7 procesos. Se usa el comando TASKKILL //PID:NUMERO //F (en consola de izq, se recuerda que es bash, y por eso es necesario usar la foobar).Se observa en consola derecha que se vuelve a habilitar para ingresar comandos y al consultar con tasklist (en consola izq) se encuentra con un solo proceso.



ELIMINAR EL CLUSTER.ISWORKER

Se inicia (en consola de Derecha) la app.js en modo CLUSTER. Se Ejecuta (en consola de Izquierda) el tasklist y devuelve 7 procesos. Se usa el comando TASKKILL //PID:NUMERO //F (en consola de izq, se recuerda que es bash, y por eso es necesario usar la foobar).Se observa en consola derecha que se despide por consola un mensaje que indica que el proceso con id 12752 fue eliminado, y de inmediato se genera un nuevo worker con pid 1032, y al consultar con tasklist (en consola izq) se encuentra con 7 procesos nuevamente.

NOTA: QUE SURJA UN NUEVO CLUSTER SE DEBE A LA SIGUIENTE PIESA DE CÓDIGO:



**FOREVER:**

**MODO FORK**

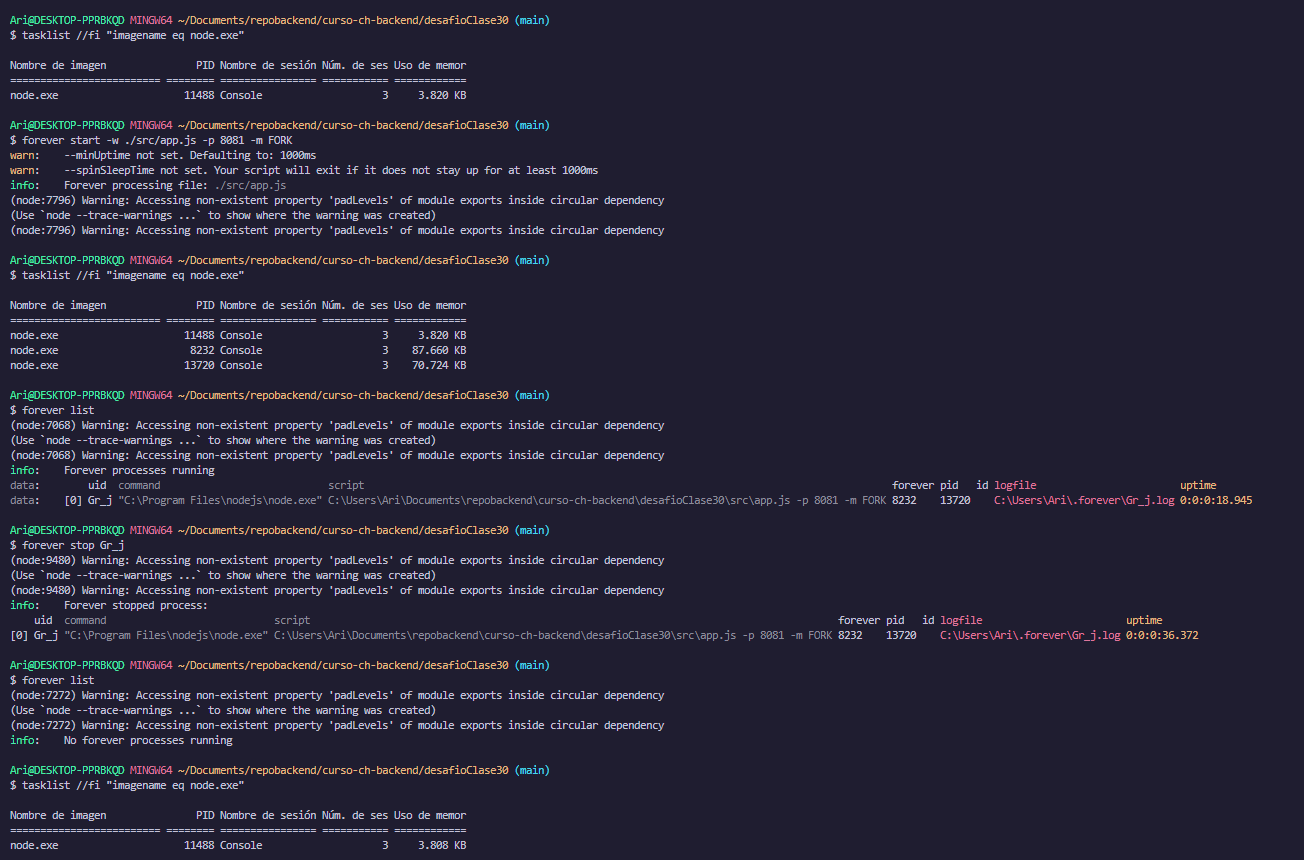
Se inicia con un tasklist //fi “imagename eq node.exe”, para ver cuantos procesos de node hay abiertos. Figura uno

Se ejecuta y da inicio a una instancia del servidor en modo FORK.

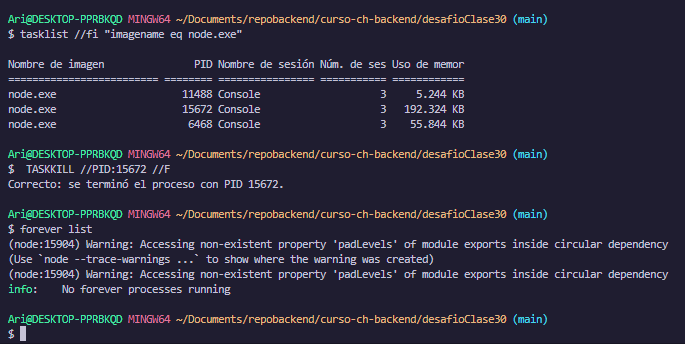
Se hace otro tasklist se ven 3 procesos

Se hace un forever list, para visualizar el servidor iniciado.

Se hace un forever stop por uid. Se verifica que el proceso se cierra con otro

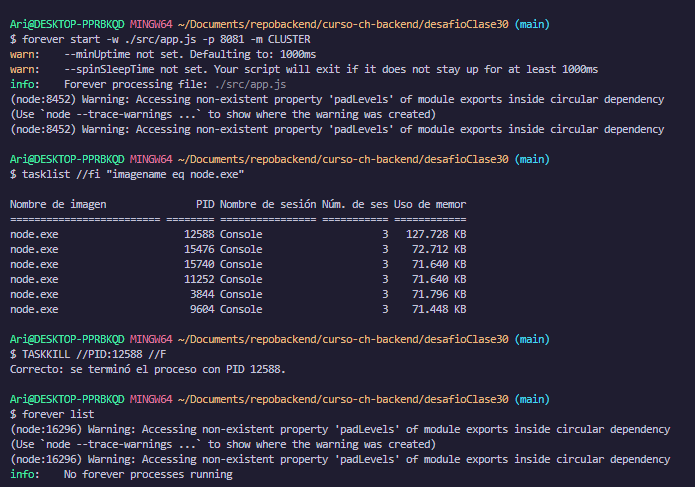


Funciona del mismo modo si se usa un taskkill (siempre que se apunte al primario, en caso de apuntar al worker se reinicia la instancia y se mantiene el proceso primario del server)

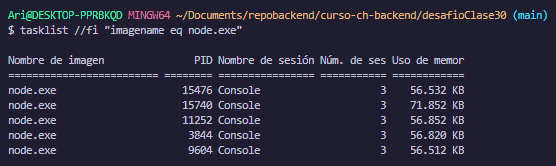


**MODO CLUSTER**

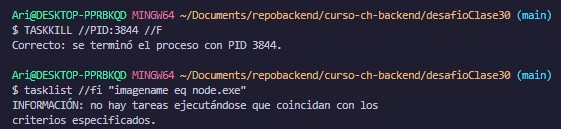
TASKKILL isPRIMARY



Forever no reconoce que quede un server corriendo. Pero node, tiene procesos que quedan pendientes



Pero al matar a uno de ellos se cierran todos

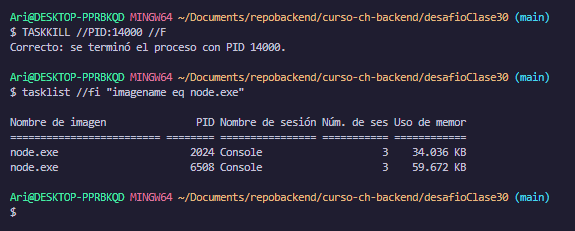


**PM2:**

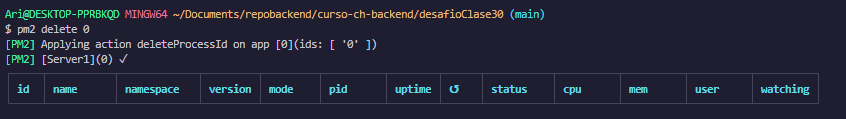
**NOTA: En PM2, los taskkill no sirven, PM2 entiende que es un cierre inesperado y lo reinicia.**

**MODO FORK**

Cuando se mata al proceso principal, se reinicia uno nuevo.

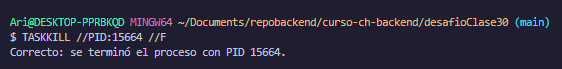


Para matar al proceso, se usa el comando pm2 delete id:

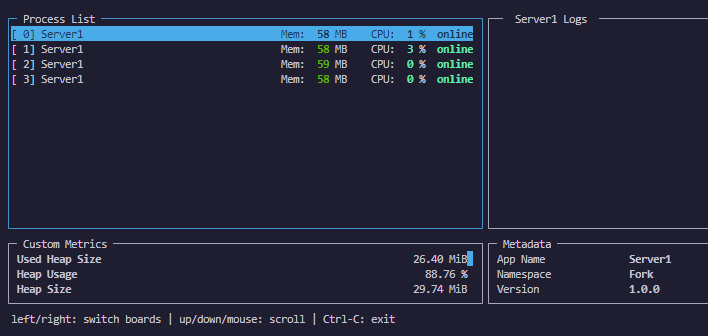


**MODO CLUSTER**

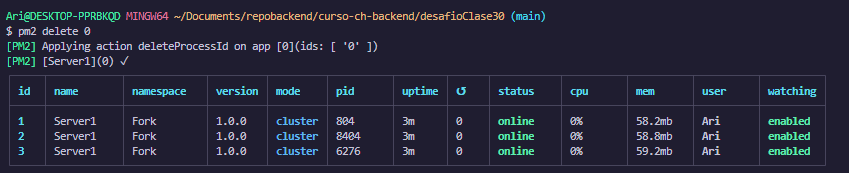
Cuando se mata un proceso, aún siendo el primario pm2, lo reemplaza con otro.



Resultado de matar un proceso(se siguen manteniendo los 4):



Para elminiar se puede hacer de a uno con pm2 delete id



o pm2 delete

