

**Computación Tolerante a Fallas**

**Sección D06**

# **Servicios y Estatus**



**Fernández Venegas David Guadalupe**

**Código: 216437352**

**Profesor: López Franco Michel Emanuel**

---

## Introducción

Existen dos tipos de procesos que puede ejecutar una computadora:

- **Aplicaciones:** aquellos procesos que se ejecutan generalmente en primer plano y responden a la interacción con el usuario. Son los programas que todo el mundo conoce. Pueden presentar interfaz gráfica o no. Si no tienen interfaz gráfica se ejecutan en el terminal. una aplicación solamente podrá ejecutarse una vez que el usuario ha iniciado sesión. Esto, a su vez, significa que una aplicación tiene los permisos del usuario que la está ejecutando.
- **Servicios:** tipo de procesos que se ejecutan de forma ajena al usuario. Los servicios no están vinculados al usuario, por lo que se pueden ejecutar antes de que se haya iniciado sesión. Al ser independientes del usuario, los servicios no pretenden interactuar con el usuario, sino realizar una función en segundo plano. No tienen interfaz gráfica y no están vinculados a un terminal.

## Proclocker: Servicio de modo de estudio.

Como demostración, se ha realizado mediante python un programa el cual se instaló como un servicio. Dicho programa permite verificar el estatus de nuestras aplicaciones para poder actuar de alguna manera. En este caso, las aplicaciones seleccionadas previamente serán cerradas al ser detectadas por el programa.

Para esta actividad, se ha utilizado este servicio para bloquear las aplicaciones de Steam y EpicGamesLauncher, dos aplicaciones que funcionan para la distribución de juegos digitales. Esto con el propósito de crear un modo de estudio o de trabajo para de esta manera bloquear aquellos juegos instalados en la computadora por medio de estos servicios, y así poder tener una mejor concentración en nuestras tareas pendientes.

## Explicación del Código.

### Librerías utilizadas:

- **Sys:** permite acceder a la consola del sistema operativo.
- **Psutil:** permite tener acceso a información de procesos que corren así como del sistema.

### Funciones:

- Verifica que tengamos argumentos en la consola (nombre de las aplicaciones) seleccionados por el usuario.

```
def check_arguments():  
    if len(sys.argv) == 1:  
        print('Este programa no funciona sin argumentos')  
        sys.exit(0)
```

- Agrega a los argumentos el formato .exe

```
def get_targets():  
  
    targets = sys.argv[1:]  
  
    i = 0  
    while i < len(targets):  
  
        if not targets[i].endswith('.exe'):  
            targets[i] = targets[i] + '.exe'  
            i += 1  
  
    return targets
```

- Localiza de todos los procesos, los programas seleccionados para cerrarlos.

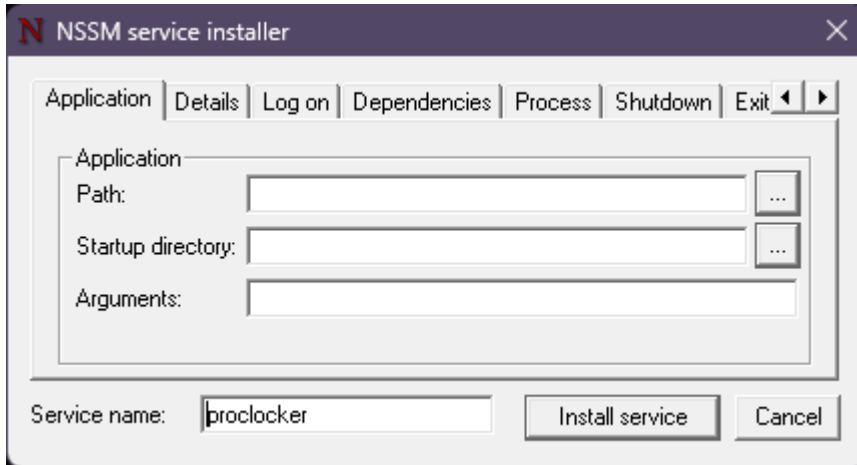
```
def lock(target):  
    for proc in psutil.process_iter():  
        if proc.name().lower() == target.lower():  
            proc.kill()
```

- Main: una vez verificado la entrada de las aplicaciones seleccionadas, pasaremos a un ciclo "infinito" en donde estaremos cerrando las aplicaciones seleccionadas repetidamente.

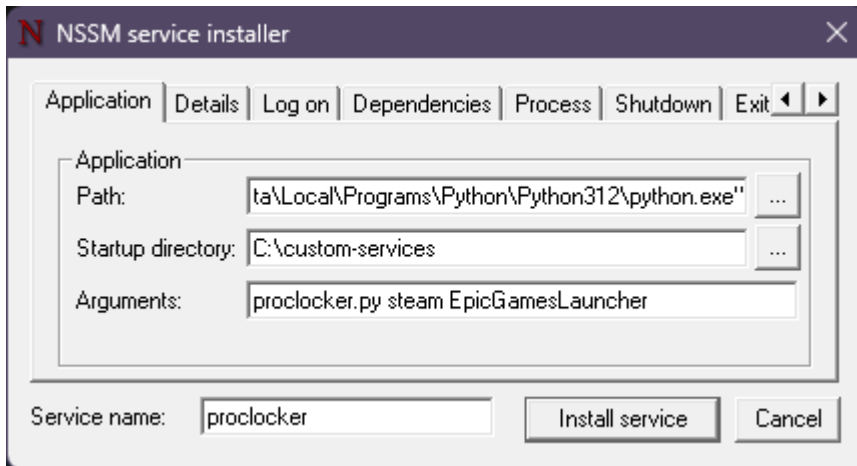
```
if __name__ == '__main__':  
  
    check_arguments()  
    targets = get_targets()  
  
    while True:  
        for target in targets:  
            lock(target)
```

## Instalación como servicio.

Una vez realizado nuestro código debemos descargar Non-Sucking Service Manager para después realizar su instalación por medio de CMD. Una vez realizada la instalación, se abrirá la siguiente ventana en donde podremos configurar nuestro servicio.



- Path: deberemos indicar la ruta completa en donde se encuentra nuestro interprete. En este caso la ruta del ejecutable de Python.
- Startup directory: deberemos indicar el directorio en donde se encuentra nuestro código.
- Arguments: deberemos indicar el nombre de nuestro programa además del nombre de las aplicaciones que deseamos bloquear.

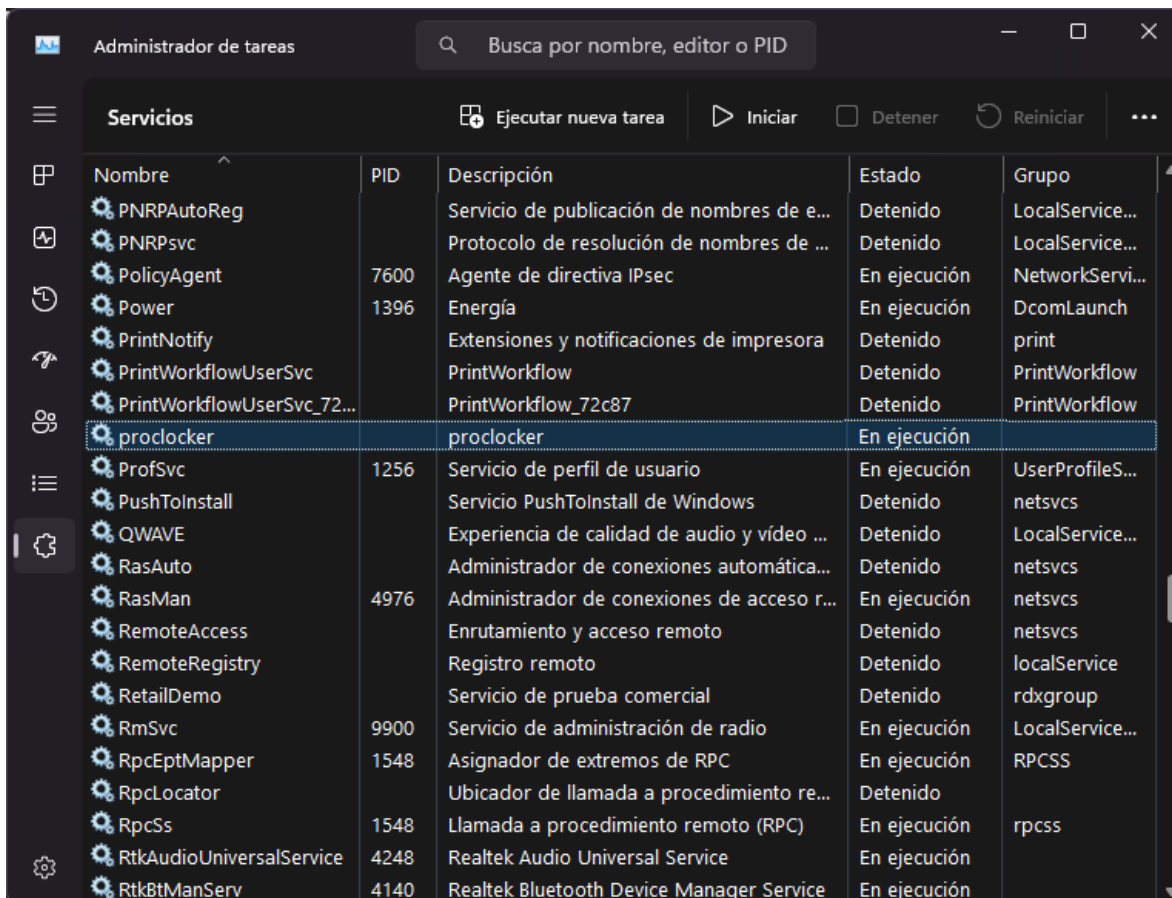


Una vez introducida toda la información necesaria, simplemente debemos hacer clic en instalar servicio y posteriormente activar el CMD mediante en comando:

```
nssm.exe start proclocker
```

Finalmente obtendremos el siguiente mensaje y podremos ver en ejecución nuestro servicio desde el administrador de tareas:

```
proclocker: START: La operación se completó correctamente.
```



The screenshot shows the Windows Task Manager 'Servicios' (Services) tab. The 'proclocker' service is highlighted in blue, indicating it is running. The table below represents the data visible in the screenshot.

Nombre	PID	Descripción	Estado	Grupo
PNRPAutoReg		Servicio de publicación de nombres de e...	Detenido	LocalService...
PNRPsvc		Protocolo de resolución de nombres de ...	Detenido	LocalService...
PolicyAgent	7600	Agente de directiva IPsec	En ejecución	NetworkServi...
Power	1396	Energía	En ejecución	DcomLaunch
PrintNotify		Extensiones y notificaciones de impresora	Detenido	print
PrintWorkflowUserSvc		PrintWorkflow	Detenido	PrintWorkflow
PrintWorkflowUserSvc_72...		PrintWorkflow_72c87	Detenido	PrintWorkflow
<b>proclocker</b>		<b>proclocker</b>	<b>En ejecución</b>	
ProfSvc	1256	Servicio de perfil de usuario	En ejecución	UserProfileS...
PushToInstall		Servicio PushToInstall de Windows	Detenido	netsvcs
QWAVE		Experiencia de calidad de audio y vídeo ...	Detenido	LocalService...
RasAuto		Administrador de conexiones automática...	Detenido	netsvcs
RasMan	4976	Administrador de conexiones de acceso r...	En ejecución	netsvcs
RemoteAccess		Enrutamiento y acceso remoto	Detenido	netsvcs
RemoteRegistry		Registro remoto	Detenido	localService
RetailDemo		Servicio de prueba comercial	Detenido	rdxgroup
RmSvc	9900	Servicio de administración de radio	En ejecución	LocalService...
RpcEptMapper	1548	Asignador de extremos de RPC	En ejecución	RPCSS
RpcLocator		Ubicador de llamada a procedimiento re...	Detenido	
RpcSs	1548	Llamada a procedimiento remoto (RPC)	En ejecución	rpcss
RtkAudioUniversalService	4248	Realtek Audio Universal Service	En ejecución	
RtkBtManServ	4140	Realtek Bluetooth Device Manager Service	En ejecución	

Con nuestro servicio activo se cerrarán las aplicaciones indica demás y además estas se volverán a cerrar cada vez que las intentemos abrir.

## Conclusión

El uso de servicios ofrece la ventaja de ejecutar procesos de manera independiente al usuario y en segundo plano, sin necesidad de que este haya iniciado sesión. Esto permite realizar funciones específicas sin interacción directa del usuario, como en el caso del servicio Proclocker utilizado para bloquear aplicaciones de distracción como Steam y EpicGamesLauncher. Esta capacidad de ejecución autónoma facilita la automatización de tareas, permitiendo no solo la aplicación de un modo estudio, si no una gran cantidad de posibilidades de servicios automáticos.