



## **Automatización de Tareas en Windows y Linux para la Administración de Bases de Datos y Manejo de Incidentes**

**MTI JESÚS E. ROMERO MORENO**

**DIVISIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN  
Y COMUNICACIÓN**

## Automatización de Tareas en Windows y Linux para la Administración de Bases de Datos y Manejo de Incidentes

La automatización de tareas es fundamental en la administración de sistemas y bases de datos, especialmente cuando se requiere una gestión eficiente, rápida y sin intervención manual constante. Herramientas como **schtasks** en Windows y **cron** en Linux permiten programar la ejecución automática de tareas, lo que optimiza el rendimiento y reduce la posibilidad de errores humanos. Estos sistemas de programación son esenciales para implementar soluciones de administración de bases de datos y respuesta a incidentes, ya que pueden ejecutarse sin necesidad de intervención constante, permitiendo una gestión más eficiente y confiable.

### Automatización en Windows con schtasks

En **Windows**, la herramienta **schtasks** es una forma robusta de automatizar la ejecución de scripts, aplicaciones o tareas del sistema. Al configurarla, es posible programar tareas que se ejecuten en intervalos específicos, como la creación de copias de seguridad de bases de datos, la ejecución de scripts de mantenimiento o el monitoreo del rendimiento del sistema.

Por ejemplo, podemos crear una tarea programada que ejecute un script de respaldo de una base de datos a las 2:00 AM todos los días:

**cmd**

```
schtasks /create /tn "BackupDB" /tr "C:\Scripts\backup_db.bat" /sc daily /st 02:00
```

Esto no solo garantiza que la base de datos esté respaldada regularmente, sino que también libera a los administradores de la necesidad de ejecutar manualmente estas tareas. Además, en el contexto de manejo de incidentes, se pueden automatizar procesos como la monitorización del sistema, alertas de errores, o la ejecución de procedimientos de recuperación ante fallos.

### Comando básico para crear una tarea programada:

**cmd**

```
schtasks /create /tn "MiTarea" /tr "C:\Ruta\al\script.bat" /sc daily /st 09:00
```

- /tn "MiTarea": Especifica el nombre de la tarea.
- /tr "C:\Ruta\al\script.bat": Define el programa o script que quieres ejecutar.
- /sc daily: Especifica que la tarea se ejecutará todos los días.
- /st 09:00: Indica que la tarea debe ejecutarse a las 9:00 AM.

### Otros parámetros útiles:

- /sc weekly para ejecutarse cada semana.
- /sc monthly para ejecutarse cada mes.

- /ru para definir el usuario bajo el cual se ejecutará la tarea.
- /f para forzar la creación de la tarea (si ya existe).

### Automatización en Linux con cron

En **Linux**, el **cron** es la herramienta estándar para la programación de tareas repetitivas. A través de su archivo de configuración crontab, se pueden definir tareas programadas que se ejecuten en función del tiempo, facilitando la automatización de tareas relacionadas con la administración de bases de datos, como la limpieza de registros antiguos, la ejecución de actualizaciones, o la verificación de la integridad de la base de datos.

Por ejemplo, podemos crear una tarea cron para verificar la integridad de la base de datos y enviar un informe por correo electrónico todos los lunes a las 3:00 AM:

**bash**

**0 3 \* \* 1 /ruta/al/script\_integridad\_db.sh**

Este tipo de automatización asegura que se detecten problemas potenciales con la base de datos antes de que afecten el rendimiento o la disponibilidad del sistema. Además, en situaciones de incidentes, cron puede ser utilizado para ejecutar procesos de remediación, como el reinicio de servicios de base de datos, la recopilación de registros o la ejecución de diagnósticos de sistema.

### Sintaxis del archivo crontab:

lua

\* \* \* \* \* /ruta/al/script

- - - - -

| | | | |

| | | | +--- Día de la semana (0 - 7) (domingo = 0 o 7)

| | | +----- Mes (1 - 12)

| | +----- Día del mes (1 - 31)

| +----- Hora (0 - 23)

+----- Minuto (0 - 59)

### Ejemplo de tarea cron:

Para ejecutar un script todos los días a las 3 AM:

**bash**

**0 3 \* \* \* /ruta/al/script.sh**

Otros ejemplos comunes de cron:

- Cada hora:

```
bash
```

```
0 * * * * /ruta/al/script.sh
```

- Cada 15 minutos:

```
bash
```

```
*/15 * * * * /ruta/al/script.sh
```

Puedes verificar tus cron jobs con el comando `crontab -l` y eliminar un cron job con `crontab -r`.

## Importancia de la Automatización para la Administración de Bases de Datos

La administración de bases de datos involucra tareas repetitivas y críticas que requieren ser gestionadas sin demora para mantener la integridad, disponibilidad y seguridad de los datos. Las herramientas **schtasks** y **cron** permiten a los administradores de bases de datos programar y automatizar tareas esenciales como:

- **Copia de seguridad regular:** Crear backups automáticos previene la pérdida de datos en caso de un desastre o incidente.
- **Mantenimiento periódico:** La ejecución de scripts de mantenimiento como la optimización de índices o la limpieza de registros antiguos ayuda a mantener el rendimiento del sistema.
- **Monitoreo constante:** Las tareas programadas permiten la supervisión continua de los sistemas, alertando sobre problemas antes de que se conviertan en incidentes críticos.

## Implementación de Acciones en Caso de Detectar un Incidente

Ambas herramientas, **schtasks** y **cron**, no solo facilitan la automatización de tareas de administración, sino que también son cruciales para implementar medidas de respuesta ante incidentes. Algunos ejemplos incluyen:

- **Notificación de incidentes:** Al detectar un fallo o comportamiento anómalo en la base de datos, se puede programar una tarea que envíe un correo electrónico o mensaje a los administradores para tomar acción inmediata.
- **Automatización de recuperación:** En caso de que un servicio de base de datos falle, se pueden configurar tareas programadas para reiniciar el servicio o ejecutar procedimientos de recuperación sin intervención manual.

- **Monitoreo proactivo:** Con herramientas como cron o schtasks, es posible ejecutar scripts que monitoricen constantemente los logs del sistema y la base de datos, generando alertas cuando se detectan errores específicos, como bloqueos de base de datos o caídas del servidor.

## Conclusión

La automatización de tareas mediante **schtasks** en Windows y **cron** en Linux representa una herramienta poderosa para la administración eficiente de bases de datos y la implementación de acciones automáticas en respuesta a incidentes. La capacidad de programar tareas críticas, como copias de seguridad, mantenimiento, o alertas de errores, no solo mejora la eficiencia operativa, sino que también incrementa la resiliencia y disponibilidad de los sistemas de base de datos. Al automatizar estos procesos, los administradores pueden centrarse en tareas estratégicas, dejando las tareas repetitivas y de monitoreo en manos de estas herramientas, lo que reduce riesgos y mejora la respuesta ante incidentes.

# Solicitud de Creación de Automatizaciones para Administración de Bases de Datos y Virtualización

## Objetivo:

El propósito de esta actividad es que cada estudiante desarrolle tres automatizaciones utilizando las opciones del shell que se han explicado en clase, con el objetivo de garantizar la ejecución de actividades sin la intervención directa del administrador del sistema. Las automatizaciones deben estar orientadas a dos áreas clave: **Administración de Bases de Datos** y **Virtualización**.

## Requisitos Generales:

1. **Automatización de Bases de Datos:** El estudiante deberá crear al menos una automatización relacionada con la administración de bases de datos. Esta podría incluir tareas como la creación de copias de seguridad, la optimización de índices o el monitoreo de la integridad de los datos.
2. **Automatización de Virtualización:** El estudiante deberá crear al menos una automatización relacionada con la gestión de entornos de virtualización, como la creación, destrucción o gestión de máquinas virtuales, así como la gestión de recursos.
3. **Permisos y Privilegios de Usuario:** Es esencial que, al programar estas actividades, se tomen en cuenta los permisos y privilegios de usuario necesarios para garantizar que las tareas se ejecuten correctamente sin comprometer la seguridad del sistema. Asegúrese de que las tareas automatizadas se ejecuten con el usuario adecuado o los permisos necesarios para evitar errores de ejecución o problemas de seguridad.
4. **Investigación y Pruebas:** Cada automatización debe ser investigada, probada y demostrada en un entorno real o simulado, validando su funcionalidad antes de presentarla. El estudiante debe documentar cómo realizó la investigación y las pruebas, explicando cómo verificó que la automatización funciona de manera correcta y confiable.
5. **Plataforma de Ejecución:** Los estudiantes pueden optar por utilizar **Windows** o **Linux** para realizar las automatizaciones, según sus preferencias y el entorno que estén utilizando.
6. **Trabajo Individual:** Esta actividad debe ser realizada de manera **individual**. No se permite la colaboración con otros compañeros del salón, y la solución entregada debe ser única para cada estudiante. Las automatizaciones no deben ser repetidas entre estudiantes.

## Entrega:

1. **Documentación Técnica:** Incluya una breve descripción de cada automatización creada, explicando qué tarea automatiza, cómo se ha configurado y qué resultado se espera de su ejecución.
2. **Código Fuente:** Adjunte los scripts o comandos utilizados para configurar cada una de las automatizaciones, con los comentarios necesarios para explicar el funcionamiento de cada línea.
3. **Demostración de Funcionalidad:** Proporcione evidencias de que las automatizaciones funcionan correctamente, ya sea mediante capturas de pantalla, registros de salida o videos cortos mostrando su ejecución.

#### **Evaluación:**

- **Investigación y Pruebas:** Se evaluará la calidad de la investigación realizada para elegir la solución adecuada, así como la exhaustividad de las pruebas realizadas.
- **Funcionalidad:** Se evaluará si las automatizaciones cumplen su propósito de manera eficiente y sin requerir intervención del administrador.
- **Seguridad y Permisos:** Se tendrá en cuenta que las tareas automatizadas respeten las políticas de seguridad del sistema, con los permisos y privilegios adecuados.
- **Documentación y Presentación:** La claridad y calidad de la documentación será parte de la evaluación, así como la estructura de los scripts presentados.

**CIERRE DE TEMA**

# AUTOMATIZACIÓN DE TAREAS EN WINDOWS PARA LA ADMINISTRACIÓN DE BASES DE DATOS Y MANEJO DE INCIDENTES

Por Martínez de la Cruz José Julián

Fecha: 12/06/2025

Palabras clave: Schtask, CLI, VirtualBox, VBoxManage, Automatización, Programar tareas, Programador de Tareas, Windows, Snapshots, Backups, Shell, VM, Máquina Virtual, mkdir.

## 1. Objetivo del Tema

El propósito de esta práctica es lograr la implementación de tres automatizaciones de tareas enfocadas a la virtualización utilizando las opciones de la línea de comandos de VBoxManage y Shell de Windows, con el objetivo de garantizar la ejecución de actividades de respaldo y monitoreo de máquinas virtuales sin la intervención directa del administrador del sistema.

## 2. Resumen

Para la realización y cumplimiento de esta práctica se decidió implementar un menú intuitivo que dependiendo de la opción seleccionada cree un nuevo archivo .bat mediante el comando mkdir, esto permite la creación automática de un script que realiza las siguientes automatizaciones enfocadas a la gestión de máquinas virtuales mediante el CLI (Línea de comandos) de VBoxManage y después crea una tarea programada usando el comando schtask en base al script generado previamente. De modo que cada una de estas tareas responda a una necesidad real de mantenimiento evitando los errores que pueda traer consigo la intervención manual.

### 1. Programar una tarea para realizar Snapshots de una Máquina Virtual

La implementación de esta automatización permite crear puntos de restauración en caso de que ocurran fallos y de esta manera tenerlos controlados. Esto asegura que exista siempre una copia reciente de la máquina virtual.

### 2. Programar una tarea que permita eliminar los Snapshots de una Máquina Virtual

Esta automatización permite eliminar Snapshots en un cierto periodo de tiempo para evitar los consumos innecesarios de recursos generados por Snapshots mal gestionadas evitando que el rendimiento de la VM se reduzca.

### 3. Programar una tarea para realizar Backups periódicos de una Máquina Virtual

Esta tarea permite generar una copia externa completa de una VM de modo que cuando se requiera pueda ser restaurada en caso de que la Máquina Virtual original sufra daños.



#### 4. Programar una tarea que permita eliminar periódicamente los Backups de una Máquina Virtual

De la misma manera que con la eliminación de Snapshots, tener una tarea programada que permita la eliminación periódica de Backups permite preservar el espacio y evitar problemas relacionados con el rendimiento del sistema.

#### 5. Opción para eliminar una tarea schtask directamente desde el Shell de Windows

Esta acción permite a los usuarios que no están experimentados con la programación de tareas de Windows, una manera intuitiva de eliminar una tarea programada creadas por el usuario desde la consola de esta, de esta manera se reduce el riesgo de errores generados por usuarios sin experiencia.

La realización de estas automatizaciones presenta sin duda alguna una mejora significativa la manera en que una empresa realiza sus operaciones y les permite la creación de políticas de respaldo y monitoreo para la realización de un entorno de pruebas controlado.

### 3. Conclusión

La automatización de tareas permite a las empresas realizar mantenimiento de sus sistemas en un entorno controlado y mantener su continuidad operativa. En este sentido, la implementación de la herramienta schtask de Windows permite la programación de tareas periódicas para gestionar correctamente estos entornos virtuales sin depender totalmente de la intervención manual, lo que mejora totalmente estos procesos. Además, permite a los usuarios que no están experimentados en el área poder realizar estas tareas de manera intuitiva, de esta manera no se depende en su totalidad del Administrador de Base de Datos.

### 4. Referencias

- Presentación: *Automatización de Tareas en Windows y Linux para la Administración de Base de Datos y Manejo de Incidentes*, MTI Jesús E. Romero, UTTEC.
- Microsoft. (2023). *schtasks*. Microsoft Learn.
- Microsoft. (2023). *mkdir*. Microsoft Learn.

### 5. Reflexiones Finales

1. **Los archivos bat automatizados reducen tiempo y errores humanos:** La intervención manual constante en tareas repetitivas como Backups o Snapshots aumenta el margen de error y consume tiempo operativo. La automatización permite mantener entornos funcionales incluso en ausencia del personal técnico, garantizando la continuidad del servicio.
2. **Usar herramientas del SO sin instalar algo más:** Utilizar funcionalidades ya integradas en el sistema operativo como lo es schtask o mkdir reduce la complejidad del entorno. Esto es especialmente útil en organizaciones con recursos limitados que buscan soluciones efectivas sin agregar más aplicaciones a sus operaciones.