

Hypervisor Remote Control: HRC

Manual de Usuario

**Carlos Eduardo Gómez Montoya^{1,2}, PhD(c).
Jaime Alberto Chavarriaga Lozano¹, PhD.
Harold Enrique Castro Barrera¹, PhD.**

¹Universidad de Los Andes

²Universidad del Quindío

2018

CONTENIDO

1	INTRODUCCIÓN	1
2	VISTA GENERAL DE LA SOLUCIÓN	3
3	ATRIBUTOS DE CALIDAD	4
4	REQUERIMIENTOS DE EJECUCIÓN	6
5	CÓMO USAR EL SISTEMA	7
5.1	CONFIGURACIÓN	7
5.2	EJECUCIÓN	7
5.2.1	HRCServer	7
5.2.2	HRCClient	8
6	CONJUNTO DE COMANDOS DISPONIBLES	9
6.1	INICIAR UNA MÁQUINA VIRTUAL	9
6.2	INICIAR UNA MÁQUINA VIRTUAL SIN INTERFAZ DE USUARIO	10
6.3	APAGAR UNA MÁQUINA VIRTUAL	10
6.4	APAGAR UNA MÁQUINA VIRTUAL – BOTÓN ACPI	11
6.5	APAGAR UNA MÁQUINA VIRTUAL GUARDANDO EL ESTADO	11
6.6	CLONAR UNA MÁQUINA VIRTUAL	12
6.7	REGISTRAR UNA MÁQUINA VIRTUAL	13
6.8	DESREGISTRAR UNA MÁQUINA VIRTUAL	13
6.9	RESTAURAR UNA MÁQUINA VIRTUAL EN EL ÚLTIMO SNAPSHOT	14
6.10	RESTABLECER LA DIRECCIÓN MAC EN UNA MÁQUINA VIRTUAL	15
6.11	ENVIAR UN ARCHIVO AL SERVIDOR	15
6.12	EJECUTAR UN <i>SCRIPT</i>	16
7	NUEVOS COMANDOS	17

1 INTRODUCCIÓN

Las tecnologías de virtualización están evolucionando constantemente. Productos como *Oracle VirtualBox* lanzan constantemente nuevas versiones con características innovadoras y mejoras en la seguridad y el rendimiento. Aunque *VirtualBox* ofrece un servicio web y una API de programación, el diseño y la realización de experimentos de investigación en esta plataforma resultan complicados y propensos a errores. Este documento presenta *HRC (Hypervisor Remote Control)* una solución para controlar el hipervisor y realizar experimentos de investigación. Este programa no solo admite operaciones para crear, modificar y controlar máquinas virtuales, sino que también admite secuencias de comandos de operaciones complejas y tiempos de medición y recursos consumidos. Aquí describimos la herramienta de software y discutimos algunos experimentos que hemos realizado con ella.

HRC (Hypervisor Remote Control): es un sistema de control remoto de hipervisor para ejecutar experimentos de virtualización. Es una aplicación creada para facilitar la administración remota de máquinas virtuales (MVs) a través de comandos para automatizar las funciones principales del hipervisor.

El sistema tiene una arquitectura cliente-servidor. En cada computador donde se ejecute el hipervisor, se ejecuta también el programa servidor (*HRCServer*) el cual queda a la espera de comandos enviados por el programa cliente (*HRCClient*) para su ejecución local. El caso de uso principal de este sistema es la posibilidad de enviar una secuencia de comandos desde el programa cliente para administrar MVs en una sala de cómputo. El cliente especifica el comando y el parámetro, y el servidor traduce el comando a uno que entienda el hipervisor y pueda ejecutarlo.

Mediante el HRC se puede, por ejemplo, encender o apagar una MV, enviando el comando desde un computador para que la MV especificada se encienda o apague en el computador donde está alojada. Dado que el sistema se ejecuta a través de la línea de comandos del sistema operativo, estos comandos se pueden incluir en un *script* para automatizar fácilmente un grupo de MVs en una sala de cómputo. Adicionalmente, el sistema permite enviar archivos al servidor y, para extender aún más su funcionalidad, permite que el programa cliente envíe un comando para ejecutar un *script* previamente creado para que se ejecute de manera local en el computador donde corre el programa *HRCServer*. De esta forma, funciones que no estén incluidas en el *HRC* pueden ser ejecutadas sin necesidad de modificar el código fuente del sistema.

La aplicación *Hypervisor Remote Control*, simple desde el punto de vista del desarrollo de software, es bastante útil para la administración efectiva de MVs, ya que la automatización de tareas ahorra tiempo significativo y reduce la probabilidad de errores.

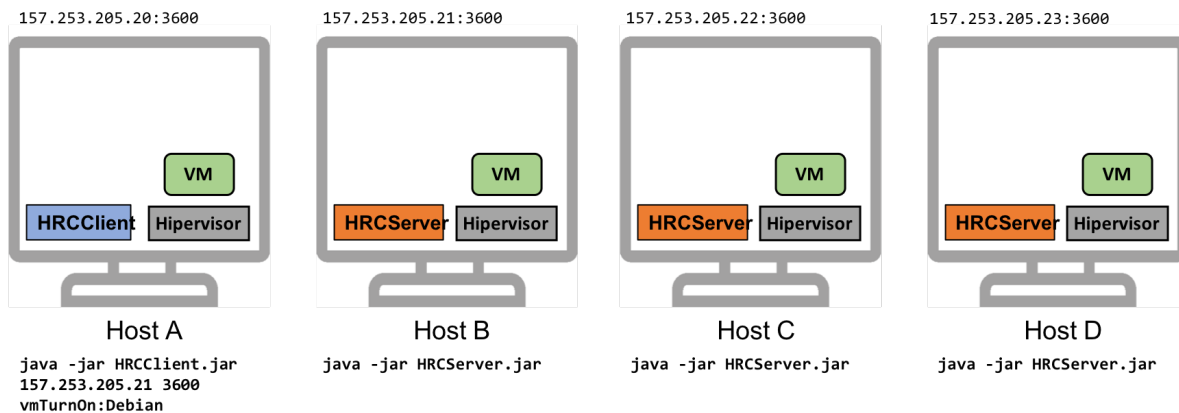
El sistema se ha desarrollado como parte de la tesis doctoral “Confiabilidad de máquinas virtuales en ambientes desktop cloud oportunistas” que actualmente se encuentra en desarrollo por el estudiante *Carlos Eduardo Gómez Montoya* en la *Universidad de Los Andes* de Bogotá, Colombia.

El presente manual fue diseñado con el propósito de señalar las funcionalidades que tiene el sistema para ofrecer a posibles usuarios.

2 VISTA GENERAL DE LA SOLUCIÓN

La solución está conformada por dos aplicaciones *stand-alone*: (1) *RTP – Ring Token Passing*, del cual se ejecuta una instancia en cada MV; y (2): *TM – Test Manager* que se puede ejecutar en una MV o en una MF. El RTP se ejecuta de manera automática y no requiere interfaz de usuario, mientras que el TM se ejecuta de manera interactiva y es donde se definen los parámetros de configuración de cada ejecución. La Figura 1 muestra una vista general de la solución.

Figura 3. Hypervisor Remote Control – Vista general de la solución.



3 ATRIBUTOS DE CALIDAD

A continuación, se presentan brevemente los atributos de calidad tenidos en cuenta durante el diseño de la aplicación.

Extensibilidad: Como mecanismo para facilitar la extensibilidad dentro del sistema se utilizaron patrones de diseño que fomentaran dicho atributo. Por ejemplo, para mantener la aplicación lo más desacoplada posible, configurando la aplicación a través de archivos de propiedades, con algunas propiedades requeridas y otras opcionales que, en caso de no incluirse, toman un valor por defecto. Dado que el sistema permite realizar un conjunto determinado de acciones en el host remoto, esta aplicación se puede extender usando scripts, a través de los cuales, se puede realizar lo que no está soportado como comandos de la aplicación.

Desempeño: Esta es una aplicación que corre a través de la red y es controlada mediante el paso de mensajes de los clientes a los servidores. Hemos realizado pruebas de ejecución de actividades de administración de máquinas virtuales, por ejemplo, para el despliegue de 448 MVs en menos de 10 minutos. Luego de ejecutar un comando, en el log del servidor queda registrado el tiempo que tardó la ejecución de cada comando.

Usabilidad: En el lado del servidor, la aplicación se configura fácilmente mediante el archivo de propiedades. Se pueden modificar las rutas de las carpetas requeridas para su ejecución, y el número de puerto. Esta aplicación se utiliza a través de la línea de comandos y permite la creación de *scripts* para ejecutar una secuencia de tareas en una sala de cómputo.

Integridad: En este sentido, la aplicación está pensada para la realización enviar comandos remotos al servidor, el cual los traduce y reenvía al hipervisor. Cualquier error en el uso de los comandos, son responsabilidad directa del usuario y no de la aplicación de control remoto.

Portabilidad: El software fue desarrollado en Java y se utilizaron librerías estándar, por lo que puede correr sin modificaciones en distintas plataformas, tales como Windows, Mac OS X y Linux.

Compatibilidad: No aplica, ya que se trata de la primera versión del software.

Mantenimiento: El código fuente sigue el estándar de codificación que se utiliza en la industria del software para proyectos en Java. El código además está debidamente documentado.

Documentación: Para la documentación del software desarrollado se hizo un proceso de ingeniería de software ligero con levantamiento de requerimientos a partir de las

historias de usuario, y se escribieron los comentarios para la generación de la documentación en formato Javadoc.

4 REQUERIMIENTOS DE EJECUCIÓN

Aunque el sistema fue desarrollado para su ejecución en ambientes Windows, este programa puede correr de manera correcta en cualquier plataforma que tenga instalado Java Runtime Environment (JRE) versión 7 o superior y Oracle Virtual Box versión 4.3 o superior.

5 CÓMO USAR EL SISTEMA

5.1 CONFIGURACIÓN

El programa *HRCServer* recibe su configuración de un archivo de propiedades. Las propiedades necesarias, como se puede apreciar en la Figura 1, son: el número de puerto donde el programa va a escuchar las peticiones, la ruta donde se encuentra el archivo *VboxManage* del hipervisor Oracle VirtualBox, o el equivalente en otro hipervisor, y la ruta donde se encuentran las MVs en el computador donde corre el servidor.

Figura 1. Propiedades de configuración del programa *HRCServer*.

hrc.properties
port=3600
virtualMachinesHome=C:\\Users\\Profesor\\VirtualBox VMs\\
virtualBoxHome=C:\\Program Files\\Oracle\\VirtualBox\\

Por su parte, el programa *HRCClient* no requiere configuración, sino que recibe por la línea de comandos los parámetros necesarios para enviar los comandos a las instancias de *HRCServer* que se encuentran en ejecución en los equipos de una sala de cómputo.

5.2 EJECUCIÓN

5.2.1 HRCServer

Este programa *HRCServer* se ejecuta desde la línea de comandos. Al ejecutar, usando **java -jar HRCServer.jar**, se puede ver en la consola (Figura 2), el servidor indica: (1) que se encuentra en ejecución, (2) la forma como un cliente puede enviar comando, ya que muestra la dirección IP de la máquina donde corre el servidor y su número de puerto, y (3) el conjunto de comandos que ofrece.

Figura 2. Ejecución del programa *HRCServer*.

```
11-22-2018 13:05:52,539 INFO HRCServer:Hypervisor Remote Control Server is
running ... in ther port: 3600 Last update: November 22nd, 2018
Usage: HRCClient 157.253.199.206 3600 command

Supported commands:
vmTurnOn:name      Starts a VM
vmTurnOnH:name     Starts a headless VM
vmTurnOff:name     Stops a VM
vmTurnOffACPI:name Stops a VM with ACPI power button
vmTurnOffSave:name Stops a VM saving the state of the VM
vmClone:name       Clones a VM
```

vmRegister:name	Registers a VM
vmUnregister:name	Unregisters a VM
vmRestore:name	Resumes a VM in last snapshot
vmMacReset:name	Resets the mac address of the first NIC
sendFile:name	Sends a file
script.bat	Executes a .bat file

5.2.2 HRCClient

Al igual que el programa *HRCServer*, el programa *HRCClient* también se ejecuta desde la línea de comandos, usando **java -jar HRCClient.jar**, seguido de la dirección IP y el número de puerto del servidor que recibirá y ejecutará el comando. La Figura 3 muestra un ejemplo de la ejecución del programa cliente, encendiendo la MV cuyo nombre es **Debian**.

Figura 3. Ejecución del programa *HRCClient*. Comando para iniciar una MV.

```
Java -jar HRCClient 157.253.199.206 3600 vmTurnOn:Debian
```

Los mensajes que se ven en la consola, tanto del cliente como del servidor tienen el formato de registro de eventos (*log*) y quedan almacenados en la carpeta Logs.

En la siguiente sección se describe con más detalle el conjunto de comandos que se han incluido en esta primera versión del sistema

6 CONJUNTO DE COMANDOS DISPONIBLES

Al iniciar el programa servidor, el servidor queda esperando por comandos del cliente, tal como se muestra en la Figura 2 y como se mencionó en la *Sección 3.2.1*. Cuando el servidor recibe un mensaje del cliente, lo interpreta para determinar cuál es el comando, y si está dentro del conjunto de comandos conocidos, lo ejecuta de manera local. Cualquier mensaje de respuesta que el llamado a un comando cualquiera produzca, el servidor lo envía al cliente como respuesta para que el cliente lo muestre en su consola. La sintaxis definida para usar entre el cliente y el servidor es **comando:parámetro**, donde **comando** es uno de los comandos establecidos y **parámetro** es un valor cuyo significado depende del comando particular. En su mayoría, el parámetro se refiere al nombre de la MV a la que se refiere el comando.

A continuación se describe cada uno de los comandos soportados por esta versión del sistema *Hypervisor Remote Control*.

6.1 INICIAR UNA MÁQUINA VIRTUAL

El comando **vmTurnOn** inicia, en el computador donde corre el programa *HRCServer*, la MV especificada como parámetro. La Figura 3 muestra el comando utilizado en el cliente, la Figura 4 muestra la salida en la consola del servidor y la Figura 5 presenta la salida en la consola del cliente. En este caso, el cliente desea iniciar la MV llamada **Debian**. Si la MV existe en el computador donde corre el servidor, está registrada en el catálogo de MVs del hipervisor y no está en ejecución, la MV se inicia y, como se puede notar en la Figura 5, no hay un mensaje de respuesta. Por el contrario, si la MV no se puede iniciar, el hipervisor responde con un mensaje de error, el cual es redirigido al cliente.

Figura 4. Iniciar una MV – Lado Servidor.

```
11-22-2018 11:31:44,259 INFO HRCServer:Connection incoming ...
11-22-2018 11:31:44,274 INFO HRCServer:vmTurnOn:Debian
11-22-2018 11:31:44,274 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage startvm Debian
```

Figura 5. Iniciar una MV – Lado Cliente.

```
11-22-2018 11:31:43,659 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmTurnOn:Debian
11-22-2018 11:31:49,775 INFO HRCClient:FROM SERVER:
```

6.2 INICIAR UNA MÁQUINA VIRTUAL SIN INTERFAZ DE USUARIO

En algunas ocasiones, las MVs no son creadas para interactuar directamente con los usuarios, sino recibiendo acceso remoto a través de un protocolo como *SSH (Secure Shell)*. Estas MVs son iniciadas con el comando **vmTurnOnH**. El comportamiento tanto del cliente como del servidor es igual que en el comando **vmTurnOn**, descrito en la sección anterior.

Figura 6. Comando para iniciar una MV sin interfaz de usuario

```
Java -jar HRCClient 157.253.199.206 3600 vmTurnOnH:Debian
```

La Figura 6 muestra el comando para iniciar una MV sin interfaz de usuario. En las Figuras 7 y 8 se pueden ver los mensajes que salen en la consola del servidor y cliente, respectivamente.

Figura 7. Iniciar una MV sin interfaz de usuario – Lado Servidor.

```
11-22-2018 11:32:49,002 INFO HRCServer:Connection incoming ...
11-22-2018 11:32:49,049 INFO HRCServer:vmTurnOnH:Debian
11-22-2018 11:32:49,049 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage startvm Debian --type headless
```

Figura 8. Iniciar una MV sin interfaz de usuario – Lado Cliente.

```
11-22-2018 11:32:48,465 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmTurnOnH:Debian
11-22-2018 11:32:54,059 INFO HRCClient:FROM SERVER:
```

6.3 APAGAR UNA MÁQUINA VIRTUAL

El comando **vmTurnOff** apaga la máquina virtual especificada. El ejemplo de la Figura 9 muestra el comando empleado para apagar la MV **Debian**.

Figura 9. Comando para apagar una MV.

```
Java -jar HRCClient 157.253.199.206 3600 vmTurnOff:Debian
```

Las Figuras 10 y 11 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 9 para apagar una MV.

Figura 10. Apagar una MV – Lado Servidor.

```
11-22-2018 11:32:40,111 INFO HRCServer:Connection incoming ...
11-22-2018 11:32:40,152 INFO HRCServer:vmTurnOff:Debian
11-22-2018 11:32:40,152 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage controlvm Debian poweroff
11-22-2018 11:32:40,849 INFO
HRCServer:0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

Figura 11. Apagar una MV – Lado Cliente.

```
11-22-2018 11:32:39,556 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmTurnOff:Debian
11-22-2018 11:32:40,322 INFO HRCClient:FROM SERVER:
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

6.4 APAGAR UNA MÁQUINA VIRTUAL – BOTÓN ACPI

El comando **vmTurnOffACPI** envía una señal de apagado ACPI a la MV, como si el botón de encendido/apagado de un computador hubiera sido presionado.

Figura 12. Comando para apagar una MV – Botón ACPI.

```
Java -jar HRCClient 157.253.199.206 3600 vmTurnOffACPI:Debian
```

Las Figuras 13 y 14 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 12 para apagar una MV con el botón ACPI.

Figura 13. Apagar una MV – Botón ACPI – Lado Servidor.

```
11-22-2018 11:33:30,930 INFO HRCServer:Connection incoming ...
11-22-2018 11:33:30,971 INFO HRCServer:vmTurnOffACPI:Debian
11-22-2018 11:33:30,971 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage controlvm Debian acpipowerbutton
```

Figura 14. Apagar una MV – Botón ACPI – Lado Cliente.

```
11-22-2018 11:33:30,374 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmTurnOffACPI:Debian
11-22-2018 11:33:30,990 INFO HRCClient:FROM SERVER:
```

6.5 APAGAR UNA MÁQUINA VIRTUAL GUARDANDO EL ESTADO

El comando **vmTurnOffSave** guarda el estado de la MV y la apaga.

Figura 15. Comando para apagar una MV guardando el estado.

```
Java -jar HRCClient 157.253.199.206 3600 vmTurnOffSave:Debian
```

Las Figuras 16 y 17 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 15 para apagar una MV guardando el estado.

Figura 16. Apagar una MV guardando el estado – Lado Servidor.

```
11-22-2018 11:34:41,338 INFO HRCServer:Connection incoming ...
11-22-2018 11:34:41,379 INFO HRCServer:vmTurnOffSave:Debian
11-22-2018 11:34:41,380 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage controlvm Debian savestate
11-22-2018 11:34:42,588 INFO
HRCServer:0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

Figura 17. Apagar una MV guardando el estado – Lado Cliente.

```
11-22-2018 11:34:40,781 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmTurnOffSave:Debian
11-22-2018 11:34:42,062 INFO HRCClient:FROM SERVER:
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

6.6 CLONAR UNA MÁQUINA VIRTUAL

El comando **vmClone** produce una MV igual a la especificada.

Figura 18. Comando para clonar una MV.

```
Java -jar HRCClient 157.253.199.206 3600 vmClone:Debian
```

Las Figuras 19 y 20 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 18 para clonar una MV.

Figura 19. Clonar MV – Lado Servidor.

```
11-22-2018 11:36:01,880 INFO HRCServer:Connection incoming ...
11-22-2018 11:36:01,927 INFO HRCServer:vmClone:Debian
11-22-2018 11:36:01,927 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage clonevm Debian --name Debian-clone --mode
machine --register
11-22-2018 11:36:53,662 INFO
HRCServer:0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```


Figura 20. Clonar una MV – Lado Cliente.

```
11-22-2018 11:36:01,337 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmClone:Debian
11-22-2018 11:36:53,239 INFO HRCClient:FROM SERVER:
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

6.7 REGISTRAR UNA MÁQUINA VIRTUAL

El comando **vmRegister** registra una MV igual en la lista de MVs del hipervisor. La MV que se desea registrar debe existir y no estar registrada en el catálogo de MVs.

Figura 21. Comando para registrar una MV.

```
Java -jar HRCClient 157.253.199.206 3600 vmRegister:Debian-clone
```

Las Figuras 22 y 23 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 21 para registrar una MV.

Figura 22. Registrar MV – Lado Servidor.

```
11-22-2018 12:00:18,648 INFO HRCServer:vmRegister:Debian-clone
11-22-2018 12:00:18,648 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage registervm "C:\Users\Profesor\VirtualBox
VMs\Debian-clone\Debian-clone.vbox"
```

Figura 23. Registrar una MV – Lado Cliente.

```
11-22-2018 12:00:18,045 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmRegister:Debian-clone
11-22-2018 12:00:18,660 INFO HRCClient:FROM SERVER:
```

6.8 DESREGISTRAR UNA MÁQUINA VIRTUAL

El comando **vmUnregister** eliminar una MV (desregistra) de la lista de MVs del hipervisor. La MV que se desea desregistrar debe estar registrada en el catálogo de MVs.

Figura 24. Comando para registrar una MV.

```
Java -jar HRCClient 157.253.199.206 3600 vmUnregister:Debian-clone
```

Las Figuras 25 y 26 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 24 para desregistrar una MV.

Figura 25. Desregistrar MV – Lado Servidor.

```
11-22-2018 11:59:57,257 INFO HRCServer:vmUnregister:Debian-clone
11-22-2018 11:59:57,257 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage unregistervm Debian-clone
```

Figura 26. Desregistrar una MV – Lado Cliente.

```
11-22-2018 11:59:56,648 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmUnregister:Debian-clone
11-22-2018 11:59:57,305 INFO HRCClient:FROM SERVER:
```

6.9 RESTAURAR UNA MÁQUINA VIRTUAL EN EL ÚLTIMO SNAPSHOT

El comando **vmRestore** restaura una MV a partir del último *snapshot*. La MV que se desea restaurar debe tener al menos un *snapshot*. Luego de este comando, la MV debe ser iniciada, por ejemplo, usando el comando **vmTurnOn**, descrito en la sección 4.1.

Figura 27. Comando para restaurar una MV.

```
Java -jar HRCClient 157.253.199.206 3600 vmRestore:Debian
```

Las Figuras 28 y 29 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 27 para desregistrar una MV.

Figura 28. Restaurar MV – Lado Servidor.

```
11-22-2018 12:06:05,312 INFO HRCServer:Connection incoming ...
11-22-2018 12:06:05,344 INFO HRCServer:vmRestore:Debian
11-22-2018 12:06:05,344 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage snapshot Debian restorecurrent
11-22-2018 12:06:06,078 INFO
HRCServer:0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

Figura 29. Restaurar una MV – Lado Cliente.

```
11-22-2018 12:06:04,760 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: vmRestore:Debian
11-22-2018 12:06:05,541 INFO HRCClient:FROM SERVER:
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

6.10 RESTABLECER LA DIRECCIÓN MAC EN UNA MÁQUINA VIRTUAL

El comando **vmMacReset** restablece la dirección MAC de la interfaz de red de una MV.

Figura 30. Comando para restablecer la dirección MAC en una MV

```
Java -jar HRCClient 157.253.199.206 3600 vmMacReset:Debian
```

Las Figuras 31 y 32 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 30 para restablecer la dirección MAC en una MV.

Figura 31. Restablecer la dirección MAC de la interfaz de red de una MV – Lado Servidor.

```
11-22-2018 12:06:05,312 INFO HRCServer:Connection incoming ...
11-22-2018 12:06:05,344 INFO HRCServer:vmRestore:Debian
11-22-2018 12:06:05,344 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage snapshot Debian restorecurrent
11-22-2018 12:06:06,078 INFO
HRCServer:0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

Figura 32. Restablecer la dirección MAC de la interfaz de red de una MV – Lado Cliente.

```
11-22-2018 12:08:29,078 INFO HRCServer:Connection incoming ...
11-22-2018 12:08:29,125 INFO HRCServer:vmMacReset:Debian
11-22-2018 12:08:29,125 INFO HRCServer:Running: C:\Program
Files\Oracle\VirtualBox\VBoxManage modifyvm Debian --macaddress1 auto
```

6.11 ENVIAR UN ARCHIVO AL SERVIDOR

El comando **sendFile** envía un archivo al servidor. El archivo a enviar es especificado mediante su nombre y debe estar en la carpeta **Source**. Al llegar al servidor, se almacena en la carpeta **Destination**.

Figura 33. Comando para enviar un archivo al servidor

```
Java -jar HRCClient 157.253.199.206 3600 sendFile:Foto-01.PNG
```

Las Figuras 34 y 35 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 33 para enviar un archivo al servidor.

Figura 34. Enviar un archivo – Lado Servidor.

```
11-22-2018 12:10:29,669 INFO HRCServer:Connection incoming ...
11-22-2018 12:10:29,701 INFO HRCServer:sendFile:Foto-01.PNG
11-22-2018 12:10:29,701 INFO HRCServer:Running: sendFile:Foto-01.PNG
11-22-2018 12:10:29,716 INFO HRCServer:Creando el archivo Foto-01.PNG
11-22-2018 12:10:29,716 INFO HRCServer:File size: 32618bytes    File transfer
time: 0.013067766 s
```

Figura 35. Enviar un archivo – Lado Cliente.

```
11-22-2018 12:10:29,116 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: sendFile:Foto-01.PNG
11-22-2018 12:10:29,163 INFO HRCClient:Size: 32618
11-22-2018 12:10:29,163 INFO HRCClient:Number of blocks: 32
11-22-2018 12:10:29,163 INFO HRCClient:Size of last block: 874
11-22-2018 12:10:29,178 INFO HRCClient:FROM SERVER: File size: 32618bytes
File transfer time: 0.013067766 s
```

6.12 EJECUTAR UN *SCRIPT*

Cuando el cliente le envía al servidor un mensaje que no corresponde a alguno de los comandos disponibles, asume que se trata de un *script* que se desea ejecutar en el servidor. Los comandos deben estar almacenados en la misma carpeta donde se encuentra el servidor.

Figura 36. Comando para ejecutar un *script* en el servidor

```
Java -jar HRCClient 157.253.199.206 3600 copiar.bat
```

Las Figuras 37 y 38 muestran los mensajes de respuesta que se ven en la consola del servidor y cliente, respectivamente al ejecutar el comando de la Figura 36 para ejecutar un *script*.

Figura 37. Ejecutar un *script*– Lado Servidor.

```
11-22-2018 12:13:42,386 INFO HRCServer:Running: copiar.bat
11-22-2018 12:13:42,698 INFO HRCServer:
```

Figura 38. Ejecutar un script – Lado Cliente.

```
11-22-2018 12:13:41,793 INFO HRCClient:Running a remote task: Ip address:
157.253.205.21 Port: 3600 Command: copiar.bat
11-22-2018 12:13:42,152 INFO HRCClient:FROM SERVER:
```

7 NUEVOS COMANDOS

Algunos comandos se han incorporado recientemente, en modo experimental, los cuales pueden contribuir con la administración remota de una sala de cómputo. Entre los nuevos comandos se encuentran:

`vmModify:name:profile` para modificar el perfil de hardware de una máquina virtual. Los perfiles predefinidos son los siguientes: (1) 1 CPU, 1 GB RAM, (2) 2 CPU, 2 GB RAM, (3) 4 CPU, 4 GB RAM, (4) 6 CPU, 8 GB RAM.

`ServerOff:` para apagar el servidor HRC.

`vmRegisterDisk:disk` para registrar un disco en el administrador de medios virtuales del hipervisor Oracle VirtualBox.

`vmUnRegisterDisk:disk` para eliminar un disco del administrador de medios virtuales del hipervisor Oracle VirtualBox.

`vmListDisks` para obtener un listado de los discos registrados en el administrador de medios virtuales del hipervisor Oracle VirtualBox.

`serverRun` para ejecutar un comando básico en el servidor.

`vmRun` para ejecutar un comando en la máquina virtual. Se usa mediante la siguiente sintaxis `vmRun:name:user:password:command:options`

`serverListFiles` para ver un listado de archivos y directorios en el lado del servidor.