

# Shell Scripting TIC – TAC

i

1 Vote

**Saludos:** Integrantes de la **Gran Comunidad de Usuarios del Software Libre** (No necesariamente gratis) y **Usuarios del Linux Post Install – Script Bicentenario (CMSL / U-LPI-SB)**, Compañeros, Colegas y Autodidactas del **Siglo XXI**, Ingenieros, Licenciados y Técnicos de áreas afines a las Tecnologías de la Información.



([https://proyectotictac.files.wordpress.com/2013/10/1385044\\_10151749140259538\\_1049479195\\_n.jpg](https://proyectotictac.files.wordpress.com/2013/10/1385044_10151749140259538_1049479195_n.jpg))

**RECORDATORIO:** Recuerda que muchas veces el carácter de doble guión “ – ” en una orden de comando escrita en el Blog se visualiza como un solo guión largo “ — ”, el carácter comillas simple “ ’ ” se visualiza como un acento “ ‘ ” y el carácter comillas doble recta “ ” ” se visualiza como comillas doble inclinadas “ ” ”, por lo que si al momento de copiar la orden de comando desde el Blog y pegarla en una Consola / Terminal no se ejecuta correctamente la misma, verifique y corrija si el problema es causado por un carácter mal traducido guión largo que debe ser escrito como doble guión. Lo mismo puede llegar a suceder con algún otro carácter no alfanumérico, así que tenga cuidado al copiar y pegar en una Terminal / Consola.

Esta sección del Blog tiene como objetivo servir de repositorio de **ordenes de comando útiles** para la facilitar la creación de **Scripts** en un **Shell tipo Bash** o compatible.

Aprender sobre el **Shell Scripting**, es decir, a **ejecutar** ordenes complejas sobre el **Terminal (Consola) de GNU/Linux** es muy útil para automatizar actividades rutinarias e importantes dentro de nuestro **Sistema Operativo GNU/Linux** para así optimizar nuestros **Recursos y Tiempo**, es decir, con lo aquí contenido exploraremos como desde el terminal podemos ejecutar ordenes de comando manuales que luego nos permitan **programar/automatizar** actividades sobre un equipo ahorrando **Horas/Labor** de ejecución manual o presencial, implementando dichas ordenes en un Script de Bash Shell o compatible y explicado de una manera, practica y sencilla.

Todo esto con la intención de seguir intentar **desmitificar** otro de esos **paradigmas** sobre **GNU/Linux** que muchos de los **Usuarios de Sistemas Operativos Privativos** y uno que otro **Linuxero Novato** tienen todavía metido entre ceja y ceja. El mito de **"GNU/Linux es muy complejo porque todo lo importante es manual vía terminal"** y **"No sé como realizar un Script de Bash Shell en un Sistema Operativo GNU/Linux desde un Terminal"**.

En fin, abarcaremos como en una sola orden de comando podemos extraer y mostrar valores / información del Sistema Operativo / Hardware, que luego podamos implementar dentro de un **Script de Bash Shell** para automatizar una tarea especifica. Concentrándonos en las **Mejores Practicas (Best Practices)** necesarias para obtener un excelente diseño de un Script de la manera **más eficiente y practica**.

Bueno aquí lo tienen, y demuéstrenle a los escépticos e incrédulos que no es cierto, que solo es desconocimiento o falta de experiencia técnica. Estos son pequeños **"Tips"** que nos facilitan realizar esas operaciones técnicas reservadas por lo general para personas **expertas (Administradores de Sistemas y/o Servidores)** en **DEBIAN** o nuestro **Sistema Operativo Nacional** basado en **DEBIAN**, **"Canaima GNU/Linux"**.

---

Recuerda donar por lo menos 1\$ US a este u otro Proyecto de apoyo o desarrollo de Software Libre, GNU, Linux o Codigo Abierto.

Ayúdanos a mantener nuestro proyecto



([https://www.paypal.com/cgi-bin/\\_webscr?cmd=s-xclick&hosted\\_button\\_id=M2JM3Y9XU9FYS](https://www.paypal.com/cgi-bin/_webscr?cmd=s-xclick&hosted_button_id=M2JM3Y9XU9FYS))

Dona al Proyecto TIC - TAC

---

Bueno aquí les va la información:

**VARIABLES CON INFORMACIÓN SOBRE LA FECHA, HORA, VERSIÓN, ARQUITECTURA DEL SISTEMA OPERATIVO Y SUS USUARIOS, ENTRE OTROS PARÁMETROS:**

**2. Nombre del Equipo.****1. FECHA\_INICIO=\$(date +"%D")****2. Fecha actual del Sistema****1. FECHA\_ALFABET=\$(date +"%d-%b-%y")****2. Fecha actual del Sistema****1. FECHA\_ALFANUM=\$(date +"%d-%m-%y")****2. Fecha Numérica actual del Equipo****1. FECHA\_ACTUAL\_EXT=\$(date "+%d-%m-%y\_%H-%M-%S")****2. Fecha actual extendida del Sistema****1. HORA\_INICIO=\$(date +"%T")****2. Hora actual del Sistema****1. HORA\_ACTUAL=\$(date +"%H-%M")****2. Hora actual del Sistema****1. HORA\_ACTUAL\_EXT=\$(date +"%H-%M-%S")****2. Fecha actual extendida del Sistema****1. HORA=\$(date +"%H")****2. Hora del Sistema****1. MINUTO=\$(date +"%M")****2. Minuto del Equipo****1. SEGUNDOS=\$(date +"%S")****2. Segundos del Sistema****1. DIA=\$(date +"%d")****2. Día actual del Equipo****1. MES\_ALFABET=\$(date +"%b")****2. Mes alfabético actual del Equipo****1. MES\_ALFANUM=\$(date +"%m")****2. Mes numérico actual del Equipo****1. AÑO\_2C=\$(date +"%y")****2. Año (con 2 cifras) actual del Equipo****1. AÑO\_4C=\$(date +"%Y")****2. Año (con 4 cifras) actual del Equipo****1. NOMBRE\_USUARIO=\$(cat /etc/passwd | grep 1000 | cut -d: -f1)****2. Nombre de Usuario creado por UID/GID (EJM. 1000) - VERSIÓN SIMPLE**

1. **NOMBRE\_USUARIO=\$(awk -F: '{if (\$3==1000) print \$1}' /etc/passwd)**
2. **Nombre de Usuario creado por UID/GID (EJM. 1000) - VERSIÓN COMPLEJA**

1. **NOMBRE\_USUARIO=\$(awk -F: '{if (\$4==1000) print \$1}' /etc/passwd)**
2. **Nombre de Usuario creado por UID/GID (EJM. 1000) - VERSIÓN COMPLEJA**

1. **IDSUPERUSER=\$(id -u | awk '{print \$1}')**
2. **IUD del Superusuario root**

1. **IDSUPERUSER=\$(echo \$UID)**
2. **IUD del Superusuario root**

1. **IDSUPERUSER=\$(getent passwd \$USER | cut -d: -f3)**
2. **IUD del Superusuario root**

1. **USER\_NAME\_1=\$(cat /etc/passwd | grep 1000 | cut -d: -f1 | awk '{print\$1}')**
2. **Nombre del usuario 1000**

1. **HOME\_USER\_NAME\_1=/home/\$USER\_NAME\_1**
2. **/home del usuario 1000 utilizando la variable anterior**

1. **USER\_NAME\_2=\$(cat /etc/passwd | grep 1001 | cut -d: -f1 | sed -n '2p' | awk '{print \$1}')**
2. **Nombre del usuario 1001**

1. **HOME\_USER\_NAME\_2=/home/\$USER\_NAME\_2**
2. **/home del usuario 1001 utilizando la variable anterior**

1. **ID\_FIRST\_USERNAME=\$(for i in 1000 1001 1002 1003 1004; do grep \$i /etc/passwd | awk '{print "\t" \$1}' | tr -d '[A-Za-z]' | sed 's/::x\*//' | cut -c 1-5 | awk '{print \$1}' ; done | sed -n '1p')**
2. **Almacenar el ID del Nombre del primer usuario creado en el sistema entre los ID del 1000 al 1005. Nota: Solo funciona con la opción -o desactivada del MODO BASH ESTRICTO ó SEGURO (Bash Strict Mode / BSM)**

1. **FIRST\_ID\_USERNAME=\$(for i in \$(seq 1000 1999); do grep \$i /etc/passwd | awk '{print "\t" \$1}' | tr -d '[A-Za-z]' | sed 's/::x\*//' | cut -c 1-5 | awk '{print \$1}' ; done | sed -n '1p')**
2. **Almacenar el ID del Nombre del primer usuario creado en el sistema. Nota: Solo funciona con la opción -o desactivada del MODO BASH ESTRICTO ó SEGURO (Bash Strict Mode / BSM)**

1. **QUIENSOY=\$(who -m | awk '{print \$5}' | awk 'NR==1')**
2. **Nombre de quien esta logueado en el Sistema.**

1. **DIR\_INICIO=\$(echo \$PWD)**
2. **Ruta de ejecución del Script.**

1. **NOM\_DISTRO\_N**=\$(cat /etc/os-release | grep ID | grep -v "VERSION" | cut -f2 -d\=)

2. **Nombre (Normal/Extendido) de la Distro detectada con todas las letras en minúscula.**

1. **NOM\_DISTRO\_N**=\$(cat /etc/os-release | grep ID | grep -v "VERSION\_ID" | grep -v "ID\_LIKE" | sed 's/ID=/' | sed 's/'//g')

2. **Nombre (Normal/Extendido) de la Distro detectada con todas las letras en minúscula.**

1. **NOM\_DISTRO\_N**=\$(lsb\_release -d | sed 's/Description:/' | awk '{print \$1}')

2. **Nombre (Normal/Extendido) de la Distro detectada con la primera letra en mayúscula.**

1. **NOM\_DISTRO\_N**=\$(lsb\_release -i | awk '{print \$3}')

2. **Nombre (Normal/Extendido) de la Distro detectada con la primera letra en mayúscula.**

1. **NOM\_DISTRO\_N**=\$(cat /etc/os-release | grep NAME | grep -v "VERSION" | sed -n '2p' | cut -f2 -d\" | awk '{print \$1}')

2. **Nombre (Normal/Extendido) de la Distro detectada con la primera letra en mayúscula.**

1. **NOM\_DISTRO\_E**=\$(lsb\_release -d | sed 's/Description:/' | awk '{print \$1, \$2, \$3, \$4}')

2. **Nombre (Normal/Extendido) de la Distro detectada con la primera letra en mayúscula.**

1. **NOM\_DISTRO\_E**=\$(cat /etc/os-release | grep NAME | grep -v "VERSION" | sed -n '2p' | cut -f2 -d\" )

2. **Nombre (Normal/Extendido) de la Distro detectada con la primera letra en mayúscula.**

1. **NUM\_ID\_DISTRO\_N**=\$(cat /etc/os-release | grep VERSION= | sed -n '1p' | sed 's/VERSION=/' | sed 's/'//g' | awk '{print \$1}')

2. **Número de Identificación (Normal/Extendido) de la Distro detectada.**

1. **NUM\_ID\_DISTRO\_N**=\$(cat /etc/os-release | grep VERSION\_ID= | sed -n '1p' | sed 's/VERSION\_ID=/' | sed 's/'//g')

2. **Número de Identificación (Normal/Extendido) de la Distro detectada.**

1. **NUM\_ID\_DISTRO\_E**=\$(lsb\_release -r | sed 's/Release:/' )

2. **Número de Identificación (Normal/Extendido) de la Distro detectada.**

1. **NUM\_ID\_DISTRO\_E**=\$(lsb\_release -d | awk '{print \$4}')

2. **Número de Identificación (Normal/Extendido) de la Distro detectada.**

1. **NOM\_COD\_DISTRO**=\$(lsb\_release -c | sed 's/Codename:/' )

**2. Nombre Código de la Distro detectada.**

```
1. NOM_COD_DISTRO=$(cat /etc/os-release | grep VERSION= | sed -n '1p' | sed 's/VERSION=/' | sed 's/"//g' | awk '{print $2}' | sed 's/(//g' | sed 's/)//g')
```

**2. Nombre Código de la Distro detectada.**

```
1. TIPO_DISTRO=$(uname -o)
```

**2. Nombre del tipo de Sistema Operativo Linux encontrado.**

```
1. TIPO_KERNEL=$(uname -s)
```

**2. Nombre del Tipo de Kernel del Sistema Operativo encontrado.**

```
1. RELEASE_KERNEL=$(uname -r)
```

**2. Versión del Kernel del Sistema Operativo encontrado.**

```
1. VERSION_KERNEL=$(uname -v | awk '{print $4}')
```

**2. Versión del Kernel del Sistema Operativo encontrado.**

```
1. ARQ_DISTRO_COD=$(uname -m)
```

**2. Arquitectura de la Distro detectada en forma codificada.**

```
1. ARQ_DISTRO_COD=$(uname -m) ; if [[ "$ARQ_DISTRO_COD" = "x86" ]]; then ARQ_DISTRO_HUM=32; else ARQ_DISTRO_HUM=64; fi
```

**2. Convertir el Tipo de Arquitectura de la Distro detectada en forma codificada**

```
1. VERSION_BASH=$(bash --version | grep bash | awk '{print $4}')
```

**2. Versión del Bash Shell.**

```
#+++++
```

**VARIABLES CON INFORMACIÓN SOBRE LA MEMORIA RAM Y LAS PARTICIONES DEL DISCO DURO DEL EQUIPO:**

```
1. MEM_TOTAL=$(free -h | sed '1 d' | grep Mem: | awk '{print $2}')
```

**2. Memoria total del Equipo.**

```
1. MEM_USADA=$(free -h | sed '1 d' | grep Mem: | awk '{print $3}')
```

**2. Memoria usada del Equipo.**

```
1. MEM_LIBRE=$(free -h | sed '1 d' | grep Mem: | awk '{print $4}')
```

**2. Memoria libre del Equipo.**

```
1. MEM_COMPARTIDA=$(free -h | sed '1 d' | grep Mem: | awk '{print $5}')
```

**2. Memoria compartida del Equipo.**

```
1. MEM_ALMACENADA=$(free -h | sed '1 d' | grep Mem: | awk '{print $6}')
```

**2. Memoria almacenada del Equipo.**

1. **MEM\_CACHEADA=\$(free -h | sed '1 d' | grep Mem: | awk '{print \$7}')**

2. **Memoria cacheada del Equipo.**

1. **SWAP\_TOTAL=\$(free -h | sed '1 d' | grep Swap: | awk '{print \$2}')**

2. **Memoria Swap total del Equipo.**

1. **SWAP\_USADA=\$(free -h | sed '1 d' | grep Swap: | awk '{print \$3}')**

2. **Memoria Swap usada del Equipo.**

1. **SWAP\_LIBRE=\$(free -h | sed '1 d' | grep Swap: | awk '{print \$4}')**

2. **Memoria Swap libre del Equipo.**

#+++++

1. **ESPACIO\_TOTAL=\$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print \$2}')**

2. **Espacio total de una partición predefinida.**

1. **ESPACIO\_TOTAL=\$(df -h | sed '1 d' | grep disk | awk '{print \$2}')**

2. **Espacio total de una partición predefinida.**

1. **ESPACIO\_USADO=\$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print \$3}')**

2. **Espacio usado de una partición predefinida.**

1. **ESPACIO\_USADO=\$(df -h | sed '1 d' | grep disk | awk '{print \$3}')**

2. **Espacio usado de una partición predefinida.**

1. **ESPACIO\_DISPONIBLE=\$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print\$4}')**

2. **Espacio disponible de una partición predefinida.**

1. **ESPACIO\_DISPONIBLE=\$(df -h | sed '1 d' | grep disk | awk '{print \$4}')**

2. **Espacio disponible de una partición predefinida.**

1. **ESPACIO\_PORCENTUAL=\$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print\$5}')**

2. **Espacio porcentual de una partición predefinida.**

1. **ESPACIO\_PORCENTUAL=\$(df -h | sed '1 d' | grep disk | awk '{print \$5}')**

2. **Espacio porcentual de una partición predefinida.**

1. **ESPACIO\_PUNTOMONTAJE=\$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print \$6}')**

2. **Espacio de un punto de montaje de una partición predefinida.**

1. **ESPACIO\_PUNTOMONTAJE=\$(df -h | sed '1 d' | grep disk | awk '{print \$6}')**

2. **Espacio de un punto de montaje de una partición predefinida.**

1. **ESPACIO\_TOTAL\_RAIZ=\$(df -h | sort -k 6 | sed -n '1p' | awk '{print \$2}')**

2. **Espacio total de la raíz " / " del Sistema Operativo.**

1. **ESPACIO\_USADO\_RAIZ=\$(df -h | sort -k 6 | sed -n '1p' | awk '{print \$3}')**

2. **Espacio usado de la raíz " / " del Sistema Operativo.**

1. **ESPACIO\_DISPONIBLE\_RAIZ=\$(df -h | sort -k 6 | sed -n '1p' | awk '{print \$4}')**

2. **Espacio disponible de la raíz " / " del Sistema Operativo.**

1. **ESPACIO\_PORCENTUAL\_RAIZ=\$(df -h | sort -k 6 | sed -n '1p' | awk '{print \$5}')**

2. **Espacio porcentual de la raíz " / " del Sistema Operativo.**

#+++++

#### VARIABLES CON INFORMACIÓN SOBRE APLICACIONES INSTALADAS EN EL SISTEMA OPERATIVO DEL EQUIPO USANDO DPKG O APTITUDE:

1. **VERSION\_SSH=\$(dpkg -l | grep "ssh " | egrep '(ii)' | sed -n '1p' | awk '{print \$3}')**

2. **Versión de un Paquete Instalado usando DPKG**

1. **VERSION\_LINUX\_H=\$(dpkg -l | grep linux-headers-`uname -r` | egrep '(ii)' | awk '{print \$3}')**

2. **Versión de un Paquete Instalado usando DPKG**

1. **VERSION\_SSH=\$(aptitude show ssh | sed -n '5p' | awk '{print \$2}')**

2. **Versión de un Paquete Instalado usando APTITUDE**

1. **VERSION\_LINUX\_H=\$(aptitude show linux-headers-`uname -r` | sed -n '5p' | awk '{print \$2}')**

2. **Versión de un Paquete Instalado usando APTITUDE**

1. **aptitude show \$VERSION\_SSH \$VERSION\_LINUX\_H | egrep '(Paquete|Estado|Versión)'**

2. **Ver información relativa sobre un paquete instalado utilizando las variables anteriores.**

1. **aptitude show \$VERSION\_SSH \$VERSION\_LINUX\_H | egrep '(Package|State|Version)'**

2. **Ver información relativa sobre un paquete instalado utilizando las variables anteriores.**

#+++++

#### VARIABLES CON INFORMACIÓN SOBRE PARAMETROS DE RED EN EL SISTEMA OPERATIVO DEL EQUIPO:

1. **PROXY\_GATEWAY=\$(route -n | sed -n '3p' | awk '{print \$2}')**

2. **Proxy / Gateway del Sistema Operativo.**

1. **DOMINIO=\$(cat /etc/resolv.conf | sed '2 d' | grep search | sed -n '1p' | awk '{print \$2}')**

2. **Dominio del Sistema Operativo.**



1. **IP\_SERVIDOR\_DNS=\$(cat /etc/resolv.conf | sed '1 d' | awk '{print \$2}')**

2. **IP del Servidor DNS del Sistema Operativo.**

1. **CONEXION\_INTERNET=(if ping -c 1 8.8.8.8 &> /dev/null; then RESULTADO=Habilitado; else RESULTADO=Deshabilitado; fi ; echo \$RESULTADO)**

2. **Estado de la Conexión a Internet.**

1. **TEST\_PING=\$(ping 8.8.8.8 -c 5 | grep packet | awk '{print \$6}' | cut -f1 -d%)**

2. **Validación de perdida de paquetes hacia Internet.**

1. **TEST\_LATENCIA=\$(ping 8.8.8.8 -c 5 | grep packet | awk '{print \$10}' | cut -f1 -d%)**

2. **Validación de Latencia de paquetes hacia Internet.**

1. **TEST\_LATENCIA=\$(ping 8.8.8.8 -c 1 | grep time= | awk '{print \$7}' | sed 's/time=//')**

2. **Validación de Latencia de paquetes hacia Internet.**

#+++++

1. **IP\_INTERNA=\$(hostname -I)**

2. **Dirección IP unica Interna del Equipo.**

1. **IP\_LAN=\$(hostname -I | awk '{print \$1}')**

2. **Primera Dirección IP (eth0 - LAN) del Equipo.**

1. **IP\_INTERNA=\$(hostname -I | awk '{print \$2}')**

2. **Segunda Dirección IP (eth1 - WAN) del Equipo.**

1. **IP\_ETH0=\$(ifconfig eth0 | grep inet | grep -v inet6 | cut -d ":" -f 2 | cut -d " " -f 1)**

2. **Dirección IP (eth0 - LAN) del Equipo.**

1. **IP\_ETH0=\$(/sbin/ifconfig | sed '/Bcast/!d' | awk '{print \$2}' | awk '{print \$2}' FS=":" | sed -n '1p')**

2. **Dirección IP (eth0 - LAN) del Equipo.**

1. **IP\_ETH0=\$(ifconfig|grep Bcast|awk ' { print \$2 }'|awk -F":" ' { print \$2 }' | sed -n '1p') ; echo \$IP\_ETH0**

2. **Dirección IP (eth0 - LAN) del Equipo.**

1. **MAC\_ETH0=\$(ifconfig eth0 | sed -n '1p' | awk '{print \$5}'); echo\$MAC\_ETH0**

2. **Dirección física MAC (eth0 - LAN) del Equipo.**

1. **IP\_ETH0=\$(ifconfig eth0 | grep inet | grep -v inet6 | cut -d ":" -f 2 | cut -d " " -f 1) && IP\_ETH0=\$(ifconfig eth0 | grep inet | grep -v inet6 | cut -d ":" -f 2 | cut -d " " -f 1) ; echo IP\_ETH0: \$IP\_ETH0 - MAC\_ETH0:**

**2. Aparejamiento de IP/MAC de red eth0 del Equipo.**

#+++++

**VARIABLES CON INFORMACIÓN SOBRE PARAMETROS DEL SERVIDOR SSH DEL SISTEMA OPERATIVO DEL EQUIPO:**

1. **PUERTO\_SSH**=\$(cat /etc/ssh/sshd\_config | egrep '(Port )' | sed 's/^\.//g' | awk '{print \$2}')

2. **Numero de Puerto configurado para conexiones ssh.**

1. **QUIENSSH\_IP**=\$(echo "\$SSH\_CONNECTION" | awk '{print \$1}')

2. **Dirección IP de quien esta logueado en el Sistema vía SSH.**

1. **QUIENSSH\_PUERTO**=\$(echo "\$SSH\_CONNECTION" | awk '{print \$2}')

2. **Puerto de conexión SSH de quien esta logueado en el Sistema.**

1. **USUARIOS\_CONECTADOS**=\$(who | awk '{print \$1}')

2. **Usuarios conectados vía ssh al Equipo.**

1. **USER\_ONLINE**=\$(who | awk '{print \$1}' | wc -w)

2. **Cantidad de Usuarios conectados vía ssh al Equipo.**

1. **USER\_ONLINE**=\$(top -n 1 -b | grep "load average:" | awk '{print \$6}')

2. **Cantidad de Usuarios conectados vía ssh al Equipo.**

1. **ID\_USERNAME\_SESSION**=\$(for i in \$(who | sed 1q | cut -c 1-8); do grep \$i /etc/passwd | awk '{print "\t" \$1}' | tr -d '[A-Za-z]' | sed 's/::x\*//' | cut -c 1-5 | awk '{print \$1}' ; done | sed 1q)

2. **ID del Nombre del primer usuario (alfabeticamente) detectado conectado (con sesión abierta) en el sistema.**

1. **USERNAME\_ID\_SESSION**=\$(for i in \$(who | sed 1q | cut -c 1-8); do grep \$i /etc/passwd | awk '{print "\t" \$1}' | tr -d '[A-Za-z]' | sed 's/::x\*//' | cut -c 1-5 | awk '{print \$2}' ; done ; echo \$i)

2. **Nombre del primer usuario (alfabeticamente) detectado conectado (con sesión abierta) en el sistema.**

#+++++

**VARIABLES CON INFORMACIÓN SOBRE PARAMETROS DE CARPETAS DEL SISTEMA DE FICHERO DEL SISTEMA OPERATIVO DEL EQUIPO:**

1. **NOMBRES\_CARPETAS**=\$(ls -l /home | sed '1 d' | awk '{print \$9}')

2. **Nombres de carpetas de usuarios creados.**

1. **NUMERO\_CARPETAS**=\$(ls -l /home | sed '1 d' | awk '{print \$9}' | wc -w )

2. **Numeros de carpetas de usuarios creados.**

**\$9}')****2. Usuario con más cantidad de archivos en la Carpeta HOME****1. DATA\_USUARIO1=\$(du -sh /home/\* | sort -r | sed q | awk '{print \$1}')****2. Tamaño de la carpeta de Usuario con más cantidad de archivos en la Carpeta HOME****1. DATA\_ROOT=\$(du -sh /root | awk '{print \$1}')****2. Tamaño de la carpeta del Superusuario root****1. DATA\_CARPETA1=\$(du -sh /var | awk '{print \$1}')****2. Tamaño de una Carpeta predefinida. Ejemplo: La carpeta /VAR**

#+++++

**VARIABLES CON INFORMACIÓN SOBRE PARAMETROS DE CARGA, PROCESOS Y TIEMPOS DEL SISTEMA OPERATIVO DEL EQUIPO:****1. CARGA\_1MIN=\$(top -n 1 -b | grep "load average:" | awk '{print \$12}' | sed 's/,//2')****2. Carga del Sistema Operativo en 1 Minuto, usando el comando top.****1. CARGA\_5MIN=\$(top -n 1 -b | grep "load average:" | awk '{print \$13}' | sed 's/,//2')****2. Carga del Sistema Operativo en 5 Minuto, usando el comando top.****1. CARGA\_15MIN=\$(top -n 1 -b | grep "load average:" | awk '{print \$14}' | sed 's/,//2')****2. Carga del Sistema Operativo en 15 Minuto, usando el comando top.****1. CARGA\_1MIN=\$(uptime | awk '{print \$10}' | sed 's/,//2')****2. Carga del Sistema Operativo en 1 Minuto, usando el comando uptime.****1. CARGA\_5MIN=\$(uptime | awk '{print \$11}' | sed 's/,//2')****2. Carga del Sistema Operativo en 5 Minuto, usando el comando uptime.****1. CARGA\_15MIN=\$(uptime | awk '{print \$12}' | sed 's/,//2')****2. Carga del Sistema Operativo en 15 Minuto, usando el comando uptime.****1. PROC\_ZOMBIE=\$(top -n 1 -b | grep "zombie" | awk '{print \$10}')****2. Cantidad de procesos Zombies.****1. TIEMPO\_ENCENDIDO=\$(uptime | awk '{print \$3,\$4}' | cut -f1 -d,)****2. Tiempo de encendido del Equipo.**

#+++++

**VARIABLES CON INFORMACIÓN SOBRE ELEMENTOS DE HARDWARE DEL SISTEMA OPERATIVO DEL EQUIPO:**

1. **FAB\_TVIDEO=\$(lspci -v | grep "VGA" | cut -d " " -f05)**

2. **Fabricante de la tarjeta de video.**

1. **MEM\_TVIDEO=\$(lspci -v -s `lspci | awk '/VGA/{print \$1}'` | sed -n '/Memory.\*/, prefetchable/s/.\*/[size=([^\]]+\s)M]/\1/p')**

2. **Cantidad de Memoria de la tarjeta de video.**

1. **DRV\_TVIDEO=\$(lspci -nnk | grep -i vga -A3 | grep 'in use' | cut -d " " -f05)**

2. **Controlador de la tarjeta de video.**

1. **A3D\_TVIDEO=\$(glxinfo | grep "direct rendering: Yes" | awk '{print \$3}')**

2. **Comprobación de la Habilitación de la Aceleración 3D de la tarjeta de video.**

#+++++

1. **FABRICANTE\_CPU=\$(grep "vendor\_id" /proc/cpuinfo | sed q | awk '{print\$3}')**

2. **Fabricante del Procesador (CPU) del Equipo.**

1. **MODELO\_CPU=\$(grep "model name" /proc/cpuinfo | sed q | cut -d ":" -f 2 | awk '{print \$0}')**

2. **Modelo del Procesador (CPU) del Equipo.**

1. **NUM\_CPU=\$(grep "processor" /proc/cpuinfo | sort -r | sed q | awk '{print \$3}'); TOTAL\_CPU=\$((`expr \$NUM\_CPU + 1`))**

2. **Cantidad de Procesadores (CPU) del Equipo.**

1. **NUM\_CPU=\$(grep "processor" /proc/cpuinfo | sort -r | sed q | awk '{print \$3}'); TOTAL\_CPU=\$((`expr \$NUM\_CPU + 1`)); NUCLEO\_CPU=\$(grep "cpu cores" /proc/cpuinfo | sed q | awk '{print \$4}'); TOTAL\_NUCLEO\_CPU=\$((`expr \$TOTAL\_CPU \\* \$NUCLEO\_CPU`))**

2. **Cantidad Total de Núcleos disponibles en los Procesadores (CPU) disponibles del Equipo.**

1. **CACHE\_CPU=\$(grep "cache size" /proc/cpuinfo | sed q | cut -d ":" -f 2 | awk '{print \$0}')**

2. **Cantidad de Memoria Caché de los Procesadores (CPU) del Equipo.**

#+++++

## VARIABLES CON INFORMACIÓN SOBRE X APLICACIÓN INSTALADA EN EL SISTEMA OPERATIVO DEL EQUIPO:

1. **NOMBRE\_ZABBIX=\$(cat /etc/zabbix/zabbix\_agentd.conf | egrep '(Hostname)' | sed '/#/ d')**

2. **Nombre configurado del Equipo en el Cliente ZABBIX instalado.**

12 de **VERSION\_VBOX=\$(dpkg -l | grep virtualbox | egrep '(ii)' | awk '{print\$2}')** 07/06/2019 12:38 PM

**2. Versión del VirtualBox instalado en el Equipo.**

1. **`VERSION_VBOX=$(dpkg -l | grep virtualbox | egrep '(ii)' | awk '{print $2}')`  
**`&& aptitude show $VERSION_VBOX | egrep '(Paquete|Estado|Versión)'`****
2. **Versión del VirtualBox instalado en el Equipo.**

1. **`VERSION_VBOX=$(dpkg -l | grep virtualbox | egrep '(ii)' | awk '{print $2}')`  
**`&& aptitude show virtualbox-5.0 | egrep '(Package|State|Version)'`****
2. **Versión del VirtualBox instalado en el Equipo.**

# PRONTO MUCHOS MÁS...

Aquí les dejo esta encuesta para medir el apoyo logrado:

#####

**¿Has encontrado agradable el Blog y útil su contenido?**

Si, es excelente. Es muy completo Si, es un buen Blog sobre Linux, Si, pero no me parece nada innovador. Lo seguiré y consultare Ciencia, Tecnología y Educación. asombroso. De vez en cuando lo frecuentemente. Lo visitare a menudo. revisaré.

No. Y no creo volver dentro de mucho tiempo a consultarlo! **Vote View Results Polldaddy.com (<https://polldaddy.com/signup-free/?ad=poll-front>)**

#####

**Recuerda donar por lo menos 1\$ US a este u otro Proyecto de apoyo o desarrollo de Software Libre, GNU, Linux o Código Abierto.**

Ayúdanos a mantener nuestro proyecto



([https://www.paypal.com/cgi-bin/webscr?cmd=s-xclick&hosted\\_button\\_id=M2JM3Y9XU9FYS](https://www.paypal.com/cgi-bin/webscr?cmd=s-xclick&hosted_button_id=M2JM3Y9XU9FYS))

Dona al Proyecto TIC – TAC

**Atrévete, usa Software Libre y GNU/Linux, sobre todo Canaima GNU / Linux, el Sistema Operativo Libre oficial de la República Bolivariana de Venezuela.**

**Todos a luchar por un Mundo Mejor.**

CREA TU PROPIO BLOG CON WORDPRESS.COM.