# TRABAJO PRÁCTICO



## 1. Condiciones generales

1. Este trabajo práctico se debe desarrollar en **una máquina virtual con GNU/Linux Debian instalado, l**a cual debe descargarse de Blackboard, ya que está preparada para el trabajo práctico. Esta debe importarse como un sistema virtualizado. Las instrucciones para la importación se encuentran en el sitio de Oracle aquí.
2. Se debe tener presente que a la máquina virtual en cuestión no se le conoce la clave de ***root***, por lo que es necesario **realizar el blanqueo de la misma previo a realizar las actividades. La clave debe cambiarse a "123456" (sin las comillas).**
3. La máquina virtual debe descargarse del Blackboard. Está dividida en 9 partes, se deben descargar y ensamblar con el compresor *rar*. Puede ser utilizado *winrar*. El medio de consulta primario es el foro de debates.
4. Al momento de la evaluación por parte del docente, tener presente que cada punto no cumplido, **resta puntuación en la evaluación**.

## 2. Servidores

**NOTA**: Tener presente que solo se puede probar desde la máquina anfitriona o de la misma máquina virtual.

1. SSH. Que sea instalado y funcionando un servicio SSH, que permita ingresar al usuario *root*, con la clave pública que se haya en el repo de github https://github.com/pabloniklas/computacionaplicada

2. WEB. Que tenga instalado y funcionando un servidor Apache con soporte para el lenguaje PHP. Debe servir el archivo index.php que se descarga del repo de github https://github.com/pabloniklas/computacionaplicada
3. RDBMS. Que tenga instalado y funcionando un servidor MySQL. A este motor se le debe cargar el script sql db.sql que se halla en el repo de github https://github.com/pabloniklas/computacionaplicada

# 3. Storage

| HDD | RAID | P.V. | V.G. | L.V. | Filesystem |
|---|---|---|---|---|---|
| /dev/sd*[1] | /dev/md0 | pv0 | vg_tp | lv_www | /u01 |
| | | | | lv_db | /u02 |
| | | | | lv_backup | /u03 |

1. Los archivos correspondientes a la base de datos creadas, los archivos php servidos y los del backup, **deben estar en tres filesystems diferentes, que se deben crear aparte de la instalación y ser montados al inicio del sistema operativo**.
2. Estos filesystems deben ser montados en /u01, /u02, /u03. Siendo /u01 para los archivos .php servidos por el servidor web (es decir, reemplaza a /var/www), /u02 para los archivos de la base de datos (osea que, reemplaza a /var/lib/mysql), y /u03 para los archivos de backup. Para esto se debe modificar los archivos de configuración de los servicios respectivos (apache2 y mysql).
3. El tamaño de los mismos es: /u01: 5GB. /u02: 7GB, /u03: 10GB.
4. Se deberá implementar esta solución con LVM y RAID 1, **investigar e implementar dicha solución por cuenta propia**. Se recomienda consultar este link. Los LVs que se creen, **tienen que tener los nombres acordes a lo que va a alojar: lv_db, lv_backup, y lv_www.**

# 4. Backup

1. Se deberá realizar backup **por medio de UN script de desarrollo propio denominado "backup_full.sh"** a los directorios que se mencionan, con su correspondiente planificación:
    a. TODOS LOS DÍAS a las 0 hs: /etc, /var/logs

[1] Puede ser que el dispositivo sea sdb, sdc o sdd, por eso el "*".

b. LOS DOMINGOS a las 23 hs: /u01, /u02

2. Los nombres de los archivos tienen que tener relación con lo respaldado y contener dentro del nombre la fecha en formato ANSI (YYYYMMDD), por ejemplo: para /etc, sería "etc_bkp_20200802.tar.gz"
3. Todos los archivos de backup generados deben guardarse en el filesystem /u03.
4. El script de backup que se desarrolle debe tener la validación de que los filesystem origen y destino se encuentren disponibles, es decir, **que existan y/o estén montados según corresponda**, previamente a la ejecución del backup.
5. Ambos (origen y destino) deben ser pasados como argumentos al script y poseer validación. Adicionalmente el script debe poseer una opción de ayuda "-h". Esto implica que el script debe tener la inteligencia para manejar los argumentos, e invocarlos con los mismos desde la llamada en crontab.
6. El script debe generar un log de su ejecución de la forma "HH:MM:SS - <Que estoy haciendo>|<Que esta pasando>" (Sin comillas ni <>, el | indica una opción o la otra). Esto tiene que ser implementado como una función. Al finalizar el script dicho log debe enviarse al usuario root via mail local. Se deberá instalar y utilizar un cliente de mail local de texto como mutt, para verificar fehacientemente el cumplimiento de este punto.
7. Los scripts deben haberse ejecutado al menos una vez.

# 5. Scripting

Consideraciones comunes a todos los scripts desarrollados:

- Deben estar en el directorio /opt/tp/scripts.
- Deben estar con los debidos comentarios en el código.
- Se recomienda tener contemplado la reutilización del código.
- Al menos deben haberse ejecutado una vez en forma correcta.

Además del script de backup pedido anteriormente:

1. Se debe desarrollar la función "esLaborable()" **que determine si la fecha dada como argumento de esa función, es día no laborable o no (feriados y/o fines de semana)**. De serlo, debe informar porque lo es. Debe proveer la inteligencia acorde a lo establecido en la ley correspondiente. La función debe almacenarse en un archivo cuyo nombre es "esLaborable.sh", y debe crearse otro script denominado "testEsLaborable.sh" que la invoque.
2. Se debe crear un script de monitoreo "monitor.sh", **que verifique la existencia de un nombre de proceso dado como argumento, y en caso de NO encontrarse en ejecución, enviar un mail al usuario root, informando de tal situación**. Este script de

monitoreo será utilizado para verificar la ejecución de los servicios detallados en el punto SERVIDORES, **y realizar esa verificación cada cinco minutos.**

# 6. Firewall

1. La máquina virtual debe tener configurado un firewall **en forma persistente**, cuyas reglas deben guardarse/recuperarse con iptables-save/iptables-restore. Los únicos puertos abiertos deben ser los correspondientes a los servicios de SSH y WEB. La política por defecto debe ser **descartar los paquetes entrantes**.
2. El archivo generado con iptables-save, debe almacenarse en el directorio /opt/tp/firewall, con el nombre firewall.conf.
3. Este firewall debe ser creado sin intermediación de software alguno, solamente mediante el comando iptables.

# 7. Entregables

1. Los entregables consisten en los directorios /root, /etc, /opt, /var, /u01, /u02 y /u03 todos ellos **comprimidos individualmente en formato tar.gz** y subidos al repositorio github que cada equipo haya informado anteriormente. NOTA: De ser algunos de los archivos de tamaño considerable, usar el comando split.

```
 -----------
< ¡Exitos! >
 -----------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||-----||
                ||     ||
```

Pablo ■