

## Introducción a la Informática

# Ejercitación

En las mesas de trabajo realizamos los siguientes pasos:

Procederemos a instalar Git a través de la consola de comandos.

- Lo primero que debemos hacer es verificar que tengamos conexión a Internet, para esto utilizaremos el comando **ping**:

**ping -c 2 [www.digitalhouse.com.ar](http://www.digitalhouse.com.ar)**

```
usuario@ubuntu-intro:~$ ping -c 2 www.digitalhouse.com.ar
PING digitalhouse.com.ar (54.37.156.117) 56(84) bytes of data.
64 bytes from 117.ip-54-37-156.eu (54.37.156.117): icmp_seq=1 ttl=45 time=279 ns
64 bytes from 117.ip-54-37-156.eu (54.37.156.117): icmp_seq=2 ttl=45 time=292 ns

--- digitalhouse.com.ar ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ns
rtt min/avg/max/mdev = 279.946/286.415/292.884/6.469 ns
usuario@ubuntu-intro:~$
```

- Si existiera algún tipo de problema, revisar que el tipo de conexión de la MV esté en modo nat.
- Si no ingresamos con el usuario root, podemos cambiarnos al mismo utilizando el comando **su root**, a continuación, debemos introducir la contraseña establecida. Si la máquina virtual es nueva deberían agregar la clave root con el comando **sudo passwd root**.

```

usuario@ubuntu-intro:~$ sudo passwd root
[sudo] password for usuario:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
usuario@ubuntu-intro:~$ su root
Contraseña:
root@ubuntu-intro:/home/usuario# _

```

- En caso de tener que agregarlo recomendamos que la contraseña sea **root** (usuario: root contraseña: root)
- Ahora procedemos a instalar Git a través del comando: **apt-get install git**
  - Puede que en algún momento nos pida una confirmación para seguir la instalación. Basta con poner la letra **S** o **Y** seguido de **Enter** para continuar.
- Luego probamos que Git haya sido instalado a través del comando **git --version**

```

root@ubuntu-intro:/home/usuario# git --version
git version 2.7.4

```

- En **formato TXT**, resolver el **siguiente cuestionario**:

**Con toda la mesa de trabajo debatan sobre las siguientes preguntas y contesten en conjunto:**

- ¿Por qué un lenguaje de programación sólo puede utilizarse en algunos sistemas operativos y en otros no?.

Porque un lenguaje de programación compila a lenguaje ensamblador el cual es ejecutado por el sistema operativo, este lenguaje ensamblador depende de 2 cosas, de la arquitectura del procesador y la interpretación que le de el sistema operativo. Si en la computadora cambia la arquitectura del procesador el lenguaje ensamblador es totalmente diferente y no va a correr.

- ¿Qué tipo de máquina virtual es virtualBox?.

**VirtualBox es una máquina virtual de sistemas. Permite crear sistemas virtuales indistintamente del sistema operativo que estemos utilizando.**

- Si tengo más de una máquina virtual instalada, y una se rompe, ¿esto afecta a las demás? ¿por qué?

**R: Si una máquina virtual se rompe o experimenta un problema, generalmente no afectará a las demás máquinas virtuales que estén instaladas en el mismo sistema. Esto se debe a que cada máquina virtual se ejecuta de manera aislada y se considera como un entorno independiente dentro del hipervisor.**

- Subir este archivo a la mochila del viajero.

**Sacar una captura de pantalla de los commits hechos y el cuestionario resuelto y subirlos a la mochila**

**3**

### **Opcional:**

- Clonar la mochila del viajero personal dentro de la Máquina virtual y subir el archivo de la ejercitación desde la misma.
- Para crear el TXT debemos usar el comando **touch** y luego modificarlo a través de **GNU Nano**.
- **EN CASO DE NECESITAR TOKEN PARA HACER PUSH, SOLO PUEDE SER COLOCADO COPIANDO LETRA POR LETRA DEL MISMO**

```

[sudo] password for bernardo:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
bernardo@ubuntu-CLI:~$ apt-get install git
E: No se pudo abrir el fichero de bloqueo «/var/lib/dpkg/lock-frontend» - open (13: Permiso denegado)
)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
bernardo@ubuntu-CLI:~$ y
y: no se encontró la orden
bernardo@ubuntu-CLI:~$ root
El programa «root» no está instalado. Puede instalarlo escribiendo:
sudo apt install root-system-bin
bernardo@ubuntu-CLI:~$ su root
Contraseña:
root@ubuntu-CLI:/home/bernardo# apt-get install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch
  git-cvs git-mediawiki git-svn
Se actualizarán los siguientes paquetes:
  git
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 183 no actualizados.
Se necesita descargar 3 312 kB de archivos.
Se utilizarán 32.8 kB de espacio de disco adicional después de esta operación.
Des:1 http://mx.archive.ubuntu.com/ubuntu xenial-updates/main i386 git i386 1:2.7.4-0ubuntu1.10 [3 3
12 kB]
Descargados 3 312 kB en 1s (1 880 kB/s)
(Leyendo la base de datos ... 58020 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../git_1:2.7.4-0ubuntu1.10_i386.deb ...
Desempaquetando git (1:2.7.4-0ubuntu1.10) sobre (1:2.7.4-0ubuntu1.6) ...
Configurando git (1:2.7.4-0ubuntu1.10) ...
root@ubuntu-CLI:/home/bernardo# git --version
git version 2.7.4
root@ubuntu-CLI:/home/bernardo#

```

```

(Reading database ... 53075 files and directories currently installed.)
Preparing to unpack .../perl-modules-5.22_5.22.1-9ubuntu0.9_all.deb ...
Unpacking perl-modules-5.22 (5.22.1-9ubuntu0.9) ...
Selecting previously unselected package libperl5.22:i386.
Preparing to unpack .../libperl5.22_5.22.1-9ubuntu0.9_i386.deb ...
Unpacking libperl5.22:i386 (5.22.1-9ubuntu0.9) ...
Selecting previously unselected package perl.
Preparing to unpack .../perl_5.22.1-9ubuntu0.9_i386.deb ...
Unpacking perl (5.22.1-9ubuntu0.9) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../liberror-perl_0.17-1.2_all.deb ...
Unpacking liberror-perl (0.17-1.2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1:2.7.4-0ubuntu1.10_all.deb ...
Unpacking git-man (1:2.7.4-0ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1:2.7.4-0ubuntu1.10_i386.deb ...
Unpacking git (1:2.7.4-0ubuntu1.10) ...
Selecting previously unselected package patch.
Preparing to unpack .../patch_2.7.5-1ubuntu0.16.04.2_i386.deb ...
Unpacking patch (2.7.5-1ubuntu0.16.04.2) ...
Selecting previously unselected package rename.
Preparing to unpack .../archives/rename_0.20-4_all.deb ...
Unpacking rename (0.20-4) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up perl-modules-5.22 (5.22.1-9ubuntu0.9) ...
Setting up libperl5.22:i386 (5.22.1-9ubuntu0.9) ...
Setting up perl (5.22.1-9ubuntu0.9) ...
update-alternatives: using /usr/bin/rename to provide /usr/bin/rename (rename) in auto mode
Setting up liberror-perl (0.17-1.2) ...
Setting up git-man (1:2.7.4-0ubuntu1.10) ...
Setting up git (1:2.7.4-0ubuntu1.10) ...
Setting up patch (2.7.5-1ubuntu0.16.04.2) ...
Setting up rename (0.20-4) ...
update-alternatives: using /usr/bin/file-rename to provide /usr/bin/rename (rename) in auto mode
Processing triggers for libc-bin (2.23-0ubuntu11) ...
root@martin:/home/martin#

```

