



**Tecnológico de Costa Rica**  
**Área académica de Ingeniería En Computadores**

**Laboratorio 1 - Conociendo Linux y las máquinas virtuales**

**Curso**

CE 4303 - Principios de Sistemas Operativos

**Realizado por**

Luis López Salas - 2015088115

**Profesor**

Jason Leitón Jiménez

*Marzo 2021*

## Teoría

1. Mencione 3 máquinas virtuales que se utilizan actualmente y haga un cuadro comparativo con las ventajas y desventajas de cada una de ellas.

VM	Ventajas	Desventajas
VirtualBox	<ul style="list-style-type: none"><li>• Es gratis.</li><li>• Capaz de soportar una gran variedad de sistemas operativos.</li><li>• Fácil de usar.</li><li>• Es simple.</li><li>• Está actualizado para los sistemas operativos más recientes.</li></ul>	<ul style="list-style-type: none"><li>• Tiene problemas con el modo de pantalla entera.</li><li>• Es simple, lo que puede provocar problemas si se desea hacer cambios.</li><li>• Es poco eficiente.</li><li>• Es capaz de asignar más recursos que la máquina host.</li></ul>
VMware	<ul style="list-style-type: none"><li>• Es capaz de correr varias máquinas virtuales a la vez.</li><li>• Tiene la capacidad de hacer rollback.</li><li>• Agregar/borrar una nueva máquina virtual es simple.</li><li>• Es capaz de limitar los recursos de cada máquina.</li></ul>	<ul style="list-style-type: none"><li>• Su rendimiento no es muy alto. Está diseñado para sistema de gama alta.</li><li>• No tiene tantas opciones para hacer ajustes.</li><li>• No es posible extraer la máquina virtual directamente.</li><li>• Falla de vez en cuando con las redes.</li><li>• Es difícil solucionar problemas de algunas partes específicas.</li></ul>
Citrix	<ul style="list-style-type: none"><li>• Corre de forma remota.</li><li>• Relativamente barato.</li><li>• Permite virtualizar sólo los apps, pero también una máquina virtual.</li><li>• Deployment de varios apps.</li><li>• Datos de forma centralizada.</li></ul>	<ul style="list-style-type: none"><li>• Es dependiente de la red.</li><li>• Es de paga.</li><li>• Los datos no están en el host.</li><li>• Difícil solucionar problemas, más debido a que está de forma remota.</li><li>• Puede ser lento dependiendo en que se esté realizando.</li><li>• No es tan bueno para MacOS</li></ul>

2. Cual es la diferencia entre simular, emular y virtualizar.

Emulación: Copia por completo el programa o su comportamiento. Esto lo puede hacer convirtiendo el código assembly de una arquitectura a otra (emulación a bajo nivel). Otra opción es que puede aproximarse por medio de funciones predefinidas y se asemeja por medio de estas (emulación de alto nivel). [1]

Simulación: Es similar a la emulación, pero este no intenta copiar el hardware de la máquina original. Si no el comportamiento como tal.

Virtualización: Permite la creación de varios ambientes simulado[2] desde un solo sistema físico. Para esto divide los recursos del sistema y le da acceso a las aplicaciones que lo requieran.

3. ¿Por qué se dice que una máquina virtual no es una simulación de una PC?  
Una máquina virtual no es una simulación de una PC ya que está emulando el hardware. Esto se debe a un concepto de encapsulamiento, donde la máquina virtual es independiente.

4. Busque el comando para instalar SSH en un entorno de Linux, así como el de generar las claves.

Para instalar el el ssh: `sudo apt-get install openssh-server`.

Para generar las claves: `ssh-keygen -t rsa`

5. Para qué se utiliza SSH, explique su funcionamiento y brinde 3 usos del mismo. El ssh se utiliza para la conexión entre un cliente y un servidor (este es su esquema). La diferencia importante es que el ssh es encriptado. 3 usos son:

- Se puede utilizar para compartir datos.
- Se puede utilizar para controlar un sistema de manera remota.
- Permite tunelaje, el uso de un protocolo que la red no permite. E.g. IPv6 en Plv4.

6. Explique 1 desventaja de las máquinas virtuales remotas.

Que depende de las conexiones. Si se trabaja por medio de internet y hay algún problema de conexion sea por parte del servidor o el cliente no se puede utilizar esta (lo cual indica doble punto de fallo). Hay que estar al tanto de que puede haber problemas por un sin número de problemas. Esto se mitiga si se utiliza en una red local, sin embargo no es infalible.

## Creación y conexión de la máquina virtual remota

Creación de máquina virtual (la llave pública se creó previamente utilizando el comando del punto 4):

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Azure subscription 1

Resource group \* ⓘ

(New) LuisLResourceGroup

[Create new](#)

### Instance details

Virtual machine name \* ⓘ

Operativos-LuisL

Region \* ⓘ

(US) East US

Availability options ⓘ

No infrastructure redundancy required

Image \* ⓘ

 Ubuntu Server 18.04 LTS - Gen1

[See all images](#)

Size \* ⓘ

Standard\_D2s\_v3 - 2 vcpus, 8 GiB memory (\$70.08/month)

[See all sizes](#)

### Administrator account

Authentication type ⓘ

- ☒ SSH public key  
☐ Password

**i** Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username \* ⓘ

LuisL ✓

SSH public key source

Use existing public key ▼

SSH public key \* ⓘ

/aGhqkrErVmbeM6KSik8LGDUjKu3dySqSIlBazly0XazinhZWrZia8s7CJ5Xi3cs  
u+oxgq3h4W+EmhlvtbvhsPX3cnODsNMQLz12idnzvXiwlO5ZB9fqly16eSPlo  
9UIO62KGenfV1olipBs+ xRP8e+ pSW5IARbpic5WSdAxcc+ 15ecvraAtzVPfGly  
6s= luis@luis-virtual-machine ✓

**i** [Learn more about creating and using SSH keys in Azure](#) ↗

### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ

- ☐ None  
☒ Allow selected ports

Select inbound ports \*

SSH (22) ▼

**⚠** **This will allow all IP addresses to access your virtual machine.** This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Conexión a la maquina virtual desde el sistema local:

```

luis@luis-virtual-machine:~/.ssh$ ssh -i ~/.ssh/id_rsa LuisL@52.146.41.123
The authenticity of host '52.146.41.123 (52.146.41.123)' can't be established.
ECDSA key fingerprint is SHA256:gUxLDpalUnjY9Jjb6DNP0JaWbx0gW3Hxxd596ksKt04.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '52.146.41.123' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1039-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Mar  8 21:19:40 UTC 2021

System load:  0.09               Processes:            129
Usage of /:   4.5% of 28.90GB    Users logged in:     0
Memory usage: 2%                IP address for eth0: 10.0.0.4
Swap usage:   0%

0 packages can be updated.
0 of these updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

```

Hostname y uname:

```

LuisL@Operativos-LuisL:~$ hostname
Operativos-LuisL
LuisL@Operativos-LuisL:~$ uname -a
Linux Operativos-LuisL 5.4.0-1039-azure #41~18.04.1-Ubuntu SMP Mon Jan 18 14:00:
01 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
LuisL@Operativos-LuisL:~$

```

## Programa en C y en Python

Código en C:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  double factorial(double ans);
6
7  int main (){ //Main function from where the system is called
8      char buffer[256];
9      printf ("Enter a number: ");
10     fgets (buffer, 256, stdin); //recieve number
11     int factnum = atoi (buffer);
12     double ans = (double) factnum;
13     double start = clock();
14     ans = factorial(ans); //call function
15     double end = clock();
16     printf("El factorial es de %lf \n",ans);
17     double cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
18     printf("Tiempo de duraci6n %lf \n",cpu_time_used);
19 }
20
21 double factorial(double ans){
22     //ans will be the answer
23     if(ans == 1)
24         return ans;
25     return ans * factorial(ans - 1);
26 }
```

Compilaci6n y ejecuci6n ambos duraron lo mismo:

```
luis@operativos-luis:~/Ejemplo$ gcc factorial.c -o factorial
luis@operativos-luis:~/Ejemplo$ ./factorial
Ingrese un n6mero: 100
El factorial es de 93326215443944102188325666108575267240944254854960571509166910400407995064242937148632694030450512898042989296944474898258737204311236641477561877016501813248.000000
Tiempo de duraci6n 0.000002
luis@operativos-luis:~/Ejemplo$
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ gcc factorial.c -o factorial
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Ingrese un n6mero: 100
El factorial es de 93326215443944102188325666108575267240944254854960571509166910400407995064242937148632694030450512898042989296944474898258737204311236641477561877016501813248.000000
Tiempo de duraci6n 0.000003
```

5 corridas en la m6quina local:

```
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ gcc factorial.c -o factorial
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Enter a number: 1234
El factorial es de inf
Tiempo de duraci6n 0.000027
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Enter a number: 100
El factorial es de 93326215443944102188325666108575267240944254854960571509166910400407995064242937148632694030450512898042989296944474898258737204311236641477561877016501813248.000000
Tiempo de duraci6n 0.000003
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ gcc factorial.c -o factorial
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ gcc factorial.c -o factorial
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Ingrese un n6mero: 100
El factorial es de 93326215443944102188325666108575267240944254854960571509166910400407995064242937148632694030450512898042989296944474898258737204311236641477561877016501813248.000000
Tiempo de duraci6n 0.000003
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Ingrese un n6mero: 66
El factorial es de 544344939077443069445496060275635856761283034568718387417404234993819829995466026946857533440.000000
Tiempo de duraci6n 0.000003
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Ingrese un n6mero: 89
El factorial es de 16507955160908452497218052643056785820348586593118454131228292379023767876373998621963681432374396360185860939835599722920061648806071040.000000
Tiempo de duraci6n 0.000005
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Ingrese un n6mero: 106
El factorial es de 114620856373470783195262178791869885158372134974981857049798601542754676287616326162938700437174657932394775194874661651509774298443489164851042101334162066851708352331776.000000
Tiempo de duraci6n 0.000004
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ ./factorial
Ingrese un n6mero: 155
El factorial es de 4789142901463391195071321120698705430443839346927766590243024791904930493455109325996023636166356544252726245873730366125871827618421207030650513093306720390752236820198520765
00372117214987400110541168443256609744538354280562895021882635083915363955979735593929220590403584.000000
Tiempo de duraci6n 0.000004
```

5 corridas en la máquina remota:

```

luisl@operativos-Luisl:~/Ejemplo$ gcc factorial.c -o factorial
luisl@operativos-Luisl:~/Ejemplo$ ./factorial
Ingrese un número: 100
El factorial es de 933262154439441021832150601085752672494425485496057150916691040004799506424293714863269403045051289804298929694447489825873720431123664147756187016501813248.000000
Tiempo de duración 0.000002
luisl@operativos-Luisl:~/Ejemplo$ ./factorial
Ingrese un número: 66
El factorial es de 544344939077443069445496060275635856761283034568718387417404234993819829995466026946857533440.000000
Tiempo de duración 0.000003
luisl@operativos-Luisl:~/Ejemplo$ ./factorial
Ingrese un número: 89
El factorial es de 16507955160908452497218052643056785820348586593118454131228292379023767876373998621963681432374396360185860939835599722920061648806871040.000000
Tiempo de duración 0.000004
luisl@operativos-Luisl:~/Ejemplo$ ./factorial
Ingrese un número: 106
El factorial es de 114628956373470783195262178791869885150372134974981857049798601542754676287616326162938700437174657932394775194874661651509774290443489164051042104316200605170035231776.000000
Tiempo de duración 0.000003
luisl@operativos-Luisl:~/Ejemplo$ ./factorial
Ingrese un número: 155
El factorial es de 478914290146339119507132112069870543044383934692770659024302479190493049345510932599062363616635654424527262458737303661258718276184212070306505130933067203907522368201985207650037211214987400110541168443250609744538354280562895021882635083915363955979735593929220590403584.000000
Tiempo de duración 0.000015

```

## Python:

```
import time
def factorial(ans):
    if(ans == 1):
        return 1
    return ans*factorial(ans-1)

print('Ingrese un número: ')
ans = input()
start=time.time()
ans = factorial(int(ans))
print('El factorial es de '+str(ans))
print('Tiempo de duración '+str(time.time() - start))
```

5 corridas locales:

[illegible]

### 5 corridas en máquina remota:

[illegible]

Como se observa en la mayoría de los casos, la máquina local ejecutó el programa más rápido. Realmente es un proceso un poco relativo pero se mantiene un poco similar. Claramente los resultados pueden variar para una aplicación más compleja. Hay muchos factores determinantes en cómo una máquina ejecuta código, desde el clima dónde está (si está caliente, humedo...) hasta cómo está configurada (cuantos recursos tiene, la calidad de las partes que utiliza...). Sin embargo, relativamante ambos se encuentran en un rango aceptable y de usarse para aplicaciones como la del ejercicio no hay gran problema, debico incluso a que no se nota un lag.

Copiando de máquina local a máquina virtual

```
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ scp factorial.c LuisL@52.146.41.123:/home/LuisL/Ejemplo
factorial.c                                100% 730      0.7KB/s   00:00
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$ scp factorial.py LuisL@52.146.41.123:/home/LuisL/Ejemplo
factorial.py                              100% 280      0.3KB/s   00:01
luis@luis-virtual-machine:~/Documents/SistemasOperativos/Lab1$
```

Resultado:

```
LuisL@Operativos-LuisL:~/Ejemplo$ ls
factorial factorial.c factorial.py
```



## Referencias:

- [1] Team, T. (2019, May 22). Virtualization vs. emulation vs. simulation: What's the difference? Retrieved March 08, 2021, from <https://www3.technologyevaluation.com/sd/category/virtualization-virtual-machine/articles/virtualization-vs-emulation-vs-simulation>
- [2] What is virtualization? (n.d.). Retrieved March 08, 2021, from <https://www.redhat.com/en/topics/virtualization/what-is-virtualization>
- [3] Nakivo. (2020, November 24). A complete comparison of vmware and virtualbox. Retrieved March 09, 2021, from <https://www.nakivo.com/blog/vmware-vs-virtual-box-comprehensive-comparison/>