

Solucionario Cuaderno

Tareas Recuperación

Javier Arturo Lucero Ilabaca

26 de octubre del 2023

Índice

Índice	1
Introducción	2
Programación con Python	2
Descripción de algoritmos.	2
Código 1	2
Código 2	3
Código 3	5
Desarrollo de algoritmos	6
Enlace a códigos	6
Enunciado 1	6
Enunciado 2	7
Enunciado 3	7
Enunciado 4	8
Enunciado 5	9
LINUX	10
Supuesto práctico linux	10
Comandos LINUX	10
Ejercicio 1	10
Ejercicio 2	10
Ejercicio 3	11
Windows	14
Supuestos práctico	14
Gestión de recursos y otros	14
Comandos gestion	14
Licencia 1	15
Licencia 2	16
Virtualización	17
Instalación dual	17
Red NAT	17

Introducción

Este cuaderno será dividido en 5 puntos principales siendo los bloques a resolver, subiendo con capturas y texto, simultáneamente creando un árbol de directorios en github donde estarán los archivos de todos los ejercicios.

Programación con Python

Nos encontramos con dos apartados.

Descripción de algoritmos.

Código 1

```
Proceso un_proceso
  definir a, b, c, final Como Entero
  final = 1
  Mientras final ≠ 0 Hacer
    Escribir "escribir tres números distintos, por favor"
    leer a,b,c
    Si a>b Entonces
      Si a>c Entonces
        Escribir " a mayor " a
      Sino
        Escribir "c mayor " c
      Fin Si
    Sino
      Si b>c Entonces
        Escribir "b mayor " b
      Sino
        Escribir "c mayor " c
      Fin Si
    Fin Si
    Escribir "quiere finalizar el proceso teclea 0  ?"
    Leer final
  Fin Mientras
FinProceso
```

Enunciado:

#Realiza un script en python que compare 3 números insertados por el usuario, en un bucle donde nos de el mayor de los 3 hasta que el usuario ingrese el 0 y finalice

```
while final != 0:
    print("Escribe tres numeros distintos")
    a = int(input("Ingresa el primer numero: "))
    b = int(input("Ingresa el segundo numero: "))
    c = int(input("Ingresa el tercer numero: "))
    if a > b:
        if a > c:
            print("a es el mayor:", a)
        else:
            print("c es el mayor:", c)
    else:
        if b > c:
            print("b es el mayor:", b)
        else:
            print("c es el mayor:", c)

    print(" Teclee 0 para salir o cualquiera para continuar.")
    final = int(input())

proceso()
```

Código 2

```
Proceso un_proceso
  Definir a,r,c Como Entero
  definir pr Como Caracter
  escribir " desea ejecutar el programa s/n"
  leer pr
  si pr = "s" entonces
    c ← 0
    Repetir
      Leer a
      c ← c+1
      r ← a MOD 2
      Si r=0 Entonces
        Escribir a, ' seleccionado'
      SiNo
        Escribir a, ' no seleccionado'
      FinSi
      Escribir r
    Hasta Que c=10
  FinSi
  escribir "ya terminamos"
FinProceso
```

Enunciado:

#Realizar un script en python que pida al usuario si quiere o no realizar el programa y si lo ejecuta pida un numero, con ese número el programa le dirá si es divisible entre 2, y repetiremos el proceso hasta en 10 ocasiones

```
if pr.lower() == "s":
    c = 0
    while c < 10:
        a = int(input("Ingrese un número: "))
        c += 1
        r = a % 2
        if r == 0:
            print(f"{a} seleccionado")
        else:
            print(f"{a} No seleccionado")
        print(r)

    print("YA TERMINAMOS")
```

Código 3

```
Proceso un_proceso
    definir i, j, r Como Entero
    definir tecla Como Caracter
    i=1
    j=1
    mientras i≠ 0
        escribir "Escribe numero. Pulsar 0 para fin"
        leer i
        si i≠0 entonces
            mientras j<11
                r = i*j
                escribir i " x " j "=" r
                j=j+1
            finmientras
        fin si
        j=1
    fin mientras
FinProceso
```

Enunciado:

#Realiza un script que de las tablas de multiplicación del número que ingrese el usuario, tantas veces que quiera el usuario hasta que finalice el programa pulsando el 0

```
def un_proceso():
    i = 1
    j = 1

    while i != 0:
        print("Escribe número o pulsa 0 para finalizar")
        i = int(input())

        if i != 0:
            while j < 11:
                r = i * j
                print(f"{i} x {j} = {r}")
                j = j + 1
            j=1

un_proceso()
```

Desarrollo de algoritmos

Enlace a códigos

https://github.com/ArturoLucero28/Lucero_Arturo_ReclSO/tree/bf0a21ca0f8ef372609fc5cf1bf789dade7380fa/Python/Desarrollo%20de%20enunciados

Enunciado 1

Desarrolla un algoritmo que pida al usuario las calificaciones de 5 módulos de un alumno. El algoritmo mostrará en pantalla la calificación mayor, la calificación menor, y la media de las calificaciones.

```
def enunciado1():
    #Creamos una variable en modo lista
    notas = []
    print("Ingresa las calificaciones de los 5 modulos:")
    #un bucle de rango 5 al pedirnos 5 modulos
    for i in range(5):
        modulo = float(input(f"Ingresa una nota: "))
        notas.append(modulo)
        i= i +1
    #Buscamos el mayor,menos y suma de la lista
    mayor = max(notas)
    menor = min(notas)
    suma = sum(notas)
    #Y para la media la conseguimos haciendo la suma de todos diviendolo entre todos los modulos
    media = suma / 5

    print(f"Calificacion mayor: {mayor}")
    print(f"Calificacion menor: {menor}")
    print(f"Media de calificaciones: {media}")

enunciado1()
```

Enunciado 2

Escribe un algoritmo que, dado un número introducido por el usuario, indique si el número es capicua o no.

```
#Crear variables y asignar cadena el numero
numero=input("Introduce un número:")
cadena=str(numero)
#Se invierte
numero_invertido=cadena[::-1]
if cadena == numero_invertido:
    print("El numero es capicúa")
else:
    print("El numero no es capicúa")
```

Enunciado 3

Implementa un algoritmo que busque el nombre de una ciudad entre un conjunto de diez ciudades predefinidas. Al inicio se creará el conjunto con el nombre de las diez ciudades. Posteriormente se pedirá el nombre de una ciudad al usuario. El programa indicará si la ciudad se encuentra en el conjunto o no. Deberá dar por válidas entradas en minúsculas, mayúsculas. Por ejemplo, Si tenemos almacenada “Mérida”, se dará por válido “Mérida”, “Mérida”, “MÉRIDA”, “Mérida”, etc.

```
1 #Introducimos las ciudades en una lista
2 ciudades = {"Mérida","Montijo","Guadiana","Caceres","Puebla de la calzada","Lácar","Barbaño","Valdelacalzada","La Roca","Calamonte"}
3 ciudad = input("Introduce el nombre de la ciudad")
4 #para que de igual como lo ingresa el usurio hacemos lower
5 ciudad=ciudad.lower()
6 #verificamos
7 if ciudad in ciudades:
8     print("La ciudad está")
9 else:
10    print("La ciudad no está")
```


Enunciado 4

Desarrolla un programa que pida al usuario un número del 1 al 50 y que muestre la representación de dicho número en binario, octal y hexadecimal.

```
1  #pedir numero
2  numero=int(input("Introduce un numero del 1 al 50"))
3  if 0 < numero < 50:
4      #usamos funciones preestablecidas
5      binario= bin(numero)
6      octal= oct(numero)
7      hexadecimal=hex(numero)
8      print(f"Representacion binaria: {binario}")
9      print(f"Representacion octal: {octal}")
10     print(f"Representacion hexadecimal: {hexadecimal}")
11 else:
12     print("Ese numero no se puede")
```

Enunciado 5

Implementa un programa que pida al usuario una cantidad de dinero hasta que dicha cantidad sea múltiplo de 5 y, como máximo 3000. Una vez que se cumplan estas condiciones se tiene que mostrar al usuario el número de billetes de 100, 50, 20, 10 y 5 que se necesitan para obtener la cantidad de dinero indicada.

```
cantidad = 0
#bucle que tiene que cumplirse que el resto de cantidad entre 5 sea diferente que 0 o cantidad este entre 0 y 3000
while cantidad % 5 != 0 or cantidad <= 0 or cantidad > 3000:
    cantidad = int(input("Introduce una cantidad de dinero (múltiplo de 5 y como máximo 3000): "))
#realizamos los calculos
billetes_100 = cantidad // 100
cantidad %= 100
billetes_50 = cantidad // 50
cantidad %= 50
billetes_20 = cantidad // 20
cantidad %= 20
billetes_10 = cantidad // 10
cantidad %= 10
billetes_5 = cantidad // 5
#Y los mostramos en pantalla
print(f"La cantidad de ${cantidad}")
print(f"- {} billetes de 100".format(billetes_100))
print(f"- {} billetes de 50".format(billetes_50))
print(f"- {} billetes de 20".format(billetes_20))
print(f"- {} billetes de 10".format(billetes_10))
print(f"- {} billetes de 5".format(billetes_5))
```

LINUX

Supuesto práctico usuarios, grupos y permisos(vídeo en github)

Comandos LINUX

Ejercicio 1

Indica las instrucciones necesarias para mostrar en la pantalla, en el formato indicado, la siguiente

información de tu usuario:

Nombre de usuario: **\$USER**

Id de usuario: **id -u**

Id de grupo: **id -g**

Ruta de su directorio personal: **\$HOME**

Ejercicio 2

sudo useradd [opciones] [nombre]

usaremos en opciones -c para añadir nombre y apellido,luego el -e YYYY-MM-DD nombre para añadir la fecha de caducidad y con el -m -d es por si la ruta no existe la crearemos.

sudo useradd -m -d /ruta/ -c "nombre apellido" -e YYYY-MM-DD nombre

Con el sudo obligamos a que realice la opción y el usuario añadimos el usuario.

sudo userdel -r nombredeusuario

usamos userdel para eliminar el usuario añadiendo el -r [como opción] y el nombre del usuario

Ejercicio 3

- Crear un directorio dentro de Documentos llamado ejercicio permisos

```
a123@123:~$ mkdir ~/Documentos/ejercicio_permisos
a123@123:~$ cd ~/Documentos/ejercicio_permisos
a123@123:~/Documentos/ejercicio_permisos$
```

- Dentro del directorio crear 9 archivos (archiv 1,archiv 2,etc.)

```
a123@123:~/Documentos/ejercicio_permisos$ touch archiv{1..9}
a123@123:~/Documentos/ejercicio_permisos$ ls
archiv1 archiv2 archiv3 archiv4 archiv5 archiv6 archiv7 archiv8 archiv9
```

- Establecer a cada archivo los siguientes permisos:

- archiv1 -rwx-----
- archiv2 -rw-----
- archiv3 -rwxrwxrwx
- archiv4 -rwxrw-r--
- archiv 5 -drwxr-----
- archivo 6 -r-xr-r--
- archiv7 -r-----x
- archiv8 -rw-r—r--
- archiv9 -rw-rw-r—

```

a123@123:~/Documentos/ejercicio_permisos$ chmod 700 archiv1
a123@123:~/Documentos/ejercicio_permisos$ chmod 600 archiv2
a123@123:~/Documentos/ejercicio_permisos$ chmod 777 archiv3
a123@123:~/Documentos/ejercicio_permisos$ chmod 764 archiv4
a123@123:~/Documentos/ejercicio_permisos$ chmod 704 archiv5
a123@123:~/Documentos/ejercicio_permisos$ chmod 614 archiv6
a123@123:~/Documentos/ejercicio_permisos$ chmod 400 archiv7
a123@123:~/Documentos/ejercicio_permisos$ chmod 644 archiv8
a123@123:~/Documentos/ejercicio_permisos$ chmod 664 archiv9
a123@123:~/Documentos/ejercicio_permisos$ ls -l
total 0
-rwx----- 1 a123 a123 0 ene 21 14:47 archiv1
-rw----- 1 a123 a123 0 ene 21 14:47 archiv2
-rwxrwxrwx 1 a123 a123 0 ene 21 14:47 archiv3
-rwxrw-r-- 1 a123 a123 0 ene 21 14:47 archiv4
-rwx---r-- 1 a123 a123 0 ene 21 14:47 archiv5
-rw---xr-- 1 a123 a123 0 ene 21 14:47 archiv6
-r----- 1 a123 a123 0 ene 21 14:47 archiv7
-rw-r--r-- 1 a123 a123 0 ene 21 14:47 archiv8
-rw-rw-r-- 1 a123 a123 0 ene 21 14:47 archiv9

```

- Añade y/o retirar permisos sobre los archivos anteriores utilizando los operadores + y - del

'chmod' para que queden de la siguiente manera:

- archiv1 -rwx---r--
- archiv2 -r-----
- archiv3 -rw-rw-rw
- archiv 4 -rwx-w----
- archiv5 -rwx----wx
- archivo 6 -rwxrw----
- archiv7 -rw---x-w
- archiv8-----r--
- archiv 9 -rwx-----

- Crear un directorio y quitarle todos los permisos de ejecución.

```
a123@123:~/Documentos/ejercicio_permisos$ mkdir directorio_ejemplo
a123@123:~/Documentos/ejercicio_permisos$ chmod 000 directorio_ejemplo
a123@123:~/Documentos/ejercicio_permisos$ cd directorio_ejemplo/
bash: cd: directorio_ejemplo/: Permiso denegado
a123@123:~/Documentos/ejercicio_permisos$
```

- **Explicar qué pasa al intentar entrar al directorio con el comando cd.**

Me da error al no tener el permiso “x” el de ejecución

- **Explicar el significado de los permisos r,w y x para directorios.**

r : Permite listar el contenido del directorio

w : Permite crear, eliminar y renombrar archivos en el directorio, así como modificar su contenido

x : Permite entrar al directorio

Windows

Supuestos práctico(vídeo en github)

Gestión de recursos y otros

Comandos para la gestión de procesos y memoria

Ps:Muestra información sobre los procesos en ejecución

```
a123@123:~$ ps
  PID TTY          TIME CMD
 3467 pts/0    00:00:00 bash
 3596 pts/0    00:00:00 ps
```

Top: Muestra una lista actualizada periódicamente de los procesos y su uso de recursos

```
top - 10:44:46 up 4 min,  2 users,  load average: 2,60, 2,29, 1,04
Tareas: 354 total,  1 ejecutar, 283 hibernar,  0 detener,  0 zombie
%Cpu(s):  0,4 usuario,  0,7 sist,  0,0 adecuado, 98,8 inact,  0,1 en espera,  0,
KiB Mem : 8695660 total, 5363440 libre, 2173672 usado, 1158548 búfer/caché
KiB Intercambio: 1925164 total, 1925164 libre,  0 usado. 6239056 dispon
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
3646	a123	20	0	800380	34800	28724	S	4,7	0,4	0:00.14	deja-dup-m+
2656	a123	20	0	52836	7380	3932	S	1,0	0,1	0:03.61	dbus-daemon
2842	a123	20	0	5051188	458360	106720	S	1,0	5,3	0:53.97	gnome-shell
777	message+	20	0	52844	6824	4076	S	0,7	0,1	0:11.56	dbus-daemon
2074	artu	20	0	249396	2696	2364	S	0,7	0,0	0:02.99	VBoxClient
2110	artu	20	0	220784	6856	6136	S	0,7	0,1	0:00.24	at-spi2-re+
2639	a123	20	0	1296040	70524	44064	S	0,7	0,8	0:12.32	Xorg
2761	a123	20	0	314932	2644	2312	S	0,7	0,0	0:01.86	VBoxClient
10	root	20	0	0	0	0	I	0,3	0,0	0:08.22	rcu_sched
13	root	20	0	0	0	0	I	0,3	0,0	0:00.97	kworker/0:+
259	root	0	-20	0	0	0	I	0,3	0,0	0:00.63	kworker/0:+
1932	artu	20	0	1298324	81744	47512	S	0,3	0,9	0:08.83	Xorg
2066	artu	20	0	248880	2704	2376	S	0,3	0,0	0:01.69	VBoxClient
2177	artu	20	0	5049436	447380	107732	S	0,3	5,1	0:45.37	gnome-shell
2629	a123	20	0	284420	8348	7400	S	0,3	0,1	0:00.41	gnome-keyr+
2659	a123	20	0	768744	14976	12388	S	0,3	0,2	0:01.48	gnome-sess+
2755	a123	20	0	248880	2748	2420	S	0,3	0,0	0:00.95	VBoxClient

Killall: Elimina todos los procesos

```
xwayland
a123@123:~$ killall thunderbird
a123@123:~$
```

free: Muestra la memoria libre y utilizada del sistema

```
a123@123:~$ free
              total        usado       libre   compartido búfer/caché  disponible
Memoria:      8695660      2162536      4948736          34980      1584388      6219964
Swap:         1925164           0       1925164
```

Licencia 1

OEM (Original Equipment Manufacturer)

Esta licencia se utiliza para instalar un Software en algún hardware nuevo. Principalmente están unidas al hardware específico que se instala.

Un ejemplo de uso sería comprar el OEM de photoshop para reinstalar el software en varios ordenadores antes de ser vendidos.

Retail

Esta licencia está dirigida a una venta al público, están diseñadas para ser instaladas en un solo dispositivo para un usuario que compre el servicio.

Comprar las licencias por G2A(por ejemplo) de windows y la instalo en mi dispositivo

Licencias por volumen



Estas licencias son más para empresas que necesitan instalar el software en muchos dispositivos, se puede instalar en muchos dispositivos teniendo una comprada.

Tengo una empresa que necesita un software en común para todos los trabajadores y compró las licencias por volúmenes.

Licencia 2



- CC0 1.0

Significa que renuncia a todos sus derechos de autor sobre la obra, permitiendo que la obra pase al dominio público.

recuperacion by arturo is marked with CC0 1.0  




-CC BY 4.0

La BY significa que los que utilicen la obra tienen que dar créditos al autor y el 4.0 internacional limita su alcance.

recuperacion by arturo is licensed under CC BY 4.0  




- CC BY-SA 4.0

Aquí añade el SA indican que las licencias tienen herencias, es decir, las restricciones de la de arriba pasan a las de abajo.

recuperacion by arturo is licensed under CC BY-SA 4.0   

- CC BY-ND 4.0

Añadimos ND que significa que no se puede modificar la obra

recuperacion de arturo tiene licencia CC BY-ND 4.0   

No comercial (NC)

Esta licencia permite la modificación de la obra pero solo con fines no comerciales, además tienen de dar créditos al autor original.

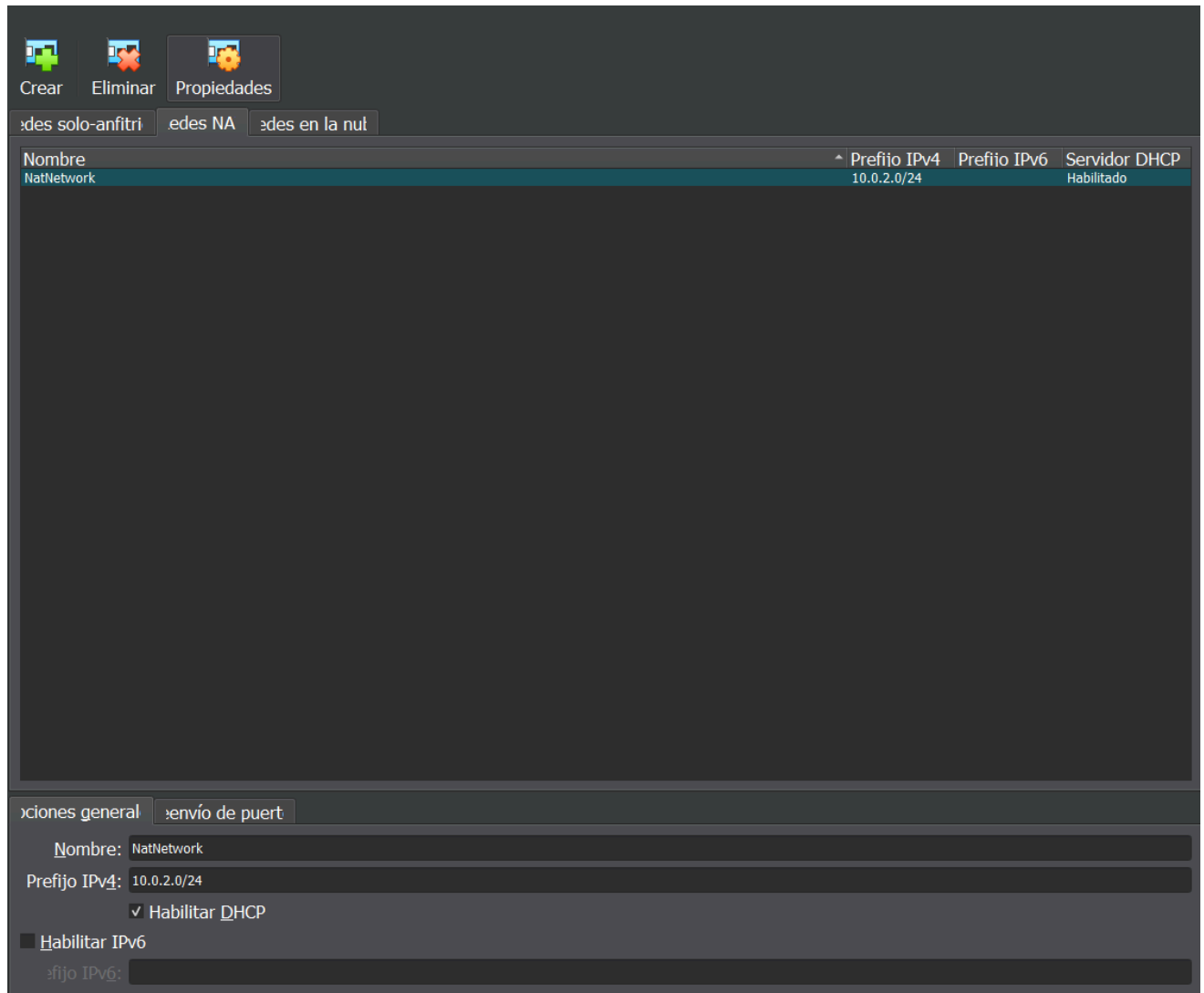
No copy © 2024 by arturo is licensed under CC BY-NC 4.0   

Luego quedan dos pero que se mezclan CC BY-NC-SA 4.0 CC BY-NC-ND 4.0

Virtualización

Instalación dual(video en github)

Red NAT(video en github)



```
dual5 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal - arturo@arturo-Vir...
Terminal - arturo@arturo-VirtualBox: ~
Archivo Editar Ver Terminal Pestañas Ayuda
arturo@arturo-VirtualBox:~$ ifconfig
enp0s3  Link encap:Ethernet  direcciónHW 08:00:27:67:23:37
        Direc. inet:10.0.2.4  Difus.:10.0.2.255  Másc:255.255.255.0
        Dirección inet6: fe80::7bc6:58d0:dd42:9eb2/64  Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
        Paquetes RX:8 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:46 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1000
        Bytes RX:1308 (1.3 KB)  TX bytes:5940 (5.9 KB)

lo      Link encap:Bucle local
        Direc. inet:127.0.0.1  Másc:255.0.0.0
        Dirección inet6: ::1/128  Alcance:Anfitrión
        ACTIVO BUCLE FUNCIONANDO  MTU:65536  Métrica:1
        Paquetes RX:196 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:196 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1000
        Bytes RX:14506 (14.5 KB)  TX bytes:14506 (14.5 KB)

arturo@arturo-VirtualBox:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.709 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.933 ms
```

```
preuba [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal - a123@preuba: ~
Terminal - a123@preuba: ~
File Edit View Terminal Tabs Help
a123@preuba:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:75:7f:21
        inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::609:4fe8:8c62:6cce/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:22 errors:0 dropped:0 overruns:0 frame:0
        TX packets:106 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:3690 (3.6 KB)  TX bytes:12852 (12.8 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128  Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:382 errors:0 dropped:0 overruns:0 frame:0
        TX packets:382 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:25964 (25.9 KB)  TX bytes:25964 (25.9 KB)

a123@preuba:~$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=0.828 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.959 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=1.05 ms
```