

11-17 linguaxes scripts

Emilio Cabo Gomis

Índice

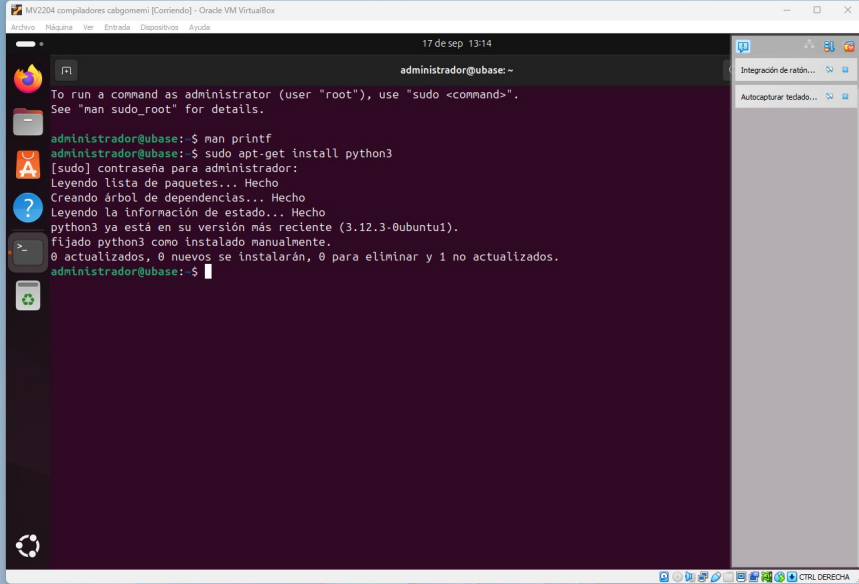
Python3.....	3
Linux.....	3
Windows 11.....	5
Control de flujo en python.....	9
Creamos un programa que calcule el área del triángulo.....	10
PERL.....	11
Linux.....	11
Windows.....	12
Ejecución de programas en Windows 11.....	14
Ejecución de programas en Linux.....	16
Hola mundo.....	16
Variables.....	16
Ver nombre PC.....	17
C.....	18
Datatypes en c.....	20
Definición de variables y trabajo con loops.....	21
Definición de array y trabajo con loops.....	22
STDIN a STDOUT y array.....	22
BASH.....	24
Hola mundo.....	24
Variables en BASH.....	24
Bucle for.....	25
Bucle While.....	26
Input y output.....	26
Powershell.....	27
Habilitar Scripts Powershell.....	27
Cambiar nombre de la máquina.....	28
Cambiar la configuración de red del sistemas.....	29
Crear punto de restauración del sistema.....	29
Desactivar IPv6.....	30
Bibliografía.....	31

Python3

Linux

Iniciamos la máquina virtual Ubuntu 22.04 que tenemos lista para comenzar a trabajar.

Realizamos la instalación de python mediante el comando **“apt-get install python3”**.

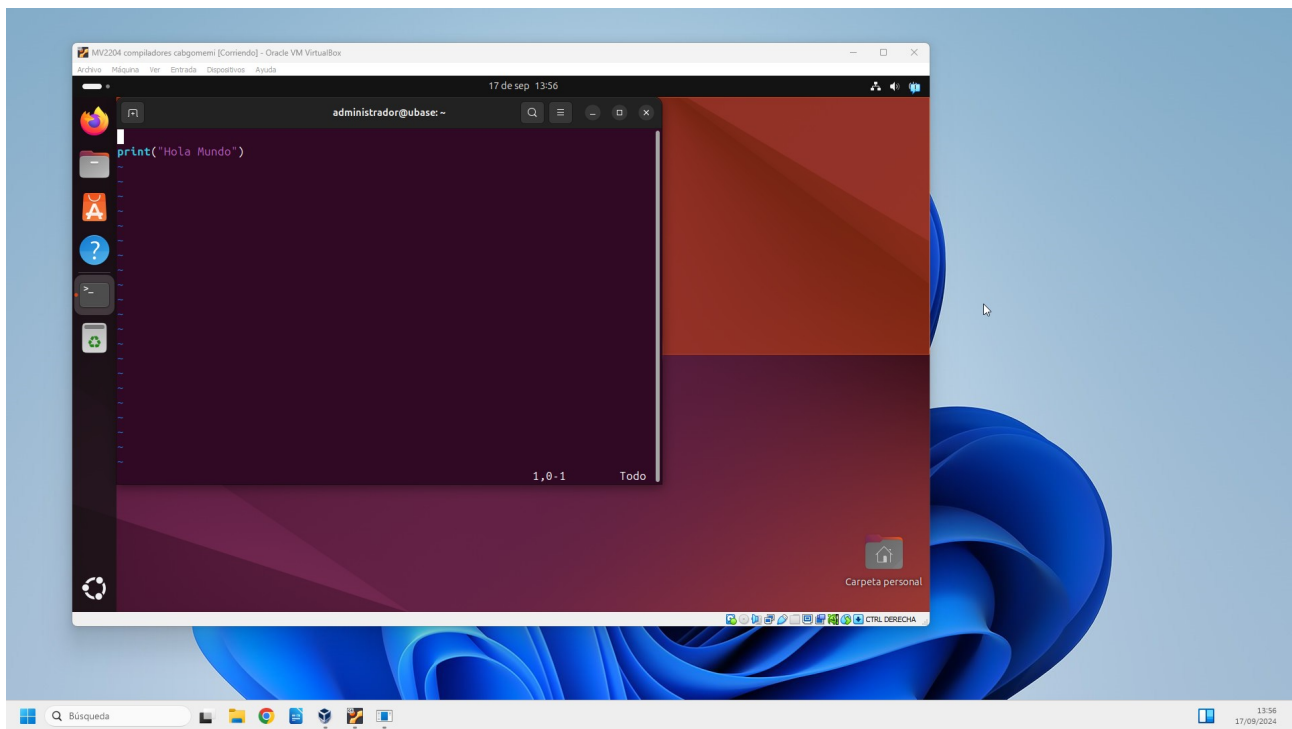


```
17 de sep. 13:14
administrador@ubase: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

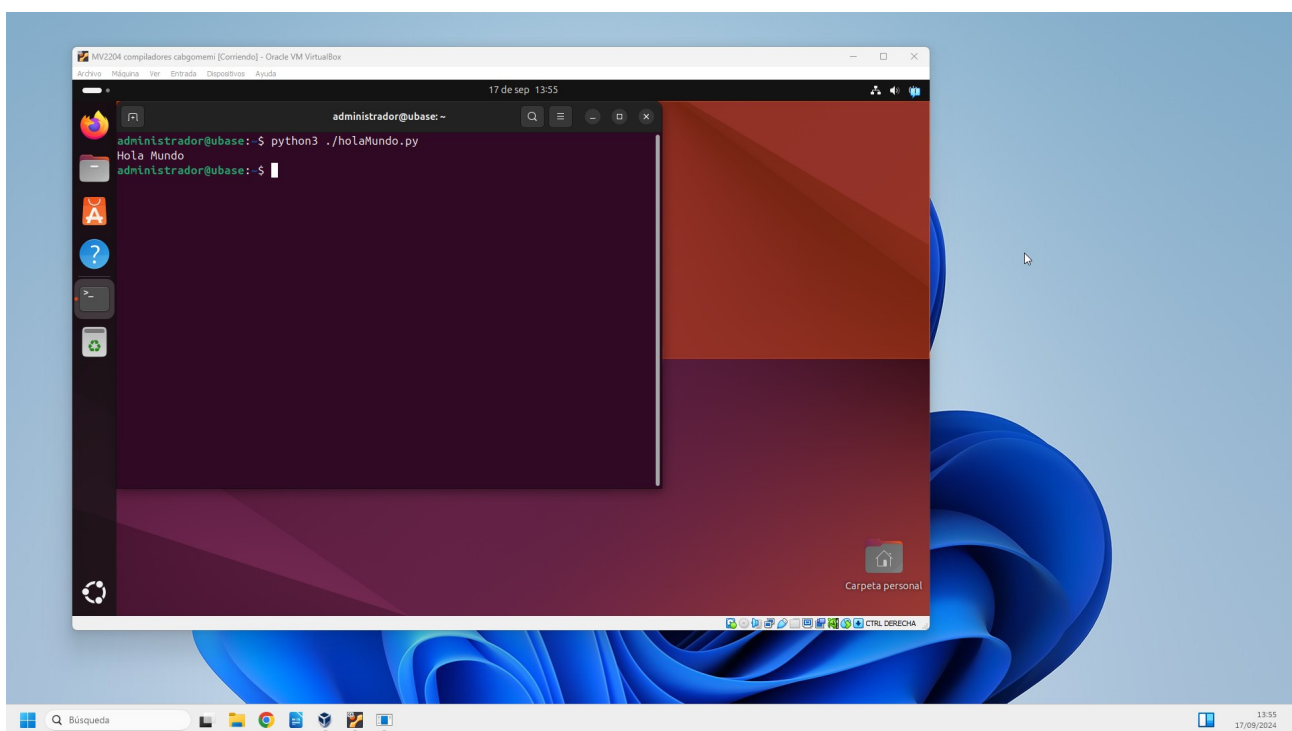
administrador@ubase:~$ man printf
administrador@ubase:~$ sudo apt-get install python3
[sudo] contraseña para administrador:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
python3 ya está en su versión más reciente (3.12.3-0ubuntu1).
fijado python3 como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
administrador@ubase:~$
```

En mi caso ya lo tengo instalado.

Creamos un simple script que printea a stdout. (Importante hacer un script mediante un editor de texto)

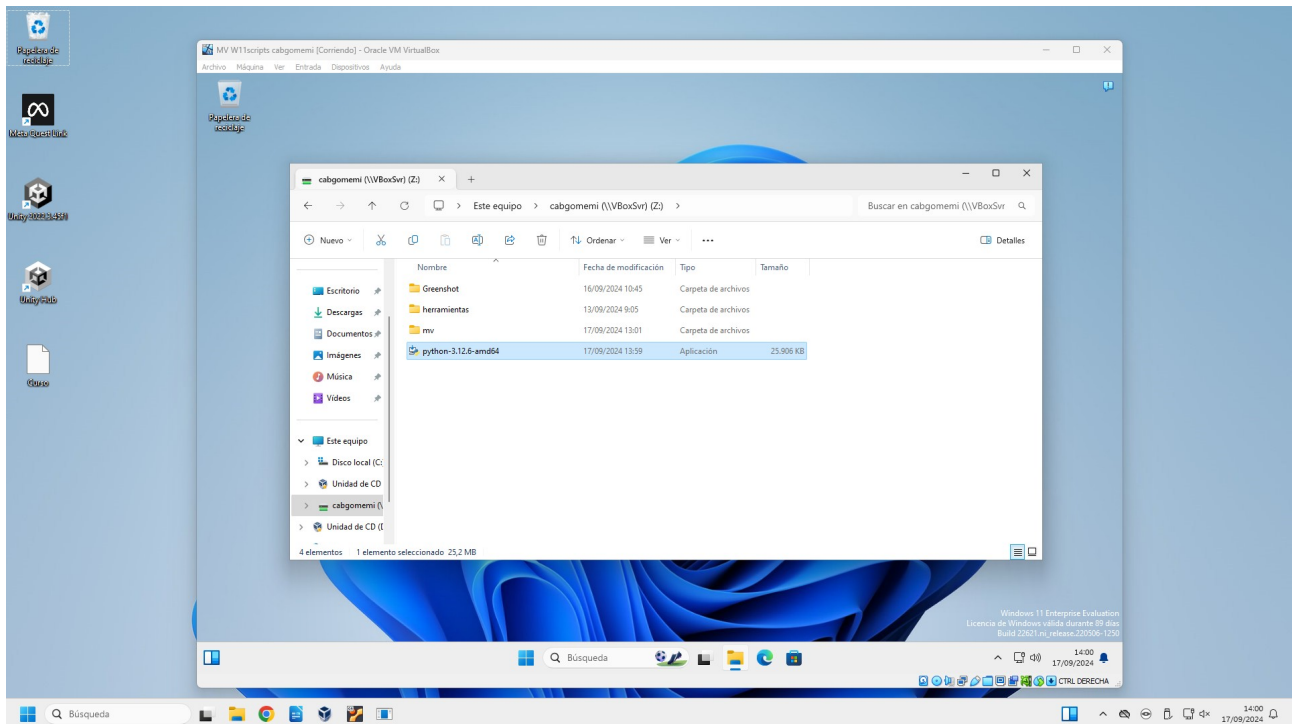


Comprobamos la ejecución del programa.

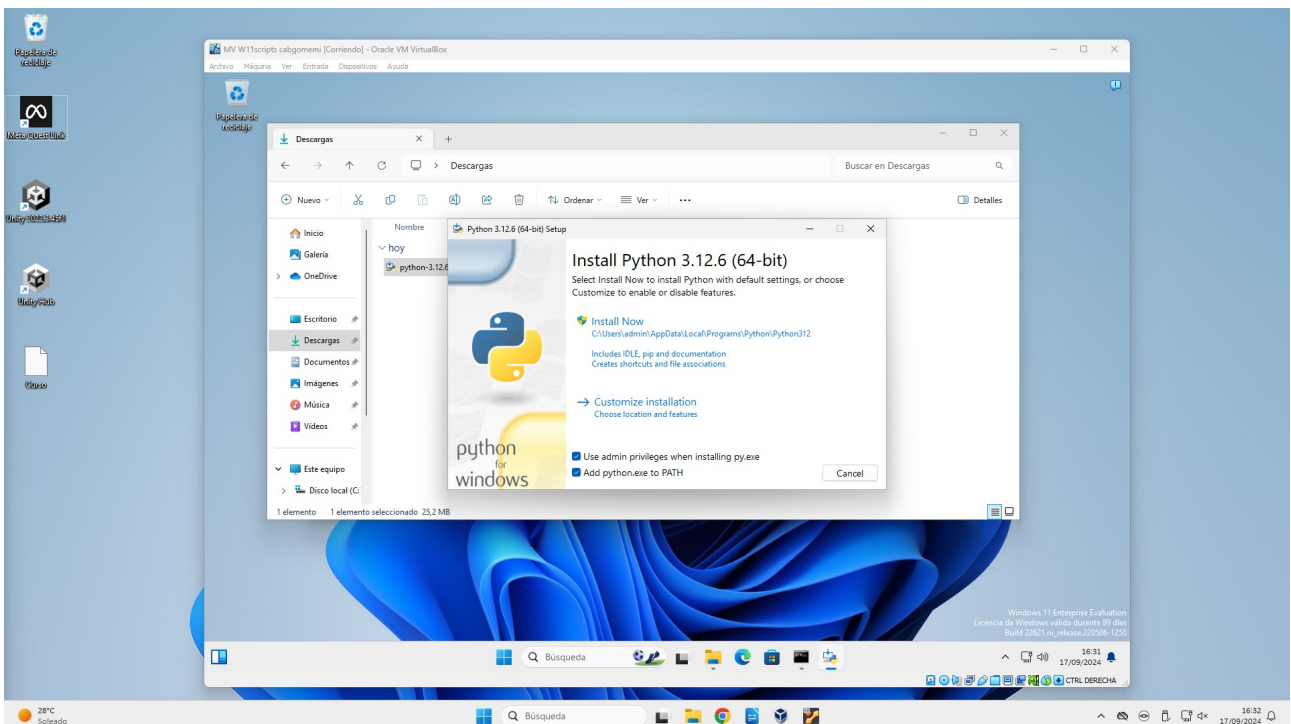


Windows 11

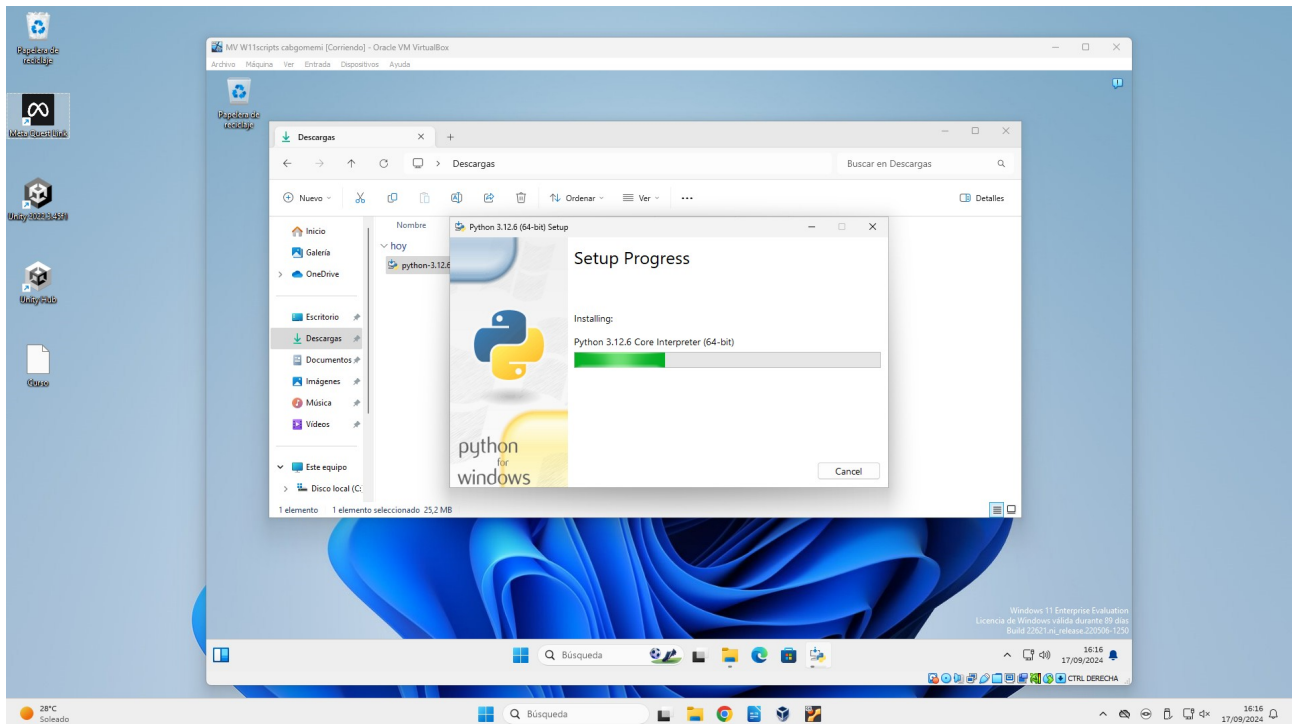
Descargamos el instalador de python3 de la web oficial.



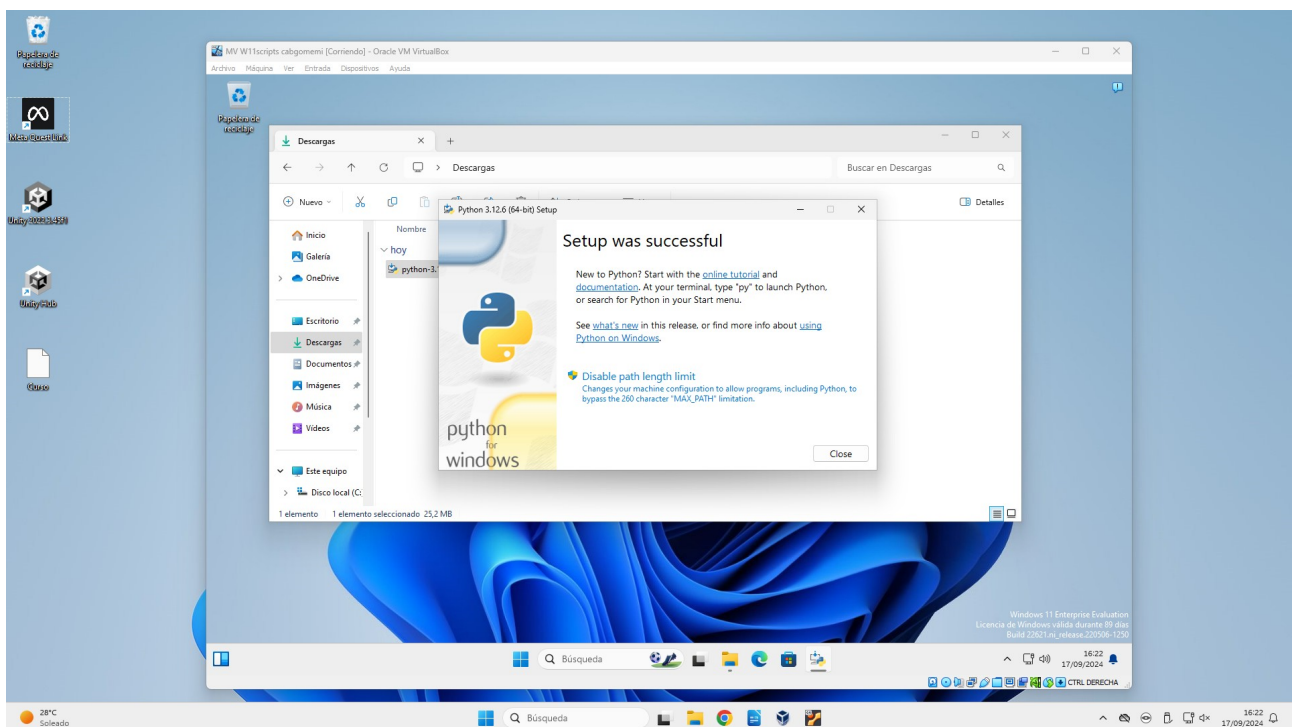
Importante marcar la opción que añade Python a el **PATH**.



Comenzamos a instalar python mediante el instalador.

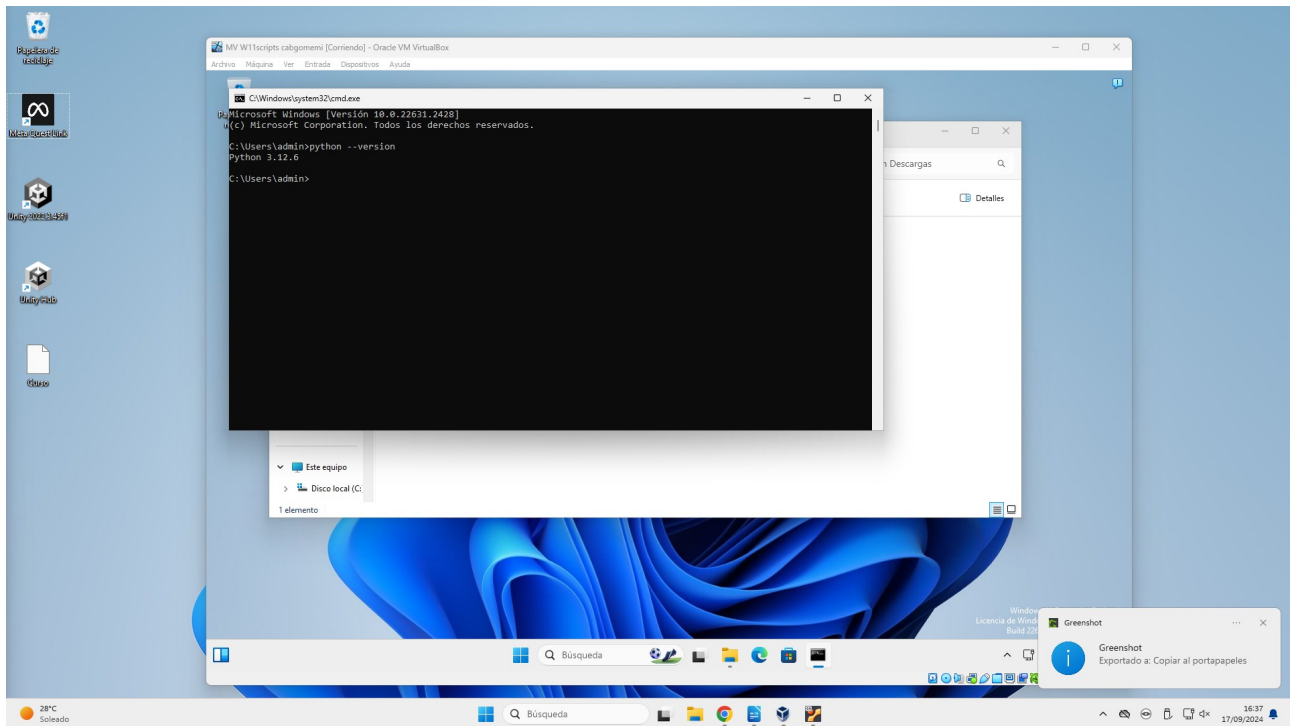


Seleccionaremos la opción “Disable path lenght limit”



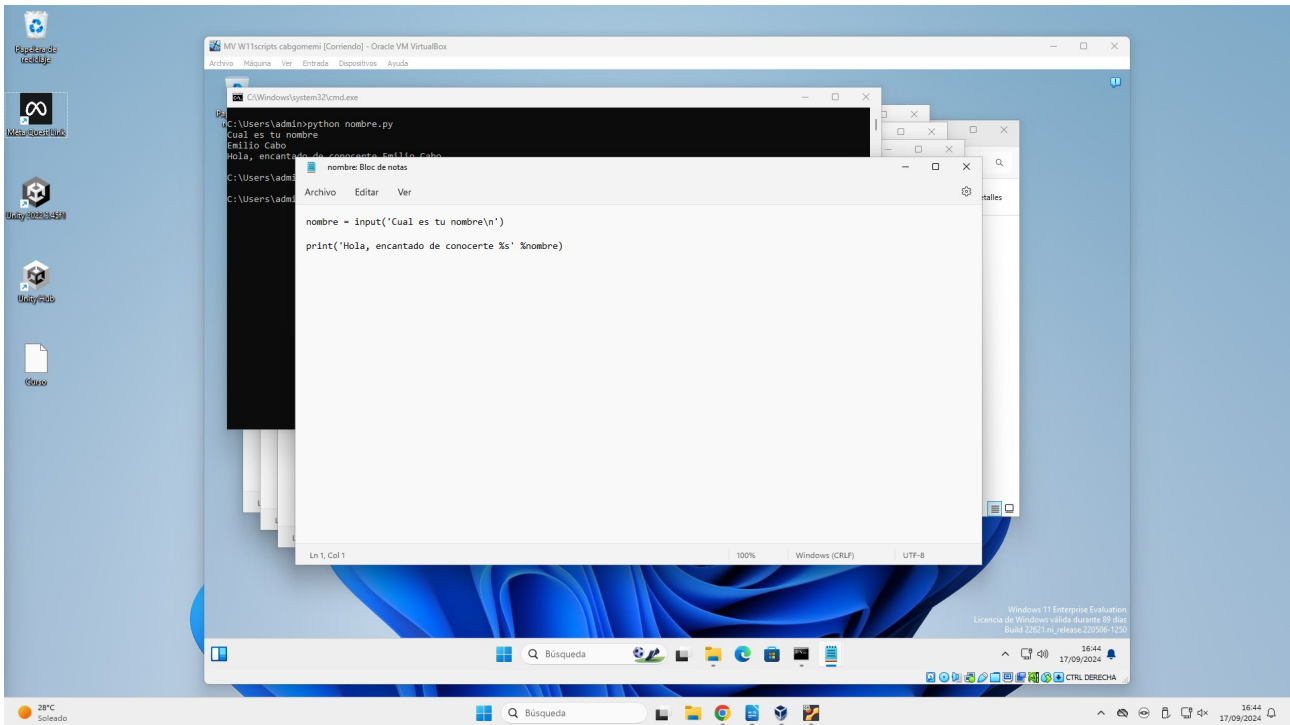
Necesitaremos permisos de administración para esto

Una vez completados todos estos pasos, la instalación estará completa.

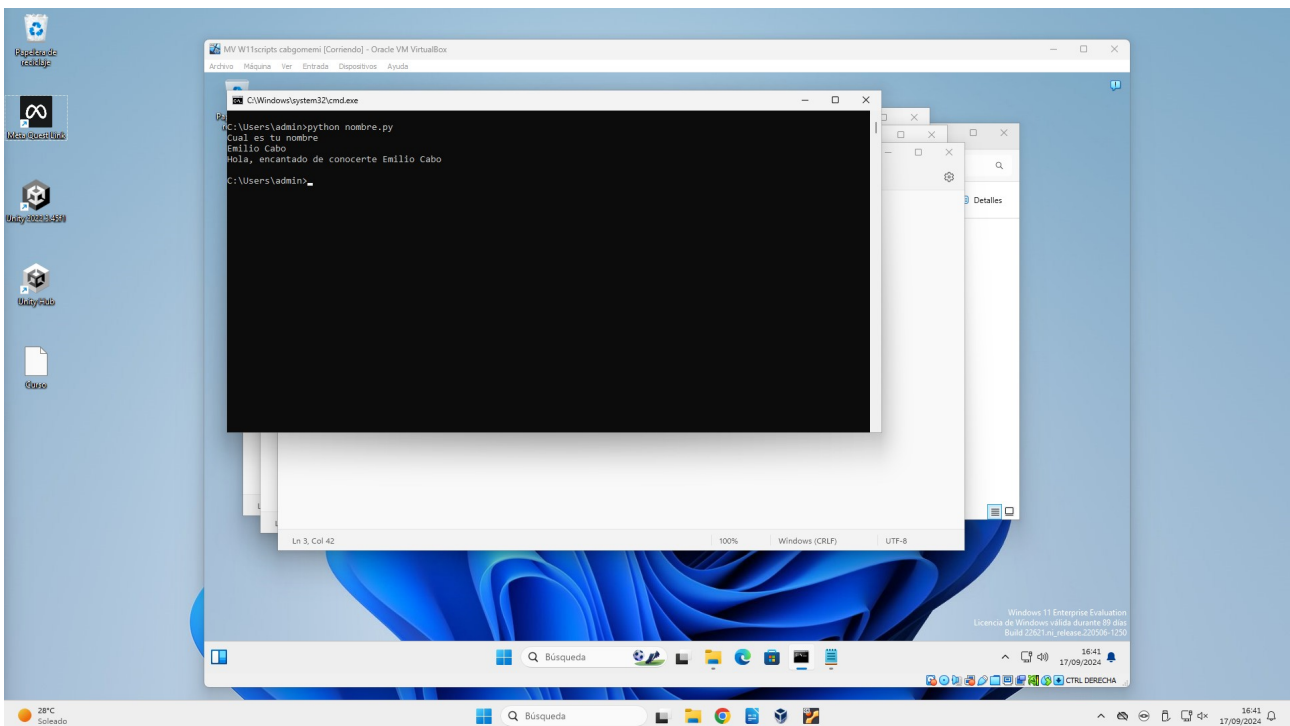


Vamos a ejecutar un scripts algo distinto que el de linux.

Debido a que es un lenguaje basado en C la sintaxis de la programación es muy similar.



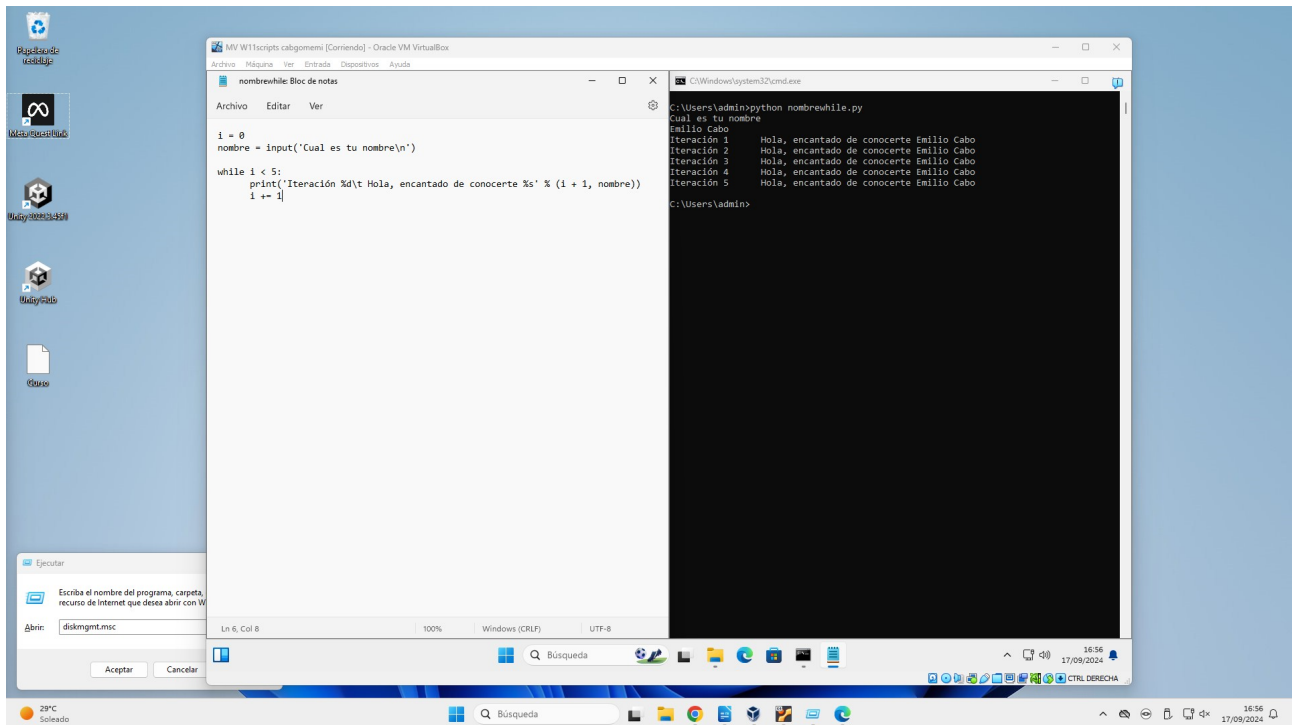
Contenido del programa



Ejecución del programa

Control de flujo en python

Vamos a crear nuestro primer loop while en python.



The screenshot shows a Windows 10 desktop with a virtual machine environment. A Notepad window titled 'nombrewhile Bloc de notas' contains the following Python code:

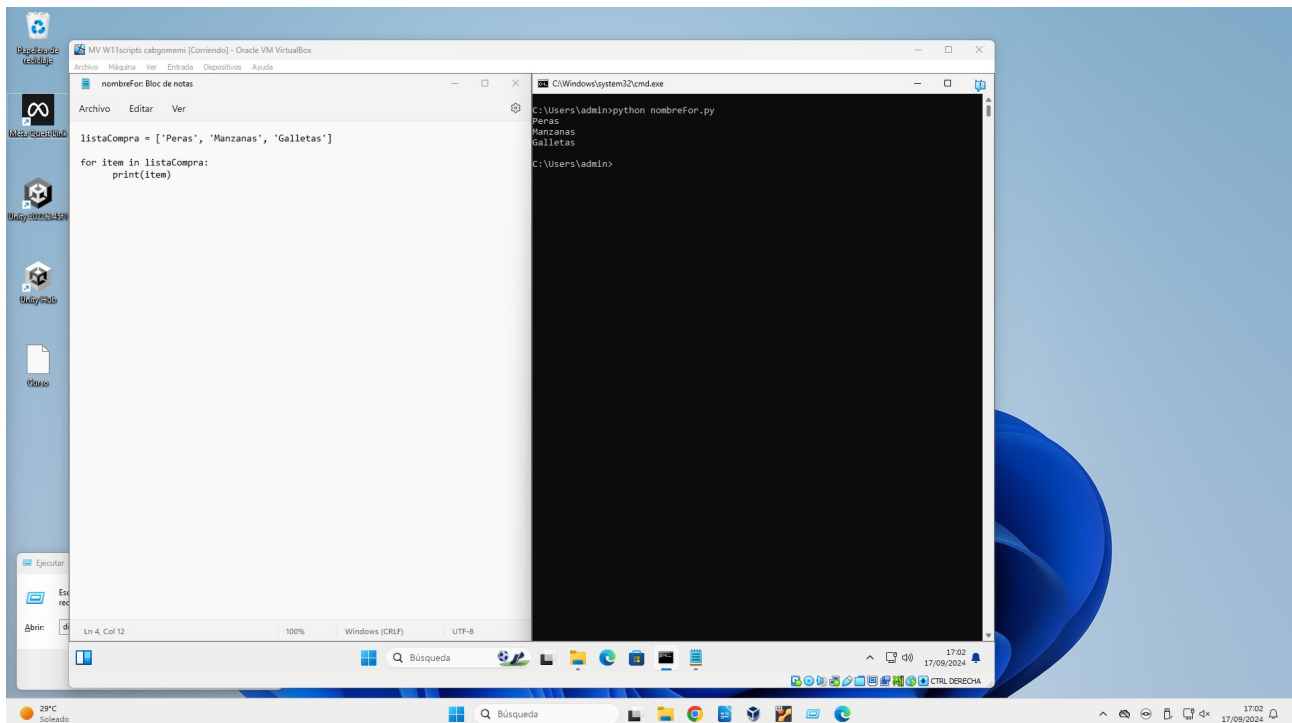
```
i = 0
nombre = input('Cual es tu nombre')
while i < 5:
    print('Iteración %d\t Hola, encantado de conocerte %s' % (i + 1, nombre))
    i += 1
```

A command prompt window titled 'C:\Windows\system32\cmd.exe' shows the execution of the program:

```
C:\Users\admin>python nombrewhile.py
Cual es tu nombre
Emilio Cabo
Iteración 1      Hola, encantado de conocerte Emilio Cabo
Iteración 2      Hola, encantado de conocerte Emilio Cabo
Iteración 3      Hola, encantado de conocerte Emilio Cabo
Iteración 4      Hola, encantado de conocerte Emilio Cabo
Iteración 5      Hola, encantado de conocerte Emilio Cabo
C:\Users\admin>
```

The taskbar at the bottom shows the system clock as 16:56 on 17/09/2024.

Vamos a crear nuestro primer array y nuestro primer bucle for en un programa de “Lista de compra”



The screenshot shows the same Windows 10 desktop environment. A Notepad window titled 'nombrefor Bloc de notas' contains the following Python code:

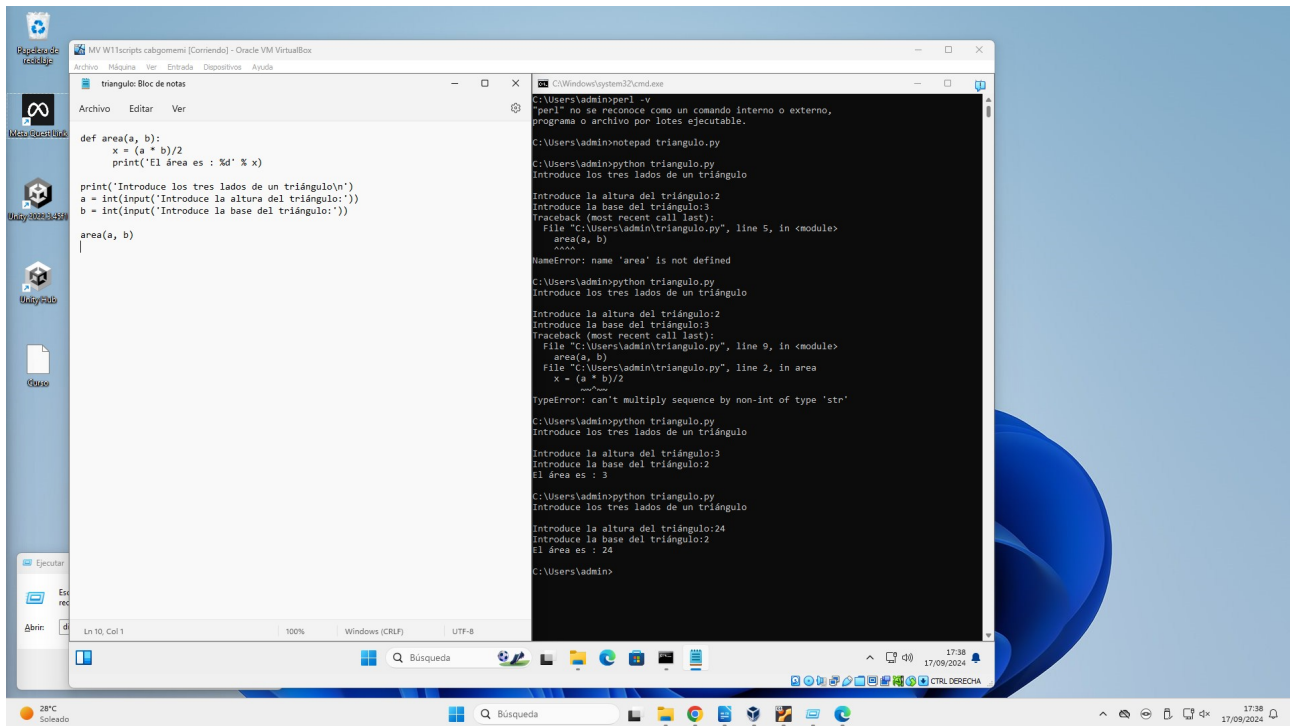
```
listaCompra = ['Peras', 'Manzanas', 'Galletas']
for item in listaCompra:
    print(item)
```

A command prompt window titled 'C:\Windows\system32\cmd.exe' shows the execution of the program:

```
C:\Users\admin>python nombrefor.py
Peras
Manzanas
Galletas
C:\Users\admin>
```

The taskbar at the bottom shows the system clock as 17:02 on 17/09/2024.

Creamos un programa que calcule el área del triángulo.



The screenshot shows a Windows 10 desktop with a blue background. In the foreground, there are two windows. The left window is a Notepad application titled "triángulo: Bloc de notas". It contains the following Python code:

```
def area(a, b):  
    x = (a * b)/2  
    print("El área es : %d" % x)  
  
print("Introduce los tres lados de un triángulo\n")  
a = int(input("Introduce la altura del triángulo:"))  
b = int(input("Introduce la base del triángulo:"))  
  
area(a, b)
```

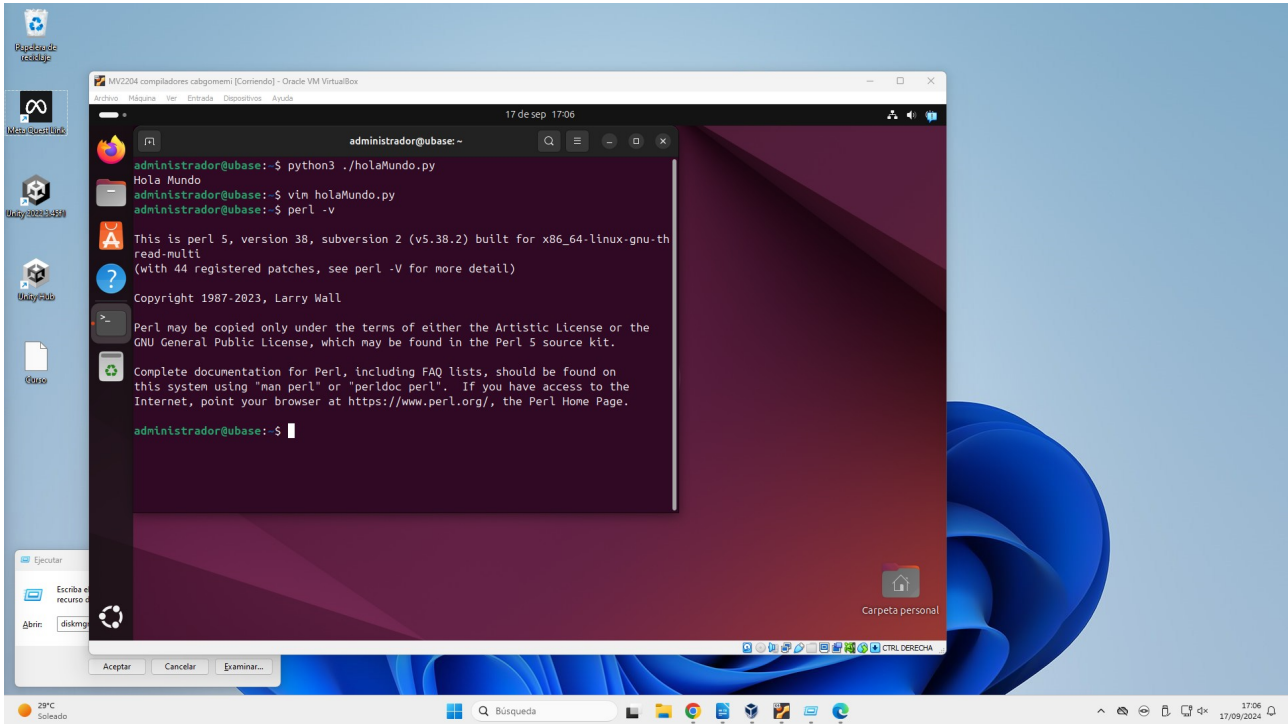
The right window is a Command Prompt titled "C:\Windows\system32\cmd.exe". It shows the execution of the script with the following output:

```
C:\Users\admin>perl -v  
"perl" no se reconoce como un comando interno o externo,  
programa o archivo por lotes ejecutable.  
  
C:\Users\admin>notepad triángulo.py  
  
C:\Users\admin>python triángulo.py  
Introduce los tres lados de un triángulo  
Introduce la altura del triángulo:2  
Introduce la base del triángulo:3  
Traceback (most recent call last):  
  File "C:\Users\admin\triángulo.py", line 5, in <module>  
    area(a, b)  
NameError: name 'area' is not defined  
  
C:\Users\admin>python triángulo.py  
Introduce los tres lados de un triángulo  
Introduce la altura del triángulo:2  
Introduce la base del triángulo:3  
Traceback (most recent call last):  
  File "C:\Users\admin\triángulo.py", line 9, in <module>  
    area(a, b)  
  File "C:\Users\admin\triángulo.py", line 2, in area  
    x = (a * b)/2  
TypeError: can't multiply sequence by non-int of type 'str'  
  
C:\Users\admin>python triángulo.py  
Introduce los tres lados de un triángulo  
Introduce la altura del triángulo:3  
Introduce la base del triángulo:2  
El área es : 3  
  
C:\Users\admin>python triángulo.py  
Introduce los tres lados de un triángulo  
Introduce la altura del triángulo:24  
Introduce la base del triángulo:2  
El área es : 24  
  
C:\Users\admin>
```

PERL

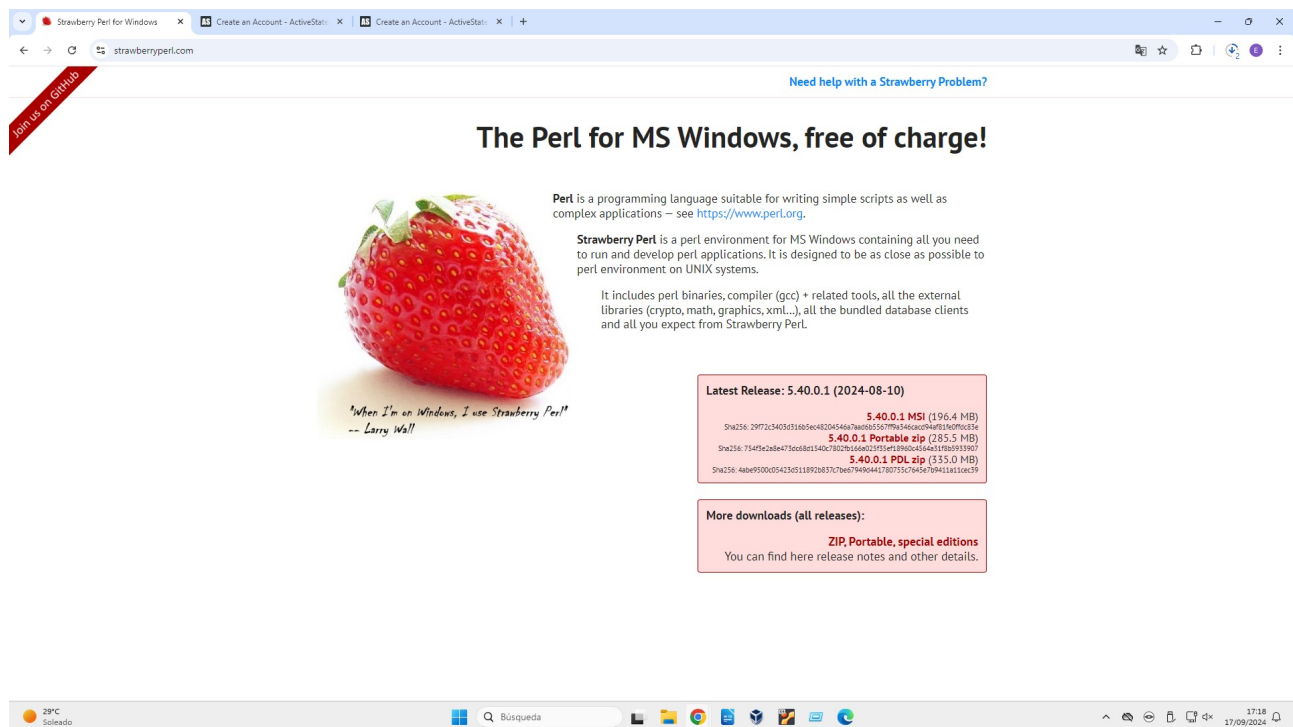
Linux

Comprobamos que tenemos instalado perl mediante el comando “**perl -v**”



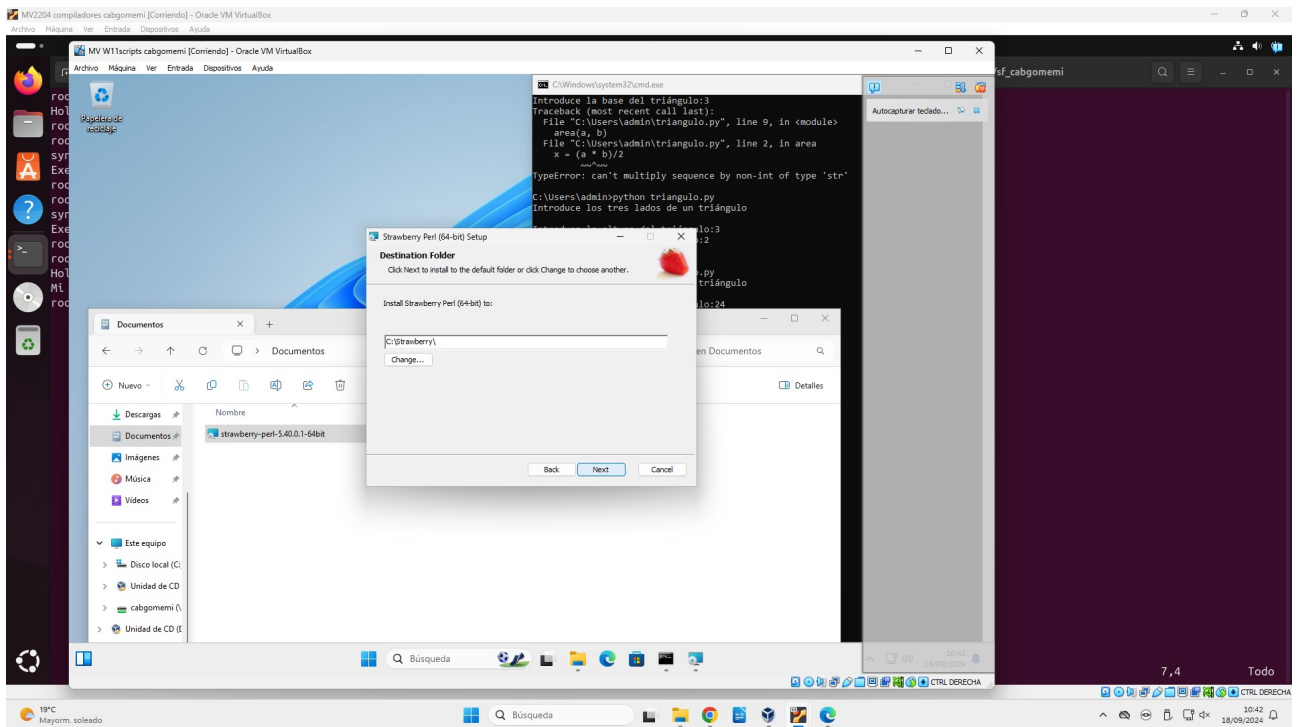
Windows

Descargamos perl desde la [web oficial](#)

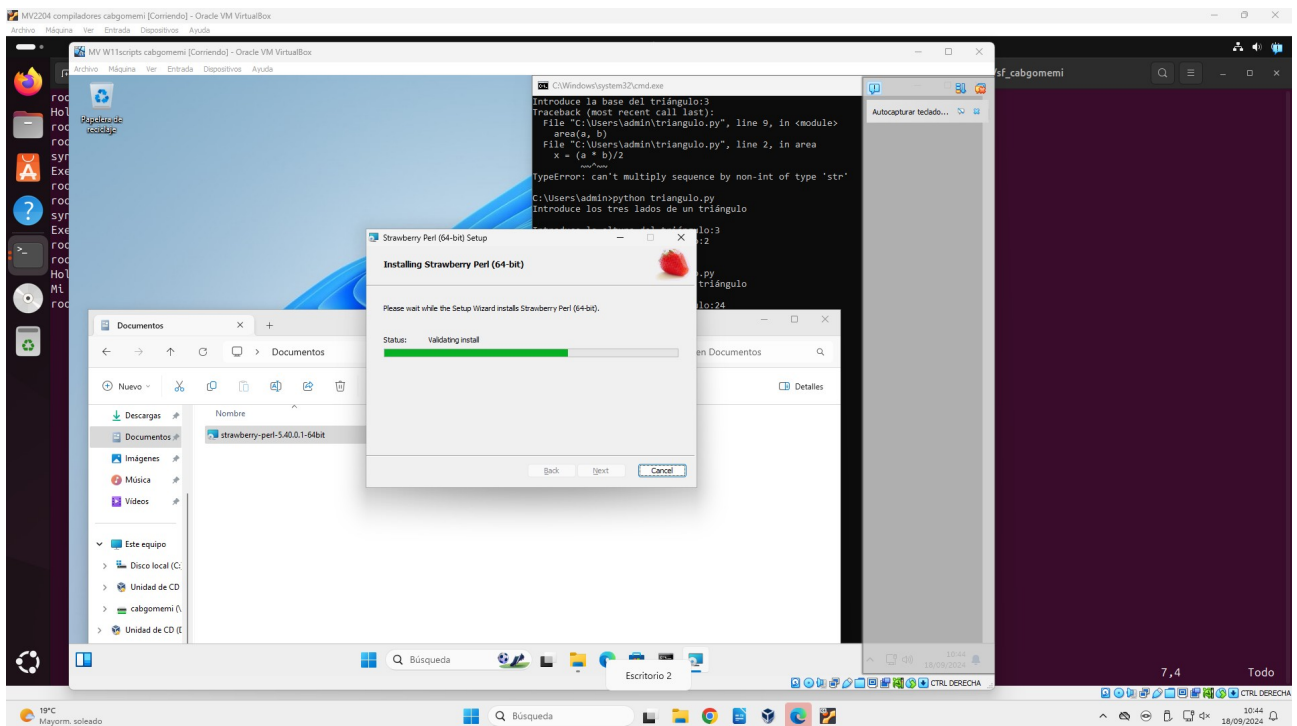


Ejecutamos el instalador.

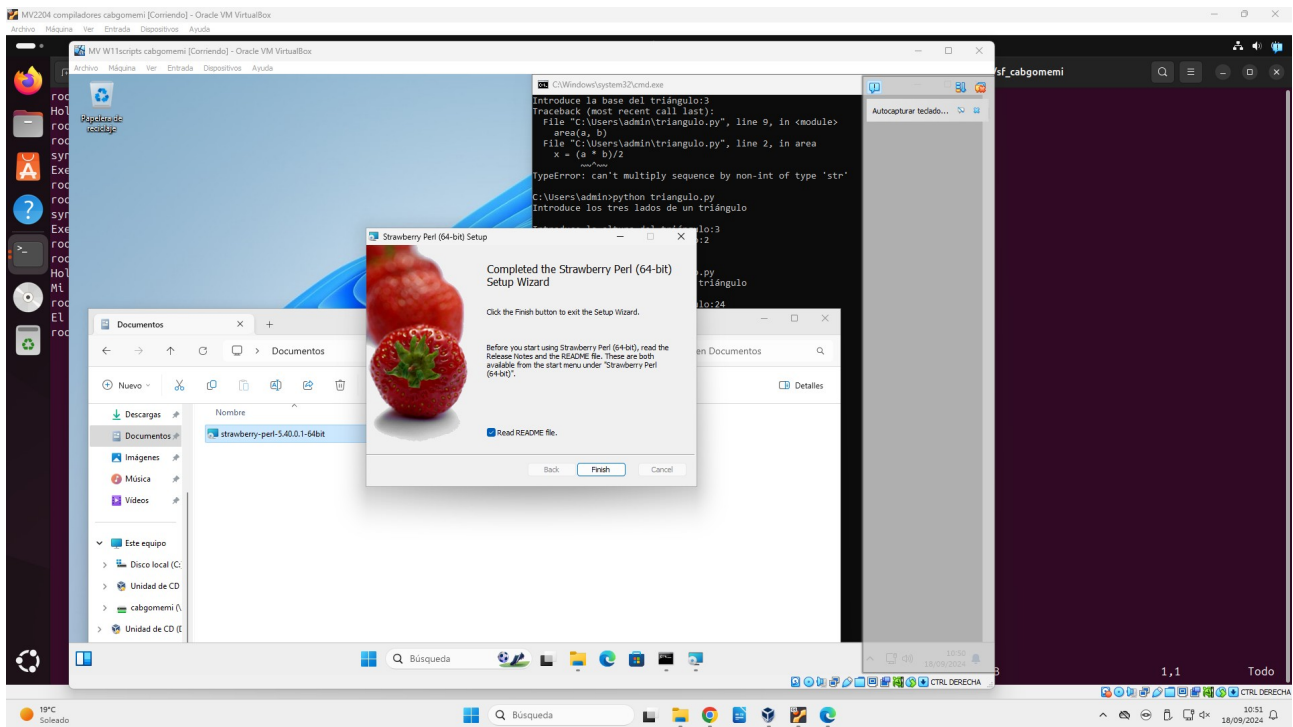
Seleccionamos una ubicación para instalar el intérprete.



Damos permisos de administrador y la instalación continuará sin problemas.

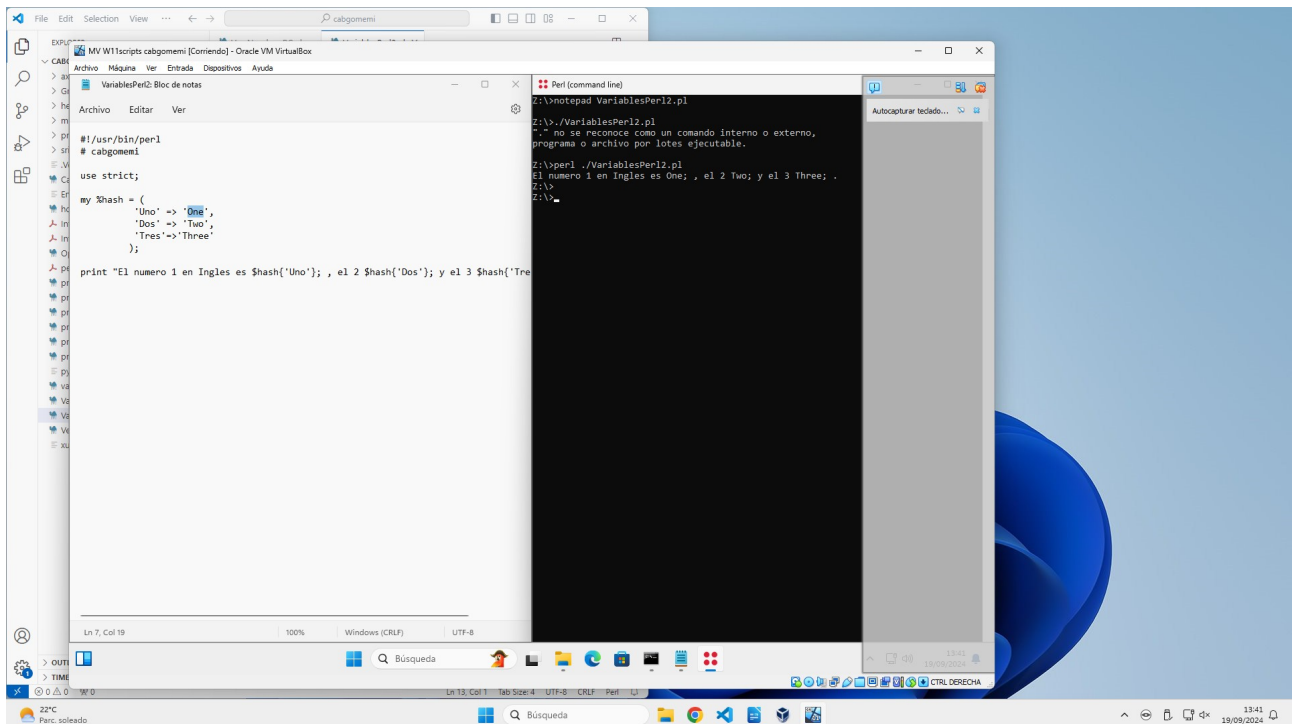


Instalación de PERL completada en Windows.



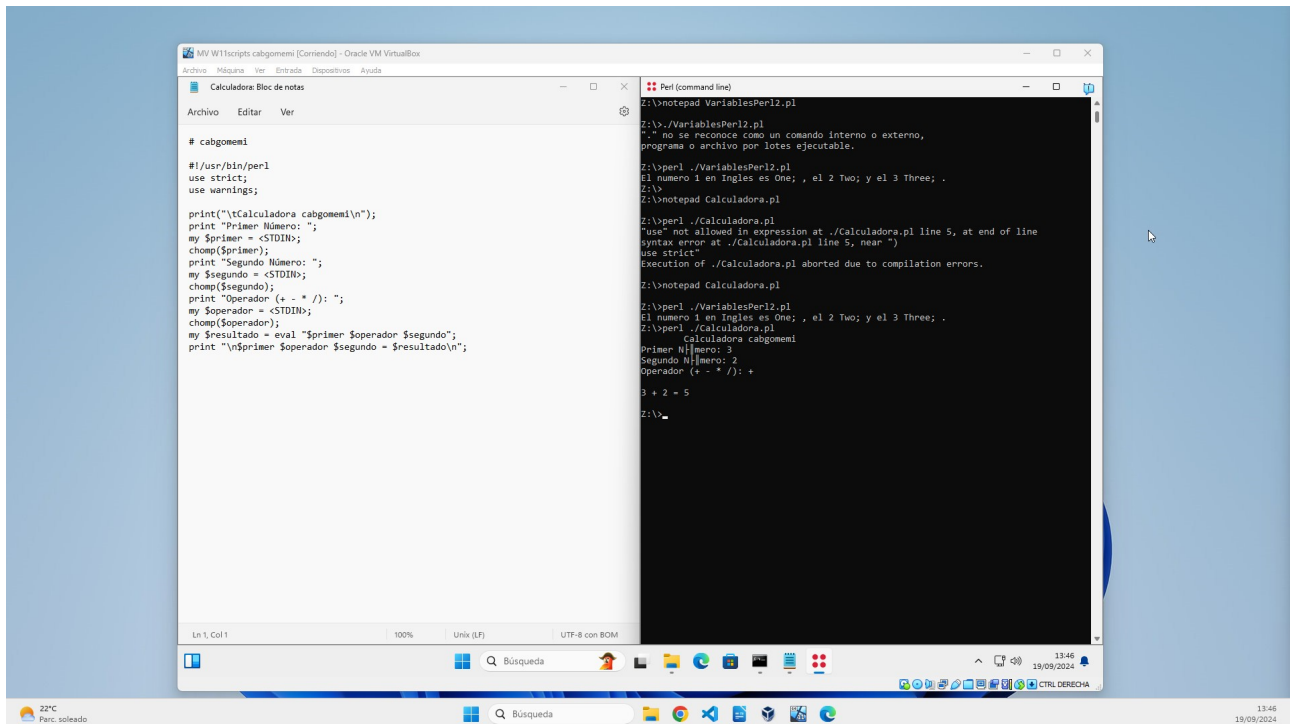
Ejecución de programas en Windows 11

Este es un script que muestra un uso de variables más avanzado.



Calculadora en PERL

Este programa es una calculadora que calcula el resultado de operaciones con dos números que nosotros le introduzcamos. Nos permite además elegir el operador.



The screenshot shows a Windows Virtual Machine environment. On the left, a Notepad window titled 'Calculadora: Bloc de notas' contains a Perl script named 'cabgomem1'. The script prompts the user for two numbers and an operator, then calculates the result. On the right, a 'Per (command line)' window shows the execution of the script. It displays the command to run the script, followed by the user's input: 'El numero 1 en Ingles es One; , el 2 Two; y el 3 Three; .', and the resulting output: 'Primer N: 3', 'Segundo N: 2', 'Operador (+ - * /): +', and '3 + 2 = 5'.

```
# cabgomem1

#!/usr/bin/perl
use strict;
use warnings;

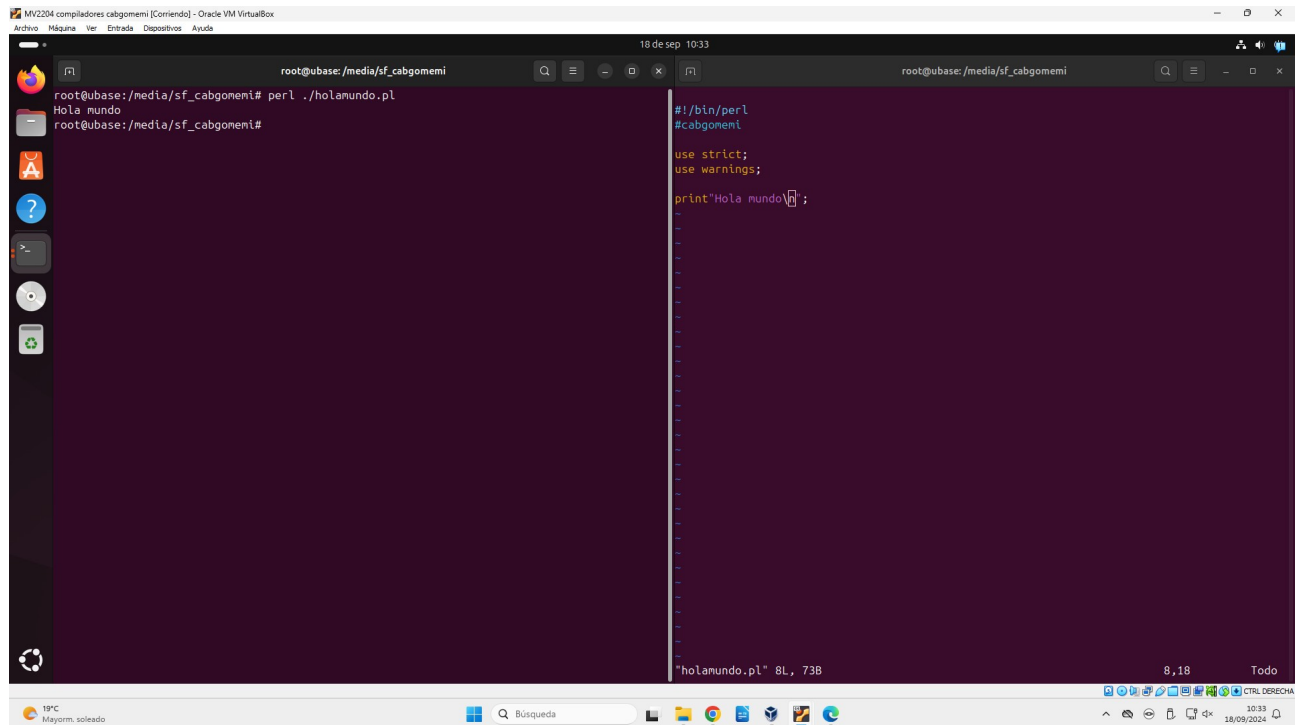
print("\tCalculadora cabgomem1\n");
print "Primer Número: ";
my $primer = <STDIN>;
chomp($primer);
print "Segundo Número: ";
my $segundo = <STDIN>;
chomp($segundo);
print "Operador (+ - * /): ";
my $operador = <STDIN>;
chomp($operador);
my $resultado = eval "$primer $operador $segundo";
print "\n$primer $operador $segundo = $resultado\n";
```

```
Z:\>notepad VariablesPerl2.pl
Z:\>./VariablesPerl2.pl
"." no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
Z:\>perl ./VariablesPerl2.pl
El numero 1 en Ingles es One; , el 2 Two; y el 3 Three; .
Z:\>
Z:\>notepad Calculadora.pl
Z:\>perl ./Calculadora.pl
"use" not allowed in expression at ./Calculadora.pl line 5, at end of line
syntax error at ./Calculadora.pl line 5, near ")"
use strict"
Execution of ./Calculadora.pl aborted due to compilation errors.
Z:\>notepad Calculadora.pl
Z:\>perl ./VariablesPerl2.pl
El numero 1 en Ingles es One; , el 2 Two; y el 3 Three; .
Z:\>perl ./Calculadora.pl
Calculadora cabgomem1
Primer N: 3
Segundo N: 2
Operador (+ - * /): +
3 + 2 = 5
Z:\>
```


Ejecución de programas en Linux

Hola mundo

Ejecutamos el primer nuestro “Hola Mundo” en PERL



The screenshot shows a terminal window with a dark purple background. The prompt is `root@ubase: /media/sf_cabgomemi#`. The user enters `perl ./holamundo.pl`, and the output is `Hola mundo`. The terminal also shows the contents of the `holamundo.pl` file, which is a Perl script that prints "Hola mundo".

```
root@ubase: /media/sf_cabgomemi# perl ./holamundo.pl
Hola mundo
root@ubase: /media/sf_cabgomemi#
```

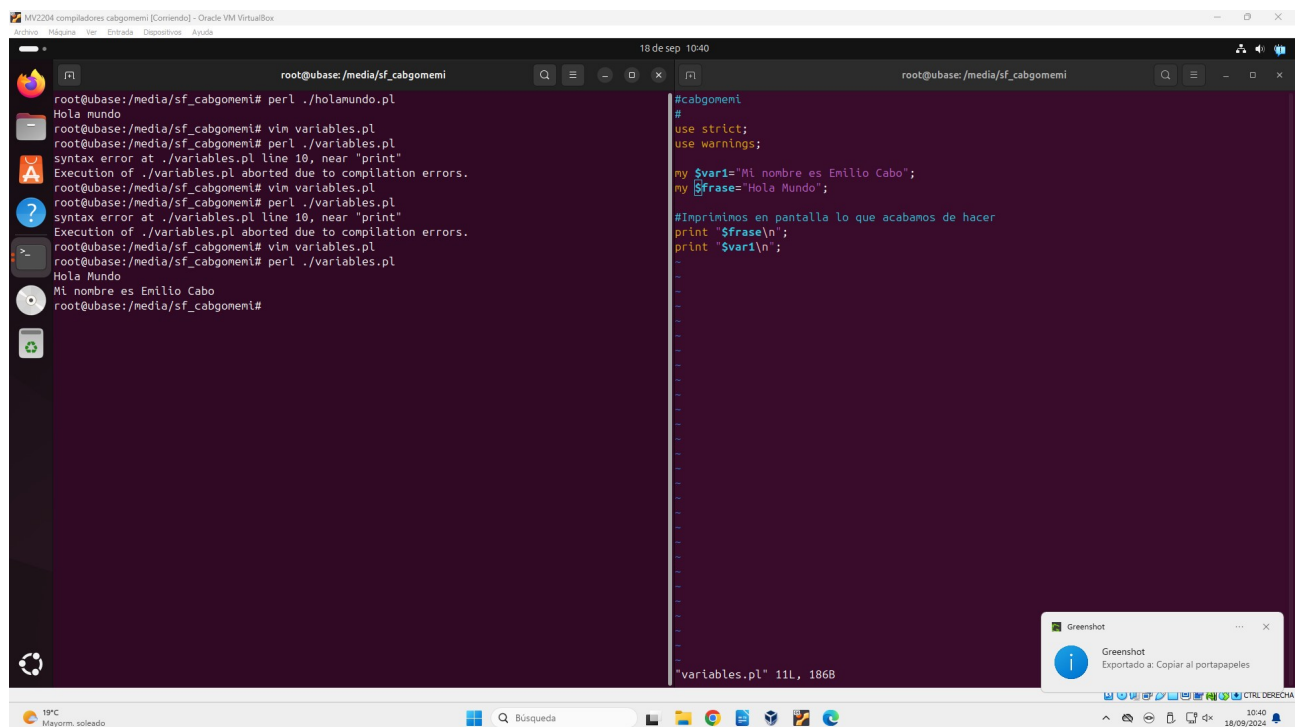
```
#!/bin/perl
#cabgomemi

use strict;
use warnings;

print"Hola mundo\n";
```

Variables

Este programa Imprimiremos en pantalla desde una variable.



The screenshot shows a terminal window with a dark purple background. The prompt is `root@ubase: /media/sf_cabgomemi#`. The user enters `perl ./holamundo.pl`, and the output is `Hola mundo`. The user then enters `vim variables.pl`, and the terminal shows the contents of the `variables.pl` file, which is a Perl script that prints "Hola Mundo" and "Mi nombre es Emilio Cabo". The user then enters `perl ./variables.pl`, and the output is `Hola Mundo` and `Mi nombre es Emilio Cabo`.

```
root@ubase: /media/sf_cabgomemi# perl ./holamundo.pl
Hola mundo
root@ubase: /media/sf_cabgomemi# vim variables.pl
root@ubase: /media/sf_cabgomemi# perl ./variables.pl
syntax error at ./variables.pl line 10, near "print"
Execution of ./variables.pl aborted due to compilation errors.
root@ubase: /media/sf_cabgomemi# vim variables.pl
root@ubase: /media/sf_cabgomemi# perl ./variables.pl
syntax error at ./variables.pl line 10, near "print"
Execution of ./variables.pl aborted due to compilation errors.
root@ubase: /media/sf_cabgomemi# vim variables.pl
root@ubase: /media/sf_cabgomemi# perl ./variables.pl
Hola Mundo
Mi nombre es Emilio Cabo
root@ubase: /media/sf_cabgomemi#
```

```
#cabgomemi
#
use strict;
use warnings;

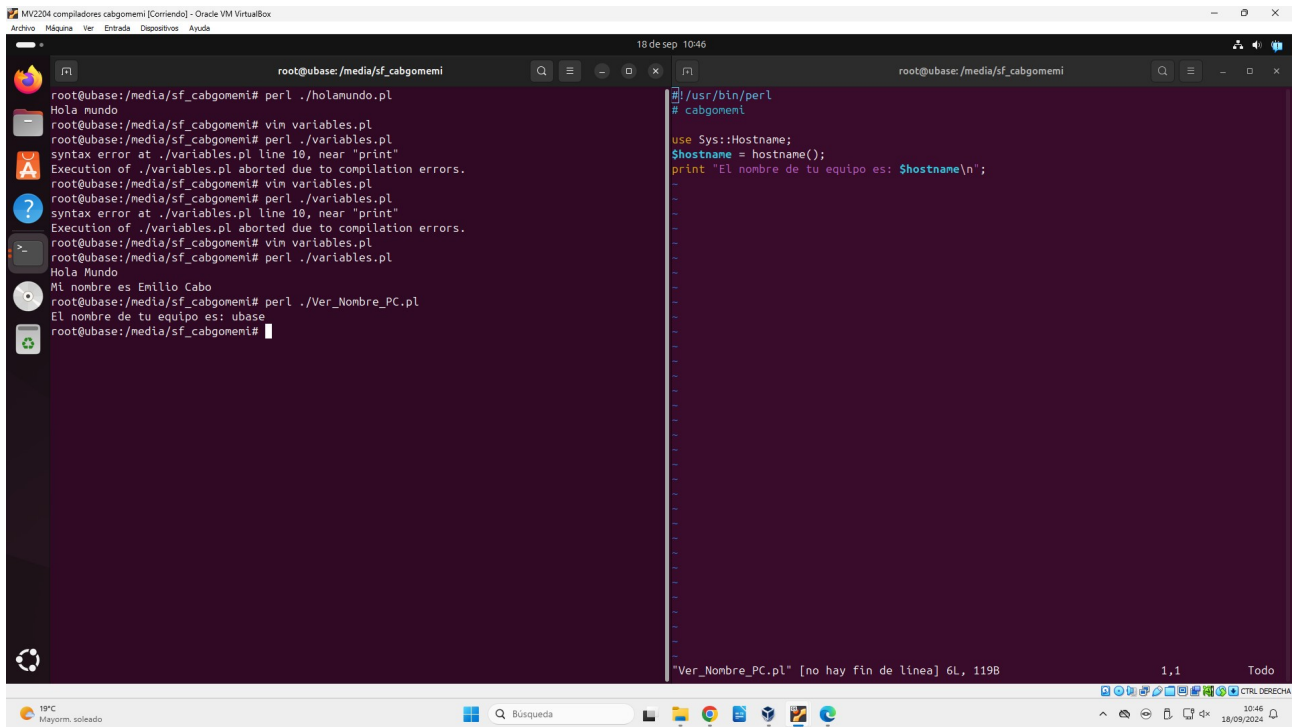
my $var1="Mi nombre es Emilio Cabo";
my $frase="Hola Mundo";

#Imprimimos en pantalla lo que acabamos de hacer
print "$frase\n";
print "$var1\n";
```

Ver nombre PC

Este programa nos permite ver el nombre del PC en el que estamos trabajando.

Este script está hecho para trabajar en Ubuntu.



The screenshot shows a terminal window titled 'root@ubase:/media/sf_cabgomemi' with the following commands and output:

```
root@ubase:/media/sf_cabgomemi# perl ./holamundo.pl
Hola mundo
root@ubase:/media/sf_cabgomemi# vim variables.pl
root@ubase:/media/sf_cabgomemi# perl ./variables.pl
syntax error at ./variables.pl line 10, near "print"
Execution of ./variables.pl aborted due to compilation errors.
root@ubase:/media/sf_cabgomemi# vim variables.pl
root@ubase:/media/sf_cabgomemi# perl ./variables.pl
syntax error at ./variables.pl line 10, near "print"
Execution of ./variables.pl aborted due to compilation errors.
root@ubase:/media/sf_cabgomemi# vim variables.pl
root@ubase:/media/sf_cabgomemi# perl ./variables.pl
Hola Mundo
Mi nombre es Emilio Cabo
root@ubase:/media/sf_cabgomemi# perl ./Ver_Nombre_PC.pl
El nombre de tu equipo es: ubase
root@ubase:/media/sf_cabgomemi#
```

Overlaid on the terminal is a text editor window showing the content of `Ver_Nombre_PC.pl`:

```
#!/usr/bin/perl
# cabgomem

use Sys::Hostname;
$hostname = hostname();
print "El nombre de tu equipo es: $hostname\n";
```

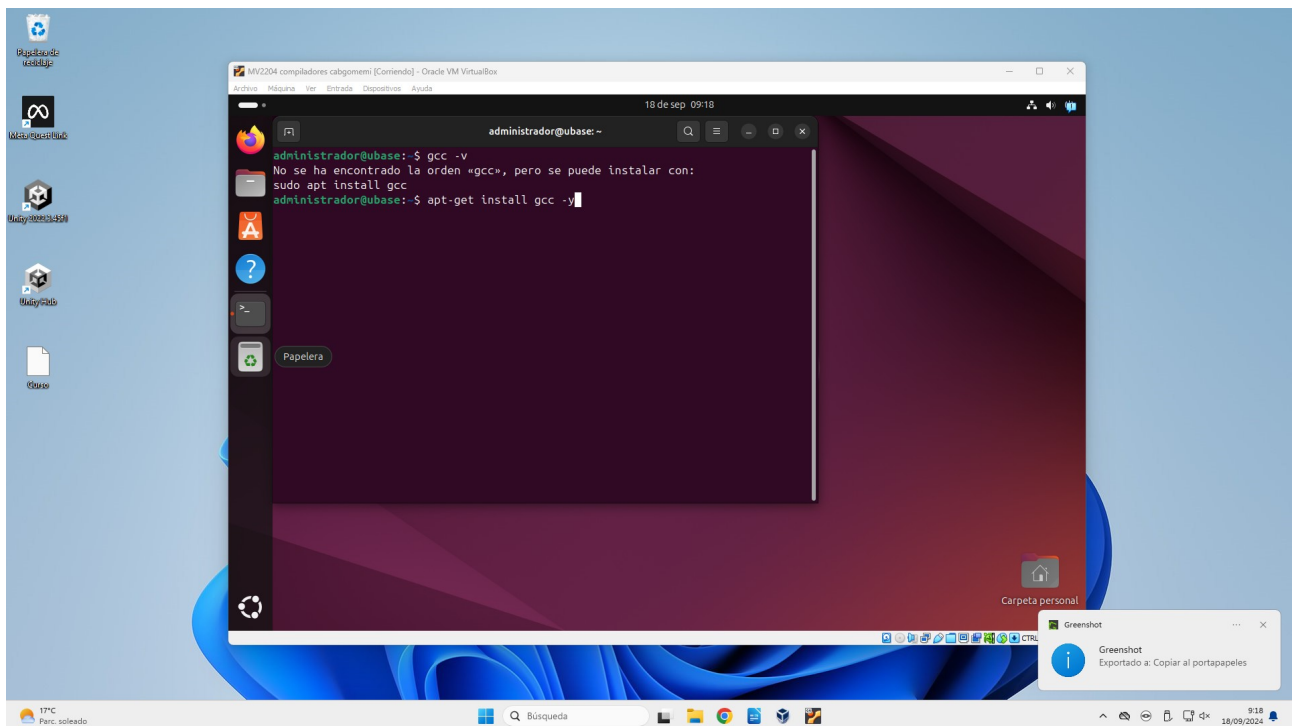
The bottom of the image shows the Windows taskbar with the date 18/09/2024 and time 10:46.

C

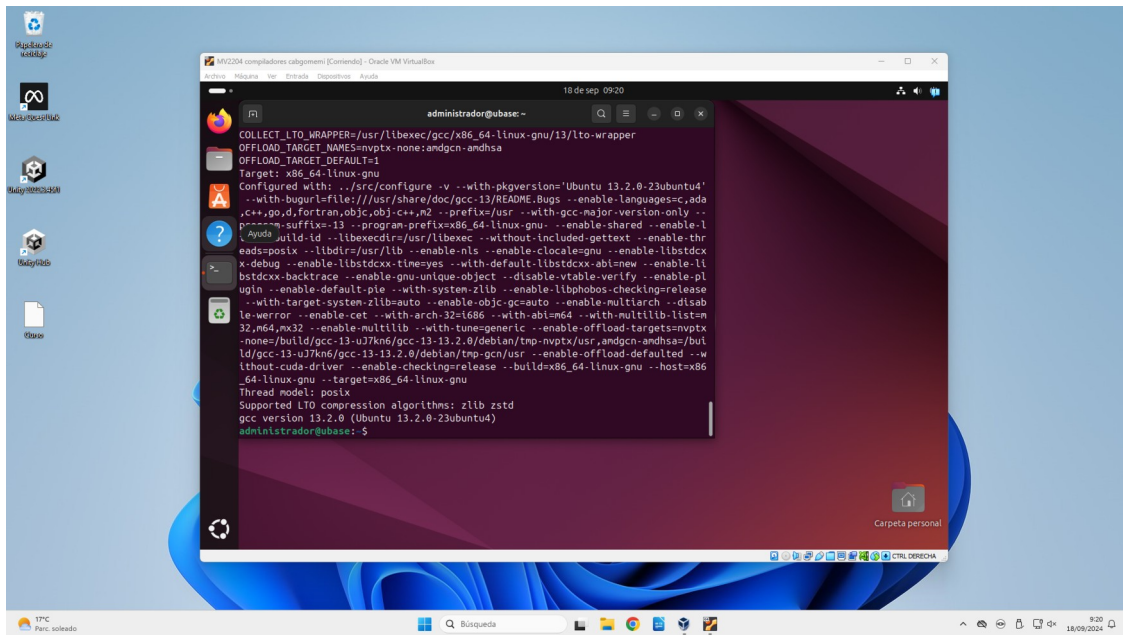
Hasta ahora los lenguajes que hemos ido empleando han sido lenguajes interpretados, es decir, cada línea iba siendo traducida a código máquina por orden.

Este es un lenguaje compilado, es decir, se traduce todo el programa en su conjunto a código máquina que luego ejecutamos.

En mi caso emplearé el gcc. En ubuntu debería venir instalado. Aunque si no lo tenemos instalado mediante el siguiente comando lo podremos instalar. **“apt-get install gcc”**



Comprobamos que el compilador se ha instalado correctamente.



Este es un programa que incluye la estructura básica de los programas en C.

```
#include <stdio.h>
```

```
int main () {  
    return 0;  
}
```

Hacemos un hello world.

Lo de arriba es una librería de standard input and output. `#include <stdio.h>`

Mediante el siguiente comando compilamos el código. `gcc 'basic structure.c'`

```
root@ubase:/media/sf_programasC# ls | grep basic
basic structure.c
root@ubase:/media/sf_programasC# gcc 'basic structure.c'
root@ubase:/media/sf_programasC# gcc 'basic structure.c'
root@ubase:/media/sf_programasC# ./a
bash: ./a: No existe el archivo o el directorio
root@ubase:/media/sf_programasC#
Display all 2096 possibilities? (y or n)
root@ubase:/media/sf_programasC# ./a
adder1.c arrayasArgument.c arraytoPointer.c
a.out arrayasReference.c atm2.c
array1.c arrayofStructs.c atm.c
root@ubase:/media/sf_programasC# ./a
adder1.c arrayasArgument.c arraytoPointer.c
a.out arrayasReference.c atm2.c
array1.c arrayofStructs.c atm.c
root@ubase:/media/sf_programasC# ./a
adder1.c arrayasArgument.c arraytoPointer.c
a.out arrayasReference.c atm2.c
array1.c arrayofStructs.c atm.c
root@ubase:/media/sf_programasC# ./a.out
Hola Mundo
root@ubase:/media/sf_programasC#
```

Con los argumentos especificados ejecutamos el código mediante el comando. “./a.out”

Datatypes en c

Estos son algunos de los tipos de datos con los que trabajamos en C

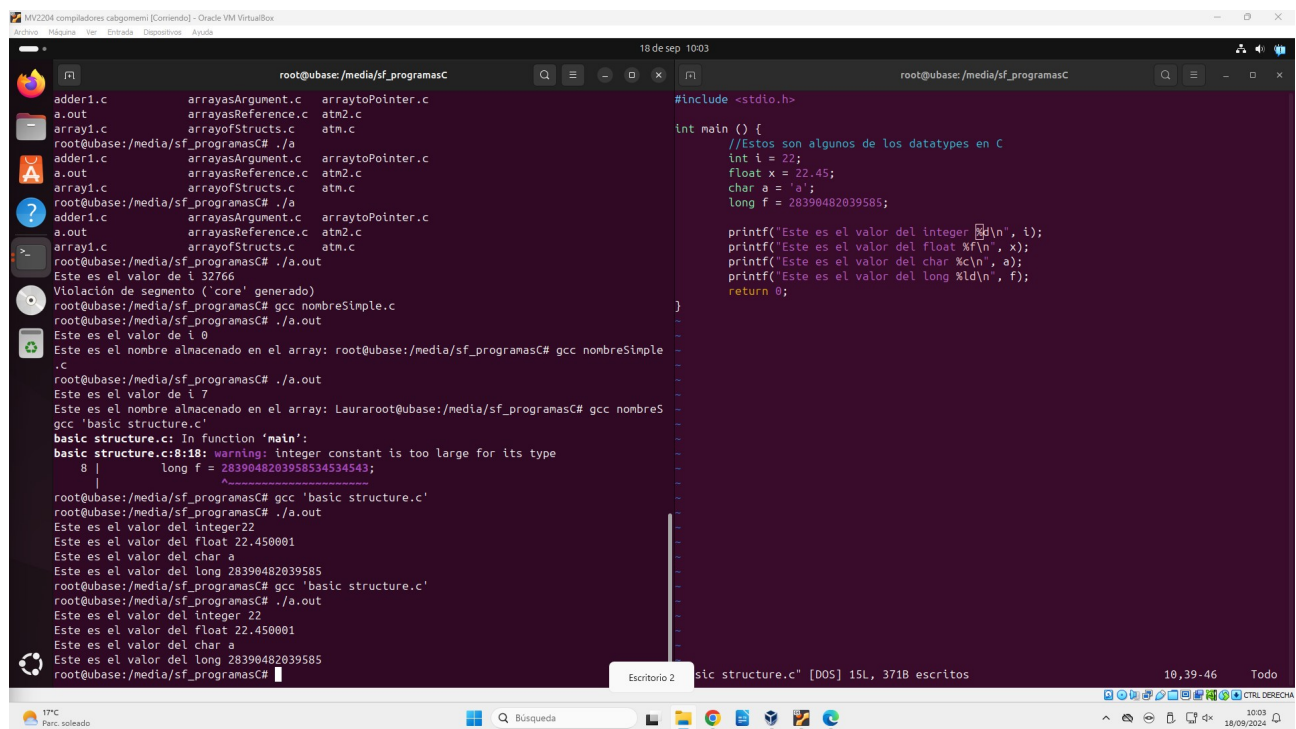
En los programas debemos especificar de que tipo son las variables que vamos a usar y cuánto queremos que ocupen en memoria. En este ejemplo pongo prácticamente todos los datatypes y los imprimo en pantalla.

```
root@ubase:/media/sf_programasC# gcc nombreSimple.c
root@ubase:/media/sf_programasC# ./a.out
Este es el valor de i 32766
Violación de segmento ('core' generado)
root@ubase:/media/sf_programasC# gcc nombreSimple.c
root@ubase:/media/sf_programasC# ./a.out
Este es el valor de i 0
Este es el nombre almacenado en el array: root@ubase:/media/sf_programasC# gcc nombreSimple.c
root@ubase:/media/sf_programasC# ./a.out
Este es el valor de i 7
Este es el nombre almacenado en el array: Lauraroot@ubase:/media/sf_programasC# gcc nombreSimple.c
root@ubase:/media/sf_programasC# gcc 'basic structure.c'
basic structure.c: In function 'main':
basic structure.c:8:18: warning: integer constant is too large for its type
8 |     long f = 283904820395853453453;
  |
root@ubase:/media/sf_programasC# gcc 'basic structure.c'
root@ubase:/media/sf_programasC# ./a.out
Este es el valor del integer 22
Este es el valor del float 22.450001
Este es el valor del char a
Este es el valor del long 28390482039585
root@ubase:/media/sf_programasC# gcc 'basic structure.c'
root@ubase:/media/sf_programasC# ./a.out
Este es el valor del integer 22
Este es el valor del float 22.450001
Este es el valor del char a
Este es el valor del long 28390482039585
root@ubase:/media/sf_programasC#
```

Definición de variables y trabajo con loops

En este programa trabajaremos con loops y variables.

En este caso voy a programar un programa que cuenta de 5 en 5 hasta 100.



The screenshot shows a terminal window in a virtual machine environment. The terminal is divided into two panes. The left pane shows the execution of various C programs, including 'nombreSimple.c' and 'basic structure.c'. The right pane shows the source code for 'basic structure.c', which includes standard headers and defines variables of different types (int, float, char, long) and prints their values. The output of the programs is visible in the left pane, showing the values of the variables and the results of the printf statements.

```
#include <stdio.h>

int main() {
    //Estos son algunos de los datatypes en C
    int i = 22;
    float x = 22.45;
    char a = 'a';
    long f = 28390482039585;

    printf("Este es el valor del integer %d\n", i);
    printf("Este es el valor del float %f\n", x);
    printf("Este es el valor del char %c\n", a);
    printf("Este es el valor del long %ld\n", f);
    return 0;
}
```

Output from the terminal:

```
root@ubase:/media/sf_programasC# gcc nombreSimple.c
Este es el valor de i 32766
Violación de segmento ('core' generado)
root@ubase:/media/sf_programasC# gcc nombreSimple.c
Este es el valor de i 0
Este es el nombre almacenado en el array: root@ubase:/media/sf_programasC# gcc nombreSimple.c
Este es el valor de i 7
Este es el nombre almacenado en el array: Lauraroot@ubase:/media/sf_programasC# gcc nombreSimple.c
basic structure.c: In function 'main':
basic structure.c:8:18: warning: integer constant is too large for its type
      8 |         long f = 28390482039585345343;
          |
root@ubase:/media/sf_programasC# gcc 'basic structure.c'
root@ubase:/media/sf_programasC# ./a.out
Este es el valor del integer 22
Este es el valor del float 22.450001
Este es el valor del char a
Este es el valor del long 28390482039585
root@ubase:/media/sf_programasC# gcc 'basic structure.c'
root@ubase:/media/sf_programasC# ./a.out
Este es el valor del integer 22
Este es el valor del float 22.450001
Este es el valor del char a
Este es el valor del long 28390482039585
root@ubase:/media/sf_programasC#
```

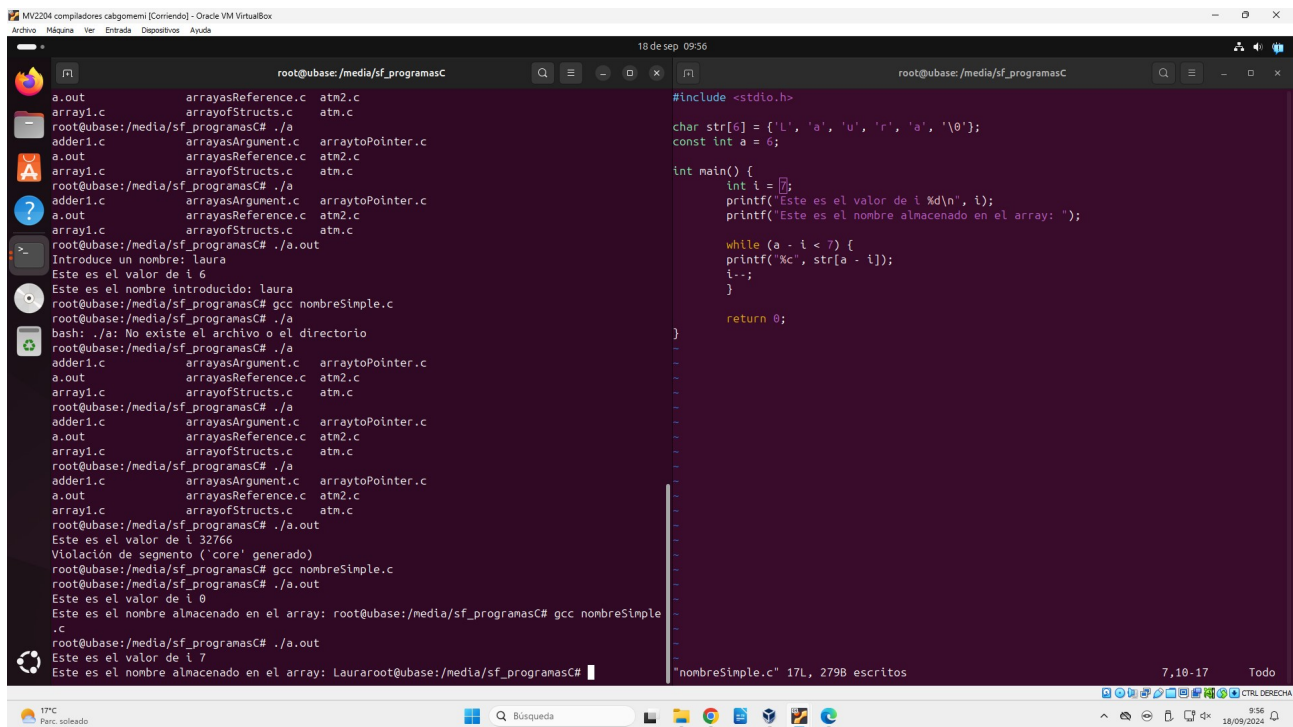

Definición de array y trabajo con loops

Para comprender el funcionamiento de este programa debemos entender que en C no existen los “Strings” sino que se almacenan como un array de caracteres.

Por ejemplo:

En nuestro caso vamos a crear un array lo suficientemente largo para almacenar “Laura”.

Debemos añadir un espacio extra para un carácter en el array llamado “Null terminating character” para decirle al ordenador que ese es el último de los caracteres del array.



The image shows a screenshot of a virtual machine environment. On the left, a terminal window displays the execution of a C program. The program prompts the user to enter a name and a value. The user enters 'Laura' and '6'. The program then prints the value and the name stored in an array. On the right, a code editor shows the source code of the program. The code includes `<stdio.h>`, defines a character array `str[6]` with the string "Laura", and a constant `int a = 6;`. The `main` function prints the value of `a` and the string stored in `str`.

```
#include <stdio.h>

char str[6] = {'L', 'a', 'u', 'r', 'a', '\0'};
const int a = 6;

int main() {
    int i = 0;
    printf("Este es el valor de i %d\n", i);
    printf("Este es el nombre almacenado en el array: ");

    while (a - i < 7) {
        printf("%c", str[a - i]);
        i--;
    }

    return 0;
}
```

STDIN a STDOUT y array

En este caso vamos a avanzar un poco en el anterior programa y vamos a hacer que el contenido del array de caracteres sea cogido mediante input del usuario.

Debemos introducir un nombre de como máximo 5 caracteres de longitud.

Las líneas con “//” son comentarios que incluyen implementaciones antiguas del código que a alguien le pueden resultar útiles.

Programar es un proceso creativo. Existen múltiples soluciones para un determinado problema.

MV2204 compiladores cabgomi [Corriendo] - Oracle VM VirtualBox

18 de sep 09:45

```
root@ubase:/media/sf_programasC
root@ubase:/media/sf_programasC# ./a
adder1.c      arrayasArgument.c  arraytoPointer.c
a.out         arrayasReference.c  atm2.c
array1.c      arrayofStructs.c   atm.c
root@ubase:/media/sf_programasC# ./a
adder1.c      arrayasArgument.c  arraytoPointer.c
a.out         arrayasReference.c  atm2.c
array1.c      arrayofStructs.c   atm.c
root@ubase:/media/sf_programasC# ./a
adder1.c      arrayasArgument.c  arraytoPointer.c
a.out         arrayasReference.c  atm2.c
array1.c      arrayofStructs.c   atm.c
root@ubase:/media/sf_programasC# ./a.out
Hola Mundo
root@ubase:/media/sf_programasC# gcc 23_nombre.c
23_nombre.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
6 | main() {
  | ^~~~~~
root@ubase:/media/sf_programasC# gcc 23_nombre.c
root@ubase:/media/sf_programasC# ./a
adder1.c      arrayasArgument.c  arraytoPointer.c
a.out         arrayasReference.c  atm2.c
array1.c      arrayofStructs.c   atm.c
root@ubase:/media/sf_programasC# ./a
adder1.c      arrayasArgument.c  arraytoPointer.c
a.out         arrayasReference.c  atm2.c
array1.c      arrayofStructs.c   atm.c
root@ubase:/media/sf_programasC# ./a
adder1.c      arrayasArgument.c  arraytoPointer.c
a.out         arrayasReference.c  atm2.c
array1.c      arrayofStructs.c   atm.c
root@ubase:/media/sf_programasC# ./a.out
Introduce un nombre: laura
Este es el valor de i 6
Este es el nombre introducido: laura
root@ubase:/media/sf_programasC#
```

```
#include <stdio.h>

char str[6];
const int a = 6;

int main() {
    int i;
    printf("Introduce un nombre: ");

    for (i = 0; i < 6; i++) {
        scanf("%c", &str[i]);
    }

    //Null termination del string LAURA en posicion 6
    str[6] = '\0';

    printf("Este es el valor de i %d\n", i);
    //printf("%c%c%c%c%c%c",str[0], str[1], str[2], str[3], str[4], str[5] );
    printf("Este es el nombre introducido: ");

    while (a - i < 7) {
        //int b = a - i;
        printf("%c", str[a - i]);
        i--;
    }

    return 0;
}
```

"23_nombre.c" 28L, 481B escritos

28,0-1 Todo

17°C Parc. soleado

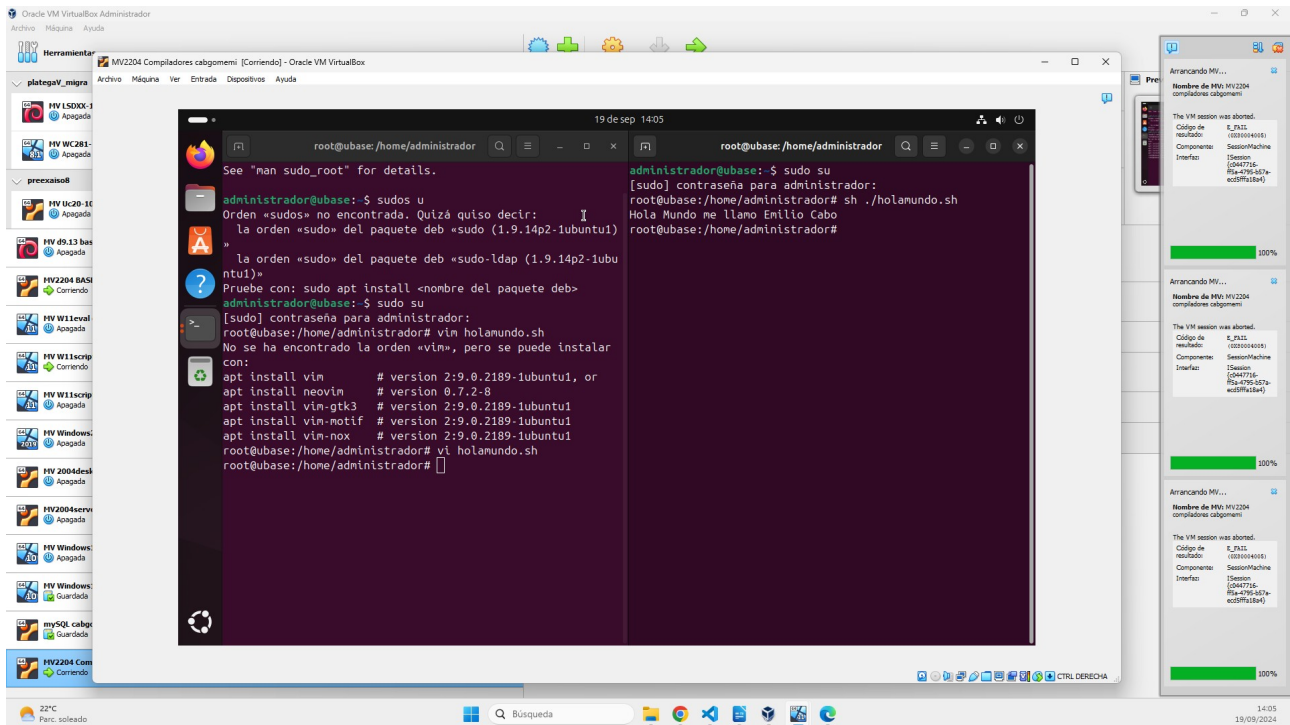
Búsqueda

9:45 18/09/2024

BASH

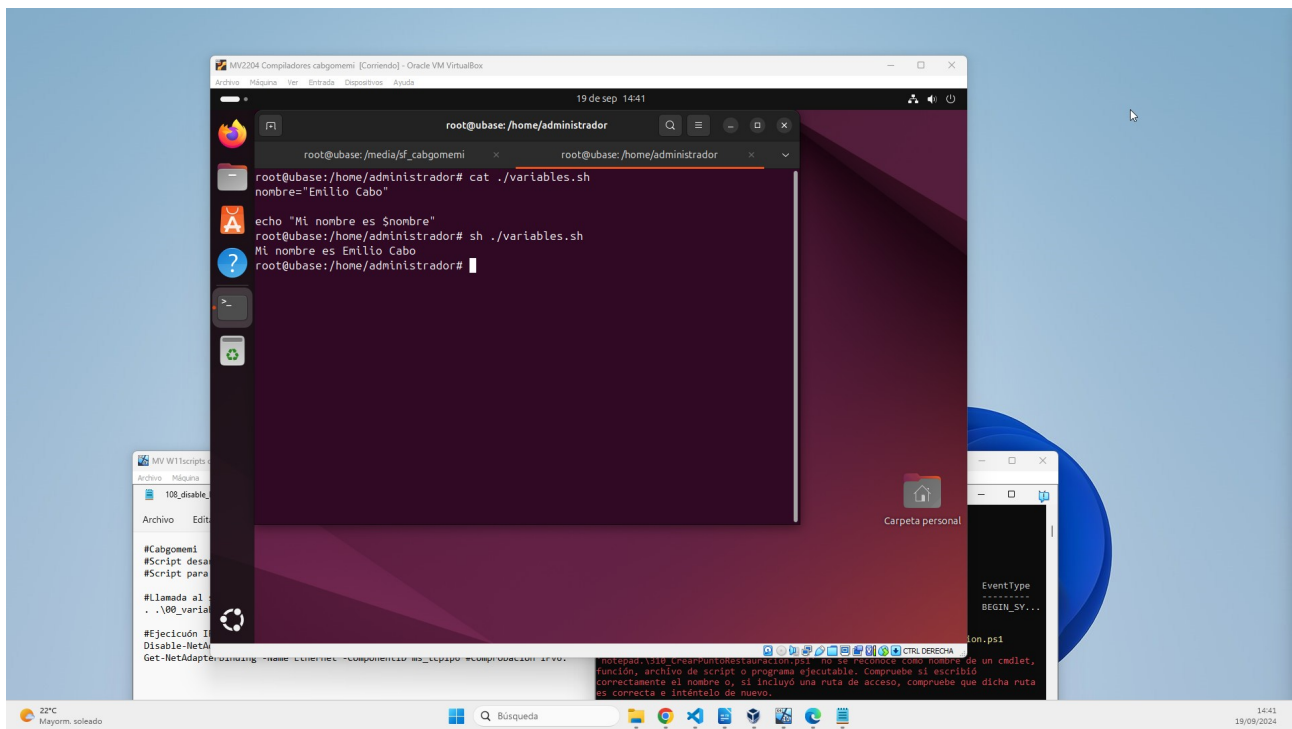
Hola mundo

Ejecutamos un hola mundo sencillo.

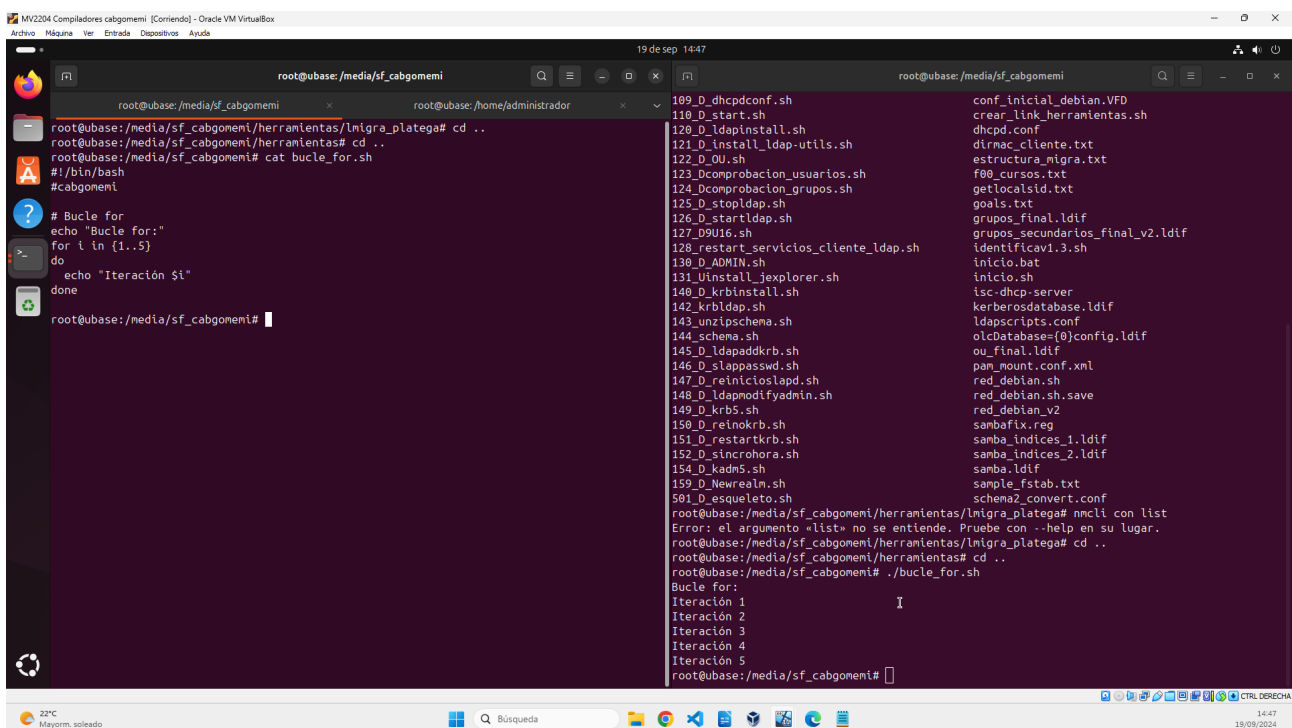


Variables en BASH

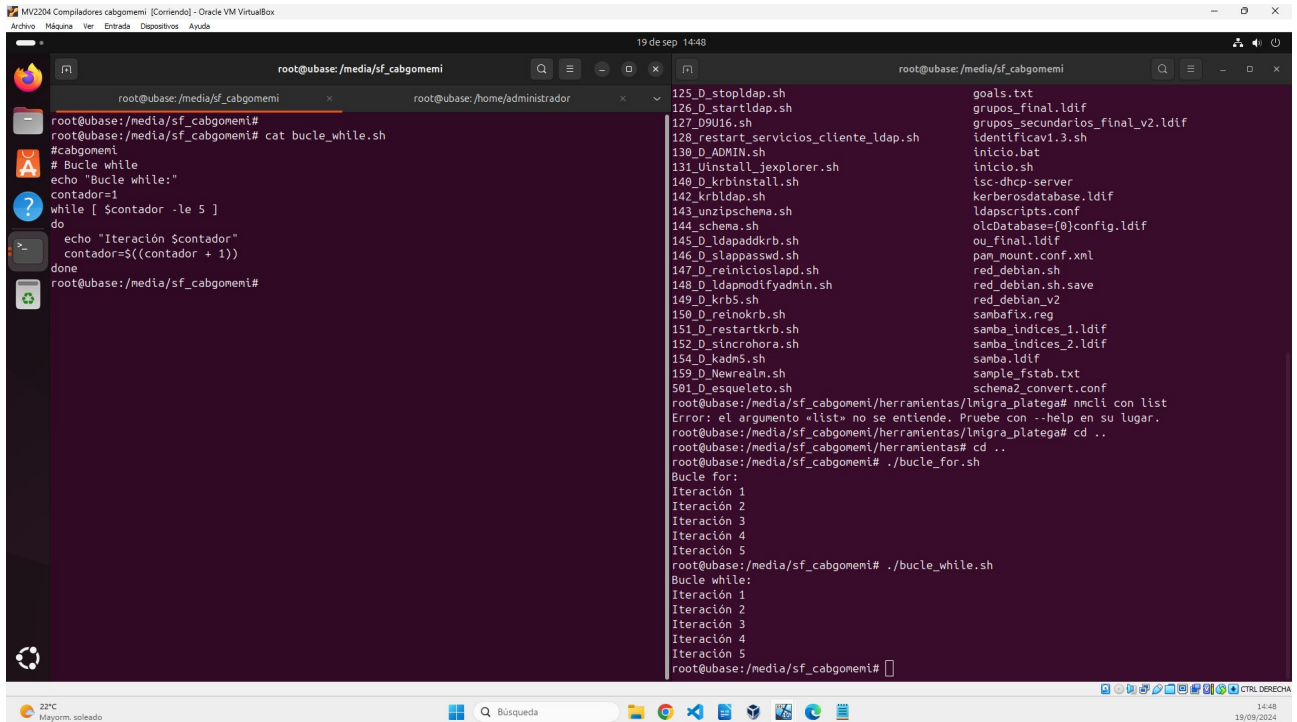
Almacenamos nuestro nombre en una variable y luego lo imprimimos en pantalla.



Bucle for



Bucle While



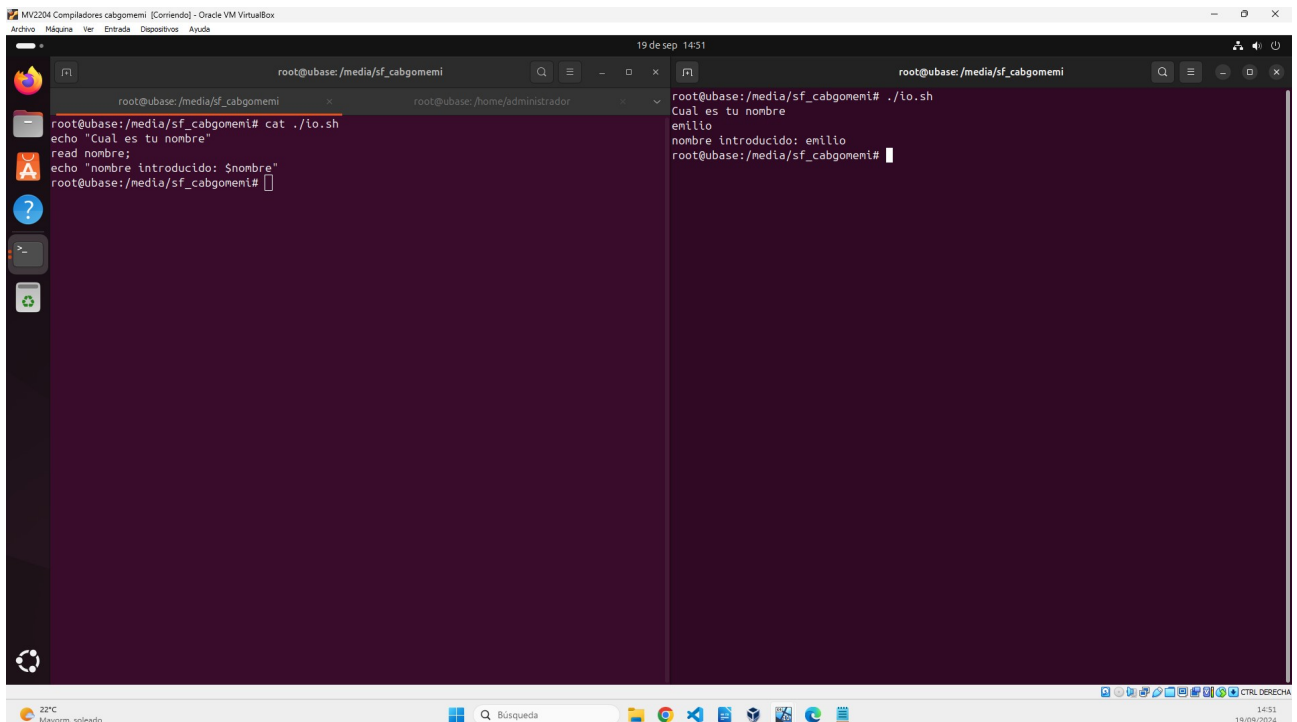
The screenshot shows a terminal window with a dark background. The prompt is `root@ubase:/media/sf_cabgonemi#`. The user enters `cat bucle_while.sh`, and the terminal displays the contents of the script. The script is a while loop that runs 5 iterations, printing "Iteración" followed by the iteration number. The user then enters `./bucle_while.sh`, and the terminal shows the output of the script, which prints "Iteración 1" through "Iteración 5".

```
root@ubase:/media/sf_cabgonemi# cat bucle_while.sh
# Bucle while
echo "Bucle while:"
contador=1
while [ $contador -le 5 ]
do
    echo "Iteración $contador"
    contador=$((contador + 1))
done
root@ubase:/media/sf_cabgonemi#

root@ubase:/media/sf_cabgonemi# ./bucle_while.sh
Bucle while:
Iteración 1
Iteración 2
Iteración 3
Iteración 4
Iteración 5
root@ubase:/media/sf_cabgonemi#
```

Input y output

En este script le solicitamos al usuario que introduzca un nombre y luego lo imprimimos en el terminal.



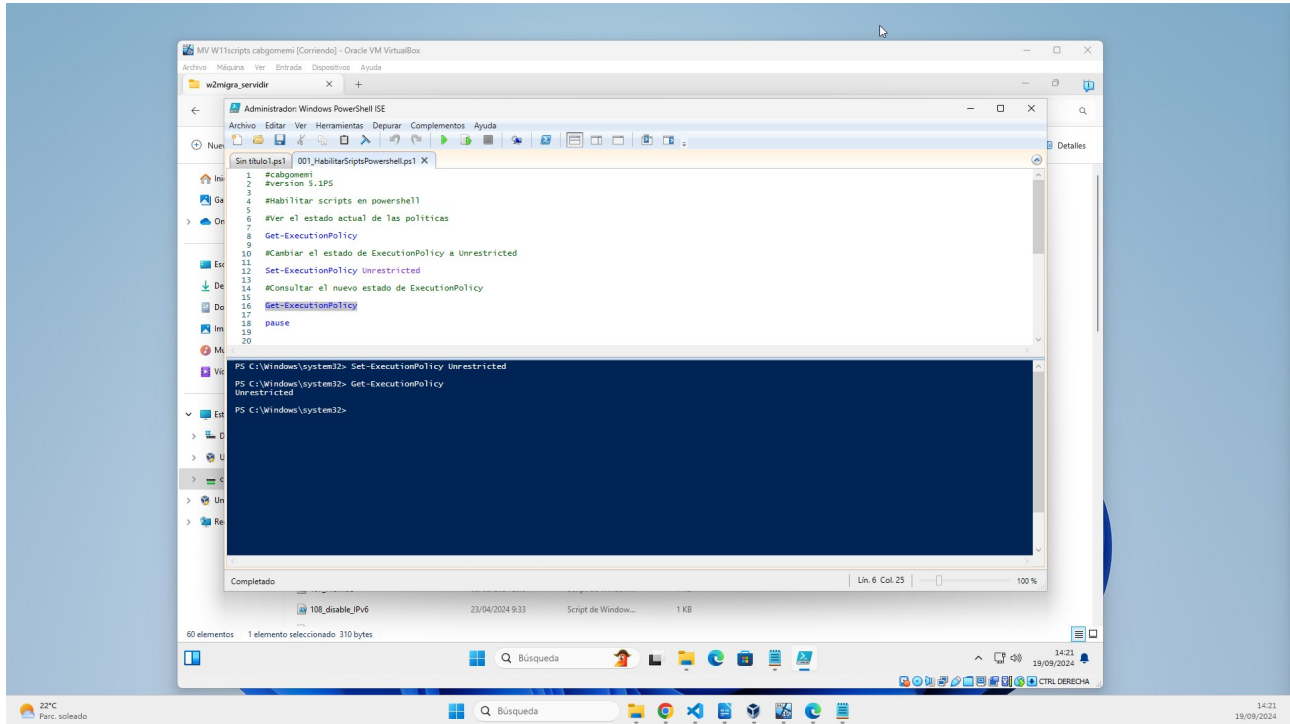
The screenshot shows a terminal window with a dark background. The prompt is `root@ubase:/media/sf_cabgonemi#`. The user enters `cat ./to.sh`, and the terminal displays the contents of the script. The script prompts the user for their name and prints it. The user then enters `./to.sh`, and the terminal shows the output, which prompts the user for their name and prints "nombre introducido: emilio".

```
root@ubase:/media/sf_cabgonemi# cat ./to.sh
echo "Cual es tu nombre"
read nombre;
echo "nombre introducido: $nombre"
root@ubase:/media/sf_cabgonemi#

root@ubase:/media/sf_cabgonemi# ./to.sh
Cual es tu nombre
emilio
nombre introducido: emilio
root@ubase:/media/sf_cabgonemi#
```

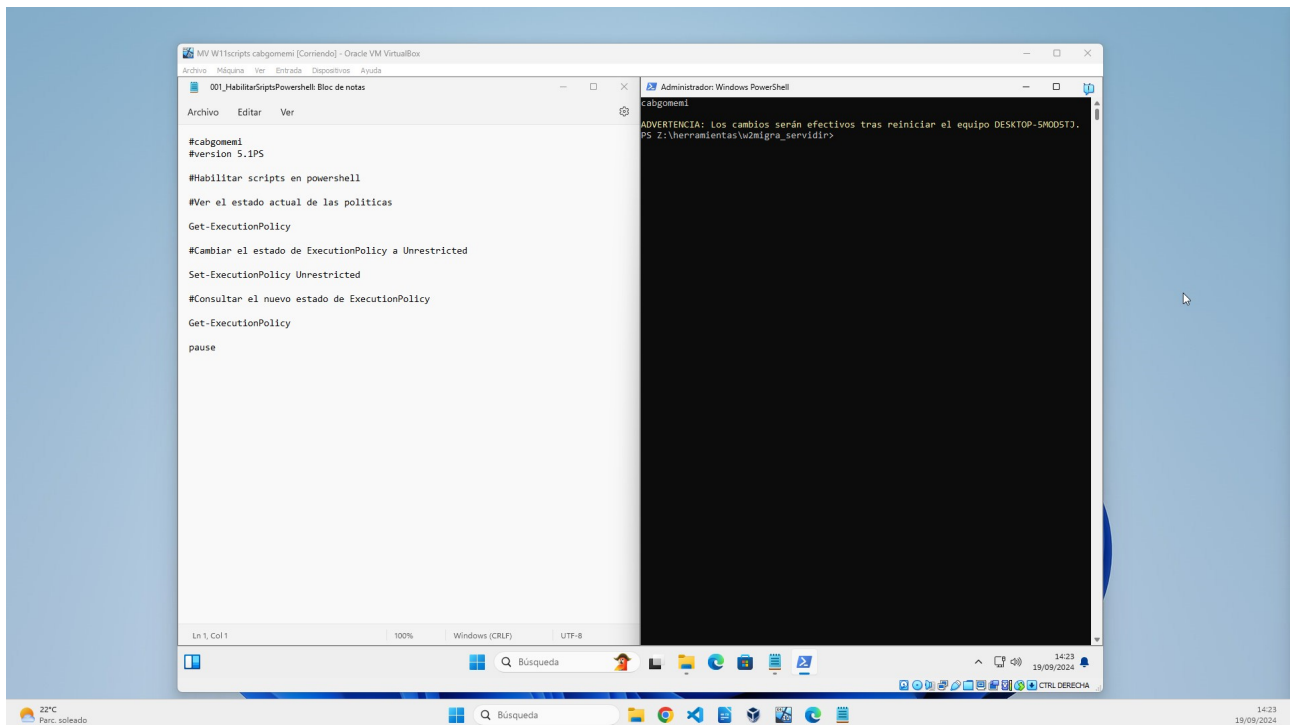
Powershell

Habilitar Scripts Powershell

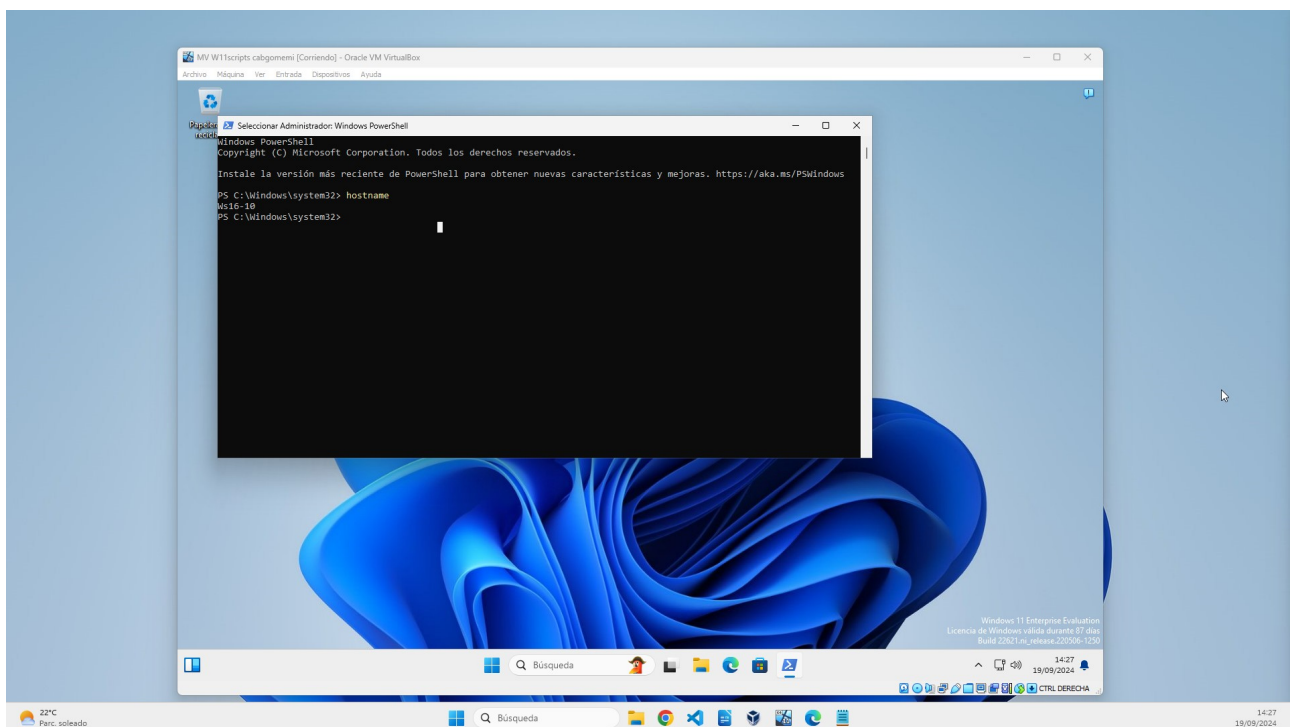


Cambiar nombre de la máquina

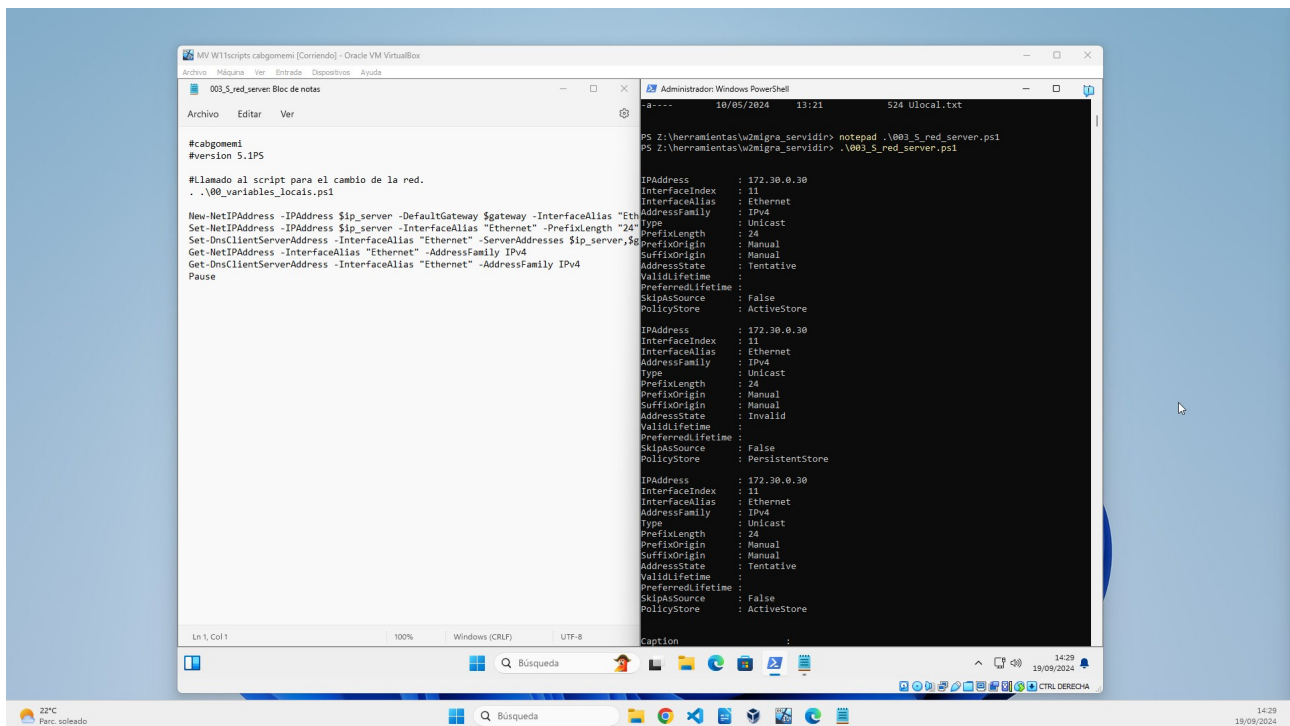
Deberemos reiniciar la máquina para que los cambios sean efectivos.



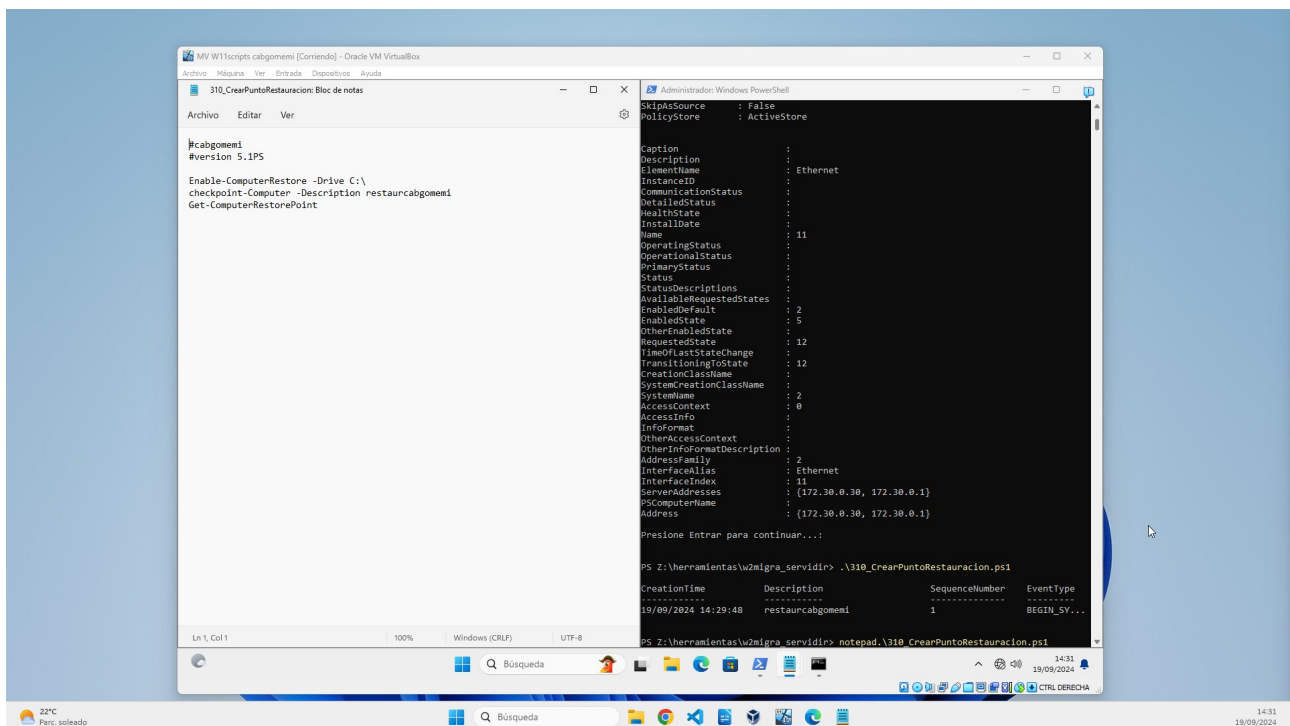
Comprobación de la ejecución del programa.



Cambiar la configuración de red del sistemas

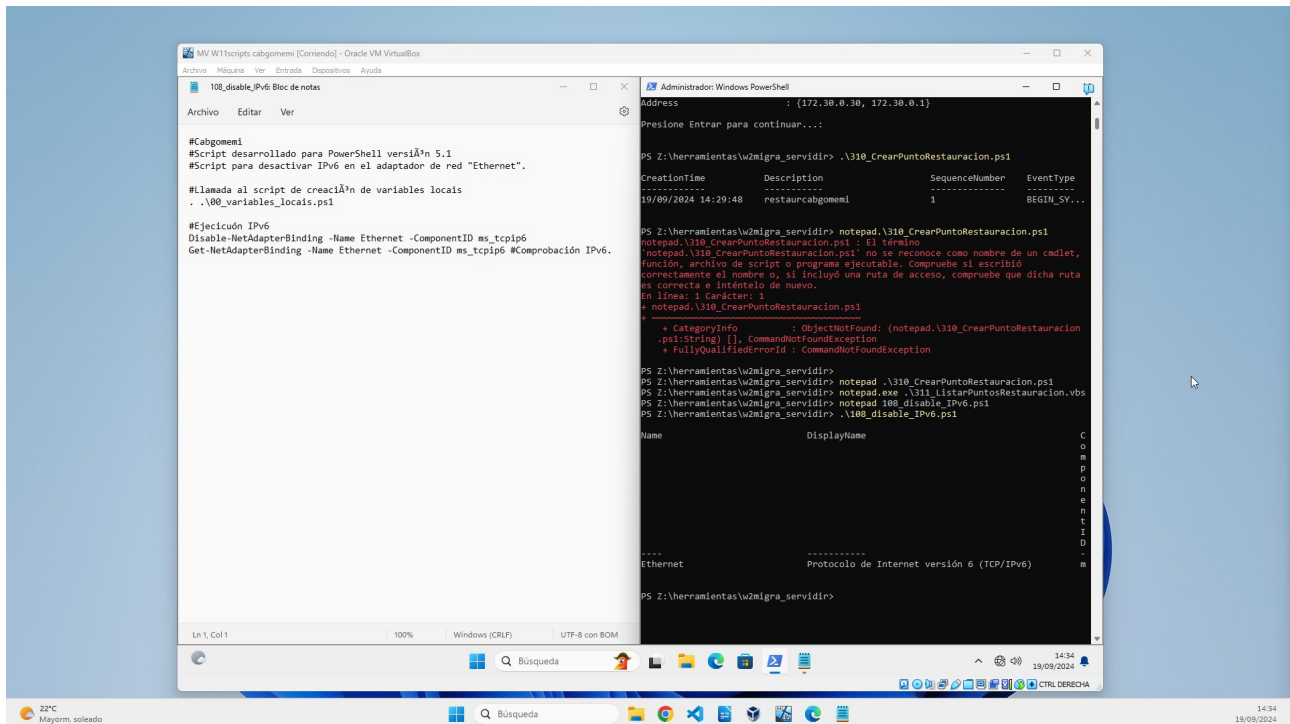


Crear punto de restauración del sistema



Desactivar IPv6

Este es un script para el desactivado de IPv6



Bibliografía

Python

<https://wiki.python.org/moin/BeginnersGuide>

PERL

[/aso/1.Administracion%20del%20Servizo%20Directorio/11.Perl1/perl.pdf](#)