

## Práctica de laboratorio: programación básica de Python (Versión para el instructor)

**Nota para el instructor:** El color de fuente rojo o las partes resaltadas en gris indican texto que solamente aparece en la copia del instructor.

### Objetivos

**Parte 1:** inicie VirtualBox e introduzca la VM del servidor I2IoT

**Parte 2:** aspectos básicos de Python

**Parte 3:** IDLE para Python

### Aspectos básicos

Python, un lenguaje de programación, permite instrucciones más simples. Python es muy fácil de usar, potente y versátil. Se ha convertido en el lenguaje de preferencia de muchos desarrolladores de IoT. Uno de los motivos principales de la popularidad de Python es la comunidad de desarrolladores. Los desarrolladores de Python han creado y puesto a disposición muchos módulos específicos que pueden importarse a cualquier programa para prestar inmediatamente mayor funcionalidad.

### Situación

En esta práctica de laboratorio, aprenderá algunos tipos de programación básica de Python y realizará ejercicios sobre esto. Más específicamente, usaremos la versión 3 de Python en la práctica de laboratorio.

### Recursos necesarios

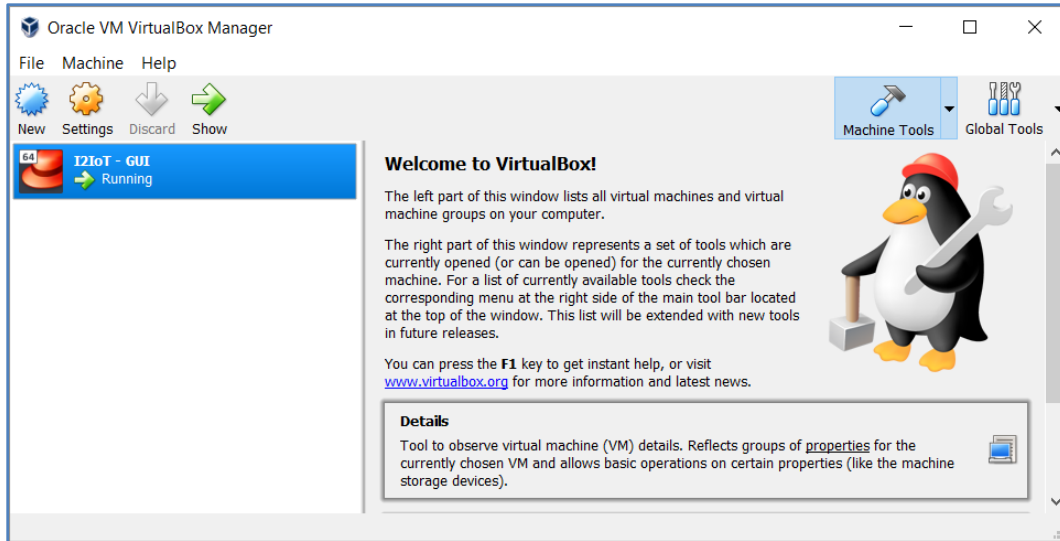
- Una computadora personal moderna con acceso a Internet y suficiente RAM.
- VirtualBox con servidor I2IoT instalado.

## Parte 1: Inicie VirtualBox e ingrese la VM del servidor I2IoT

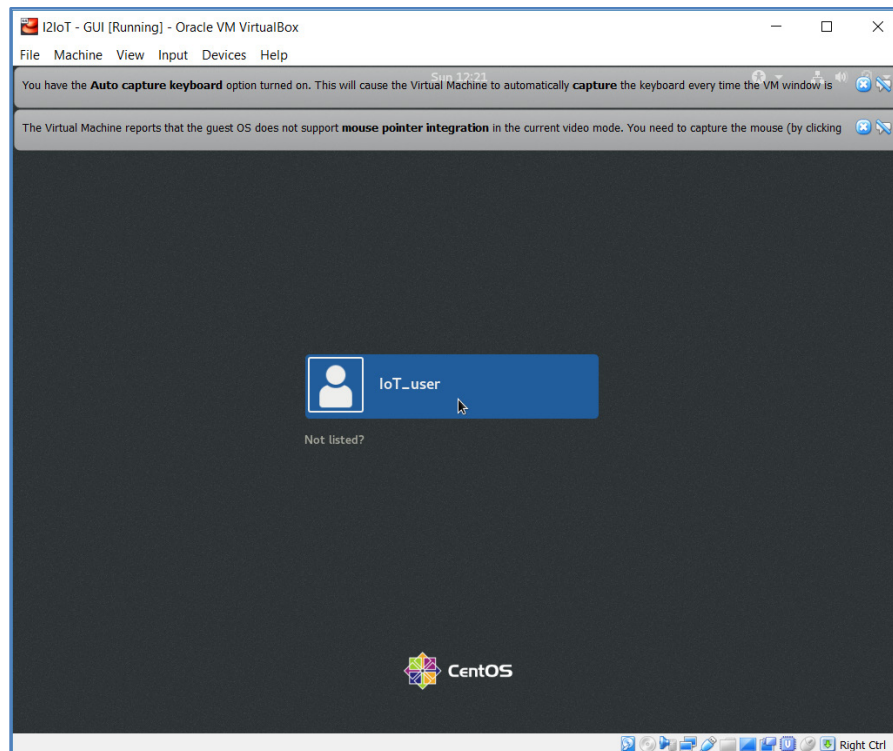
En la parte 1, debe ejecutar el software de virtualización VirtualBox e iniciar sesión en la VM del servidor I2IoT.

### Paso 1: Inicie VirtualBox.

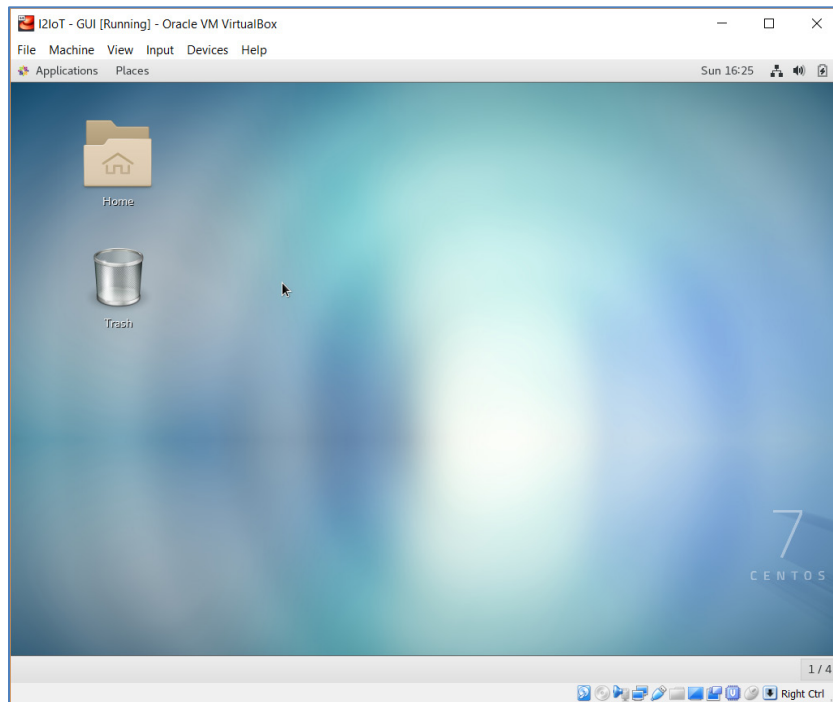
- a. Una vez que se haya instalado VirtualBox (consulte la práctica de laboratorio 2.1.3.6 ), el icono de VirtualBox debería aparecer en el escritorio. Haga clic en el icono para iniciar VirtualBox.



- b. Haga clic en **I2IoT – GUI** en el panel izquierdo para iniciar la VM del servidor.

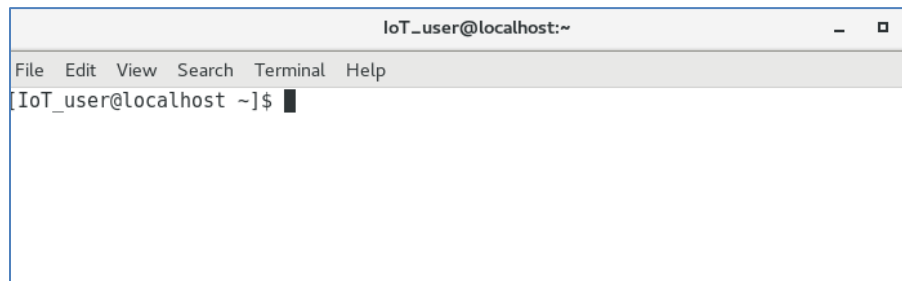


- c. El nombre de usuario predeterminado es `IoT_user`, sin contraseña. Haga clic en la barra azul **IoT\_user** que está en el medio de la pantalla para iniciar sesión en la VM.

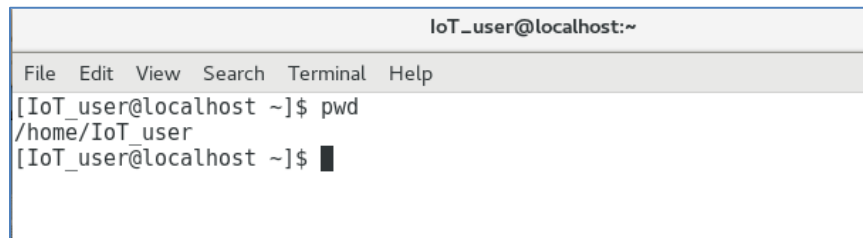


### Paso 2: Navegue hasta el directorio de documentos del usuario

- a. Para acceder a la interfaz de línea de comandos, haga clic en **Application** (Aplicación) en la barra de menús y elija **Terminal**.



- b. Utilice el comando **pwd** para ver el directorio actual.



- c. Utilice el comando **ls** para ver la lista de contenido en el directorio actual. Utilice el comando **ls** con la opción **-l** para ver información detallada sobre el contenido.

```
IoT_user@localhost:~  
File Edit View Search Terminal Help  
[IoT_user@localhost ~]$ pwd  
/home/IoT_user  
[IoT_user@localhost ~]$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
[IoT_user@localhost ~]$ ls -l  
total 0  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Desktop  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Documents  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 15:24 Downloads  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Music  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Pictures  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Public  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Templates  
drwxr-xr-x. 2 IoT_user IoT_user 6 Apr 16 10:55 Videos
```

- d. Utilice el comando **cd Documents** para cambiar el directorio al directorio `/home/IoT_user/Documents`. Verifique mediante el comando **pwd**.

```
IoT_user@localhost:~/Documents  
File Edit View Search Terminal Help  
[IoT_user@localhost ~]$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
[IoT_user@localhost ~]$ cd Documents  
[IoT_user@localhost Documents]$ pwd  
/home/IoT_user/Documents  
[IoT_user@localhost Documents]$
```

- e. Para verificar la versión de Python instalada en la VM, emita el comando **python3 --version**.

```
IoT_user@localhost:~/Documents  
File Edit View Search Terminal Help  
[IoT_user@localhost ~]$ cd Documents/  
[IoT_user@localhost Documents]$ python3 --version  
Python 3.6.5  
[IoT_user@localhost Documents]$
```

## Parte 2: Aspectos básicos de Python

En la parte 2, aprenderá algunos tipos de programación básica de Python y realizará ejercicios sobre esto.

### Paso 1: Programación de Python en el intérprete interactivo.

Como lenguaje interpretado, los comandos de Python se pueden emitir en un intérprete interactivo.

- a. Utilice el comando **python3** para iniciar el intérprete de Python.

```
File Edit View Search Terminal Help
[IoT_user@localhost ~]$ cd Documents/
[IoT_user@localhost Documents]$ python3 --version
Python 3.6.5
[IoT_user@localhost Documents]$ python3
Python 3.6.5 (default, Apr 16 2018, 15:31:49)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

- b. Realice los cálculos.

```
>>> 1 + 2
3
>>> 2 * 4
8
>>> 6 / 2
3.0
>>>
```

- c. Escriba una cadena de texto.

```
>>> "¿Cómo está?"
'¿Cómo está?'
>>>
```

- d. Utilice el comando **type()** para determinar el tipo de datos básico: int, float, string, Booleano.

```
>>> type(65)
<class 'int'>
>>> type(45.6)
<class 'float'>
>>> type("¡Hola!")
<class 'str'>
>>> type(True)
<class 'bool'>
>>> 1<2
True
>>> 1<1
False
>>> 1==1
True
>>> 1>=1
True
>>>
```

- e. Cree una variable.

```
>>> x=3
>>> x*5
15
>>> "¡Bien!"*x
'¡Bien!¡Bien!¡Bien!'
```

- f. Combine varias cadenas e introdúzcalas en una sola.

```
>>> str1="Cisco"
>>> str2="Networking"
>>> str3="Academy"
>>> space=" "
>>> print(str1+space+str2+space+str3)
Cisco Networking Academy
>>>
```

- g. Convierta el tipo de dato de numérico a una cadena.

```
>>> x=5
>>> str(x)
>>> '5'
>>> y=4.2
>>> str(y)
>>> y='4.2'
>>>
```

- h. Tenga en cuenta que los números enteros no se redondean al realizar la conversión de números decimales. Se omite el número decimal.

```
>>> int(8.21)
8
>>> int(8.99)
8
>>> int(8.21) + int(8.99)
16
>>>
```

- i. Convierta un número entero en un número decimal.

```
>>> x=5
>>> x
5
>>> float(x)
5
>>> type(x)
<class 'int'>
>>> x=float(x)
>>> type(x)
<class 'float'>
>>> x
5
>>>
```

- j. Obtenga la entrada del usuario.

```
>>> name=input("¿Cuál es su nombre? ")
¿Cuál es su nombre? John
>>> print("Hola " + nombre + ", mucho gusto")
Hola John, mucho gusto.
>>>
```

- k. Use **quit()** para salir del intérprete interactivo.

## Parte 3: IDLE para Python

IDLE significa Desarrollo Integrado y Entorno de Aprendizaje en inglés. Es admitido y está incluido en el paquete de Python. Algunas de las características clave de IDLE para Python son las siguientes:

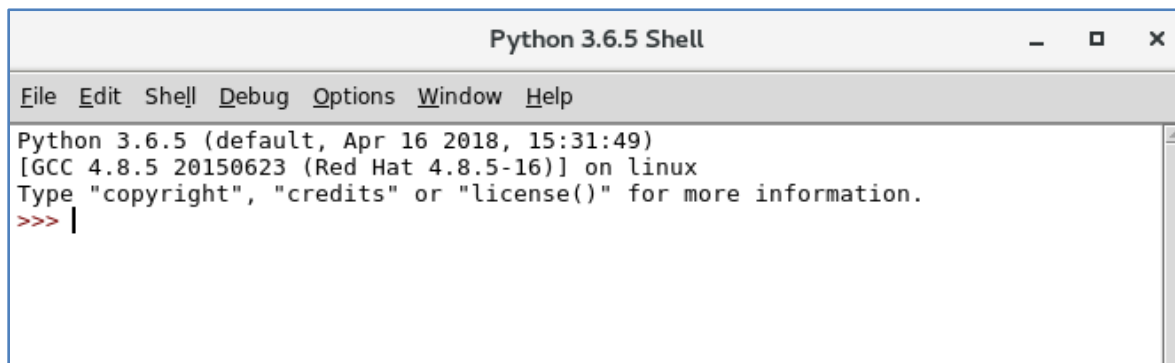
- Una ventana de shell de Python (intérprete interactivo) con coloreado de mensajes de error, entrada y salida de códigos
- Un editor de texto multiventana con varias opciones de deshacer, coloreado de Python, sangría inteligente, consejos de llamada, finalización automática y otras características
- La capacidad para realizar búsquedas dentro de una ventana, realizar reemplazos dentro de las ventanas del editor y realizar búsquedas a través de varios archivos (grep)
- Un depurador con puntos de interrupción persistentes, paso a paso, y visualización de espacios de nombres locales y globales
- Configuración, navegadores y otros cuadros de diálogo

En la parte 3, iniciará IDLE y creará un script simple.

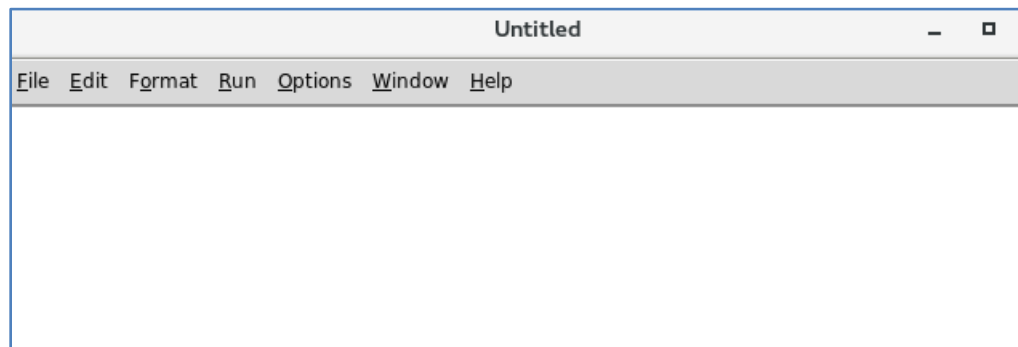
### Paso 1: Inicie IDLE.

- a. Utilice el comando **idle3** para iniciar IDLE. De manera predeterminada, se inicia en la ventana de shell de Python, o el intérprete interactivo. Ya está familiarizado con el intérprete interactivo.

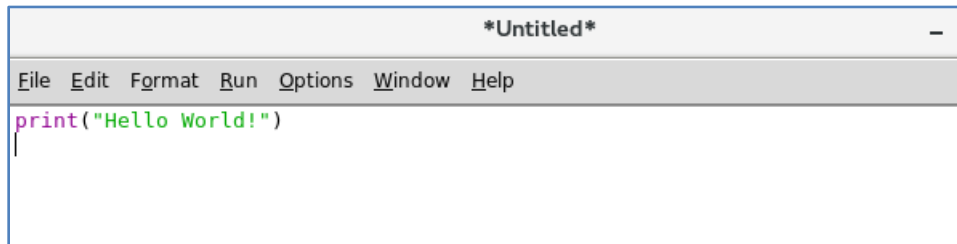
```
[IoT_user@stueverj-vm2 Documents]$ idle3
```



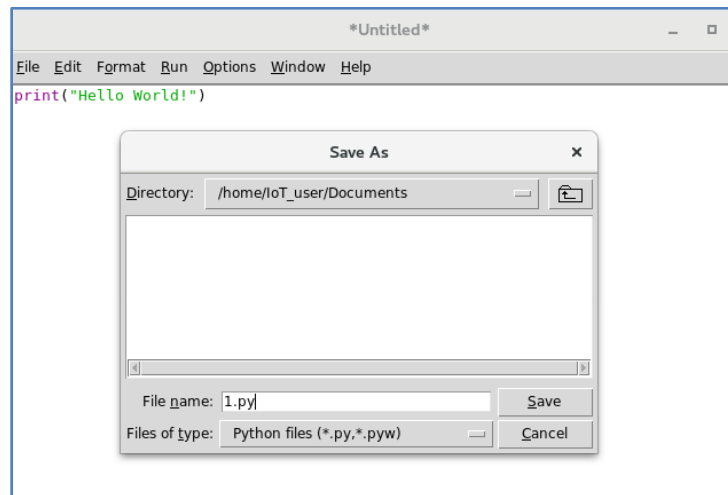
- b. Haga clic en **File > New File** (Archivo > Nuevo Archivo) para abrir un nuevo script de Python (sin título).



- c. Escriba el código en el script `print("¡Hola mundo!")`; observe que los códigos están coloreados y contienen paréntesis.



- d. Haga clic en **File > Save** (Archivo > Guardar), guarde el script actual como 1.py en el directorio actual. Haga clic en el botón **Save** (Guardar).



- e. Haga clic en **Run > Run Module** (Ejecutar > Ejecutar módulo) (o presione F5). La ventana de shell muestra el resultado.

