

## Práctica de laboratorio: cree un juego simple con IDLE 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:** cree un juego simple con IDLE 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 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, cree un juego simple utilizando IDLE de Python.

### Recursos necesarios

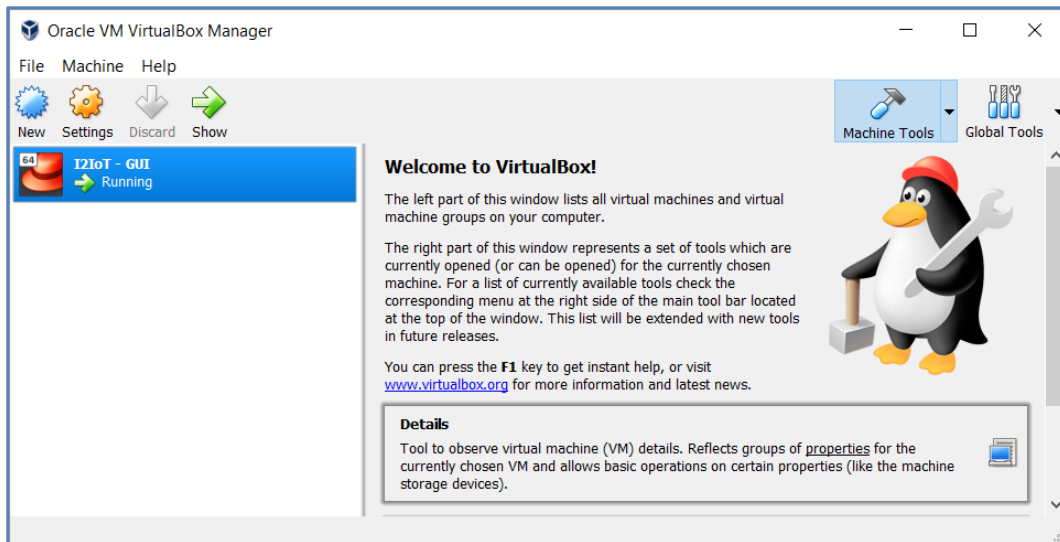
- Una computadora personal moderna con suficiente RAM y con acceso a Internet.
- VirtualBox con servidor I2IoT instalado.

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

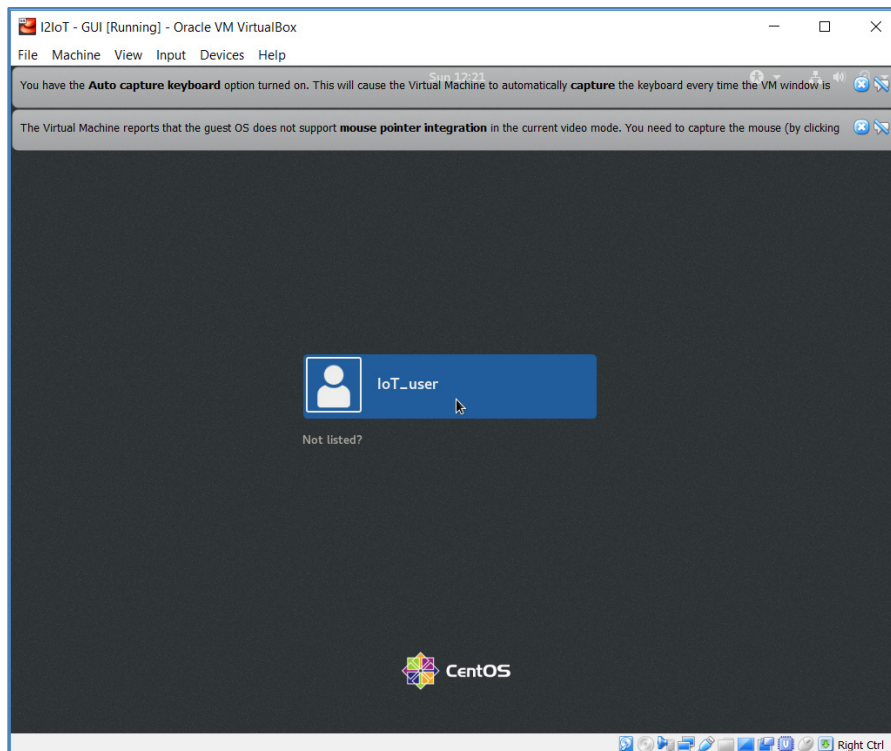
En la parte 1, debe ejecutar el software de virtualización VirtualBox e iniciar sesión en la VM del servidor I2IoT.

### Paso 1: Inicie VirtualBox.

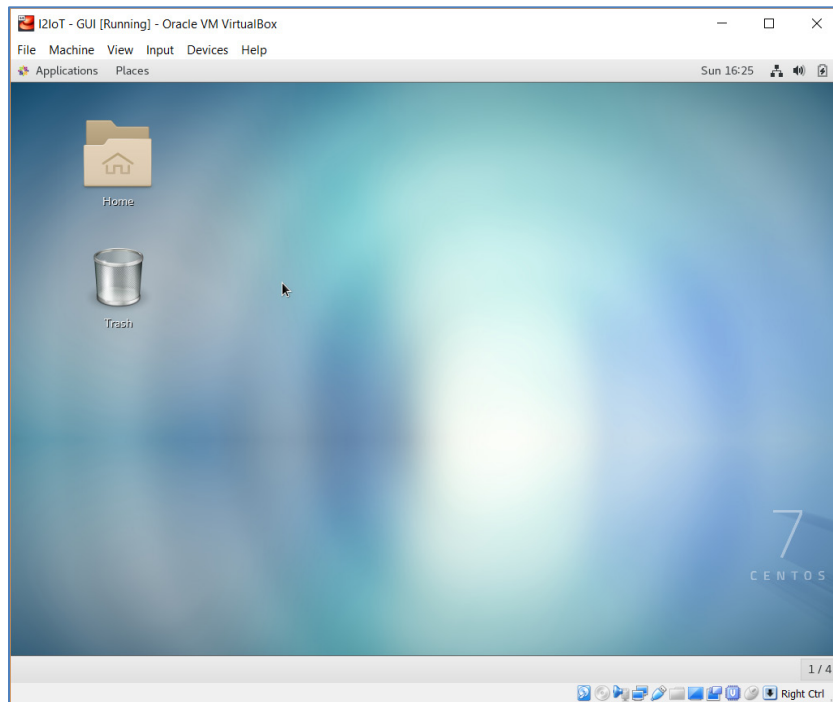
- a. Una vez que se haya instalado VirtualBox (consulte la práctica de laboratorio 2.1.3.6), el icono de VirtualBox debería aparecer en el escritorio. Haga clic en el icono para iniciar VirtualBox.



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

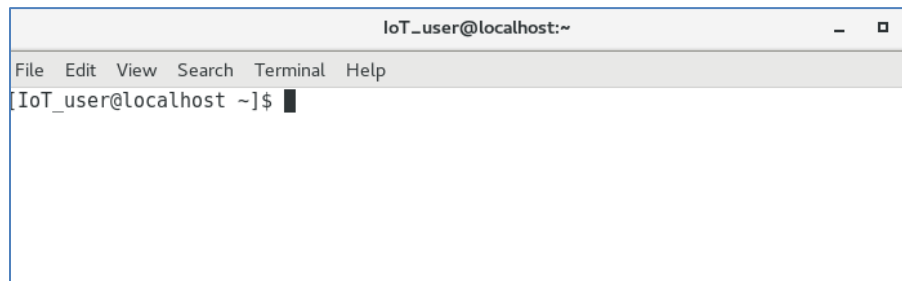


- c. El nombre de usuario predeterminado es `IoT_user`, sin contraseña. Haga clic en la barra azul **IoT\_user** que está en el medio de la pantalla para iniciar sesión en la VM.

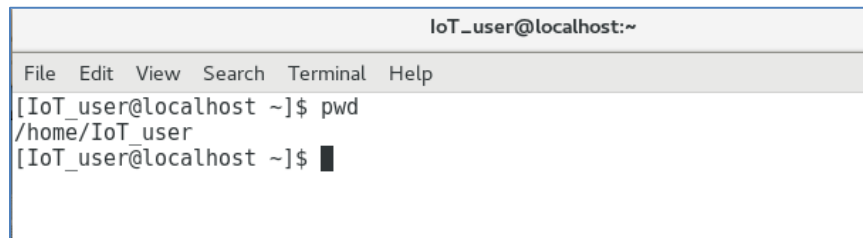


### Paso 2: Navegue hasta el directorio de documentos del usuario

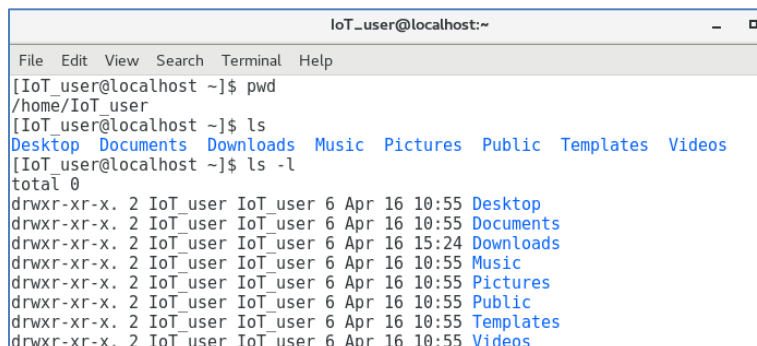
- a. Para acceder a la interfaz de línea de comandos, haga clic en **Application** (Aplicación) en la barra de menús y elija **Terminal**.



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

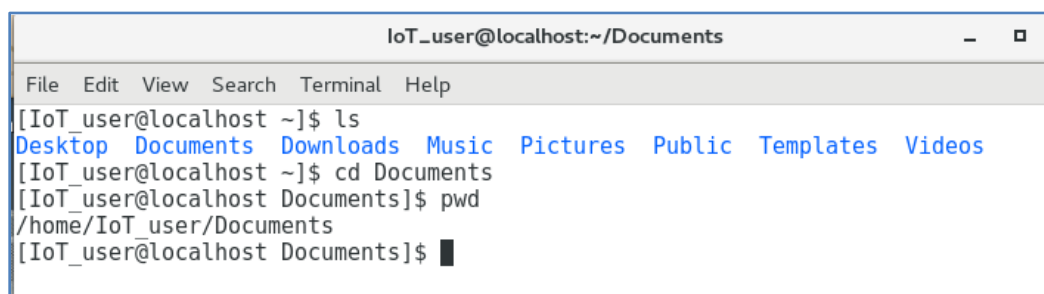


- c. Utilice el comando **ls** para ver la lista de contenido en el directorio actual. Utilice el comando **ls** con la opción **-l** para ver información detallada sobre el contenido.



```
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]$
```

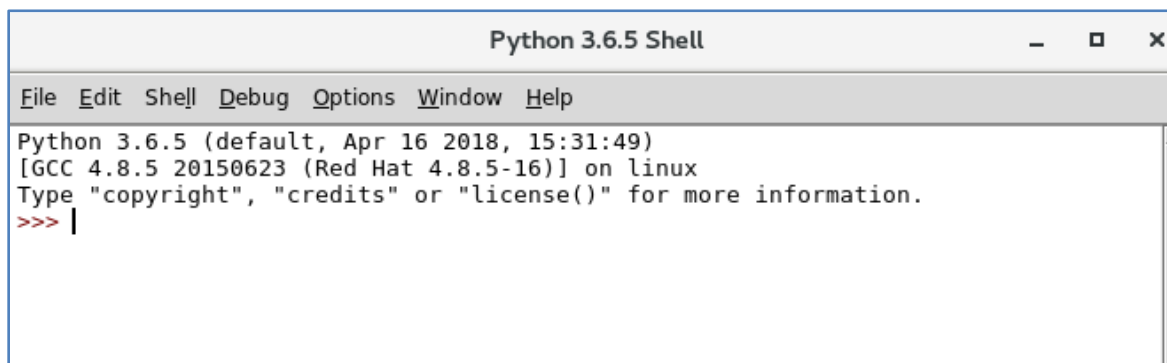
## Parte 2: Cree un juego simple con IDLE de Python

En la parte 2, creará un juego simple. El objetivo del juego es buscar un número elegido por un jugador de 0 a 1024 mediante el método de bisección.

### Paso 1: Inicie IDLE.

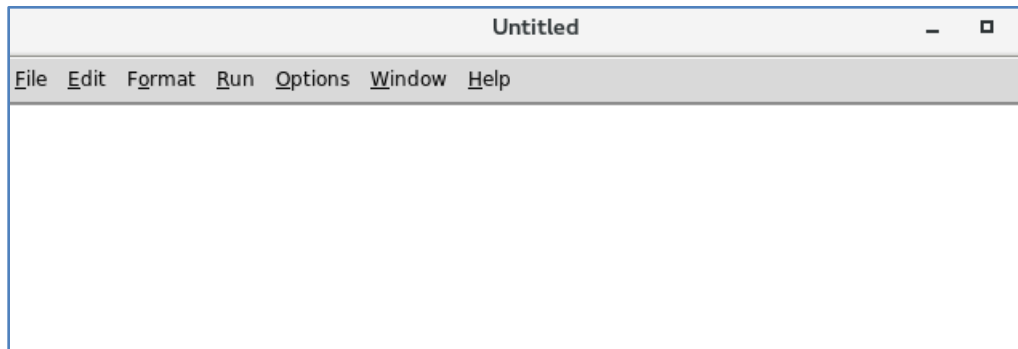
- a. Utilice el comando **idle3** para iniciar IDLE. De manera predeterminada, se inicia en la ventana de shell de Python, o el intérprete interactivo. Ya está familiarizado con el intérprete interactivo.

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



```
Python 3.6.5 Shell  
File Edit Shell Debug Options Window Help  
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 "copyright", "credits" or "license()" for more information.  
>>> |
```

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



- c. Escriba los códigos en el script, y observe que los códigos tienen color y están entre paréntesis.

A screenshot of the Python IDLE window showing a script for a binary search game. The window title is '2.py - /home/loT\_user/Documents/2.py (3.6.5)'. The menu bar is the same as the previous window. The code is as follows:

```
x=input("Enter an integer between 0 and 1024 -- ")
x=int(x)
a=0
b=1024
test=True
if x == 0:
    print("Your bumber is 0, thank you for playing.")
    test=False
else:
    if x == 1024:
        print("Your number is 1024, thank you for playing.")
        test=False
    while test == True:
        m=int((a+b)/2)
        if m == x:
            print("Your number is ", m, ", thank you for playing.")
            break
        else:
            if m < x:
                a=m
            else:
                b=m
```

- d. Haga clic en **File > Save** (Archivo > Guardar) y guarde el script actual como 2.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.



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024
Your number is 63 , thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 --0
Your number is 0, thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 -- 1024
Your number is 1024, thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 -- 72
Your number is 72 , thank you for playing.
>>>
===== RESTART: /home/IoT_user/Documents/2.py =====
Enter an integer between 0 and 1024 -- 12
Your number is 12 , thank you for playing.
>>>
```

- f. Si se produce un error durante la evaluación de la sintaxis del código, solucione el problema.

### Reflexión

¿Cómo detectar si el jugador ingresa un número fuera del rango de 0 a 1024?

---

---

Se debe verificar que la entrada esté dentro del rango correcto inmediatamente después de que el reproductor ingrese la cantidad.

¿Cómo detectar si el jugador ingresa un número con decimales?

---

---

Se debe verificar que la entrada sea un número entero inmediatamente después de que el reproductor ingrese la cantidad.