



ISIS-1221

INTRODUCCIÓN A LA PROGRAMACIÓN

Nivel 1 - Laboratorio 1 Ambientes básicos de trabajo

Objetivos

El objetivo de este taller es que el estudiante se familiarice con varios de los ambientes de trabajo que tendrá que utilizar a lo largo del semestre:

- Brightspace
- Intérprete de Python (REPL): Python (ipython)
- Spyder

Actividad 0: Familiarización con sistemas operativos

Un **sistema operativo** es un programa cuya función principal es actuar como el intermediador entre software (programas como Word, Spyder, etc.) y el hardware (procesador, memoria, etc.) de un computador. Todos los programas que hacemos se comunican con el sistema operativo, y el sistema operativo es el que se encarga de convertir esos programas en instrucciones que el procesador y los demás componentes (físicos) pueden interpretar.

Existen varias “familias” muy populares de sistemas operativos, y entre las más importantes están Microsoft Windows y los sistemas basados en Unix. Entre estos últimos se encuentran por ejemplo todas las variantes de Linux y OS X. Note que todas estas nociones pueden ser extrapoladas a todo tipo de computadores, en el caso de los computadores móviles (smartphones) algunos sistemas operativos son IOS y Android.

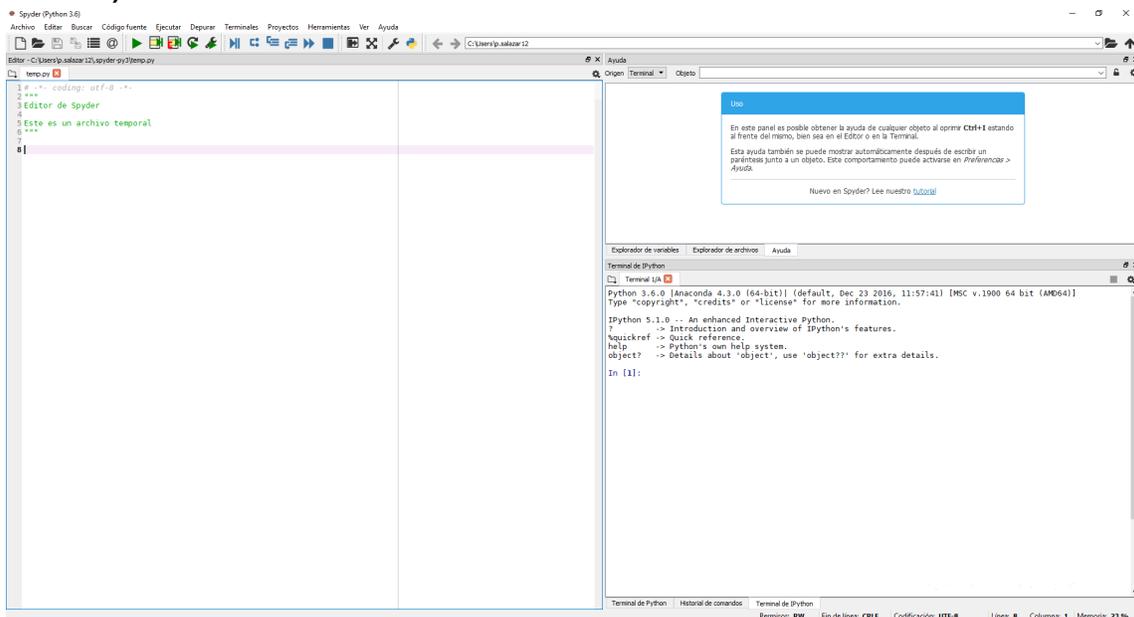
Hoy en día la mayoría de la interacción que tienen los usuarios con el sistema operativo se hace a través de interfaces gráficas (ventanas, diálogos, botones) usando un ratón. Sin embargo, también es posible interactuar con el sistema operativo a través de un programa especial llamado la *línea de comandos* o *el terminal*. Más adelante en el curso usaremos la línea de comandos bien sea de Windows (se ejecuta con el comando ‘cmd’) de OS X (se ejecuta con el comando ‘terminal’) o de Linux (está siempre disponible).

Actividad 1: Interactuar con el intérprete de Python en Spyder

El **intérprete de Python** es un programa que se encarga de convertir código de Python (línea por línea) en código que el sistema operativo puede entender y ejecutar. Por medio de este intérprete usted puede escribir instrucciones de Python y ver inmediatamente el resultado de su evaluación. El intérprete de Python también es llamado un **REPL** por las acciones que hace todo el tiempo: Read (lee una instrucción que escribió el usuario), Evaluate (evalúa la expresión para calcular un resultado), Print (imprime en la pantalla el resultado de la instrucción), y Loop (repite todo el ciclo).

En esta actividad vamos a interactuar con el intérprete de Python pero no lo vamos a ejecutar directamente. En lugar de eso vamos a usar un ambiente llamado **Spyder** que es un IDE (Integrated Development Environment – Ambiente Integrado de Desarrollo) que nos permite acceder directamente al intérprete.

1. Ejecute el programa llamado **“Spyder”**. También puede abrir el programa que se llama **“Anaconda”** y desde allí lanzar **Spyder** (puede demorarse un poco en abrir).



2. El intérprete de Python es la ventana blanca con el título **“Terminal de IPython”** (en la esquina inferior derecha de Spyder) que se muestra a continuación².

¹ Si le interesa leer más sobre las diferentes maneras en que los lenguajes de programación interactúan con los sistemas operativos le recomendamos la siguiente lectura: <https://www.geeksforgeeks.org/compiler-vs-interpreter-2/>

² Los nombres **“Terminal de Python”** e **“Intérprete de Python”** son equivalentes.

Juegue con el intérprete de Python, usando varias de las operaciones matemáticas del lenguaje.

```

Terminal de IPython
Terminal 1/A
Python 3.6.0 |Anaconda 4.3.0 (64-bit)| (default, Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.

In [1]:
  
```

Operadores matemáticos en Python			
Operación	Descripción	Ejemplo	Resultado
$x + y$	Suma dos valores	$3 + 5$	8
$x - y$	Resta un valor de otro	$3 - 5$	-2
$x * y$	Multiplica dos valores	$3 * 5$	15
x / y	Cociente de la división de dos valores	$13 / 5$	2.6
$x // y$	Piso del cociente de la división de dos valores	$13 // 5$	2
$x \% y$	Residuo de la división entre dos valores.	$13 \% 5$	3
$-x$	Cambio de signo	$-(-3)$	3
$x ** y$	Eleva x a la potencia y	$3 ** 5$	243

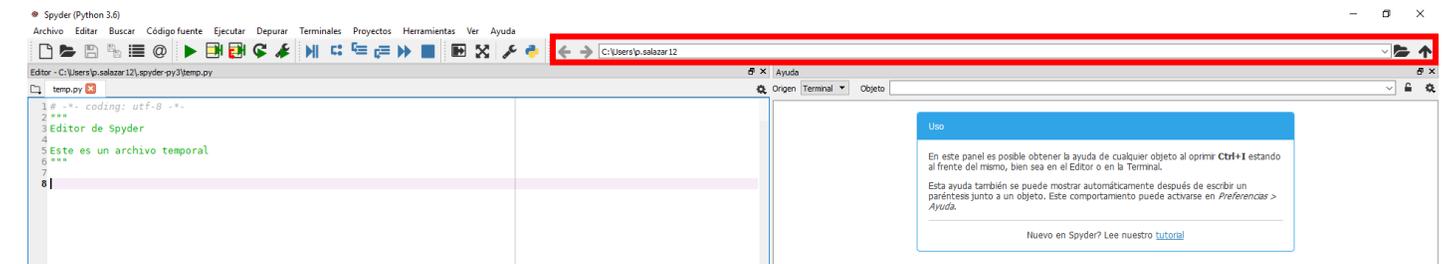
Notas: $x + y * z$ es igual a $x + (y * z)$
 Para valores positivos, $x // y$ equivale a la división entera. Para valores negativos se debe considerar la función piso³.

³ La función piso(x) tiene como resultado el mayor número entero menor o igual a x. Por ejemplo, piso(1.2)=1, piso(2)=2 y piso(-3.5)=-4

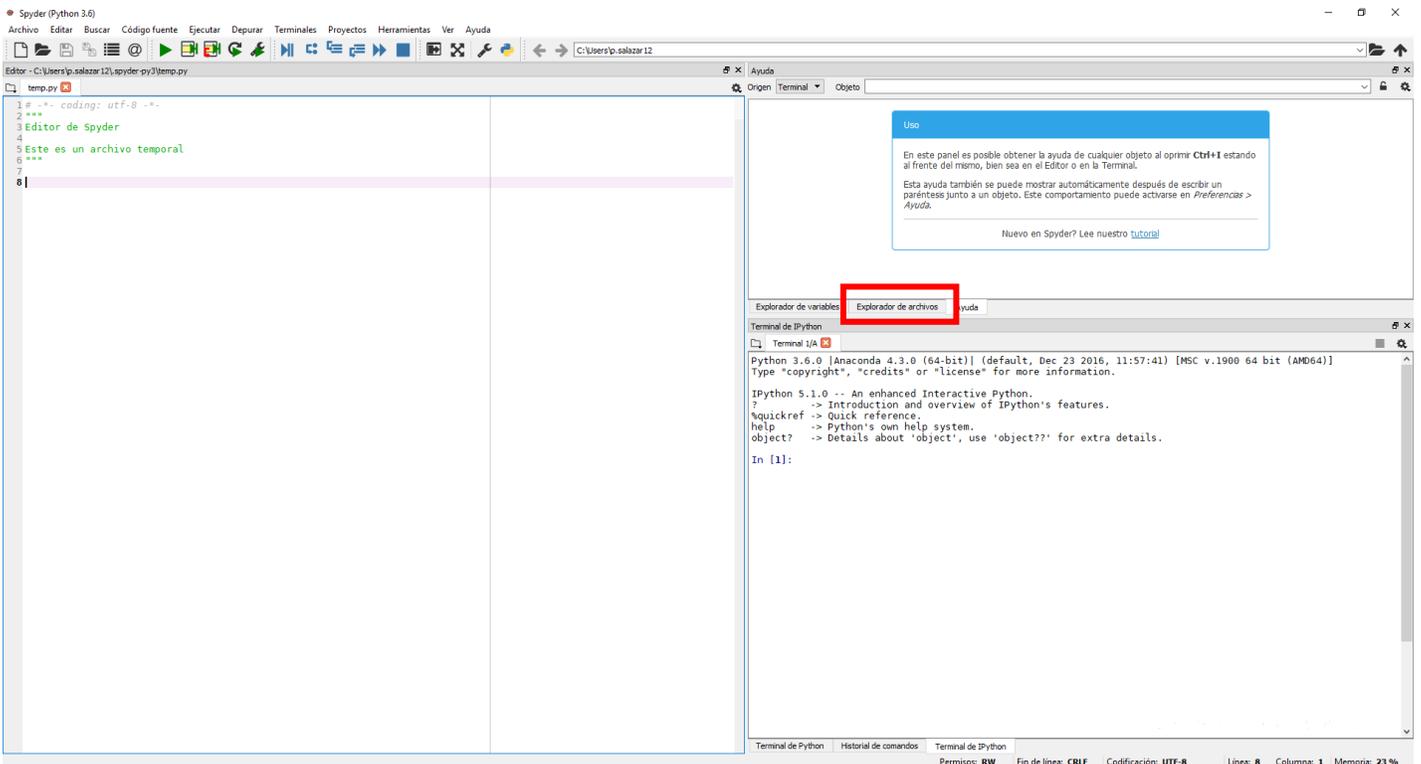
3. Pruebe la función **print** para imprimir algún mensaje en la consola. Imprima mensajes que mezclen texto con el resultado de operaciones matemáticas. Por ejemplo: `print("La suma de 2 y 3 es" + str(2+3))`

Actividad 2: Escribir un programa usando Spyder

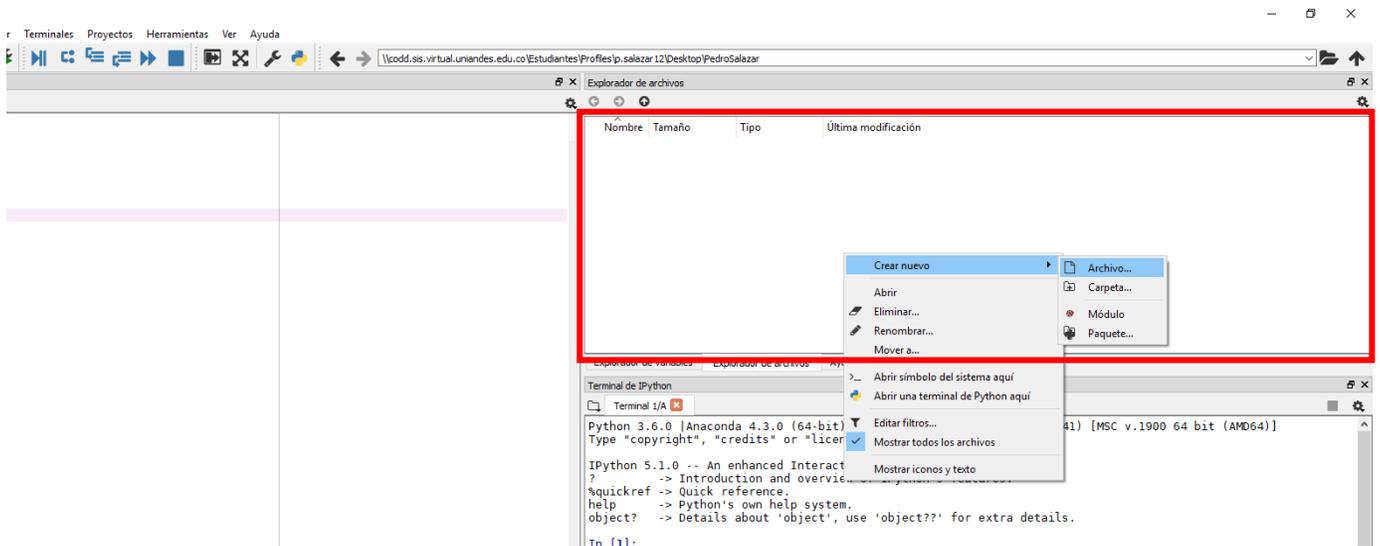
4. Cree una carpeta en el escritorio usando su nombre (sin espacios).
5. En la parte superior de **Spyder** hay un espacio para escoger la carpeta de trabajo. Use el botón que se encuentra a su derecha (el que tiene una carpeta como ícono) para seleccionar la carpeta que usted creó con su nombre en el paso anterior.



6. Identifique la vista que se llama “Explorador de archivos” (File Explorer en inglés) y ábrala haciendo click en su nombre.



7. Haga **click derecho** en la vista “Explorador de archivos” y seleccione la opción “Crear nuevo, archivo” (New File en inglés). Seleccione el nombre “actividad2.py” para el nuevo archivo.



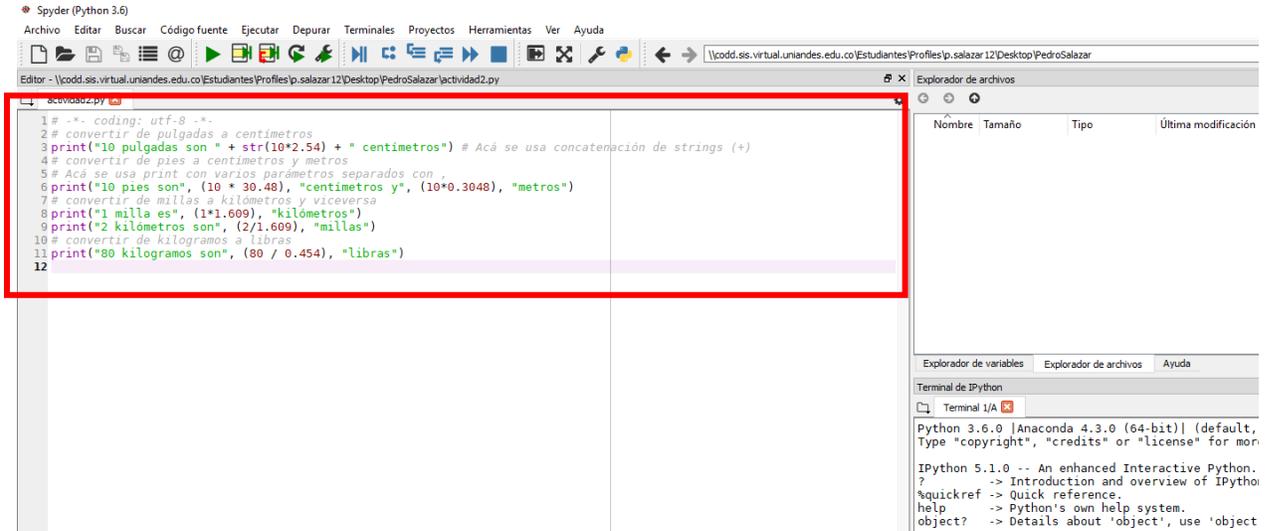
8. Copie en el nuevo archivo las siguientes instrucciones:

```
# convertir de pulgadas a centímetros  
print("10 pulgadas son " + str(10*2.54) + " centímetros") # Acá se usa concatenación de strings (+)
```

```
# convertir de pies a centímetros y metros  
# Acá se usa print con varios parámetros separados con ,  
print("10 pies son", (10 * 30.48), "centímetros y", (10*0.3048), "metros")
```

```
# convertir de millas a kilómetros y viceversa  
print("1 milla es", (1*1.609), "kilómetros")  
print("2 kilómetros son", (2/1.609), "millas")
```

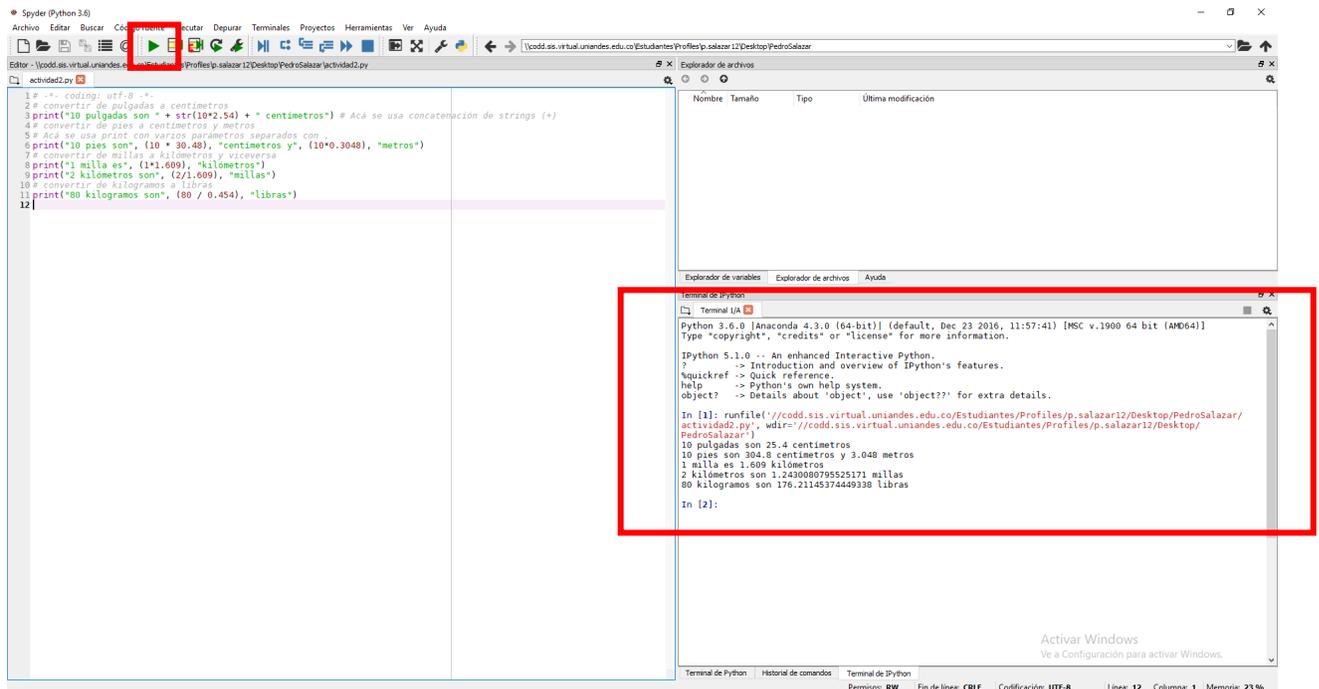
```
# convertir de kilogramos a libras  
print("80 kilogramos son", (80 / 0.454), "libras")
```



9. Ejecute el programa “actividad2.py” que está construyendo. Para eso asegúrese de que el cursor se encuentra en la ventana con el programa y presione la tecla F5. También puede hacer clic en el botón para ejecutar que se encuentra en la parte superior de la ventana.

Los resultados de la ejecución aparecerán en la ventana con la terminal de Python: revise que los resultados obtenidos sean los correctos y que no aparezcan mensajes de error.

Si encuentra algún error, corrijalo en su programa y vuelva a ejecutarlo.



10. Usted también puede ir probando línea por línea su programa usando la tecla F9 en **Spyder**: esto ejecutará la instrucción en la que esté ubicado el cursor en ese momento.
11. Modifique ahora su programa “actividad2.py” para que las conversiones ya no sean de 10 pulgadas sino 100 y no sean 2 kilómetros sino 20.
12. Entregue el archivo modificado actividad2.py a través de Brightspace en **la tarea designada como L1: Ambientes básicos de trabajo**.