



SKILLING  
CENTER

**TECMILENIO**



# Visualización y Programación en Python

Introducción a la programación  
con Python





Ernesto es un profesional financiero que busca crecimiento, comenzando a buscar opciones descubre que los profesionales científicos de datos analizan grandes cantidades de datos y son de las profesiones mejor pagadas en todo el mundo.

En este tema, Ernesto descubrirá que la combinación de datos y el lenguaje de programación Python es una de las combinaciones más valiosas en la actualidad. La ciencia de datos se encuentra en prácticamente todas las áreas de conocimiento, especialmente en las finanzas.

En esta experiencia académica aprenderás las bases del lenguaje Python y cómo encontrar soluciones informáticas a problemas cotidianos a través de algoritmos.



- Python es un lenguaje de programación creado a principios de la década de los noventa.
- Es más viejo que el lenguaje Java y más joven que el C.
- La sintaxis es muy limpia y favorece la legibilidad del código.
- Actualmente, Python es de los lenguajes más utilizados principalmente en los siguientes entornos:

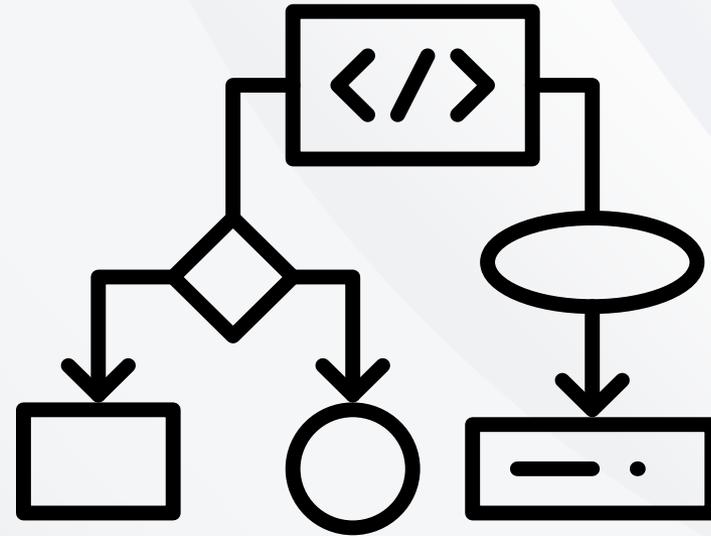




Python es un software sin costo y se puede instalar desde la página oficial de Python.org.

Existen dos formas de ejecutar Python:

- Línea de comando. Se ejecuta línea por línea.
- Entornos interactivos. Se crean líneas de comando y se ejecutan en conjunto. Ofrece diversos atributos que facilitan el desarrollo del código, como el análisis de la sintaxis, autocompletado de código, identificación de comandos de código por colores, depuración, controlar versiones, entre otras.

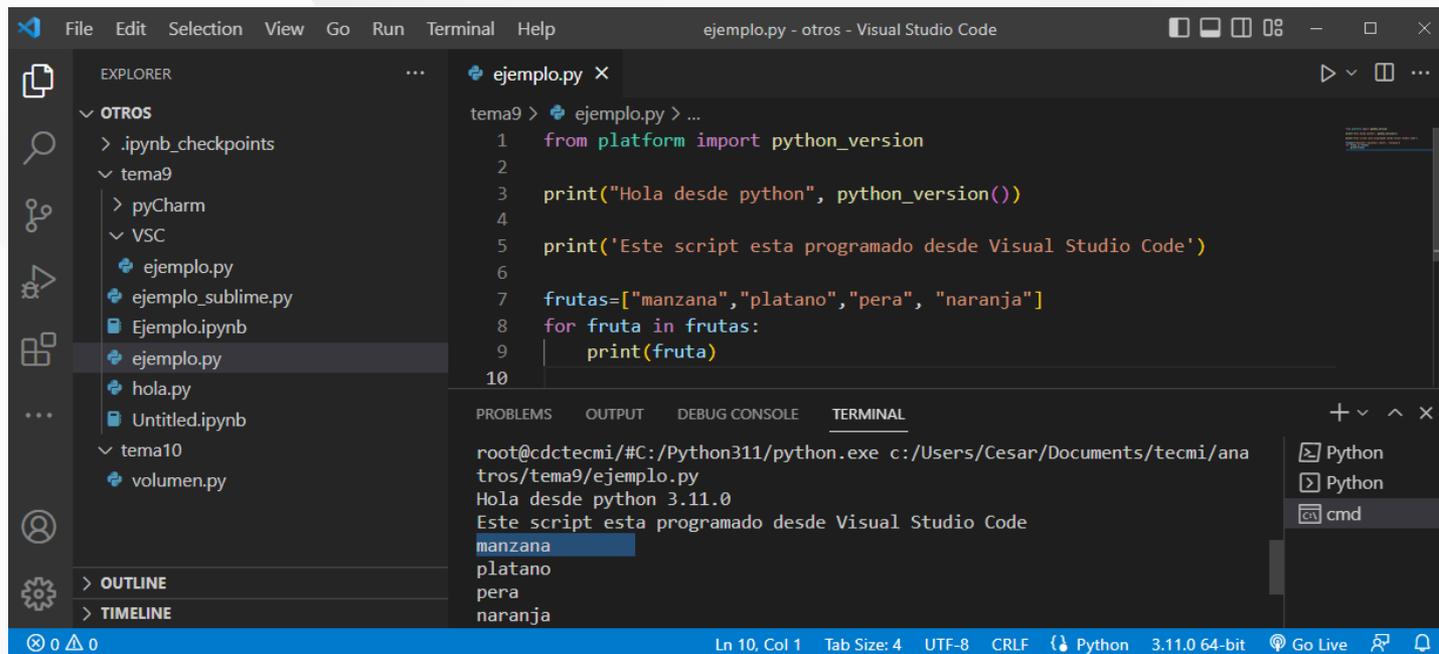




Una vez concluido el código, se puede guardar con extensión `.py` como un conjunto de líneas de comando, o con extensión `.ipynb` como un cuaderno de trabajo con código y texto.

Según Heineman (2021), un algoritmo es un método de resolución de problemas paso a paso implementado como un programa de computadora que devuelve un resultado correcto en una cantidad de tiempo predecible.

Una opción recomendada para desarrollar código Python y ejecutarlo es Visual Studio Code de Microsoft. Cuenta con utilerías como el enlace con repositorios como Gtithub e instalaciones de extensiones o plugins que se obtienen dentro del mismo editor.



The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with folders like 'OTROS' and 'tema9', and files such as 'ejemplo.py'. The main editor window shows the content of 'ejemplo.py' with the following Python code:

```
1 from platform import python_version
2
3 print("Hola desde python", python_version())
4
5 print('Este script esta programado desde Visual Studio Code')
6
7 frutas=["manzana","platano","pera", "naranja"]
8 for fruta in frutas:
9     print(fruta)
10
```

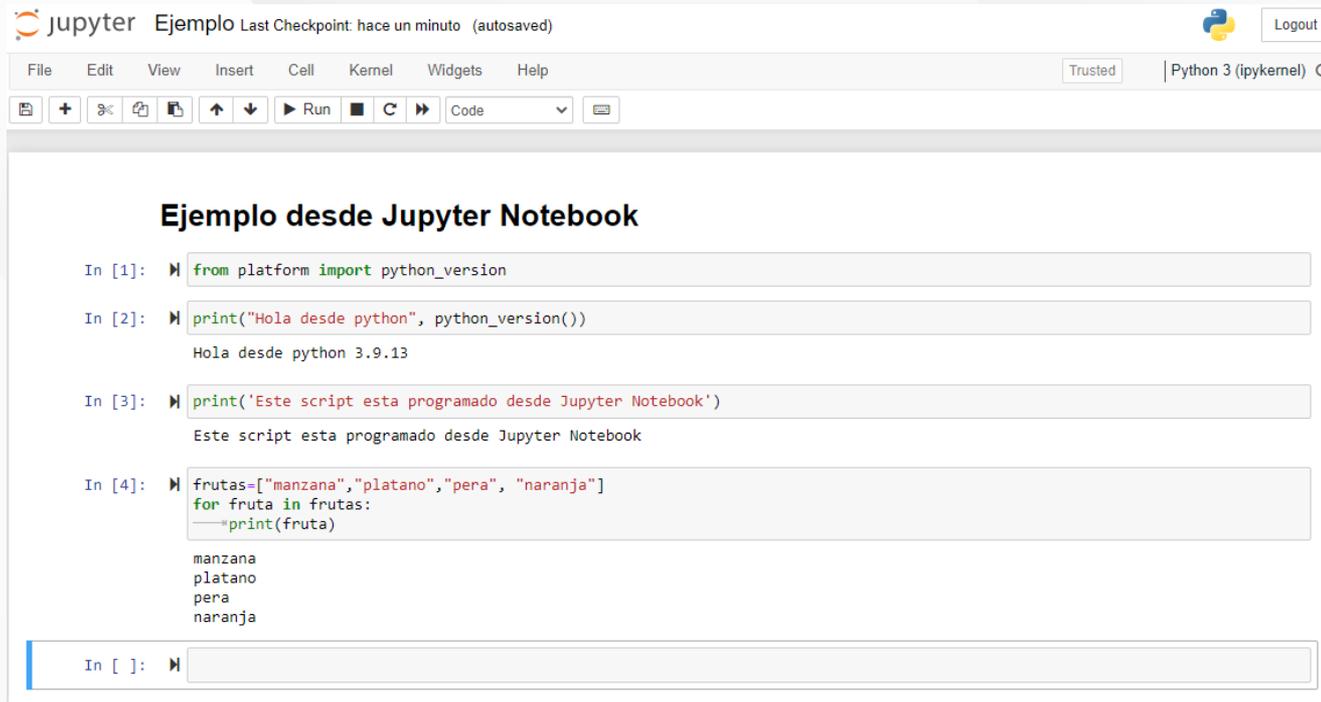
Below the editor, the Terminal panel shows the execution output:

```
root@cdctecmi/#C:/Python311/python.exe c:/Users/Cesar/Documents/tecmi/ana
tros/tema9/ejemplo.py
Hola desde python 3.11.0
Este script esta programado desde Visual Studio Code
manzana
platano
pera
naranja
```

The status bar at the bottom indicates the current file is 'ejemplo.py' at line 10, column 1, using UTF-8 encoding and the Python 3.11.0 64-bit extension.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Otra opción es la plataforma Jupyter Notebook, que permite la creación de cuadernos que contienen celdas con código y celdas con texto y/o imágenes para explicar lo que se está desarrollando, facilitando la lectura e interpretación del código.



The screenshot shows a Jupyter Notebook interface with the following content:

```
Jupyter Ejemplo Last Checkpoint: hace un minuto (autosaved) Python 3 (ipykernel) Logout
```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

### Ejemplo desde Jupyter Notebook

```
In [1]: from platform import python_version
```

```
In [2]: print("Hola desde python", python_version())
```

Hola desde python 3.9.13

```
In [3]: print('Este script esta programado desde Jupyter Notebook')
```

Este script esta programado desde Jupyter Notebook

```
In [4]: frutas=["manzana","platano","pera", "naranja"]
for fruta in frutas:
    print(fruta)
```

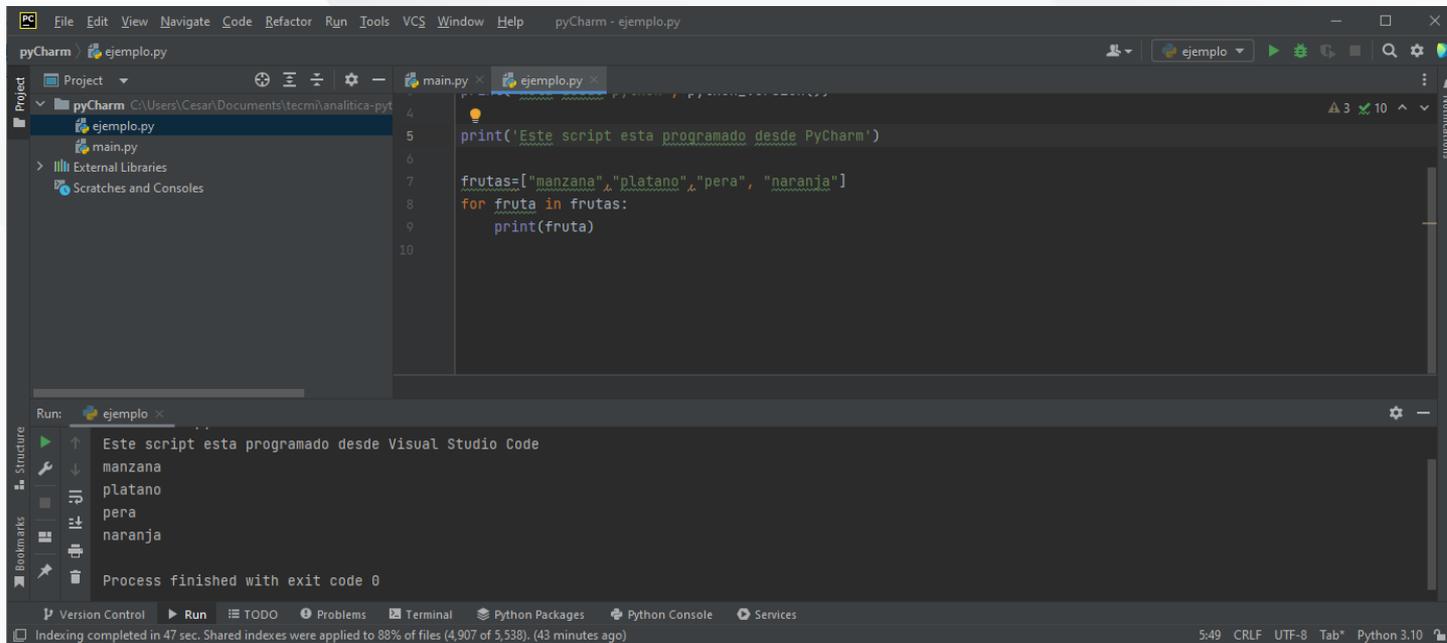
manzana  
platano  
pera  
naranja

```
In [ ]:
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Adicionalmente, se tiene la opción de la plataforma PyCharm.

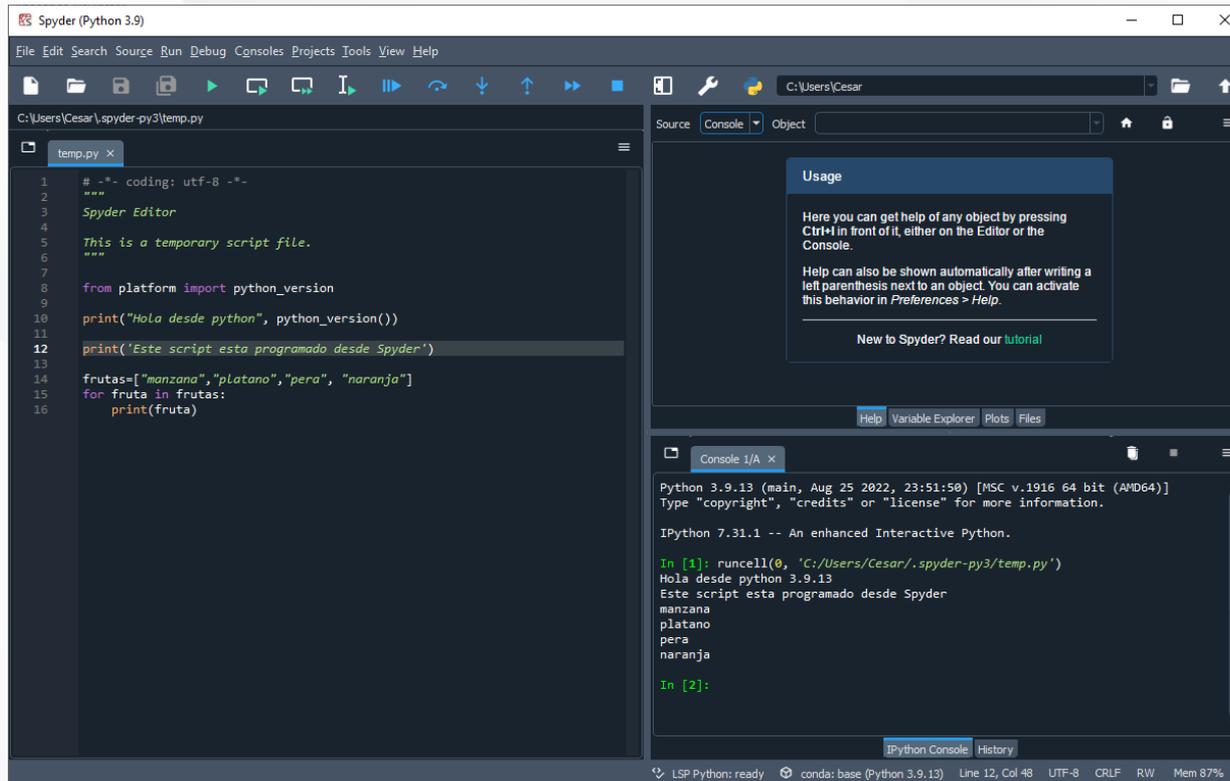
Tiene la versión profesional que es de pago. Ofrece Python, HTML, CSS, JS y SQL, además la versión community para el desarrollo de Python puro.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pyCharm - ejemplo.py
pyCharm ejemplo
Project
  pyCharm C:\Users\Cesar\Documents\tecm\analitica-pyt
  ejemplo.py
  main.py
  External Libraries
  Scratches and Consoles
Run: ejemplo
Este script esta programado desde Visual Studio Code
manzana
platano
pera
naranja
Process finished with exit code 0
Version Control Run TODO Problems Terminal Python Packages Python Console Services
Indexing completed in 47 sec. Shared indexes were applied to 88% of files (4,907 of 5,538). (43 minutes ago) 5:49 CRLF UTF-8 Tab* Python 3.10
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Spyder, que está incluido en Anaconda Navigator, es un excelente compilador para Python. Cuenta con un nivel avanzado de funcionalidades para edición, depuración y explotación de los datos, exploración de variables, además de varios complementos.



The screenshot displays the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.9)". The top menu bar includes "File", "Edit", "Search", "Source", "Run", "Debug", "Consoles", "Projects", "Tools", "View", and "Help". The toolbar contains icons for file operations, running, and debugging. The left pane shows a file explorer with a folder named "temp.py" and a file named "temp.py". The code editor displays the following Python code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4 This is a temporary script file.
5 """
6
7
8 from platform import python_version
9
10 print("Hola desde python", python_version())
11
12 print('Este script esta programado desde Spyder')
13
14 frutas=["manzana","platano","pera", "naranja"]
15 for fruta in frutas:
16     print(fruta)
```

The right pane is divided into two sections. The top section is titled "Usage" and contains the following text:

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our [tutorial](#)

The bottom section is titled "Console 1/A" and contains the following output:

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

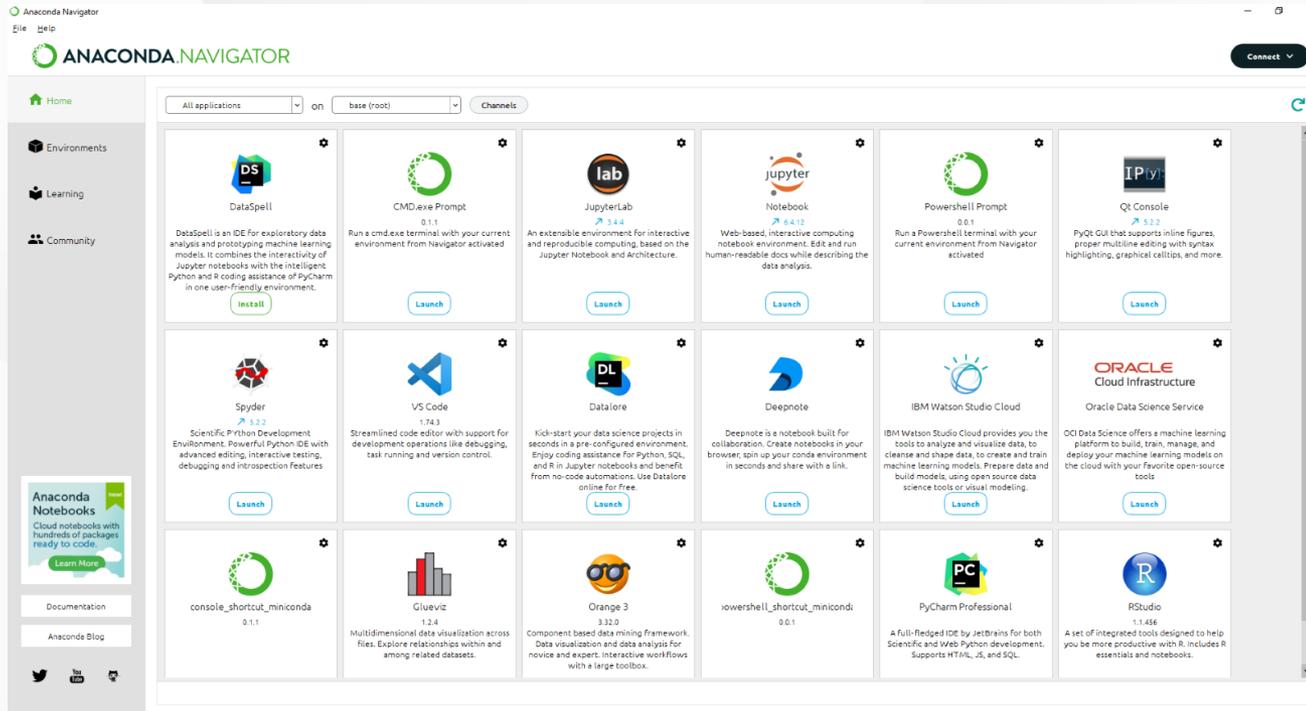
In [1]: runcell(0, 'C:/Users/Cesar/.spyder-py3/temp.py')
Hola desde python 3.9.13
Este script esta programado desde Spyder
manzana
platano
pera
naranja

In [2]:
```

The bottom status bar shows "LSP Python: ready", "conda: base (Python 3.9.13)", "Line 12, Col 48", "UTF-8", "CRLF", "RW", and "Mem 87%".

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Otros entornos interactivos incluidos en Anaconda Navigator:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

## Fundamentos de algoritmos Python

- Tipos de datos:
  - Boleanos: True, False, and, or, not.
  - Números enteros, punto flotante y complejos.
  - Cadenas de texto.
- Te permite definir variables con nombre.
  - Pueden contener caracteres como letras del abecedario, números del 0 al 9 y guion bajo.
  - Distingue entre mayúsculas y minúsculas.
  - Deben comenzar con letra o guion bajo, no con número.
  - No se pueden usar palabras reservadas de Python, como True, False, while, return, entre otros.

Operador	Descripción	Ejemplo
==	¿a y b son iguales?	r = 5 == 3 # r es False
!=	¿a y b son distintos?	r = 5 != 3 # r es True
<	¿a es menor que b?	r = 5 < 3 # r es False
>	¿a es mayor que b?	r = 5 > 3 # r es True
<=	¿a es menor o igual que b?	r = 5 <= 5 # r es True
>=	¿a es mayor o igual que b?	r = 5 >= 3 # r es True



Los operadores aritméticos en Python son:

Operador	Descripción	Ejemplo
+	Suma	<code>r = 3 + 2</code> <code># r es 5</code>
-	Resta	<code>r = 4 - 7</code> <code># r es -3</code>
*	Multiplicación	<code>r = 2 * 6</code> <code># r es 12</code>
**	Exponente	<code>r = 2 ** 6</code> <code># r es 64</code>
/	División	<code>r = 3.5 / 2</code> <code># r es 1.75</code>
//	División entera	<code>r = 3.5 // 2</code> <code># r es 1.0</code>
%	Módulo	<code>r = 7 % 2</code> <code># r es 1</code>



- Las cadenas de texto se pueden manejar indistintamente con comillas simples ('cadena') o comillas dobles ("cadena").
- Dentro de estas cadenas se utilizan caracteres especiales con la diagonal invertida (\), tales como `\n` para nueva línea en la impresión, `\t` para tabulación.
- Una cadena precedida del carácter `u` o `r`, significa que se trata de una cadena que utiliza codificación unicode, por ejemplo:
  - `Unicode = u"ääè"`
  - `raw = r"\n"`
- Se usan comillas triples `"""` para explicar o comentar el código.
- Para concatenar cadenas se pueden usar los operadores `+` y `*`, por ejemplo:
  - **`a = "uno"`**
  - **`b = "dos"`**
  - **`c = a + b # c es la nueva cadena "unodos"`**
  - **`d = a * 3 # d es la nueva cadena "unounouno"`**
- Como recomendación, el primer paso antes de elaborar el código de un algoritmo es hacer el diagrama de flujo con la lógica que seguirá el proceso.



Para reforzar tu aprendizaje del tema, contesta las siguientes preguntas:

1. Menciona tres ventajas de utilizar entornos interactivos en Python en lugar de línea por línea.
2. Menciona tres cualidades de las siguientes plataformas para editar código Python:
  - a) Visual Studio Code
  - b) Jupyter Notebook
  - c) Pycharm
  - d) Spyder
3. Desarrolla el algoritmo en Python para calcular el valor presente de una inversión que promete recibir la cantidad de \$15,000 en 3 años y una tasa de interés anual del 9 %.



Con el aprendizaje de este tema, Ernesto descubre las bases de la programación en Python y con esto, comienza a generar propuestas para resolver problemas, así como explorar áreas de oportunidad en su especialidad, que son las finanzas.

Adicionalmente Ernesto percibe que la ciencia de datos apoyada con Python le incrementará sus habilidades para facilitar la toma de decisiones en finanzas y en muchas otras funciones de la empresa.

Esta experiencia le brindará a Ernesto y a su organización, el crecimiento profesional que busca.



- Heineman, G. (2021). *Learning Algorithms: A Programmer's Guide To Writing Better Code*. Estados Unidos: O'Reilly Media.

# Visualización y Programación en Python

Condicionales





```
25
26 def check_db():
27     if not os.path.isfile(FILE_URI):
28         db.create_all()
29
30 @app.route("/")
31 def home():
32     check_db()
33     all_books = db.session.query(Book).all()
34     return render_template("index.html", books=all_books)
35
36 @app.route("/edit", methods=["GET", "POST"])
37 def edit():
38
39     if request.method == 'POST':
40         book_id = request.form["id"]
41         book_to_update = Book.query.get(book_id)
42         book_to_update.rating = request.form["rating"]
43         db.session.commit()
44         return redirect(url_for("home"))
```

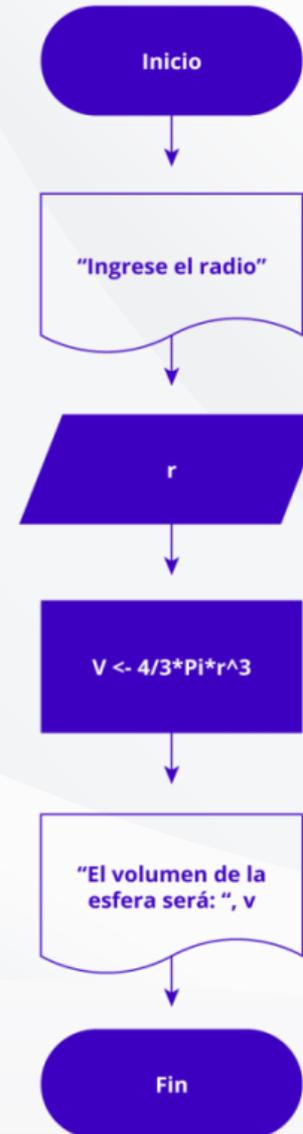
Considera que quieres ingresar a un concurso para desarrollar aplicaciones para Big Data en el Tecmilenio, para lo cual investigas y descubres que el lenguaje Python incrementa tus oportunidades para desarrollar los algoritmos que tendrá la aplicación que tienes en mente.

En este tema darás un paso adelante para dominar el lenguaje Python con las estructuras de condicionantes para controlar una gran variedad de procesos analíticos.

Para comprender mejor el uso de condicionantes, conviene plantear el diagrama de flujo de los cálculos que se van a realizar. Considera que quieres calcular el volumen de una esfera a partir de la longitud del radio.

En primer lugar, planteas el flujo de acciones para determinar el resultado:

- Variables:  $v$ =volumen y  $r$ =radio
- Constantes:  $\pi=3.1416$
- Flujo:
  - Inicio: Ingresar el radio
  - Calcular el volumen  $v = \frac{4\pi r^3}{3}$
  - Imprimir el resultado: "El volume de la esfera es: ",  $v$
- Fin





El código en Python para el cálculo del volumen de la esfera puede ser el siguiente:

```
# -*- coding: utf-8 -*-
#Decoración: Nombre del Algoritmo
print("-----")
print("Complemento1: CALCULAR VOLUMEN DE LA ESFERA.")
print("-----")
#Constantes
PI = 3.1416
#Entradas
r = float( input("Ingrese Radio: "))
#Proceso
v = 4/3 * PI * r**3
#Salida
print("\nSALIDA: ")
print("-----")
print("El volumen de la esfera es:", v)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



En la mayoría de los algoritmos se requiere tomar decisiones durante el proceso de cálculo, indicando la ruta a partir en un punto determinado, para lo cual se utilizan sentencias condicionantes. González (s.f.) establece que las sentencias condicionantes o estructuras selectivas comprueban condiciones para establecer el comportamiento de los algoritmos dependiendo de que se cumpla la condición.

Tipos de condicionantes:

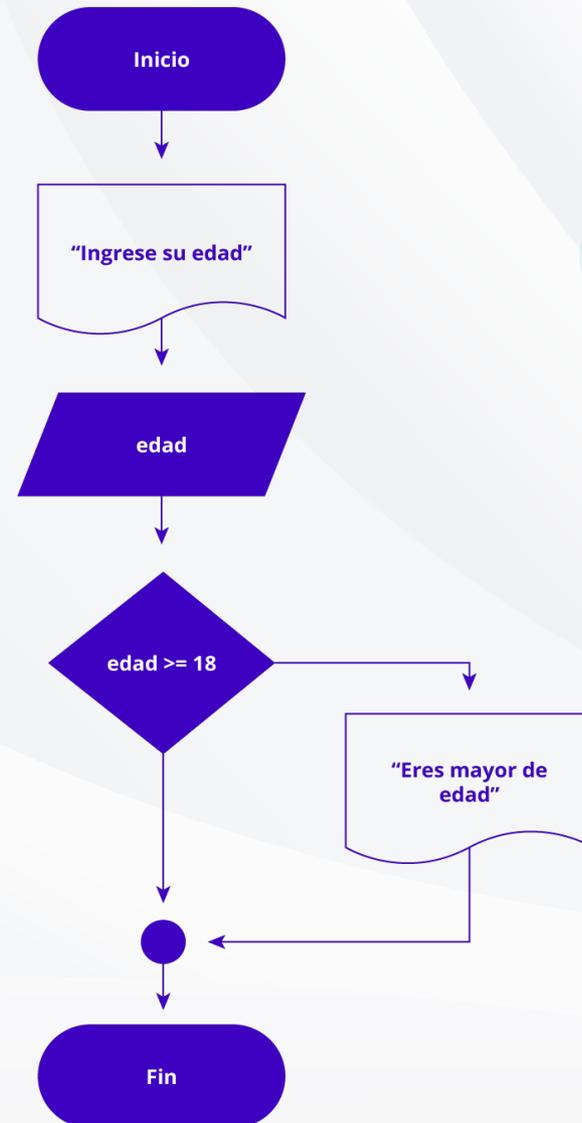
1. Relacionales (igual =, mayor > , menor <)
2. Operadores lógicos (and, or, not)
3. De tipo booleano (True o False)



Las sentencias condicionales utilizan palabras reservadas en Python:

- "if", condicionante simple

Cuando es verdadero, ejecuta las acciones consideradas en la estructura, de lo contrario no hace nada y continúa con el resto del código.





Ejemplo: Determinar si una persona es mayor de 18 años, en su caso, mostrar “Eres mayor de edad”.

En Python, se puede escribir el problema con el siguiente código:

```
# -*- coding: utf-8 -*-  
#Decoración: Nombre del Algoritmo  
print("-----")  
print("EVALUAR SI ERES MAYOR DE EDAD.")  
print("-----")  
#Entradas  
print("Ingrese su edad: ")  
x = int( input())  
#Proceso  
if x >= 18 :  
    print("Eres mayor de edad")
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Hay que tomar en cuenta que una de las reglas de Python es la indentación, motivo por el cual el último renglón de código anterior tiene espacio en blanco antes del print, porque es parte del condicionante if (El Libro de Python, s.f.).

- Doble (if ... else)

Permiten el control de acciones, primero se evalúa si la condición es verdadera o falsa y luego se ejecutan las acciones en consecuencia.

Por ejemplo, dados 2 números, determina e imprime el mayor de ellos.

```
# -*- coding: utf-8 -*-
#Decoración: Nombre del Algoritmo
print("-----")
print("Ejemplo2: IMPRIMIR EL MAYOR DE DOS NÚMEROS.")
print("-----")
#Entradas
print("Ingrese 2 números: ")
num1 = int( input("Primer Número: "))
num2 = int( input("Segundo Número: "))
#Salida
print("\nSALIDA: ")
print("-----")
if num1 > num2 :
#Esto esta dentro del IF
    print("El mayor es:", num1)
else:
#Esto esta dentro del ELSE
    print("El mayor es:", num2)
```

```
-----
Ejemplo2: IMPRIMIR EL MAYOR DE DOS NÚMEROS.
-----
Ingrese 2 números:
Primer Número: 12
Segundo Número: 24

SALIDA:
-----
El mayor es: 24
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

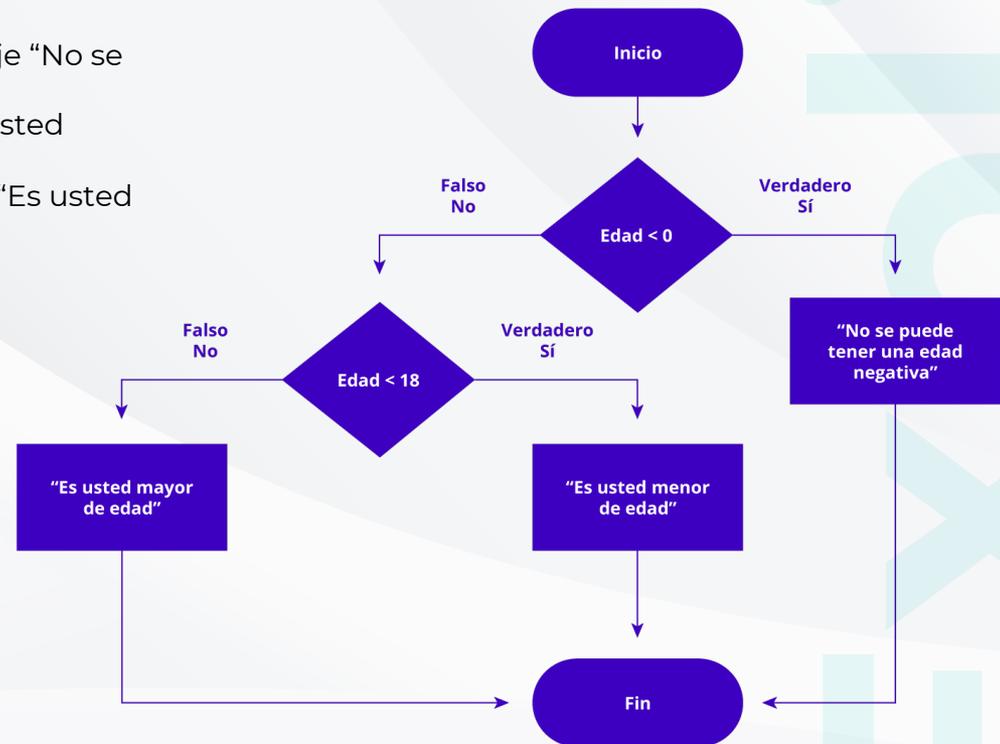
- Múltiple o anidado (if ... elif ... elif ... else)

Cuando existan dos o más posibles resultados en la condición, se utiliza *if* para la primera, y en las siguientes se utiliza *elif* y *else* para la última que se ejecuta cuando las anteriores no se cumplen.

Por ejemplo, dada la edad de una persona, evaluar las siguientes condiciones:

- Si el valor es negativo, mostrar el mensaje “No se puede tener una edad negativa”
- Si el valor está entre 0 y 17, mostrar “Es usted menor de edad”
- Si el valor es mayor o igual a 18, mostrar “Es usted mayor de edad”

```
# -*- coding: utf-8 -*-  
#Decoración: Nombre del Algoritmo  
print("-----")  
print("EVALUAR SI ERES MAYOR DE EDAD.")  
print("-----")  
#Entradas  
edad = int(input("Ingrese su edad: "))  
#Proceso  
if edad < 0:  
    print("No se puede tener una edad negativa")  
elif edad < 18:  
    print("Es usted menor de edad")  
else:  
    print("Es usted mayor de edad")
```



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.



Ejecutando el código anterior, con valores de 12, -12 y 45 años, se tienen las siguientes salidas:

```
-----  
EVALUAR SI ERES MAYOR DE EDAD.  
-----
```

```
Ingrese su edad: 12  
Es usted menor de edad  
12
```

```
-----  
EVALUAR SI ERES MAYOR DE EDAD.  
-----
```

```
Ingrese su edad: -12  
No se puede tener una edad negativa  
-12
```

```
-----  
EVALUAR SI ERES MAYOR DE EDAD.  
-----
```

```
Ingrese su edad: 45  
Es usted mayor de edad  
45
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

El uso de condicionales te ayudará a estructurar los programas que desarrolles para que sean más inteligentes y puedan reaccionar dependiendo de los parámetros de entrada, eligiendo por ellos mismos el camino que deben seguir.



Para reforzar tu aprendizaje del tema, contesta las siguientes preguntas:

1. Describe tres situaciones en tu área de trabajo en donde se apliquen condicionantes para algún flujo de trabajo, ya sea en procesos productivos o en procesos administrativos. Muestra cada situación en un diagrama de flujo, explicando los propósitos del condicionante en particular.
2. Elabora el diagrama de flujo y desarrolla el código en Python para clasificar el censo demográfico de los empleados de una organización en cuanto a género y edad como preparación de los datos para la pirámide poblacional. En la variable edad considera cinco bloques: entre 18 y 22 años, entre 23 y 30 años, entre 31 y 40 años, entre 41 y 50 años y mayor de 50 años.

Nota: Incluye el condicional para cuando se registra un número negativo.



En este tema aprendiste los tres tipos de condicionantes para los algoritmos en Python, los cuales se utilizan con palabras reservadas para realizar comparaciones, con lo cual tienes un paso firme hacia el dominio del lenguaje Python.

Como sabes, todo lo que se aprende requiere de práctica para lograr la correcta aplicación en diversas situaciones del mundo real, así como en tu área de trabajo y en toda la organización.





- El Libro de Python. (s.f.). *Test, documentación y consejos*. Recuperado de <https://ellibrodepython.com/test-python>
- González, R. (s.f.). *Python para todos*. Recuperado de <https://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>

*Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, estas se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.*

---

*Todos los derechos reservados @ Universidad Tecmilenio*

*La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor. El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO. Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.*