



Diplomado en Ciencia de Datos con Python

Módulo 1

Profesor: Javier Alberto Pérez Garza

¿Qué es la ciencia de datos?

La ciencia de datos es un campo dedicado a la extracción del conocimiento en los datos mediante el uso de métodos, procesos, algoritmos y sistemas científicos.

Un científico de datos utiliza conocimientos:

- Matemáticos y estadísticos.
- Ciencias de la computación.
- Del área de investigación, industria o negocio.
- Comunicación y visualización.

¿Por qué ciencia de datos?



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

¿?

Mundo de datos

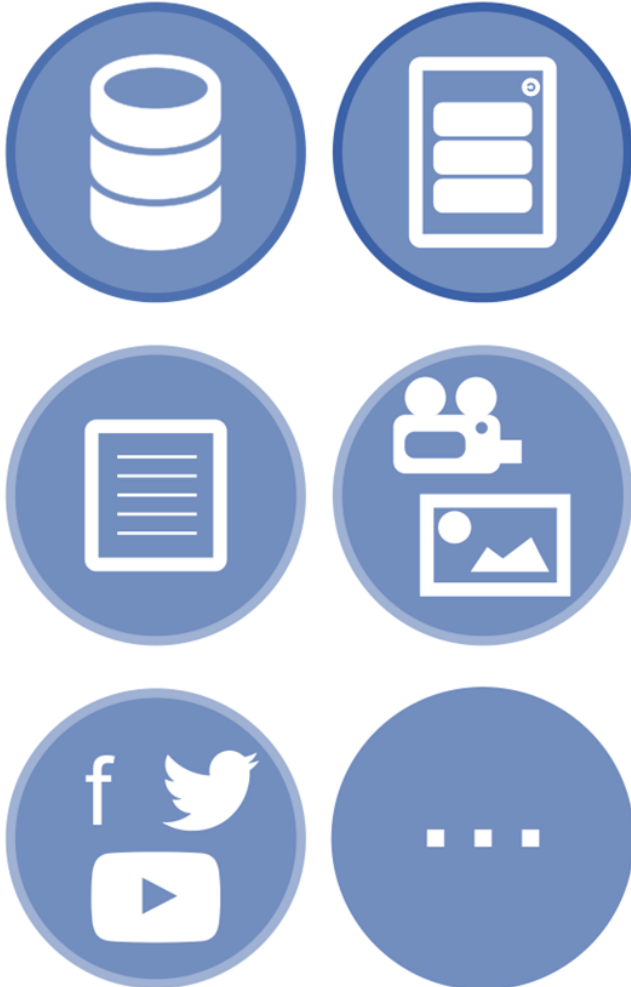


UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Flujo de datos

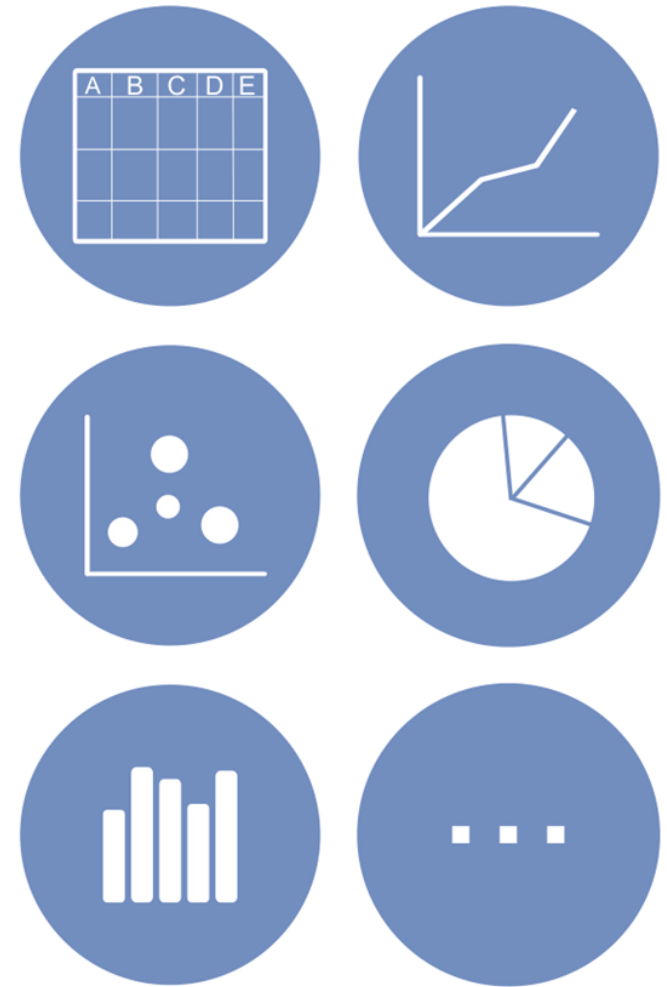
ETL



Análisis



Visualización



Herramientas y tecnologías



¿Qué es Python?

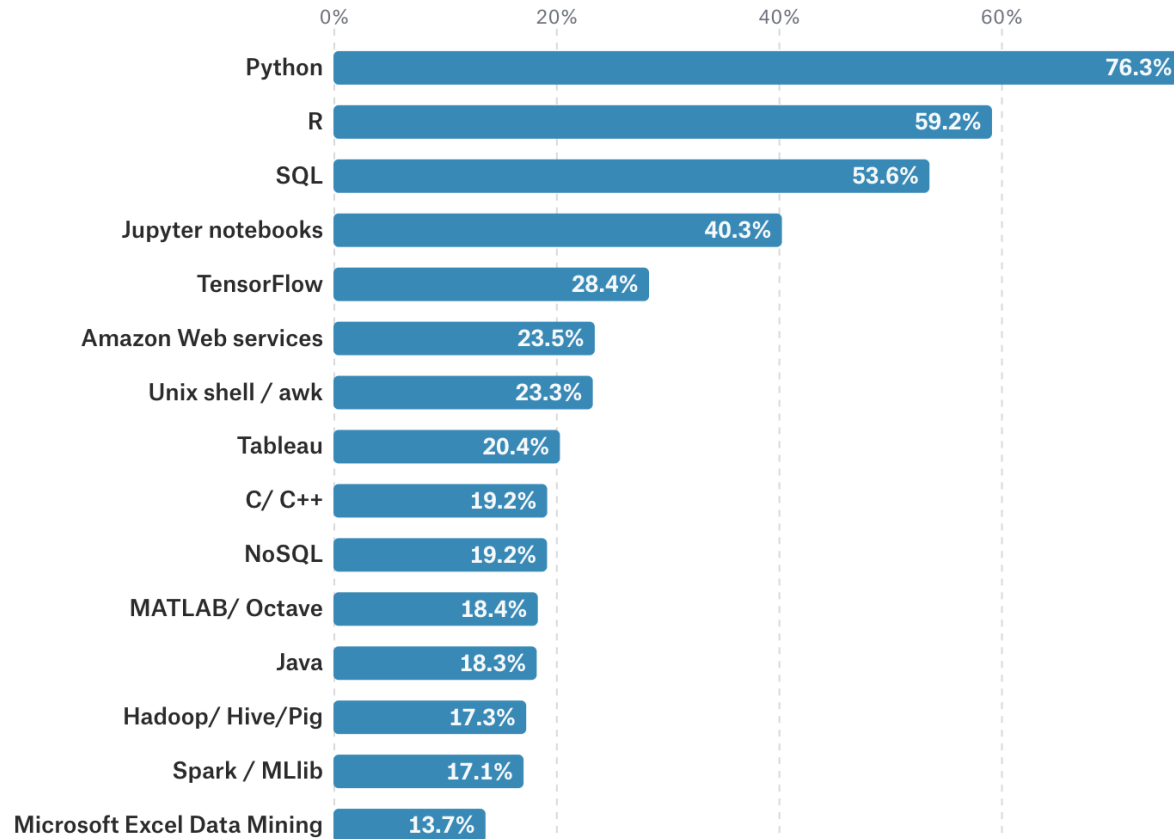
Python es un lenguaje de programación interpretado de alto nivel que permite trabajar rápidamente para una integración de sistemas mucho más eficiente.



Ventajas de Python

- Código fácil de leer.
 - Hello world: `print("Hello, world!")`
- Propósito general (back-end, front-end, análisis de datos, ...).
- Múltiples paradigmas de programación.
 - Imperativo, Orientado a objetos y funcional.
- Código abierto.
- Extensible (extensiones en C y C++).
- Muchas librerías disponibles para todo tipo de problemas.

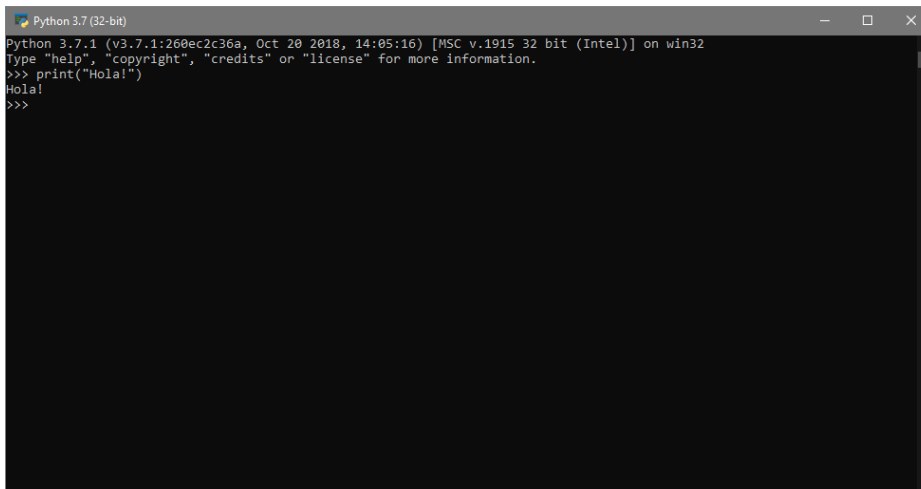
Python en ciencia de datos



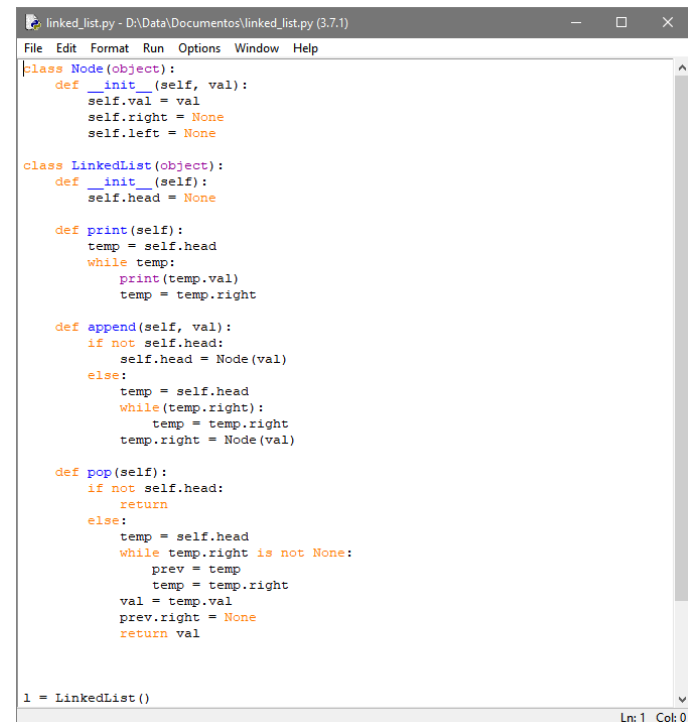
Fuente: towardsdatascience.com
(Gráfica generada usando datos de LinkedIn)

Formas de utilizar Python

- Desde la línea de comandos de manera interactiva.
- Desde un editor de texto para crear programas ejecutables.
- Desde un IDE para crear programas ejecutables.
- Desde notebook para la ejecución interactiva de código.



```
Python 3.7 (32-bit)
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hola!")
Hola!
>>>
```



```
linked_list.py - D:\Data\Documentos\linked_list.py (3.7.1)
File Edit Format Run Options Window Help

class Node(object):
    def __init__(self, val):
        self.val = val
        self.right = None
        self.left = None

class LinkedList(object):
    def __init__(self):
        self.head = None

    def print(self):
        temp = self.head
        while temp:
            print(temp.val)
            temp = temp.right

    def append(self, val):
        if not self.head:
            self.head = Node(val)
        else:
            temp = self.head
            while temp.right:
                temp = temp.right
            temp.right = Node(val)

    def pop(self):
        if not self.head:
            return
        else:
            temp = self.head
            while temp.right is not None:
                prev = temp
                temp = temp.right
            val = temp.val
            prev.right = None
            return val

l = LinkedList()

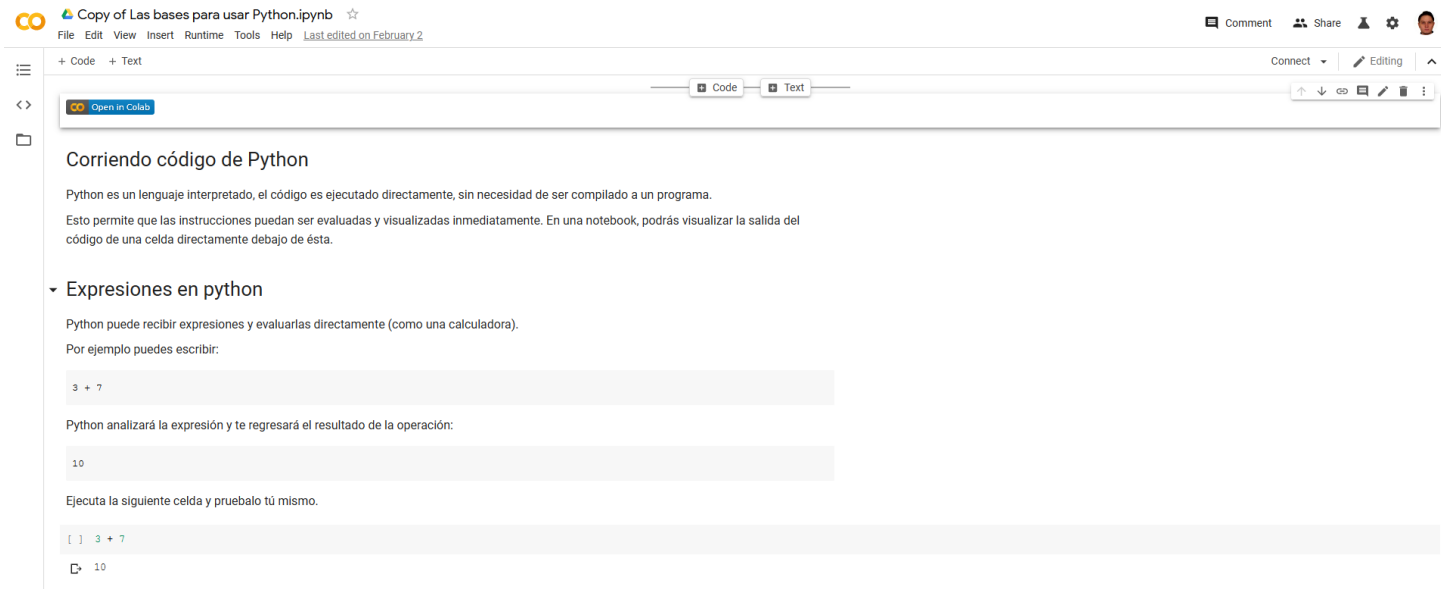
Ln: 1 Col: 0
```

Notebooks

Las notebook son una interfaz de programación que permite integrar procesamiento de texto y la capacidad de ejecutar código del lenguaje de programación.

Las notebooks son particularmente útiles para:

- El análisis de datos, ya que se puede documentar, analizar y visualizar los datos en un mismo entorno.
- Cursos, por la capacidad de combinar texto y código en un mismo entorno.



The screenshot shows a Google Colab notebook interface. At the top, the title bar reads "Copy of Las bases para usar Python.ipynb" with a star icon and a "Last edited on February 2" timestamp. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". On the right side of the title bar are icons for "Comment", "Share", "Settings", and a user profile. The main area of the notebook is divided into two tabs: "Code" and "Text". The "Text" tab is currently selected. The content of the notebook is as follows:

Corriendo código de Python

Python es un lenguaje interpretado, el código es ejecutado directamente, sin necesidad de ser compilado a un programa. Esto permite que las instrucciones puedan ser evaluadas y visualizadas inmediatamente. En una notebook, podrás visualizar la salida del código de una celda directamente debajo de ésta.

▼ **Expresiones en python**

Python puede recibir expresiones y evaluarlas directamente (como una calculadora). Por ejemplo puedes escribir:

```
3 + 7
```

Python analizará la expresión y te regresará el resultado de la operación:



```
10
```

Ejecuta la siguiente celda y pruebalo tú mismo.


```
[ ] 3 + 7
```


Below the code cell, the output "10" is displayed.

Notebooks en la nube con Colaboratory


 Welcome To Colaboratory 

File Edit View Insert Runtime Tools Help

Share 

+ Code + Text  Copy to Drive

Rectangular Snip Connect Editing


 **Welcome to Colaboratory!**

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

[] **Introducing Colaboratory**

This 3-minute video gives an overview of the key features of Colaboratory:



▼ **Getting Started**

The document you are reading is a [Jupyter notebook](#), hosted in Colaboratory. It is not a static page, but an interactive environment that lets you write and execute code in Python and other languages.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

[]

```
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

Python en tu computadora

Además de la opción en la nube, y si ya tienes experiencia usando otros lenguajes, quizá estés interesado en conocer las opciones para instalar Python y sus paquetes localmente, algunas de éstas son:

www.python.org



pypi.org/project/pip/
(paquetes)



www.anaconda.com



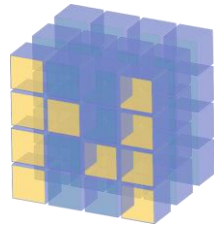
O mediante el manejador de paquetes de tu Sistema Operativo.

Propósito del módulo 1

- Familiarizarse con Python como lenguaje de programación general.
- Conocer las herramientas de Python para el análisis de datos.
- Entender la importancia del uso eficiente de recursos computacionales para el procesamiento de datos.
- Otorgarles un “cheat-sheet” de los paquetes esenciales para el procesamiento científico y visualización de Python.

Contenido del módulo

1. Las bases de Python.
2. Módulos y programación orientada a objetos.
3. Numpy.
4. Pandas.
5. Paquetes de visualización.



NumPy



pandas

matplotlib

Seaborn

Material del diplomado

El material del diplomado estará disponible en un repositorio en GitHub. Desde ese repositorio podrás acceder directamente a las notebooks desde Google Colab.



https://github.com/jhermosillo/diplomado_CDD2019

En los siguientes módulos

- Análisis estadístico de datos.
- Preprocesamiento y Normalización.
- Clasificación y Agrupamiento.
- Aprendizaje profundo.
- Taller experimental.