



**CARRERA: TECNICATURA SUPERIOR DE CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL**

**ESPACIO CURRICULAR: PROGRAMACIÓN I**

**PROFESORA: SILVIA PEROTTI**

**COHORTE: 2022**

## Módulo I: Introducción a Python y a la programación



"La computadora es capaz de procesar mecánicamente símbolos que representan información. La tarea de los programadores es construir aquellos programas que manipulan esos símbolos. Lo que concierne a la informática es la interrelación entre la manipulación de símbolos mecánica y la manipulación de símbolos humana, generalmente referidas como computación y programación, respectivamente." (Dijkstra, 1989)

### ¿Qué es programar?

```
ht=float(input("Horas trabajadas: "));  
hu=float(input("precio x hora : "));  
pagar=round(ht * hu, 3);  
print("El pago es: " + str(pagar));
```

Horas trabajadas: 12  
precio x hora : 500  
El pago es: 6000.0  
Ingrese el valor de n:

Programar es escribir un algoritmo en un lenguaje de programación concreto que la computadora pueda ejecutar.

### ¿Cómo funciona un programa de computadora?

Un programa hace que una computadora sea utilizable. Sin un programa, una computadora, no es más que un objeto, aunque esto no está más allá de los límites posibles en un futuro próximo.

La palabra clave es el **lenguaje**.



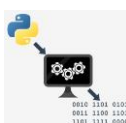
## Lenguajes de programación.

Los programas, podríamos afirmar, son como textos que interpretan las computadoras. Así como escribimos textos (una monografía, una solicitud, un CV) por medio de un programa que nos ayuda a hacerlo (un editor de textos como Word o Libreoffice Writer), también tenemos programas que nos ayudan a programar.

Para esto, se necesita:

- **Un lenguaje de programación:** es decir, una forma concreta de escritura, con reglas y símbolos.
- **Un IDE (Entorno de Desarrollo Integrado, en inglés):** un programa que nos ayuda a escribir con un determinado lenguaje de programación.

En el siguiente gráfico, se muestran ejemplos de diferentes lenguajes de programación, desde el código binario (lo que entiende nativamente la computadora) a lenguajes de bloques, utilizados para la enseñanza de la programación.



**PARA SABER MÁS:**

[Sistema binario:](#)

[Lenguajes de Bajo y Alto nivel:](#)

## ¿Qué hace a un lenguaje?

Podemos decir que cada idioma (máquina o natural, **no importa**) consta de los siguientes **elementos**:

- **ALFABETO**: Un *conjunto de símbolos* utilizados para formar palabras de un determinado idioma.
- **LÉXICO**: Un *conjunto de palabras* que el idioma ofrece a sus usuarios.
- **SINTAXIS**: Un *conjunto de **REGLAS*** utilizadas para precisar si una determinada cadena de palabras forma una oración válida.
- **SEMÁNTICA**: Un *conjunto de REGLAS* que determinan si una frase tiene SENTIDO.

```
miNomb=input("Hola como te llamas?: ");
numal= randint(1,20);
print("Bueno, "+miNomb + ", estoy pensando en un número entre 1 y 20");
intUs=0;
while(intUs<6):
    est =int(input("Intenta adivinar el número: " ));
    intUs= intUs +1;
    if(est < numal):
        print("Tu estimacion es muy baja");
    if(est > numal):
        print("Tu estimacion es muy alta");
    if(est == numal):
        break;

if(est == numal):
    intUs=str(intUs);
    print("Buen trabajo, "+miNomb+" Has adivinado el núm en "+intUs+ "intentos")
```

Fig.: Código fuente escrito en el lenguaje de Programación de alto nivel Python.

## Compilador e Intérprete.

Existen muchos lenguajes y herramientas de programación que nos permiten desarrollar por medio de bloque de código, sentencias algorítmicas que luego son traducidas a lenguaje máquina.

Estos cumplen la función de intermediarios entre el desarrollador y el hardware.



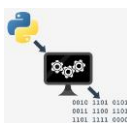
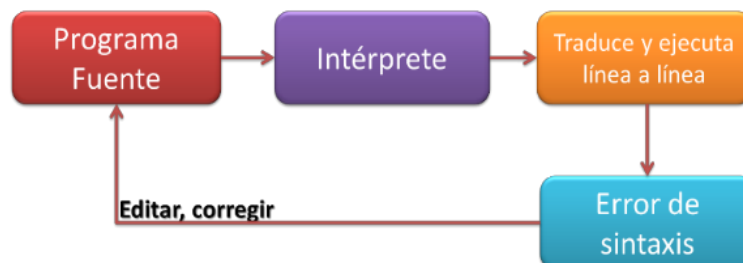
Fig.: Tipos de Lenguaje de Programación, Fuente: conogasi

Hay 2 formas diferentes de transformar un programa de un lenguaje de programación de alto nivel a un lenguaje de máquina:

**COMPILACIÓN:** El programa fuente se traduce una vez (sin embargo, esta ley debe repetirse cada vez que se modifique el código fuente) obteniendo un archivo que contiene el código de la máquina.



**INTERPRETE:** Es un programa que tiene como objetivo ejecutar código fuente de un lenguaje de alto nivel o scripting **pero sin tener que obtener código máquina como resultado final.**



**PARA SABER MÁS:**

[Compilación vs Interpretación](#) - Ventajas y Desventajas.

## ¿Qué es Python?

Python es un lenguaje de programación popular. Fue creado por Guido van Rossum y lanzado en 1991.

Se utiliza para:

- desarrollo web (lado del servidor),
- desarrollo de software,
- matemáticas,
- secuencias de comandos del sistema.



## ¿Qué puede hacer Python?

- Python se puede usar en un servidor para crear aplicaciones web.
- Python se puede usar junto con el software para crear flujos de trabajo.
- Python puede conectarse a sistemas de bases de datos. También puede leer y modificar archivos.
- Python se puede usar para manejar big data y realizar operaciones matemáticas complejas.
- Python se puede utilizar para la creación rápida de prototipos o para el desarrollo de software listo para la producción.

## ¿Por qué Python?

- Python funciona en diferentes plataformas (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python tiene una sintaxis simple similar al idioma inglés.
- Python tiene una sintaxis que permite a los desarrolladores escribir programas con menos líneas que otros lenguajes de programación.
- Python se ejecuta en un sistema de interpretación, lo que significa que el código se puede ejecutar tan pronto como se escribe. Esto significa que la creación de prototipos puede ser muy rápida.
- Python se puede tratar de forma procesal, orientada a objetos o funcional.


## Bueno saber

- La versión principal más reciente de Python es Python 3. Sin embargo, Python 2, aunque no se actualiza con nada más que actualizaciones de seguridad, sigue siendo bastante popular.
- Es posible escribir Python en un entorno de desarrollo integrado, como Thonny, Pycharm, Netbeans, Eclipse, Visual studio Code, Sublime Text, Google Colaboraty, que son particularmente útiles cuando se administran colecciones más grandes de archivos de Python.

## Sintaxis de Python en comparación con otros lenguajes de programación

- Python fue diseñado para facilitar la lectura y tiene algunas similitudes con el idioma inglés con influencia de las matemáticas.
- Python usa nuevas líneas para completar un comando, a diferencia de otros lenguajes de programación que a menudo usan punto y coma o paréntesis.
- Python se basa en la sangría, usando espacios en blanco, para definir el alcance; como el alcance de los bucles, funciones y clases. Otros lenguajes de programación a menudo usan corchetes para este propósito.

## ¿Comenzando tú trabajo con Python?

 Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

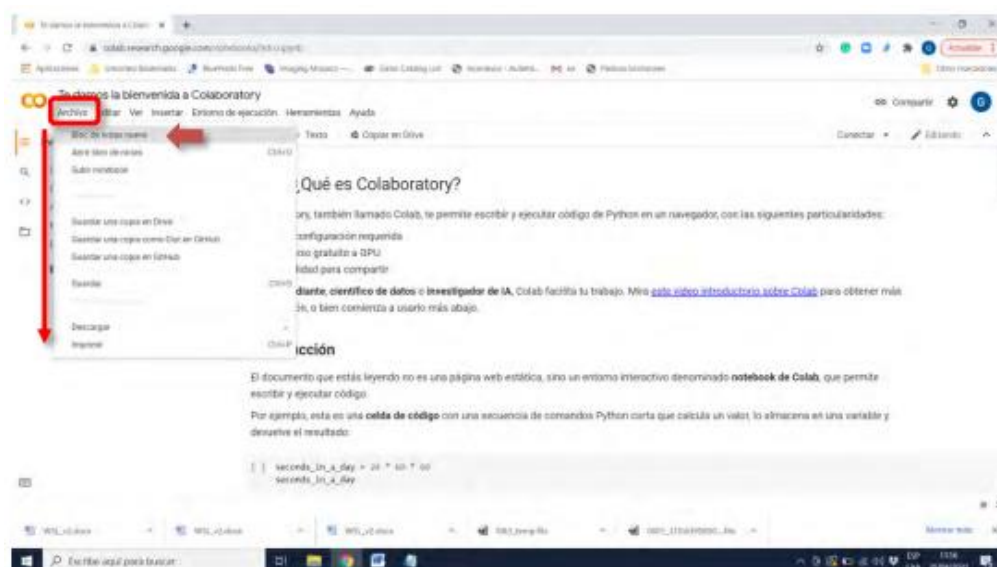
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29  
tel)] on win32  
Type "help", "copyright", "credits" or "license()" for m  
>>> |
```

Si bien existen muchos entornos de trabajo con Python, para este año te proponemos trabajar con Google Colaboraty. Si ya programas con anterioridad y estás acostumbrado a realizarlo en otro entorno, pues sigue con el mismo!

## Entorno de trabajo: Google Colaboratory

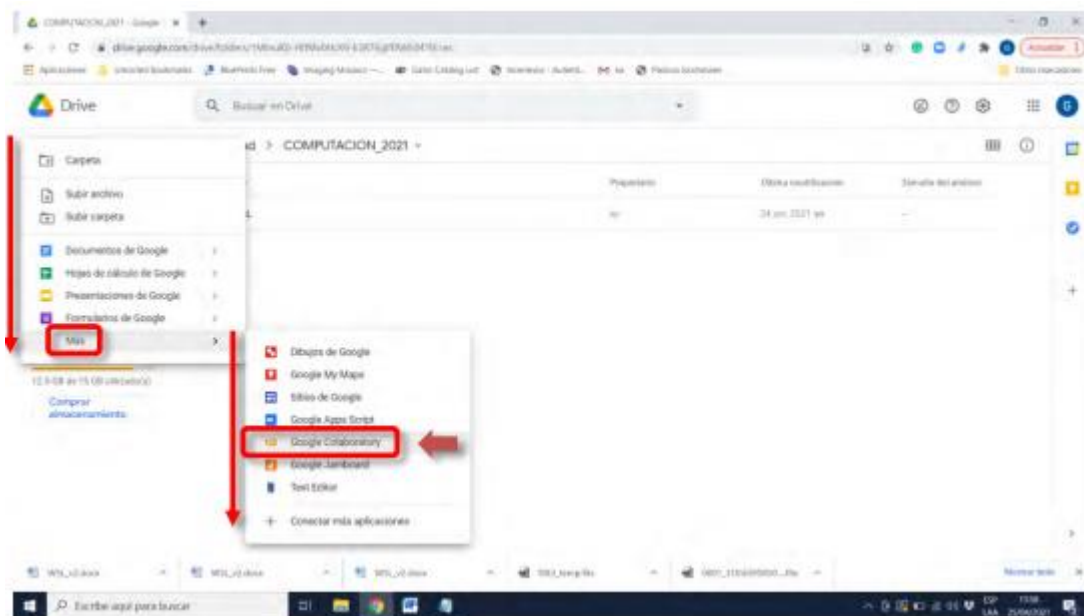


**"Google Colab"** es una herramienta para escribir y ejecutar código Python en la nube de Google. También es posible incluir texto enriquecido, "links" e imágenes. En caso de necesitar altas prestaciones de cómputo, el entorno permite configurar algunas propiedades del equipo sobre el que se ejecuta el código. En definitiva, el uso de "Google Colab" permite disponer de un entorno para llevar a cabo tareas que serían difíciles de realizar en un equipo personal. Por otro lado, siguiendo la idea de "Drive", "Google Colab" brinda la opción de compartir los códigos realizados lo que es ideal para trabajos en equipo. II. Elementos Básicos 1. Acceso a "Google Colab" Existen algunas variantes para acceder a "Google Colab", estas son: Opción A: Ir al sitio de "Google Colab" (<https://colab.research.google.com/>), ir a "Archivo" (arriba a la izquierda) y elegir la opción "Block de notas nuevo"



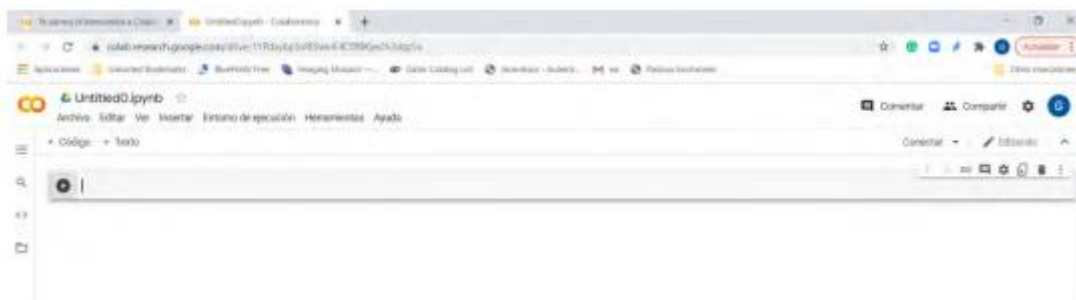


Opción B: Ir a **"Google Drive"** (<https://drive.google.com/drive/my-drive?hl=es>), hacer "click" en "Nuevo" (arriba a la izquierda), elegir la opción "Mas" (la última del menú) y seleccionar **"Google Colaboratory"**.



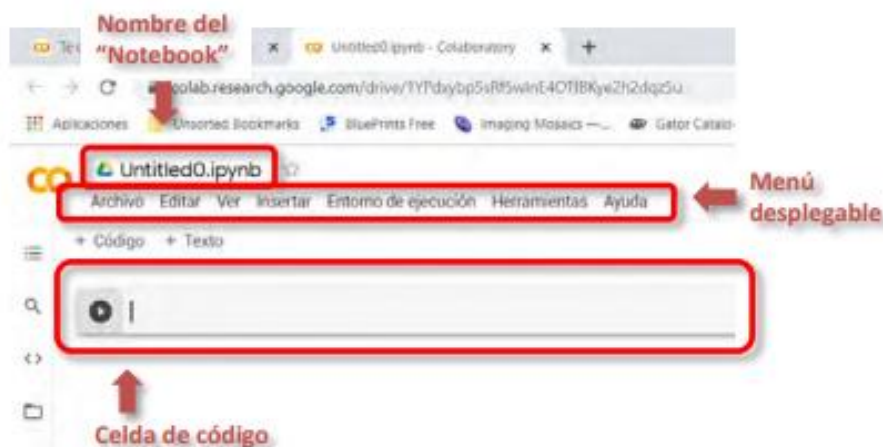
## Notebook

Independiente de la opción adoptada para acceder a **"Google Colab"**, al final se abrirá una nueva pestaña en el navegador y en ella se encontrará el entorno de trabajo.



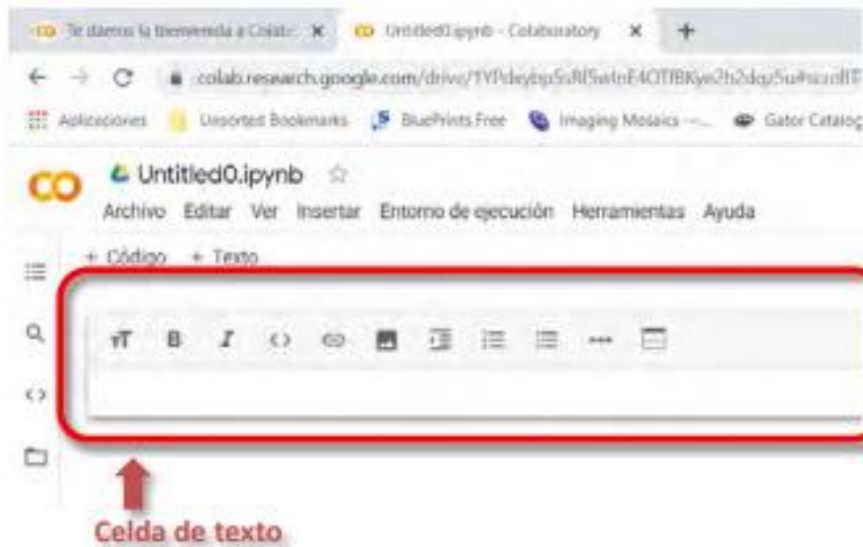
Este entorno se denomina "**Notebook**" y puede ser almacenado como un archivo tanto en "**Drive**" como en el equipo local. La estructura del "Notebook" es la siguiente:

- En la parte superior se indica el nombre del archivo correspondiente. La extensión del archivo será **".ipnb ("Interactive Python Notebook")**
- Debajo se encuentra un menú desplegable con varias opciones
- Luego se halla una "**celda**" que es el lugar donde se puede escribir el texto o el código deseado. Una "**celda**" es la unidad mínima de ejecución dentro del "**Notebook**". Usualmente no se utiliza solo una "**celda**", sino una sucesión de "celdas".



En una "**celda**" es posible escribir texto o código Python. No obstante, la "**celda**" presentada en el archivo creado es una "**celda de código**" y solo admite la escritura de código Python.

Para poder escribir texto es necesario transformarla en una "**celda de texto**" localizando el cursor en ella y utilizando las teclas "**Ctrl+M M**" (con "Ctrl+M Y" es posible volver a una "celda de código").

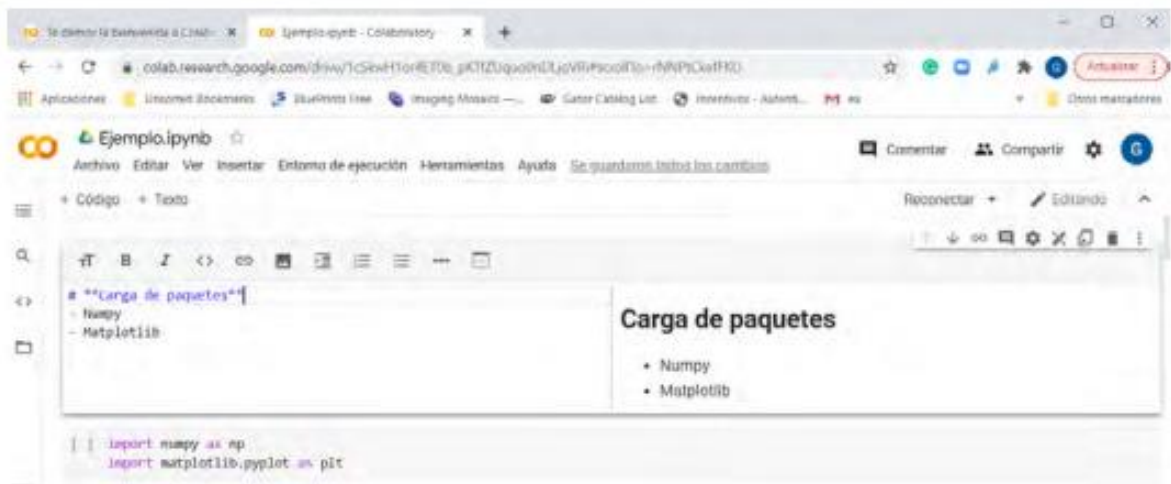


Se pueden indicar entonces las siguientes características de las "celdas":

- Luego de escribir texto o código en la “**celda**” (según corresponda), ella se puede ejecutar con el ícono pulsando las teclas “**Shift+Enter**”.
- En el caso de tratarse de una “**celda de texto**”, se admiten comandos básicos y fórmulas de “**Latex**”. También se pueden insertar “links” e imágenes
- En el caso de una “**celda de código**”, al ejecutarla, aparecerá eventual resultado debajo de la “celda”.
- En ambos tipos de celdas, al utilizar las teclas “**Shift+Enter**”, se produce su ejecución y se crea a continuación una nueva “celda de código”.
- Al situarse el cursor en una celda (de código o de texto) aparece un menú en su ángulo superior derecho que brinda diversas opciones



Las **"celdas de texto"** intercaladas entre las **"celdas de código"** permiten ir documentando el código que se está desarrollando, permitiendo su mejor comprensión. A continuación se presenta un ejemplo simple de un **"Notebook"** previo a su ejecución y conteniendo ambos tipos de **"celdas"**:



**Estadística básica**

- Definición de un conjunto de números al azar con distribución uniforme  
 $x_1, x_2, \dots, x_N$
- Valor medio o promedio  

$$prom = \frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N}$$
- Desviación estándar  

$$stdev = \sqrt{\frac{\sum_{i=1}^N (x_i - prom)^2}{N-1}}$$

**Vista previa del resultado de la "celda de texto"**

```
# Estadística básica
- Definición de un conjunto de números al azar con distribución
uniforme \\\
$ {x_1, x_2, ..., x_N} $

- Valor medio o promedio \\\
$ prom = \frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N} $

- Desviación estándar \\\
$ stdev = \sqrt{\frac{\sum_{i=1}^N (x_i - prom)^2}{N-1}} $

# -----
# estadística básica con numpy
# -----

# Conjunto de N valores con distribución uniforme entre 0 y A
N = 1000
A = 50
x = A * np.random.rand(N)

# Cálculos estadísticos
prom = np.mean(x)
stdev = np.std(x)
x_min = np.min(x)
x_max = np.max(x)

# -----
# Salida de resultados
# -----

print ('-----')
print ('N <=', N)
print ('promedio =', prom)
print ('desviación estándar =', stdev)
print ('valor mínimo =', x_min)
print ('valor máximo =', x_max)
print ('-----')
```

**Gráfico con Matplotlib**

Histograma de los datos junto con su valor medio y desviación estándar

Se nota que las **"celdas de texto"** indican en su parte derecha el futuro resultado que ellas producirán al ser ejecutadas,

Finalmente, una vez ejecutadas todas las celdas, el resultado obtenido es:

The screenshot shows a Google Colab notebook with the following content:

- Carga de paquetes**
  - Numpy
  - Matplotlib
- [1] Import numpy as np**  
**Import matplotlib.pyplot as plt**
- Estadística básica**
  - Definición de un conjunto de números al azar con distribución uniforme  $x_1, x_2, \dots, x_N$
  - Valor medio o promedio  
$$prom = \frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N}$$
  - Desviación estándar  
$$stdev = \sqrt{\frac{\sum_{i=1}^N (x_i - prom)^2}{N - 1}}$$
- [2] # Estadística básica con Numpy**  
**# Conjunto de N valores con distribución uniforme entre 0 y A**  
**N = 1000**  
**A = 50**  
**x = A \* np.random.rand(N)**  
**# Cálculos estadísticos**  
**prom = np.mean(x)**  
**stdev = np.std(x)**  
**x\_min = np.min(x)**  
**x\_max = np.max(x)**
- [3] # Salida de resultados**  
**print('-----')**  
**print('N =', N)**  
**print('promedio =', prom)**

Annotations in the image:

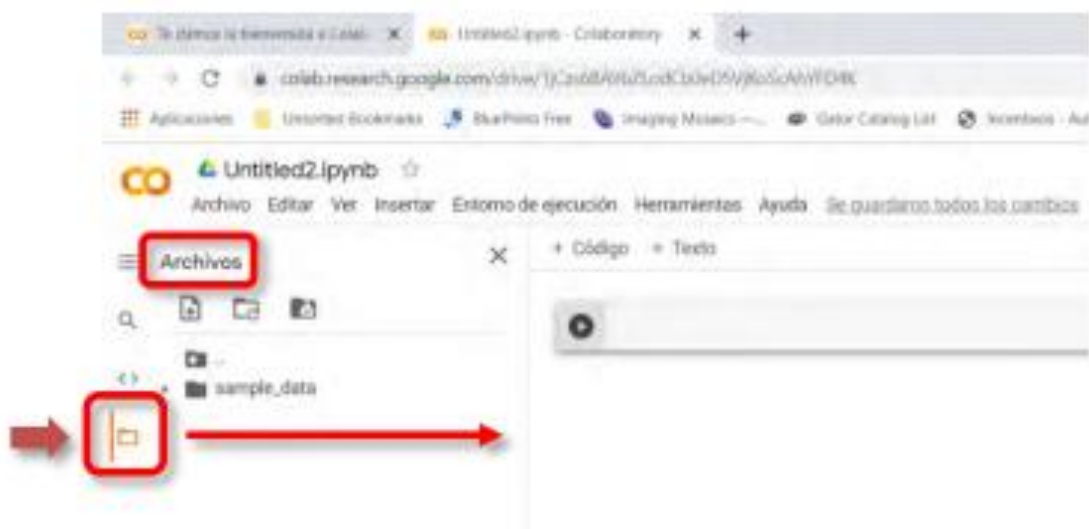
- A red box highlights the "Estadística básica" section.
- A red arrow points from the text "Resultado de la 'celda de texto'" to the "Estadística básica" section.
- A red arrow points from the text "En este caso la 'celda de código' anterior solo realiza cálculos y no produce ningún resultado" to the code cell [2].

Entorno de ejecución **"Google Colab"** permite ejecutar código Python (versión 3) sobre diferentes configuraciones del equipo, o sea que éste sólo utilice CPU que posea además algún acelerador (utilizando GPU o TPU). Además, cabe notar que existe un límite de tiempo de uso del servicio de **"Google Colab"** (unas 12 horas corridas) y de almacenamiento en RAM.

Sin embargo, tango el uso de aceleradores como los límites indicados solo se tornan relevantes para el caso de códigos muy complejos. No es necesario preocuparse por ellos para el caso de desarrollar códigos simples.

## Acceso a los archivos de datos

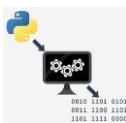
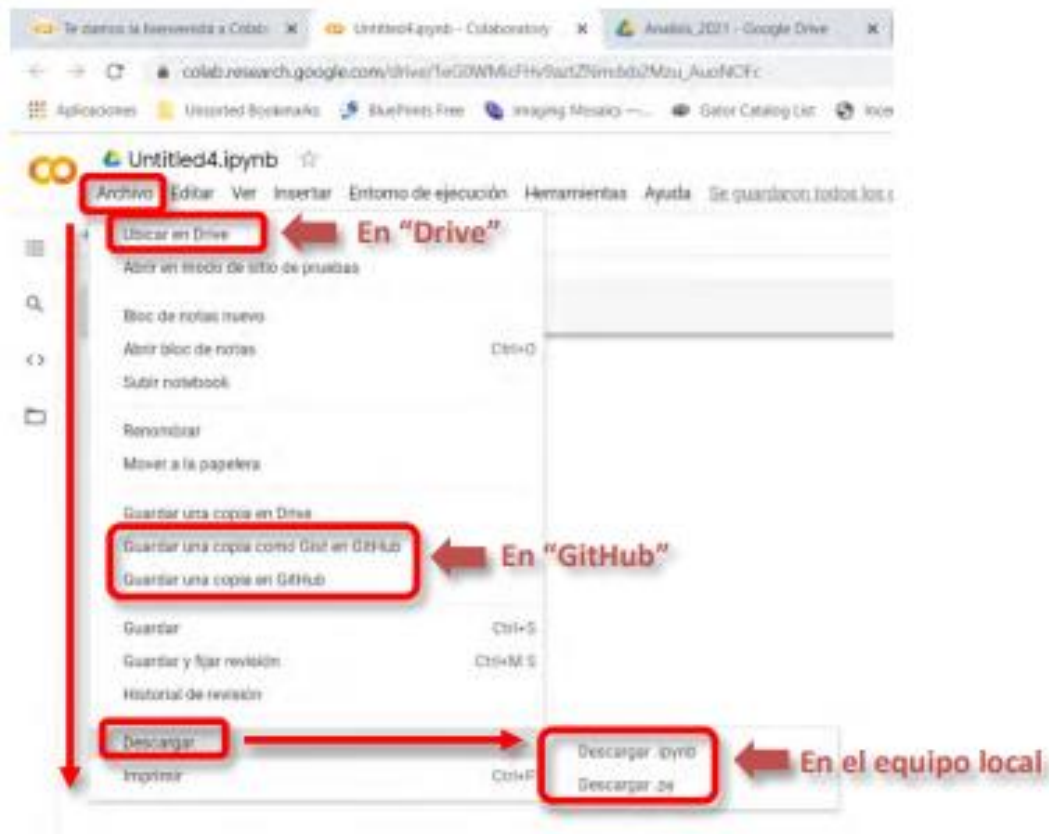
Los archivos de datos que a los que el código puede acceder se pueden visualizar haciendo "click" en el ícono de una carpeta que se encuentra a la izquierda de la pantalla, entonces se abrirá hacia la izquierda una pestaña denominada "Archivos".



## Guardando la "Notebook"

Una vez abierta la "**Notebook**" se crea un archivo con el nombre indicado en la parte superior. Ese nombre puede cambiarse simplemente haciendo "click" en ese lugar y escribiendo el nombre deseado. Si bien la "**Notebook**" es guardada regularmente en forma automática por "**Google Drive**", también es posible hacerlo manualmente utilizando con la opción "Archivo" el menú superior. Se desplegarán entonces varias opciones entre las que se encuentran la posibilidad de guardar el "**Notebook**" en "**Drive**", en "**GitHub**" o descargarla en el equipo local. En este último caso puede ser en formato ".ipnb" o ".py".





**PARA SABER MÁS:**



[https://colab.research.google.com/?hl=es-419#scrollTo=GJBs\\_fIRovLc](https://colab.research.google.com/?hl=es-419#scrollTo=GJBs_fIRovLc)



## **Resumen de los Objetivos abordados en el Módulo I:**

- Los fundamentos de la programación de computadoras, es decir, como funciona la computadora, como se ejecuta el programa, como se define y construye el lenguaje de programación.
- La diferencia entre compilación e interpretación.
- La información básica sobre Python y cómo se posiciona entre otros lenguajes de programación, y qué distingue a sus diferentes versiones.
- La descarga e Instalación del IDE Python 3.x y unas primeras líneas de código.