



unab

UNIVERSIDAD NACIONAL
GUILLERMO BROWN

Introducción a La programación

Estructuras de Datos

Introducción a la Programación

- ✓ Programa
- ✓ Lenguajes de programación
- ✓ Orígenes del lenguaje Python
- ✓ IDLE
- ✓ Instalación Python, IDLE
- ✓ Algunas herramientas

Programa

Un programa es una **secuencia de instrucciones** que especifica cómo realizar una computación. La computación puede ser:

- Resolver un sistema de ecuaciones.
- Encontrar las raíces de un polinomio.
- Buscar y reemplazar texto en un documento.
- Procesar una imagen.
- Reproducir un video.
- Etc.

Lenguaje de programación

Ahora bien.. si hago uso del lenguaje natural para escribir un conjunto de instrucciones... ¿lo puedo cargar en la computadora y ejecutar? Ejemplo:

~~Leer del teclado el número ingresado
Sumarle 3~~

NO.....

Debo hacer uso de “**lenguajes de programación**”

Lenguajes de Programación



```
else:
    print("no tenemos productos de la marca ingresada, por favor ingrese una existente")
    mar=input("ingrese la marca de los productos a actualizar, por ejemplo, 'molto': ")
    mar= mar.lower()
    por=int(input("ingresar el porcentaje de incremento del producto: "))

return (lis)

#funcion de la opcion 4: 4. Agregar un nuevo producto
def agregar (todos_los_productos):
    cod=int(input("ingrese el codigo del nuevo producto: "))
    mar=input("ingrese la marca del producto, por ejemplo, 'molto': ")
    mar= mar.lower()
    t_prod=input("ingrese el tipo de producto, por ejemplo, 'lata': ")
    t_prod= t_prod.lower()
    pre_com=float(input("ingrese el precio de compra del producto: "))
    #si ingresan una coma en vez de un punto ?
    pre_ven= float(input("ingrese el precio de venta del producto: "))
    stock_ini=int(input("ingrese el stock inicial del producto: "))
    cant_v=int(input("ingrese la cantidad vendida del producto: "))
    tipo_pre= input("ingrese el tipo de presentacion del producto: ")
    tipo_pre= tipo_pre.lower()
    while tipo_pre!="paquete" and tipo_pre!="lata" and tipo_pre!="caja":
        print("el formato utilizado para ingresar el tipo de presentacion del producto es incorrecto")
        print()
        tipo_pre= input("ingrese el tipo de presentacion del producto en el formato paquete/lata/caja: ")
        tipo_pre= tipo_pre.lower()

    prov= input("ingrese el nombre de la empresa proveedora del producto: ")
    prov= prov.upper()
```

Un programa es un conjunto de pasos lógicos escritos en un lenguaje de programación que nos permita realizar una tarea específica

Origen del lenguaje PYTHON:

- ☐ El lenguaje fué desarrollado por Guido Van Rossum a finales de los años 80 en el centro de investigación en Matematicas CWI en Holanda.
- ☐ El nombre proviene del grupo de cómicos ingleses “Monty Python”.
- ☐ Fué creado para suceder al lenguaje de programación ABC.
- ☐ En la primera versión ya estaban presentes elementos que permiten programación compleja como ser: clases con herencia, manejo de excepciones, funciones, y los tipos modulares: list, dict, str.
- ☐ Es Software libre.

Origen del lenguaje PYTHON:

Se lo define en base a:

- **Libertad 0:** Libertad de **usar** el programa, con cualquier propósito
- **Libertad 1:** Libertad de **estudiar** cómo funciona el programa, y adaptarlo a necesidades propias.
- **Libertad 2:** Libertad de **distribuir** copias
- **Libertad 3:** Libertad de **mejorar** el programa y **hacer públicas** las mejoras a los demás, de modo que toda la comunidad se beneficie.

Se distribuye con el **código fuente.**

Usos del Lenguaje

- ☐ Se ha utilizado y se lo utiliza ampliamente para:
- ☐ Programar tareas de administración de sistemas. Ej. En distribuciones Linux.
- ☐ Producción de efectos especiales de películas. Ej. Empresa: Industrial Light & Magic.
- ☐ En sistemas informáticos de la NASA.
- ☐ Gestión de grupos de discusión (ej. Yahoo).
- ☐ Parte de componentes de rastreadores Web y motores de búsqueda (ej. Google).
- ☐ Realización de juegos de computadora.
- ☐ En la bioinformática.
- ☐ Enseñanza de la programación, etc.

Algunas características

- ☐ Lenguaje de alto nivel, fácil de aprender. Expresivo y legible.
- ☐ Sintaxis elegante y tipado dinámico y fuerte.
- ☐ Es multiparadigma, puede usarse como lenguaje imperativo procedimental o como lenguaje orientado a objetos. Además tiene características de los lenguajes funcionales.
- ☐ Es Interpretado.
- ☐ Es Multiplataforma.

Python en la actualidad:

- Es un lenguaje que en los últimos años ha crecido de manera constante.

Sitio Oficial:

<http://python.org/>

Documentación en español:

<https://wiki.python.org/moin/SpanishLanguage>

Python Argentina:

<http://python.org.ar/>

Hay dos grandes versiones de Python

- Python 2.x y la 3.x (la que se va a utilizar aquí)

Instalar python:

- Para poder aplicar lo que se enseña en teoría, y realizar los ejercicios prácticos, es necesario instalar Python.
- Hay versiones para Windows, Linux, Android, etc.
- Se descarga desde: <https://www.python.org/downloads/>



IDLE (Integrated DeveLopment Environment)

- Un IDLE **es un entorno gráfico** que permite editar y ejecutar programas. No es un lenguaje.
- El IDLE trae funciones extras como el auto completar código y el coloreado de la sintaxis del lenguaje, etc
- Python trae incorporado un IDLE para escribir los programas y ejecutarlos, pero también se puede instalar otros.
- Hay varios IDLEs para poder programar en Python como el Geany, Thonny, Notepad ++.
- El GEANY se puede descargar desde:

<https://www.geany.org/download/releases/>

Viene versiones para Windows, Mac OSX y Linux.

geany.org



geany

Acerca de ▾

Descargar ▾

Documentación ▾

Apoyo ▾

Contribuir ▾

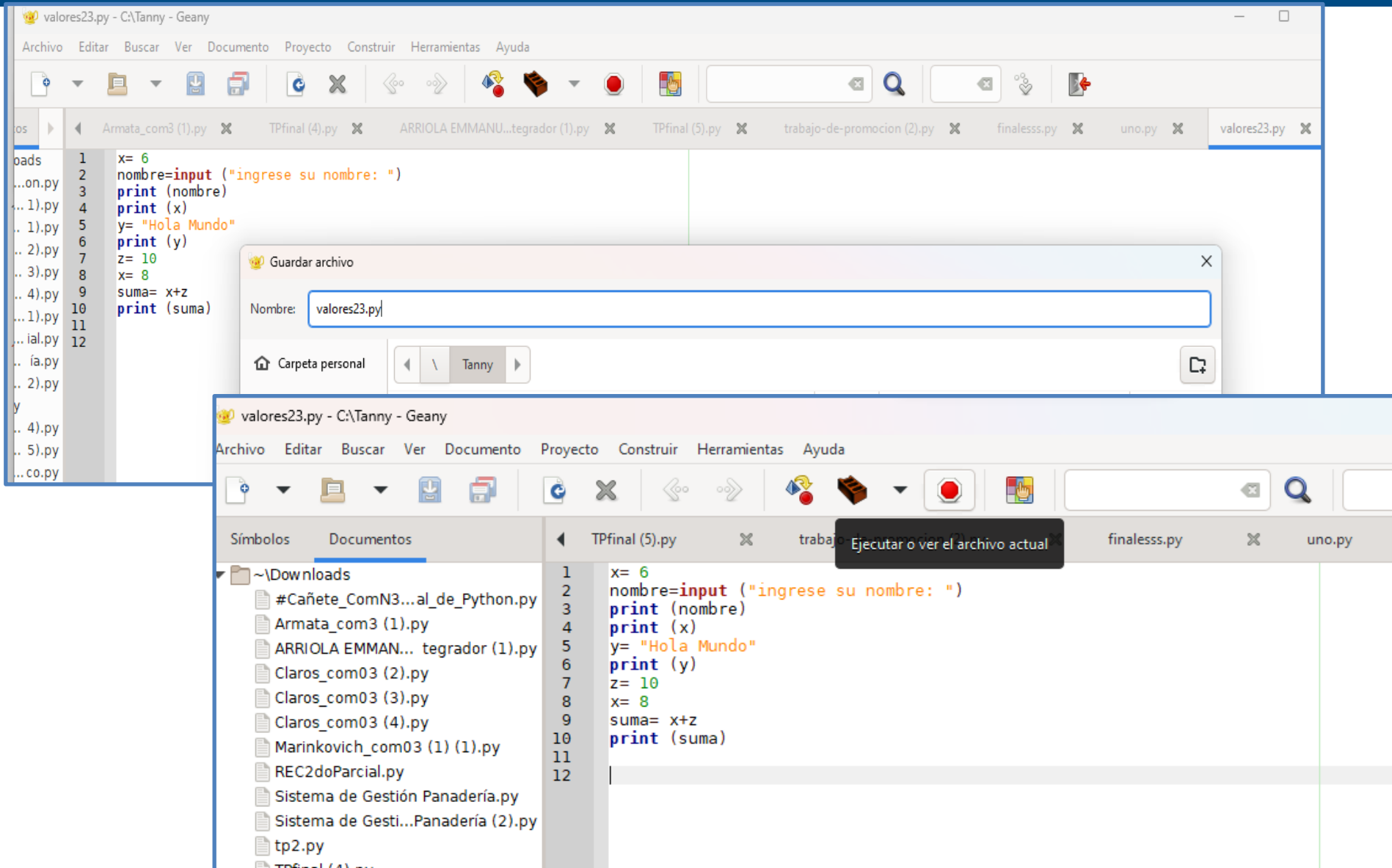
Fork me on GitHub

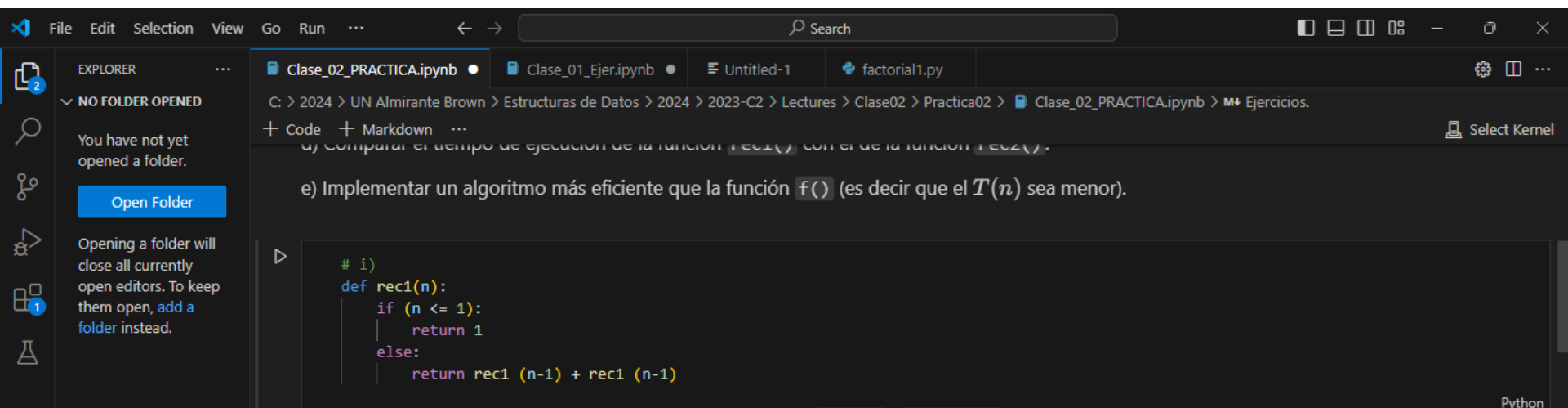
Geany - El IDE de peso mosca

Geany es un editor de texto para programadores potente, estable y ligero que proporciona toneladas de funciones útiles sin atascar su flujo de trabajo. Se ejecuta en Linux, Windows y macOS, está traducido a más de 40 idiomas y tiene soporte integrado para más de 50 lenguajes de programación.

[Descargar Geany 1.38.0 »](#)

```
editor.c - /home/ban/progs/geany/src - Geany
File Edit Search View Document Project Build Tools Help
Symbols Documents
  Functions
    add_kb [213]
    append_tooltip [1041]
    at_eol [2629]
    auto_close_chars [99]
    auto_close_chars [1520]
    auto_multiline [98]
    auto_multiline [2518]
    auto_update_margin_width [989]
    autocomplete_check_html [2088]
    autocomplete_check_html [2088]
    autocomplete_scope [693]
    autocomplete_tags [2071]
    brace_match [1563]
    change_tab_indentation [5210]
    check_line_breaking [537]
    check_partial_completion [1018]
    close_block [100]
    close_block [1604]
    copy [5345]
    count_indent_size [104]
    count_indent_size [2771]
    create_new_sci [4898]
  editor.c
441 void editor_toggle_fold(GeanyEditor *editor, gint line, gint mod
442 {
443     ScintillaObject *sci;
444     gint header;
445     g_return_if_fail(editor != NULL);
446     sci = editor->sci;
447     /* when collapsing a fold range whose starting line is offset
448      * scroll the starting line to display at the top of the view
449      * Otherwise it can be confusing when the document scrolls do
450      * the folded lines. */
451     if ((sci_get_fold_level(sci, line) & SC_FOLDLEVELNUMBERMASK)
452         & sci_get_fold_level(sci, line) & SC_FOLDLEVELHEADERFLAG)
453     {
454         gint parent = sci_get_fold_parent(sci, line);
455         gint first = sci_get_first_visible_line(sci);
456         parent = SCM(sci, SCI_VISIBLEFROMDOOLINE, parent, 0);
457         if (first > parent)
458             SCM(sci, SCI_SETFIRSTVISIBLELINE, parent, 0);
459     }
460     /* find the fold header of the given line in case the one cli
461     if (sci_get_fold_level(sci, line) & SC_FOLDLEVELHEADERFLAG)
462         header = line;
463     else
464         header = sci_get_fold_parent(sci, line);
465 }
14:20:25: This is Geany 1.36 (git - 4bc9db44).
14:20:25: File /home/ban/progs/geany/src/editor.c opened (1).
Status
line: 449 / 5356 col: 4 sel: 0 INS TAB mode: LF encoding: UTF-8 filetype: C scope: editor.toggle_fold
```





Google Colab

A screenshot of the Google Colab homepage. The browser address bar shows 'colab.google'. The page features the Google Colab logo, navigation links for 'Blog', 'Notas de la versión', 'Cuadernos', and 'Recursos', and three buttons: 'Colaboración abierta', 'Nuevo cuaderno', and 'Inscribirse'. The main content area has a background image of a server room and a white text box with the title 'Colaboratorio de Google' and a description of the service. At the bottom of this text box are two buttons: 'Colaboración abierta' and 'Nuevo cuaderno'.

← → ↻ colab.google

Google colab Blog Notas de la versión Cuadernos Recursos

Colaboración abierta Nuevo cuaderno Inscribirse

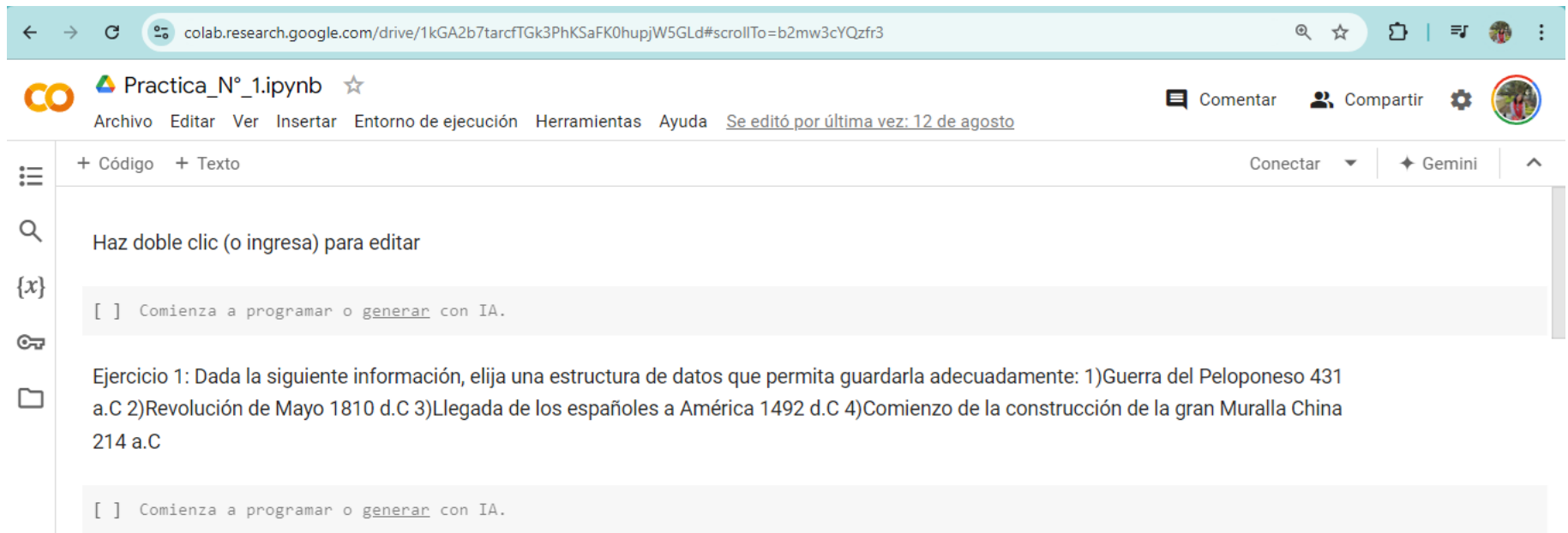
Colaboratorio de Google

Colab es un servicio alojado de Jupyter Notebook que no requiere configuración para su uso y brinda acceso gratuito a recursos informáticos, incluidas GPU y TPU. Colab es especialmente adecuado para el aprendizaje automático, la ciencia de datos y la educación.

Colaboración abierta Nuevo cuaderno

Algunas herramientas

Google Colab



The screenshot displays the Google Colab web interface. The browser address bar shows the URL: `colab.research.google.com/drive/1kGA2b7tarcfTGk3PhKSaFK0hupjW5GLd#scrollTo=b2mw3cYQzfr3`. The notebook title is "Practica_N°_1.ipynb". The top navigation bar includes options like "Archivo", "Editar", "Ver", "Insertar", "Entorno de ejecución", "Herramientas", and "Ayuda". On the right, there are buttons for "Comentar", "Compartir", and a settings gear. The main workspace is divided into two tabs: "+ Código" (active) and "+ Texto". The code cell contains the following text:

```
[ ] Comienza a programar o generar con IA.
```

Ejercicio 1: Dada la siguiente información, elija una estructura de datos que permita guardarla adecuadamente: 1)Guerra del Peloponeso 431 a.C 2)Revolución de Mayo 1810 d.C 3)Llegada de los españoles a América 1492 d.C 4)Comienzo de la construcción de la gran Muralla China 214 a.C

[] Comienza a programar o generar con IA.

Algunas herramientas



<http://programacion.espol.edu.ec/%23/gu%C3%ADa-de-instalaci%C3%B3n/instalaci%C3%B3n-de-jupyter-notebook/>

The screenshot shows the JupyterLab web interface. At the top, there's a Jupyter logo and a browser address bar showing 'localhost:8888/tree'. Below the browser bar is a navigation bar with 'File', 'View', 'Settings', and 'Help' menus. The main area is divided into two tabs: 'Files' and 'Running'. The 'Files' tab is active, showing a file browser. On the right side of the file browser, there are three buttons: 'New', 'Upload', and a refresh icon. The 'Upload' button is highlighted with a red box. Below the buttons, there's a table of files and folders. The table has columns for 'Name', 'Last Modified', and 'File Size'. The files listed are 'Music', 'OneDrive', 'OpenVPN', 'Saved Games', 'Searches', 'Videos', 'Clase_01_Ejer.ipynb', 'Clase_01_Objeto.ipynb', and 'Clase_02_Examples.ipynb'. The 'Clase_01_Ejer.ipynb' and 'Clase_01_Objeto.ipynb' files are highlighted with a red box.

jupyter

localhost:8888/tree

jupyter

File View Settings Help

Files Running

Select items to perform actions on them.

New Upload

Name	Last Modified	File Size
Music	last year	
OneDrive	37 minutes ago	
OpenVPN	2 years ago	
Saved Games	last year	
Searches	last year	
Videos	last month	
Clase_01_Ejer.ipynb	1 hour ago	5 KB
Clase_01_Objeto.ipynb	2 days ago	16.5 KB
Clase_02_Examples.ipynb	48 minutes ago	3.9 KB



jupyter Clase_02_Examples Last Checkpoint: 54 minutes ago

File Edit View Run Kernel Settings Help

Trusted

 Markdown ▾

JupyterLab   Python 3 (ipykernel)  

Ejemplos:

Ejemplo 1:

Analizar el tiempo de ejecución del siguiente algoritmo.

```
[1]: def es_par(n):  
      """Determina si un número es par de forma recursiva"""  
  
      if (n == 1):  
          return False  
      elif (n == 0):  
          return True  
      else:  
          return es_par(n // 2)
```

Casos Base: lineas 4 y 6