

**Universidad Nacional de San Agustín
Facultad de Ingeniería
Escuela Profesional de Ingeniería de
Sistemas**



**Curso: Introducción al Desarrollo Web
Laboratorio**

Grupo: “D”

**Docente: Carlo Jose Luis Corrales
Delgado**

**Título de la Actividad: Laboratorio 21 –
PYTHON: Tópicos Avanzados**

**Estudiante: Alessandro Josue Justo
Vilca**

2025

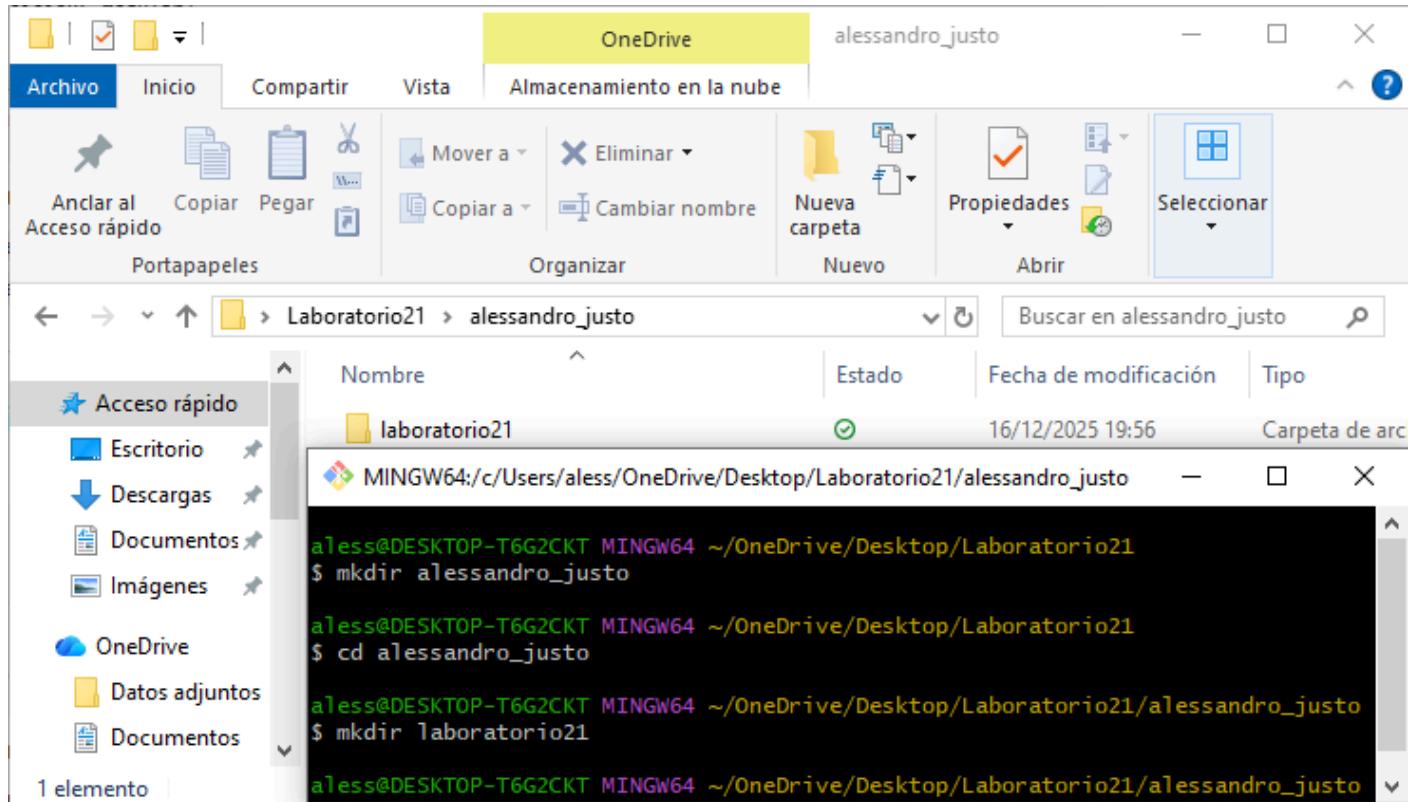
1. Crear un directorio que tenga su nombre y un subdirectorio laboratorio21. Recomendación: usar minúsculas, sin espacios, sin tildes ni “ñ” y con guiones medios o bajos

// Laboratorio Nro 21 - Ejercicio1

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 1min



2. Utilizar los atajos de teclado o combinación de teclas para agilizar su trabajo

// Laboratorio Nro 21 - Ejercicio2

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 1min

Atajos de teclado que use y tuve en cuenta en VS Code:

Ctrl + Ñ: abre la terminal integrada.

Ctrl + /: comentar línea seleccionada.

Alt + ↑: mover línea hacia arriba.

Alt + ↓: mover línea hacia abajo.

Ctrl + Alt + ↓: seleccionar con el tabulador varias líneas para modificar simultáneamente las líneas.

Shift + Alt + ↓: duplicar línea.

Atajos de teclado útiles (recomendados)

Ctrl + S → guardar

Ctrl + C / Ctrl + V → copiar / pegar

Ctrl + Z / Ctrl + Y → deshacer / rehacer

Ctrl + F → buscar texto en archivo

Ctrl + Shift + F → buscar en todo el proyecto

Ctrl + P → abrir archivo por nombre (VSCode)

Ctrl + ` → abrir terminal integrado (VSCode)

Ctrl + Shift + T → reabrir pestaña cerrada (navegador)

Alt + Tab → cambiar de aplicación

3. Usando una jerarquía de clases que permita calcular el área y el perímetro de Rectángulo, Triángulo y Círculo. Crear una lista con un objeto de cada figura y mostrar sus datos

// Laboratorio Nro 21 - Ejercicio3
// Autor: Alessandro Josue Justo Vilca
// Colaboró : sin colaboradores
// Tiempo : 28min

Código:

The screenshot shows a code editor window with the title "ejercicio03.py". The code is a Python script defining three classes: Figura, Rectangulo, and Triangulo, which inherit from Figura. The Figura class has abstract methods for area and perimeter. The Rectangulo class has an __init__ method and overrides area and perimeter. The Triangulo class has an __init__ method and overrides area and perimeter. The Circulo class has an __init__ method and overrides area and perimeter.

```
ejercicio03 > ejercicio03.py > ...
1 import math
2
3 class Figura:
4     def area(self):
5         pass
6
7     def perimetro(self):
8         pass
9
10
11 class Rectangulo(Figura):
12     def __init__(self, base, altura):
13         self.base = base
14         self.altura = altura
15
16     def area(self):
17         return self.base * self.altura
18
19     def perimetro(self):
20         return 2 * (self.base + self.altura)
21
22
23 class Triangulo(Figura):
24     def __init__(self, lado):
25         self.lado = lado
26
27     def area(self):
28
29         return (math.sqrt(3) / 4) * self.lado ** 2
30
31     def perimetro(self):
32         return 3 * self.lado
33
34
35 class Circulo(Figura):
36     def __init__(self, radio):
37         self.radio = radio
38
39     def area(self):
40         return math.pi * self.radio ** 2
41
42     def perimetro(self):
43         return 2 * math.pi * self.radio
```

```

43
44     figuras = [
45         Rectangulo(4, 5),
46         Triangulo(6),
47         Circulo(3)
48     ]
49
50     for f in figuras:
51         print("Figura:", f.__class__.__name__)
52         print("Área:", round(f.area(), 2))
53         print("Perímetro:", round(f.perimetro(), 2))
54         print("-" * 20)

```

Ejecución:

```

Seleccionar Python
Figura: Rectangulo
Área: 20
Perímetro: 18
-----
Figura: Triangulo
Área: 15.59
Perímetro: 18
-----
Figura: Circulo
Área: 28.27
Perímetro: 18.85
-----
```

4. Crea un programa que permita gestionar libros en una biblioteca usando clases y objetos.

De los libros me interesa título, autor, año y ver si está disponible. Cada libro se debe poder prestar (si está disponible) y devolver() (si no está disponible). Crear la clase biblioteca que administre los libros, debe manejar un conjunto de libros y se debe poder agregar libro nuevo, listar libros para mostrar todos los libros con su disponibilidad, buscar por autor y prestar libro (busca el libro por título y lo presta si está disponible). Crear una clase Libro Digital que herede de Libro y tenga propiedades adicionales de formato (ej: "PDF", "EPUB") y tamañoMB. Sobrescribir el método prestar para que siempre esté disponible (los libros digitales no se prestan físicamente).

Crear al menos 3 libros físicos y 2 libros digitales, agregarlos a la biblioteca y probar los métodos:

- Listar todos los libros
- Prestar un libro físico
- Prestar un libro digital 5 veces
- Intentar prestar un libro ya prestado
- Buscar libros por autor

// Laboratorio Nro 21 - Ejercicio4

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 47min

Código:

```

ejercicio04.py x
ejercicio04 > ejercicio04.py > ...
1  class Libro:
2      def __init__(self, titulo, autor, anio):
3          self.titulo = titulo
4          self.autor = autor
5          self.anio = anio
6          self.disponible = True

```

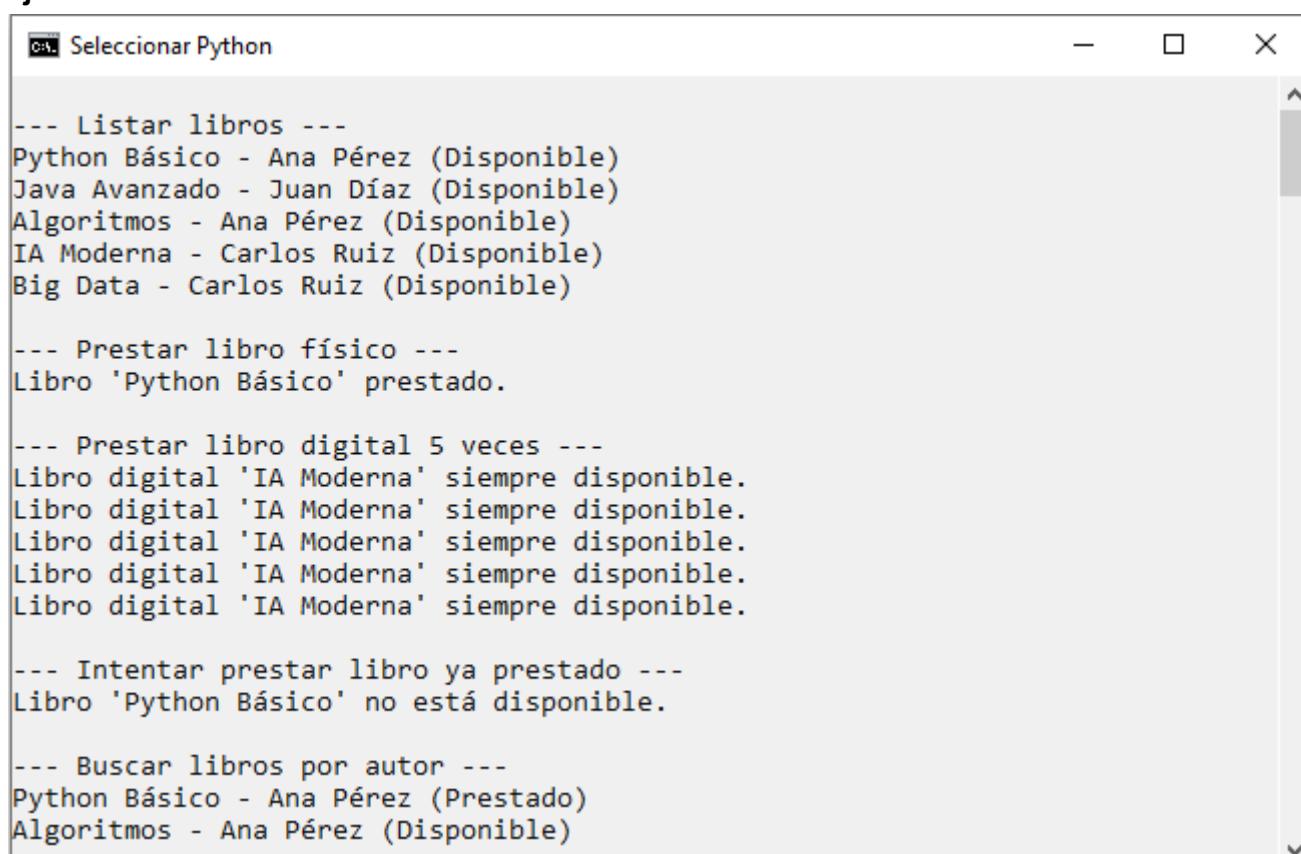
```
7
8     def prestar(self):
9         if self.disponible:
10            self.disponible = False
11            print(f"Libro '{self.titulo}' prestado.")
12        else:
13            print(f"Libro '{self.titulo}' no está disponible.")
14
15    def devolver(self):
16        self.disponible = True
17        print(f"Libro '{self.titulo}' devuelto.")
18
19    def __str__(self):
20        estado = "Disponible" if self.disponible else "Prestado"
21        return f"{self.titulo} - {self.autor} ({estado})"
22
23
24 class LibroDigital(Libro):
25     def __init__(self, titulo, autor, anio, formato, tamañoMB):
26         super().__init__(titulo, autor, anio)
27         self.formato = formato
28         self.tamañoMB = tamañoMB
29
30     def prestar(self):
31         print(f"Libro digital '{self.titulo}' siempre disponible.")
32
33
34 class Biblioteca:
35     def __init__(self):
36         self.libros = []
37
38     def agregar_libro(self, libro):
39         self.libros.append(libro)
40
41     def listar_libros(self):
42         for libro in self.libros:
43             print(libro)
44
45     def buscar_por_autor(self, autor):
46         for libro in self.libros:
47             if libro.autor == autor:
48                 print(libro)
49
50     def prestar_libro(self, titulo):
51         for libro in self.libros:
52             if libro.titulo == titulo:
53                 libro.prestar()
54             return
55         print("Libro no encontrado.")
56
57 biblio = Biblioteca()
```

```

59     l1 = Libro("Python Básico", "Ana Pérez", 2020)
60     l2 = Libro("Java Avanzado", "Juan Díaz", 2019)
61     l3 = Libro("Algoritmos", "Ana Pérez", 2018)
62
63     d1 = LibroDigital("IA Moderna", "Carlos Ruiz", 2022, "PDF", 5)
64     d2 = LibroDigital("Big Data", "Carlos Ruiz", 2021, "EPUB", 8)
65
66     biblio.agregar_libro(l1)
67     biblio.agregar_libro(l2)
68     biblio.agregar_libro(l3)
69     biblio.agregar_libro(d1)
70     biblio.agregar_libro(d2)
71
72     print("\n--- Listar libros ---")
73     biblio.listar_libros()
74
75     print("\n--- Prestar libro físico ---")
76     biblio.prestar_libro("Python Básico")
77
78     print("\n--- Prestar libro digital 5 veces ---")
79     for _ in range(5):
80         biblio.prestar_libro("IA Moderna")
81
82     print("\n--- Intentar prestar libro ya prestado ---")
83     biblio.prestar_libro("Python Básico")
84
85     print("\n--- Buscar libros por autor ---")
86     biblio.buscar_por_autor("Ana Pérez")

```

Ejecución:



```

Seleccionar Python

--- Listar libros ---
Python Básico - Ana Pérez (Disponible)
Java Avanzado - Juan Díaz (Disponible)
Algoritmos - Ana Pérez (Disponible)
IA Moderna - Carlos Ruiz (Disponible)
Big Data - Carlos Ruiz (Disponible)

--- Prestar libro físico ---
Libro 'Python Básico' prestado.

--- Prestar libro digital 5 veces ---
Libro digital 'IA Moderna' siempre disponible.

--- Intentar prestar libro ya prestado ---
Libro 'Python Básico' no está disponible.

--- Buscar libros por autor ---
Python Básico - Ana Pérez (Prestado)
Algoritmos - Ana Pérez (Disponible)

```

5. Leer la operación como un solo string

Ejemplo: "10 / 2"

Separar los componentes y validar:

- numero1 y numero2 deben poder convertirse a float
- operador debe ser uno de: + - * /

Realizar la operación, pero manejar las excepciones:

- División entre cero → mostrar mensaje personalizado
- Valores inválidos → manejar ValueError
- Operador inválido → lanzar una excepción personalizada (crear clase que herede de Exception)

// Laboratorio Nro 21 - Ejercicio5

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 26min

Código:

```
ejercicio05 > ejercicio05.py > ...
1  class OperadorInvalidoError(Exception):
2      pass
3
4  operacion = input("Ingresa la operación (ej: 10 / 2): ")
5
6  try:
7      partes = operacion.split()
8      num1 = float(partes[0])
9      operador = partes[1]
10     num2 = float(partes[2])
11
12     if operador not in ["+", "-", "*", "/"]:
13         raise OperadorInvalidoError("Operador inválido")
14
15     if operador == "+":
16         print(num1 + num2)
17     elif operador == "-":
18         print(num1 - num2)
19     elif operador == "*":
20         print(num1 * num2)
21     elif operador == "/":
22         if num2 == 0:
23             raise ZeroDivisionError("No se puede dividir entre cero")
24         print(num1 / num2)
25
26 except ValueError:
27     print("Error: valores inválidos")
28 except ZeroDivisionError as e:
29     print("Error:", e)
30 except OperadorInvalidoError as e:
31     print("Error:", e)
```

Ejecución:

Python

Ingresa la operación (ej: 10 / 2): 10 / 4
2.5

```
Python
Ingresá la operación (ej: 10 / 2): 10 / 0
Error: No se puede dividir entre cero
```

```
Python
Ingresá la operación (ej: 10 / 2): 10 % 0
Error: Operador inválido
```

6. Crear un módulo `geometria.py` con lo hecho en el ejercicio 3 y utilizarlo desde otro archivo

// Laboratorio Nro 21 - Ejercicio6

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 12min

Código:

```
geometria.py X main.py
ejercicio06 > geometria.py > ...
1 import math
2
3 class Rectangulo:
4     def __init__(self, b, h):
5         self.b = b
6         self.h = h
7
8     def area(self):
9         return self.b * self.h
```

```
geometria.py main.py ...
ejercicio06 > main.py > ...
1 from geometria import Rectangulo
2
3 r = Rectangulo(5, 4)
4 print("Área:", r.area())
```

Ejecución:

```
Python
Área: 20
```

7. Crear un programa que permita Copiar el contenido de un archivo a otro. 2 versiones: una para archivos de texto y otra para archivos binarios

// Laboratorio Nro 21 - Ejercicio7

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 25min

Código:

```
ejercicio07
ejercicio07.py
imagen.jpg
origen.txt
```

```
ejercicio07 > ejercicio07.py > ...
1  def copiar_texto(origen, destino):
2      with open(origen, "r") as f1:
3          contenido = f1.read()
4      with open(destino, "w") as f2:
5          f2.write(contenido)
6
7
8  def copiar_binario(origen, destino):
9      with open(origen, "rb") as f1:
10         data = f1.read()
11     with open(destino, "wb") as f2:
12         f2.write(data)
13
14
15 copiar_texto(
16     r"C:\Users\aless\OneDrive\Desktop\Laboratorio21\alejandro_justo\laboratorio21\ejercicio07\origen.txt",
17     r"C:\Users\aless\OneDrive\Desktop\Laboratorio21\alejandro_justo\laboratorio21\ejercicio07\destino.txt"
18 )
19
20 copiar_binario(
21     r"C:\Users\aless\OneDrive\Desktop\Laboratorio21\alejandro_justo\laboratorio21\ejercicio07\imagen.jpg",
22     r"C:\Users\aless\OneDrive\Desktop\Laboratorio21\alejandro_justo\laboratorio21\ejercicio07\imagen_copia.jpg"
23 )
24
25 print("Archivos copiados correctamente.")
```

```
ejercicio07 > origen.txt
1  Hola mundo
2  Programación en Python
3  EPIS - UNSA
```



Ejecución:

A screenshot of a terminal window. The title bar says 'Python'. The main area shows the command 'Archivos copiados correctamente.' (Files copied correctly.). Below the terminal window, a file browser shows a folder named 'ejercicio07' containing several files: 'destino.txt', 'ejercicio07.py', 'imagen_copia.jpg', 'imagen.jpg', and 'origen.txt'.

```
Python
Archivos copiados correctamente.

ejercicio07
├── destino.txt
└── ejercicio07.py
    ├── imagen_copia.jpg
    ├── imagen.jpg
    └── origen.txt
```

```
ejercicio07.py destino.txt imagen_copia.jpg origen.txt ...  
ejercicio07 > destino.txt  
1 Hola mundo  
2 Programación en Python  
3 EPIS - UNSA
```

```
ejercicio07.py destino.txt imagen_copia.jpg origen.txt ...  
ejercicio07 > imagen_copia.jpg
```



8. Crear una lista de diccionarios donde cada diccionario represente un equipo de fútbol, con las siguientes claves:
Nombre, país, nivelAtaque, nivelDefensa
Luego, convierte la lista completa a una cadena JSON y muéstralala en pantalla con formato legible.
// Laboratorio Nro 21 - Ejercicio8

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

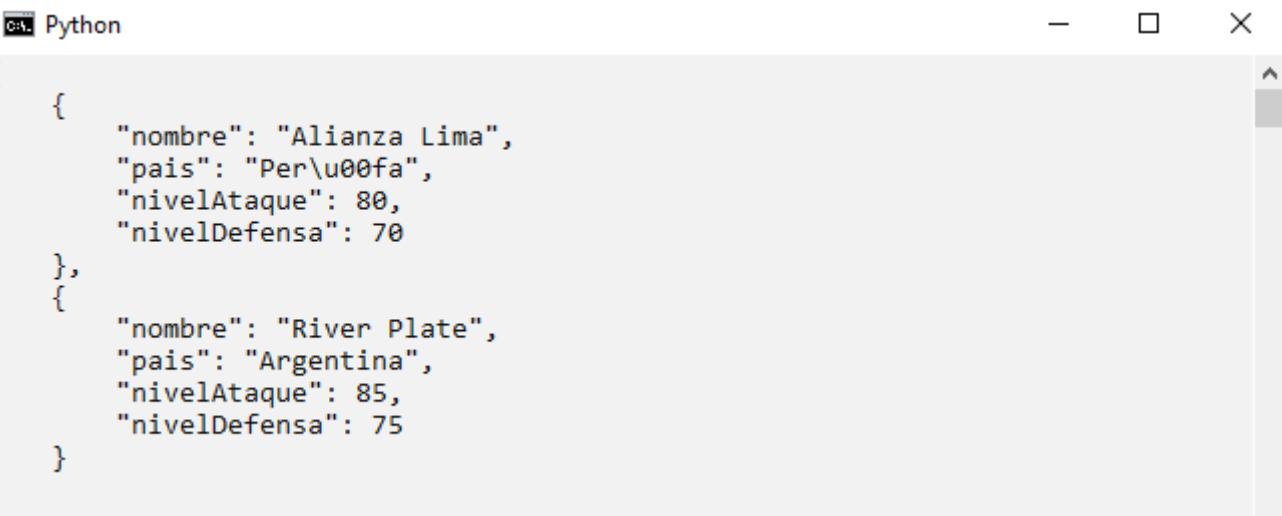
// Tiempo : 14min

Código:



```
ejercicio08 > ejercicio08.py > ...
1 import json
2
3 equipos = [
4     {"nombre": "Alianza Lima", "pais": "Perú", "nivelAtaque": 80,
5      "nivelDefensa": 70},
6     {"nombre": "River Plate", "pais": "Argentina", "nivelAtaque": 85,
7      "nivelDefensa": 75}
8 ]
9
10 texto_json = json.dumps(equipos, indent=4)
11 print(texto_json)
```

Ejecución:



```
[{"nombre": "Alianza Lima", "pais": "Per\u00fa", "nivelAtaque": 80, "nivelDefensa": 70}, {"nombre": "River Plate", "pais": "Argentina", "nivelAtaque": 85, "nivelDefensa": 75}]
```

9. Simular el comportamiento de un programa que realiza 3 consultas a una base de datos remota, cada una con un tiempo de respuesta variable entre 1 y 5 segundos. Se quiere comparar la concurrencia usando hilos, tareas asíncronas y procesos

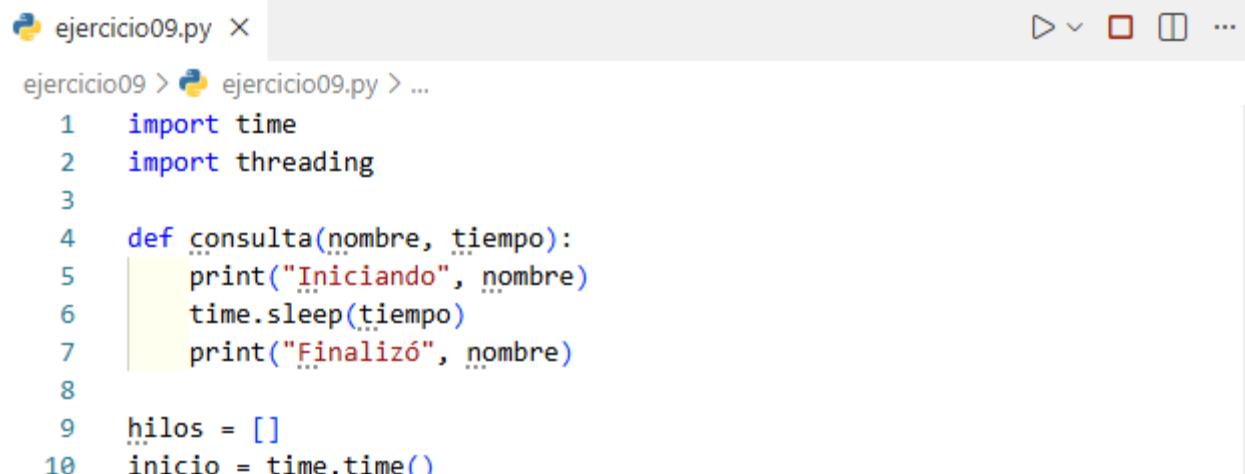
// Laboratorio Nro 21 - Ejercicio9

// Autor: Alessandro Josue Justo Vilca

// Colaboró : sin colaboradores

// Tiempo : 38min

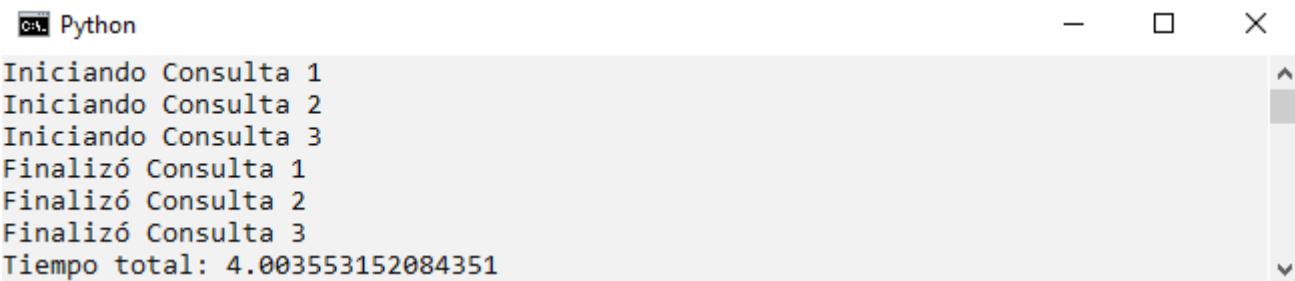
Código:



```
ejercicio09 > ejercicio09.py > ...
1 import time
2 import threading
3
4 def consulta(nombre, tiempo):
5     print("Iniciando", nombre)
6     time.sleep(tiempo)
7     print("Finalizó", nombre)
8
9 hilos = []
10 inicio = time.time()
```

```
11
12     for i in range(3):
13         t = threading.Thread(target=consulta, args=(f"Consulta {i+1}", i+2))
14         t.start()
15         hilos.append(t)
16
17     for h in hilos:
18         h.join()
19
20     print("Tiempo total:", time.time() - inicio)
```

Ejecución:



The screenshot shows a Jupyter Notebook cell with a Python logo icon and the word "Python". The cell contains the following text:

```
Iniciando Consulta 1
Iniciando Consulta 2
Iniciando Consulta 3
Finalizó Consulta 1
Finalizó Consulta 2
Finalizó Consulta 3
Tiempo total: 4.003553152084351
```

10. Crear un repositorio remoto en GitHub y subir tu repositorio local. Compartir URL y pdf con captura de pantalla del código de los archivos js y de la ejecución

// Laboratorio Nro 21 - Ejercicio1
// Autor: Alessandro Josue Justo Vilca
// Colaboró : sin colaboradores
// Tiempo : 5min

Ajustov / **laboratorio21-IDWeb**

Type: Initialized empty Git repository in C:/Users/aless/OneDrive/Desktop/Laboratorio21/alessandro_justo/laboratorio21/.git/

<> **Code** Issues Pull requests Actions Projects Wiki Security

laboratorio21-IDWeb Public

master 1 Branch 0 Tags Go to file

Ajustov Primer commit laboratorio21

ejercicio03 Primer commit laboratorio21

ejercicio04 Primer commit laboratorio21

ejercicio05 Primer commit laboratorio21

ejercicio06 Primer commit laboratorio21

ejercicio07 Primer commit laboratorio21

ejercicio08 Primer commit laboratorio21

ejercicio09 Primer commit laboratorio21

README

Add a README

Help people interested in this repository understand your project.

Add a README

```
MINGW64:/c/Users/aless/OneDrive/Desktop/Laboratorio21/alessandro_justo/laboratorio21 (master)
$ git remote add origin https://github.com/Ajustov/laboratorio21-IDWeb.git
aless@DESKTOP-T6G2CKT MINGW64 ~/OneDrive/Desktop/Laboratorio21/alessandro_justo/laboratorio21 (master)
$ git add .
warning: in the working copy of 'ejercicio03/ejercicio03.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ejercicio04/ejercicio04.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ejercicio05/ejercicio05.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ejercicio06/geometria.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ejercicio06/main.py', LF will be replaced by CR LF the next time Git touches it
warning: in the working copy of 'ejercicio07/ejercicio07.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ejercicio07/origen.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ejercicio08/ejercicio08.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ejercicio09/ejercicio09.py', LF will be replaced by CRLF the next time Git touches it
aless@DESKTOP-T6G2CKT MINGW64 ~/OneDrive/Desktop/Laboratorio21/alessandro_justo/laboratorio21 (master)
$ git commit -m "Primer commit laboratorio21"
[master (root-commit) 666c7c2] Primer commit laboratorio21
13 files changed, 255 insertions(+)
create mode 100644 ejercicio03/ejercicio03.py
create mode 100644 ejercicio04/ejercicio04.py
create mode 100644 ejercicio05/ejercicio05.py
create mode 100644 ejercicio06/__pycache__/geometria.cpython-311.pyc
create mode 100644 ejercicio06/geometria.py
create mode 100644 ejercicio06/main.py
create mode 100644 ejercicio07/destino.txt
create mode 100644 ejercicio07/ejercicio07.py
create mode 100644 ejercicio07/imagen.jpg
create mode 100644 ejercicio07/imagen_copia.jpg
create mode 100644 ejercicio07/origen.txt
create mode 100644 ejercicio08/ejercicio08.py
create mode 100644 ejercicio09/ejercicio09.py
aless@DESKTOP-T6G2CKT MINGW64 ~/OneDrive/Desktop/Laboratorio21/alessandro_justo/laboratorio21 (master)
$ git push -u origin master
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (21/21), 68.93 KiB | 6.27 MiB/s, done.
Total 21 (delta 0), reused 0 (delta 0), pack-reused 0 (From 0)
To https://github.com/Ajustov/laboratorio21-IDWeb.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
```

URL: <https://github.com/Ajustov/laboratorio21-IDWeb>