

TP Python N°1

Objetivos:

Realizar un programa en Python que integre por lo menos cinco ejemplos de codificación vistos en clase, tema libre.

- Instalar Python
- Selección a ejecutar por lo menos 5 programas de la guía
- Crear su primer programa tema libre en Python
- Subir el programa a la plataforma indicando nombre de los integrantes del grupo en archivo Python. Con el siguiente formato para el nombre del programa
`<título del programa>_apellido_TG1.py`

Código fuente Python: https://gitlab.com/nbalich/python_basico_robotica/-/tree/master

Descripción general

El presente programa en Python fue desarrollado como un trabajo integrador de ejemplos básicos de codificación, tomando como base los códigos del repositorio `python_basico_robotica` (https://gitlab.com/nbalich/python_basico_robotica/-/tree/master)

Se implementó un menú principal que organiza las funcionalidades en dos grandes módulos:

- **Módulo Aritmético:** incluye operaciones matemáticas y manejo de excepciones.
- **Módulo Sistema:** incluye funciones relacionadas con la interacción básica con el usuario y el sistema.

De esta manera, el código permite recorrer distintos conceptos de programación estructurada y orientada a objetos en Python, integrando seis ejemplos de codificación.

Este programa fue desarrollado en Google Colab, con la asistencia de ChatGPT para la integración del código.

Link al notebook:

<https://colab.research.google.com/drive/1BpHOZJeHr2Q2Hur4h3mvaB3gYDvpfZNi?usp=sharing>

Funcionalidades implementadas

1. Clases y Funciones

- Archivo de origen: `clases_y_funciones.py`.
- Se definió la clase `Operaciones` con dos métodos:
 - `cuadrado(x)`: devuelve el cuadrado de un número.
 - `seno(ang_en_radianes)`: calcula el seno de un ángulo en radianes.
- Estas funciones se invocan desde el submenú aritmético.
- Además, se agregó la función `mostrar_fecha()` que imprime la fecha y hora actual tomando como referencia la zona horaria de Buenos Aires (Argentina).

2. Suma con entrada del usuario

- Archivo de origen: **sumas_con_constants.py**.
- En lugar de depender de un archivo externo **constants.py** con valores fijos, se implementó la función **suma_con_constants()** para que el usuario ingrese dos números (A y B) y se muestre su suma.

3. Factorial recursivo

- Archivo de origen: **factorial.py**.
- Se implementó la función **factorial(n)** de manera recursiva, con caso base en **n == 0**.
- El usuario ingresa el valor y el programa valida que sea mayor o igual a 0, mostrando un error en caso contrario.

4. Manejo de errores (try/except)

- Archivo de origen: **trapeo_errores.py**.
- Se reimplementó el ejemplo para hacerlo interactivo: el usuario ingresa un número y el programa intenta dividir **10 / n**.
- Se manejan los siguientes casos:
 - Si el número es distinto de cero: muestra el resultado correcto.
 - Si el número es cero: captura la excepción **ZeroDivisionError** e informa que no se puede dividir por cero.
 - Si el usuario ingresa texto u otro valor no numérico: captura el **ValueError** e informa el error.

5. Entrada de datos (string e integer)

- Ejemplo agregado (basado en la práctica de ingreso de datos).
- El programa solicita al usuario que ingrese su **nombre** (string) y su **edad** (entero).

Luego muestra un mensaje personalizado: “**Hola, Juan tu edad es 20 años.**”

- Se validan los datos para evitar errores en caso de que la edad no sea un número.

6. Menús y organización del programa

- El código integra todas las funciones mediante:
 - **Módulo Aritmético:** cuadrado, seno, suma, factorial, manejo de errores.
 - **Módulo Sistema:** mostrar fecha/hora y entrada de nombre/edad.
- El usuario navega entre opciones de manera interactiva, pudiendo ejecutar cada ejemplo de forma independiente.

Código

```
# -*- coding: UTF-8 -*-

# Programa integrador de ejemplos (basado en repos de
python_basico_robotica)

# Incluye: clases/funciones, fecha/hora (zona Argentina), suma con
input, factorial, manejo de errores interactivo, entrada de datos.


import math

from datetime import datetime

import pytz    # librería para manejar zonas horarias


# -----
# (1) CLASES Y FUNCIONES (clases_y_funciones.py)
# -----


class Operaciones:

    def cuadrado(self, x):

        return x * x


    def seno(self, ang_en_radianes):
```

```
        return math.sin(ang_en_radianes)

def mostrar_fecha():
    """Imprime la fecha/hora actual en Argentina (Buenos Aires)."""
    tz = pytz.timezone("America/Argentina/Buenos_Aires")
    now = datetime.now(tz)
    print("Fecha y hora actual (Buenos Aires):", now.strftime("%Y-%m-%d %H:%M:%S"))

def setup():
    """Inicializa variables de control (internas)."""
    global g_controlador, g_now
    g_controlador = "INICIALIZADO"
    g_now = datetime.now(pytz.timezone("America/Argentina/Buenos_Aires"))

# -----
# (2) SUMA CON ENTRADA DEL USUARIO (basado en sumas_con_constants.py)
# -----

def suma_constants():
    """Pide al usuario dos números y muestra su suma."""
    try:
        a = float(input("Ingresá el valor de A: "))
        b = float(input("Ingresá el valor de B: "))
        print(f"A = {a}, B = {b}")
        print("A + B =", a + b)
    except ValueError:
        print("Error: Debés ingresar números.")

# -----
```

```
# -----
# (3) FACTORIAL (factorial.py)

# -----
def factorial(n):

    if n < 0:

        raise ValueError("El factorial no está definido para negativos.")

    if n == 0:

        return 1

    return n * factorial(n - 1)

# -----
# (4) TRAPEO DE ERRORES (trapeo_errores.py)

#     Ahora interactivo: el usuario ingresa el divisor

# -----
def demo_trapeo_errores():

    try:

        n = float(input("Ingresá un número para dividir 10 por él: "))

        resultado = 10 / n

        print("División correcta =", resultado)

    except ZeroDivisionError:

        print("Error: no se puede dividir por cero.")

    except ValueError:

        print("Error: ingresá un número válido.")

# -----
```

```
# (5) ENTRADA DE DATOS (nuevo agregado)

# ----

def pedir_nombre_y_edad():
    """Pide nombre y edad al usuario y los muestra."""
    try:
        nombre = input("Ingresá tu nombre: ")
        edad = int(input("Ingresá tu edad: "))
        print("Hola,", nombre, "tu edad es", edad, "años.")
    except ValueError:
        print("Error: la edad debe ser un número entero.")

# ----

# Utilidades de entrada segura

# ----

def pedir_float(msg):
    while True:
        try:
            return float(input(msg))
        except ValueError:
            print("Entrada inválida. Ingresá un número.")

def pedir_int(msg):
    while True:
        try:
            return int(input(msg))
        except ValueError:
            print("Entrada inválida. Ingresá un entero.")
```

```
# -----
# SUBMENÚS
# -----
def submenu_aritmetica():
    ops = Operaciones()

    while True:

        print("\n--- MÓDULO ARITMÉTICO ---")

        print("[1] Cuadrado (Operaciones.cuadrado)")

        print("[2] Seno (en radianes) (Operaciones.seno)")

        print("[3] Suma con números ingresados por el usuario")

        print("[4] Factorial (recursivo)")

        print("[5] Manejo de errores (división por cero)")

        print("[0] Volver")

        op = input("Elegí una opción: ").strip()

        if op == "1":

            x = pedir_float("Ingresá un número: ")

            print("cuadrado(", x, ") =", ops.cuadrado(x))

        elif op == "2":

            ang = pedir_float("Ingresá un ángulo en radianes: ")

            print("seno(", ang, ") =", ops.seno(ang))

        elif op == "3":

            suma_con_constants()

        elif op == "4":

            n = pedir_int("Ingresá un entero ( $\geq 0$ ): ")

            try:
```

```
        print(f"n{! =", factorial(n))

    except ValueError as e:

        print("Error:", e)

    elif op == "5":

        demo_trapeo_errores()

    elif op == "0":

        return

    else:

        print("Opción inválida.")

def submenu_sistema():

    while True:

        print("\n--- MÓDULO SISTEMA ---")

        print("[1] Mostrar fecha/hora actual (Argentina)")

        print("[2] Ingresar nombre y edad")

        print("[0] Volver")

        op = input("Elegí una opción: ").strip()

        if op == "1":

            mostrar_fecha()

        elif op == "2":

            pedir_nombre_y_edad()

        elif op == "0":

            return

        else:

            print("Opción inválida.")
```

```
# -----#
# PROGRAMA PRINCIPAL#
# -----#
def main():
    setup() # inicializa variables internas
    while True:
        print("\n==== PROGRAMA INTEGRADO (6 EJEMPLOS) ====")
        print("[1] Módulo Aritmético")
        print("[2] Módulo Sistema")
        print("[0] Salir")
        op = input("Elegí una opción: ").strip()
        if op == "1":
            submenu_aritmetica()
        elif op == "2":
            submenu_sistema()
        elif op == "0":
            print("¡Chau!")
            break
        else:
            print("Opción inválida.")

if __name__ == "__main__":
    main()
```