

Examen – Programación en Python

Requisitos Generales

- Se debe crear un entorno llamado **P3py** con **Python en su version 3.11**
 - El proyecto debe estar **modularizado correctamente**, con cada operación en un archivo separado dentro de un paquete con `__init__.py`.
 - El menú principal debe implementarse en un **bucle infinito**, permitiendo seleccionar entre distintas funcionalidades.
 - La última opción del menú debe permitir **salir del programa**.
 - Se valorará el uso adecuado de:
 - Funciones
 - Bucles (`for`, `while`)
 - Condicionales (`if`, `elif`, `else`)
 - Entrada y salida de datos
 - Nombres de funciones y archivos descriptivos
 - Comentarios y buena documentación
 - El uso de Git y GitHub se considerará como **valor adicional (bonus)** si el repositorio está correctamente organizado.
-

Consigna

Desarrollar un programa que muestre el siguiente menú:

```
----- MENÚ DE FUNCIONES -----  
  
1. Calcular la suma de todos los divisores de un número N (excluyendo el propio  
número).  
  
2. Generar un triángulo de caracteres con letras del alfabeto hasta una altura N.  
  
3. Mostrar los primeros N números primos.  
  
4. Generar la secuencia de los primeros N términos de la serie de Fibonacci  
inversa.  
  
5. Salir del programa.  
  
Seleccione una opción:
```

Detalles por opción

Opción 1:

Función que reciba un número entero positivo **N** y calcule la suma de todos sus **divisores propios** (números

menores que **N** que lo dividen exactamente).

Opción 2:

Función que reciba un entero **N** e imprima un triángulo formado por letras del alfabeto.

Ejemplo para N = 4:

```
A
A B
A B C
A B C D
```

Opción 3:

Función que reciba un entero **N** y muestre los **primeros N números primos**, uno por línea.

Opción 4:

Función que genere los primeros **N** términos de la **serie de Fibonacci inversa**, es decir, mostrar la secuencia en orden descendente.

Ejemplo para N = 5 → **3, 2, 1, 1, 0**

Opción 5:

Mostrar un mensaje de despedida y finalizar el programa correctamente.

Estructura del proyecto esperada

```
examen_final/
├── main.py
├── funciones/
│   ├── __init__.py
│   ├── divisores.py
│   ├── triangulo_letras.py
│   ├── primos.py
│   └── fibonacci_inverso.py
├── environment.yml
├── estructura.txt
├── dist/
│   └── examen_final (ejecutable generado por PyInstaller)
```

Entregables

1. Archivo **.yml** del entorno virtual (**environment.yml**)
2. Archivo de texto con la estructura del proyecto generado con **tree** (**estructura.txt**)
3. Ejecutable generado con **PyInstaller** (**dist/<examen_final>**)

4. Comprimir el proyecto completo en un archivo **.zip** y subirlo a la plataforma
 5. (Opcional, +10 puntos): Subir el proyecto completo a un repositorio en GitHub con commits claros (se determinara un numero minimo de commits) y una estructura ordenada, SIGUIENDO LA METODOLOGIA GIT FLOW (main, develop, features/porOpcion)
-