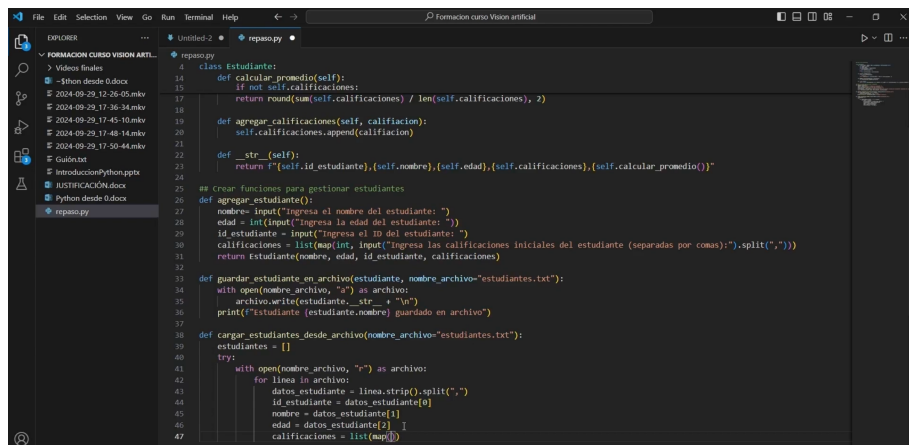




# Programación en Python: Introducción y Manejo de Librerías Fundamentales

## JUSTIFICACIÓN

Los profesionales de ingeniería que trabajan en áreas como el diseño de sistemas, la automatización de procesos, el control de calidad, la Inteligencia Artificial y la gestión de proyectos, enfrentan la necesidad de desarrollar soluciones personalizadas y optimizar flujos de trabajo. Python, con su amplio ecosistema de librerías como NumPy, Pandas y Matplotlib, permite realizar estas tareas de forma eficiente, reduciendo tiempos y costos operativos.



```
4 class Estudiante:
5     def calcular_promedio(self):
6         if not self.calificaciones:
7             return round(sum(self.calificaciones) / len(self.calificaciones), 2)
8
9     def agregar_calificaciones(self, calificacion):
10        self.calificaciones.append(calificacion)
11
12    def __str__(self):
13        return f"{self.id_estudiante}, {self.nombre}, {self.edad}, {self.calificaciones}, {self.calcular_promedio()}"
14
15    # Crear funciones para gestionar estudiantes
16    def agregar_estudiante():
17        nombre = input("Ingresa el nombre del estudiante: ")
18        edad = int(input("Ingresa la edad del estudiante: "))
19        id_estudiante = input("Ingresa el ID del estudiante: ")
20        calificaciones = list(map(int, input("Ingresa las calificaciones iniciales del estudiante (separadas por comas):").split(",")))
21        return Estudiante(nombre, edad, id_estudiante, calificaciones)
22
23    def guardar_estudiante_en_archivo(estudiante, nombre_archivo="estudiantes.txt"):
24        with open(nombre_archivo, "a") as archivo:
25            archivo.write(estudiante.__str__ + "\n")
26        print(f"Estudiante {estudiante.nombre} guardado en archivo")
27
28    def cargar_estudiantes_desde_archivo(nombre_archivo="estudiantes.txt"):
29        estudiantes = []
30        try:
31            with open(nombre_archivo, "r") as archivo:
32                for linea in archivo:
33                    datos_estudiante = linea.strip().split(",")
34                    id_estudiante = datos_estudiante[0]
35                    nombre = datos_estudiante[1]
36                    edad = datos_estudiante[2]
37                    calificaciones = list(map(int,
38
```

## OBJETIVOS

- Adquirir los fundamentos básicos de programación con Python
- Familiarizarse con herramientas y librerías avanzadas
- Desarrollar habilidades prácticas para la automatización de procesos
- Adentrarse en entornos web
- Dominar las bases del testing automatizado con Pytest

## CONTENIDOS

### Módulo 1: Introducción a la Programación con Python

- Unidad 1.1 Introducción a Python
- Unidad 1.2 Introducción Google colab
- Unidad 1.3 Mi primer programa
- Unidad 1.4 Variables y Reglas Para Nombrar Variables
- Unidad 1.5 Tipos de Datos en Python
- Unidad 1.6 Cadenas de Caracteres y algunos métodos
- Unidad 1.7 Recibiendo Datos del Usuario
- Unidad 1.8 Operadores en Python
- Unidad 1.9 Sentencias Condicionales en Python
- Unidad 1.10 Comentarios en Python
- Unidad 1.11 Listas y tuplas
- Unidad 1.12 Diccionarios
- Unidad 1.13 Ciclos
- Unidad 1.14 Funciones
- Unidad 1.15 Recursividad
- Unidad 1.16 Archivos
- Unidad 1.17 Importaciones
- Unidad 1.18 Errores y excepciones
- Unidad 1.19 Programación orientada a objetos Archivo
- Unidad 1.20 GUI
- Unidad 1.21 Ejercicio General
- Unidad 1.22 Gemini
- Unidad 1.23 Peticiones HTTP
- Unidad 1.24 Multiprocesamiento
- Unidad 1.25 Multihilo
- Unidad 1.26 Tests y TDD
- Unidad 1.27 Web
- Unidad 1.28 Bases de datos

### Módulo 2: Librerías y herramientas avanzadas

- Unidad 2.1 Numpy
- Unidad 2.2 Pandas
- Unidad 2.3 Matplotlib
- Unidad 2.4 Seaborn



70 horas /  
8 semanas



Nivel  
de  
profundidad:  
Básico\*

Modalidad:  
*e-learning*

**Ampliar información:**

web: [www.ingenierosformacion.com](http://www.ingenierosformacion.com)  
e-mail: [secretaria@ingenierosformacion.com](mailto:secretaria@ingenierosformacion.com)  
Tlf: 985 73 28 91

\* Partiendo de la base de que los cursos están dirigidos a un perfil mínimo de Ingeniero