



*Instituto Superior*  
**SANTO DOMINGO**

## PLANIFICACIÓN

**CARRERA:** TÉCNICO SUPERIOR EN DESARROLLO DE SOFTWARE

**MATERIA:** PROGRAMACION 1

**SEMESTRE:** 1° SEMESTRE

**PROFESOR:** EMMANUEL GALIOTTI

**AÑO:** 2025

## METAS DE COMPRESIÓN

### Generales

- Comprender los fundamentos de la lógica de programación y el pensamiento algorítmico.
- Desarrollar algoritmos utilizando diagramas de flujo y código Python.
- Dominar las estructuras de control (secuenciales, condicionales y repetitivas).
- Comprender y utilizar tipos de datos, variables, listas, tuplas y diccionarios en Python.
- Introducir el concepto de funciones y el uso de bibliotecas.

### Específicas

*Al finalizar esta asignatura, los alumnos habrán adquirido las siguientes habilidades teórico-prácticas:*

- Definir y aplicar conceptos de lógica, algoritmo, variable.
- Representar algoritmos mediante diagramas de flujo.
- Instalar y configurar el entorno de desarrollo Python.
- Utilizar tipos de datos (numéricos, carácter, cadena, lógicos) y operadores en Python.
- Implementar entrada/salida en consola.
- Utilizar estructuras condicionales (simples, compuestas y anidadas).
- Utilizar estructuras repetitivas (while, for).
- Manipular cadenas de caracteres, listas, tuplas y diccionarios en Python.
- Definir y utilizar funciones (con/sin retorno, con/sin parámetros).
- Comprender el concepto de parámetros y retorno de datos en funciones.
- Importar y utilizar bibliotecas (módulos) en Python.

## PROGRAMA DE CONTENIDOS

### UNIDAD I: Fundamentos de la Programación

- **Tema 1:** Introducción. (Clase 1)
  - La lógica de la programación.
  - Algoritmos.
  - Diagramas de flujo.
  - Variables.
  - Instalación de Python.

- Codificación (primeros pasos).
- **Tema 2:** Elementos básicos y Entrada/Salida. (Integración con temas anteriores)
  - Constantes y variables (repaso).
  - Identificadores.
  - Tipos de datos simples (numéricos, carácter, cadena, lógicos).
  - Operadores (matemáticos, relacionales, lógicos).
  - Entrada y salida de datos (input(), print()).
  - Asignación.
  - Bloques de código (indentación).

## **UNIDAD II: Estructuras de Control: Condicionales**

- **Tema 1:** Condicionales. (Clase 2)
  - Estructuras condicionales simples y anidadas (if, elif, else).
  - Operadores relacionales y lógicos.
  - Operadores "Or" (or) y "And" (and).

## **UNIDAD III: Estructuras de Control: Repetitivas**

- **Tema 1:** Estructura while. (Clase 3)
  - Estructuras repetitivas.
  - Estructura while.
  - Contadores.
  - Acumuladores.
- **Tema 2:** Estructura for. (Clase 4)
  - Estructura for.
  - Comentarios del código.

## **UNIDAD IV: Estructuras de Datos: Cadenas, Listas, Tuplas y Diccionarios**

- **Tema 1:** Cadenas de caracteres. (Clase 5)
  - Variables tipo "cadenas de caracteres".
  - Función len().
  - Métodos propios de las cadenas de caracteres.
- **Tema 2:** Listas (Parte 1). (Clase 6)
  - Estructura de datos tipo lista.
  - Función append().
  - Mayores y menores en una lista.
- **Tema 3:** Listas (Parte 2). (Clase 7)
  - Listas paralelas.
  - Ordenamiento de listas.
  - Carga de variables tipo lista.

- Eliminación de elementos de tipo lista.
- **Tema 4:** Tuplas. (Clase 10)
  - Estructura de datos tipo tupla.
  - Listas y tuplas anidadas.
  - Variantes repetitivas para recorrer tuplas y listas.
- **Tema 5:** Diccionarios. (Clase 11)
  - Datos tipo Diccionario.
  - Diccionarios con valores tipo lista y tuplas.

#### **UNIDAD V: Programación Modular: Funciones y Bibliotecas**

- **Tema 1:** Introducción a las funciones. (Clase 8)
  - Programación secuencial.
  - Definición de funciones.
  - Parámetros y retorno de datos.
  - Función return.
- **Tema 2:** Funciones y estructuras de datos. (Clase 9)
  - Recibir parámetros de tipo lista.
  - Retorno de estructura de datos tipo lista.
- **Tema 3:** Bibliotecas.(Clase 12)
  - Importar Bibliotecas de Python.
  - Uso de funciones de la biblioteca.
  - Importar funcionalidades de bibliotecas

## **BIBLIOGRAFÍA**