

Semana	Unidad	Tema(s)
Semana 1	Unidad 1: Introducción a la Resolución de Problemas y al Entorno de Programación	Tema 1: Introducción a la Resolución de Problemas
Semana 2	Unidad 2: Manejo de Datos, Algoritmos y Diagramas de Flujo	Tema 2: Entorno de Programación Introducción
Semana 3	Unidad 3: Lógica de Programación	Tema 1: Introducción al Manejo de Datos
Semana 4	Unidad 4: Estructura de Datos y Funciones	Tema 2: Algoritmos y Diagramas de Flujo
Semana 5	Proyecto Integrador: Planificación y Diseño	Tema 1: Introducción a las Estructuras de Decisión
Semana 6	Proyecto Integrador: Desarrollo e Implementación	Tema 2: Bucles
Semana 7	Proyecto Integrador: Pruebas, Depuración y Documentación	Tema 1: Introducción a las Estructuras de Datos
Semana 8	Proyecto Integrador: Entrega y Presentación	Tema 2: Funciones

## Actividades/Entregas

- \* Investigación sobre metodologías de resolución de problemas. Ejercicios prácticos de análisis y planteamiento de soluciones.
- \* Instalación y configuración del entorno de desarrollo Python. Primeros ejercicios con la sintaxis básica de Python. Exploración de la documentación y recursos de Python.
- \* Ejercicios de identificación y clasificación de tipos de datos. Prácticas de almacenamiento y manipulación de datos en Python.
- \* Diseño y representación de algoritmos mediante diagramas de flujo. Implementación de algoritmos sencillos en Python.
- \* Ejercicios prácticos con sentencias condicionales (if, elif, else). Resolución de problemas que requieran la toma de decisiones.
- \* Implementación de bucles for y while. Ejercicios de repetición de tareas y control de flujo.
- \* Exploración y uso de listas, tuplas, diccionarios y conjuntos en Python. Ejercicios de manipulación y acceso a datos en estructuras.
- \* Definición y uso de funciones en Python. Ejercicios de creación de funciones reutilizables.
- \* Lluvia de ideas y selección del tema del proyecto. Definición de los objetivos y alcance del proyecto. Elaboración de un documento de requisitos inicial.
- \* Diseño de la arquitectura del sistema. Elaboración de diagramas de flujo y/o pseudocódigo. Configuración del repositorio en Github.
- \* Implementación de la estructura básica del proyecto. Codificación de las funcionalidades principales. Primeras pruebas unitarias.
- \* Implementación de funcionalidades secundarias y mejoras. Integración de las diferentes partes del proyecto. Pruebas de integración.
- \* Pruebas exhaustivas del proyecto. Depuración y corrección de errores. Optimización del código.
- \* Elaboración de la documentación final (README). Comentarios en el código.
- \* Entrega del proyecto final en el repositorio de Github. Preparación de la presentación.
- \* Presentación del proyecto ante la clase.

## Observaciones

Se introducen estructuras de datos más complejas para el almacenamiento y organización de información.

Se aprende a modularizar el código mediante funciones.

Se inicia formalmente el proyecto, enfocándose en la planificación inicial.

Se diseña la solución técnica al problema planteado.

Se comienza la codificación del proyecto, implementando las funcionalidades clave.

Se completan las funcionalidades y se integran las diferentes partes del proyecto.

Se asegura la calidad del proyecto mediante pruebas y corrección de errores.

Se documenta el proyecto para facilitar su comprensión y mantenimiento.

Se entrega el proyecto final y se prepara la presentación.

Se presenta el proyecto y se comparten los resultados.