

Curso: Programación – Introducción a la Lógica y Desarrollo en Python

Contenidos Mínimos

- Pensamiento lógico y algoritmos.
- Tipos de datos, variables y operadores.
- Estructuras de control: condicionales, bucles.
- Funciones y modularización.
- Python
- Proyecto integrador simple.

Duración total: 32 horas

Modalidad: Teórico–Práctico

Formato: 16 clases de 2 horas

Nivel: Inicial

Requisitos: Manejo básico de computadora

Introducción

El curso de Introducción a la Programación tiene como objetivo desarrollar el pensamiento lógico y computacional mediante la resolución de problemas utilizando un lenguaje de programación accesible (Python).

La programación no se enseña como “código”, sino como una forma estructurada de pensar, analizar y automatizar soluciones. Durante el curso se trabajará con ejercicios prácticos progresivos, culminando en un proyecto integrador funcional.

Objetivo General

Desarrollar habilidades básicas de programación y pensamiento lógico aplicadas a la resolución de problemas simples mediante Python.

Objetivos Específicos

- Comprender qué es un algoritmo.
- Aplicar estructuras básicas de programación.
- Utilizar variables y tipos de datos.

- Implementar estructuras condicionales y repetitivas.
- Crear funciones simples.
- Resolver problemas prácticos básicos mediante código.

Desarrollo por Clases (16 clases – 2 horas cada una)

Clase 1 – ¿Qué es programar?

Se introducirá el concepto de algoritmo como secuencia ordenada de pasos para resolver un problema. Se trabajará con ejemplos cotidianos (por ejemplo, una receta de cocina) para entender la lógica paso a paso. Se explicará qué es un lenguaje de programación y se presentará Python como herramienta de trabajo.

Clase 2 – Instalación y primer programa

Se explicará la estructura básica de un programa en Python y se escribirá el primer programa utilizando la función `print()`. Se introducirán los errores comunes y cómo interpretarlos. Se explicará qué es una librería y se realizará la instalación de las principales.

Clase 3 – Variables y tipos de datos

Se explicará el concepto de variable como espacio de almacenamiento. Se trabajará con tipos básicos: enteros, decimales y texto. Se realizarán ejercicios de ingreso y muestra de datos utilizando `input()` y `print()`.

Clase 4 – Operadores y cálculos

Se abordarán operadores matemáticos (+, -, *, /) y su aplicación en cálculos simples. Se desarrollarán programas que realicen operaciones básicas como suma de dos números o cálculo de promedio.

Clase 5 – Estructuras condicionales I

Se introducirá la estructura `if`, permitiendo tomar decisiones en el programa. Se desarrollarán ejemplos como determinar si un número es positivo o negativo.

Clase 6 – Estructuras condicionales II

Se ampliará el uso de `if`, `elif` y `else`, resolviendo problemas con múltiples condiciones, como clasificación de notas (aprobado o desaprobado).

Clase 7 – Estructuras repetitivas I

Se enseñará el uso del ciclo `while`, explicando el concepto de repetición controlada. Se desarrollarán ejercicios como contar del 1 al 10 automáticamente.

Clase 8 – Estructuras repetitivas II

Se trabajará con el ciclo `for` y la función `range()`. Se desarrollarán programas que impriman tablas de multiplicar o acumulen resultados.

Clase 9 – Listas

Se introducirá el concepto de lista como colección de datos. Se enseñará cómo agregar, modificar y recorrer elementos de una lista.

Clase 10 – Funciones

Se explicará qué es una función y por qué se utiliza para organizar código. Se desarrollarán funciones simples que realicen cálculos y retornen resultados.

Clase 11 – Manejo básico de errores

Se abordará el concepto de errores comunes (sintaxis y ejecución). Se explicará cómo interpretar mensajes de error y corregirlos.

Clase 12 – Programación modular

Se trabajará en la organización del código mediante funciones y bloques reutilizables. Se desarrollará un pequeño programa estructurado en partes.

Clase 13 – Proyecto guiado I

Se iniciará un proyecto integrador, por ejemplo un sistema simple de gestión de notas o una calculadora extendida. Se diseñará el algoritmo antes de programar.

Clase 14 – Proyecto guiado II

Se continuará la implementación del proyecto integrando condicionales, ciclos y funciones.

Clase 15 – Mejora y optimización

Se revisará el proyecto incorporando validación de datos y mejoras en la estructura del código.

Clase 16 – Presentación del proyecto final

Los estudiantes presentarán su programa funcionando. Se realizará una revisión colectiva del código y cierre conceptual del curso.

Evaluación

La evaluación será práctica y continua, considerando el desempeño en ejercicios y el proyecto final integrador.

Competencias Adquiridas

Al finalizar el curso, el estudiante será capaz de analizar problemas simples y traducirlos en algoritmos estructurados, utilizar variables y tipos de datos básicos (enteros, flotantes y cadenas de texto), aplicar operadores matemáticos y lógicos, implementar estructuras condicionales (if, elif, else) y repetitivas (while, for), manipular listas como estructuras de almacenamiento de datos, crear y utilizar funciones para organizar el código de manera modular, interpretar y corregir errores básicos de programación, y desarrollar programas funcionales simples en Python aplicando buenas prácticas iniciales de organización y validación de datos. Asimismo, desarrollará pensamiento lógico, capacidad de descomposición de problemas y autonomía en la construcción de soluciones programadas.