



dev Senior Code

Python Junior Developer

Python Junior Developer



Modalidad
Online - En Vivo



Nivel de dificultad
Inicial



Dedicación
Media



Proyecto Final.
Sistema completo en Python
(consola + GitHub)



Acerca del Programa.

Este programa introduce al estudiante en el desarrollo profesional con Python, abordando desde los fundamentos de la programación hasta la construcción de sistemas completos.

A lo largo del proceso, el estudiante desarrolla habilidades clave en lógica de programación, estructuras de control, programación orientada a objetos y manejo de estructuras de datos, aplicándolas en ejercicios progresivos y proyectos prácticos.

Además, se trabaja con herramientas utilizadas en la industria como Git y GitHub, preparando al estudiante para entornos reales de desarrollo.

Al finalizar, el estudiante será capaz de construir aplicaciones funcionales en Python y desarrollar proyectos completos listos para portafolio profesional.

Objetivo principal del programa.

El objetivo principal de este programa es capacitar a los estudiantes para iniciar su camino como desarrolladores Python, dominando los fundamentos de la programación, el pensamiento lógico y la construcción de aplicaciones funcionales.

Se busca que el estudiante sea capaz de desarrollar soluciones estructuradas, aplicar buenas prácticas de código y utilizar herramientas modernas de desarrollo para mejorar su productividad y calidad.

Esto incluye:**1. Fundamentos de Programación con Python**

Aprenderás a construir la base del pensamiento lógico computacional, comprendiendo variables, estructuras de control, ciclos y funciones. También desarrollarás la capacidad de analizar problemas, identificar errores y mejorar la calidad del código desde el inicio.

2. Programación Orientada a Objetos (POO)

Dominarás el paradigma central del desarrollo en Python, creando clases, objetos y sistemas estructurados. Aplicarás encapsulamiento, herencia y buenas prácticas para construir soluciones reutilizables y escalables.

3. Manejo de Estructuras de Datos

Comprenderás cómo organizar y manipular información de manera eficiente utilizando listas, diccionarios, tuplas y conjuntos. Implementarás soluciones que te permitirán trabajar con datos de forma estructurada en aplicaciones reales.

4. Persistencia de Datos y Manejo de Archivos

Aprenderás a guardar y recuperar información utilizando archivos, permitiendo que tus aplicaciones mantengan datos entre ejecuciones. Además, trabajarás validaciones y manejo básico de errores.

5. Desarrollo de Proyecto Real en Python

Aplicarás todos los conocimientos en la construcción de un sistema completo en Python. Aprenderás a estructurar proyectos, organizar tu código, implementar funcionalidades reales y publicar tu trabajo en GitHub como parte de tu portafolio profesional.

Estructura semanal por módulo

- 2 clases en vivo por semana (2 horas cada una)
- 1 tutoría de refuerzo técnico semanal (2 horas)

Cada bloque del programa está diseñado para que el estudiante:

- Aplique los conceptos en un entorno práctico
- Trabaje con herramientas y flujos de trabajo utilizados en la industria
- Construya aplicaciones funcionales y bien estructuradas en Python



Duración del Programa

- 4 meses (1 bloque por mes)
- 2 clases por semana
- 2 horas por clase
- Total por bloque: 16 horas

MÓDULO 1: Fundamentos Modernos de Programación con Python

En este módulo, el estudiante desarrolla las bases esenciales de la programación en Python. Se enfoca en el pensamiento lógico, la comprensión de estructuras fundamentales y la aplicación práctica mediante ejercicios progresivos.

El estudiante comienza a programar desde el primer día, adoptando buenas prácticas y trabajando con herramientas utilizadas en entornos reales de desarrollo.

● CLASE 1: Configuración del Entorno y Primer Programa

Objetivos de aprendizaje

- Configurar un entorno de desarrollo profesional
- Comprender el flujo de ejecución de un programa en Python
- Conocer las herramientas básicas de desarrollo

Contenidos

- Instalación de Python
- Configuración de entorno (VS Code + extensiones Python)
- Instalación y configuración de Git
- Creación de cuenta en GitHub
- Comandos básicos de Git: init, add, commit, push
- Estructura de un programa en Python
- Creación del primer programa (Hello World)

Ejercicio práctico

- Crear estructura de proyecto organizada
- Inicializar repositorio Git
- Subir primer programa a GitHub

● CLASE 2: Variables y Tipos de Datos

Objetivos de aprendizaje

- Comprender cómo se almacenan los datos
- Utilizar variables correctamente

Contenidos

- Variables en Python
- Tipos de datos: int, float, str, bool
- Entrada y salida de datos (input / print)
- Conversión de tipos

Ejercicio práctico

- Crear programa que capture y muestre datos del usuario

● CLASE 3: Operadores y Expresiones

Objetivos de aprendizaje

- Aplicar operaciones en programas

Contenidos

- Operadores aritméticos
- Operadores relacionales
- Operadores lógicos
- Evaluación de expresiones

Ejercicio práctico

- Resolver ejercicios de cálculo y validación

● **CLASE 4: Estructuras Condicionales**

Objetivos de aprendizaje

- Implementar lógica de decisión

Contenidos

- if, elif, else
- Comparaciones
- Validación de datos

Ejercicio práctico

- Sistema de validación (edad, acceso, notas, etc.)

● **CLASE 5: Ciclos (Bucles)**

Objetivos de aprendizaje

- Comprender procesos repetitivos

Contenidos

- for
- while
- break y continue
- Contadores y acumuladores

Ejercicio práctico

- Crear programas con repetición (menú simple, tablas, etc.)

● **CLASE 6: Funciones**

Objetivos de aprendizaje

- Organizar código de forma modular

Contenidos

- Definición de funciones
- Parámetros
- Retorno de valores
- Reutilización de código

Ejercicio práctico

- Crear programa usando funciones

● CLASE 7: Integración de Conceptos

Objetivos de aprendizaje

- Aplicar todos los conceptos juntos

Contenidos

- Uso combinado de variables, condicionales, ciclos y funciones
- Resolución de problemas completos
- Buenas prácticas

Ejercicio práctico

- Crear programa funcional con múltiples opciones

● CLASE 8: Mini Proyecto del Módulo

Objetivos de aprendizaje

- Construir un sistema completo básico

Contenidos

- Diseño del programa
- Implementación de lógica
- Validaciones
- Organización del código

Ejercicio práctico

- Desarrollo de mini sistema en consola (ej: agenda, calculadora avanzada o gestión simple)
- Publicación en GitHub

MÓDULO 2: Programación Orientada a Objetos en Python

En este módulo, el estudiante aprende los fundamentos de la Programación Orientada a Objetos (POO), el paradigma central en el desarrollo profesional con Python.

Se enfoca en la creación de sistemas estructurados, reutilizables y escalables mediante clases y objetos. Se introducen conceptos clave como encapsulamiento, herencia y polimorfismo, aplicados en ejercicios prácticos y proyectos progresivos.

● CLASE 1: Introducción a la Programación Orientada a Objetos

Objetivos de aprendizaje

- Comprender el paradigma de Programación Orientada a Objetos
- Diferenciar programación estructurada vs orientada a objetos
- Crear las primeras clases y objetos en Python

Contenidos

- ¿Qué es la Programación Orientada a Objetos?
- Conceptos clave: clase, objeto, atributo, método
- Creación de clases en Python
- Instanciación de objetos
- Relación entre objetos y comportamiento

Ejercicio práctico

- Crear una clase (ej: Estudiante o Producto)
- Instanciar objetos y mostrar información

● **CLASE 2: Encapsulamiento y Buenas Prácticas**

Objetivos de aprendizaje

- Aplicar encapsulamiento en el diseño de clases
- Proteger la integridad de los datos

Contenidos

- Atributos públicos y privados
- Métodos getter y setter
- Principio de encapsulamiento
- Validación de datos dentro de clases

Ejercicio práctico

- Implementar encapsulamiento en una clase
- Validar datos antes de asignarlos

● **CLASE 3: Constructores y Métodos Especiales**

Objetivos de aprendizaje

- Inicializar objetos correctamente

Contenidos

- Método constructor (init)
- Inicialización de atributos
- Métodos dentro de clases
- Uso de self

Ejercicio práctico

- Crear clase con constructor
- Inicializar objetos con datos

● **CLASE 4: Relaciones entre Objetos**

Objetivos de aprendizaje

- Modelar sistemas con múltiples objetos

Contenidos

- Relación entre clases
- Uso de objetos dentro de otros objetos
- Diseño básico de sistemas

Ejercicio práctico

- Crear sistema con múltiples clases relacionadas (ej: Cliente - Pedido)

● CLASE 5: Herencia

Objetivos de aprendizaje

- Reutilizar código mediante herencia

Contenidos

- Concepto de herencia
- Clases padre e hija
- Reutilización de atributos y métodos

Ejercicio práctico

- Crear jerarquía de clases
(ej: Persona → Estudiante)

● CLASE 6: Polimorfismo

Objetivos de aprendizaje

- Aplicar comportamiento dinámico en objetos

Contenidos

- Polimorfismo en Python
- Métodos sobrescritos
- Diferente comportamiento según el objeto

Ejercicio práctico

- Implementar polimorfismo en un sistema

● CLASE 7: Abstracción

Objetivos de aprendizaje

- Diseñar sistemas más organizados

Contenidos

- Concepto de abstracción
- Métodos abstractos (básico)
- Separación de responsabilidades

Ejercicio práctico

- Crear sistema estructurado usando abstracción

● CLASE 8: Proyecto Integrador Orientado a Objetos

Objetivos de aprendizaje

- Aplicar todos los conceptos de POO

Contenidos

- Diseño de sistema orientado a objetos
- Organización del código
- Aplicación de buenas prácticas

Ejercicio práctico

- Desarrollo de sistema completo orientado a objetos
(ej: gestión de productos o estudiantes)
- Publicación en GitHub

MÓDULO 3: Estructuras de Datos y Colecciones en Python

En este módulo, el estudiante aprende a organizar, almacenar y manipular datos de manera eficiente utilizando estructuras de datos fundamentales en Python.

Se introducen listas, diccionarios, conjuntos y tuplas, junto con algoritmos básicos de búsqueda y ordenamiento. Se enfatiza la resolución de problemas y el desarrollo de lógica estructurada aplicada a escenarios reales.

● CLASE 1: Listas y Manejo de Datos

Objetivos de aprendizaje

- Comprender cómo almacenar múltiples datos en memoria
- Manipular listas en Python
- Recorrer y procesar información de forma eficiente

Contenidos

- ¿Qué es una lista?
- Creación de listas
- Acceso a elementos por índice
- Recorrido de listas (for)
- Modificación de valores
- Operaciones básicas

Ejercicio práctico

- Crear programa que almacene datos en una lista
- Calcular promedio, valor máximo y mínimo

● CLASE 2: Listas Avanzadas

Objetivos de aprendizaje

- Profundizar en el uso de listas

Contenidos

- Métodos de listas (append, remove, pop)
- Listas dinámicas
- Ordenamiento básico
- Uso de listas en problemas reales

Ejercicio práctico

- Crear sistema de gestión simple con listas

● CLASE 3: Diccionarios

Objetivos de aprendizaje

- Manejar estructuras clave-valor

Contenidos

- ¿Qué es un diccionario?
- Creación y uso
- Métodos principales (get, keys, values)
- Recorrido de diccionarios

Ejercicio práctico

- Crear sistema que almacene datos por clave (ej: usuarios por ID)

● CLASE 4: Conjuntos (Sets)

Objetivos de aprendizaje

- Gestionar datos sin duplicados

Contenidos

- ¿Qué es un set?
- Eliminación de duplicados
- Operaciones básicas
- Comparación con listas

Ejercicio práctico

- Crear sistema que evite registros duplicados

● CLASE 5: Tuplas

Objetivos de aprendizaje

- Comprender estructuras inmutables

Contenidos

- ¿Qué es una tupla?
- Diferencias con listas
- Uso en programas
- Casos prácticos

Ejercicio práctico

- Implementar tuplas en un sistema simple

● CLASE 6: Algoritmos de Búsqueda

Objetivos de aprendizaje

- Encontrar información dentro de estructuras

Contenidos

- Búsqueda lineal
- Búsqueda en listas y diccionarios
- Introducción a eficiencia

Ejercicio práctico

- Implementar búsqueda en estructuras

● CLASE 7: Algoritmos de Ordenamiento

Objetivos de aprendizaje

- Organizar datos de forma eficiente

Contenidos

- Ordenamiento básico
- Uso de sorted()
- Ordenamiento por criterios

Ejercicio práctico

- Ordenar datos (ej: productos por precio)

● CLASE 8: Integración de Estructuras de Datos

Objetivos de aprendizaje

- Aplicar múltiples estructuras en conjunto

Contenidos

- Uso combinado de listas, diccionarios y sets
- Organización de información
- Resolución de problemas complejos

Ejercicio práctico

- Crear sistema que combine múltiples estructuras (ej: inventario básico)

MÓDULO 4: Desarrollo de Sistema Profesional en Python (Consola)

En este módulo, el estudiante desarrolla un sistema completo en Python aplicando todos los conocimientos adquiridos durante el programa: lógica de programación, Programación Orientada a Objetos, estructuras de datos y manejo de archivos. El enfoque está en simular el trabajo real de un desarrollador junior, construyendo una aplicación funcional, organizada y lista para portafolio profesional.

PROYECTO OFICIAL DEL PROGRAMA

Sistema de Gestión de Ventas e Inventario (Consola)

El sistema permitirá:

- Registrar productos
- Gestionar stock
- Realizar ventas
- Consultar información
- Eliminar registros
- Validar operaciones
- Guardar y cargar datos desde archivos

Objetivo del módulo

Que el estudiante sea capaz de:

- Construir un sistema completo desde cero
- Aplicar Programación Orientada a Objetos en un caso real
- Manejar datos utilizando listas y diccionarios
- Implementar funcionalidades tipo CRUD
- Guardar y recuperar información con archivos
- Organizar código de forma clara y profesional

CLASE 1: Análisis del Sistema

Objetivos de aprendizaje

- Comprender el problema a resolver
- Definir el alcance del sistema

Contenidos

- Qué es un sistema real
- Flujo de una aplicación
- Definición de funcionalidades
- Identificación de datos necesarios

Ejercicio práctico

- Definir funcionalidades del sistema
- Diseñar qué datos se van a manejar

● **CLASE 2: Diseño de la Solución**

Objetivos de aprendizaje

- Planificar antes de programar

Contenidos

- Diseño de clases (Producto, Venta)
- Relación entre objetos
- Estructura del programa
- Organización del código

Ejercicio práctico

- Crear estructura base del sistema

● **CLASE 3: Estructura del Sistema**

Objetivos de aprendizaje

- Construir la base del programa

Contenidos

- Menú interactivo
- Uso de while
- Uso de funciones
- Control de flujo

Ejercicio práctico

- Implementar menú funcional

● **CLASE 4: CRUD de Productos**

Objetivos de aprendizaje

- Implementar operaciones principales

Contenidos

- Crear productos
- Listar productos
- Buscar productos
- Eliminar productos

Ejercicio práctico

- Implementar CRUD completo

● **CLASE 5: Sistema de Ventas**

Objetivos de aprendizaje

- Aplicar lógica de negocio real

Contenidos

- Registro de ventas
- Validación de stock

- Actualización de inventario
- Cálculo de totales

Ejercicio práctico

- Implementar proceso de ventas

● **CLASE 6: Persistencia de Datos**

Objetivos de aprendizaje

- Guardar información del sistema

Contenidos

- Lectura y escritura de archivos
- Guardado de productos
- Carga de datos al iniciar
- Manejo de datos persistentes

Ejercicio práctico

- Guardar y cargar datos del sistema

● **CLASE 7: Validaciones y Mejora del Sistema**

Objetivos de aprendizaje

- Hacer el sistema robusto

Contenidos

- Validación de entradas
- Manejo de errores (try/except)
- Mejora de flujo
- Optimización del código

Ejercicio práctico

- Mejorar sistema completo

● **CLASE 8: Entrega Profesional del Proyecto**

Objetivos de aprendizaje

- Preparar el proyecto para portafolio

Contenidos

- Creación de README
- Organización del proyecto
- Explicación del sistema
- Buenas prácticas en GitHub

Ejercicio práctico

- Subir proyecto final
- Documentar funcionalidades
- Presentar sistema



dev Senior Code

“Tu ruta ya está trazada. El futuro empieza contigo.”
La industria te está esperando.