

-  Modalidad: Sincrónico / Asincrónico
-  Curso grabado
-  Incluye certificado digital
-  1 Lección - 3 horas c / semana (24 horas en total)

### Lección 1: Introducción a la Programación Orientada a Objetos

Conceptos básicos de POO: clases, objetos, atributos y métodos.  
Encapsulamiento, herencia y polimorfismo.  
Comparación entre programación orientada a objetos y programación estructurada.

### Lección 2: Definición de Clases y Objetos en Python

Sintaxis para definir clases en Python.  
Creación de instancias de objetos.  
Métodos de inicialización (init) y de instancia.

### Lección 3: Atributos y Métodos de Clase

Definición y uso de atributos de clase.  
Métodos de clase: definición, decoradores y usos.

### Lección 4: Herencia y Polimorfismo

Definición y uso de la herencia en Python.  
Herencia múltiple y el método de resolución de métodos (MRO).  
Polimorfismo: uso de interfaces y clases abstractas.

-  Modalidad: Sincrónico / Asincrónico
-  Curso grabado
-  Incluye certificado digital
-  1 Lección - 3 horas c / semana (24 horas en total)

### Lección 5: Métodos Especiales (Métodos Mágicos)

Introducción a los métodos especiales en Python.  
Uso de métodos especiales para sobrecargar operadores (+, -, \*, /, etc.).  
Otros métodos especiales: str, repr, len, etc.

### Lección 6: Encapsulamiento y Modificadores de Acceso

Encapsulamiento en Python: convenciones de nomenclatura y uso de atributos privados.  
Modificadores de acceso: public, protected y private.

### Lección 7: Herencia Avanzada y Composición

Herencia múltiple y problemas asociados.  
Composición vs. herencia: pros y contras.  
Uso de composición para crear relaciones entre objetos.

### Lección 8: Diseño Orientado a Objetos y Buenas Prácticas

Principios SOLID y su aplicación en Python.  
Buenas prácticas de diseño orientado a objetos.  
Refactorización de código para mejorar el diseño y la legibilidad.