Nivel: Bachillerato	Grado: Sexto	Fecha de entrega:
Asignatura: Programación	**Tema:Programación orientada a objetos en Python: UML y herencia.**	**Temporalidad: 29 Septiembre-10 Octubre 2025**

Campo De Formación Académica: Programación

Propósito: Que los estudiantes comprendan cómo representar clases mediante diagramas UML y traduzcan estas representaciones a código en Python, aplicando conceptos básicos de la programación orientada a objetos (POO). Además, que reconozcan la importancia de la herencia en la reutilización y extensión de código.

Desempeño: Los estudiantes serán capaces de interpretar un diagrama UML y escribir el código correspondiente en Python, verificando su correcto funcionamiento, además de poder abstraer atributos adicionales para cada clase

Estándar: Aplicación de conceptos de programación orientada a objetos como clases, atributos y herencia en Python, apoyándose en representaciones gráficas (UML) para planificar y organizar programas.

Estrategia del modelo constructivista a implementar: Se aplicará una estrategia de aprendizaje por construcción de modelos, donde los estudiantes representarán gráficamente los conceptos mediante diagramas UML y los llevarán al código en Python. Se dará un acompañamiento constante, promoviendo la reflexión sobre cómo las clases y la herencia permiten organizar el programa y mejorar su comprensión.

Materiales: Cuaderno, dispositivo móvil o computadora portátil con acceso a python

Evidencias o productos que entregará: Código y diagrama UML para ejercicios planteados en clase

	Sesión (60 minutos)	**Estrategia de evaluación**
Sesión 1	Se presentará un ejemplo de diagrama UML y, a partir de él, los estudiantes observarán la equivalencia de código correspondiente en Python. Cada estudiante deberá ejecutar y comprobar el correcto funcionamiento de su programa.	Revisión individual del código desarrollado, comprobando que el estudiante logre traducir el diagrama UML a Python y que el programa funcione según lo esperado.
Sesión 2	Los estudiantes crearán un diagrama UML a partir de un código ejemplo en python y de forma inversa con otro ejemplo, crearán el código a partir de un ejemplo UML, además de incorporar atributos adicionales en las clases hija	Análisis de los diagramas UML y del código escrito por cada estudiante, verificando que las clases hijas hereden correctamente de la clase padre y que se incorporen atributos adicionales en la implementación.