



TECNICATURA SUPERIOR EN ANÁLISIS DE SISTEMAS

Paradigmas de Programación - Tema 1

Profesor Bonini Juan Ignacio

Apellido y nombre:

DNI:

Fecha:

1. **Diseñar clases utilizando Programación Orientada a Objetos y Composición**
 - 1.1. Imagina que estás desarrollando un videojuego de aventuras épicas. Tu tarea es diseñar un sistema de clases orientadas a objetos para gestionar a los personajes, sus movimientos y habilidades. Debes crear clases para representar un 'Personaje', 'Movimiento', 'Habilidad', 'Equipo'. Además deberás heredar del personaje para tener distintas razas. Utiliza la composición y asegúrate de que el diseño sea flexible y expansible para futuras características del juego. Describe las propiedades y métodos esenciales de cada clase y cómo se relacionan entre sí en el contexto del juego. Aplicar al menos un ejemplo de cada pilar de la POO.
2. **Programación orientada a objetos**
 - 2.1. ¿Cuáles son los pilares? Explicar cada uno de ellos
 - 2.2. ¿Cuál es la diferencia entre una clase y un objeto en POO?
 - 2.3. ¿Qué son los métodos de acceso y los métodos de modificación? ¿Cómo se usan en Python?
 - 2.4. ¿Cuál es la diferencia entre variable de clase y de instancia?
 - 2.5. ¿Qué es el polimorfismo y cómo se implementa?
 - 2.6. ¿Cuáles son los tipos de visibilidades? ¿Cómo se implementan en Python?
3. **Diseño de clases**
 - 3.1. ¿Qué es la relación "Es un"? Explicar y dar ejemplo en código
 - 3.2. ¿Qué es la relación "Tiene un"? Explicar y dar ejemplos en código
4. **Buenas prácticas**
 - 4.1. ¿Qué son las buenas prácticas de programación? ¿Por qué son importantes?
 - 4.2. ¿Por qué es esencial gestionar adecuadamente las importaciones en un proyecto?
5. **Algoritmos**
 - 5.1. ¿Qué es la recursividad? Explicar y dar ejemplos de uso
 - 5.2. ¿Qué son los administradores de contexto? ¿Cuáles son sus principales usos?
 - 5.3. ¿Qué son los decoradores?
 - 5.4. ¿Cuándo utilizarías un decorador en vez de administrador de contexto?
 - 5.5. ¿Qué tipos de imports existen?

Sistema de calificaciones

1 (puntaje 25%)	2 (puntaje 25%)	3 (puntaje 15%)	4 (puntaje 15%)	5 (puntaje 20%)

60%=4 70%=5 80%=6 85%=7 90%=8 95%=9 100%=10



```
class Movimiento:
    def __init__(self, velocidad, destreza):
        """Acá podían poner los atributos que querían"""
        self._velocidad = velocidad
        self.destreza = destreza

    @property
    def velocidad(self):
        return self._velocidad

    def incrementar_velocidad(self, valor):
        if valor >= self._velocidad:
            self._velocidad = valor

class Habilidad:
    def __init__(self, nombre, daño, tiempo_casteo):
        """Acá podían poner los atributos que querían"""
        self.nombre = nombre
        self.daño = daño
        self.tiempo_casteo = tiempo_casteo

class Equipo:
    def __init__(self, nombre, tipo, es_equipable, valor):
        """Acá podían poner los atributos que querían"""
        self.nombre = nombre
        self.tipo = tipo
        self.es_equipable = es_equipable
        self.valor = valor
```



```
class Personaje:
    def __init__(self, nombre, nivel, vida, mana):
        """Acá podían poner los atributos que querían"""
        self.nombre = nombre
        self.nivel = nivel
        self.vida = vida
        self.mana = mana
        self.movimiento = Movimiento(velocidad=10, destreza=5)
        self.habilidad = Habilidad("Ataque básico", 10, 1.0)
        self.equipo = Equipo("Espada", "Arma", True, 20)

    def atacar(self, enemigo):
        print(f"{self.nombre} ataca a {enemigo.nombre}")

    def usar_habilidad(self, habilidad, objetivo):
        """Abstracción, un método que no tiene implementación"""
        pass

class Orco(Personaje):
    """Aplicamos herencia"""
    def __init__(self, nombre, nivel, vida, mana):
        super().__init__(nombre, nivel, vida, mana)

    def atacar(self, enemigo):
        print(f"Golpea con un hacha a {enemigo.nombre}")

# Ejemplo de uso (esto no se pedía en el parcial)
mi_personaje = Personaje("Ejemplo", 1, 100, 50)
enemigo = Personaje("Enemigo", 1, 100, 50)

mi_personaje.movimiento.velocidad = 10
mi_personaje.usar_habilidad("Rayo", enemigo)
equipo = Equipo("Armadura", "ArmaduraCompleta", True, 30)
mi_personaje.equipo.equipar(equipo)

orco = Orco("Thrall", 5, 200, 50)

# Ejemplo del uso del de polimorfismo
orco.atacar(mi_personaje)
```