

# Tarea 2 - Refactorización de Código

Trabajo individual

## Objetivo

Refactorizar el sistema de gestión de biblioteca aplicando buenas prácticas de ingeniería de software vistas en el curso.

## Instrucciones

1. Analice el código provisto identificando al menos 5 problemas de diseño
2. Implemente las mejoras en un repositorio Git (opcional) o documento markdown para subir a Mediación Virtual.
3. Documente los cambios realizados explicando:
  - Problemas identificados
  - Soluciones implementadas
  - Buenas prácticas aplicadas
4. Incluya en dicho documento:
  - Explicación técnica de al menos 3 mejoras significativas
  - Enlace al repositorio si aplica o el código con los cambios respectivos.

## Código

```
class Libro:
    def __init__(self, titulo, autor, genero, paginas, anio_publicacion,
disponible=True):
        self.titulo = titulo
        self.autor = autor
        self.genero = genero # 'novela', 'ciencia', 'historia'
        self.paginas = paginas
        self.anio_publicacion = anio_publicacion
        self.disponible = disponible

    def calcular_popularidad(self):
        if self.genero == 'novela':
            base = 50
            extra = self.paginas / 10
        elif self.genero == 'ciencia':
            base = 70
            extra = self.paginas / 5
        elif self.genero == 'historia':
            base = 40
            extra = self.paginas / 8
        else:
            base = 10
            extra = 0
        return base + extra

    def es_antiguo(self):
```

```

        if self.anio_publicacion < 1980:
            return True
        else:
            return False

def imprimir_datos(self):
    print(f"Título: {self.titulo}")
    print(f"Autor: {self.autor}")
    print(f"Género: {self.genero}")
    print(f"Páginas: {self.paginas}")
    print(f"Año: {self.anio_publicacion}")
    print(f"Disponible: {'Sí' if self.disponible else 'No'}")
    print(f"Popularidad: {self.calcular_popularidad()}")
    print(f"Es antiguo: {'Sí' if self.es_antiguo() else 'No'}")
    print("-----")

```

---

```

from libro import Libro

class Biblioteca:
    def __init__(self):
        self.libros = []

    def agregar_libro(self):
        titulo = input("Título: ")
        autor = input("Autor: ")
        genero = input("Género (novela/ciencia/historia): ").lower()
        paginas = int(input("Número de páginas: "))
        anio = int(input("Año de publicación: "))

        l = Libro(titulo, autor, genero, paginas, anio)
        self.libros.append(l)
        print("Libro agregado!")

    def generar_reporte(self):
        total = len(self.libros)
        antiguos = 0
        disponibles = 0
        popularidad_total = 0

        for l in self.libros:
            l.imprimir_datos()
            if l.es_antiguo():
                antiguos += 1
            if l.disponible:
                disponibles += 1
            popularidad_total += l.calcular_popularidad()

        print("\nREPORTE BIBLIOTECA:")
        print(f"Total libros: {total}")
        print(f"Disponibles: {disponibles}")
        print(f"Antiguos: {antiguos}")
        print(f"Promedio de popularidad: {popularidad_total / total if total > 0
else 0}")

```

