MOD 4 Lección 4: HERENCIA y POLIMORFISMO

EJERCICIO 1: Muy Básico - "Las Figuras Geométricas"

Contexto:

Crea un sistema simple para calcular áreas de figuras geométricas usando herencia básica.

Clase Padre: `Figura`

•`color` (string)
•`nombre` (string)

Métodos:

•`mostrar_info()`: muestra color y nombre
•`calcular_area()`: devuelve 0 (será sobrescrito)

Clases Hijas:

1.`Cuadrado` (hereda de Figura)

•`lado` (float)

Métodos:

•`calcular_area()`: devuelve lado × lado (sobrescribe método padre)

2.`Circulo` (hereda de Figura)

•`radio` (float)

Métodos:

•`calcular_area()`: devuelve π × radio^2 (sobrescribe método padre)

3.`Triangulo` (hereda de Figura)

•`base` (float)

•altura (float)

Métodos:

•calcular_area(): devuelve (base × altura) / 2 (sobrescribe método padre)

Requisitos:

•Todas las clases hijas deben sobrescribir calcular_area()
•Usar super() en los constructores de las hijas
•Crear una función que acepte cualquier Figura y muestre su área (polimorfismo)

Salida esperada:

- Figura: Cuadrado Rojo - Lado: 5
  Área: 25.0 unidades²

- Figura: Círculo Amarillo - Radio: 3
  Área: 28.27 unidades²

- Figura: Triángulo Verde - Base: 6, Altura: 4
  Área: 12.0 unidades²

EJERCICIO 2: Intermedio - "Sistema de Medios Digitales"

Contexto:

Una plataforma digital gestiona diferentes tipos de contenido multimedia.

Clase Padre: ContenidoDigital

•titulo (string)
•autor (string)
•fecha_publicacion (string)
•duracion (int) - minutos
•calificacion (float) - 0.0 a 5.0

Métodos:

•reproducir(): imprime "Reproduciendo [titulo]"
•calificar(nueva_calificacion): actualiza calificación
•es_largo(): devuelve True si dura > 30 minutos

Clases Hijas:

1. Video (hereda de ContenidoDigital)

•formato (string) - "mp4", "avi", "mov"
•resolucion (string) - "HD", "Full HD", "4K"
•subir_calidad(): aumenta la resolución si es posible
•agregar_subtitulos(idioma): método propio

2. Podcast (hereda de ContenidoDigital)

•invitados (lista de strings)
•tema (string)
•agregar_invitado(nombre): agrega un invitado
•generar_transcripcion(): método propio

3. LibroDigital (hereda de ContenidoDigital)

•num_paginas (int)
•formato_lectura (string) - "epub", "pdf", "mobi"

•marcar_pagina(pagina): marca una página
•calcular_tiempo_lectura(): devuelve num_paginas / 2 (minutos)

Clases NUEVAS que heredan de las hijas:

4.Tutorial (hereda de Video)

•dificultad (string) - "principiante", "intermedio", "avanzado"
•materiales_necesarios (lista)
•agregar_material(material): método propio
5.Entrevista (hereda de Podcast)

•entrevistado (string)
•empresa (string)
•generar_resumen(): método propio

Requisitos:

•Herencia multinivel: Tutorial → Video → ContenidoDigital
•Sobrescribir reproducir() en al menos 2 clases hijas
•Crear método obtener_tipo() que devuelva el tipo específico
•Lista que contenga diferentes tipos de contenido (polimorfismo)

Salida esperada:

CONTENIDO: Aprende Python en 30 días (Tutorial)
  Autor: Carlos Méndez | Duración: 45 min | Calificación: 4.8
  Formato: mp4 | Resolución: Full HD | Dificultad: principiante
  Materiales: Python, editor de código

CONTENIDO: Historia de la IA (Podcast)
  Autor: Tech Radio | Duración: 60 min | Calificación: 4.5
  Tema: Tecnología | Invitados: Dr. Smith, Dra. Johnson

CONTENIDO: POO para todos (Libro Digital)
  Autor: Ana Torres | Páginas: 300 | Formato: epub
  Tiempo lectura estimado: 150 minutos

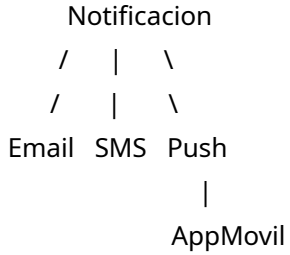
ESTADÍSTICAS:
• Total contenidos: 8
• Promedio calificación: 4.6
• Contenidos largos (>30 min): 6

EJERCICIO INTERMEDIO: "Sistema de Notificaciones"

Contexto:

Una aplicación necesita enviar diferentes tipos de notificaciones a usuarios, cada una con comportamiento distinto.

Jerarquía de Herencia:



Clase Padre: `Notificacion`

•`destinatario` (string)
•`mensaje` (string)
•`fecha_envio` (string)
•`prioridad` (string) - "baja", "media", "alta"

Métodos:

•`enviar()`: imprime "Enviando notificación básica"
•`validar()`: devuelve True si destinatario no está vacío
•`obtener_tipo()`: devuelve "Notificación genérica"
•`formatear_mensaje()`: devuelve el mensaje en mayúsculas

Clases Hijas:

1. Email (hereda de Notificacion)

•asunto (string)
•remitente (string)
•adjuntos (lista de strings)

Métodos:

•enviar(): sobrescribe - imprime "📧 Enviando EMAIL a [destinatario]"
•obtener_tipo(): sobrescribe - devuelve "Email"
•agregar_adjunto(archivo): método propio
•formatear_mensaje(): sobrescribe - añade "Estimado/a," al inicio

2. SMS (hereda de Notificacion)

•numero_telefono (string)
•operadora (string) - "Claro", "Movistar", "Entel"

Métodos:

•enviar(): sobrescribe - imprime "📱 Enviando SMS al [numero_telefono]"
•obtener_tipo(): sobrescribe - devuelve "SMS"
•validar(): sobrescribe - valida que el número tenga 8 dígitos
•acortar_mensaje(): si mensaje > 160 chars, lo corta

3. Push (hereda de Notificacion)

•dispositivo (string) - "iOS", "Android", "Web"
•icono (string) - nombre del icono

Métodos:

•enviar(): sobrescribe - imprime "🔔 Enviando PUSH a dispositivo [dispositivo]"
•obtener_tipo(): sobrescribe - devuelve "Notificación Push"
•mostrar_en_pantalla(): método propio
•formatear_mensaje(): sobrescribe - añade emoji según prioridad

Clase EXTRA: AppMovil (hereda de Push)

•aplicacion (string) - "WhatsApp", "Telegram", "Facebook"
•chat_id (string)

Métodos:

•enviar(): sobrescribe - imprime "□□ Enviando por [aplicacion] a [chat_id]"
•obtener_tipo(): sobrescribe - devuelve "Mensaje App Móvil"
•verificar_conexion(): método propio
•formatear_mensaje(): sobrescribe - añade formato de la app

REQUISITOS CON isinstance():

Parte 1: Filtrar notificaciones

Crear función filtrar_por_tipo(lista_notificaciones, tipo) que:

•Use isinstance() para identificar el tipo
•Devuelva solo las notificaciones del tipo especificado

Parte 2: Estadísticas

Función generar_estadisticas(lista_notificaciones) que:

1.Cuente cuántas notificaciones de cada tipo hay usando isinstance()
2.Separe por prioridad usando condicionales
3.Identifique las AppMovil específicamente (especial atención)

Parte 3: Procesamiento especial

Función procesar_notificaciones(lista_notificaciones) que:

•Si es Email: agregar firma automática
•Si es SMS: verificar límite de caracteres
•Si es Push pero NO AppMovil: añadir sonido

•Si es AppMovil: verificar conexión primero

Parte 4: Validación múltiple

Función validar_todas(lista_notificaciones) que:

•Use isinstance() para aplicar validación específica por tipo
•Email: validar que tenga @ en destinatario
•SMS: validar número de teléfono
•Push/AppMovil: validar que dispositivo no esté vacío

PROGRAMA DE PRUEBA:

python

```python
# Crear lista con diferentes tipos de notificaciones
notificaciones = [
    Email("ana@gmail.com", "Reunión importante", "jefe@empresa.com"),
    SMS("71234567", "Tu código es 1234", "Claro"),
    Push("usuario123", "Nueva actualización", "Android"),
    AppMovil("usuario456", "Hola ¿cómo estás?", "Android", "WhatsApp", "chat123"),
    Email("pepe@hotmail.com", "Factura pendiente", "contabilidad@empresa.com"),
    SMS("69876543", "Oferta especial hoy", "Entel")
]

# Demostrar isinstance()
print("🔍 USANDO isinstance():")

for i, notif in enumerate(notificaciones, 1):
    print(f"\n{i}. {notif.obtener_tipo()}:")

    if isinstance(notif, Email):
        print(f"   Es Email: ✓")
        print(f"   Remitente: {notif.remitente}")

    elif isinstance(notif, SMS):
        print(f"   Es SMS: ✓")
        print(f"   Operadora: {notif.operadora}")

    elif isinstance(notif, AppMovil):  # ¡IMPORTANTE! Verificar primero
```

```
        print(f"   Es AppMovil (y también Push): ✓")
        print(f"   Aplicación: {notif.aplicacion}")

    elif isinstance(notif, Push):
        print(f"   Es Push (pero NO AppMovil): ✓")
        print(f"   Dispositivo: {notif.dispositivo}")

    else:
        print(f"   Tipo desconocido")

# Filtrar usando isinstance()
emails = [n for n in notificaciones if isinstance(n, Email)]
print(f"\n Total emails: {len(emails)}")

# Verificar herencia
app1 = AppMovil("test", "msg", "iOS", "Telegram", "chat1")
print(f"\n ¿AppMovil ES Push? {isinstance(app1, Push)}")  # True
print(f" ¿Push ES AppMovil? {isinstance(Push('a','b','c'), AppMovil)}")  # False
```

SALIDA ESPERADA:

 USANDO isinstance():

1. Email:
   Es Email: ✓
   Remitente: jefe@empresa.com

2. SMS:
   Es SMS: ✓
   Operadora: Claro

3. Notificación Push:
   Es Push (pero NO AppMovil): ✓
   Dispositivo: Android

4. Mensaje App Móvil:
   Es AppMovil (y también Push): ✓
   Aplicación: WhatsApp

5. Email:

Es Email: ✓
Remitente: contabilidad@empresa.com

6. SMS:
Es SMS: ✓
Operadora: Entel

 Total emails: 2

離 ¿AppMovil ES Push? True
離 ¿Push ES AppMovil? False

 ESTADÍSTICAS:
- Total notificaciones: 6
- Emails: 2
- SMS: 2
- Push: 1
- AppMovil: 1
- Prioridad alta: 0 | media: 3 | baja: 3

 VALIDACIÓN:
- 4/6 notificaciones válidas
- 2 SMS con número inválido