

# **ANALISIS Y DISEÑO – LAB 4,**

## **Polimorfismo**

POO – sección 50

César Espinoza 25059

Juan Pablo Román 25627

### **1. Propósito de las clases**

#### **Modelo:**

Clase	Propósito
<b>Contenido (abstracta)</b>	Clase base que representa cualquier tipo de contenido del sistema. Contiene los atributos y métodos comunes como id, titulo, autor, fechaCreacion, categoria, y métodos como publicar(), editar(), mostrarDetalles(). Evita repetir código en las subclases.
<b>Artículo</b>	Hereda de Contenido. Representa publicaciones escritas (artículos académicos o informativos). Sobrescribe los métodos para mostrar el texto del artículo y definir cómo se publica.
<b>Video</b>	Hereda de Contenido. Representa material audiovisual (videos del estudio). Incluye atributos como url, duracion, resolucion.
<b>Imagen</b>	Hereda de Contenido. Representa imágenes o fotografías. Contiene atributos como url, formato, resolucion.
<b>Usuario (abstracta)</b>	Clase base para representar a cualquier persona que usa el CMS. Contiene atributos comunes como nombre, correo, rol. Define métodos generales como iniciarSesion().
<b>Administrador</b>	Hereda de Usuario. Tiene permisos para crear, editar, publicar y eliminar contenidos. Controla el flujo general del sistema.
<b>Editor</b>	Hereda de Usuario. Puede crear y editar contenidos, pero no publicarlos ni eliminarlos. Está limitado en permisos.

**Controlador:**

Clase	Propósito
<b>ContenidoController</b>	Controla las operaciones sobre los contenidos (crear, editar, eliminar, filtrar, buscar). Aplica polimorfismo para manejar diferentes tipos de contenido (Artículo, Video, Imagen) a través de la clase base Contenido.
<b>UsuarioController</b>	Gestiona los usuarios del sistema. Permite iniciar sesión, registrar usuarios y asignar roles (Administrador o Editor).

**Vista:**

Clase	Propósito
<b>VistaCMS</b>	Representa la interfaz gráfica o de consola. Muestra menús o botones para que el usuario cree, edite, elimine o busque contenidos. También permite generar reportes.

## 2. Organizacion de la jerarquía

**a. Modelo (Model):**

- Contiene todas las clases que representan la lógica y los datos del sistema.
- Contenido es una clase abstracta que sirve como base para los diferentes tipos de contenido (Articulo, Video, Imagen).
- Usuario también es abstracta y da origen a las subclases Administrador y Editor, que definen distintos niveles de permisos.

**b. Controlador (Controller):**

- Contiene las clases encargadas de manejar la comunicación entre el modelo y la vista.
- ContenidoController administra la creación, edición, eliminación, filtrado y búsqueda de los contenidos.
- UsuarioController se encarga de la gestión de usuarios y de la asignación de roles.

**c. Vista (View):**

- Presenta la interfaz gráfica o consola con la que interactúa el usuario final.
- VistaCMS permite visualizar, crear y gestionar los contenidos del sistema.

**d. Relaciones y jerarquía**

- Contenido es superclase abstracta de Articulo, Video e Imagen.
- Usuario es superclase abstracta de Administrador y Editor.

- Los controladores (ContenidoController y UsuarioController) tienen asociaciones con las clases del modelo para ejecutar acciones sobre los datos.
- La vista (VistaCMS) interactúa únicamente con los controladores, nunca directamente con el modelo, cumpliendo el principio de baja acoplamiento y alta cohesión del patrón MVC.

### 3. Propósito de las interfaces

- a. El uso de interfaces en este sistema permite aplicar el principio de polimorfismo y mantener el diseño extensible y flexible. De esta forma, nuevas clases pueden integrarse al sistema sin modificar las existentes, cumpliendo con el principio “Open/Closed” de la programación orientada a objetos.
- b. Interfaz propuesta: **IReportable**
  - Propósito general:**
    - La interfaz IReportable define el contrato para todas las clases que necesiten generar reportes o resúmenes dentro del CMS. Cualquier clase que implemente esta interfaz deberá definir su propia forma de generar dichos reportes.
  - Métodos principales:**
    - generarReporte(): genera un报告 o resumen del contenido o actividad del usuario.
  - Clases que pueden implementarla:**
    - ContenidoController: para generar reportes generales de todos los contenidos publicados.
    - Administrador: para generar reportes de usuarios, publicaciones y estadísticas.
    - Contenido: para generar reportes específicos de cada tipo de contenido (por ejemplo, artículos o videos).

