

## Práctica: Gestión de un Videoclub

En esta práctica, vamos a implementar una estructura de clases que simula la gestión de un videoclub. El objetivo es aplicar los conceptos de **herencia**, **composición**, **métodos**, y **polimorfismo** en un entorno práctico.

---

### Descripción de las clases y relaciones

#### 1. Clase Soporte

Esta clase es la base para todos los tipos de productos que puede ofrecer el videoclub, como cintas de video, DVDs y videojuegos.

- **Atributos:**
  - titulo: Título del soporte.
  - numero: Número identificador del soporte.
  - precio: Precio base del soporte.
- **Métodos:**
  - getPrecio(): Devuelve el precio del soporte.
  - getPrecioConIva(): Devuelve el precio con IVA incluido.
  - getNumero(): Devuelve el número identificador.
  - muestraResumen(): Muestra un resumen con la información básica del soporte.

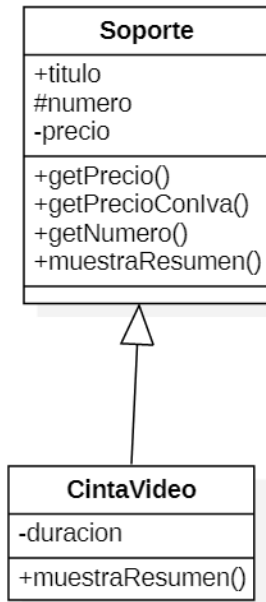
Soporte
+titulo #numero -precio
+getPrecio() +getPrecioConIva() +getNumero() +muestraResumen()

---

#### 2. Clase CintaVideo

Hereda de Soporte y añade un atributo específico para almacenar la duración de la cinta.

- **Atributos adicionales:**
  - duracion: Duración de la cinta en minutos.
- **Métodos:**
  - Sobrescribe muestraResumen() para incluir la duración.

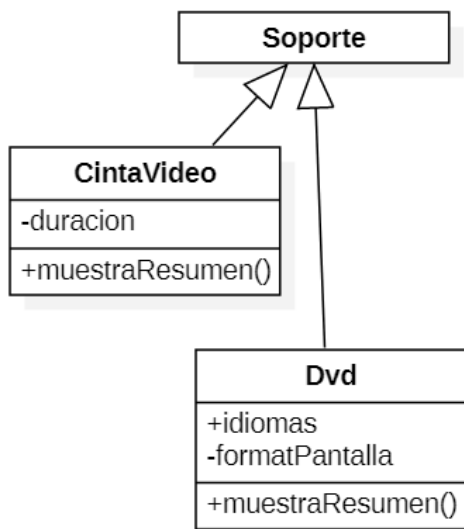


---

### 3. Clase Dvd

Hereda de Soporte y añade información específica para DVDs.

- **Atributos adicionales:**
  - idiomas: Idiomas disponibles en el DVD.
  - formatPantalla: Formato de pantalla (ej.: widescreen, estándar).
- **Métodos:**
  - Sobrescribe `muestraResumen()` para incluir los idiomas y formato de pantalla.



#### 4. Clase Juego

Hereda de Soporte y añade características específicas para videojuegos.

- **Atributos adicionales:**
  - `consola`: Consola para la que está disponible el juego.
  - `minNumJugadores`: Número mínimo de jugadores.
  - `maxNumJugadores`: Número máximo de jugadores.
- **Métodos:**
  - `muestraJugadoresPosibles()`: Muestra el rango de jugadores permitido.
  - Sobrescribe `muestraResumen()` para incluir la consola y



## 5. Clase Cliente

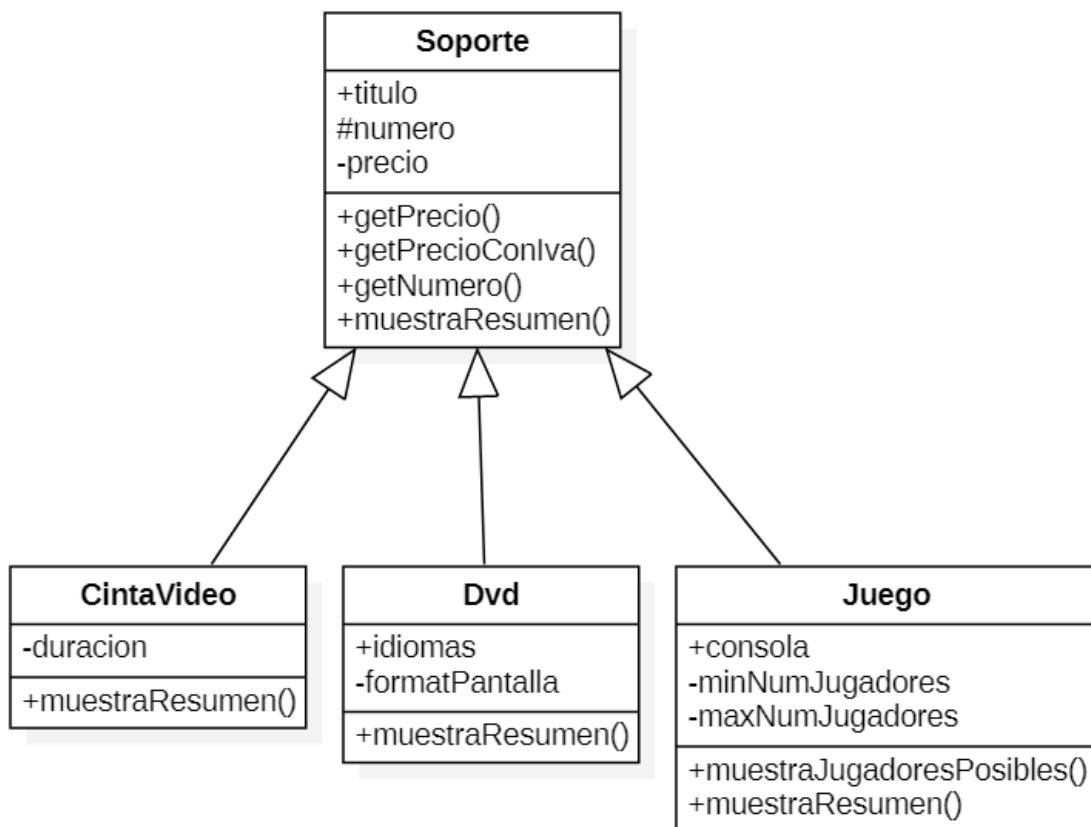
Representa a los clientes del videoclub, que pueden alquilar varios soportes con un límite establecido.

- **Atributos:**

- nombre: Nombre del cliente.
- numero: Número identificador del cliente.
- soportesAlquilados[]: Lista de soportes alquilados.
- numSoportesAlquilados: Número actual de soportes alquilados.
- maxAlquilerConcurrente: Máximo número de soportes que puede alquilar al mismo tiempo.

- **Métodos:**

- alquilar(soporte: Soporte): Permite alquilar un soporte.
- tieneAlquilado(soporte: Soporte): bool: Indica si el cliente ya tiene alquilado el soporte.
- devolver(numSoporte: int): Devuelve un soporte.
- listaAlquileres(): Lista los soportes alquilados.
- muestraResumen(): Muestra información del cliente.



Cliente
+nombre -numero -soportesAlquilados[] -numSoportesAlquilados -maxAlquilerConcurrente

---

## 6. Clase Videoclub

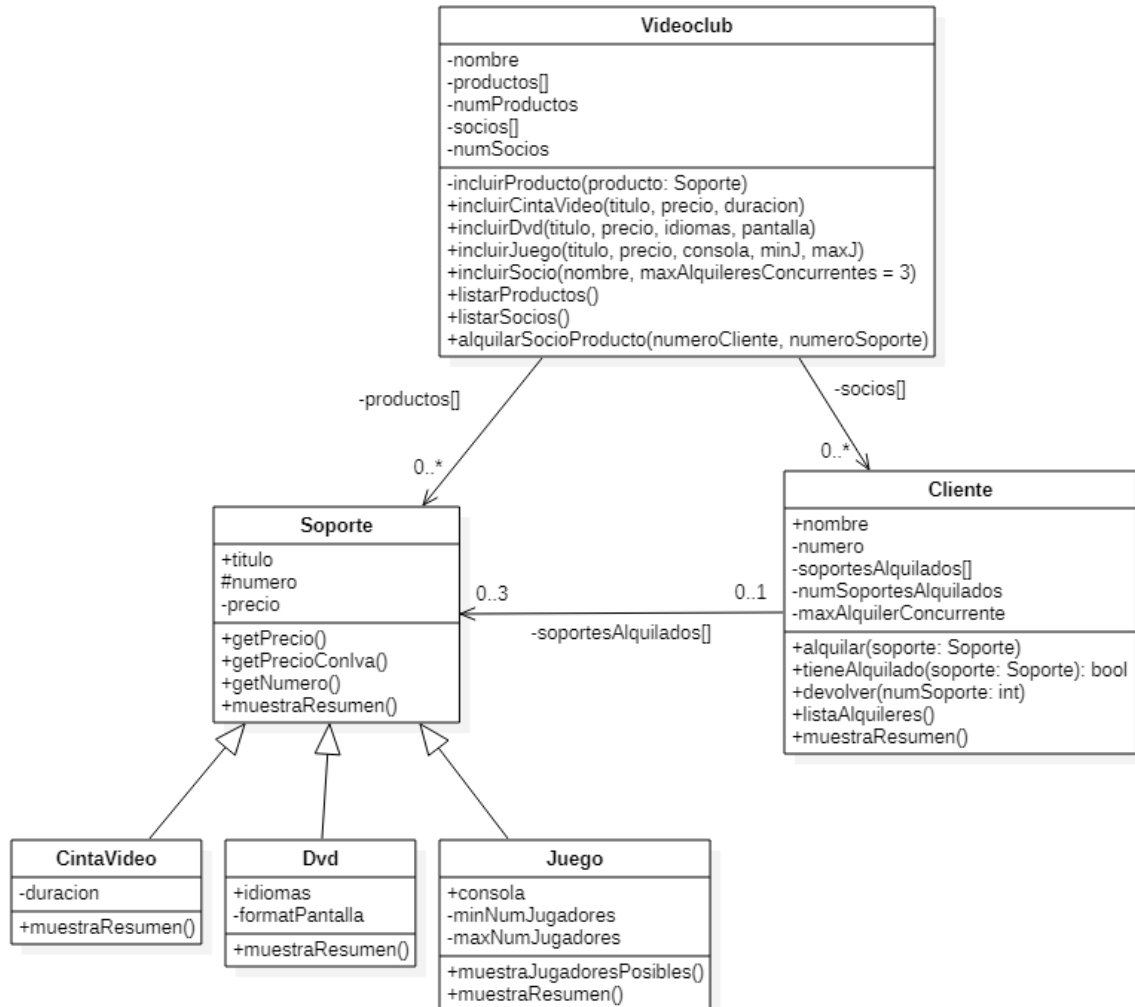
Esta clase gestiona todos los aspectos del videoclub, como los productos disponibles y los socios registrados.

- **Atributos:**

- nombre: Nombre del videoclub.
- productos[]: Lista de productos disponibles.
- numProductos: Número total de productos.
- socios[]: Lista de socios registrados.
- numSocios: Número total de socios.

- **Métodos:**

- incluirProducto(producto: Soporte): Añade un producto a la lista.
- incluirCintaVideo(titulo, precio, duracion): Añade una cinta de video.
- incluirDvd(titulo, precio, idiomas, pantalla): Añade un DVD.
- incluirJuego(titulo, precio, consola, minJ, maxJ): Añade un videojuego.
- incluirSocio(nombre, maxAlquileresConcurrentes = 3): Añade un socio.
- listarProductos(): Lista todos los productos.
- listarSocios(): Lista todos los socios.
- alquilarSocioProducto(numeroCliente, numeroSoporte): Permite a un cliente alquilar un producto.



## Objetivos de la práctica

1. Implementar las clases con los atributos y métodos indicados.
2. Aplicar la **herencia** y el **polimorfismo** para sobrescribir y extender métodos.
3. Gestionar la lógica de alquileres entre los clientes y el videoclub.
4. Practicar el uso de arreglos (array) para almacenar productos y soportes.

## Tareas específicas

1. Crea las clases siguiendo los diagramas proporcionados.
2. Implementa las relaciones de herencia entre las clases.
3. Sobrescribe el método `muestraResumen()` en cada clase derivada.

4. Añade la lógica necesaria para que un cliente pueda alquilar y devolver soportes en el videoclub.
5. Realiza pruebas creando instancias de las clases y mostrando resúmenes de cada una.