

Desarrollo Web en Entorno Servidor

4.- PHP Objetos / Prácticas

IES Severo Ochoa



Ejercicio 1

- 401Empleado.php: Crea una clase Empleado con su nombre, apellidos y sueldo. Encapsula las propiedades mediante getters/setters y añade métodos para:
 - Imprimir su nombre completo
 - `getNombreCompleto() : string`
 - Que devuelva un booleano indicando si debe o no pagar impuestos (se pagan cuando el sueldo es superior a 3333€)
 - `debePagarImpuestos() : bool`

Ejercicio 2

- 402EmpleadoT.php: Copia la clase del ejercicio anterior y modifícala.
- Añade una propiedad privada que almacene un array de números de telefonos.
- Añade los siguientes métodos:
 - `public function anyadirTelefono(int $telefono) : void`
 - Añade un teléfono al array
 - `public function listarTelefonos() : string`
 - Muestra los teléfonos separados por comas
 - `public function vaciarTelefonos() : void`
 - Elimina todos los teléfonos

Ejercicio 3

- `403EmpleadoC.php`: Copia la clase del ejercicio anterior y modifícala.
- Elimina los setters de nombre y apellidos, de manera que dichos datos se asignan mediante el constructor.
- Si el constructor recibe un tercer parámetro, será el sueldo del Empleado. Si no, se le asignarán 1000€.

Ejercicio 4

- `404EmpleadoC.php`: Copia la clase del ejercicio anterior y modifícala.
- Añade una constante `SUELDO_TOPE` con el valor del sueldo que debe pagar impuestos, y modifica el código para utilizar la constante

Ejercicio 5

- `405Empleados.php`: Copia la clase del ejercicio anterior y modifícala.
- Cambia la constante por una variable estática `sueldoTope`, de manera que mediante *getter/setter* puedas modificar su valor.

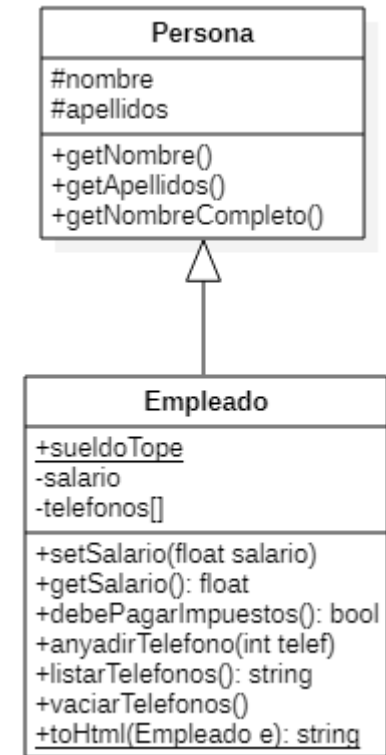
Ejercicio 6

- 406EmpleadoE.php: Copia la clase del ejercicio anterior y modifícala.
- Completa el siguiente método con una cadena HTML que muestre los datos de un empleado y muestre todos los teléfonos mediante una lista ordenada:

```
public static function  
toHtml(Empleado $emp): string
```

Ejercicio 7

- 407Persona.php: Copia la clase del ejercicio anterior y modifícala.
- Crea una clase Persona que sea padre de Empleado, de manera que Persona contenga el nombre y los apellidos, y en Empleado quede el salario y los teléfonos.



Ejercicio 8

- `408PersonaH.php`: Copia las clases del ejercicio anterior y modifícalas.
- Crea en `Persona` el método estático `toHtml(Persona $p)`
- Modifica en `Empleado` el mismo método `toHtml()`, pero cambia la firma para que reciba una `Persona` como parámetro.

Ejercicio 9

- `409PersonaE.php`: Copia las clases del ejercicio anterior y modifícalas.
- Añade en `Persona` un atributo `edad`
- A la hora de saber si un empleado debe pagar impuestos, lo hará siempre y cuando tenga más de 21 años.
- Modifica todo el código necesario para mostrar y/o editar la edad cuando sea necesario.

Ejercicio 10

- `410PersonaS.php`: Copia las clases del ejercicio anterior y modifícalas.
- Añade nuevos métodos que hagan una representación de todas las propiedades de las clases `Persona` y `Empleado`, de forma similar a los realizados en HTML, pero sin que sean estáticos.

```
function public toString(): string
```

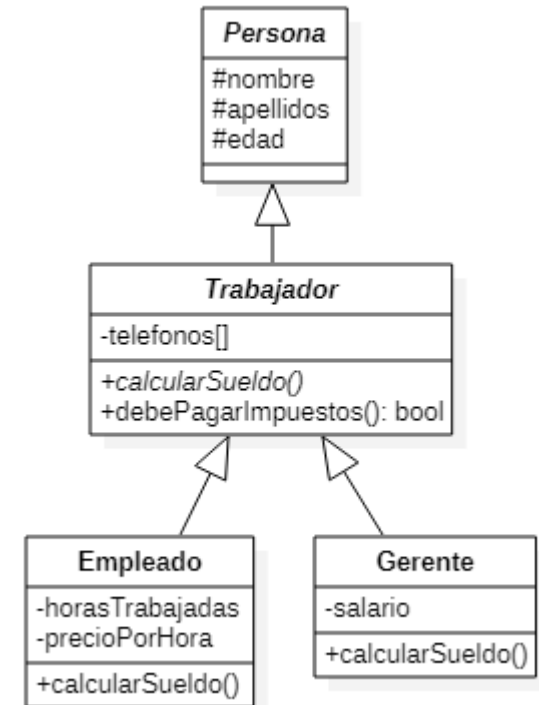
De manera que obtenga los datos mediante `this`

Ejercicio 11

- `411PersonaA.php`: Copia las clases del ejercicio anterior y modifícalas.
- Transforma `Persona` a una clase abstracta donde su método `toString()` tenga que ser redefinido en todos sus hijos.

Ejercicio 12

- 412Trabajador.php: Copia las clases del ejercicio anterior y modifícalas.
- Cambia la estructura de clases conforme al gráfico respetando todos los métodos que ya están hechos.
- Trabajador es una clase abstracta donde calcularSueldo es un método abstracto
 - El sueldo de un Empleado se calcula a partir de las horas trabajadas y lo que cobra por hora
 - Para los Gerentes, su sueldo se incrementa porcentualmente en base a su edad: $\text{salario} + \text{salario} * \text{edad} / 100$



Ejercicio 13

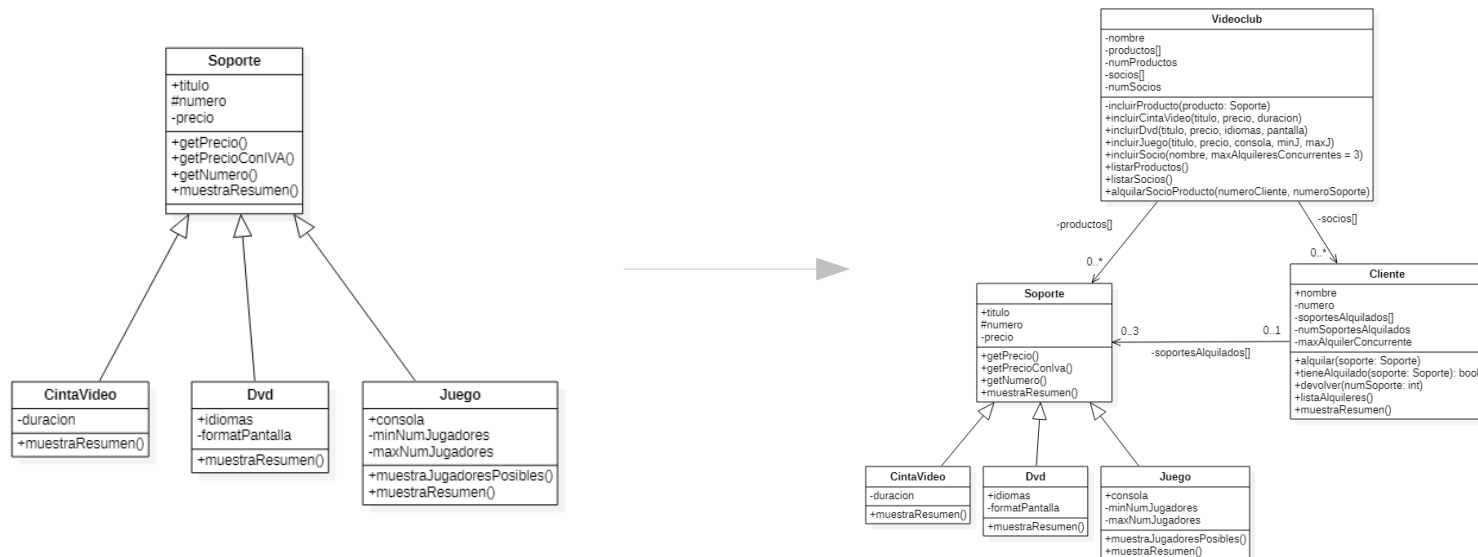
- `413Empresa.php`: Copia las clases del ejercicio anterior y modifícalas.
- Crea una clase `Empresa` que además del `nombre` y la `dirección`, contenga una propiedad con un array de `Trabajadores`, ya sean `Empleados` o `Gerentes`.
- Añade *getters/setters* para el `nombre` y `dirección`.
- Añade métodos para añadir y listar los trabajadores.
 - `public function anyadirTrabajador(Trabajador $t)`
 - `public function listarTrabajadoresHtml() : string`
- Añade un método para obtener el coste total en nóminas.
 - `public function getCosteNominas() : float`

Ejercicio 14

- 414EmpresaI.php: Copia las clases del ejercicio anterior y modifícalas.
- Crea un interfaz `JSerializable`, de manera que ofrezca los métodos:
 - `toJSON(): string` // utiliza `json_encode(mixed)`
 - `toSerialize(): string` // utiliza `serialize(mixed)`
- Haz que todas las clases que no son abstractas implementen el interfaz

Proyecto videoclub

- En los siguientes ejercicios vamos a simular un pequeño proyecto de un Videoclub.
- Desarrollo incremental
- Cada clase en un archivo php
 - Se muestra la estructura y la salida por pantalla



Ejercicio 420

- Crear una clase Soporte (Soporte.php)
 - Crear constructor
 - Definir una constante IVA = 1.16
- Crear un archivo para usar las clases: videoclub.php

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

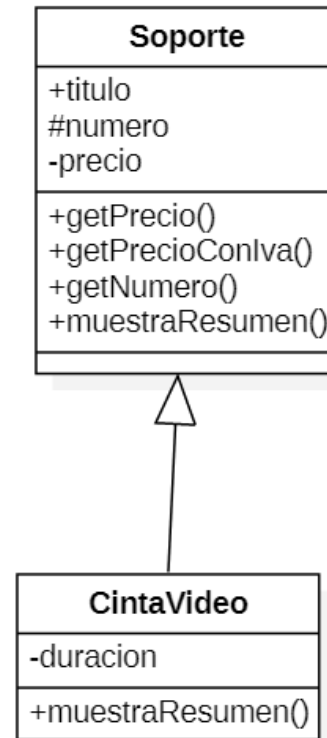
```
<?php
include "Soporte.php";

$soporte1 = new Soporte("Tenet", 22, 3);
echo "<strong>" . $soporte1->titulo . "</strong>";
echo "<br>Precio: " . $soporte1->getPrecio() . " euros";
echo "<br>Precio IVA incluido: " . $soporte1->getPrecioConIVA() . " euros";
$soporte1->muestraResumen();
```

```
Tenet
Precio: 3 euros
Precio IVA incluido: 3.48 euros
Tenet
3 (IVA no incluido)
```

Ejercicio 421

- Crear la clase `CintaVideo`
 - Hereda de `Soporte`
 - Añadir atributo `duracion`
- Sobreescibir
 - `constructor`
 - `muestraResumen`



```
include "CintaVideo.php";
```

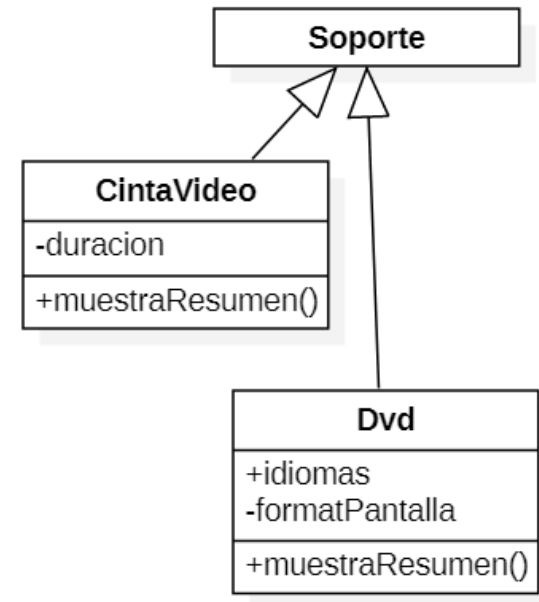
```
$miCinta = new CintaVideo("Los cazafantasmas", 23, 3.5, 107);
echo "<strong>" . $miCinta->titulo . "</strong>";
echo "<br>Precio: " . $miCinta->getPrecio() . " euros";
echo "<br>Precio IVA incluido: " . $miCinta->getPrecioConIva() . " euros";
$miCinta->muestraResumen();
```

```

Los cazafantasmas
Precio: 3.5 euros
Precio IVA incluido: 4.06 euros
Película en VHS:
Los cazafantasmas
3.5 (IVA no incluido)
Duración: 107 minutos
  
```

Ejercicio 422

- Crear la clase Dvd
 - Hereda de Soporte
 - Añadir atributo idiomas, formatoPantalla
- Sobrecribir
 - constructor
 - muestraResumen



```

include "Dvd.php";
$miDvd = new Dvd("Origen", 24, 15, "es,en,fr", "16:9");
echo "<strong>" . $miDvd->titulo . "</strong>";
echo "<br>Precio: " . $miDvd->getPrecio() . " euros";
echo "<br>Precio IVA incluido: " . $miDvd->getPrecioConIva() . " euros";
$miDvd->muestraResumen();
  
```

Origen
 Precio: 15 euros
 Precio IVA incluido: 17.4 euros
 Película en DVD:
 Origen
 15 (IVA no incluido)
 Idiomas: es,en,fr
 Formato Pantalla: 16:9

Ejercicio 423

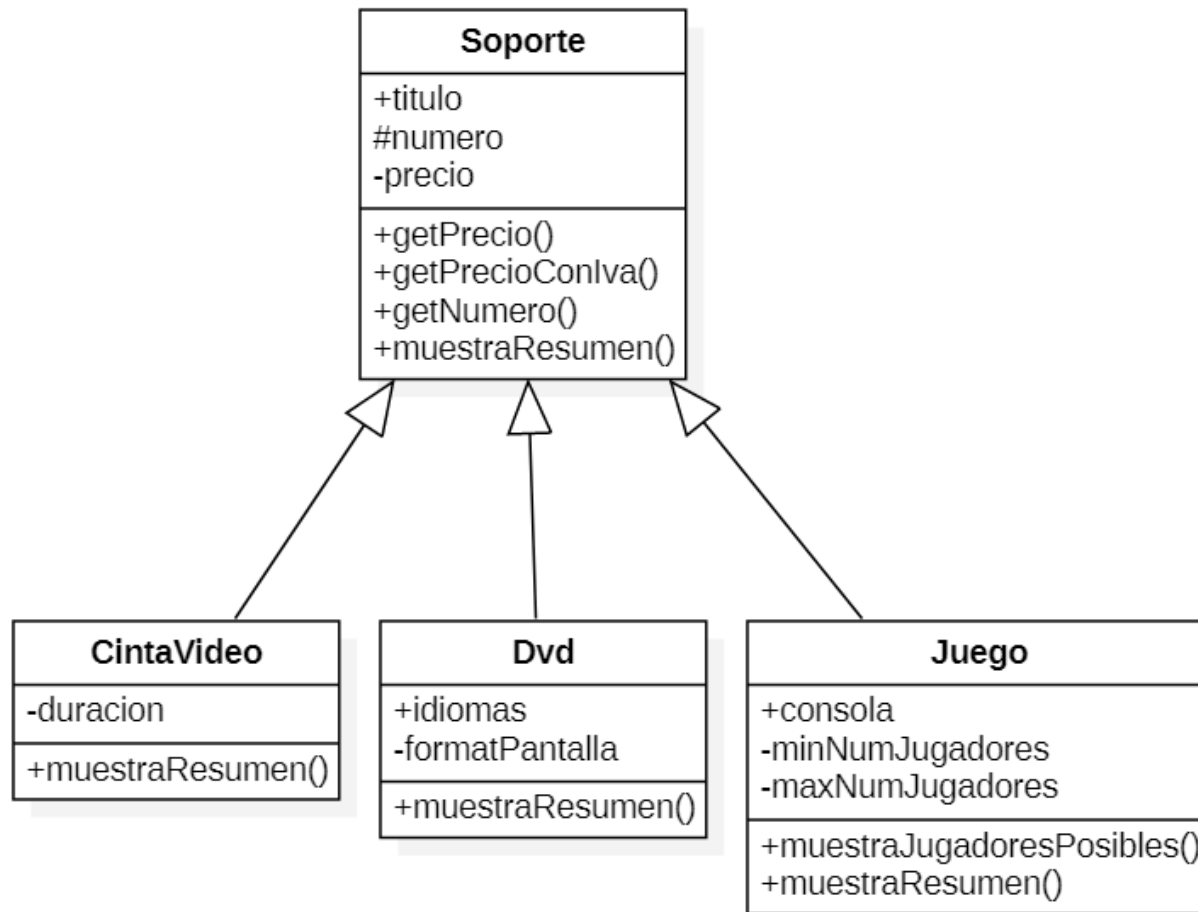


- Crear la clase Juego
 - Hereda de Soporte
 - Añadir atributo `consola`, `minNumJugadores` y `maxNumJugadores`
 - Añadir método `muestraJugadoresPosibles`
 - Debe mostrar "Para un jugador", "Para X jugadores", "De X a Y jugadores"
- Sobreescribir constructor y `muestraResumen`

```
include "Juego.php";
$miJuego = new Juego("The Last of Us Part II", 26, 49.99, "PS4", 1, 1);
echo "<strong>" . $miJuego->titulo . "</strong>";
echo "<br>Precio: " . $miJuego->getPrecio() . " euros";
echo "<br>Precio IVA incluido: " . $miJuego->getPrecioConIva() . " euros";
$miJuego->muestraResumen();
```

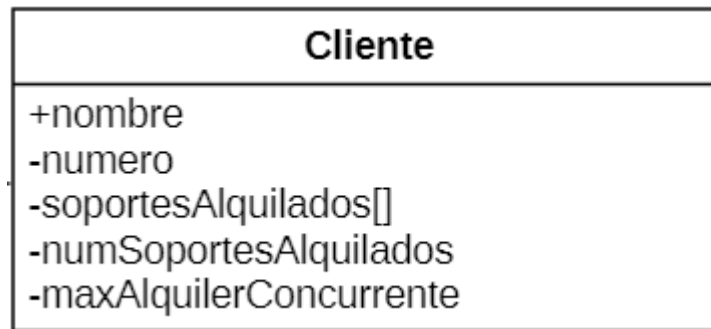
The Last of Us Part II
Precio: 49.99 euros
Precio IVA incluido: 57.9884 euros
Juego para: PS4
The Last of Us Part II
49.99 (IVA no incluido)
Para un jugador

Modelo actual



Ejercicio 424

- Crear la clase `Cliente`
- El constructor recibirá `nombre`, `numero` y `maxAlquilerConcurrente`, este último pudiendo ser opcional y tomando como valor por defecto 3.
- Añadir getter/setter únicamente a `numero`.
- Añade operación `muestraResumen` que muestre el nombre y la cantidad de alquileres.



Ejercicio 425

- Dentro de `Cliente`, añade las siguientes operaciones:
- `tieneAlquilado(Soporte $s) : bool`
 - Recorre el array de soportes y comprueba si está el soporte
- `alquilar(Soporte $s) : bool`
 - Debe comprobar si el soporte está alquilado y si no ha superado el cupo de alquileres.
 - Al alquilar, incrementará el `numSoportesAlquilados` y almacenará el soporte en el array
 - Para cada caso debe mostrar un mensaje informado de lo ocurrido

Ejercicio 426

- Dentro de `Cliente`, añade las operaciones para:
- `devolver(int $numSoporte) : bool`
 - Debe comprobar que el soporte estaba alquilado y actualizar la cantidad de soportes alquilados.
 - Para cada caso debe mostrar un mensaje informado de lo ocurrido
- `listarAlquileres() : void`
 - Informa de cuantos alquileres tiene el cliente y los muestra

Probando I

```
<?php
include_once "CintaVideo.php";
include_once "Dvd.php";
include_once "Juego.php";
include_once "Cliente.php";

//instanciamos un par de objetos cliente
$cliente1 = new Cliente("Bruce Wayne", 23);
$cliente2 = new Cliente("Clark Kent", 33);

//mostramos el numero de cada cliente creado
echo "<br>El identificador del cliente 1 es: " . $cliente1-
>getNumero();
echo "<br>El identificador del cliente 2 es: " . $cliente2-
>getNumero();

//instancio algunos soportes
$soporte1 = new CintaVideo("Los cazafantasmas", 23, 3.5, 107);
$soporte2 = new Juego("The Last of Us Part II", 26, 49.99, "PS4", 1,
1);
$soporte3 = new Dvd("Origen", 24, 15, "es,en,fr", "16:9");
$soporte4 = new Dvd("El Imperio Contraataca", 4, 3, "es,en", "16:9");

//alquilo algunos soportes
$cliente1->alquilar($soporte1);
$cliente1->alquilar($soporte2);
$cliente1->alquilar($soporte3);
```

El identificador del cliente 1 es: 23

El identificador del cliente 2 es: 33

Alquilado soporte a: Bruce Wayne

Película en VHS:

Los cazafantasmas

3.5 (IVA no incluido)

Duración: 107 minutos

Alquilado soporte a: Bruce Wayne

Juego para: PS4

The Last of Us Part II

49.99 (IVA no incluido)

Para un jugador

Alquilado soporte a: Bruce Wayne

Película en DVD:

Origen

15 (IVA no incluido)

Idiomas:es,en,fr

Formato Pantalla:16:9

Probando II

El cliente ya tiene alquilado el soporte **Los cazafantasmas**

Este cliente tiene 3 elementos alquilados. No puede alquilar más en este videoclub hasta que no devuelva algo

No se ha podido encontrar el soporte en los alquileres de este cliente

No se ha podido encontrar el soporte en los alquileres de este cliente

Este cliente tiene 3 elementos alquilados. No puede alquilar más en este videoclub hasta que no devuelva algo

El cliente tiene 3 soportes alquilados

Película en VHS:

Los cazafantasmas

3.5 (IVA no incluido)

Duración: 107 minutos

Juego para: PS4

The Last of Us Part II

49.99 (IVA no incluido)

Para un jugador

Película en DVD:

Origen

15 (IVA no incluido)

Idiomas:es,en,fr

Formato Pantalla:16:9

Este cliente no tiene alquilado ningún elemento

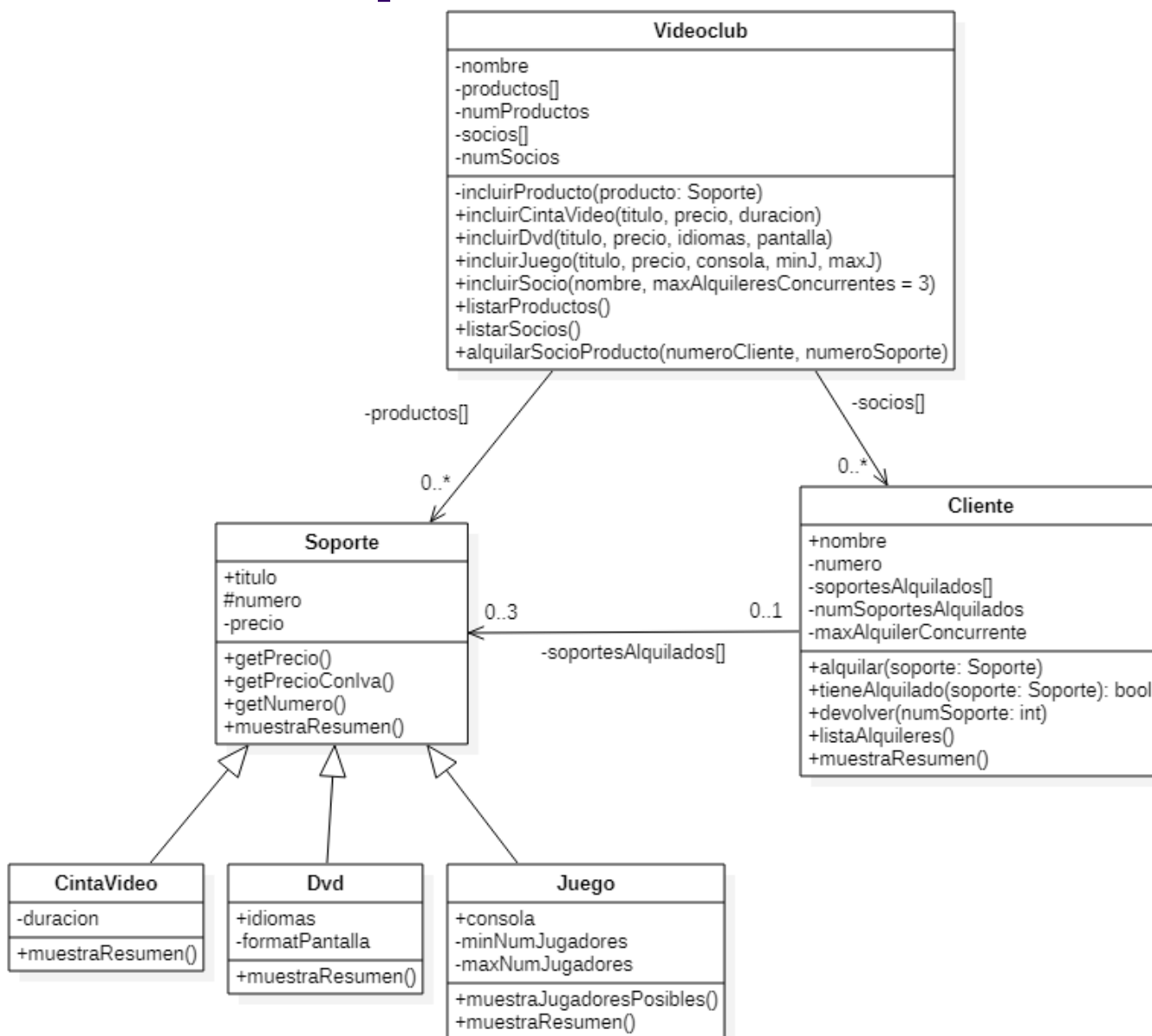
```
//  
voy a intentar alquilar de nuevo un soporte que ya tiene alquila  
do  
$cliente1->alquilar($soporte1);  
//el cliente tiene 3 soportes en alquiler como máximo  
//este soporte no lo va a poder alquilar  
$cliente1->alquilar($soporte4);  
//este soporte no lo tiene alquilado  
$cliente1->devolver(4);  
//devuelvo un soporte que sí que tiene alquilado  
$cliente1->devolver(2);  
//alquilo otro soporte  
$cliente1->alquilar($soporte4);  
//listo los elementos alquilados  
$cliente1->listaAlquileres();  
//este cliente no tiene alquileres  
$cliente2->devolver(2);
```

Ejercicio 427

- Crear la clase Videoclub
 - productos es un array de Soportes
 - socios es una array de Clientes
 - Los métodos públicos de incluir algún soporte, crearán la clase y llamarán al método privado
 - Lo inserta en el array

Videoclub
-nombre -productos[] -numProductos -socios[] -numSocios
-incluirProducto(producto: Soporte) +incluirCintaVideo(titulo, precio, duracion) +incluirDvd(titulo, precio, idiomas, pantalla) +incluirJuego(titulo, precio, consola, minJ, maxJ) +incluirSocio(nombre, maxAlquileresConcurrentes = 3) +listarProductos() +listarSocios() +alquilarSocioProducto(numeroCliente, numeroSoporte)

Modelo completo



Probando I

```
<?php
include_once "Videoclub.php";

$vc = new Videoclub("Video Gaga");

//voy a incluir unos cuantos soportes de prueba
$vc->incluirJuego("God of War", 19.99, "PS4", 1, 1);
$vc->incluirJuego("The Last of Us Part II", 49.99, "PS4", 1,
$vc->incluirDvd("Torrente", 4.5, "es", "16:9");
$vc->incluirDvd("Origen", 4.5, "es,en,fr", "16:9");
$vc->incluirDvd("El Imperio Contraataca", 3, "es,en", "16:9");
$vc->incluirCintaVideo("Los cazafantasmas", 3.5, 107);
$vc->incluirCintaVideo("El nombre de la Rosa", 1.5, 140);

//listo los productos
$vc->listarProductos();
```

Incluido soporte 0
Incluido soporte 1
Incluido soporte 2
Incluido soporte 3
Incluido soporte 4
Incluido soporte 5
Incluido soporte 6

Listado de los 7 productos disponibles:

- 1.-
Juego para: PS4
God of War
19.99 (IVA no incluido)
Para un jugador
- 2.-
Juego para: PS4
The Last of Us Part II
49.99 (IVA no incluido)
Para un jugador
- 3.-
Película en DVD:
Torrente
4.5 (IVA no incluido)
Idiomas:es
Formato Pantalla:16:9
- 4.-
Película en DVD:
Origen
4.5 (IVA no incluido)
Idiomas:es,en,fr
Formato Pantalla:16:9
- 5.-
Película en DVD:
El Imperio Contraataca
3 (IVA no incluido)
Idiomas:es,en
Formato Pantalla:16:9
- 6.-
Película en VHS:
Los cazafantasmas
3.5 (IVA no incluido)
Duración: 107 minutos
- 7.-
Película en VHS:
El nombre de la Rosa
1.5 (IVA no incluido)
Duración: 140 minutos

Probando II

Incluido socio 0
Incluido socio 1

Alquilado soporte a: Pablo Picasso

Película en DVD:
Torrente
4.5 (IVA no incluido)
Idiomas:es
Formato Pantalla:16:9

Alquilado soporte a: Pablo Picasso

Película en DVD:
Origen
4.5 (IVA no incluido)
Idiomas:es,en,fr
Formato Pantalla:16:9

El cliente ya tiene alquilado el soporte **Torrente**

Este cliente tiene 2 elementos alquilados. No puede alquilar más en este videoclub hasta que no devuelva algo

Listado de 2 socios del videoclub:
1.-

Cliente 0: Amancio Ortega
Alquileres actuales: 0
2.-

Cliente 1: Pablo Picasso
Alquileres actuales: 2

```
// creamos algunos socios
$vc->incluirSocio("Amancio Ortega");
$vc->incluirSocio("Pablo Picasso", 2);

$vc->alquilaSocioProducto(1, 2);
$vc->alquilaSocioProducto(1, 3);
// alquilo otra vez el soporte 2 al socio 1.
// no debe dejarme porque ya lo tiene alquilado
$vc->alquilaSocioProducto(1, 2);
// alquilo el soporte 6 al socio 1.
//
no se puede porque el socio 1 tiene 2 alquileres como máximo
$vc->alquilaSocioProducto(1, 6);

// listo los socios
$vc->listarSocios();
```

Ejercicio 428

- Pasar `Soporte` a una clase abstracta.
- Crea un interfaz `Resumible`, de manera que las clases que lo implementen deben ofrecer el método `muestraResumen()`

¿Alguna pregunta?