

# Proyecto 2º evaluación

1º de grado superior  
Desarrollo de aplicaciones multiplataforma  
Programación



Pedro Ángel Serrano Rodríguez  
Raúl Pantoja Porras  
Rafael Ángel Gilarte Núñez  
Víctor Páez Anguita  
Juan Manuel Gómez Wals

# Índice

<b>Descripción</b>	<b>2</b>
<b>Requisitos mínimos</b>	<b>2</b>
<b>Mejoras a la base</b>	<b>2</b>
<b>Diagramas</b>	<b>3</b>
<b>Enlace al github</b>	<b>7</b>

# Descripción

Este proyecto se basa en un sistema de introducción de datos que almacena productos(películas, series, documentales, etc), sus copias y los clientes que compran en el establecimiento. También se ha implementado un sistema que permite al usuario que lo usa añadir, modificar, ver o eliminar tanto productos como sus copias y los clientes. Todos los datos ya mencionados se guardan en archivos xml a modo de base de datos.

## Requisitos mínimos

Los requisitos mínimos para este proyecto han sido:

- Programación orientada a objetos: encapsulamiento, herencia, sobreescritura, polimorfismo, interfaces, clases abstractas, etc.
  - Tenemos herencia, métodos sobreescritos, interfaces y clases abstractas.
- Tipos genéricos.
  - Campo cumplido.
- Tratamiento de cadenas y expresiones regulares.
  - Campo no cumplido.
- Estructuras complejas de datos.
  - Hemos desarrollado ArrayList y HashMap.
- Tratamiento básico de xml.
  - Campo cumplido.

## Mejoras a la base

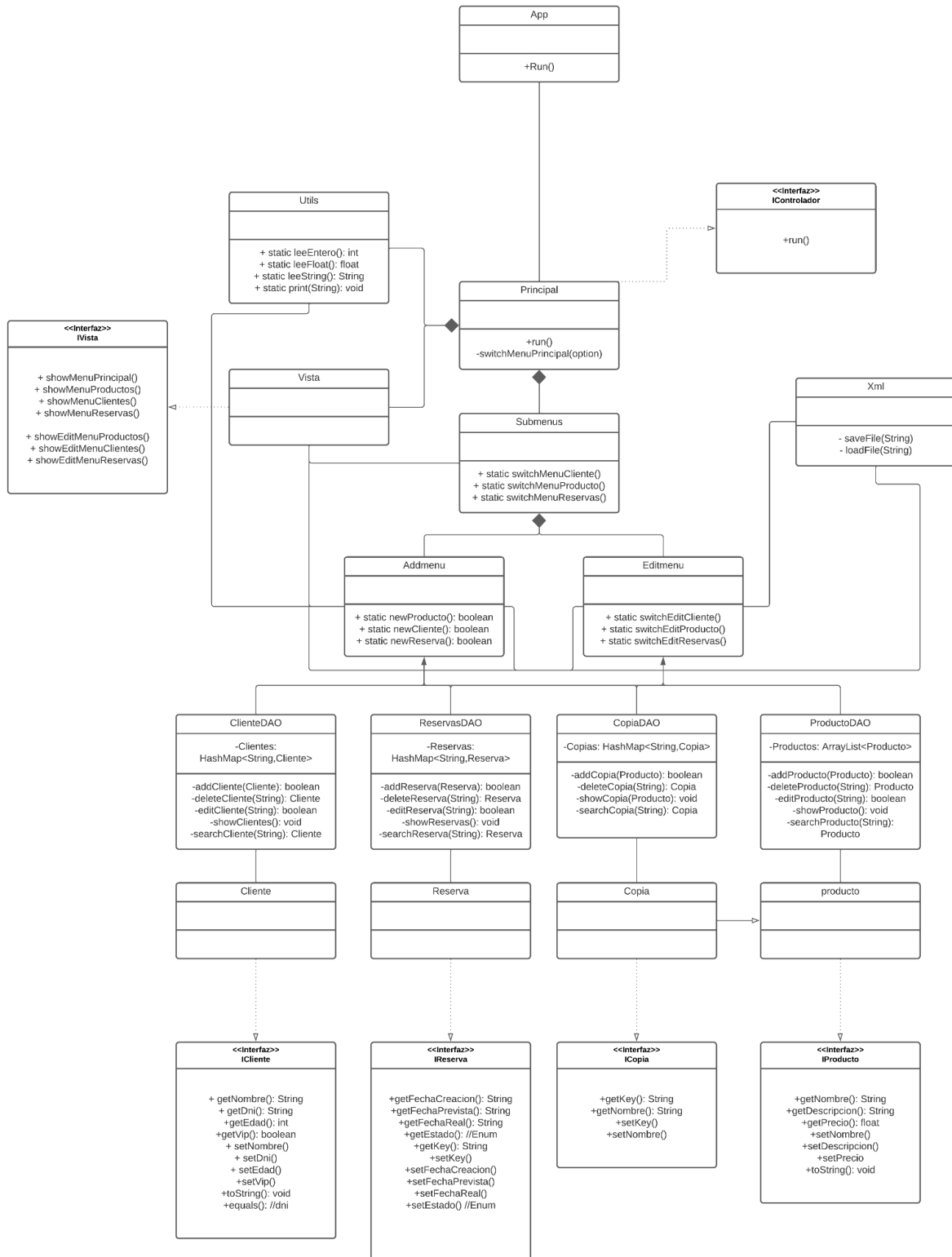
Como mejora hemos creado un paquete “DAO” que nos servirá como repositorio para las listas de cada clase, con ello podremos hacer uso de forma eficaz en el controlador y para que no haya conflicto al crear los modelos que requiere el programa.

Dentro de esta, hemos creado una clase superior en los repositorios la cual nos permite tener una clase aparte y genérica en la que utilizar métodos de guardar y cargar archivos en xml.

El controlador lo hemos dividido en varias clases para tener una mejor organización del programa a la vez que nos permite un mejor control de versión.

# Diagramas

- Diagrama de clases

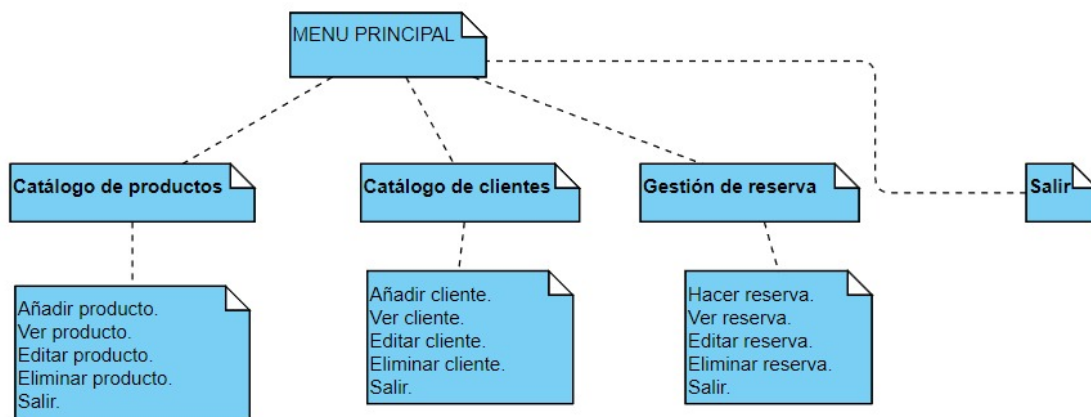


Nuestro video club va a tener las siguientes clases:

- App. va a ser la clase con el main, encargada de ejecutar el programa.
- Principal. va a contener el método run() que es el que ejecuta la app y el método switchMenuPrincipal que es el que permite al usuario coger una opción del menú principal.
- IControlador. Es la interfaz de la clase principal.
- Utils. es la clase que contiene los métodos más útiles para desarrollar las demás clases, los cuáles son:
  - leeEntero. Permite al usuario introducir un número entero sin equivocarse.
  - leeFloat. Permite al usuario introducir un número float sin equivocarse.
  - leeString. Permite al usuario introducir un String sin equivocarse.
  - print. Sirve para mostrar mensajes por pantalla.
- Vistas. es la clase que contiene todos los menús que el usuario va a ver por pantalla.
- IVistas. es la interfaz de la clase Vistas.
- Submenus. Esta clase contiene los métodos siguientes:
  - switchMenuCliente. Permite al usuario elegir una opción del menú clientes.
  - switchMenuproducto. Permite al usuario elegir una opción del menú producto.
  - switchMenuReservas. Permite al usuario elegir una opción del menú reservas.
- XML. es la clase que contiene los métodos que nos permiten guardar y cargar datos de un fichero xml
- ClienteDao. es la clase que contiene los métodos que nos permiten hacer distintas opciones con los clientes, que son las siguientes:
  - addCliente. nos permite agregar un nuevo cliente al hashmap de clientes
  - deleteCliente. nos permite borrar un cliente del hashmap de clientes a través de su id.
  - editCliente. nos permite escoger un cliente a través del id y editarlo.
  - searchCliente. nos permite buscar un cliente a través de su id y mostrarlo por pantalla.
  - showClientes. nos muestra todos los clientes que hay en el hashmap de clientes.
- ReservaDao. es la clase que contiene los métodos que nos permiten hacer distintas opciones con las reservas, que son las siguientes:
  - addReserva. nos permite agregar una nueva reserva al hashmap de reservas
  - deleteReserva. nos permite borrar una reserva del hashmap de reservas a través de su id.
  - editReserva. nos permite escoger una reserva a través del id y editarla.
  - searchCliente. nos permite buscar una reserva a través de su id y mostrarla por pantalla.
  - showReservas. nos muestra todas las reservas que hay en el hashmap de reservas.
- CopiaDao. es la clase que contiene los métodos que nos permiten hacer distintas opciones con las copias, que son las siguientes:
  - addCopia. nos permite agregar una nueva copia al hashmap de copias
  - deleteCopia. nos permite borrar una copia del hashmap de copias a través de su id.
  - searchCopia. nos permite buscar una copia y mostrarla por pantalla.

- showCopia. nos muestra todas las copias que hay en el hashmap de copias.
- ProductoDao. es la clase que contiene los métodos que nos permiten hacer distintas opciones con los productos, que son las siguientes:
  - addProducto. nos permite agregar un nuevo producto al hashmap de productos
  - deleteProducto. nos permite borrar un producto del hashmap de productos
  - editProducto. nos permite escoger un producto a través del id y editarlo.
  - searchProducto. nos permite buscar un producto a través de su id y mostrarlo por pantalla.
  - showProducto. nos muestra todos los productos que hay en el hashmap de productos.
- Cliente. Es la clase donde se crea el objeto cliente. Esta clase tiene un constructor por defecto y uno en el que se rellenan manualmente los datos. También contiene los getters y setters de cliente y un toString.
- Reserva. Es la clase donde se crea el objeto reserva. Esta clase tiene un constructor por defecto y uno en el que se rellenan manualmente los datos. También contiene los getters y setters de reserva y un toString.
- Copia. Es la clase donde se crea el objeto copia. Esta clase tiene un constructor por defecto y uno en el que se rellenan manualmente los datos. También contiene los getters y setters de copia y un toString.
- Producto. Es la clase donde se crea el objeto producto. Esta clase tiene un constructor por defecto y uno en el que se rellenan manualmente los datos. También contiene los getters y setters de producto y un toString.
- ICliente. Es la interfaz de la clase cliente.
- IReserva. Es la interfaz de la clase reserva.
- ICopia. Es la interfaz de la clase copia.
- IProducto. Es la interfaz de la clase producto.

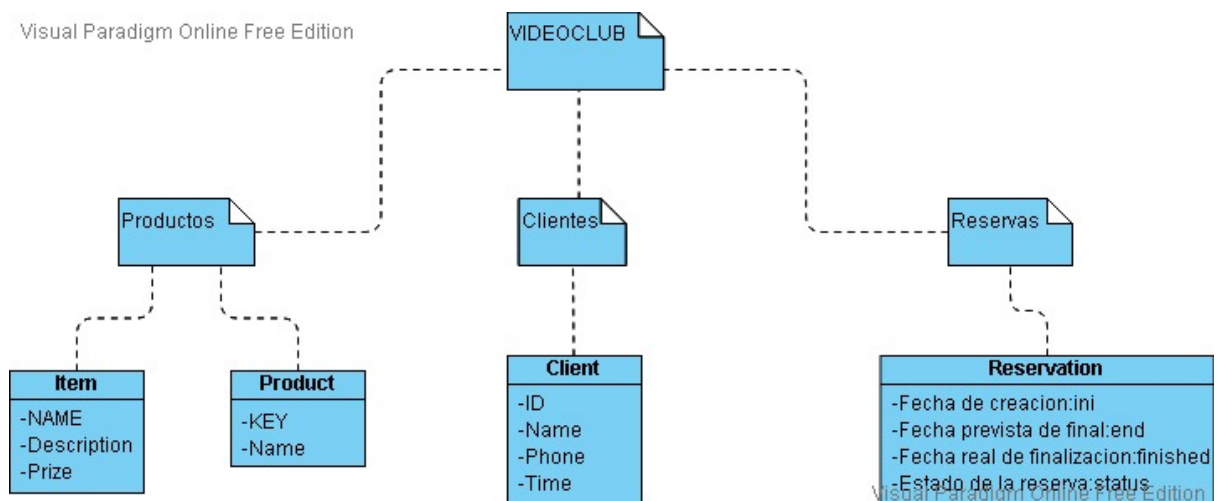
## ● Menú Principal



Nuestro **menú principal** va a tener 3 opciones:

1. Catálogo de productos. Dentro de este, tenemos:
  - a. *Añadir producto*. Nos permite agregar un producto al videoclub.
  - b. *Ver producto*. Nos permite buscar un producto a través del nombre
  - c. *Editar producto*. Nos permite elegir a través del nombre del objeto cual queremos editar y editar sus diferentes campos.
  - d. *salir*. Volver al menú principal.
2. Catálogo de clientes
  - a. *Añadir cliente*. Nos permite agregar un cliente al videoclub.
  - b. *Ver cliente*. Nos permite buscar un cliente a través del dni
  - c. *Editar cliente*. Nos permite elegir a través del dni del cliente cual queremos editar y editar sus diferentes campos.
  - d. *salir*. Volver al menú principal.
3. Gestión de reserva
  - a. *Hacer reserva*. Nos permite agregar una reserva al videoclub.
  - b. *Ver cliente*. Nos permite buscar una reserva a través del id
  - c. *Editar reserva*. Nos permite elegir a través del id de la reserva cual queremos editar y editar sus diferentes campos.
  - d. *salir*. Volver al menú principal.
4. Salir. Salir de la aplicación

## ● Diseño de datos



Lo que nuestra aplicación va a gestionar son los productos, clientes y reservas de un videoclub:

1. Productos: Será las películas que se vendan en el videoclub. Estos productos pueden ser:
  - a. *Item*. Son las películas que hay en el videoclub, estas contienen 3 atributos:
    - i. *Name*. Nombre de la película
    - ii. *Description*. Una breve descripción de la película

- iii. *Precio*. El precio que tiene la película
- b. *Product*. Es el disco y la carátula, es decir, la película concreta, que el cliente va a alquilar. Este va a tener 2 atributos:
  - i. *Key*. Clave del producto para identificarlo sobre los demás.
  - ii. *Name*. Nombre del product.

<https://github.com/Paez11/VideoClub>