

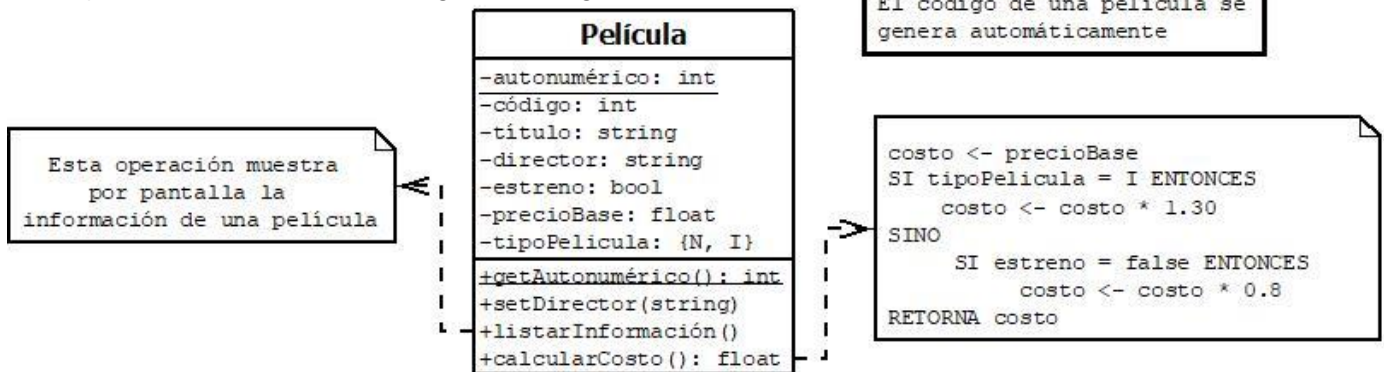
PARADIGMAS DE PROGRAMACION

Trabajo Práctico N° 6

Fecha: 26/09/25

Tema: Constructores y Destructores. Conceptos varios

1) Teniendo en cuenta el siguiente diagrama UML.



- Implemente en C++ la clase **Película** detallada en el diagrama. Determine qué operaciones pueden implementarse como **funciones const**.
 - Agregue a la clase **Película**:
 - Constructor por defecto** que cree un objeto con código 000 y valores por defecto en el resto de sus atributos.
 - Constructor con parámetros** que reciba la información correspondiente para crear e inicializar un objeto.
 - Constructor copia** que permita generar una copia del objeto **Película** recibido.
 - Destructor** que destruya el objeto y escriba por pantalla un mensaje indicando el código del objeto destruido.
 - Escriba un programa de prueba y compruebe el funcionamiento de su clase.
- 2) Teniendo en cuenta la Clase **CuentaObjeto** que permite registrar la cantidad de objetos creados y objetos destruidos:
- Analice el programa de prueba y el código de implementación de la Clase **CuentaObjeto** disponible en el Aula Virtual.
 - Ejecute el programa de prueba y analice la información que presenta el método `mostrarResumen()` en cada una de las invocaciones e identifique en el código el momento en que se crean y destruyen los objetos.
 - ¿Existe algún objeto que haya sido creado en el programa de prueba y no haya sido destruido al finalizar el programa? ¿A qué se debe esto?
- 3) Teniendo en cuenta la Clase **Vector** implementada en el *Trabajo Práctico N° 5 – Ejercicio 3*:
- Modifique la operación `crearVector` para transformarla en un **constructor** con parámetros por defecto.
 - Modifique la operación `destruir` para transformarla en un **destructor** de la Clase **Vector**.
 - Utilice el programa de prueba dado para comprobar el funcionamiento de su clase.
 - Como usuario de la Clase **Vector**, cree una operación **ContarPares** que reciba un objeto del tipo **Vector** y retorne la cantidad de elementos pares se encuentran almacenados dentro del mismo.
 - Habilite las líneas de código que se encuentran bajo el comentario `// PUNTO 3) e`. ¿Qué sucede con el **Vector** luego de la invocación a la función `ContarPares`? ¿Por qué sucede esto?
 - Agregue a la Clase **Vector** un **constructor copia** adecuado que permita crear correctamente una copia de los objetos del tipo **Vector**.
 - Pruebe el funcionamiento del constructor copia y analice nuevamente lo que sucede con el **Vector** luego de la invocación a la función `ContarPares`.