


|  |   |
|--|---|
|  Proyecto CUPi2 | <b>ISIS-1204 Algorítmica y Programación Enunciado</b> |
| Ejercicio:   |   |
| Autor:   |   |
| Fecha:   |   |

## Descripción Global

Se quiere construir una aplicación que maneje los clientes y el alquiler de las películas en una videotienda.

## Objetivos del ejercicio

En el presente ejercicio usted trabajará en:

- Asignación de responsabilidades a clases.
- Definición y uso de contratos de métodos.
- Utilización de la herramienta de documentación Javadoc.
- Uso de excepciones (clase `Exception`) para indicar situaciones de error en la aplicación.

## Preparación

1. Descargue del sitio del curso el archivo .zip del ejercicio.
2. Descomprima el archivo y cree el proyecto en Eclipse llamado **n4\_videotienda** con el contenido.
3. Lea el enunciado del problema disponible en `n4_videotienda/docs/specs/Descripcion.doc`.
4. Revise el documento de requerimientos funcionales de la aplicación `RequerimientosFuncionales.doc`, también disponible en `n4_videotienda/docs/specs/`.
5. Estudie el modelo del mundo diseñado para este ejercicio, disponible en `n4_videotienda/docs/specs/Modelo.mdl` o en las imágenes ubicadas en el mismo directorio. Identifique las clases, relaciones entre clases, atributos y métodos.
6. Lea la documentación html de las clases, métodos y atributos del mundo para conocer las responsabilidades de cada elemento del mundo. La documentación está disponible en `n4_videotienda/docs/api/index.html`.

## Desarrollo

### Instrucciones generales:

- Complete las clases del mundo del problema para que la aplicación funcione de acuerdo a lo especificado.

- Lea en cada clase, toda la documentación disponible de contratos de los métodos que debe desarrollar y cúmplalos.
- Complete toda la documentación de las clases con Javadoc (comentarios de clase, constantes, atributos y métodos). Por cada método que usted cree, describa su contrato con un grupo de los siguientes elementos que apliquen:
  - `<b>pre:</b>`
  - `<b>post:</b>`
  - `@param`
  - `@return`
  - `@exception`
- Ajuste (si es necesario) la invocación de los métodos de las clases del mundo, en las clases de la interfaz.
- Al finalizar la modificación de su ejercicio, genere el Javadoc del proyecto y verifique que la documentación para las clases del mundo esté correcta. Recuerde que puede hacer esta tarea utilizando el archivo ejecutable `doc.bat` localizado en el directorio `n4_videotienda/bin`.

### Instrucciones específicas:

1. Cree la clase `Copia` que representa una copia de película que se alquila. Desarrolle los métodos necesarios para cumplir con todas sus responsabilidades. Recuerde que el identificador de una copia es el nombre de la película y un consecutivo de copia que se lleva por cada película.
2. Cree la clase `Cliente` que representa un cliente de la videotienda. Desarrolle los métodos necesarios para cumplir con todas sus responsabilidades:
  - a. Mantener la información básica del cliente.
  - b. Manejar la lista de copias prestadas (que aún no ha devuelto).
  - c. Mantener el saldo de su cuenta
3. Complete la clase `Pelicula` que representa una película del catálogo de la videotienda. Desarrolle los métodos necesarios para cumplir con todas sus responsabilidades:
  - a. Agregar nuevas copias disponibles
  - b. Manejar el alquiler y devolución de sus copias
  - c. Dar información sobre sus copias
4. Complete la clase `VideoTienda` que representa a la videotienda. Desarrolle los métodos necesarios para cumplir con todas sus responsabilidades:
  - a. Actualizar la información de la tarifa
  - b. Mantener y consultar la información del catálogo de películas
  - c. Mantener y consultar la información de los clientes
  - d. Realizar los procesos de alquiler y devolución de copias de películas

### Validación

Para comprobar el funcionamiento de su ejercicio, ejecute el programa e interactúe con todas las opciones disponibles en la interfaz.

Adicionalmente, el ejercicio contiene unas clases de prueba que están ajustadas a la implementación original del ejercicio. Al finalizar las pruebas, es probable que estas clases no compilen por las diferencias de implementación que tenga su versión con la original. Opcionalmente puede tratar de ajustar las pruebas a sus métodos y ejecutarlas para probar el funcionamiento esperado.