

# Programación 4

EXAMEN DICIEMBRE 2020

*Debido a las restricciones impuestas por la pandemia COVID-19, el examen se realizó en modalidad virtual, según las pautas del documento anexo.*

## **Problema 1 (30 puntos)**

La cooperativa “Creciendo” gestiona un proyecto de acompañamiento a emprendimientos de personas en situación de vulnerabilidad socioeconómica. El acompañamiento consta de capacitación y participación en eventos comerciales coordinados por el equipo técnico de la cooperativa. Se desea modelar un sistema para la gestión del proyecto.

Los usuarios pueden ser emprendedores (personas que tienen emprendimientos) o asesores (que forman parte del equipo técnico de la cooperativa). De cada usuario interesa conocer su cédula de identidad que lo identifica, nombre, fecha de nacimiento, teléfono y correo electrónico.

De los emprendimientos se conoce el rubro, el tipo de actividad (Producción o Servicio), fecha de creación y monto de ingresos. A su vez, los emprendedores pueden ser parte de más de un emprendimiento a la vez. Cada emprendimiento tiene un asesor como responsable, pudiendo este ser responsable de tres emprendimientos como máximo.

Para ayudar a los emprendedores a aumentar sus ventas la cooperativa organiza eventos en los que participan los emprendedores. De cada evento se conoce su título, descripción, fecha y un código que lo identifica. Los eventos son dirigidos por hasta dos asesores.

Por último, para cada emprendedor que participa de un evento, es de interés conocer la hora de llegada y la hora en que se retiró.

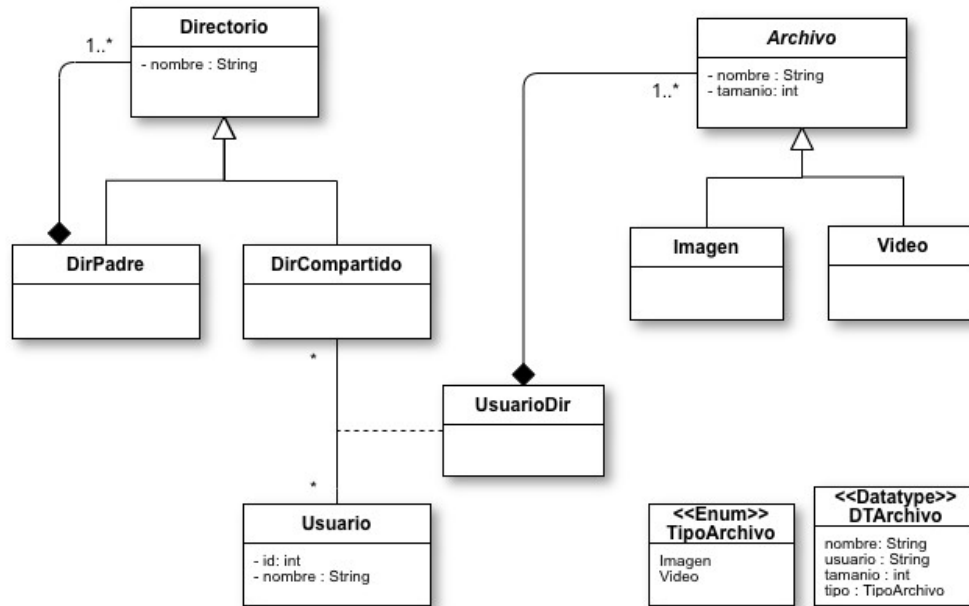
### **Se pide:**

- a) Realizar el Modelo de Dominio en UML de la realidad descrita.
- b) Escribir las restricciones en lenguaje natural.

## **Problema 2 (35 puntos)**

Se propone desarrollar un sistema de almacenamiento compartido de archivos en el que distintos usuarios pueden subir archivos dentro de un mismo directorio. Los directorios forman una estructura en forma de árbol en donde existen dos tipos de directorios: los directorios padre, que contienen un conjunto de directorios y no contienen archivos y los directorios compartidos, que solamente pueden contener archivos. Los archivos pueden ser del tipo Imagen o Video. De cada archivo se conoce su tamaño y su nombre que lo identifica dentro del directorio. De cada usuario se conoce su nombre y el id dentro del sistema, que lo identifica.

Se cuenta con el siguiente modelo de dominio:



Se definió el siguiente contrato para la primera etapa de construcción de la aplicación.

listarImágenes(usr: integer , dir : String) : Set<DTArchivo>	
Descripción	Obtiene un listado con los datos de los archivos del tipo <b>Imagen</b> del usuario con identificador <b>usr</b> , subidos en directorios compartidos dentro del directorio de nombre <b>dir</b> .
Parámetros	<b>usr</b> : El identificador del usuario en el sistema. <b>dir</b> : El nombre del directorio.
Precondiciones	Existe en el sistema una instancia de <b>Usuario</b> con id <b>usr</b> . Existe en el sistema una instancia de <b>DirCompartido</b> de nombre <b>dir</b> .
Postcondiciones	Se retorna una colección de DTArchivo que para cada elemento contiene el nombre del archivo, el tamaño, el tipo y el nombre del usuario al que pertenece.

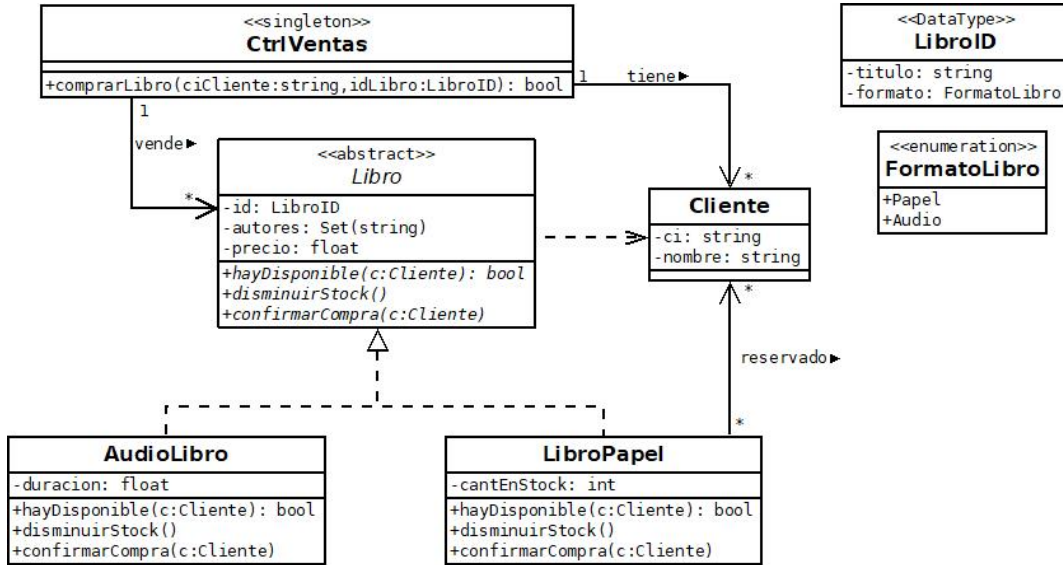
**Se pide:**

- Realizar el Diagrama de Comunicación correspondiente a la operación del sistema **listarImágenes()** incluyendo visibilidades.
- Realizar el Diagrama de Clases de Diseño (DCD) resultante.

*Nota:* No incorporar setters ni getters al DCD. Incorporar en el DCD sólo los constructores o destructores que se utilicen en los diagramas de comunicación. No incorporar al DCD las operaciones de colecciones.

**Problema 3 (35 puntos)**

El siguiente diagrama de clases muestra el diseño (parcial) de un sistema para la venta de libros. Un libro se identifica por su título y formato, que puede ser papel o audio. Para los libros en formato papel, la venta queda sujeta al stock disponible, mientras que para los audiolibros siempre hay disponibilidad. Los clientes cuentan además con la posibilidad de reservar libros en formato papel.



Se pide:

- Implementar el archivo .h de las clases CtrlVentas, Libro y LibroPapel.
- Implementar el archivo .cpp de las clases CtrlVentas y LibroPapel, donde:
  - comprarLibro():bool chequea que haya disponibilidad del libro, y en caso afirmativo disminuye su stock y confirma la compra. La operación devuelve true únicamente si la compra pudo realizarse.
  - hayDisponibilidad(): devuelve true únicamente si el cliente tiene una reserva previa del libro, o si la diferencia entre el stock actual y la cantidad de reservas existentes es mayor que cero.
  - disminuirStock(): disminuye en una unidad el stock del libro.
  - confirmarCompra(): si el cliente tiene una reserva del libro la elimina.

Observaciones:

- Puede suponer la existencia de la interface ICollectible e implementaciones de ICollection (clase List), Ikey (clase OrderedKey), IDictionary (clase OrderedDictionary) e Iterator según sea necesario.
- Es posible utilizar las clases set<T>, vector<T> o map<K,V> de la STL.
- Se deben incluir constructores y destructores en caso de aplicar.
- No implementar setters y getters.
- No incluir directivas al precompilador (#include, #define, etc).