

# Programación 4

## PARCIAL FINAL EDICIÓN 2011

Por favor siga las siguientes indicaciones:

- Escriba con lápiz
- Escriba las hojas de un solo lado
- Escriba su nombre y número de documento en todas las hojas que entregue
- Numere las hojas e indique el total de hojas en la primera de ellas
- Recuerde entregar su número de parcial junto al parcial

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

- a) Indique la diferencia entre una asociación *need-to-know* y una asociación de comprensión.
- b) Se le ha encomendado el análisis de un sistema de venta de entradas a eventos (e.g. recitales y conciertos) brindados por artistas del ambiente musical. Para ello se cuenta con la siguiente descripción de la realidad:

“El objetivo es registrar todos los eventos de los cuales la empresa vende entradas. De cada artista se conoce su nombre que lo identifica y una descripción del mismo. Los artistas pueden ser músicos o grupos. Si es un músico se conoce su fecha de nacimiento. Si es un grupo se debe conocer cada uno de los músicos que lo integran y el año de formación. De cada evento se conoce su nombre que lo identifica, el artista que participa, la fecha en la que se realiza el evento (e.g. 10 de setiembre 21.00 hs) y el lugar donde se realiza (e.g. Velódromo Municipal). Para cada lugar se conoce su nombre que lo identifica y su dirección. Cada lugar tiene diferentes tipos de localidades (e.g. Platea Alta, Campo VIP) cada una con un precio diferente para el evento. De cada tipo de localidad se conoce su nombre (que lo identifica en el lugar) y su capacidad (cantidad de asientos). Para un evento es necesario conocer el precio de las entradas y cuantas localidades se vendieron para cada tipo de localidad. Esto último se hace con la finalidad de llevar un registro de las ganancias generadas por la venta de entradas y además para verificar que no haya una sobreventa de las mismas.”

Adicionalmente se plantea el siguiente caso de uso:

Nombre:	Alta artista
Actores:	Usuario
Sinopsis:	El usuario da de alta un nuevo artista en el sistema. Para ello ingresa el nombre y la descripción. En caso de ser músico ingresa su fecha de nacimiento, mientras que si es un grupo ingresa los nombres de cada uno de sus integrantes y el año de formación.

**Se pide:**

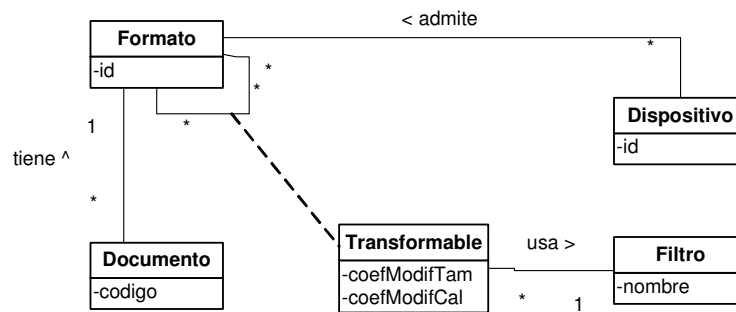
- Realizar el modelo de dominio en UML y expresar sus restricciones en lenguaje natural.
- Realizar el Diagrama de Secuencia del Sistema para el caso de uso Alta artista.

**Problema 2 (35 puntos)**

Considere un repositorio de documentos cuyo contenido es accedido desde diferentes dispositivos, cada uno con su capacidad particular de procesamiento y visualización. Los documentos tienen diferentes formatos y existe la posibilidad de utilizar filtros de conversión entre formatos, con el propósito de poder procesar y visualizar todos los documentos en todos los dispositivos.

Se desea diseñar un sistema que, ante una solicitud de visualización de un documento en un dispositivo, pueda determinar el formato más adecuado de visualización, según el tamaño del archivo que contiene la información del documento y la calidad de la visualización.

Se cuenta con el siguiente modelo de dominio, donde el tipo asociativo Transformable indica la posibilidad de transformar un Formato a otro, utilizando un Filtro. Además, mediante dos coeficientes se indica por un lado una aproximación de la modificación en el tamaño del archivo y por otro, de la calidad de visualización, una vez aplicada la transformación correspondiente.



Se plantean además los siguientes casos de uso:

Nombre:	Especificar transformaciones
Actores:	Usuario
Sinopsis:	El usuario da de alta transformaciones de un formato a otros formatos. Para ello, inicialmente ingresa el identificador del formato origen. Luego, para cada formato destino, ingresa su identificador, los coeficientes de modificación de tamaño y calidad, y el nombre del filtro asociado.

Nombre:	Determinar filtro
Actores:	Usuario
Sinopsis:	El usuario obtiene el filtro asociado a la mejor transformación posible para visualizar un documento en un dispositivo. Para ello, indica el código del documento y el identificador del dispositivo, además de un coeficiente que indica la importancia que asigna al tamaño del archivo resultante en relación a su calidad de visualización. El sistema le devuelve el nombre del filtro asociado a la transformación que mejor se ajusta al coeficiente de importancia dado, teniendo en cuenta los formatos que admite el dispositivo. La transformación que mejor ajusta según un coeficiente $c$ es aquella que maximiza el valor $(1 - c) * q - c * t$ , siendo $q$ y $t$ los coeficientes de modificación de la calidad y tamaño de la transformación, respectivamente.

Para estos casos de uso, se han determinado las siguientes operaciones del sistema, con sus correspondientes contratos parciales:

Operación	ingFormatoOrigen(idO:Integer)
Referencias	Especificar transformaciones
<b>Pre y post condiciones</b>	
pre: Existe en el sistema una instancia de Formato identificada por idO.	
post: Se recuerda la instancia de Formato identificada por idO en la memoria del sistema.	

Operación	ingDatosTransf(idD:Integer;cTam,cCal:Real;fil:String)
Referencias	Especificar transformaciones
<b>Pre y post condiciones</b>	
pre: Existe en el sistema una instancia de Formato fD identificada por idD.	
pre: Existe en el sistema una instancia de Filtro f identificada por fil.	
pre: Existe en la memoria del sistema una instancia de Formato fO recordada.	
post: Existe en el sistema una nueva instancia de Transformable t, que representa un link entre fO y fD, cuyos atributos valen cTam y cCal. Existe un nuevo link entre t y f.	

Operación	darMejorFiltro(codDoc,idDisp:Integer;impRel:Real):String
Referencias	Determinar filtro
<b>Pre y post condiciones</b>	
pre: Existe en el sistema una instancia de Documento doc identificada por codDoc.	
pre: Existe en el sistema una instancia de Dispositivo disp identificada por idDisp.	
pre: disp admite alguno de los formatos a los que se puede transformar el formato de doc.	
pre: disp no admite el formato que tiene doc.	
post: Se retorna el identificador de la instancia de Filtro asociada a la transformación t del formato que tiene doc, que minimiza el valor $v$ , entre los formatos admitidos por disp, siendo $v = (1-impRel) * t.coefModifTCal - impRel * t.coefModifTam$ .	

**Se pide:**

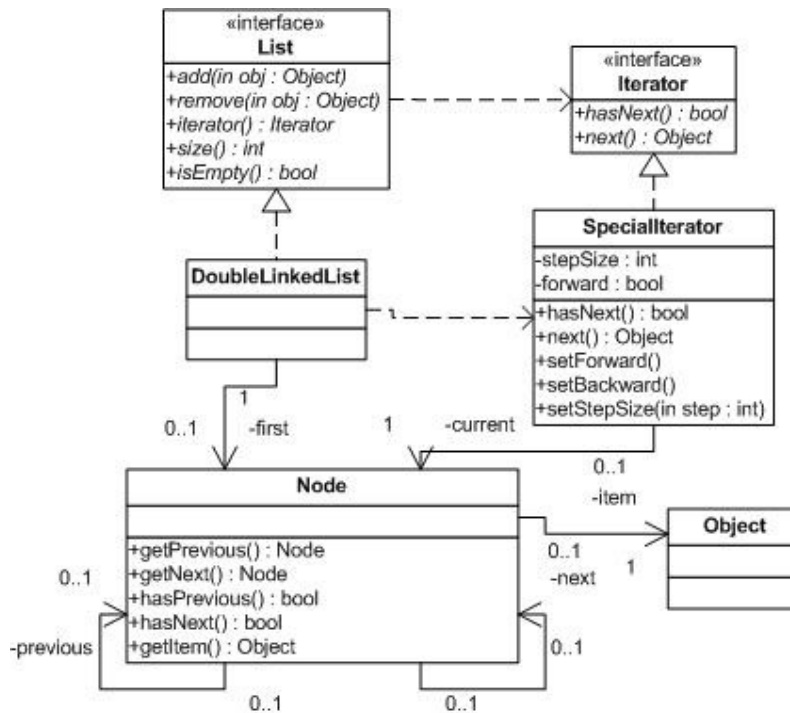
- i. Realizar los Diagramas de Comunicación de todas las operaciones del sistema.
- ii. Realizar el Diagrama de Clases de Diseño de la solución.

*Nota:* no realizar el chequeo de precondiciones, asumir la existencia de un único controlador.

**Problema 3 (35 puntos)**

Se desea implementar un iterador de lista especial, que permita tanto la recorrida de una lista hacia adelante como hacia atrás; el comportamiento del iterador debe permitir que la dirección de la recorrida pueda cambiarse incluso durante el uso del mismo. Se quiere que el iterador pueda avanzar por la lista no solo de a un elemento, sino que también se pueda cambiar el tamaño del salto en la recorrida (ej.: avanza de a un elemento, de a dos elementos, etc.). Como requerimiento del iterador se tiene que al intentar avanzar con el método `next()` se chequee la existencia del elemento a retornar y en caso de que este no exista se lance una excepción del tipo `std::domain_error`.

Con este fin se realizó el siguiente diseño completo de la solución:



Se pide:

- i. Implementar completamente en C++ la clase `SpecialIterator` (archivos `.h` y `.cpp`), de acuerdo al diseño previamente detallado.
- ii. Implementar la operación `iterator` de la clase `DoubleLinkedList`.