

# Programación 4

## EXAMEN DICIEMBRE 2013

**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 examen junto al examen

### Problema 1 (30 puntos)

- a. ¿Cuáles son los dos objetivos de la Etapa de Análisis vista en el curso?
- b. **Contexto:** La Agencia Nacional de Investigación e Innovación (ANII) del Uruguay es una entidad gubernamental que promueve y estimula la investigación y la aplicación de nuevos conocimientos a la realidad productiva y social del país (fuente: [www.anii.org.uy](http://www.anii.org.uy)). Para ello, cuenta con una serie de instrumentos, muchos de los cuales están pensados para apoyar económicamente a empresas privadas en proyectos de I+D+i (Investigación, Desarrollo e Innovación). Ejemplos de instrumentos son el Fondo Orestes Fiandra (el cual brinda un apoyo de hasta un millón de dólares para proyectos de internacionalización de empresas) o el Fondo Innovagro (el cual brinda un apoyo de hasta 120.000 dólares para proyectos en agropecuaria y agroindustria).

**Realidad:** Los instrumentos se identifican por su nombre. Las empresas postulan proyectos a los diferentes instrumentos, indicando el monto solicitado para el proyecto, la fecha de presentación del mismo y su justificación (un texto que intenta convencer a la ANII de porque debería financiar ese proyecto). El monto solicitado en el proyecto no debe superar el monto máximo financiable que indica el instrumento. La empresa postulante debe estar ingresada en el sistema de la ANII, habiendo registrado su nombre, RUT, fecha de constitución y representante legal. Cada proyecto debe ser presentado a un único instrumento, aunque no existe límite para la cantidad de proyectos que una empresa pueda presentar. Finalmente, la ANII considera más favorablemente aquellos proyectos que sean presentados en conjunto, empresa privada + institución pública (ya que promueve la vinculación Academia-Industria). Por tanto, un proyecto presentado por una empresa a un instrumento, tiene la posibilidad (no mandatorio) de presentar a una institución pública, la cual debe estar registrada con un nombre (que la identifica) y tipo (Facultad, Fundación o Instituto de Investigación).

**Se pide:**

- i. Realizar el Modelo de Dominio de la realidad planteada, con restricciones en lenguaje natural.
- ii. Realizar el Diagrama de Secuencia del Sistema para el siguiente Caso de Uso, incluyendo Datatypes en caso de ser necesario:

Caso de Uso:	Ingreso de Proyecto
Actor:	Representante de la Empresa
Descripción:	Este CU comienza con la identificación del representante de la empresa (mediante usuario y password). Si los datos son correctos, se prosigue con el CU, de lo contrario este culmina. Luego el actor solicita un listado de los instrumentos vigentes, devolviendo todos los datos de estos, de

los cuales se elige uno. Luego se ingresan los datos del proyecto que se presentara a ese instrumento (monto solicitado, fecha y justificación). Opcionalmente se puede solicitar una lista de instituciones públicas registradas en el Sistema para seleccionar la institución con la cual la empresa presenta el proyecto. Finalmente, se confirma el ingreso del proyecto.

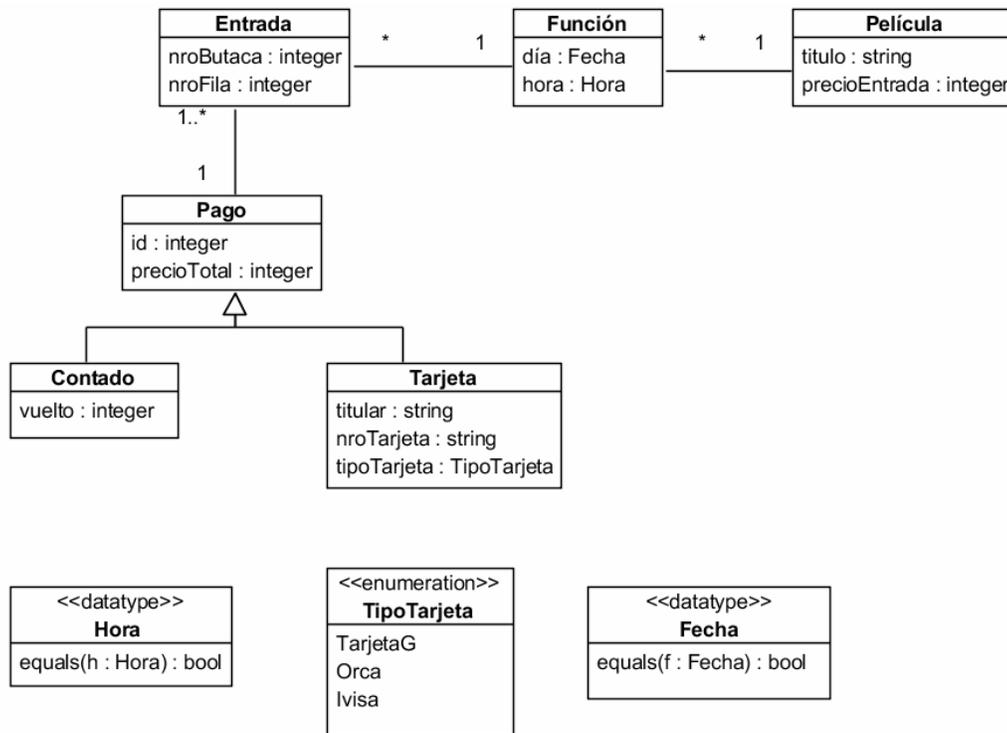
## Problema 2 (35 puntos)

a)

- i. Explique brevemente para qué sirven los criterios GRASP.
- ii. Mencione 3 criterios GRASP.

b) Usted es un desarrollador y una cadena de salas de cines le contactó para desarrollar un sistema para la gestión de las películas que ofrece cada sucursal de la cadena. Es importante mantener información acerca de los pagos que se hacen al comprar entradas y la disponibilidad de las butacas para cada función de cada película. Los pagos que realizan los clientes se hacen por un conjunto de entradas. Por ejemplo un cliente puede comprar 3 entradas para una función de una película y pagar por esas 3 entradas. Los pagos se identifican en el sistema por su id que tiene que ser autogenerado al crear el pago. El atributo id debe ir de 1 en adelante. Se determinó que los pagos pueden hacerse tanto con tarjeta de crédito como al contado. En el primer caso es necesario saber los datos de la tarjeta de crédito que se utilizó para pagar y en el segundo caso se necesita saber con cuánto efectivo pagó y cuánto dinero se dio de vuelto al cliente.

Durante la etapa de análisis se generó el siguiente diagrama con un modelo de dominio de la solución.



Es necesario diseñar el Caso de Uso *Compra de Entradas Contado* que consiste en comprar una o más entradas para una determinada función de una Película pagando al contado. Durante la etapa de análisis se detectaron las siguientes operaciones del sistema para dicho caso de uso.

Operación	<code>inicioCompra(titPelicula:string, d:Dia, h:Hora)</code>
Pre y postcondiciones	
pre: Existe en el sistema una Película <code>p</code> , cuyo atributo título tiene el valor <code>titPelicula</code> .	
pre: Existe una Función <code>func</code> asociada a <code>p</code> , cuyos atributos <code>dia</code> y <code>hora</code> coinciden con <code>d</code> y <code>h</code> respectivamente.	
post: Se recuerda la instancia <code>func</code> en la memoria del sistema.	
post: Se recuerda el valor de <code>precioTotal</code> con 0.	
post: Se recuerda el valor de <code>precioEntrada</code> de <code>p</code> .	

Operación	<code>agregarEntrada(fila: integer, butaca: integer)</code>
Pre y postcondiciones	
pre: Se recuerda una Función <code>func</code> en la memoria del sistema.	
pre: Se recuerda el valor <code>precioTotal</code> .	
pre: Se recuerda el valor <code>precioEntrada</code> .	
pre: No existe una instancia de Entrada asociada a <code>func</code> con <code>nroFila = fila</code> y <code>nroButaca = butaca</code> .	
post: Se crea una nueva instancia de Entrada llamada <code>entr</code> con atributos <code>nroFila = fila</code> y <code>nroButaca = butaca</code> .	
post: Se asocia <code>entr</code> a <code>func</code> .	
post: Se recuerda <code>entr</code> en la memoria del sistema.	
post: <code>precioTotal = precioTotal + precioEntrada</code>	

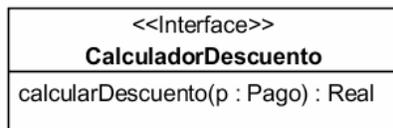
Nótese que puede invocarse a `agregarEntrada()` varias veces en el Caso de Uso.

Operación	<code>pagarAlContado(pagoCliente: integer)</code>
Pre y postcondiciones	
pre: Se recuerda un conjunto de entradas <code>entr</code> y el valor <code>precioTotal</code> en la memoria del sistema.	
pre: <code>pagoCliente &gt;= precioTotal</code>	
post: Se incrementa en uno el valor del último número de <code>id Pago</code> disponible en el Sistema.	
post: Se crea una nueva instancia de Contado <code>pago</code> con atributos <code>precioTotal</code> igual al recordado y <code>vuelto = precioTotal - pagoCliente</code>	
post: Se asocian los <code>entr</code> a <code>pago</code>	
post: Se limpia la memoria del sistema.	

**Se pide:**

- i. Realizar los Diagramas de Comunicación de las operaciones del sistema mencionadas previamente, indicando el tipo de visibilidad en todos los mensajes y definiendo los nuevos DataTypes que puedan haber surgido.
- ii. Realizar el Diagrama de Clases de Diseño resultante.

c) Debido a la creciente tendencia de aplicar descuentos a las compras en crédito se detectó la necesidad de incorporar un sistema que aplique descuentos según la tarjeta de crédito y la fecha de la función de las entradas que se compran. Por ejemplo todas las compras con TarjetaG tienen un 20% de descuento los 1ro de cada mes. Puede haber diferentes formas de calcular descuentos y quieren agregarse sin tener que modificar el código existente, los descuentos siempre se calcularán en base al pago teniendo una operación.



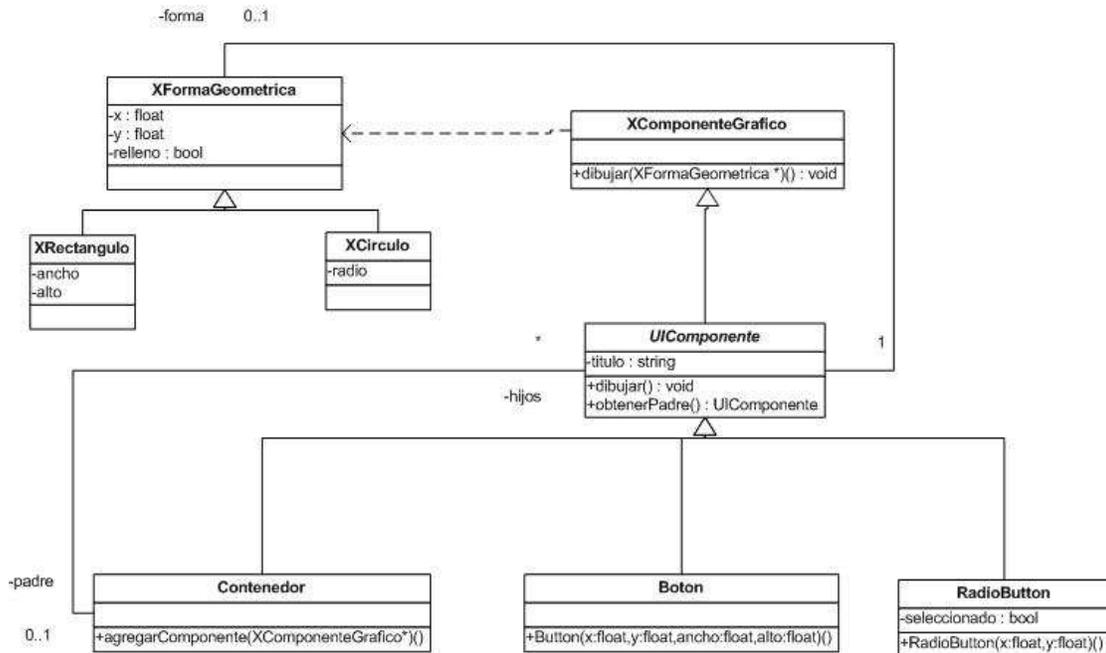
**Se pide:** Indicar el/los patrones de diseño que cree necesario para implementar dicho requerimiento indicando los roles para cada clase. Justifique su elección.

### Problema 3 (35 puntos)

Se desea implementar en C++ un framework para el desarrollo de interfaces de usuario. Para esto se comenzará implementando un prototipo que sea capaz de dibujar en un dispositivo gráfico los siguientes componentes: Botones y radio buttons.

Se decidió utilizar un conjunto de clases de una biblioteca externa para el manejo de dispositivos gráficos. Estas clases son: **XComponenteGrafico**, **XFormaGeometrica**, **XRectangulo** y **XCirculo**.

El siguiente diagrama ilustra el diseño propuesto:



La biblioteca dispone de la clase **XComponenteGrafico** que dado un elemento de tipo **XFormaGeometrica** se encarga de dibujarlo en pantalla. Se pueden dibujar en pantalla Rectángulos representados por la clase **XRectangulo** de la cual se conoce las coordenadas  $x$  e  $y$  del vértice superior izquierdo, el ancho y el alto; y también circunferencias representadas por la clase **XCirculo** de las cuales se conoce la coordenada  $x$  e  $y$  del centro, y el radio de la misma. Todas las coordenadas están dadas en pixeles.

Los objetos de la clase **XFormaGeometrica** disponen de un atributo denominado **relleno**, que si su valor es verdadero la clase **XComponenteGrafico** dibujará el elemento en pantalla relleno de color negro, en caso contrario solo dibujará el contorno.

Para representar gráficamente un botón se utilizará un rectángulo sin relleno, para representar un radio button activo se utilizará una circunferencia con relleno, y para representar un radio button desactivo una circunferencia sin relleno. Los contenedores no tienen forma geométrica, por lo que no se dibujan en pantalla, son utilizados para agrupar otros componentes. Las clases **Boton** y **RadioButton** conocen que forma geométrica deben tener asignada por lo que en el momento en que se crea cada una de ellas éstas deben ser asignadas. Por defecto, todos los radio buttons están desactivos.

**PARTE A****Se pide:**

Implementar los .h y .cpp de la clase abstracta **UIComponente** y de las clases **Boton**, **RadioButton** y **Contenedor**, con sus atributos, pseudoatributos, constructor y los métodos indicados en el diagrama. No implemente métodos set ni get.

**PARTE B**

Se desea agregar un nuevo componente al prototipo consistente en una variación del componente radio group. Los **radio group** agrupan varios componentes del tipo radio button pero esta variante permitirá que cada radio group podrá agrupar tanto radio buttons como otro radio group. Un radio button puede pertenecer a un solo radio group, lo mismo que un radio group puede pertenecer a un solo radio group. Se debe tener en cuenta que solamente uno de los radio buttons del grupo puede estar activado, y se debe considerar que cada radio group es independiente a pesar de estar contenido en otro radio group.

**Se pide:**

- i. Agregar el componente al diagrama de diseño e implementarlo.
- ii. En caso de haber utilizado algún patrón de diseño, descríballo explicando que roles realiza cada clase.

**Observaciones generales:**

- No incluir directivas al precompilador.
- Puede suponer la existencia de implementaciones de `IDictionary`, `ICollection`, `IIterator` y `KeyInt` según sea necesario.
- Es posible utilizar las clases `set<T>` y `map<K, V>` de la STL.