

Programación 4

PARCIAL FINAL EDICIÓN 2014

Por favor siga las siguientes indicaciones:

- Escriba con lápiz y de un solo lado de las hojas
- 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 numero de parcial junto al parcial
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el parcial

Problema 1 (35 puntos)

Una agencia de publicidad desea invertir en la construcción de un software que le permita gestionar las campañas de publicidad. Cada campaña involucra dos etapas: primero se configura y luego se implementa.

La definición de la campaña implica elegir qué medios de comunicación se utilizarán. Existen tres tipos de medios de comunicación: TV (interesando saber la cantidad mínima y máxima de minutos de aire permitidos), Internet (interesando saber el canal que se puede utilizar, pudiendo ser Facebook, Twitter o un Portal) y Diario (interesando saber el vendedor con el cual hay que contactarse). Todos los medios tienen un nombre y una descripción.

La implementación de la campaña implica la definición de las pautas que se utilizarán para cada medio. Una pauta contiene la fecha en la cual debe salir y un código, y existen análogamente tres tipos de pautas (pauta para TV, pauta para Internet y pauta para Diario). Por tanto, si en la campaña se configuró el uso de un diario (es decir un medio de comunicación de tipo Diario), entonces al momento de implementar la campaña se debe definir una pauta de tipo Diario, indicando el texto que debe salir impreso en el mismo. Análogamente, si en la campaña se configuró el uso de TV, entonces la implementación de la campaña debe definir una pauta para ese medio, incluyendo la cantidad de minutos de aire (la cual debe ser coherente con la cantidad mínima y máxima de minutos definidos en el medio). Finalmente, si en la campaña se configuró el uso de Internet, entonces la implementación de la campaña debe definir una pauta para ese medio, incluyendo una imagen a ser mostrada en Internet (basta con guardar solo la URL).

Casos de Uso: se han definido dos Casos de Uso: Configurar Campaña e Implementar Campaña. A continuación se describe el segundo:

Caso de Uso:	Implementar Campaña
Actores:	Usuario
Descripción:	Este Caso de Uso comienza cuando el usuario selecciona una campaña (ingresando el ID de la misma). Luego se ingresan todas las pautas, una por cada medio definido (eligiendo el medio correspondiente mediante su nombre). Si bien todas las pautas tienen fecha y código, estos datos serán calculados internamente por el Sistema (la fecha será la fecha actual del Sistema y el código será autogenerado). Para las pautas en TV se debe ingresar la cantidad de minutos de aire. Para las pautas en Internet se debe ingresar la imagen (URL). Para las pautas en Diario se debe ingresar el texto. El

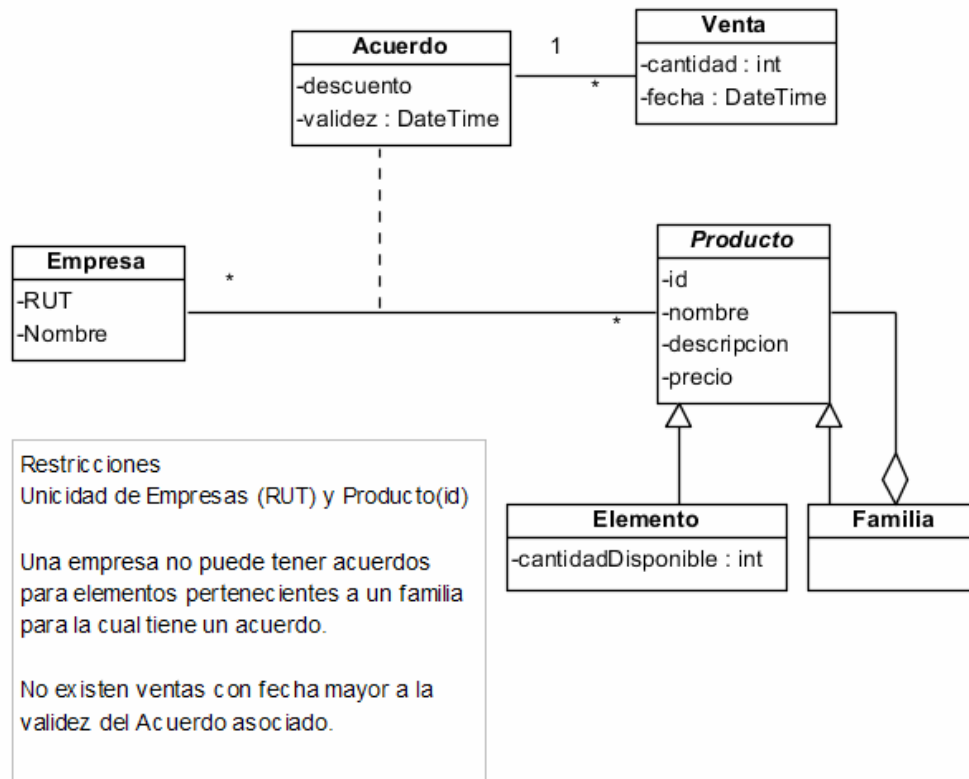
	Caso de Uso termina cuando el usuario confirma la implementacion de la campaña.
--	---

Se pide:

- i. Realizar el Modelo de Dominio de dicha realidad, incluyendo restricciones en lenguaje natural.
- ii. Realizar el Diagrama de Secuencia del Sistema correspondiente al Caso de Uso Implementar Campaña, incluyendo Datatypes en caso de ser necesario y especificando el uso o no de la memoria del Sistema.

Problema 2 (35 puntos)

Un proveedor industrial construye productos (Elementos) los cuales pueden ser agrupados sucesivamente en Familias. Como estrategia de venta, se realizan acuerdos con empresas ofreciéndoles un descuento porcentual al precio de venta del producto por determinado tiempo.

*Parte A:*

Se consideran los casos de uso **obtenerValorStock** y **calcularCompraTotal**, cada uno de los cuales es modelado con una sola operación del sistema, cuyos contratos se especifican a continuación.

obtenerValorStockDeProducto(id: int) : int	
Descripción	Dado un producto calcula el valor del stock actual utilizando la cantidad disponible y el precio del elemento o conjunto de estos que lo compongan.
Precondiciones	- Existe el producto con id <i>id</i> .
Postcondiciones	- La operación es de consulta, no hay alteración en el sistema.

calcularCompraTotal(rut: String) : int	
Descripción	Calcula y retorna el monto total de las compras realizadas por una empresa dada por su RUT. Esto es, la suma del monto de todas las ventas de todos los acuerdos de la empresa, aplicando el descuento correspondiente para cada uno de ellos.
Precondiciones	- Existe la empresa con RUT <i>rut</i> .
Postcondiciones	- La operación es de consulta, no hay alteración en el sistema.

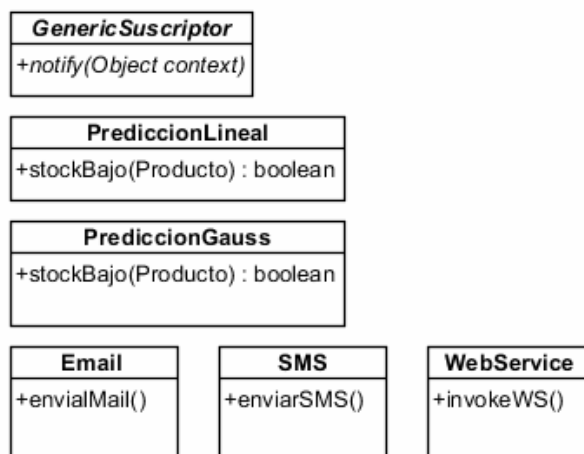
Se pide:

- i. Realizar los Diagramas de Comunicación de las dos operaciones descritas, indicando el tipo de visibilidad en todos los mensajes.
- ii. Realizar el Diagrama de Clases de Diseño resultante.

ParteB:

Se desea tener un mecanismo de alertas que dispare notificaciones de distintos tipos (SMS, Email, Webservice, etc) a distintos y múltiples tipos de suscriptores (que extienden *GenericSuscriptor*) cuando el nivel de stock de un producto es bajo. Estos suscriptores son registrados por producto, indicando el o los tipos de notificación que desean recibir (SMS, Email, Webservice). Cada vez que se realiza una venta, se ejecuta un algoritmo de predicción que define si el stock de un producto es bajo (o no) y dispara la notificación. Este algoritmo varía por producto, y es necesario que pueda definirse individualmente y poder variarlo en el tiempo.

Se definieron las siguientes clases para proveer algunas de las funcionalidades mencionadas y sobre las cuales puede realizar modificaciones para soportar los requerimientos íntegramente. Considere que el mecanismo debe prever poder agregar más algoritmos de predicción y tipos de notificaciones fácilmente.



Se pide:

- iii. Realizar las modificaciones necesarias al DCD realizado en la parte ii, para diseñar este mecanismo. Debe indicar los tipos de datos de todos los parámetros y retornos de las operaciones que defina para dar soporte a este mecanismo e indicar pseudocódigo en modo nota cuando justifique.
- iv. En caso de haber usado un Patrón de Diseño, identificarlo indicando los roles de las clases participantes.

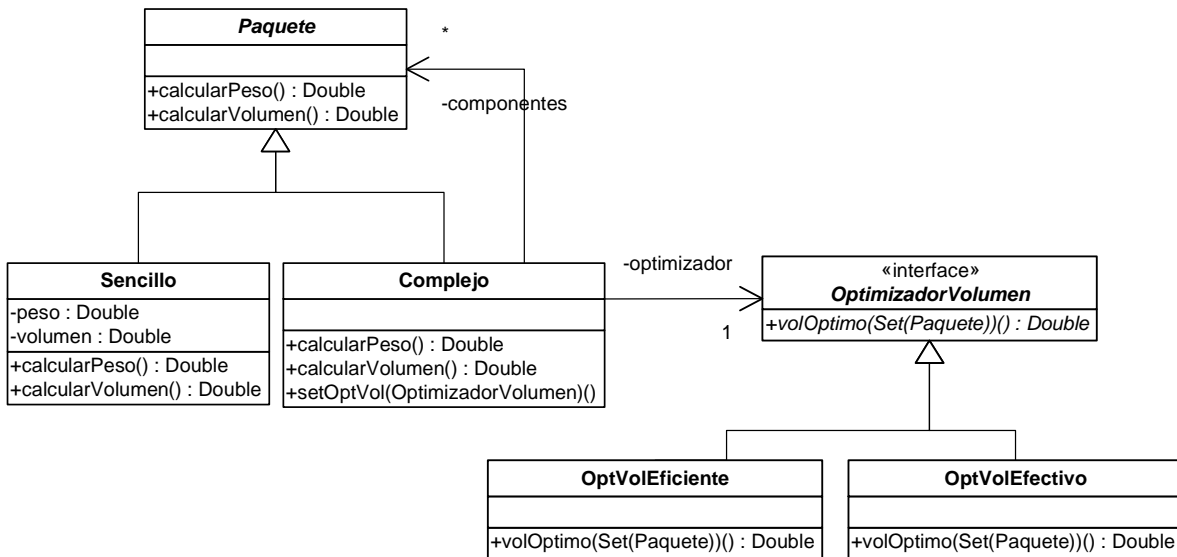
Problema 3 (30 puntos)

Parte A:

Implementar completamente en C++ una clase singleton denominada A, que tiene un atributo privado de tipo int, denominado dato.

Parte B:

Una empresa de servicios de transporte interurbano de objetos frágiles, ha encomendado el desarrollo de un sistema de apoyo a la gestión en el armado de los paquetes chicos y medianos. Los objetos son empaquetados en cajas, ya sea de forma individual o agrupados. Los paquetes así conformados, pueden a su vez ser parte de otros paquetes más grandes. La figura muestra el diseño de la estructura determinada para representar la organización de los paquetes.



Para organizar la carga en los vehículos, es de interés poder estimar el peso y volumen de los paquetes. El cálculo del peso es relativamente sencillo, mientras que el del volumen implica la ejecución de un algoritmo de optimización de ubicación de volúmenes, que tiene un alto costo computacional. Por este motivo, para la estimación del volumen de un paquete que está formado por otros, se utiliza un optimizador de volúmenes. Existen varias versiones de dicho optimizador: algunas estiman de mejor forma el volumen con un alto costo computacional (efectivas), mientras que otras hacen una estimación menos ajustada pero ejecutan de manera más rápida (eficientes).

A su vez, se deben tener en cuenta las siguientes consideraciones:

- El peso y el volumen de un paquete `Sencillo` se calculan directamente a partir de sus atributos.
- El peso de un paquete `Complejo` es la suma de los pesos de sus componentes.
- El volumen de un paquete `Complejo` se calcula invocando a la operación `valOptimo` de la interfaz `OptimizadorVolumen`, la cual recibe una colección con los componentes del paquete en cuestión.
- La construcción de un paquete `Complejo` se realiza a partir de sus componentes y de una instancia de una clase que implemente la interfaz `OptimizadorVolumen`.
- La destrucción de un paquete `Complejo` implica la destrucción de todos sus componentes.

Se pide:

Implementar completamente en C++ las clases `Paquete`, `Sencillo`, `Complejo` y la interfaz `OptimizadorVolumen`.

Observaciones:

- Puede suponer la existencia de la interfaz `ICollectionable` e implementaciones de `ICollection` (clase `List`) e `IEnumerator` según sea necesario.
- Es posible utilizar las clases `set<T>` o `vector<T>` de la STL.
- Las implementaciones deben incluir constructores, destructores y operaciones `set` y `get` donde corresponda.
- No incluir directivas al precompilador.