

Programación Avanzada

EXAMEN DICIEMBRE 2014

Por favor siga las siguientes indicaciones:

- Escriba con 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.

Problema 1 (40 puntos)

a) De desea desarrollar una plataforma que funciona principalmente en base a la comercialización de productos. Existen dos tipos de usuario que se registran a la plataforma: cliente y proveedor. El cliente se registra en la plataforma ingresando sus datos personales y luego podrá comprar productos. Además podrá realizar comentarios sobre los productos adquiridos intercambiando ideas y valoraciones con otros clientes. El proveedor se registra en la plataforma de igual manera que el usuario cliente, teniendo la posibilidad de publicar sus propios productos para comercializarlos.

De cada usuario interesa saber si es proveedor o si es cliente, su nickname (único), su nombre, su apellido, su fecha de nacimiento, su correo electrónico (único) y una imagen (opcional). En caso de ser proveedor interesa además registrar el nombre de la compañía y un link de acceso al sitio de la misma.

Como se mencionó anteriormente, un proveedor podrá publicar productos. De cada producto se conoce su nombre (único), su descripción, su especificación, su número de referencia (que lo identifica), su precio, su conjunto de imágenes (opcional) y su conjunto de categorías ("Android", "Samsung", etc.).

Las categorías se identifican por su nombre, teniendo la posibilidad de que cada categoría esté compuesta por otras categorías y así sucesivamente. A modo de ejemplo se muestran las siguientes composiciones de categorías:

- Celulares
 - SO
 - Android
 - Windows
 - Equipos
 - Samsung
 - Iphone
 - Accesorios
 - Protectores
 - Baterías

Un producto es publicado por un único proveedor y debe pertenecer al menos a una categoría.

Los clientes pueden adquirir los productos generando una orden de compra. De cada orden de compra se conoce su número (que la identifica), su fecha, su precio total y el cliente que la realiza. Una orden de compra puede incluir uno o más productos, de cada uno de los cuales se solicita una cantidad específica. Por ejemplo podría solicitar el producto "Samsung Galaxy S4 número de referencia 42 proveedor A" con cantidad igual a cinco y el producto "Nokia Lumia 1020 número de referencia 15 proveedor B" con cantidad igual a dos.

Se pide: Construya un **Diagrama de Modelo de Dominio UML** para la siguiente realidad. Las restricciones deben ser expresadas en lenguaje natural. Modele exclusivamente en base a la información presente en la descripción.

b) A continuación se presenta el **Caso de Uso Generar Orden de Compra**, el cual el equipo informático ha trabajado:

Caso de uso	Generar orden de compra
Actores	Administrador del Sistema
Descripción	El caso de uso implica crear una nueva Orden de Compra en el sistema para que luego el proveedor pueda procesarla. En primer lugar el sistema lista todos los clientes del sistema mostrando su email y nickname y el administrador selecciona uno. Luego el administrador pasa a seleccionar productos para agregar a la orden de compra. Este proceso está compuesto por los siguientes pasos: primero el sistema muestra una lista de categorías y el administrador selecciona una que contenga productos, luego el sistema muestra todos los productos de la categoría seleccionada, y el administrador selecciona el producto que se quiera agregar, indicando además la cantidad a agregar a la orden de compra. El proceso de seleccionar productos se realiza las veces que el administrador desee. Finalmente el sistema da de alta la orden de compra, mostrando el nick e email del cliente seleccionado, su fecha y hora (que se toma de la actual del sistema operativo), el número de orden auto-generado y la lista de productos seleccionados indicando para cada uno la cantidad elegida, su precio unitario y el precio total de la orden de compra.

Se pide:

Realice un **Diagrama de Secuencia del Sistema** para el Caso de Uso Generar Orden de Compra. Incluya los tipos de los parámetros y valores de retorno de las operaciones.

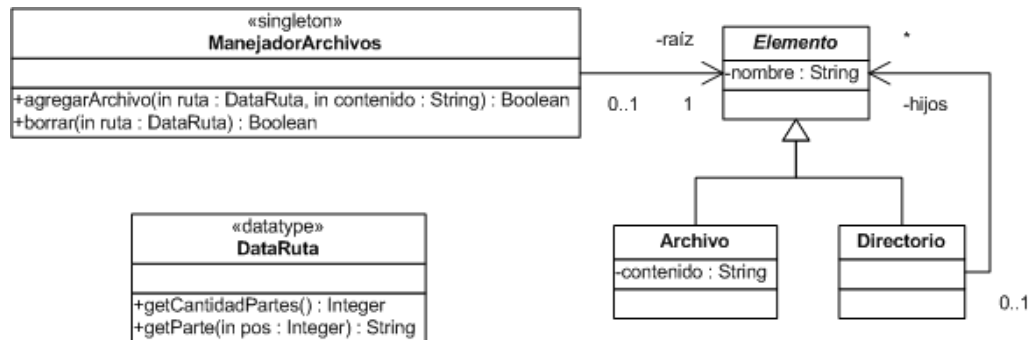
Problema 2 (30 puntos)

Dada la realidad planteada en el Ejercicio 1 se pide:

- a) Elabore el **Diagrama de Comunicación** correspondiente al **Caso de Uso Generar Orden de Compra**.
- b) Elabore el **DCD completo** (Diagrama de Clases del Diseño).

Problema 3 (30 puntos)

- a) Defina en UML las interfaces IDictionary, Iterator, IKey e ICollectible.
- b) Considere el siguiente Diagrama de Clases de Diseño parcial relativo a un sistema de manejo de archivos dentro de una estructura arborescente de directorios:



Para facilitar la tarea las rutas vienen dadas en el datatype **DataRuta** cuyas operaciones permiten acceder a cualquier trozo (o parte) de la misma. Las partes se numeran en el intervalo: 1..**getCantidadPartes()**. A modo de ejemplo, si la ruta fuera `/dir1/dir2/archivo`, las siguientes expresiones serían verdaderas:

- `getCantidadPartes() == 3`
- `getParte(1) == "dir1"`
- `getParte(3) == "archivo"`

Comportamiento esperado de las operaciones de la clase **ManejadorArchivo**:

- **agregarArchivo**: en caso de ser necesario se deben crear los directorios que falten. Dentro de un directorio no puede existir un archivo y un directorio con el mismo nombre. Devuelve `true` si logra crear el archivo, y `false` en caso contrario.
- **borrar**: borra tanto un directorio (recursivamente) como un archivo. No se puede borrar la raíz. Devuelve `true` si logra borrar el objetivo y `false` en caso contrario.

La estructura de objetos en memoria debe reflejar exactamente la estructura almacenada en disco. Para impactar los cambios hechos desde el manejador se cuenta con el **singleton**:



Tener en cuenta que los elementos pasados como parámetro deben estar correctamente ubicados en el árbol de objetos al momento de invocar tanto crear como borrar.

Se pide: Implemente en C++ completamente las clases del diseño anterior.

- No incluir el datatype **DataRuta** ni el singleton **UtilidadesIO**.
- Asuma que cuenta con una implementación de las interfaces definidas en la parte a) de este ejercicio.

Notas:

- No es necesario incluir directivas al preprocesador en el código.
- Asuma que cuenta con una implementación del datatype **String**.