

Programación 4

EXAMEN FEBRERO 2012

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)

Se desea realizar una aplicación para contabilizar la votación del concurso de carnaval uruguayo. Para ello se relevo la siguiente realidad:

La votación del concurso se realiza para distintas categorías de carnaval. Cada categoría está identificada por un nombre (Murga, Revista, Comparsa, etc.) y tiene una descripción de la categoría. Una categoría está compuesta por conjuntos que participan del concurso, los cuales tienen un nombre que los identifica y un puntaje total acumulado producto de la votación de los distintos miembros del jurado. El jurado de carnaval está integrado por doce personas, las cuales tienen un número único de jurado (de 1 a 12), un nombre y una especialidad. El jurado califica la actuación del conjunto según su especialidad que puede ser Coros, Humor, Vestuario, Actuación Global, etc. Cada integrante del jurado vota a todos los conjuntos con un puntaje (de 1 a 10). Para que el concurso sea más justo, existe una ponderación a los votos realizados por los jurados. El valor de ponderación se define para cada categoría por especialidad y es fija para todo el concurso. Por ejemplo, la categoría Murga tiene una ponderación 10 en Coros mientras que una Comparsa tiene ponderación 5 en coros.

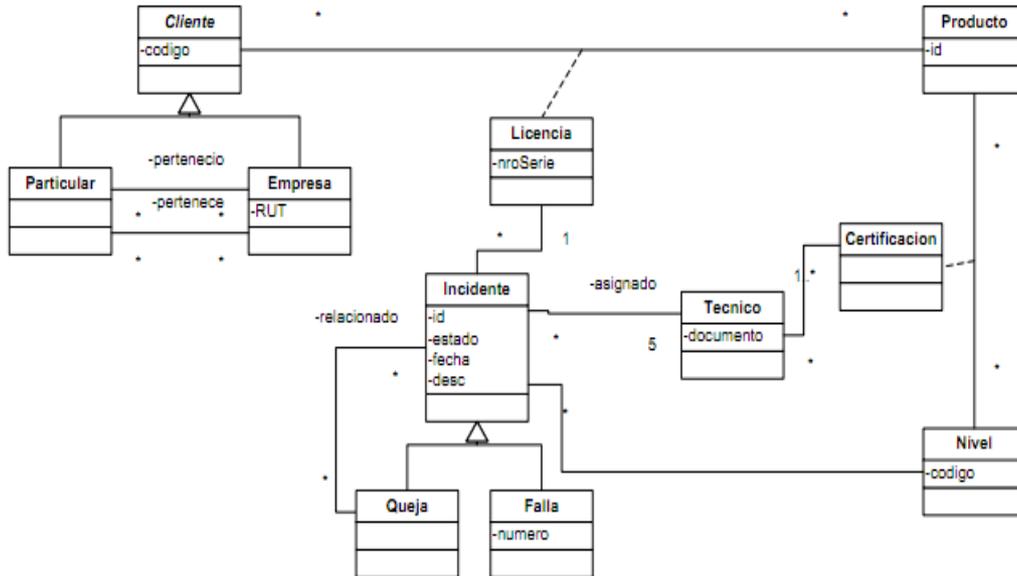
Dado que los conjuntos participan un máximo de tres rondas en el concurso, pueden llegar a obtener 3 votaciones distintas de cada jurado. Interesa saber cada voto que obtuvo el conjunto en cada ronda (por jurado y especialidad). El puntaje total acumulado de un conjunto corresponde al obtenido en todas las rondas que participó.

Se pide:

1. Modelar la realidad planteada mediante un Diagrama de Modelo de Dominio UML.
2. Expresar todas las restricciones del modelo en **lenguaje natural**.
3. En caso de que el jurado tenga más de una especialidad. Que modificaciones haría al modelo de 1). ?

Problema 2 (35 puntos)

Un sistema de gestión de incidentes se basa en el siguiente modelo de dominio para el registro y seguimiento de atención a sus clientes ante incidentes que puedan tener con sus productos. Para ello a cada incidente se le asigna un grupo de técnicos certificados tanto en el producto como en los niveles de complejidad que pueda tener el incidente.



Se pide:

- i. Realizar el Diagrama de Comunicación completo de las siguientes operaciones de consulta. Indicar claramente los parámetros y el tipo del resultado de todas las operaciones involucradas en su solución.

Operación	obtenerQuejasProducto(int idProducto, DateTime fechaInicio, DateTime fechaFin)
Descripción	Retorna una lista de identificadores de quejas para el producto con idProducto y con fecha entre el rango especificado.
Pre y Post	Pre: Existe un producto con identificador <i>idProducto</i> .

Operación	obtenerAsesores(int idQueja)
Descripción	Retorna una lista de técnicos (documento) que han participado en resolver incidentes relacionados a <i>idQueja</i> y que tengan al menos 3 certificaciones. Los incidentes relacionados deberán tener estado RESUELTO para ser considerados. Los técnicos recomendados no podrán ser los ya asignados a <i>idQueja</i> .
Pre y Post	Pre: Existe una queja con identificador <i>idQueja</i> . Post: Retorna un listado de documentos de los técnicos considerados como asesores.

- ii. Realizar el Diagrama de Clases de Diseño (DCD) correspondiente para las operaciones de i.
- iii. Se desea rediseñar el modelo de incidentes de manera que soporte distintas transiciones de estados dependiendo del tipo de incidente; e.g. si un incidente esta en estado PENDIENTE, dependerá de su tipo (Queja o Falla) el siguiente estado que tomará cuando se ejecute la operación **actualizar**. A demás se anticipa la inclusión de nuevos tipos de incidentes, estados y ajustes regulares en los métodos de transición entre estos. Asuma la existencia de los siguientes estados [PENDIENTE, PROCESANDO, RESUELTO, CERRADO].

Proponga un diseño que soporte este nuevo requerimiento minimizando el impacto en el sistema existente. En caso de utilizar algún patrón, indique cuál/es y los roles que cumplen los distintos elementos en su diseño.

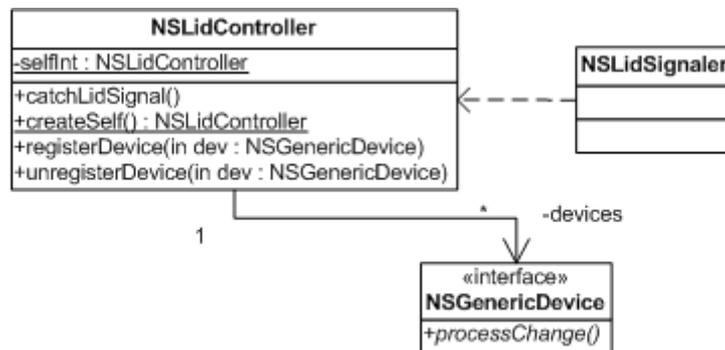
Problema 3 (35 puntos)

Se quiere desarrollar un controlador del sistema operativo que maneje funcionalidades de hardware de una notebook en función si su lid (tapa en castellano) está cerrada o abierta. Dicho controlador se llama LidController.kext y es una extensión del kernel programado en C++ que básicamente maneja la actividad de varios dispositivos de hardware que posee la maquina portátil, algunos al ser accionados por el usuario, según el estado del lid (abierto o cerrado); ejemplos de estos dispositivos pueden ser el medidor de batería, el Piloto LED de estado (idle, sleep, off), la iluminación del logo propietario, entre otros.

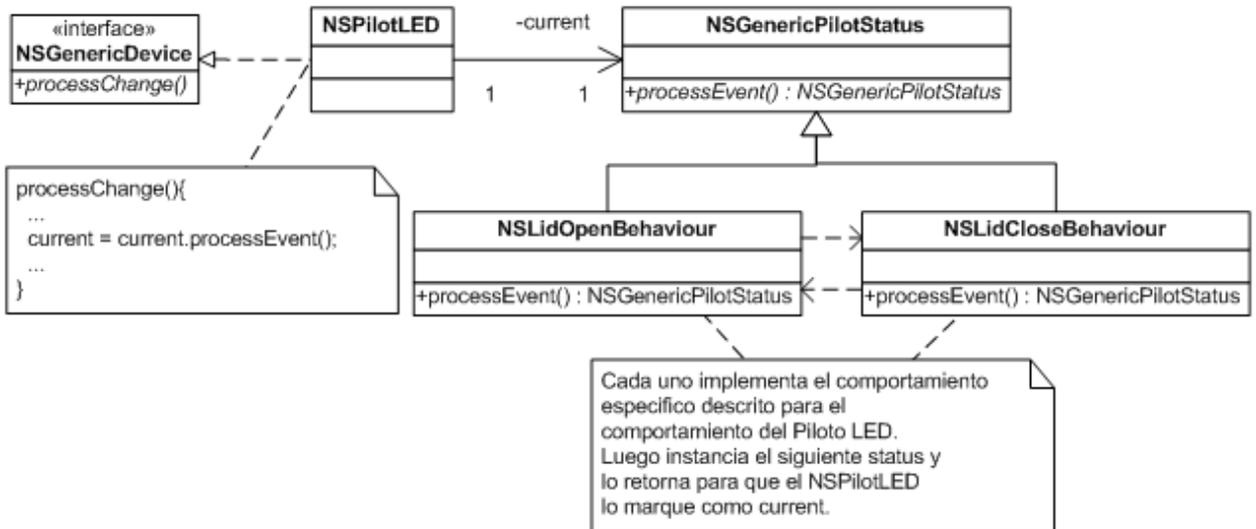
A continuación se describe el funcionamiento de los dispositivos:

- Medidor de batería – mediante un botón el usuario puede consultar cuanta batería le resta de uso al equipo (5 luces, medición de 1 a 5 donde 5 es batería llena).
- Piloto LED – muestra el estado del sistema operativo que corre el equipo, cuando está en uso el led se mantiene apagado, cuando el equipo entra en estado idle el LED se enciende (lid abierto) y al quedar en estado sleep el LED se enciende intermitentemente (automáticamente al cerrarse el lid).
- Logo – el logo se enciende solo cuando el lid está abierto y al cerrarse el mismo el logo se apaga.

En función de estos requerimientos se diseñó una clase llamada NSLidSignaler que se encarga de notificar el estado de la lid del notebook (bajo nivel, conectada al hardware, envía señales cuando la lid se cierra o se abre), un clase llamada NSLidController que recibe la señal del hardware y la propaga notificando a los interesados y una interface que permita tratar genéricamente a los dispositivos llamada NSGenericDevice ya que en el futuro se pueden agregar nuevos dispositivos. A continuación se muestra un diseño de la solución.



Por los requerimientos antes mencionados vale notar que el dispositivo Piloto LED (representado por la clase NSPilotLED) necesita un diseño particular para funcionar, el mismo tiene distintas funcionalidades en función de si la lid está abierta o cerrada. A continuación un diagrama de su diseño.



Se pide:

- i. Implementar completamente en C++ la Clase NSLidController. No es necesario incluir las directivas del preprocesador; asumir la existencia de implementaciones de ICollection, IList, IMap, IKey y las interface ICollectible en su solución.
- ii. Implementar completamente en C++ la interface NSGenericDevice y la clase NSPilotLED.
- iii. En caso de que existan, identificar los patrones de diseño utilizados en la solución. Para cada caso, indicar nombre y roles participantes (o en su defecto la colaboración completa de los mismos).