Programación 4

EXAMEN JULIO 2016

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 número de examen junto al examen.
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el parcial.

Problema 1 (35 puntos)

Se desea construir un sistema para facilitar la gestión de un taller mecánico.

De cada cliente del taller se conoce su número de cliente (que lo identifica), nombre, número de teléfono, email y dirección.

Un cliente posee uno o más vehículos que puede llevar para realizar servicios. De cada vehículo interesa registrar su número de matrícula, marca, modelo, año de fabricación, tipo de motor (nafta, diesel, eléctrico o híbrido) y descripción con detalles que puedan resultar de interés para el personal del taller. El número de matrícula identifica al vehículo. Interesa registrar todos los vehículos de cada cliente, independientemente de si se le realizaron servicios.

El taller cuenta con mecánicos, identificados por su número de empleado. Además, de cada mecánico se conoce su nombre, número de teléfono, email, dirección, servicio de emergencia y años de experiencia.

Un cliente puede tener uno o más mecánicos de referencia a quienes les puede pedir presupuestos por ciertos servicios a realizar a alguno de sus vehículos. Del presupuesto se desea registrar la fecha en que se realizó, el tipo de servicio presupuestado (puede ser reparación, chequeo o mantenimiento), el monto y el plazo estimado de realización del servicio. Cada presupuesto se identifica por un valor numérico. Además es realizado por un único mecánico de referencia. En base a ello, el cliente decidirá si realizar o no el servicio. Cada cliente podrá hacer una evaluación sobre el mecánico de referencia, que se representará por un valor numérico de 1 a 5. No interesa mantener un historial de las evaluaciones dadas por un cliente a un mecánico. La evaluación puede hacerse sin necesidad de que el mecánico haya presupuestado algún servicio.

Por último, es necesario mantener un registro de los servicios brindados a los vehículos. Interesa registrar el tipo de servicio (igual que para los presupuestos), la fecha de inicio, fecha de finalización, precio final, costo de materiales y mano de obra, y descripción. El servicio siempre se realiza por al menos un mecánico, y a un único vehículo. Puede tener uno o más presupuestos asociados, y no necesariamente los mecánicos que presupuestaron el servicio son los que lo realizaron luego.

Considere además el siguiente Caso de Uso:

Caso de Uso:	Registro de servicio
Actores:	Mecánico
Descripción:	El caso de uso comienza cuando el actor desea registrar un servicio realizado a un vehículo de un cliente. El actor ingresa el número de cliente y el sistema despliega los vehículos que posee. Si en el listado no aparece el vehículo al que se le realizó el servicio, el actor podrá registrarlo indicando su número de matrícula, marca, modelo, año, tipo de motor y descripción. El actor elige un vehículo, y luego indica los datos del servicio realizado. El sistema despliega una lista de presupuestos del mismo tipo que el servicio que se desea registrar, y que aún no se han asociado a otro servicio. El actor elige el o los presupuestos realizados para el servicio. El sistema despliega los mecánicos del taller, y el actor elige los que realizaron el servicio. El actor confirma el registro del servicio realizado, terminando el caso de uso.

Se pide:

i) Modelo de Dominio, con restricciones en lenguaje natural.

ii) Diagrama de Secuencia del Sistema para el caso de uso "Registro de servicio". Indicar explícitamente el uso o no de memoria. Se valorará el uso de comentarios que ayude al lector ante detalles importantes para su comprensión. También explicitar los tipos de datos.

<u>Problema 2</u> (30 puntos)

Parte a)

i) ¿Qué establecen las PRE y POST condiciones en un Contrato de Software?

ii) ¿Por qué se necesita de una fábrica (Factory) si se desea independizar la Capa Presentación de la Capa Lógica?

Parte b)

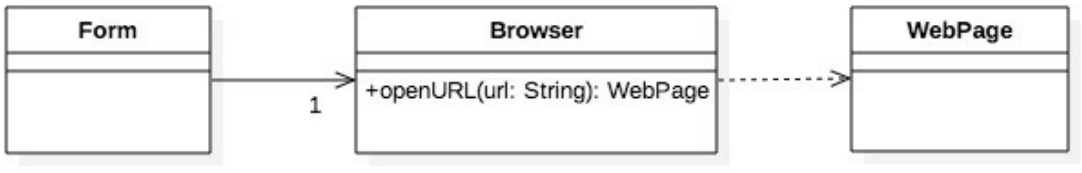
Una startup de dispositivos móviles se ha contactado con Ud. para solucionar un problema de diseño en su navegador web móvil. Sucede que el mismo demora mucho en cargar las páginas web, y el motivo es que no cuenta con un sistema de caché, es decir, que no guarda las últimas páginas web visitadas, por lo que permanentemente carga cada página de cero.

Si bien la startup podría diseñar un sistema de caché, dado que estos sistemas pueden cambiar con nuevas versiones del navegador (ej: incorporando nuevas prácticas de "cacheado" de paginas) se prefiere dejar intacta la lógica de su navegador y

realizar el diseño del sistema de caché en forma separada, de manera que ambos puedan evolucionar independientemente.

El sistema de caché funciona de la siguiente manera: antes de abrir una URL, se debe buscar si ya no se encuentra almacenada en el sistema (lo que se conoce como una página "cacheada") en cuyo caso se devuelve la página cacheada; de lo contrario se carga la página web de cero y se la almacena en el sistema de caché para futuras peticiones.

El diseño actual es el siguiente:



Se pide:

i) ¿Qué patrón(es) de diseño propondría para diseñar el sistema de caché de paginas sin modificar la clase Browser? Justifique brevemente su respuesta, indicando además los participantes de cada patrón propuesto.

ii) Realice el nuevo Diagrama de Clases incluyendo la(s) clase(s)/interfaz(ces) que sean necesarios para el/los patrones propuestos.

iii) Realice el Diagrama de Comunicación que muestra las interacciones necesarias para que el formulario solicite abrir una URL, utilizando el sistema de caché diseñado.

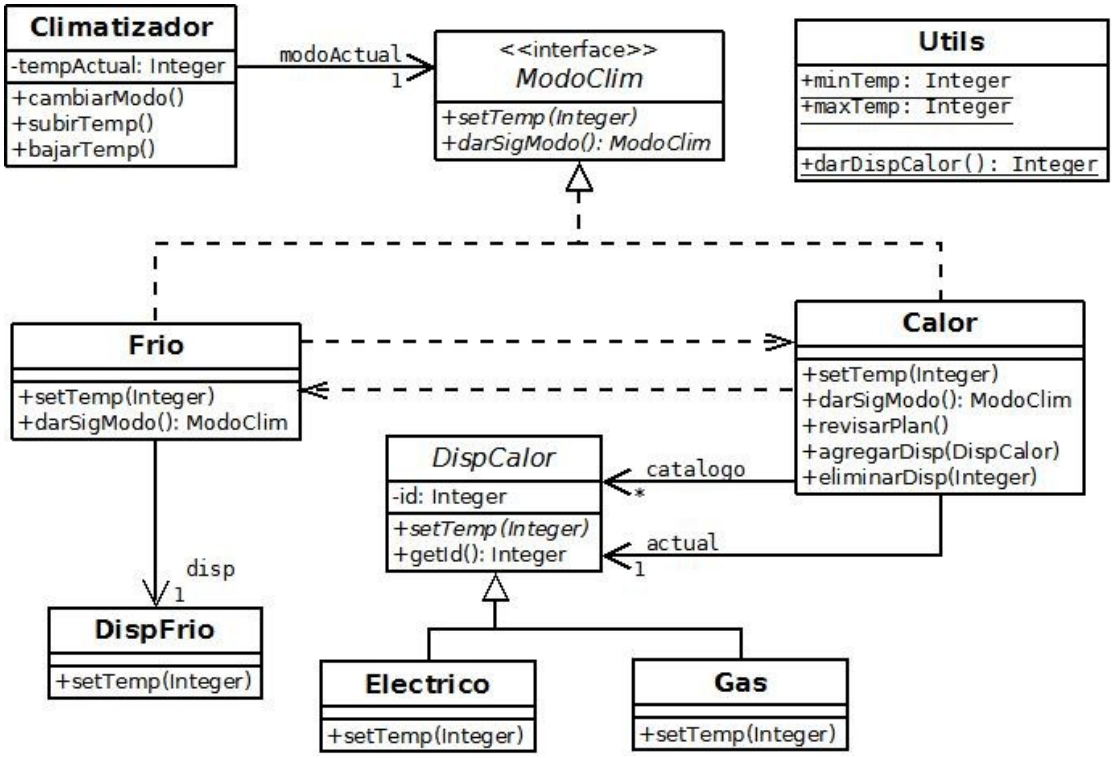
<u>Problema 3</u> (35 puntos)

Una empresa fabricante de sistemas climatizadores para hogares está diseñando un nuevo modelo cuya función es mantener una temperatura ambiente establecida, tanto enfriando como calentando, utilizando diferentes dispositivos. Para enfriar utiliza un sistema clásico de aire acondicionado mientras que para calentar tiene la capacidad de controlar un catálogo de diferentes dispositivos. De esta forma, un hogar puede disponer de diferentes métodos de calefacción (por ejemplo, gas, eléctrico), cuyos dispositivos son controlados de forma central y su uso puede alternarse de forma transparente para el usuario y haciendo un uso eficiente de la energía según la hora del día, costos, etc.

La figura muestra el Diagrama de Clases de Diseño del sistema, donde:

- La clase **Climatizador** es el punto de entrada de control para el usuario. Cuenta con operaciones para subir y bajar temperatura (en valores enteros unitarios) y una operación para alternar entre los modos frío y calor.
- Las clases **Frio** y **Calor** implementan cada modo del climatizador. Cada una tiene la capacidad de generar una nueva instancia de la otra.

- La clase **Calor** maneja un catálogo de dispositivos y setea el dispositivo actual mediante la operación **revisarPlan**, que se invoca automáticamente cada cierto intervalo de tiempo establecido.



Se pide:

i) Identificar el patrón o los patrones de diseño utilizado(s). Para cada uno establecer claramente los roles en términos del patrón, de las clases de la realidad planteada.

ii) Implementar en C++ completamente (.h y .cpp) las clases **Climatizador**, **ModoClim**, **Frio**, **Calor** y **DispCalor**. Se pide implementar exclusivamente las operaciones que están en el diagrama. No implementar getters (salvo **getId** en **DispCalor**), setters, constructores ni destructores; asumir que estas operaciones tienen sus implementaciones habituales.

Consideraciones:

- Las operaciones **subirTemp** y **bajarTemp** en **Climatizador**, deben lanzar una excepción de tipo **invalid_argument** si se intenta setear una temperatura fuera del rango **[minTemp,maxTemp]** indicado por los atributos públicos de clase, de la clase **Utils**.
- La operación **revisarPlan** en **Calor**, hace uso de la operación de clase **darDispCalor** de la clase **Utils**, que devuelve un identificador del dispositivo de calor más eficiente según las condiciones del entorno en ese momento (hora del día, condiciones climáticas externas, etc).
- Puede utilizar la clase **map** de STL o la clase **IDictionary** más las correspondientes clases auxiliares que necesite para dicha colección.
- No incluir directivas al precompilador.