UNIVERSIDAD RICARDO PALMA **ESCUELA DE INGENIERÍA INFORMÁTICA**
FACULTAD DE INGENIERIA **Ciclo 2012-I**

Examen Parcial

CURSO: Arquitectura de software
GRUPO(s): Todos

PROFESOR(s): Miguel Arrunátegui
FECHA: 10 de Mayo de 2012
HORA: 17:45
DURACION DEL EXAMEN: 2 horas
DEVOLUCION DEL EXAMEN CALIFICADO/ Fecha, hora y aula: 21-5-2012 En clase de Teoría
IMPORTANTE:

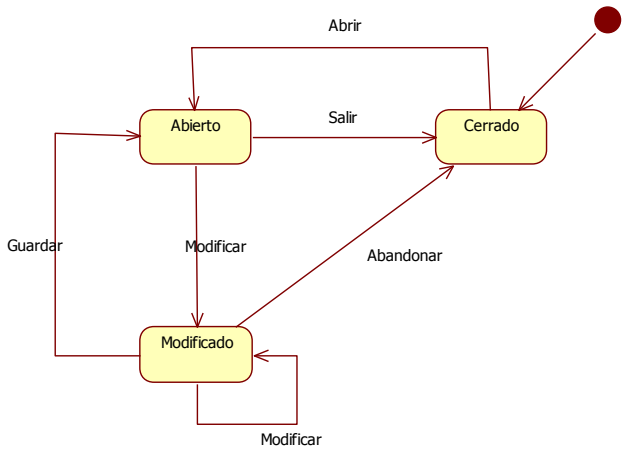
- El examen es sin copias ni apuntes.
- Está prohibido el préstamo de calculadoras, correctores, uso de celulares, consumo de bebidas, comidas y cigarrillos.

Nombre:		Número de máquina:		Nota:	

Pregunta 1 (2 Puntos, Word) ¿En cuál de las fases del RUP, la labor del Arquitecto es más importante y por qué?

Pregunta 2 (3 Puntos, Word) Explique cuáles son los inconvenientes (reales, potenciales y futuros) de un sistema no trivial que no tiene arquitectura.

Pregunta 3 (4 Puntos, Aquí mismo) Respecto al patrón Estado (State) de la entidad **Documento**. De acuerdo al siguiente diagrama de estados de la entidad **Documento**, responda lo siguiente:



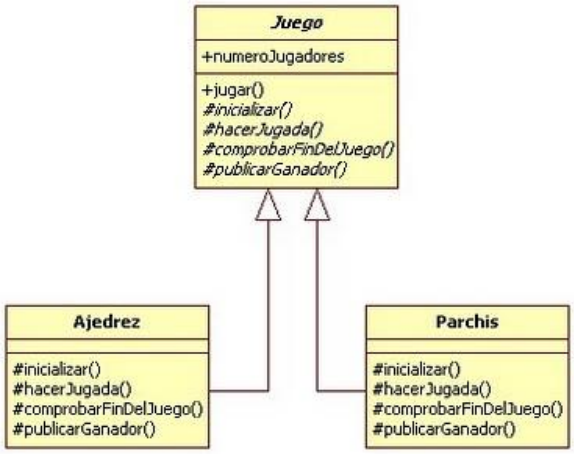
	Verdadero	Falso
El Patrón estado tiene las clases: Abrir, Salir, Modificar, Guardar y Abandonar	☐	☐
Las clases **Estado** concretas (las que heredan de **EstadoAbstracto**) tienen los métodos: Abrir, Salir, Modificar, Guardar y Abandonar	☐	☐
El método Guardar de la clase **Abierto**, no debería hacer nada.	☐	☐
El método Abierto de la clase **Guardar** no debería hacer nada.	☐	☐
La clase **EstadoAbstracto** tiene 5 métodos abstractos.	☐	☐

Pregunta 4 (6 Puntos, Carpeta de proyecto) Implementar un patrón nuevo: **Template method**

Patrones de diseño: Template Method

Este patrón define el funcionamiento general de un algoritmo en una operación de una clase, delegando en otras clases hijas (sub-Clases), a través de herencia, la implementación específica de al menos uno de los pasos del algoritmo. Permite que estas sub-Clases redefinan ciertos pasos del algoritmo sin cambiar su comportamiento general, adaptándose cada una de ellas a una necesidad específica.

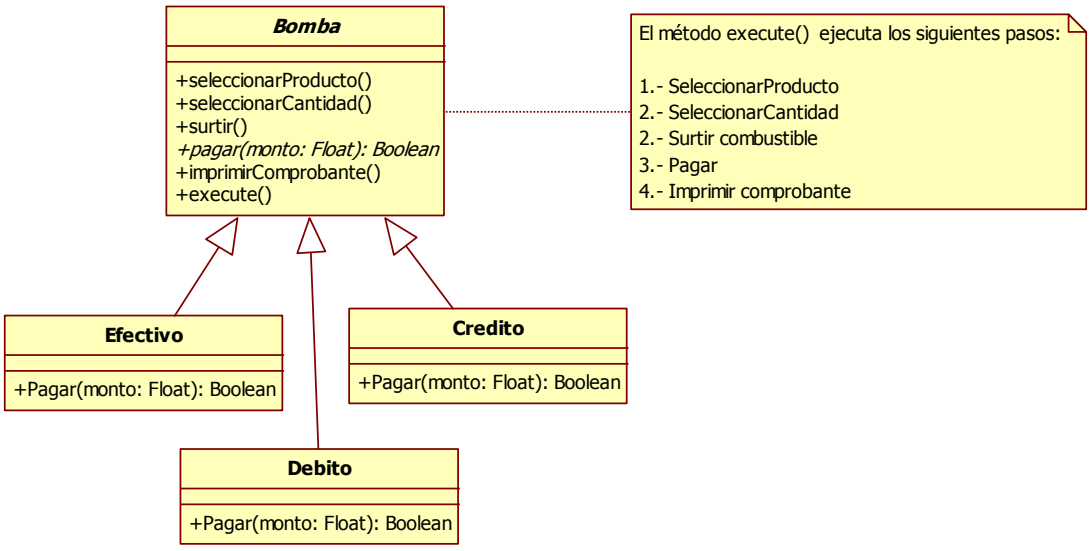
Una implementación clásica de este patrón es la que aparece en el libro "Design Patterns" del GOF, que presenta una **clase abstracta** en la que se define el comportamiento general del algoritmo en un método concreto **execute()** que llama a su vez a otros métodos abstractos en los que se definen comportamientos más específicos. Distintas subclases pueden heredar de esta clase y especificar comportamientos diferentes para los métodos abstractos, permitiendo varias implementaciones que se puedan ajustar a cada necesidad.



En la figura anterior, una clase Juego define en su método jugar() el comportamiento general de cualquier juego. El juego comienza y cada uno de los jugadores va haciendo su jugada hasta que se cumple una condición que da fin al juego y se muestra el resultado. Estas operaciones se indican mediante métodos abstractos que deben ser implementados por cada juego específico de forma apropiada, como pueden ser el Parchis o el Ajedrez.

Se pide: Haga un programa con interface gráfica, que permita aplicar el patrón Template Method en el escenario de una Estación de despacho de combustible (Bomba) de un grifo. El algoritmo de despacho se ejecuta mediante el método execute() y se ejecuta igual para todos los casos excepto por la forma de pago. Simular en la GUI, el despacho de gasolina y la selección del tipo de pago. Cuando se ejecute el despacho, se deberá mostrar el mensaje "Pagando consumo en {Efectivo| Crédito|Débito} según sea el caso.

El diseño es el siguiente:



Diseñe libremente la GUI a su criterio.

Pregunta 5 (5 Puntos, Carpeta de proyecto) Implementar la interface **IMotor** según el diseño UML mostrado. En el programa cliente (form1), instanciar la clase apropiada según el radioButton y ejecutar la carga de combustible del motor según la cantidad que se tipee en el textbox y mostrar luego el Costo incurrido.