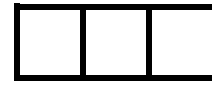


**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**  
**CCPG1009 – DISEÑO DE SOFTWARE**  
**SEGUNDA EVALUACIÓN - II TÉRMINO 2018**

**Nombre:** \_\_\_\_\_ **Paralelo:** \_\_\_\_\_

COMPROMISO DE HONOR: Al firmar este compromiso, reconozco que el presente examen está diseñado para ser resuelto de manera individual, que puedo usar un lápiz o esferográfico; que sólo puedo comunicarme con la persona responsable de la recepción del examen; y, cualquier instrumento de comunicación que hubiere traído, debo apagarlo y depositarlo en la parte anterior del aula, junto con algún otro material que se encuentre acompañándolo. Además, no debo usar calculadora alguna, consultar libros, notas, ni apuntes adicionales a los que se entreguen en esta evaluación. Los temas debo desarrollarlos de manera ordenada.  
Firmo el presente compromiso, como constancia de haber leído y aceptado la declaración anterior. "Como estudiante de ESPOL me comprometo a combatir la mediocridad y actuar con honestidad, por eso no copio ni dejo copiar".

\_\_\_\_\_  
Firma



100

\_\_\_\_\_  
Firma

**TEMA 1 – CONCEPTOS**

**(30 PUNTOS)**

**Por cada ítem, seleccione o escriba la(s) respuesta(s) correcta(s), según considere conveniente. En caso de ser falso, debe justificar su respuesta.**

- 1) Indique cual(es) de los siguientes **NO** son malos olores de programación (code smells): (3 pt)
  - a. Clase de datos
  - b. Método largo
  - c. Métodos paralelos
  - d. Cambio convergente
  - e. Legado forzado
  
- 2) Indique una recomendación para evitar el mal olor de obsesión primitiva o "Primitive Obsession". (2 pt)
  
  
  
  
  
  
  
  
  
  
- 3) Describa como reconocer si hay presencia del mal olor llamado "Data clumps": (4 pt)
  
  
  
  
  
  
  
  
  
  
- 4) El patrón de diseño \_\_\_\_\_ tiene como propósito principal, reutilizar una clase que no puede ser modificada, pero que tiene una interfaz incompatible con el sistema que se está desarrollando. (2 pt)
  
  
  
  
  
  
  
  
  
  
- 5) Describa un escenario en el que se pueda utilizar el patrón de diseño Builder en conjunto con Prototype. (4 pt)
  
  
  
  
  
  
  
  
  
  
- 6) Según el Test Driven Development, el procedimiento para corregir un fallo encontrado en el código es: corregir el error y luego generar nuevas pruebas para validar el cambio realizado. (3 pt)
  - a. Verdadero. Porque:
  - b. Falso. Porque:

- 7) Si las pruebas unitarias incrementan el tiempo de desarrollo de un sistema. ¿Por qué se aconseja realizarlas? (3 pt)
- 8) Las pruebas unitarias pueden evitar que el sistema desarrollado esté libre de fallos (bugs). Explique. (3 pt)
  - a. Verdadero.
  - b. Falso. Porque:
- 9) Explique bajo que consideraciones debe aplicar la técnica de refactorización **Reemplazar delegación con herencia "Replace Delegation with Inheritance"**. (3 pts)
- 10) Explique las ventajas de utilizar el patrón de diseño **Cadena de responsabilidad "Chain of Responsibility"**. (3 pts)

## TEMA 2 – REFACTORING

(30 PUNTOS)

- 11) Dado el siguiente código, identifique los malos olores (5 pt), explique por qué sería un problema (10 pt) y refactorice (15 pt) el código.

```

1 public class Bar {
2     public static final String ONE_BEER = "pilsener";
3     public static final String ONE_CIDER = "grans";
4     public static final String A_PROPER_CIDER = "strongbow";
5     public static final String GT = "gin_tonic";
6     public static final String BACARDI = "bacardi_special";
7     /**calcula el costo de las bebidas pedidas */
8     public int cal_co(String drink, int amount, boolean student) {
9         //Solo puede comprar máximo 2 bebidas fuertes.
10        if (amount > 2 && (drink.equals(GT) || drink.equals(BACARDI_SPECIAL))) {
11            throw new RuntimeException("Demasiadas bebidas, max 2.");
12        }
13        int price;
14        if (drink.equals(ONE_BEER)) { price = 74; }
15        else if (drink.equals(ONE_CIDER)) { price = 103; }
16        else if (drink.equals(A_PROPER_CIDER)) { price = 110; }
17        else if (drink.equals(GT)) {
18            price = ingredient6() + ingredient5() + ingredient4();
19        }else if (drink.equals(BACARDI)) {
20            price = ingredient6()/2 +ingredient1() +ingredient2() +ingredient3();
21        }
22        else { throw new RuntimeException("No such drink exists"); }
23        // Los estudiantes tienen descuento para bebidas suaves
24        if (student && (drink.equals(ONE_CIDER) || drink.equals(ONE_BEER)
25            || drink.equals(A_PROPER_CIDER))) {
26            price = price - price/10;
27        }
28        //Hay un descuento especial para GT y estudiantes
29        if (amount > 2 && drink.equals(GT) && student){ price *= 0.90; }
30        return price*amount;
31    }
32    private int ingredient1() { return 65; }
33    private int ingredient2() { return 10; }
34    private int ingredient3() { return 10; }
35    private int ingredient4() { return 10; }
36    private int ingredient5() { return 20; }
37    private int ingredient6() { return 85; }
38 }

```

12) Se solicita lo siguiente:

- Identificar los patrones de diseño que puede aplicar para implementar lo solicitado. Explique. (10 pt)
- Diagrama de clases que represente el diseño del módulo especificado, aplicando los patrones de diseño de considere apropiados. No olvide indicar multiplicidades, visibilidad de atributos y métodos y de indicar claramente si las entidades corresponden a clases concretas, abstractas o interfaces. (25 pt)
- Señale en su diagrama mediante paquetes el lugar en donde está implementando el patrón. (5 pt)

**Requerimiento:**

En el sistema de compras para politécnicos después de realizar la búsqueda de productos se muestra un listado con los resultados desde donde se permite seleccionar uno de ellos y proceder a la vista detallada del producto. Cada artículo mostrado debe tener nombre, imagen, categoría, precio, tiempo máximo de entrega y calificación del producto y del vendedor (Escala de 5 estrellas para cada uno). Asimismo, deberá permitir la compra de dicho producto. El pago se debe realizar mediante saldo registrado para el usuario actual o también puede pagar con efectivo en los quioscos donde corre el sistema. El usuario debe elegir una sola forma de pago. Una vez realizado el pago, se debe notificar automáticamente al vendedor de dicho producto, para que este realice la entrega en el lugar donde se realizó la compra (quiosco). Considere que sólo le puede comprar varios ítems pero del mismo producto.