

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y
COMPUTACIÓN
CCPG1009 – DISEÑO DE SOFTWARE
TERCERA EVALUACIÓN - I TÉRMINO 2019

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 deajo copiar". | _____ 100 |
| | _____ Firma |

TEMA 1 – CONCEPTOS

(25 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) En qué consiste el paradigma de diseño orientado a aspectos. Explique: (3 pt)

- 2) En la orientación a aspectos hay un proceso importante realizado por el WEAVER, en qué consiste este proceso: (3 pt)

- 3) Explique para qué sirve utilizar Docker dentro del desarrollo y despliegue de un sistema. (4 pt)

- 4) Explique las desventajas del patrón de diseño Builder. (4 pt)

- 5) Dibuje el diagrama de clases del patrón de diseño Observer. (5 pt)

- 6) Dibuje el diagrama de clases completo que muestre la siguiente relación: (6 pt)
Un programador puede pertenecer a 1, 2 o 3 proyectos, pero cada proyecto tiene por lo menos 5 programadores. Además, un programador es una persona (nombre, apellido, email, cédula), que posee una lista de proyectos y un método para mostrar a que proyectos pertenece.

TEMA 2 – PRINCIPIOS DE DISEÑO

(25 PUNTOS)

Dado el siguiente código, identifique los principios SOLID que se está violando, luego indique cuales son los malos olores de programación que tiene el código, explique la razón y corrija el código de tal forma que ya no se viole ningún principio y no tenga ningún mal olor. Puede crear las interfaces y clases que considere necesarias para la refactorización. **SOLID: 5 pt, Smells: 5 pt, Explicar: 5 pt y Refactoring: 10 pt.**

```
1 public enum ModoConduccion { AUTOMATICO, MANUAL }
2 public enum TipoVehiculo { TURISMO, DEPORTIVO, RALLY}
3 public class Vehiculo {
4     int maxPotencia;
5     int rpm;
6     int maxCombustible;
7     int restanteCombustible;
8     int velocidades;
9     String name;
10    String serie;
11    int colorR;
12    int colorG;
13    int colorB;
14    ModoConduccion modo;
15    TipoVehiculo tipo;
16    public Vehiculo(String nombre, String serie, int fuel,
17        int power, int vel){
18        this.serie =serie;
19        name = nombre;
20        maxCombustible = restanteCombustible = fuel;
21        maxPotencia = power;
22        velocidades = vel;
23        colorR = colorG = colorB = 255;
24        rpm = 1;
25        modo = MANUAL;
26        tipo = TURISMO;
27    }
28    public ModoConduccion getModo(){ return modo; }
29    public void setModo(ModoConduccion mode){modo = mode;}
30    public TipoVehiculo getTipo(){return tipo;}
31    public void setTipo(TipoVehiculo type){ tipo = type; }
32    public int getMaxPotencia() { return maxPotencia; }
33    public int getMaxCombustible() { return maxCombustible; }
34    public int setCombustible(final int fuel) {
35        if (this.maxCombustible >= fuel){
36            this.restanteCombustible = fuel; }
37        else{ restanteCombustible = maxCombustible; }
38    }
39    /** Permite acelerar el vehiculo de acuerdo a su
40     * modo de conducción **/
41    public void _acc_for_mod(){
42        if(rpm < maxPotencia){
43            switch (this.modo){
44                case AUTOMATICO:
45                    //Verifica que no se pase de su máxima potencia
46                    if(maxPotencia > rpm + 2 && restanteCombustible > 1 ){
47                        rpm += 2;
48                        restanteCombustible--;
49                    }
50                    //Si ya no hay combustible el carro se debe detener
51                    if(restanteCombustible == 0 ){ rpm = 0; }
52                    break;
53                case MANUAL:
54                    //Verifica que no se pase de su máxima potencia
55                    if(maxPotencia > rpm + 5 && restanteCombustible > 1 ){
56                        rpm += 5;
57                        restanteCombustible--;
58                    }
59                    //Si ya no hay combustible el carro se debe detener
60                    if(restanteCombustible == 0 ){ rpm = 0; }
61                    break;
62            }
63        }
64    }
65    public void inicializar(){
66        switch (this.tipo){
67            case TURISMO:
68                this.setMaxPotencia(50);
69                this.maxCombustible(1000);
70                this.rpm = 1;
71                this.velocidades = 5;
72                this.acelerar();
73                break;
74            case DEPORTIVO:
75                this.setMaxPotencia(80);
76                this.maxCombustible(700);
77                this.rpm = 1;
78                this.velocidades = 6;
79                this.acelerar();
80                break;
81            case RALLY:
82                this.setMaxPotencia(70);
83                this.maxCombustible(1100);
84                this.rpm = 1;
85                this.velocidades = 5;
86                this.acelerar();
87                break;
88            default:
89                this.setMaxPotencia(50);
90                this.maxCombustible(500);
91                this.rpm = 1;
92                this.velocidades = 5;
93                this.acelerar();
94                break;
95        }
96    }
97 }
98 public class Carreras{
99     Run|Debug
100    public static void main(String[] args) {
101        ArrayList <Vehiculo> autos;
102        for(int i = 0 ; i< 25 ; i++){
103            Vehiculo v = new Vehiculo("Auto_" +i,
104                "serie0"+i, 0,0,0);
105            v.setTipo(DEPORTIVO);
106            v.colorR = Random.nextInt(0,255);
107            v.colorG = Random.nextInt(0,255);
108            v.colorB = Random.nextInt(0,255);
109            System.out.println(v.name + " ("
110                + v.colorR + "," + v.colorG + "," +
111                v.colorB + ")");
112            autos.add(v);
113        }
114        // Iniciar la carrera
115        for (Vehiculo v : autos) {
116            v.inicializar();
117        }
118        //Proceso para identificar quien
119        //gana la carrera
120    }
121 }
```

TEMA 3 – APLICACIÓN DE PATRONES

(50 PUNTOS)

A usted se le ha solicitado elaborar el diseño de un sistema, considerando los siguientes requerimientos:

La empresa familiar "CasaSoñada" está dedicada hace varios años a la construcción de casas dentro de urbanizaciones cerradas, las casas que ellos ofrecen son de 4 tipos, lo que principalmente cambia es la cantidad de metros cuadrados y habitaciones, pero poseen baños, número de pisos, grifería, patio, piscina, walking closet, cocina americana, gabinetes de cocina, terreno. La casa Esmeralda tiene 85 mts² y 2 habitaciones, la casa Platino tiene 115 mts² y 3 habitaciones, la casa Europa tiene 90 mts² y 3 habitaciones y finalmente la casa Primavera tiene 140 mts² y 4 habitaciones. Todas las casas vienen con acabados sencillos, sin embargo, los clientes pueden pedir otros tipos de acabados, por ejemplo, se puede cambiar el piso de porcelanato por mármol, la grifería se puede cambiar por una italiana, a las puertas se les puede agregar seguridad Multilock, entre otros. Cada acabado tiene un costo adicional al precio final de la casa.

Usted ha sido designado para el diseño de una solución que permita a los clientes realizar una proforma, así cuando tengan una idea más clara de lo que desean, pueden recibir ayuda de uno de los vendedores. El sistema debe permitirle al usuario poder seleccionar todo lo que desee, sin embargo, si el usuario no está autenticado, no le podrá mostrar el valor presupuestado total, para poder mostrarlo, el usuario debe registrarse en la aplicación. Los vendedores, pueden ver el listado de clientes que tiene más de 15 minutos utilizando el sistema y pueden contactarlos (llamándolos por teléfono) para brindar su ayuda. Este mismo sistema debe permitir, al gerente, obtener el reporte de los clientes interesados en una casa de forma semanal para contrastarlo con las ventas realizadas.

Elaborar lo siguiente:

1. Diagrama de casos de uso. (8 pt)
2. Descripción del caso de uso: (4 pt)
 - a. Ver valor total de proforma usuario autenticado.
3. El diagrama de secuencias: (10 pt)
4. Diagrama de clases. (20 pt)
 - a. Especifique multiplicidades, relaciones, visibilidad de métodos y atributos.
 - b. Aplique patrones de diseño, identificando cada patrón usado con subpaquete.
5. Implemente 2 pruebas unitarias para el cálculo del total de una proforma. (8 pt)