

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**  
**CCPG1009 – DISEÑO DE SOFTWARE**  
**PRIMERA 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 dejo copiar".	_____ 100
	_____ Firma

**TEMA 1 – CONCEPTOS** ( \_\_\_\_ / 35 PUNTOS)

- 1) Complete. Dentro de las tendencias DevOps se suele escuchar a varias herramientas que trabajan en conjunto. Indique los nombres de las siguientes herramientas, de acuerdo con la descripción. 4pts.
  - a. \_\_\_\_\_, permite ejecutar software dentro de un contenedor con toda la configuración necesaria para que funcione correctamente.
  - b. mientras que \_\_\_\_\_ permite administrar varios contenedores de forma automática.
- 2) Elija el literal que explique de mejor manera a que se refiere el principio de diseño de Software: Cohesión. 2pts.
  - a. Es como se agrupa partes del código muy relacionadas entre sí.
  - b. Hace referencia a la forma en la que separamos partes del código.
  - c. Es el grado de interdependencia de diferentes partes del código.
  - d. Es la medida en que un objeto/módulo se relaciona con otro.
- 3) Explique la(s) diferencia(s) que hay entre el modelo y el controlador del patrón de arquitectura MVC. 3pts.
  
- 4) Complete. La mayor similitud entre las herramientas Puppet y Ansible es que permiten manejar \_\_\_\_\_ como código. 3pts.
- 5) En la Orientación a Eventos. ¿Se permite que un evento sea atendido por múltiples suscriptores, siempre y cuando sea lanzado únicamente por un sólo publisher? Asevere o niegue argumentadamente su respuesta incluso puede proveer ejemplos. 4pts.
  
- 6) Explique la principal ventaja del paradigma de Diseño por Contrato. 4pts.
  
- 7) Tangling (enredando) se refiere al problema cuando varios requerimientos se encuentran implementados en una misma porción de código. 3pts.
  - a. Verdadero.
  - b. Falso (Justifique)
- 8) Tangling (enredando) es un problema del paradigma orientado a Aspectos. 2pts.
  - a. Verdadero.
  - b. Falso.
- 9) En un diagrama de secuencias se muestran varios módulos del sistema, indicando la secuencia entre ellos. 3pts.
  - a. Verdadero.
  - b. Falso (Justifique)
- 10) Complete. Una de las características de los aspectos, en la orientación a aspectos, es que se puede reutilizar un mismo advice varias veces. La sentencia que permite esto es \_\_\_\_\_. 3pts.
- 11) Complete. Algunas herramientas permiten generar código fuente a partir de los diagramas clases y viceversa. A estos procesos se los llama \_\_\_\_\_ y \_\_\_\_\_. 4pts.

## TEMA 2 – PRINCIPIOS DE DISEÑO

(\_\_\_\_/25 PUNTOS)

Dado el siguiente código considere el caso en el que se desea agregar otro método de pago como PayPal. Identifique los **principios SOLID que se está violando**, explique la razón y corrija el código de tal forma que ya no se viole ningún principio. Usted puede crear las interfaces y clases que considere o cambiar nombres de métodos si necesario. Incluso puede utilizar **diagramas de clase**. **Identificar:** 6 pt, **Explicar:** 9 pt y **Solución:** 10 pt.

```
1 public class Pelicula {
2     private final String titulo;
3     private int codigoPrecio;
4     private Time minutoActual;
5     public Pelicula(String inTitulo, int code, Time minute) {
6         titulo = inTitulo;
7         codigoPrecio = code;
8         minutoActual = minute;
9     }
10    public int getCodigoPrecio() { return codigoPrecio; }
11    public void setCodigoPrecio(int arg) { codigoPrecio = arg; }
12    public String getTitulo() { return titulo; }
13    public int getMinutoActual() { return minutoActual; }
14    //No se permite setMinutoActual para las peliculas
15    public void play(){
16        // Reproduce la pelicula desde el minuto actual
17    }
18    public void pause(){
19        // Pausar la pelicula
20    }
21    public void buscarComentarios(String titulo){
22        // Buscar en internet informacion sobre la pelicula
23    }
24 }
25 ///////////////////////////////////////////////////MoviePPV.java//////////////////////////////////////
26 public class MoviePPV extends Pelicula {
27     private Boolean pagada;
28     public MoviePPV(String title, int code, Time t, boolean paid){
29         super(title, code, t);
30         pagada = paid;
31     }
32     public MoviePPV(int id){
33         // Cargar datos desde un archivo con id
34     }
35     public void play() throws PlayException{
36         if (pagada){
37             // Reproduce la pelicula
38         }else{
39             throw new PlayException("Esta pelicula aun no ha sido comprada");
40         }
41     }
42     public void reiniciar(){
43         // Modificar el minutoActual a 00:00 para ir al comienzo
44     }
45 }
46 ///////////////////////////////////////////////////PlayList.java//////////////////////////////////////
47 public class PlayList {
48     public static void main(String[] args) {
49         ArrayList<Pelicula> coleccion = new ArrayList<Pelicula>();
50         MoviePPV movie = new MoviePPV(int id);
51         coleccion.add(movie);
52         // Asuma que se cargan el resto de peliculas desde un archivo
53         while( coleccion.size() >0){
54             Player p = coleccion.remove(0);
55             p.play();
56         }
57     }
58 }
```

### TEMA 3 – DISEÑO ORIENTADO A OBJETOS

(\_\_\_\_/40 PUNTOS)

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

Un municipio desea implementar un sistema para venta de boletos de autobuses inter-cantoniales e inter-provinciales. Se tiene que cumplir con las siguientes funcionalidades.

Debe existir la posibilidad de crear una cuenta de usuario a cualquier persona, que permita almacenar sus datos: nombres y apellidos, cédula o pasaporte, email, teléfono. También debe permitirle guardar rutas preferidas, donde estas rutas indican la cooperativa de transporte, el lugar de partida o el lugar de destino, tomando en cuenta que los boletos vendidos inician o terminan en el terminal de transportes del municipio.

Para realizar una compra, si es un usuario registrado se puede escoger una de las rutas preferidas y luego elegir la fecha y hora del viaje y la cantidad de boletos. En cambio, si no está registrado se debe primero buscar el lugar de salida o de destino, luego escoger la cooperativa de transporte y posteriormente se debe escoger la fecha y hora del viaje y número de boletos que se desea comprar. En un viaje, cada uno de los boletos vendidos debe tener asociado un número de cédula/pasaporte distinto, de esta forma se puede conocer quien está en cada asiento antes de iniciar el viaje. Al finalizar la compra, siempre se generan los boletos impresos y la factura.

(\*) Justo antes de subir al autobús, el conductor ingresa al sistema e indica que va a iniciar el viaje, esto debe enviar a ejecutar un proceso de verificación de cédulas y pasaportes al servicio web del registro civil, de esta forma se puede verificar si alguno de los pasajeros tiene algún problema o impedimento para salir de la ciudad. Este proceso de verificación debe imprimir un listado con la identificación, nombres, foto y estado de cada pasajero según el registro civil.

También existe un vendedor de boletos para pasajeros que no desean utilizar los quioscos automáticos, pero estos boletos tienen un recargo del 12% adicional en la venta de cada boleto. Además, un administrador puede administrar usuarios, cooperativas de transporte, horarios de viajes, boletos y generar reportes.

#### **Elaborar lo siguiente:**

1. Diagrama de casos de uso. (10 pt)
2. Especificación de un caso de uso del párrafo con (\*). (5 pt)
3. Diagrama de clases. Especifique:
  - a. Multiplicidades (2 pt)
  - b. Relaciones (10 pt)
  - c. Visibilidad de métodos y atributos (3 pt)
  - d. Paquetes MVC (5 pt)
4. Un diagrama de secuencias de objetos, el mismo caso de uso que especificó: (5 pt)