

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
CCPG1009 - DISEÑO DE SOFTWARE
PRIMERA EVALUACIÓN - II TÉRMINO 2019

Nombre: _____

Matrícula: _____

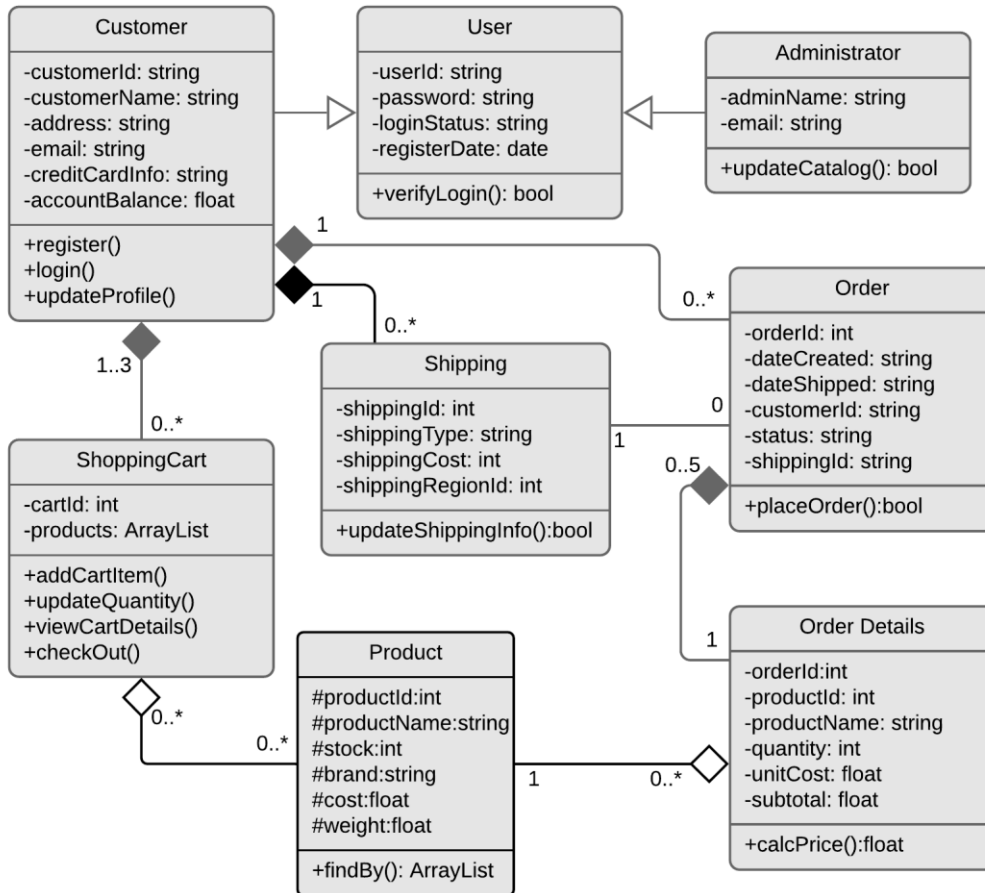
COMPROMISO DE HONOR: Al firmar este compromiso, reconozco que el presente examen está diseñado para ser resuelto de manera individual, que puedo usar un lápiz o esferográfico; que sólo puedo comunicarme con la persona responsable de la recepción del examen; y, cualquier instrumento de comunicación que 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

Sección A

1. Para el siguiente diagrama de clases de un sistema de compras en línea, indique si las siguientes afirmaciones son verdaderas o no encerrando con círculo la V o la F respectivamente. [12%]



- V – F: Una orden (*Order*) siempre debe tener un envío (*Shipping*) asociado.
- V – F: El sistema provee una funcionalidad para compartir un mismo carrito de compras (*ShoppingCart*) hasta entre 3 clientes (*Customers*).
- V – F: Si se desea eliminar un cliente (*Customer*), entonces se debe eliminar también todos los envíos (*Shipping*) asociados a este.
- V – F: Una orden (*Order*) del sistema puede tener hasta 5 productos distintos sin importar cuantas unidades se compren de cada uno.
- V – F: La clase *Administrator* puede tener un método para editar el *userId* y *password* heredados por la clase *User*, incluso si la clase *User* no provee métodos para editar dichos atributos.
- V – F: La clase *User* cumple con los principios SOLID de Single Responsibility y Open Close.

2. Complete. Una de las características de los aspectos, en la orientación a aspectos, es que se puede reutilizar un *advice*. La sentencia que permite esto se llama _____. [2%]
3. En el paradigma Orientado a Aspectos, **tangling** (enredado) se refiere al problema cuando varios requerimientos se encuentran implementados en un mismo método. Justifique su respuesta. [3%]
- Verdadero.
 - Falso.
4. En el paradigma de Diseño por Contrato existen 3 tipos de cláusulas, explique para qué sirve cada una. [3%]
- _____
 - _____
 - _____
5. Indique los nombres de las herramientas [DevOps] que permiten realizar las siguientes acciones: [4%]
- _____. Controlar la configuración de varios nodos (servidores) desde un nodo principal o master.
 - _____. Desde cada nodo (servidor), obtener las configuraciones que han sido realizadas o indicadas en el nodo principal o master.
 - _____. Crear contenedores virtuales con la configuración necesaria para que una aplicación funcione correctamente y que pueda ser portable a distintos sistemas operativos.
 - _____. Realizar pruebas de calidad sobre el código fuente de un proyecto.
6. Existen 3 etapas del diseño de software, entre las que tenemos el diseño detallado y el diseño de construcción. Explique en que se diferencian ambas etapas. Su respuesta debe ser una sola oración. [2%]
7. Describa 2 necesidades del paradigma de Estructurado que intenta suplir el paradigma Orientado a Objetos. [4%]
- _____
 - _____
8. Indique cuál de estas características NO pertenece a GIT: [2%]
- Control de acceso mediante distintos roles.
 - Manejar distintas versiones del código en paralelo.
 - Deshacer (cambios al código) ilimitado.
 - Combinar aportaciones de distintos colaboradores.
9. La cohesión es uno de los principios de diseño de software, indique si tiene alguna relación con alguno de los principios SOLID y por qué. [3%]

Sección B

10. Dado el siguiente código con cuatro clases, considere el caso en que se desea agregar una nueva forma de notificación (Telegram). Identifique **los principios SOLID que se están violando**. Para cada principio, explique la razón y corrija el código de tal forma que ya no se lo viole. Si lo considera necesario, usted puede crear interfaces, clases o nombres de métodos. Incluso puede utilizar **diagramas de clase**. Identificar [6%]. Explicar [9%]. Solución [10%].

```
public class Compra {
    private Pago pago;
    private PagoPayPal pagoPayPal;
    private List articulos;
    public Compra(Pago pago) {
        //inicializaciones
    }
    public Compra(PagoPayPal pagoPaypal) {
        //inicializaciones
    }
    public void agregarArticulo(Articulo articulo) {
        //agregar un articulo a la compra
    }
    public void removerArticulo(Articulo articulo) {
        //remover un articulo a la compra
    }
}
```

```
public class Notificacion {

    private int tipoNotificacion;

    public void notificar(Pago pago) {
        if(tipoNotificacion==1) {
            //enviarEmail();
        }
        else{
            //enviarSMS();
        }
    }
}
```

```
public class Pago {
    //atributos aquí
    public void realizarCobro(double monto) {
        //cargar el monto de compra al medio de pago
    }
    public void calcularImpuestosFactura() {
        //calcula los impuestos asociados a la compra
    }
    public void generarFactura() {
        //generar una nueva factura
    }
}
```

```
public class PagoPayPal extends Pago{
    private boolean loggedIn; //conexión a cuenta PayPal
    @Override
    public void realizarCobro(double monto) {
        if (!loggedIn) {
            return;
        }
        //cargar el monto de compra al medio de pago
    }
}
```

Sección C

11. Considere un **Sistema en Línea para Reservación de Vuelos**, en el que los usuarios pueden registrarse para el servicio de ver vuelos disponibles, reservar vuelos, ver y obtener ofertas promocionales, reservar hoteles en el destino del vuelo, pagar en línea por reserva de vuelos/hoteles usando tarjetas de crédito/débito, ver el historial de transacciones, actualizar sus perfiles, recibir notificaciones de reservas por e-mail y celulares; e imprimir los comprobantes para todo tipo de transacción efectuada. El sistema está conectado a un consorcio que representa a las compañías de tarjetas de crédito/débito.
- a. Elabore un **diagrama de caso de uso** para el Sistema en Línea de Reserva de Vuelos. Identifique todos los actores intervinientes y seis casos de uso relacionados al sistema. Haga uso de los estereotipos `<<include>>` y `<<extend>>`. Indique cualquier asunción que realice. **[10%]**
 - b. Elabore la **especificación** de un **caso de uso** (*id, título, autor, actor principal, actor secundario, precondiciones, flujo de eventos, postcondiciones*) que describa las actividades que pueden ser efectuadas por un usuario que reserva vuelos, en donde se podría reservar exitosamente el vuelo o podría no estar en capacidad de hacerse debido a la ausencia de asientos disponibles. Indique cualquier asunción que realice. **[5%]**
 - c. Elabore un **diagrama de clases**. Especifique multiplicidades, relaciones, visibilidad de métodos y atributos. Indique cualquier asunción que realice. **[15%]**
 - d. Elabore un **diagrama de secuencia** para modelar la reserva exitosa de un hotel que incluya la notificación vía email luego del pago. Indique cualquier asunción que realice. **[10%]**