

**Reglas del juego**

- Tal como se le dijo antes, durante este examen usted no puede pedir ABSOLUTAMENTE nada prestado a sus compañeros, ni hablar con ellos.
- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkelas TODAS de igual forma.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.
- Puede consultar sus notas, el libro o apuntes; pero se le recomienda que lo haga cuando esté seguro de qué quiere buscar, en caso contrario estas consultas se convierten en una distracción que le hace perder tiempo.
- No se resolverá ninguna duda durante el examen, así que si algo no le queda claro y usted requiere hacer algún tipo de supuesto, ESCRÍBALO y de una breve explicación de la razón por la cual lo supone. Tenga presente que un supuesto no puede contradecir lo que se le está diciendo.
- Debe escribir con letra clara.

La forma de evaluar este parcial es la siguiente:

| Ítem   | Valor             |
|--|-------------------|
| <b>Primer ejercicio:</b>   | <b>1.5 puntos</b> |
| Llamar de forma correcta a los métodos dentro del main   | 0.5               |
| Modificar de forma correcta el método <code>+venderProducto():void</code> según los requerimientos | 1.0               |
| <b>Segundo ejercicio:</b>  | <b>3.5 puntos</b> |
| Identificación de clases del modelo  | 0.60              |
| Identificación de atributos de las clases del modelo   | 0.30              |
| Identificación de métodos de las clases del modelo   | 0.20              |
| Relaciones de herencia (si las hay)  | 0.30              |
| Relaciones de asociación (si las hay)  | 0.30              |
| Identificación de interfaces   | 0.75              |
| Implementación de interfaces   | 0.75              |
| Identificación de atributos de la clase Ejecutable   | 0.10              |
| Identificación de métodos de la clase Ejecutable   | 0.20              |

**Punto 1**

El siguiente código corresponde a un programa encargado de gestionar los pedidos de los clientes en una empresa. El programa tiene la opción de registrar una venta, de consultar un producto y de agregar un producto. Cuando se va a registrar una venta es necesario pedir la referencia del producto y la cantidad a comprar. En caso de haber inventario, el programa registra la venta y en caso contrario anuncia que no hay inventario suficiente.

Dada esta información, analice el código que se presenta a continuación y realice lo siguiente:

1. Complete el método `+main(String[]):void` de la clase Ejecutable para que, de acuerdo con la opción escogida por el usuario, se llame al método correspondiente para proporcionar la funcionalidad esperada. Utilice el espacio proporcionado en el código para completar el main.
2. Modifique el método `+venderProducto():void` de la clase Ejecutable para que, en el caso en que no se encuentre en inventario la cantidad requerida del producto, se le pregunte al usuario el nombre y la dirección, se adjunte al mensaje de error estos datos indicándole que se le hará llegar la cantidad deseada del producto a su domicilio en cuanto se encuentre en inventario, y se vuelva a lanzar el error con el nuevo mensaje. Utilice el espacio proporcionado en el código para realizar la modificación del método

```

package modelo;

public class Producto {

    private int cantidadDisponible;
    private String referencia;

    public Producto(int cantidad, String referencia){
        this.cantidadDisponible=cantidad;
        this.referencia=referencia;
    }
    public Producto(String referencia){

        this.referencia=referencia;
    }

    public int getCantidad() {
        return cantidadDisponible;
    }

    public void setCantidad(int cantidad) {
        this.cantidadDisponible = cantidad;
    }

    public String getReferencia() {
        return referencia;
    }

    public String toString(){
        return "Hay "+cantidadDisponible+" ítems disponibles del producto con referencia
"+referencia;
    }

}

```

```

package modelo;

import java.util.*;

public class Comercializadora {

    private ArrayList<Producto> productos;

    public Comercializadora(){
        productos= new ArrayList<Producto>();
    }

    public String consultarProducto(String referencia){

        for (int contador = 0; contador < productos.size(); contador++) {
            Producto comparar = (Producto) productos.get(contador);
            if (comparar.getReferencia().equals(referencia)) {
                return comparar.toString();
            }
        }
        return null;
    }

    public int venderProducto(String referencia, int cantidad) throws Exception {
        Producto producto = null;
        for (int contador = 0; contador < productos.size(); contador++) {

```

```

        Producto comparar = (Producto) productos.get(contador);
        if (comparar.getReferencia().equals(referencia)) {
            producto = comparar;
        }
    }
    if (producto.getCantidad() >= cantidad) {
        int cantidadAnterior = producto.getCantidad();
        producto.setCantidad(cantidadAnterior - cantidad);
        return producto.getCantidad();
    }
    else{
        throw new Exception("No existe en el inventario una cantidad de producto "
+
            "suficiente para la referencia indicada");
    }
}
public void agregarProducto(String referencia, int cantidad){
    productos.add(new Producto(cantidad,referencia));
}
}
package control;

import modelo.*;

import java.util.*;

public class Ejecutable {

    private static Comercializadora comercializadora = new Comercializadora();

    private static Scanner lectorCadenas = new Scanner(System.in);

    private static Scanner lectorEnteros = new Scanner(System.in);

    public static void main(String[] args) {

        boolean salir = false;
        while (!salir) {
            System.out
                .println("Digite el número de la opción que desea
realizar:\n[1] Para registrar una venta\n[2] Para consultar un "
                    + "producto\n[3] Para agregar un producto\n[4]
Para salir");

            int opción = lectorEnteros.nextInt();
            switch (opción) {

                case 1:

                    /*
                     * En el caso uno se debe llamar al método que permite vender un
                     * producto. Utilice el espacio a continuación para llamar al método
                     * correspondiente
                     */

```

```

        break;
    case 2:
        /*
         * En el caso dos se debe llamar al método que permite consultar
         * la información de un producto. Utilice el espacio a continuación
         * para llamar al método correspondiente
         */

        break;
    case 3:
        /*
         * En el caso tres se debe llamar al método que permite agregar
         * un producto. Utilice el espacio a continuación para llamar al
         * método correspondiente
         */

        break;
    case 4:
        salir = true;
    default:
        System.out.println("Hasta luego");
        salir = true;
    }
}

}

public static void consultarProducto() {

```

```

        System.out.println("Digite la referencia del producto que desea consultar");
        String referencia = lectorCadenas.nextLine();
        System.out.println(comercializadora.consultarProducto(referencia));
    }

    public static void venderProducto() throws Exception {
        System.out.println("Digite la referencia del producto que desea comprar el
cliente");
        String referencia = lectorCadenas.nextLine();
        System.out.println("Digite la cantidad de ítems que desea llevar el cliente de
esa referencia");
        int cantidad = lectorEnteros.nextInt();
        int nuevaCantidad=comercializadora.venderProducto(referencia, cantidad);
        System.out.println("La venta fue realizada exitosamente. La cantidad "
+ "de producto que queda disponible para esa referencia es: "
+ nuevaCantidad);
    }

    /*
    * Utilice el espacio a continuación para realizar el código del método con las
    * modificaciones que se le piden.
    */

}

    public static void agregarProducto() {
        System.out.println("Digite la referencia del producto que desea agregar");
        String referencia = lectorCadenas.nextLine();
        System.out.println("Digite la cantidad de ítems disponibles del producto");
        int cantidad = lectorEnteros.nextInt();
        comercializadora.agregarProducto(referencia, cantidad);
    }

}

```

## **Punto 2**

Variedades J.J. es una empresa que empezó como una miscelánea de barrio dedicada a la importación y comercialización de productos. Con el tiempo, la empresa creció, compró plantas de fabricación y ahora produce y comercializa ciertas líneas de productos. Esto ha hecho que su portafolio sea muy variado, pues ahora produce cosméticos, bebidas, galletas y lámparas.

Todos los productos de la empresa tienen un código, una descripción, una presentación, un costo y un precio unitario. Los cosméticos pueden clasificarse en líquidos, polvos y cremas. De los cosméticos líquidos es importante saber su capacidad en mililitros, su concentración y si son para hombre o mujer; de los cosméticos en polvo es importante saber el país de origen, pues son los únicos productos de la empresa que todavía se importan. Finalmente, de los cosméticos en crema es importante saber si son para cabello o para el cuerpo y su concentración. De los productos cosméticos, las cremas son medicadas. Los productos medicados sólo se pueden vender con fórmula médica en el mercado y aunque, por ahora, la empresa sólo tiene como producto medicado las cremas, en el futuro se podría tener otro producto que fuera también medicado y sería importante poder identificar rápidamente cuáles son medicados y cuáles no pues las condiciones de venta dificultan su distribución.

En la parte alimenticia, la empresa cuenta con la posibilidad de producir galletas y bebidas. A la empresa le interesa saber, de las primeras, su fecha de vencimiento, si son dulces o saldas y los gramos que pesan, mientras que de las segundas, la empresa ha dispuesto que las gaseosas tienen una cantidad de gas (que las hace más o menos fuertes) y que los jugos tienen o no vitaminas. De ambas clases de bebidas se necesita saber el sabor, la cantidad de mililitros que tiene el envase y el tipo de envase (vidrio o plástico).

Por el lado de las lámparas, sólo es importante saber el tipo de luz que tienen (blanca o amarilla) y el voltaje máximo para conectarlas.

Dado que variedades J.J. tiene una gran diversidad de productos, estos no siempre se pueden vender en los mismos establecimientos. Por esta razón, la empresa quiere tener claramente identificado cada producto de acuerdo a los lugares dónde se puede vender para garantizar que el mismo cumpla con las condiciones necesarias según el lugar de su venta. Así, los productos que se pueden vender por el canal autoservicios deben tener la capacidad de liquidarse en las ofertas; los productos que se pueden vender por el canal tradicional deben tener la capacidad de indicar las condiciones de pago que tiene el cliente; y los productos que se pueden vender por el canal de droguerías deben tener la capacidad de mostrar las indicaciones del producto (ejemplo: champú Caspasol, está indicado para el control de la caspa).

De acuerdo a lo mencionado, es preciso saber que a excepción de los cosméticos en polvo, todos los productos de la empresa se venden en el canal autoservicios. El canal tradicional no se venden ni las cremas, ni las lámparas, y en el canal droguerías sólo se venden cosméticos en cremas y cosméticos líquidos.

Usted debe realizar el diagrama de clases completo implementando el patrón de diseño MVC (paquete control y paquete modelo), de tal forma que el sistema permita:

- Registrar los productos teniendo en cuenta toda la información propia de cada producto.
- Consultar, para un producto específico (buscado por el código) toda la información de éste, incluyendo la oferta que tiene, las condiciones de pago, las indicaciones y si es medicado o no, en caso de tener esta información de acuerdo al tipo de establecimiento donde se comercializa.