CONTEXTO

Sólo unos meses atrás, Don Juan, dueño de la tienda de venta de guitarras denominada "Guitarras Temuco", decidió digitalizar su antiguo sistema de gestión de inventario de papel, y empezar a usar un sistema de inventario computacional. Para ello, contrató una empresa de desarrollo de software, que desarrolló la primera versión de un software de escritorio. Entre los requerimientos que Don Juan plantea a la empresa, está el que sea posible realizar búsquedas que lo ayuden a seleccionar la guitarra ideal para sus clientes.

APLICACIÓN PARA LA GESTIÓN DE GUITARRAS

El siguiente diagrama da cuenta de la primera versión de la primera versión del software. En este sistema, se implementó todo lo necesario para reemplazar por completo el sistema inventario de papel y se incluyó el sistema de búsquedas requerido por el cliente. El siguiente, es parte del diagrama UML y código de la primera versión del sistema (código disponible en archivo ED_Tarea_S3_P2.zip).

Clase Guitarra

```
Guitarra
numeroSerie : String
-fabricante : String
-modelo : String
-tipo : String
-maderaPosterior : String
-maderaFrontal : String
-precio double
+Guitarra()
+getNumeroSerie() : String
+setNumeroSerie(numeroSerie: String): void
+getFabricante() : String
+setFabricante(fabricante : String) : void
+getModelo(): String
+setModelo(modelo: String): void
+getTipo() : String
+setTipo(tipo : String) : void
+getMaderaPosterior(): String
+setMaderaPosterior(maderaPosterior : String) : void
+getMaderaFrontal(): String
+setMaderaFrontal(maderaFrontal : String) : void
+getPrecio double()
 setPrecio double(precio double) : void
```

```
public class Guitarra {
           private String numeroSerie, fabricante, modelo, tipo, maderaPosterior, maderaFrontal
            private double precio:
6
            public Guitarra(String numeroSerie, double precio,
                            String fabricante, String modelo, String tipo,
                            String maderaPosterior, String maderaFrontal){
                this.numeroSerie = numeroSerie;
                this.precio = precio:
                this.fabricante = fabricante;
13
                this.modelo = modelo;
14
                this.tipo = tipo:
15
                this.maderaPosterior = maderaPosterior;
16
                this.maderaFrontal = maderaFrontal:
18
19
           public String getNumeroSerie() {
20
                return numeroSerie;
23
            public void setNumeroSerie(String numeroSerie) {
                this.numeroSerie = numeroSerie;
```

Clase Inventario

Inventario -listaGuitarras : List +Inventario() +agregarGuitarra(numeroSerie) : Guitarra +buscar(guitarraBuscada) : Guitarra

```
public class Inventario {
            public List<Guitarra> listaGuitarras;
9
10
            public Inventario() {
11
                listaGuitarras = new LinkedList();
12
13
14
            public void agregarGuitarra(String numeroSerie, double precio,
15
                                        String fabricante, String modelo, String tipo,
16
                                        String maderaPosterior, String maderaFrontal) {
17
                Guitarra guitarra = new Guitarra(numeroSerie, precio, fabricante,
18
                        modelo, tipo, maderaPosterior, maderaFrontal);
19
                listaGuitarras.add(guitarra);
20
21
22
            public Guitarra obtenerGuitarra(String numeroSerie) {
23
                Iterator<Guitarra> iterador = listaGuitarras.iterator();
                while (iterador.hasNext()) {
25
                    Guitarra guitarra = iterador.next();
                    if (guitarra.getNumeroSerie().equals(numeroSerie)) {
                        return guitarra;
28
29
30
                return null;
31
            public Guitarra buscar(Guitarra guitarraBuscada) {
```

Clase TesterBuscadorDeGuitarras

```
public class TesterBuscadorDeGuitarras {
   public static void main(String□ args) {
       Inventario inventarioGuitarStar = new Inventario();
       inicializarInventarioDeGuitarras(inventarioGuitarStar);
       Guitarra guitarraClaudia = new Guitarra( numeroSerie: "", precio: 0, fabricante: "fender",
                modelo: "stratocastor", tipo: "electric", maderaPosterior: "Alder", maderaFrontal: "Alfer");
       Guitarra auitarra = inventarioGuitarStar.buscar(auitarraClaudia):
       if(guitarra != null){
           System.out.println("Claudia, te podría gustar esta " + guitarra.getFabricante() +
                     " + guitarra.getModelo() + " " + guitarra.getTipo() + "guitarra:\n " +
                   guitarra.getMaderaPosterior() + " madera posterior,\n +" +
                   guitarra.getMaderaFrontal() + " madera frontal.\n Puedes tenerla por solo: $" +
                   guitarra.getPrecio() + "!");
       }else {
           System.out.println("Lo siento Claudia, no tenemos nada para ti.");
   private static void inicializarInventarioDeGuitarras(Inventario inventarioGuitarStar){
       // agregar guitarras al inventario
```

ESCENARIO DE USO

Suponga el escenario de uso del software, donde Claudia, una cliente de la tienda, solicita a Don Juan recomendaciones de guitarras según ciertas preferencias. Entonces, para encontrar la guitarra del gusto de Claudia, se utiliza el software que retorna una respuesta por pantalla.



Sin embargo, Don Juan no está conforme con la respuesta que entrega el sistema de inventario. El conoce su inventario y sabe que la guitarra existe y debería estar disponible en bodega**. El problema es que esta guitarra* (y otras) simplemente no se ha agregado al sistema.

```
Suponga la siguiente guitarra* es a la que refiere Don Juan

("V5693", 1550.9, "Fender", "Stratocastor", "eléctrica", "Alder", "Alder")
```

ACTIVIDADES

En las siguientes actividades asuma la posición de un desarrollador de software, al que le han encargado la tarea de mejorar el software de inventario. Dado que usted no es el desarrollador original, primero, se le hará entrega del código de la aplicación (software legado, o legacy software). Ok, ya está en su poder. Segundo, ahora tiene que empezar a trabajar con el fin de mejorar el software de Don Juan.

1.- Responda las siguientes preguntas:

1	¿Qué debe hacer primero, segundo y tercero?, es decir, ¿Cuál serán las actividades que le permitirán alcanzar el objetivo encargado?
	Lo primero que habría que hacer seria revisar si el software funciona, luego se tendrá que revisar el código apoyándose con lo comentarios y la documentación si es que esta se posea, y luego como tercer paso se debe diseñar y desarrollar el código en conforme a lo acordado con Don Juan.
2	¿Qué problemas ve en la versión actual del software?. Por ejemplo: ¿Existe código duplicado?, ¿la aplicación realizada lo que debería?, ¿el modelo de clases es útil como documento informativo?. Explícitamente, mencione cada uno de los problemas que encuentre, expliqué por qué se producen y proponga una solución para cada uno de ellos. En este último caso, si no puede proponer una solución, exponga su comentario explícitamente.
	Los problemas que se pueden observar con la versión actual del software serian la falta de comentarios en el código lo que ocasiona que la lectura de esta sea confusa ya que no se posee una estimación clara de lo que hace, otro problema es el diagrama de clases donde no se ven las interacciones entre estas lo que ocasiona un enredo de que clase instancia que clase además de no tener en claro el diseño e intención del desarrollador original, otra

	falta que ocurre es la falta de código para inicializar un inventario inicializado, lo que ocasiono el problema en primer lugar.
3	¿Qué cosas/funcionalidades/parámetros modificaría o agregaría? ¿es posible dar solucionar el problema que advierte don Juan**?
	De las cosas que agregaría seria la inicialización del inventario de guitarras ya que ese el problema por el cual ocurrió el escenario de la falta de una guitarra en el sistema, entre las mejoras que se le podrían agregar estas podrían ser un menú donde se pueda controlar el ingreso de nuevas guitarras y el egreso de la guitarras vendidas.
4	¿Es la documentación del software suficiente?. Argumente. ¿Podría mejorar su estado actual?
	La documentación actual según los estándares generales es insuficiente esto debido a que en el código faltan comentarios y el diagrama de clases no muestra las interacciones presentes.

- 2.- Implemente las observaciones de las preguntas 2 y 3 en código. Su implementación debe dar cuenta de una perfecta correlación entre argumento expuesto y código.
- 3.- Desarrolle el diagrama de clases que mejor represente el estado actual del software, y el que incluya las modificaciones que usted sugiere. Agregue ambos diagramas en la siguiente tablas.

	Diagrama de Clases - Estado Actual
Agregar Diagrama de Clases - Estado Actual - Aquí	

Diagrama de Clases - Nueva Versión

Agregar Diagrama de Clases - Nueva Versión - Aquí