**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).

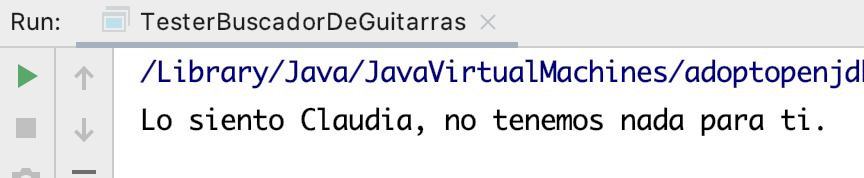
|  |  |
| --- | --- |
| **Clase Guitarra** | |
|  |  |

|  |  |
| --- | --- |
| **Clase Inventario** | |
|  |  |

|  |
| --- |
| **Clase TesterBuscadorDeGuitarras** |
|  |

**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? |
|  | 1. Ver el manual del usuario del programa, en este caso no lo tiene, por ello nos saltamos este paso.  2. Ejecutar el programa y ver las distintas interacciones con el usuario que tiene, en este caso, el programa solo lanza un mensaje sin que el usuario pueda hacer algo.  3. Leer la documentación actual del software (diagramas de clases, diagramas de caso de uso, documentación de clases y métodos, etc…), en este caso constamos con unos diagramas de clases única sin relación aparente, que apenas nos dicen los componentes de las clases y la historia contada mas arriba del documento. (Falta el diagrama de caso de uso, para marcar objetivos, y los comentarios en el código o fuera del él que expliquen el funcionamiento de cada método y clase).  4. Detectar problemas actuales del software (paso realizado en la siguiente pregunta).  5. Crear plan de trabajo en relación a problemas detectados y la documentación modificada y faltante. |
| 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. |
|  | Problemas de diseño.  1. El diseño de sus clases, todas están puestas en el éter, falta de uso de “package” (Tanto en el diagrama de clases como en el código). (S) Tratar de hacer el diagrama según módelo-vista-controlador.  2. Falta de comentarios en código y documento con explicación de los mismo. (S) Crearlos.  3. Clase Inventario no presenta los métodos presentes en el código, los métodos agregarGuitarra y obtenerGuitarra estan mezclado en nombres, parámetros y retornos.  Clase Guitarra:  4. El uso sin distinción de la clase String para la designación de tipo de atributos. (S) Conservando los actuales parámetros: numeroSerie podría ser int; fabricante y modelo dejarlas como String; tipo, maderaPosterior y maderaFrontal como Enum.  5. Falta de parámetros importantes. (S) stock, int que figura la cantidad de guitarras actuales de esa guitarra concreta.  6. Setter no utilizados. (S) Posible eliminación de aquellos constantes (numeroSerie, fabricante, modelo, tipo, maderaPosterior y madreraFrontal), y conservación en los atributos variables (precio y stock).  Clase Inventario:  7. Encapsulamiento de los atributos. (S) Ponerlos en privado.  8. Esta clase debería encargarse de cargar el inventario inicial no el testerBuscadorDeGuitarra.  9. Método buscar() siempre entrega un null, es muy largo, y solo retornaría (si lo hiciera) una, sí y solo sí, se le entrega esa misma guitarra, haciéndolo poco práctico para la búsqueda común de una guitarra por un cliente. (S) Que el método de búsqueda entregue un grupo de guitarras que cumplan con las características dadas según el campo. Crear varios métodos de búsqueda que acorten la lista dada.  Clase TesterBuscadorDeGuitarras:  10. Clase temporal que ni siquiera debería ser considerada como parte del proyecto. (S) Borrar.  Problemas externos:  11. Inexistencia de un base de datos con las guitarras y la posibilidad de actualizarlos desde la misma aplicación. (S) Creación de una clase que se encargue de leer y actualizar la base de datos. (Al no tener idea de guitarras solo se cargará, de momento, una guitarra, la pedida por Claudia) |
| 3 | ¿Qué cosas/funcionalidades/parámetros modificaría o agregaría? ¿es posible dar solucionar el problema que advierte don Juan\*\*? |
|  | Responder esto seria redundar en lo mismo, idem de las soluciones (S) de todos los puntos dados en la pregunta anterior. En cuanto a la posible solución del problema es principalmente el punto 11 de la pregunta anterior, falta de los datos de las guitarras. |
| 4 | ¿Es la documentación del software suficiente?. Argumente. ¿Podría mejorar su estado actual? |
|  | Respondido en el paso 3 de la primera pregunta, siendo más específico, en el segundo paréntesis. Y se podría mejorar modificando los actuales y agregando los nombrados. |

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 |
|  |

|  |
| --- |
| Diagrama de Clases - Nueva Versión |
| PD: Este diagrama puede no ser exactamente lo implementado en código, ya que fue la idea antes de editarlo y durante el proceso de edición se pudieron agregar uno o dos métodos complementario y parámetros no pensados hasta el final, mas, la base es esta. |