Reto 5 Java

Lenguaje

Java

Intentos

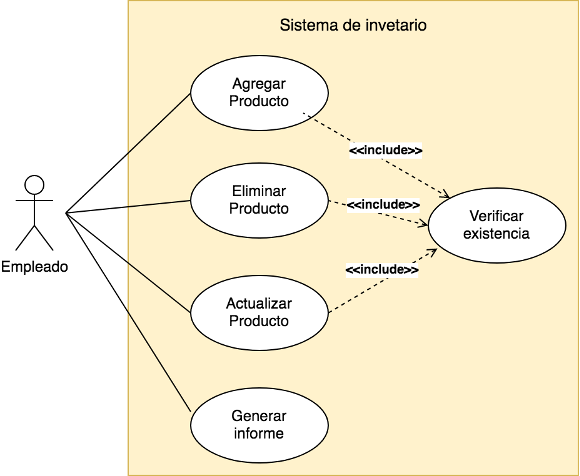
0 de 20

Puntaje requerido

90%

Enunciado

**Reto 5: Sistema de inventario usando el patrón MVC y Spring Data JDBC**  
  
En este reto continuaremos trabajando con el sistema de inventario para la cafetería de la Sergio Arboleda desarrollado en los Retos 2, 3 y 4. Nuestro objetivo es lograr implementar de forma completa la aplicación utilizando el patrón MVC y el Framework Spring Data JDBC. A continuación, se presenta nuevamente el contexto del problema.  
  
Una de las cafeterías de la Sergio Arboleda vende frutas, dulces y algunas bebidas. Con el propósito de mejorar el control sobre las ventas y el inventario de la tienda, la universidad decide construir una aplicación en Java que le permita almacenar la información de los productos y calcular algunas estadísticas sobre el inventario. Con el propósito detener una especificación más clara sobre lo que debería hacer la aplicación, el administrador de la tienda construyen el siguiente diagrama de casos de uso:



Parte 1: Creando una base de datos en MySQL  
  
En la primera parte del reto se debe utilizar MySQL para construir una base de datos que almacene la información de los productos disponibles en la tienda. La base de datos estará formada por una única tabla llamada Productos que almacenará: el código, el nombre, el precio y el inventario disponible. La Tabla 1 presenta los productos disponibles a la fecha.

|  |  |  |  |
| --- | --- | --- | --- |
| **código** | **nombre** | **precio** | **inventario** |
| 1 | Manzanas | 5000.0 | 25 |
| 2 | Limones | 2300.0 | 15 |
| 3 | Peras | 2700.0 | 33 |
| 4 | Arandanos | 9300.0 | 5 |
| 5 | Tomates | 2100.0 | 42 |
| 6 | Fresas | 4100.0 | 3 |
| 7 | Helado | 4500.0 | 41 |
| 8 | Galletas | 500.0 | 8 |
| 9 | Chocolates | 3500.0 | 80 |
| 10 | Jamon | 15000.0 | 10 |

**Tabla 1: Productos disponibles en la tienda**

Estos registros deben ingresarse a la base de datos utilizando MySQL Workbech. Serán utilizados más adelante para realizar pruebas.  
  
Parte 2: Implementando el controlador con Spring Data JDBC  
  
Para esta parte debemos seguir los siguientes pasos:

* Crear un proyecto en NetBeans llamado inventarioAPP.
* Incluir Spring Boot y agregar 'Spring Data JDBC' y 'MySQL Driver' como parte de las dependencias del proyecto
* Configurar el archivo 'application.properties' con las credenciales a la base de datos.
* Crear 3 paquetes: modelo, vista ycontrolador
* En el paquete modelo, declare una clase llamada Producto que representará la tabla en la base de datos. No olvide agregar las anotaciones a la clase para definir cuál atributo será la llave primaria.
* En el paquete modelo declare una interfaz llamada RepositorioProducto que herede de CrudRepository. Este repositorio nos permitirá utilizar las funciones CRUD sin ninguna implementación adicional.
* En el Paquete controlador, declare una clase llamada ControladorProducto. Esta clase tendrá como atributo una instancia de RepositorioProducto.
* En la clase ControladorProducto implemente los métodos CRUD para la tabla Productos y un método informe que obtenga el nombre del producto con el precio mayor; el nombre del producto con el precio menor; el promedio de precios de todos los productos y el valor total del inventario a la fecha. Este último se obtiene multiplicando el precio de cada producto por el inventario disponible y luego sumando todos los resultados.
* Verifique el correcto funcionamiento de los métodos implementados.

Parte 3:Implementando la Vista con Swing  
  
Luego de implementar el modelo y el controlador con la ayuda de Spring Data JDBC, el dueño del almacén le entrega el siguiente diseño de la interfaz gráfica (ver figura 2). En el diseño se incluyen las ventanas emergentes que surgen como resultado de realizar algunas validaciones.Esta interfaz el la misma construida en el reto 3. El código debe alojarse en el paquete vista declarado anteriormente (Esta interfaz es la misma desarrollada en el reto 3).

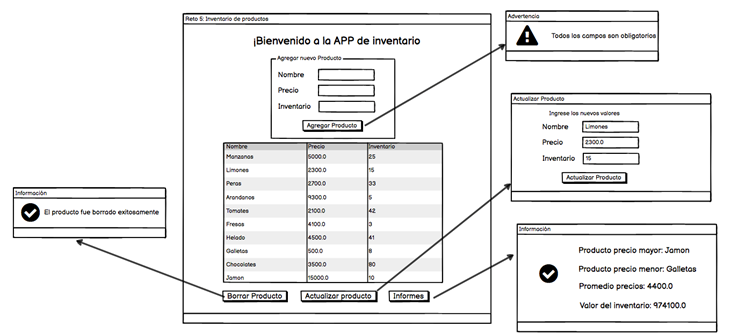


Figura 1: Diseño inicial interfaz gráfica  
  
Por ejemplo, si se trata de agregar un producto y existe algún campo vacío, se deberá generar una ventana emergente que informe que todos los campos son obligatorios. Cundo se agrega, se borra o se actualiza un producto se debe generar una ventana informando que el proceso fue exitoso. Al presionar el botón 'Actualizar Producto' se generará una ventana emergente con los campos a actualizar y presentando la información del producto que se encuentra seleccionado en la tabla. Cada que se realice una operación sobre los datos de la tabla, esta debe actualizarse nuevamente de manera automática.  
  
Por otro lado, el botón llamado 'Informes' genera una ventana emergente con los datos obtenidos al evaluar el método generar informe. Para cumplir con el patrón MVC debemos garantizar que los eventos sean capturados por el controlador y no directamente por la vista.  
  
Finalmente, para facilitar el desarrollo de la aplicación, el dueño del almacén le entrega una lista de verificación con todas las funcionalidades que se espera desarrollar.  
  
Comprobando las funcionalidades  
  
Descripción de la funcionalidad Completado  
  
1 Crear Base de datos en MySQL  
  
2 Insertar los registros Tabla 1 en la Base de datos  
  
3 Implementar los métodos acceso a datos utilizando Spring Data JDBC  
  
4 Implementar función para borrarProducto en el controlador  
  
5 Implementar función para AgregarProducto en el controlador  
  
6 Implementar función para ActualizarProducto en el controlador  
  
7 Implementar la interfaz gráfica con Swing  
  
8 Capturar los eventos generados por los botones 'Agregar Producto', 'Borrar producto' y 'actualizar producto' en el controlador  
  
9 Implementar validación sobre campos vacios para agregar producto  
  
10 Implementar validación sobre campos vacios para actualizar producto

Dificultad

Instrucciones

**Instrucciones para la calificación automática**  
  
Instrucciones para la calificación  
\* Este reto será calificado de forma manual por el formador  
\* En la plataforma MasterTech, dónde se agregan los archivos de soporte, se debe adjuntar un archivo .zip que contenga todo el proyecto.  
\* Antes de presentar la solución del reto a su formador, verifique que la aplicación cumple con todos los elementos requeridos.