DOCUMENTACION

PRUEBA 2: Creación de “Stock Management”

**OBJETIVO**

Creación de un módulo de gestión de inventarios para un supermercado, que permita listar el inventario de una tienda, así como agregar, modificar y eliminar artículos.

**DOCUMENTACION EXTERNA**

1. Primero se debe realizar la creación de la base de datos e ingresar los datos esto se realiza mediante la ejecución del script mySQLScrip que lo crea en Mysql.
2. Luego se debe configurar el servidor Tomcat en localhost en el puerto 8081
3. Posteriormente se importa el proyecto en el IDE Eclipse, se debe ejecutar maven clean, maven install y posteriormente run on the server.
4. Se mostraran los artículos que se encuentran en la base de datos, en él se muestran 3 opciones. Insertar Artículo, Modificar Artículo y eliminar Artículo que ingresaran al servicio que se encuentra en http://localhost:8081/stockManagement/services/stockManagement?wsdl.
5. En la opción Insertar Articulo, se llama al web service a la función insertItem se muestra el mensaje de confirmación y se regresa a la página inicial.
6. En la opción Modificar Articulo, se llama al web service a la función updateItem se muestra el mensaje de confirmación y se regresa a la página inicial.
7. En la opción Eliminar Articulo, se llama al web service a la función deleteItem se muestra el mensaje de confirmación y se regresa a la página inicial.

**DOCUMENTACION INTERNA**

**FRONT-END**

1. Cuando se realiza la ejecución desde el servidor, se llama a la página index.jsp donde se muestra una tabla dinámica que cambia según los datos que se encuentren en la base de datos en el momento que se carga la página, la parte lógica se encuentra en el archivo getAll.js.
2. En ella se encuentran 3 botones para insertar, modificar y eliminar un artículo cada uno de ellos envía a un nuevo formulario.
3. En la opción insertar se encuentra en el archivo llamado insertItem.jsp en el cual se encuentra una tabla que pide que se ingrese los datos correspondientes para insertar a la base de datos a través de un mensaje SOAP al webservice, la parte lógica se encuentra en el archivo insertjs y se regresa a la página inicial.
4. En la opción modificar se encuentra en el archivo llamado updateItem.jsp en el cual se encuentra una tabla que pide que se ingrese los datos correspondientes para modificar la base de datos a través de un mensaje SOAP al webservice, la parte lógica se encuentra en el archivo updatetjs y se regresa a la página inicial.
5. En la opción eliminar se encuentra en el archivo llamado deleteItem.jsp en el cual se encuentra una tabla que pide que se ingrese el código que se desea eliminar de la base de datos a través de un mensaje SOAP al webservice, la parte lógica se encuentra en el archivo deletejs y se regresa a la página inicial.

**BACK-END**

1. El WSDL llamada StockManagement contiene la información de todos los modelos y mensajes que se pueden realizar en el webservice , a partir de él se generan los archivos para poder manejar los mensajes con los parámetros correspondientes y las respuestas adecuadas.
2. Cada vez que se realiza una llamada o mensaje se ingresa a la clase stockManagement que es la interfaz de la clase stockManagementImpl donde se encuentran todas las funciones del webservice en este caso se llaman updateItem, insertItem, getAllItems y deleteItem. En esta clase se obtienen los parámetros enviados en el mensaje y se envían a la clase servicio para ser procesadas y finalmente se envía la repuesta.
3. La clase Servicio se llama ItemService que es la interfaz de ItemServiceImpl tiene las funciones getAll, update, add, delete relacionadas con las funciones que se encuentran en stockManagement. Esta clase se encarga de crear un objeto de la clase ItemRepository y enviar los parámetros que recibieron de stockManagementImpl y enviarlos a las funciones respectivas de ItemRepository. Finalmente obtiene la respuesta de la clase repository y la devuelve a la clase stockManagementImpl.
4. La clase ItemRepository está encargada de acceder a la base de datos y obtener, modificar, ingresar o eliminar los datos dependiendo de las funciones getAll, update, add y delete respectivamente. En cada una de las funciones obtiene una sesión de conexión a la base de datos que está configurada en la clase hibernateUtil y posteriormente realizar las operaciones respectivas en la sesión obtenida. Finalmente obtiene la respuesta de la base de datos y la devuelve a la clase ItemServiceImpl.
5. La Clase Hibernate útil está encargada de crear una única sesión para realizar la conexión con la base de datos según el patrón singleton. Y de enviar esta conexión a la clase ItemRepositoryImpl.

**LIBRERIAS**

Apache-CXF 3.0.2: Es Genera automáticamente los archivos necesarios para el soap webservice a partir del archivo wsdl que se indica en el archivo pom.

Maven 3.2: es un administrador de proyectos que ayuda a modelarlos, a incluir de pendencias de manera sencilla y rápida. Además permite cambiar aspectos principales del proyecto sin errores en la ejecución.

Spring 4.2.0 : Es un framework que permite realizar el desarrollo más rápidamente y fácilmente gracias a sus multiples funcionalidades.

Hibernate 4.2.15: ES un ORM que permite realizar la conexión con Base de Datos limpiamente, es decir que se puede cambiar el gestor de base de datos en el archivo de configuración llamado hibernate.cfg.xml y este seguirá funcionando. Se necesitan también archivos que realicen el mapeo objeto-relacional, para este proyecto se creo únicamente el archivo ítem.hmb.xml que mapea con la tabla ítem en la base de datos Mysql

MySQL 5.1.22: Es un gestor de base de datos relacionales desarrollado por Microsoft que se utilizo para crear y llenar la base de datos.