PROYECTO: Importación de Información Excel (xls., xlsx.) a BD en SSMS

Para la realización de este proyecto se usó Windows Forms (.NET Framework) versión 4.7.2.

Primeramente, creo la conexión a la BD, al establecer un form “VentanaPrincipal” (Inicio) como punto de partida, a su vez, agrego un botón “buttonPreparacion” (Prueba) que ejecutará las pruebas de conexión, modificación y si es posible, vaciado de la base de datos.

Luego establezco la conexión con la BD al hacer clic derecho a “Prueba\_03”, que es el nombre del proyecto, le doy a la opción de “Agregar” 🡪 “Nuevo Elemento” 🡪 y escojo en “Elementos de C#” la categoría de “Datos” 🡪 Haciendo clic en “ADO.NET Entity Data Model” y nombrándolo a mi conveniencia.

Una vez hecho esto aparece una ventana titulada “Asistente para Entity Data Model” a lo cual escojo la primera opción de contenedor de modelo. Después de ello, tengo que elegir la conexión de datos, haciendo clic en “Nueva conexión” y usando el nombre del servidor obtenido al iniciar SSMS (en mi caso LAPTOP-4LK7F4RG) sumado a que uso la autenticación de Windows y le doy check a la casilla de “Confiar en el certificado de servidor” me da la opción de escoger la Base de Datos que cree (BD\_Importador).

Luego de hacer clic en “Probar conexión” y ver que todo esta conforme acepto la conexión y dejo que muestren las tablas de la Base de Datos mostrada. Una vez terminado el proceso, en el Explorador de Soluciones deberá aparecer un archivo .edmx seguido del nombre del elemento. Finalizando en la aparición de su diagrama el cual resulta ser el diagrama de la Base de Datos.

Para establecer la conexión con la base de datos creo un método experimental llamado PruebaDatabase() donde antes tengo que especificar la cadena de conexión, que es lo que permitirá hacer puente entre el código de C# y la BD. Resulta que para crear la cadena de conexión se necesitan de 3 conceptos importantes, el servidor, la base de datos específica y la seguridad integrada; estos 3 conceptos se deben especificar en una cadena string como si codificaras múltiples variables, con ; al final de cada concepto.

En el propio método establezco dos variables, la fecha y el query. La fecha lo hago debido a que tengo un método desarrollado para obtener la fecha, así como los otros campos de la BD, pero todavía no desarrollo ni uno ni otro. Mientras que el query es una cadena string con comandos SQL que llenan con registros la tabla a especificar. Una vez hecho esto, uso “SqlConnection” para conectarme con la cadena de conexión y SqlCommand para ejecutar el query en la BD, habiendo rellenado los parámetros del query previamente. Cabe recalcar que antes de ejecutar SqlCommand y el query, tengo que “abrir” la conexión con la BD, una vez que ejecuto el query, “cierro” la conexión.

La forma de hacer la primera función resulto ser con funciones básicas y una cadena de conexión especificada, ahora usando Entity Framework es donde cambia la cosa, al establecer una conexión con un objeto de base de datos (como se hizo antes), en el apartado de “App.config” se crea una cadena de conexión con el adicional de “metadata”