**Herramienta que exporta en formatos abiertos de excel: CSV y XLSX, como lo pide la Nortic A3: 2014, Norma sobre publicación de datos abiertos del Gobierno Dominicano, desarrollada en asp.net MVC.**

**Metodo para exportar en formato abierto CSV desde c#, asp.net MVC**

Los archivos CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas o por punto y coma.

Este método lo que hace es exportar los datos de los conjuntos de datos y exportarlos en formato .ods

* **Paso 1:** En el controlador se debe agregar la siguiente librería

using System.IO;

* **Paso 2:** crear el método que permite exportar en formato csv

public void ExportarCSV()

{

* **Paso 3:** a continuación, utilizamos StringWriter, del espacio de nombres System.IO, esto permitirá más adelante que se pueda enviar la cadena de texto con la información que queremos que se visualice en Excel.

StringWriter strw = new StringWriter();

* **Paso 4:** se deben agregar los nombres de las variables o como queramos que se visualice en el encabezado de excel usando la sintaxis definida dentro de comillas dobles y utilizando (\).

strw.WriteLine("\"AÑO\";\"MES\";\"GALONES POR MES\";\"METRO CUBICO POR MES\"");

Response.ClearContent();

* **Paso 5:** en esta línea de Código se especifica el nombre y formato del archivo

Response.AddHeader("content-disposition",

string.Format("attachment;filename=Produccion\_de\_agua.csv"));

* **Paso 6:** luego con linq(**Language Integrated Query)** se hace la consulta a la tabla que queremos visualizar en Excel ordenada por su id

var listaProdAgua = db.RecursoProdAgua.OrderBy(x => x.IdRecursoPAgua).ToList();

* **Paso 7:** luego con el bucle foreach se obtienen los valores de los datos de la tabla desde la base de datos para mostrarlos en excel

foreach (var recursoProdAgua in listaProdAgua)

{

strw.WriteLine(string.Format("\"{0}\";\"{1}\";\"{2}\";\"{3}\"",

recursoProdAgua.Anio, recursoProdAgua.Mes.NombreMes, recursoProdAgua.GalonesxMes, recursoProdAgua.Mts3Mes));

}

* **Paso 8:** se debe agregar “text/csv” como tipo de dato y las demás líneas de Código.

Response.ContentType = "text/csv";

Response.ContentEncoding = System.Text.Encoding.GetEncoding("ISO-8859-1");

Response.Write(strw.ToString());

* **Paso 9:** al final para cerrar el metodo usar “Responde.End”

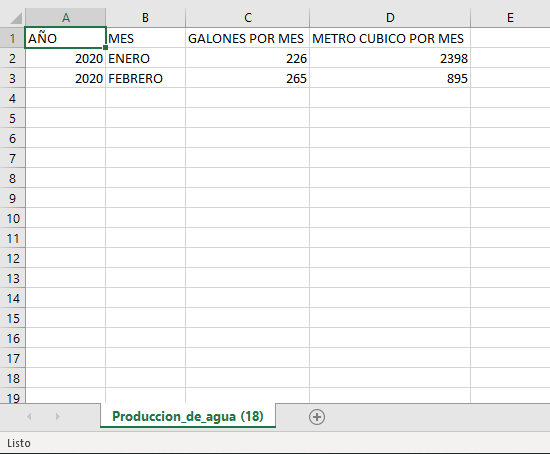
Response.End();

}

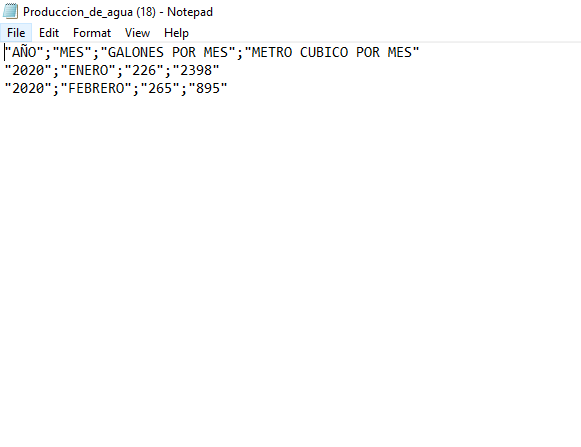
* **Paso 10:** en la vista, agregar un action link para llamar el método “ExportarCSV” creado anteriormente, utilizando el siguiente Código

@Html.ActionLink("Exportar a CSV", "ExportarCSV");

**Visualización del documento csv en excel**



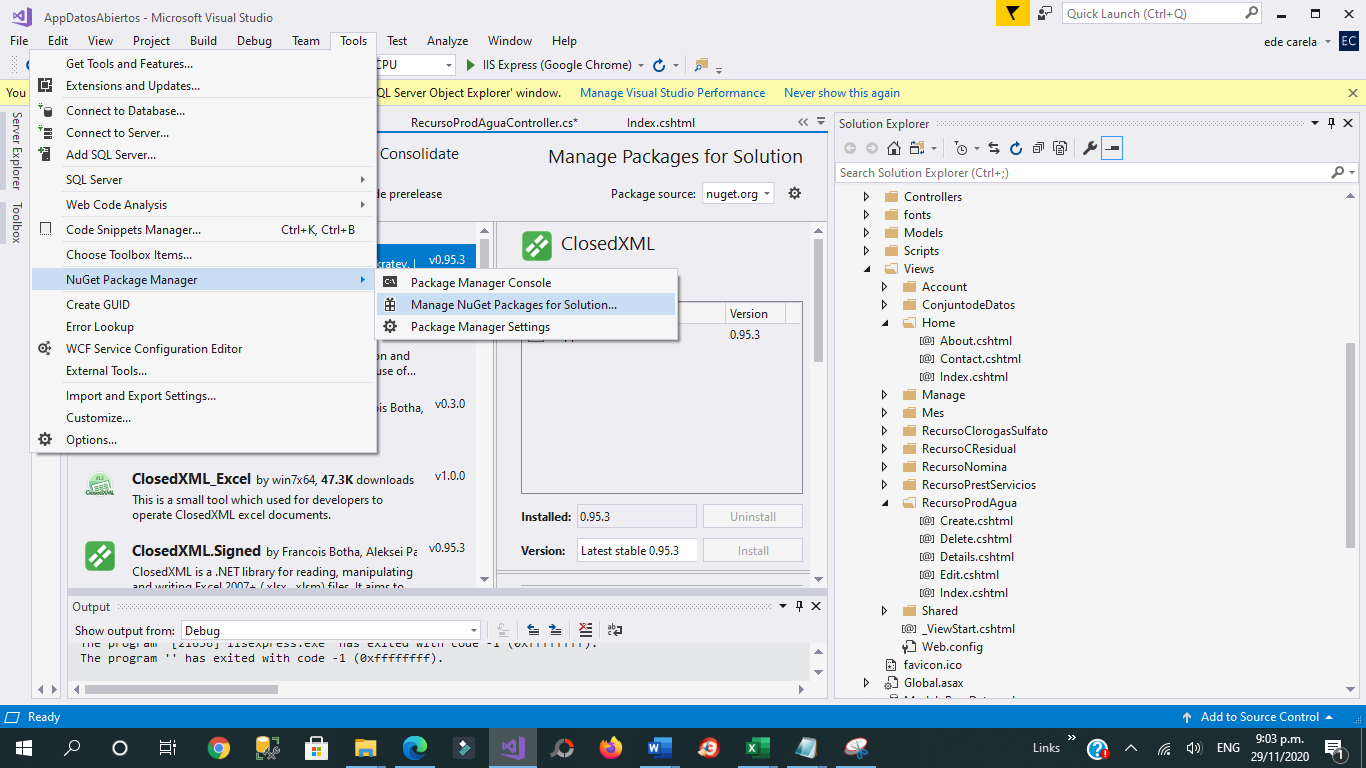
**Como se visualiza en el block de notas:**

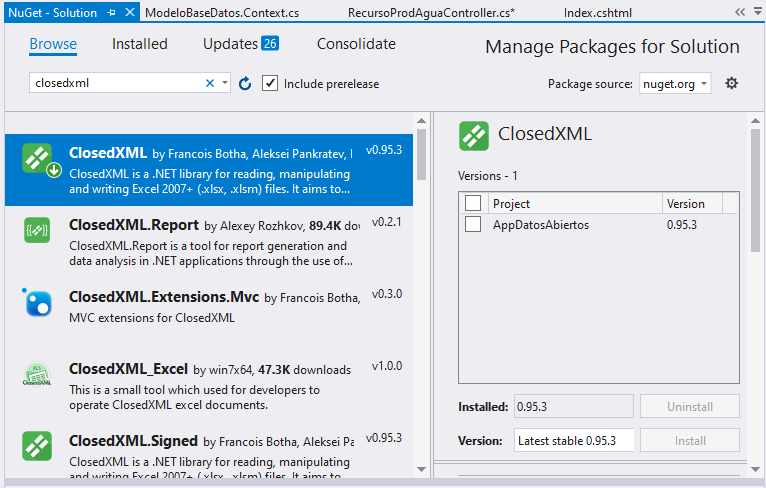


**Método para exportar a formato XLSX de Excel:**

Para exportar al formato xlsx Excel se utilizó ClosedXML, una librería utilizada para exportar en formatos de excel 2007 o posterior.

* 1-El primer paso será instalar el paquete nuget para poder utilizar el Código de esta Librería





* 2-Luego agregar las siguientes librerías en el espacio de nombres:

using ClosedXML.Excel;

* 3-Luego crear el método que nos va a permitir poder exportar los datos desde el sistema en la base de datos a excel

public ActionResult ExportarAxlsx()

{

* 4-Aquí se debe crear un nuevo Datatable

DataTable dt = new DataTable("RecursoProdAgua");

* 5-Esta línea de Código lo que hace es agregar la cantidad de columnas que va a tener nuestro archivo de excel, debe ser la cantidad de variables que se desee mostrar en dicho documento, luego, utilizando new DataColumn se deben agregar cada uno de los encabezados con los nombres que se van a mostrar en la primera columna

dt.Columns.AddRange(new DataColumn[4] { new DataColumn("AÑO", typeof(Int32)),

new DataColumn("MES", typeof(string)),

new DataColumn("GALONES POR MES", typeof(decimal)),

new DataColumn("METRO CUBICO POR MES", typeof(decimal))});

* Paso 6: luego con linq(Language Integrated Query) se hace la consulta a la tabla que queremos visualizar en Excel ordenada por su id

var listaProdAgua = db.RecursoProdAgua.OrderBy(x => x.IdRecursoPAgua).ToList();

* Paso 7: luego con el bucle foreach se obtienen los valores de los datos de la tabla desde la base de datos columna por columna

foreach (var Lista in listaProdAgua)

{

dt.Rows.Add(Lista.Anio, Lista.Mes.NombreMes, Lista.GalonesxMes, Lista.Mts3Mes);

}

* 7: ahora utilizaremos la librería closeXML crearemos una variable “wb” de la clase XLWorbook, y en las siguientes líneas de codigo agregaremos el nombre de la pagina de excel que tendrá nuestro archivo cuando sea exportado, además en este caso con la propiedad XLTableTheme.None, permite que el documento al descargarse no tenga un color de tabla predeterminado, sino que se exporte sin ningún tipo de color en las columnas.
* La propiedad ShowAutoFilter = false; va a quitar el autofiltro de la tabla que se crea automático al exportar.

using (XLWorkbook wb = new XLWorkbook())

{

var ws = wb.Worksheets.Add("Produccion\_de\_Agua");

ws.FirstCell().InsertTable(dt).Theme = XLTableTheme.None;

ws.Tables.FirstOrDefault().ShowAutoFilter = false;

* 8: Para finalizar el método en las siguientes líneas de Código agregar la clasificación con el formato de excel y también wb.SaveAs para permitir que el archivo se guarde.

using (MemoryStream stream = new MemoryStream())

{

wb.SaveAs(stream);

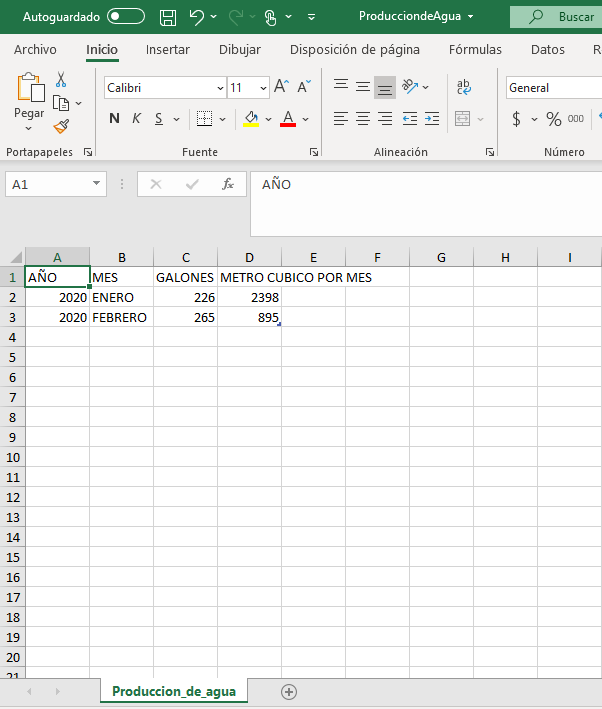
return File(stream.ToArray(), "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet", "ProducciondeAgua.xlsx");

}

}

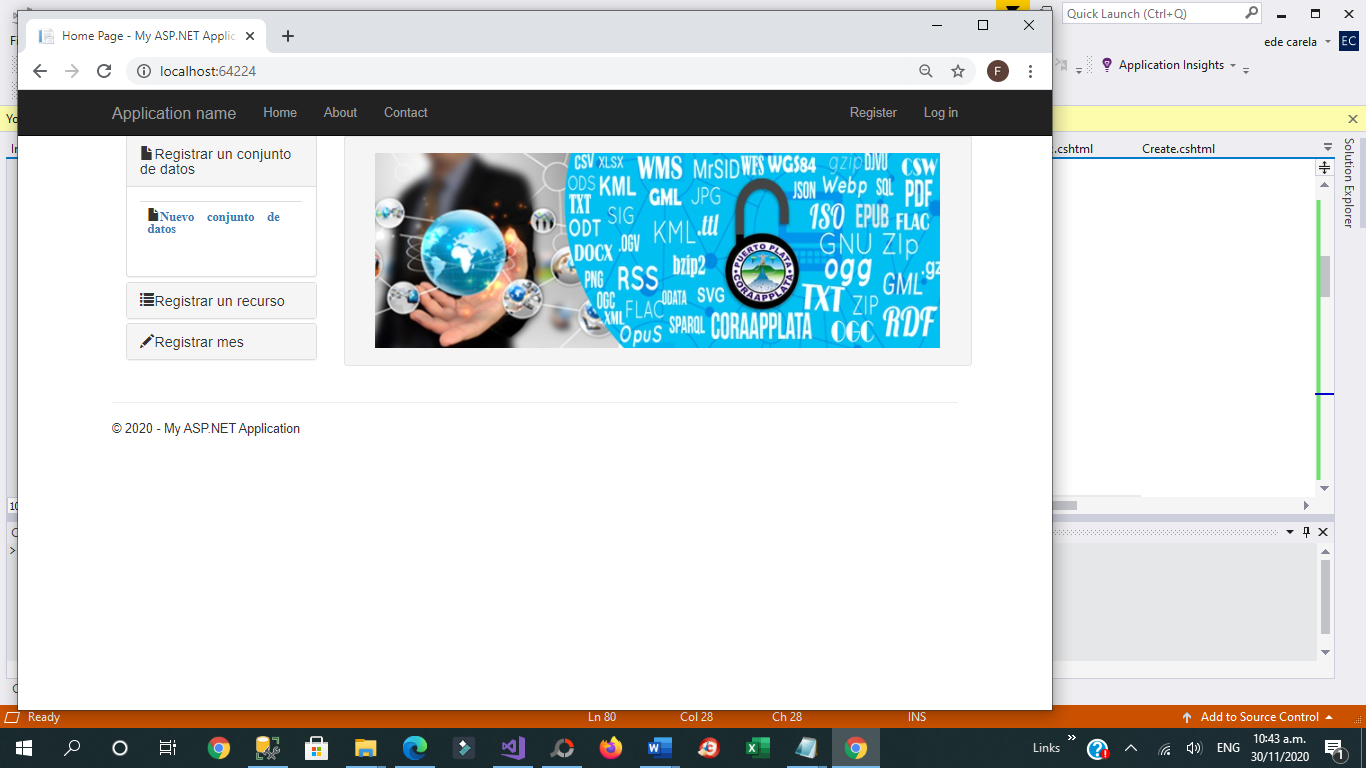
}

**Así se visualiza el archivo en excel**

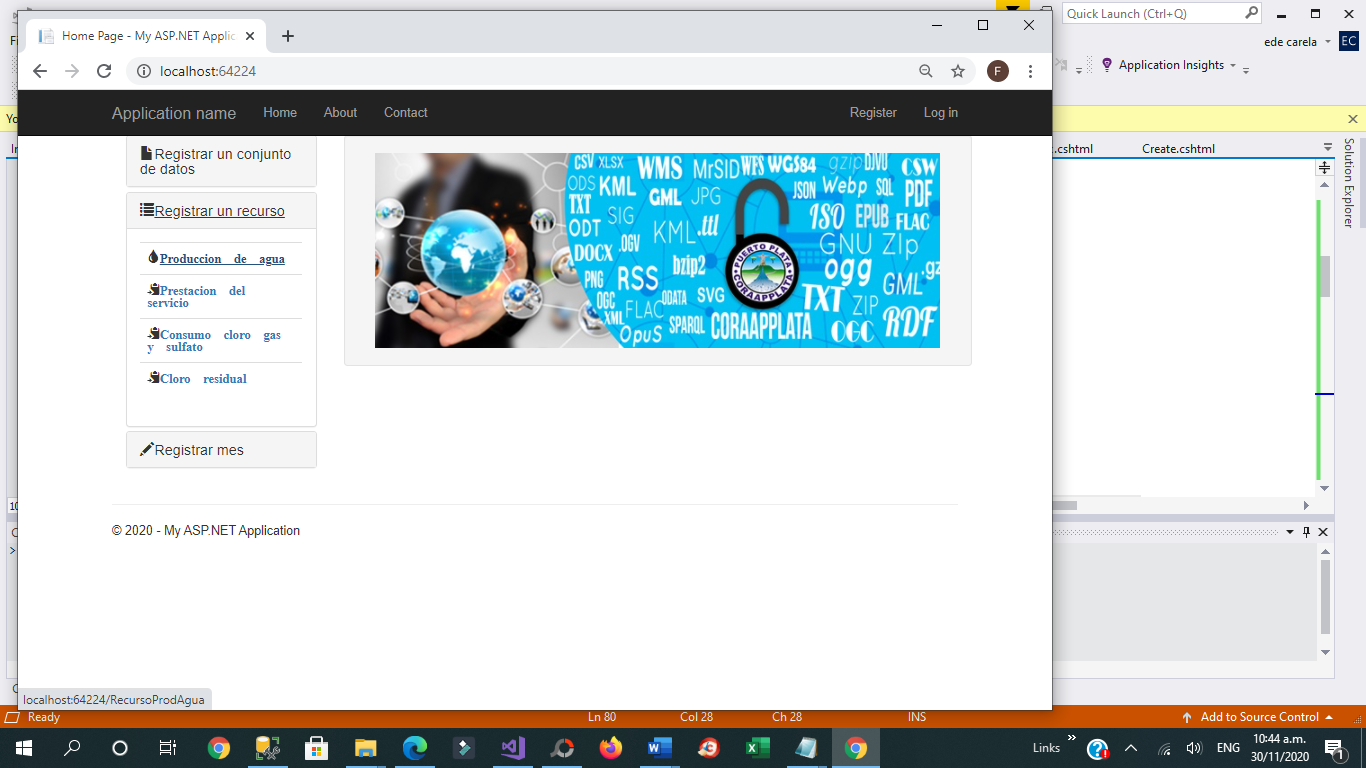


**Interfaz grafica para visualizar como se exportan los archivos a xlsx y csv**

En el menú, primero se debe agregar un nuevo conjunto de datos y el mes, para poder llenar cada uno de los recursos ya que dichas tablas están relacionadas



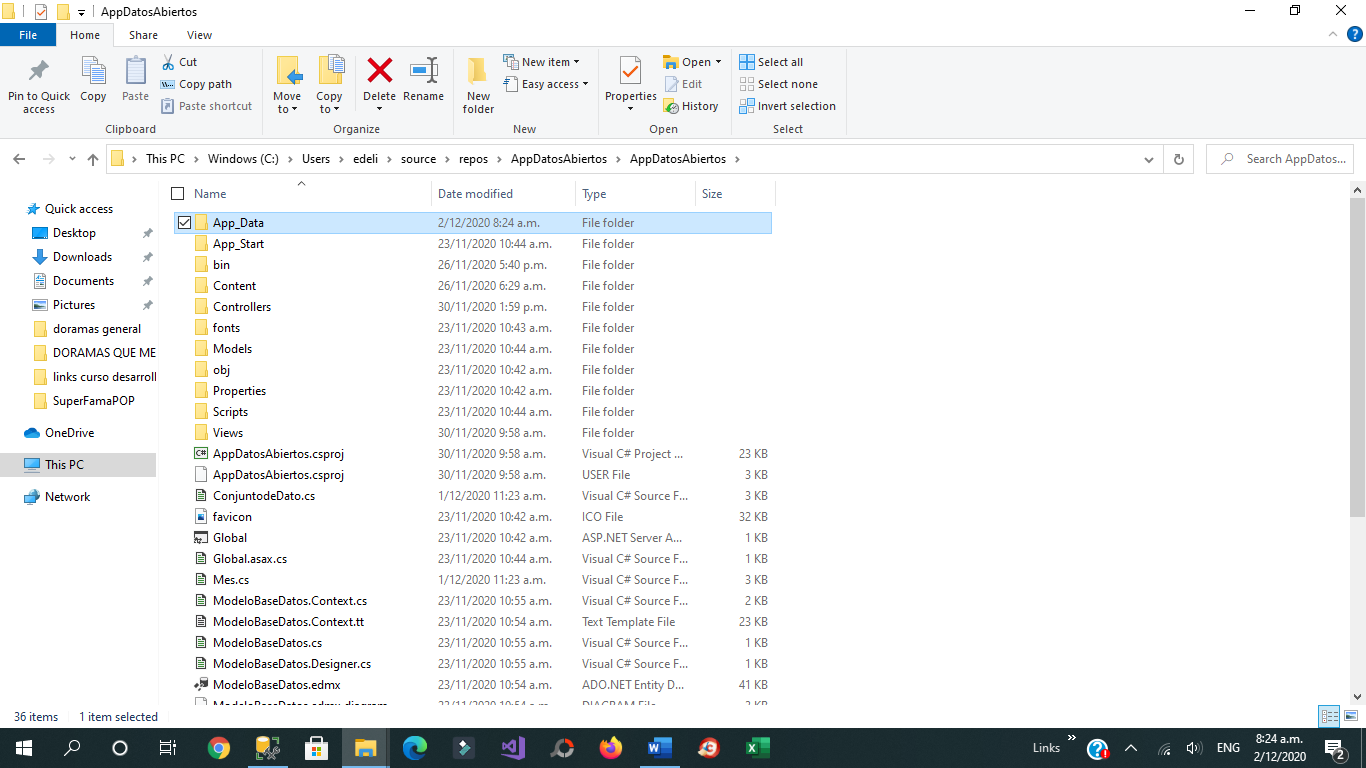
Luego seleccionar registrar un recurso para registrar y ver la información de cada uno de los recursos



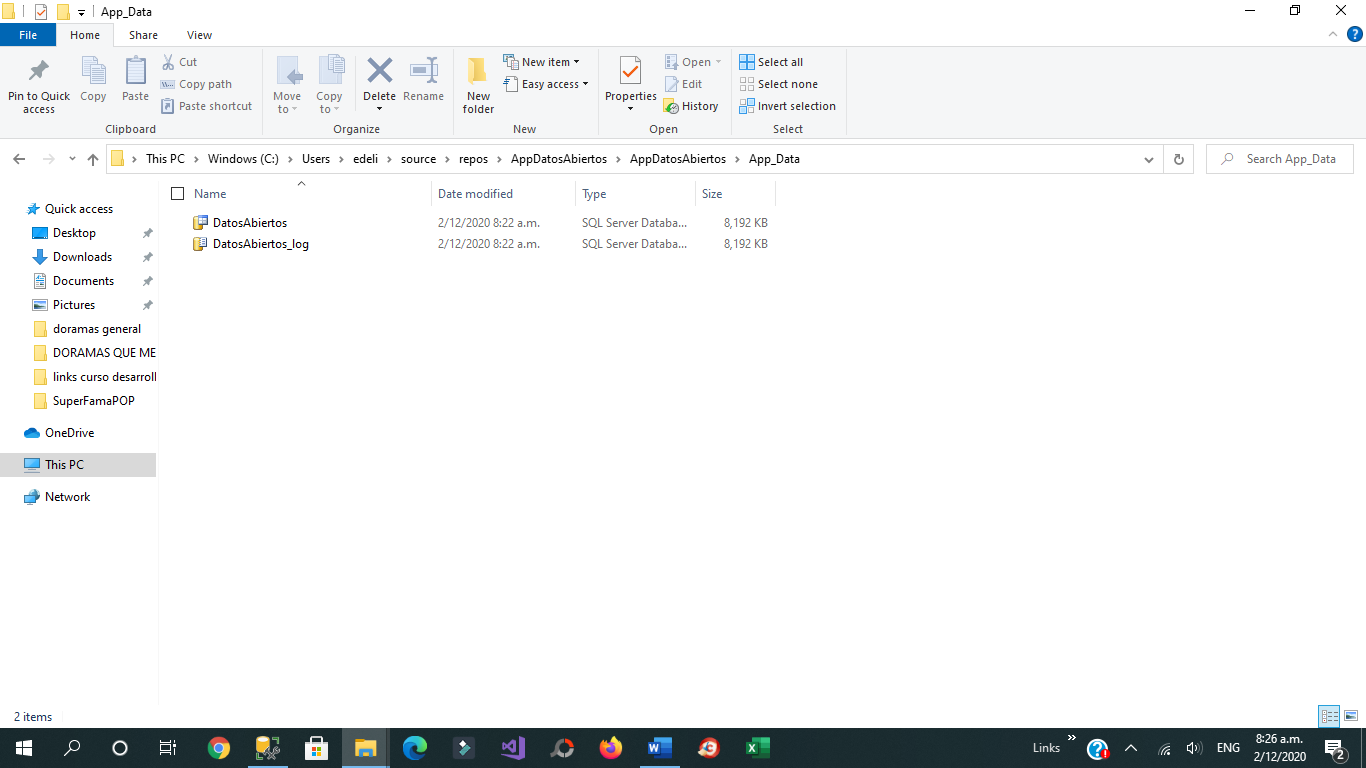
Para exportar a xlsx y a csv están los botones de cada acción:



**Para encontrar la base de datos buscar la carpeta App-Data**



Luego debe atachar los dos archivos que están dentro de la carpeta al sql server



seleccionando attach en las opciones de Databases

