

Instituto Nacional Cantón Lourdes

**Alumno:**

Fernando José Majano Aguirre

**Nombre de la Actividad:**

Documentación Todo\_2890433

**Docente:**

Francisco Arcides Canales Sorto

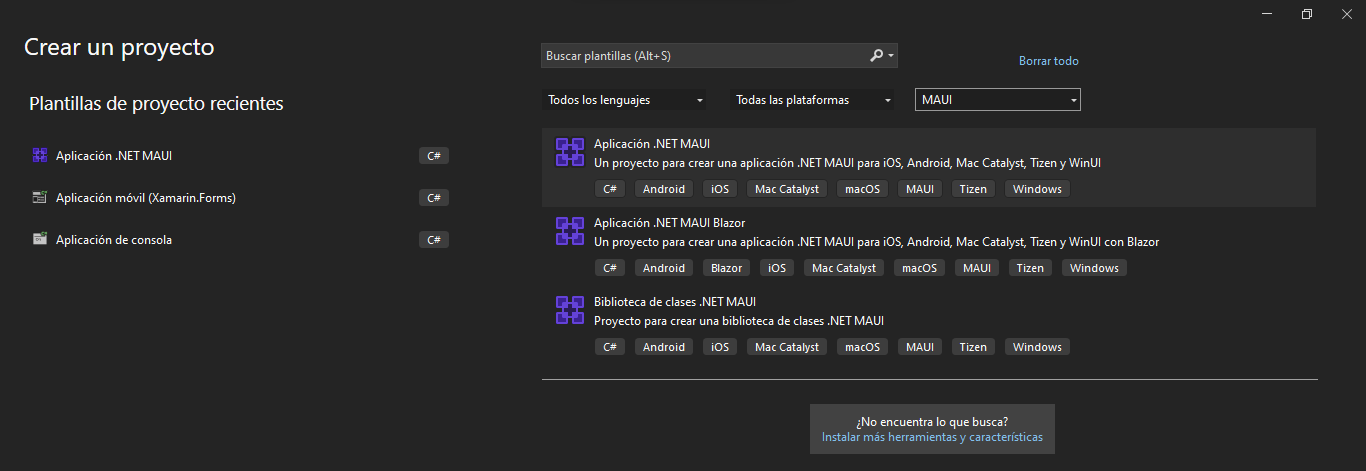
**Lourdes, 19 de agosto de 2023**

**Introducción:**

En este documento se vera paso a paso la creación de una aplicación en .Net MAUI y C# con la funcionalidad de agregar, eliminar y guardar cambios en ítems creados dentro de la misma.

También se puede observar que en la creación se utiliza ayuda externa que es por medio de paquetes NuGet para agregar referencias en las plantillas de C#.

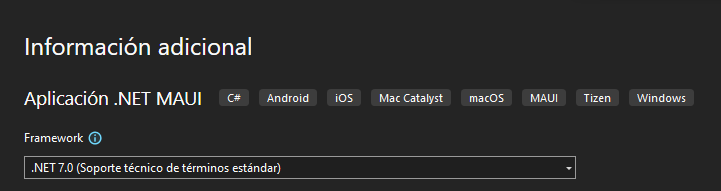
Sin más dilación, empecemos.

**Paso 1:** Entramos a Visual Studio y seleccionamos crear proyecto, luego seleccionamos Aplicación .Net MAUI y damos click en siguiente.

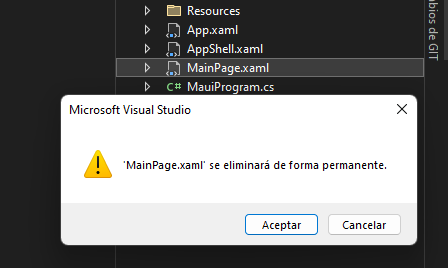
**Paso 2:** Le asignamos un nombre al proyecto, en mi caso es Todo\_2890433 porque ese es mi NIE.



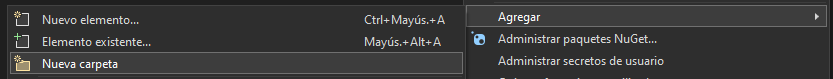
**Paso 3:** Es importante que seleccionemos un Framework más actual, en este caso el 7.0.



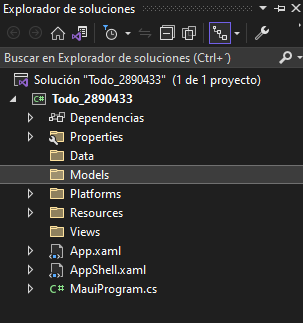
**Paso 4:** Después de crear el proyecto, eliminamos la página principal y confirmo la acción.

****

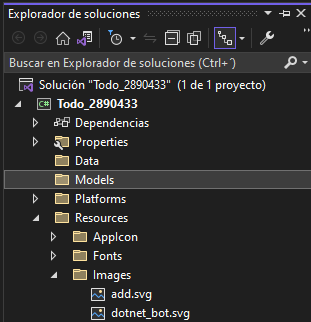
**Paso 5:** Agregamos tres carpetas en el proyecto (Data, Models, Views).

****

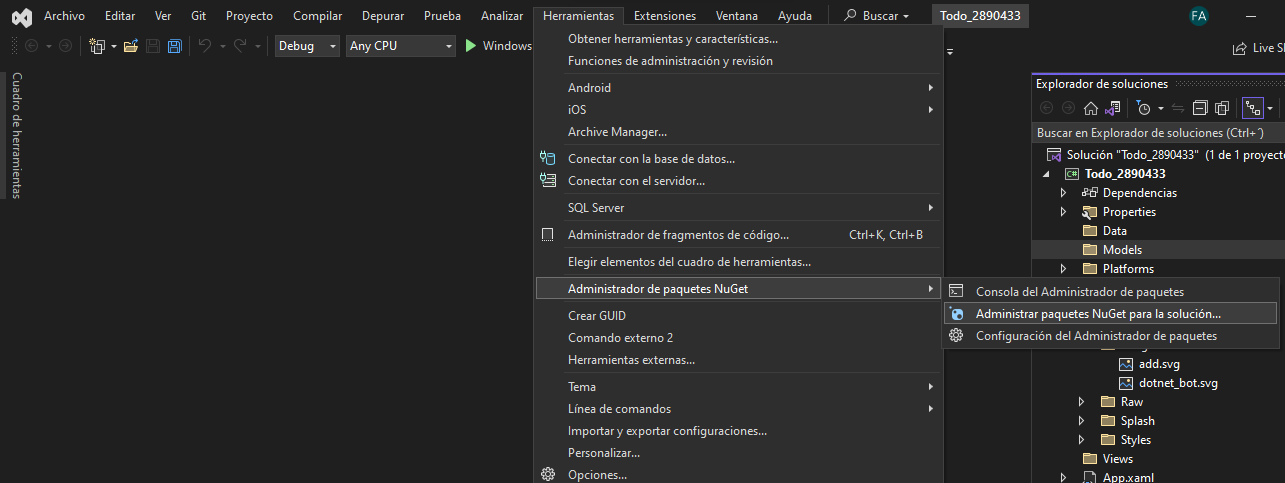
Debe quedar así:

****

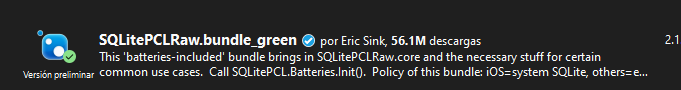
**Paso 6:** Agregamos la imagen add.svg en la carpeta Resources/Images.

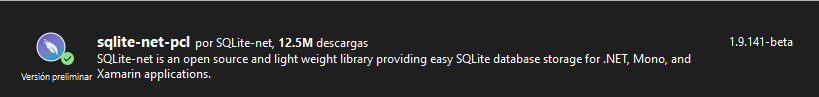
****

**Paso 7:** Nos vamos a Herramientas/Administrador de paquetes NuGet/Administrar paquetes NuGet para la solución..

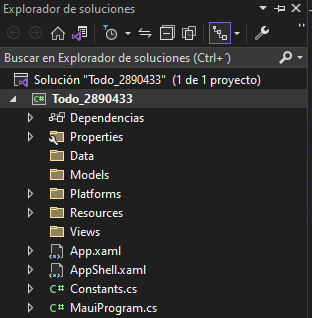
****

**Paso 8:** Estos dos paquetes son los que se tienen que instalar:

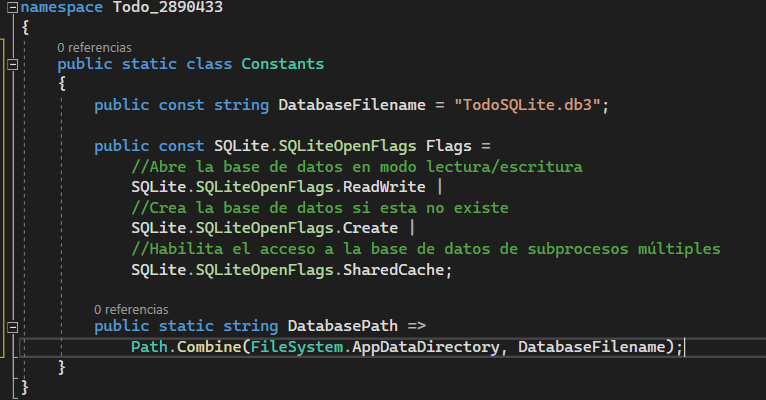




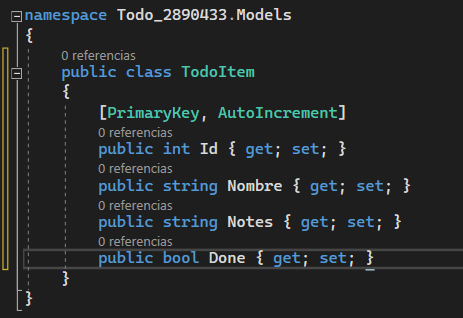
**Paso 9:** Agregamos la clase Constants al proyecto.

****

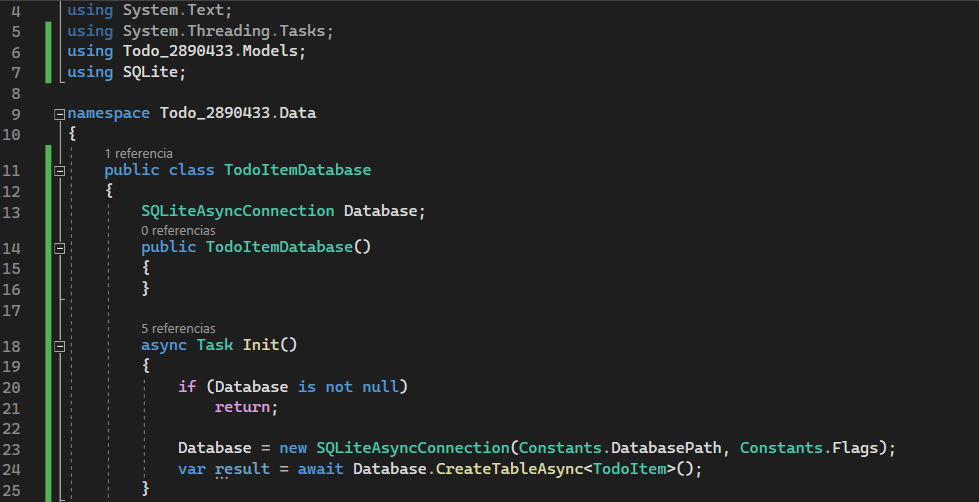
**Paso 10:** Digitamos el siguiente código en la clase Constants.cs.

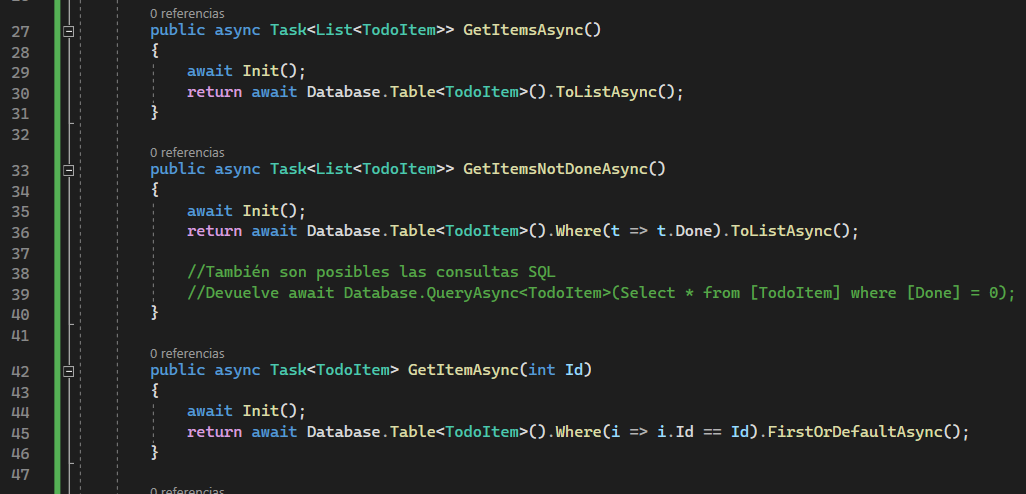
****

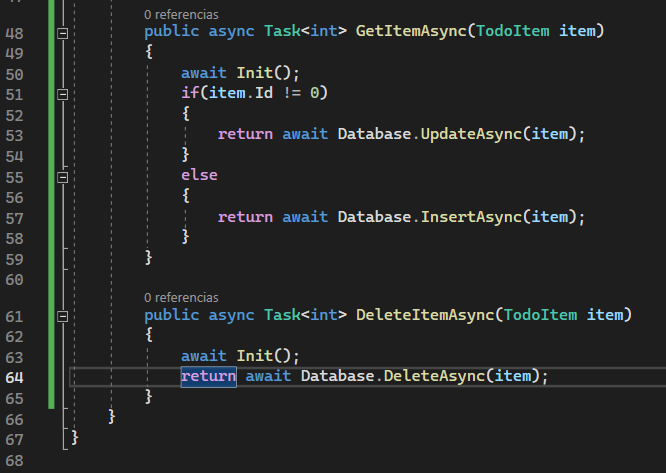
**Paso 11:** En la carpeta Models agregamos la clase TodoItem y le agregamos el siguiente código.

****

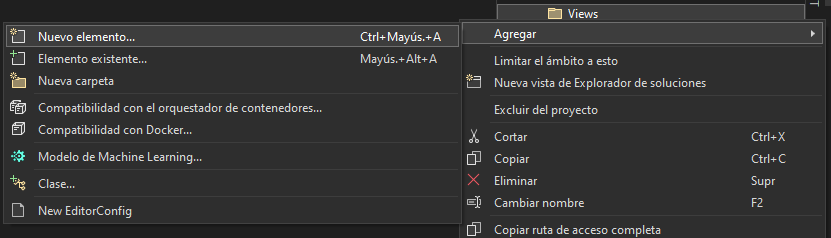
**Paso 12:** A la carpeta Data le agregamos la clase TodoItemDatabase, le ponemos sus respectivas referencias y le agregamos el siguiente código.

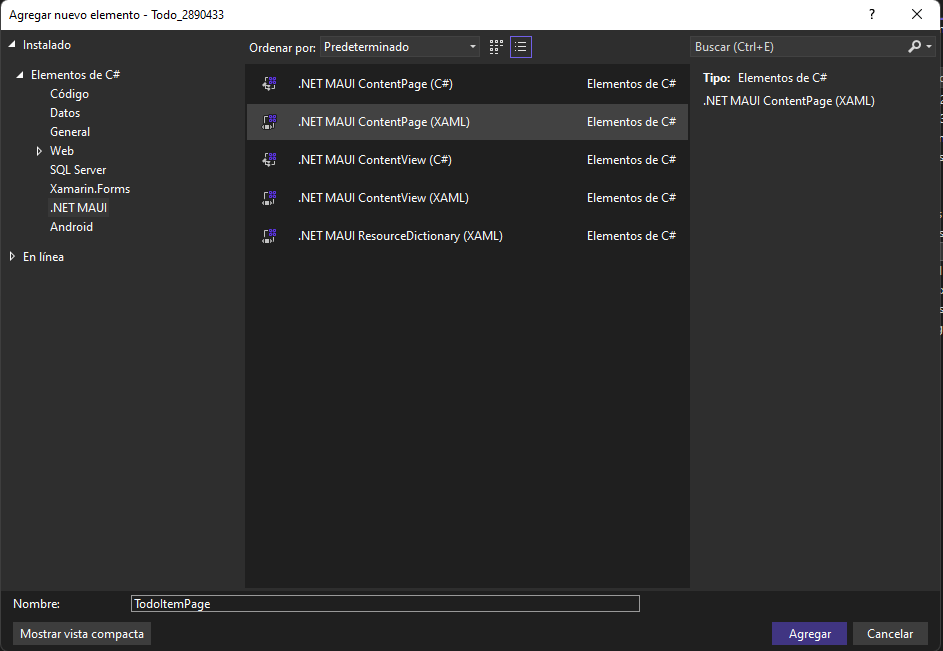
****

****

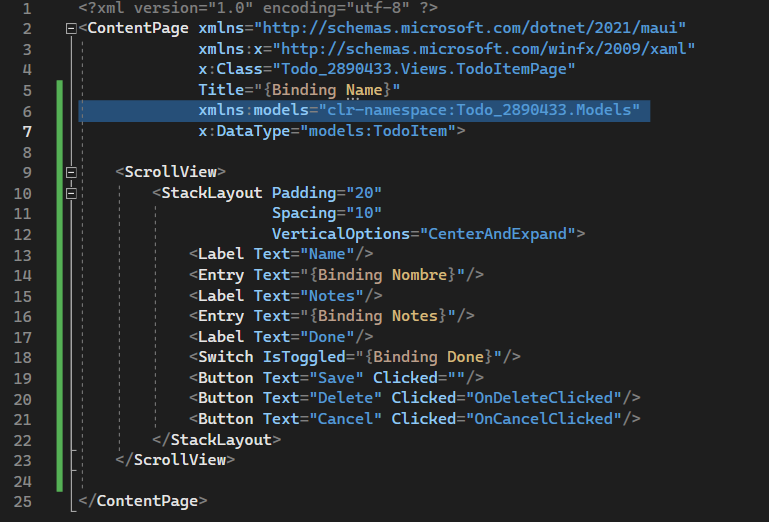
****

**Paso 13:** Empezamos a crear la interfaz agregándole una plantilla de tipo content .XAML para la aplicación y le ponemos TodoItemPage**.**

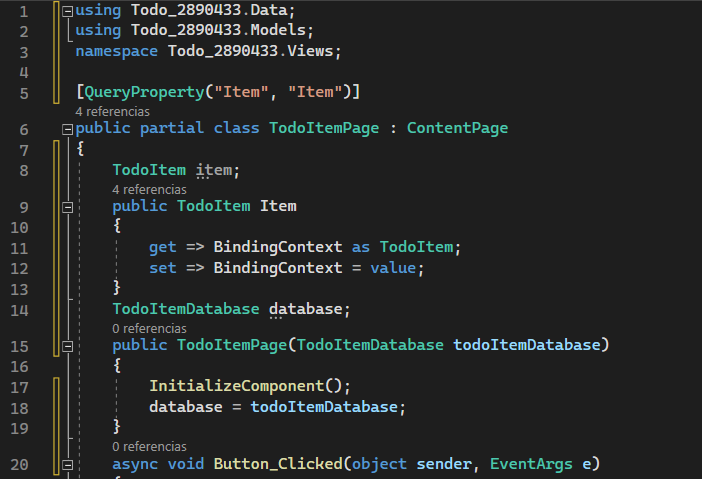
****

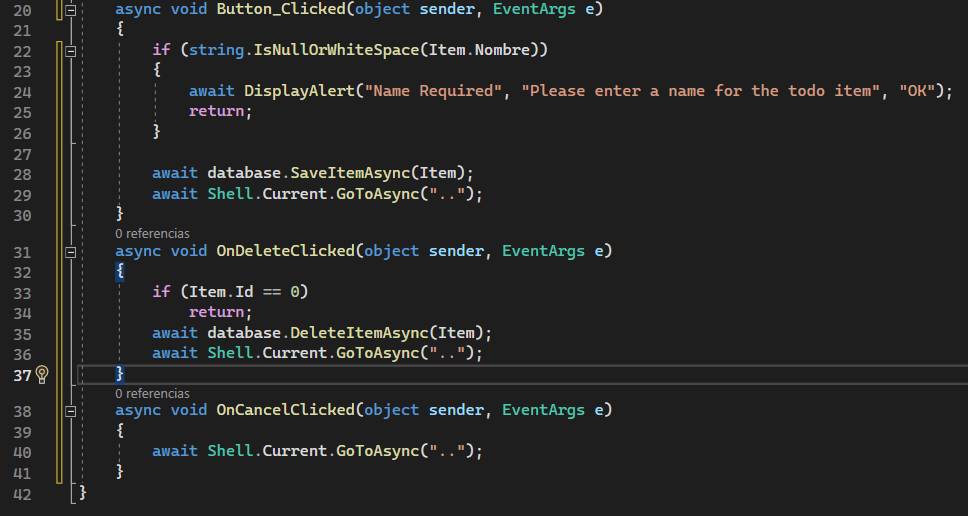
****

**Paso 14:** Le escribimos el siguiente código (la palabra binding se utiliza para traer propiedades o métodos de la clase la cual se referencia en el apartado de ContentPage.

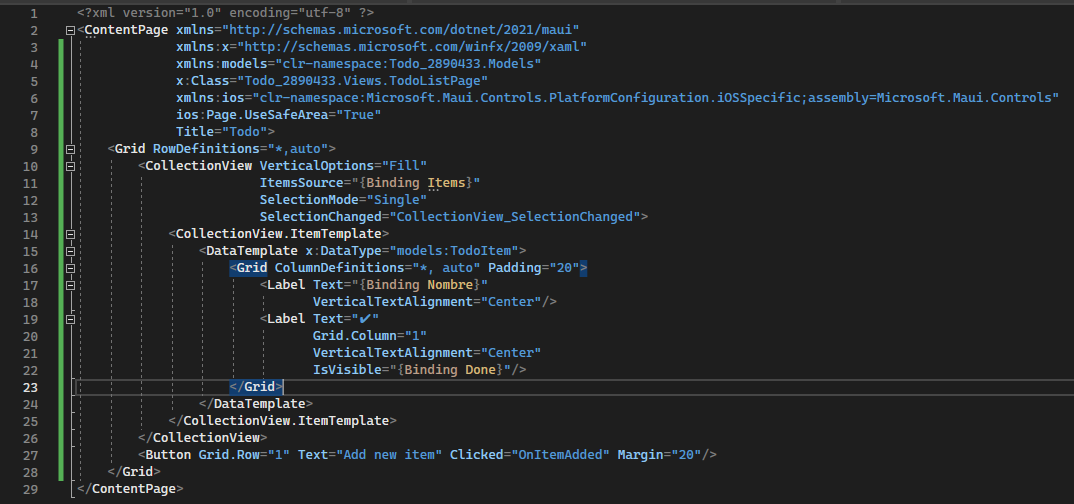
****

**Paso 15:** Nos vamos a TodoItemPage.Xaml.cs y escribimos el código, se pueden agregar las referencias manualmente o de forma dinámica.

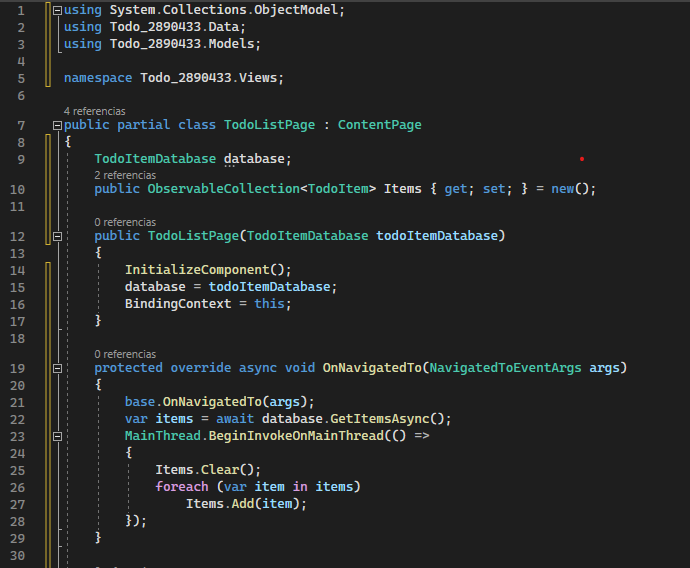
****

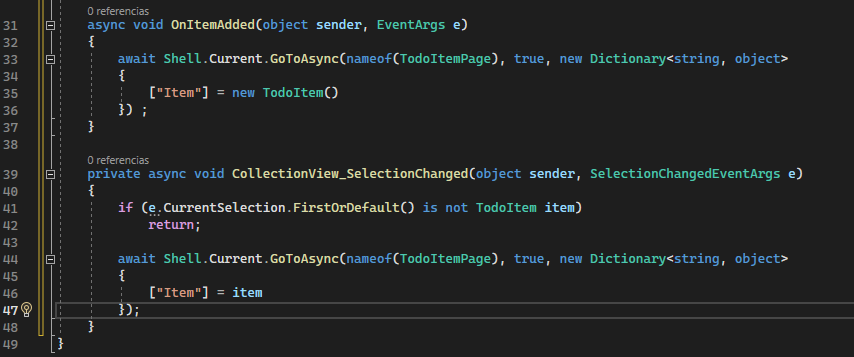
****

**Paso 16:** Escribimos el siguiente código en TodoListPage.Xaml:

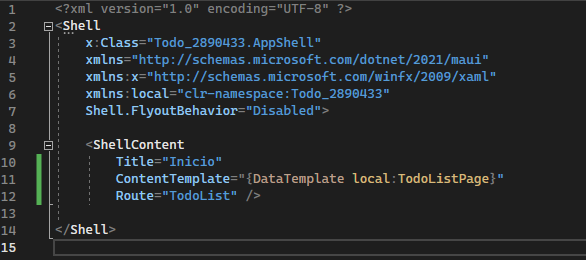
****

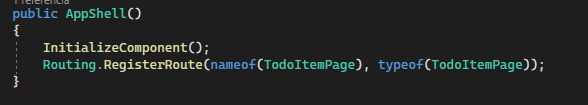
**Paso 17:** Nos vamos a TodoItemPage.Xaml.cs y digitamos el siguiente código (Nótese que primero se colocan sus respectivas referencias):

****

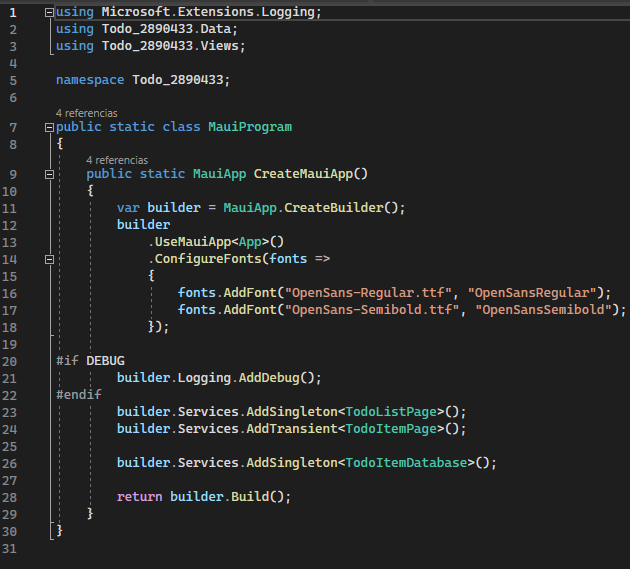
****

**Paso 18:** Vamos a AppShell.Xaml y colocamos la ruta en ContentTemplate (También se coloca xmlns:local=”clr-namespace:Todo\_2890433”):

****

**Paso 19:** Nos vamos a AppShell.Xaml.cs y colocamos la ruta:****

**Paso 20:** en **#endif;**  colocamos 3 builders, uno es para TodoListPage, el segundo es para TodoItemPage y el último es para TodoItemDatabase (No hay que olvidar las referencias):

****

**Este documento fue redactado por Fernando José Majano Aguirre con el objetivo de dar al lector una guía básica de como crear una app fácil de usar.**

**Se le agradece su atención prestada.**