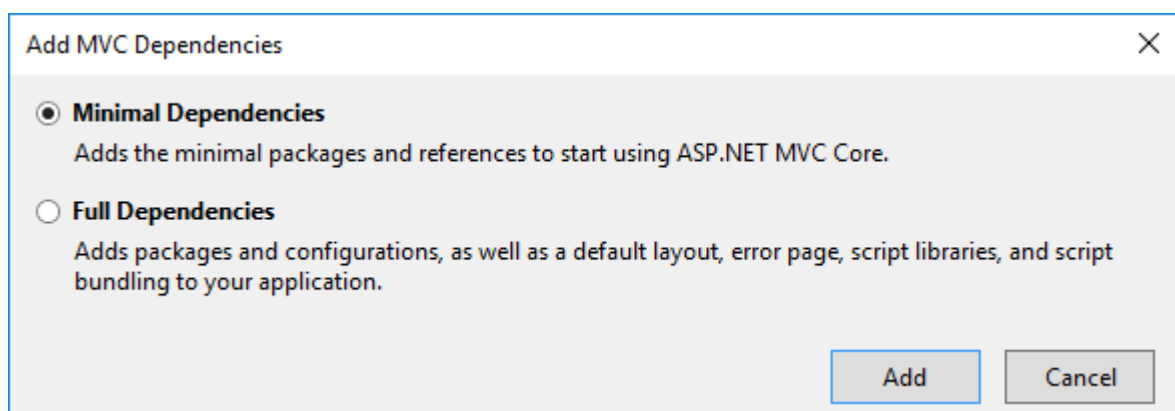


12. Controlador y vistas utilizando Scaffolding

A continuación, utilizaremos el motor scaffolding en Visual Studio para agregar un controlador MVC y vistas que utilizarán EF para consultar y guardar datos.

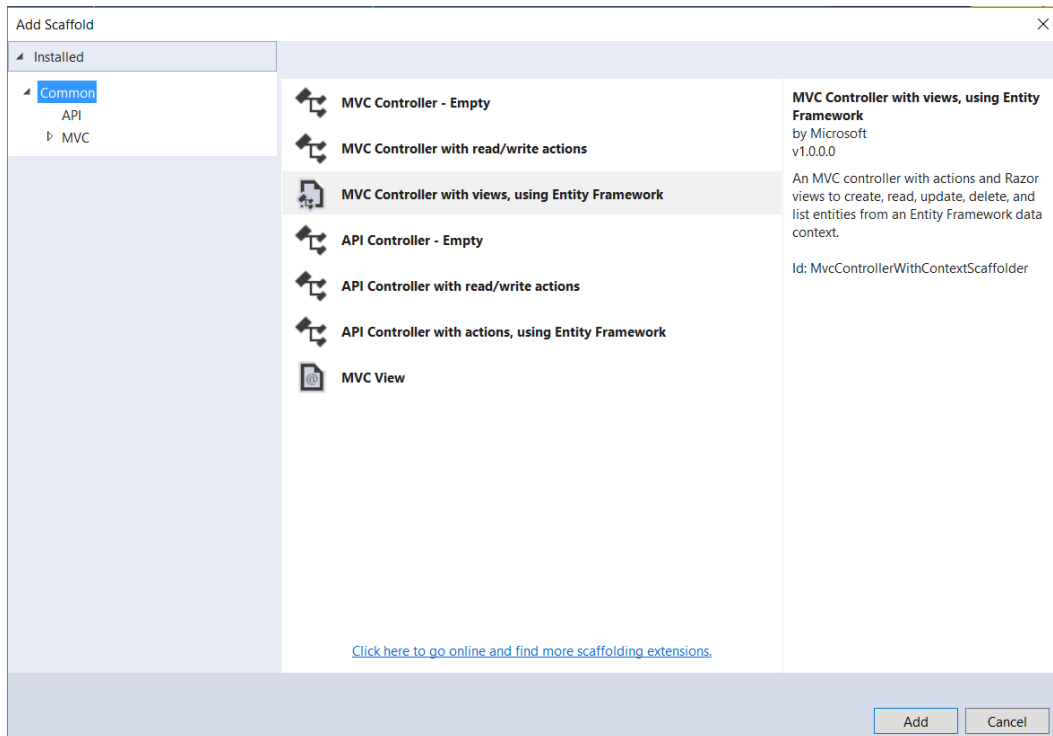
La creación automática de métodos de acción CRUD y vistas se conoce como scaffolding. El scaffolding difiere de la generación de código en que el código generado con scaffolding es un punto de partida que se puede modificar para adaptarse a sus propias necesidades, mientras que típicamente no se modifica el código generado. Cuando necesita personalizar el código generado, utiliza clases parciales o regenera el código cuando cambian las cosas.

- Click derecho en la carpeta Controllers en el Solution Explorer y seleccionar Add > New Scaffolded Item.
- En la Ventana de dialogo Add MVC Dependencies, seleccionar Minimal Dependencies, y seleccionar Add.

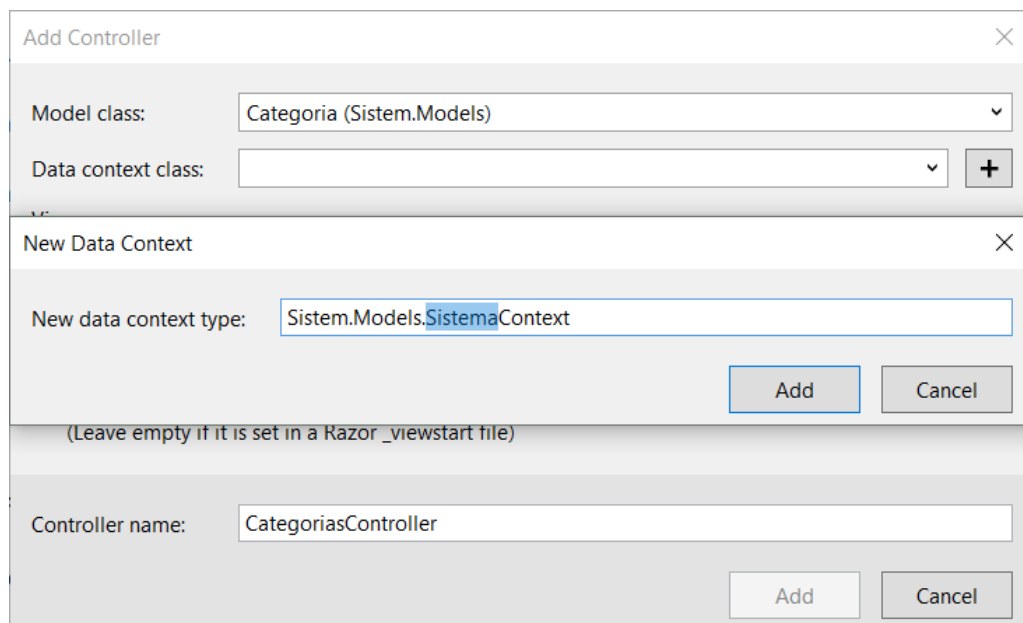


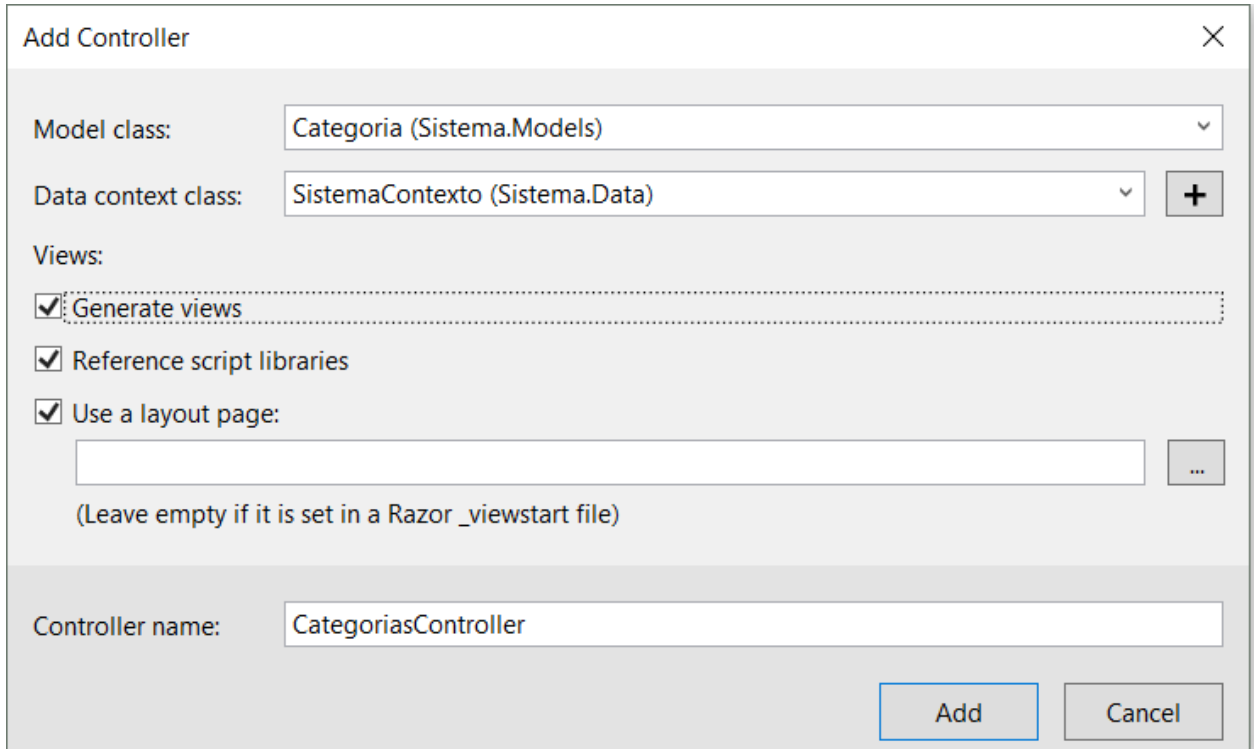
Visual Studio agrega las dependencias necesarias para el scaffolding de un controlador, incluyendo un paquete con funcionalidad EF de tiempo de diseño (`Microsoft.EntityFrameworkCore.Design`). Un paquete que se necesita sólo para scaffolding de un DbContext de una base de datos existente también se incluye (`Microsoft.EntityFrameworkCore.SqlServer.Design`). Se crea un archivo *ScaffoldingReadMe.txt* que se puede eliminar.

- Una vez más haga click con el botón derecho en la carpeta Controllers en el Solution Explorer y seleccione Add > New Scaffolded Item.
- En el cuadro de diálogo Add Scaffold:
 - Seleccione MVC controller with views, using Entity Framework.
 - Click Add.



- En el cuadro de diálogo Add Controller:
 - En Model class seleccione Categoría.
 - Click en el boton más para agregar una Data Context class, en el contexto Data context class Escriba SistemaContexto.
 - Acepte el nombre por defecto CategoríasController.
 - Click Add.





Al hacer clic en Agregar, el motor de scaffolding de Visual Studio crea un archivo `CategoriasController.cs` y un conjunto de vistas (archivos.cshtml) que funcionan con el controlador.

El motor de scaffolding también creará el contexto de la base de datos automáticamente si no lo crea manualmente.

Puede especificar una nueva clase de contexto en el cuadro Añadir controlador, haciendo clic en el signo más a la derecha de la clase de contexto Data. Visual Studio a continuación, creará su clase `DbContext`, así como el controlador y las vistas.

Notará que el controlador toma el contexto `SistemaContexto` como un parámetro de constructor.

```
namespace Sistema.Controllers
{
    public class CategoriasController : Controller
    {
        private readonly SistemaContexto _context;

        public CategoriasController(SistemaContexto context)
        {
```

```
        _context = context;  
    }
```

La inyección de dependencia de ASP.NET se encargará de pasar una instancia de `SistemaContexto` al controlador. Se ha configurado esto en el archivo `Startup.cs` automáticamente.

El controlador contiene un método de acción de `Index`, que muestra todas las categorías en la base de datos. El método obtiene una lista de categorías de la entidad `Categorias` establecida leyendo la propiedad `Categorias` de la instancia de contexto de la base de datos:

```
// GET: Categorias  
public async Task<IActionResult> Index()  
{  
    return View(await _context.Categorias.ToListAsync());  
}
```

Presione CTRL + F5 para ejecutar el proyecto o elija Debug > Start Without Debugging en el menú.

Haga clic en la ficha `Categorias` para ver los datos de prueba que el método `DbInitializer.Initialize` insertó. Dependiendo de la estrechez de la ventana del navegador, verá el vínculo de la ficha `Categorias` en la parte superior de la página o tendrá que hacer clic en el icono de navegación en la esquina superior derecha para ver el enlace.