Crear una aplicación de base de datos de la película en 15 minutos con ASP.NET MVC (C#)

El propósito de este tutorial es para darle un sentido de "lo que es" construir una aplicación ASP.NET MVC. En este tutorial, explota mediante la creación de una aplicación ASP.NET MVC toda desde el principio hasta el final. Se muestra cómo crear una aplicación de bases de datos simple que ilustra cómo puede enumerar, crear y editar registros de base de datos.
Para simplificar el proceso de creación de nuestra aplicación, a tomar ventaja de las características de andamios de Visual Studio 2008. Dejaremos que Visual Studio genera el código inicial y el contenido de nuestros controladores, modelos y vistas.
Si usted ha trabajado con Active Server Pages o ASP.NET, entonces encontrará ASP.NET MVC muy familiar. Vistas de ASP.NET MVC son muy parecido a las páginas de una aplicación de páginas Active Server. Y, al igual que una aplicación de formularios Web Forms ASP.NET tradicional, ASP.NET MVC proporciona acceso completo al conjunto de idiomas y clases de .NET framework.
Mi esperanza es que este tutorial le dará un sentido de cómo la experiencia de construir una aplicación ASP.NET MVC es similar y diferente de la experiencia de construir una aplicación de páginas Active Server o ASP.NET Web Forms.

Resumen de la aplicación de base de datos de la película

Porque nuestro objetivo es mantener las cosas simples, a construir una aplicación muy sencilla de Movie Database.Nuestra simple Movie Database aplicación nos permite hacer tres cosas:

1. Enumerar un conjunto de registros de base de datos de la película
2. Crear un nuevo registro de base de datos de la película
3. Editar un registro de base de datos existente de película

Una vez más, porque queremos mantener las cosas simples, tomaremos ventaja del número mínimo de características de ASP.NET MVC framework necesarios para construir nuestra aplicación. Por ejemplo, no tomar ventaja del desarrollo de Test-Driven.
Para crear nuestra aplicación, necesitamos completar cada uno de los siguientes pasos:

1. Crear el proyecto de aplicación Web de ASP.NET MVC
2. Crear la base de datos
3. Crear el modelo de base de datos
4. Crear el controlador de ASP.NET MVC
5. Crear el ASP.NET MVC views

Crear un proyecto de aplicación Web de ASP.NET MVC

Vamos a empezar por crear un nuevo proyecto de aplicación Web de ASP.NET MVC en Visual Studio 2008.Seleccione la opción de menú **archivo, nuevo proyecto** y verá el cuadro de diálogo nuevo proyecto en la figura 1.Seleccione C# como lenguaje de programación y seleccione la plantilla de proyecto de aplicación Web de ASP.NET MVC. Dar al proyecto el nombre MovieApp y haga clic en el botón Aceptar.

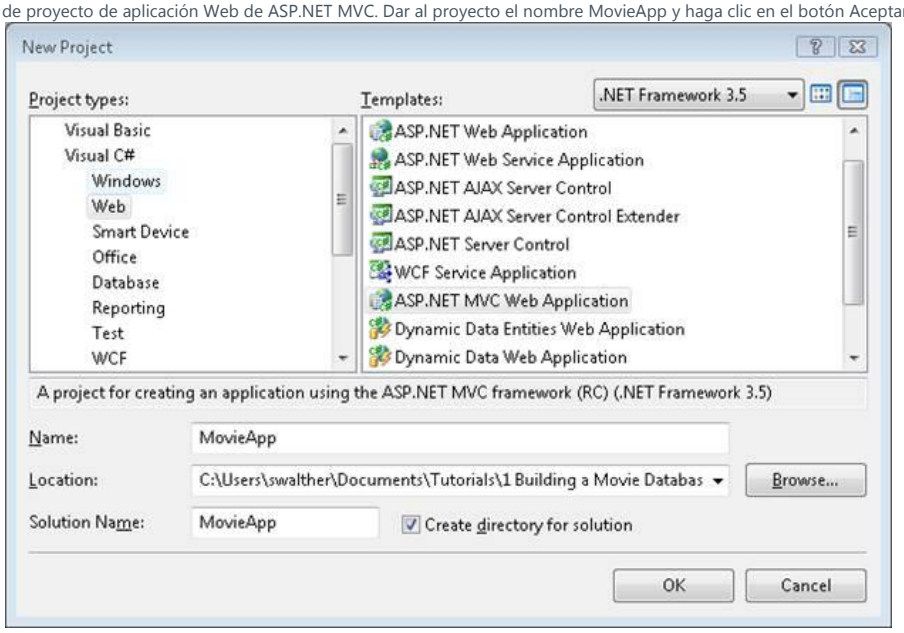

Figura 01: cuadro de diálogo nuevo proyecto (haga clic para ver imagen en tamaño completo)
Asegúrese de que selecciona el .NET Framework 3.5 en la lista desplegable en la parte superior del cuadro de diálogo nuevo proyecto o la plantilla de proyecto de aplicación Web de ASP.NET MVC no aparecerá.

Cuando se crea un nuevo proyecto de aplicación Web de MVC, Visual Studio le pedirá que cree un proyecto de prueba de unidad independiente. Aparece el cuadro de diálogo en la figura 2. Porque no se crear pruebas en este tutorial debido a limitaciones de tiempo (y, sí, debemos sentir un poco culpables por esto) no seleccione la opcióny haga clic en el botón **OK** .
Visual Web Developer no admite proyectos de prueba.

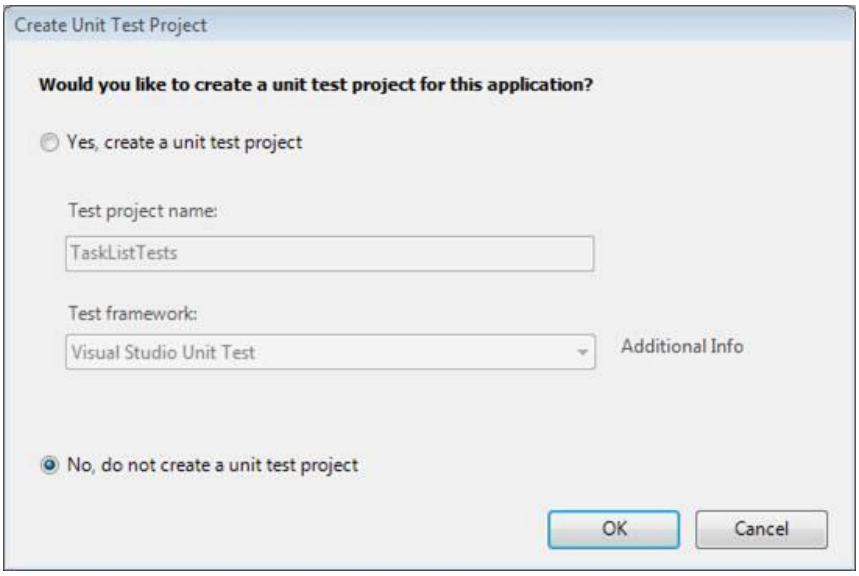

Figura 02: diálogo de unidad de crear el proyecto de prueba (haga clic para ver imagen en tamaño completo)
Una aplicación ASP.NET MVC tiene un conjunto estándar de carpetas: una carpeta modelos, vistas y controladores.Puedes ver este conjunto estándar de carpetas en la ventana Explorador de soluciones. Necesitaremos agregar archivos a cada una de las carpetas de modelos, vistas y controladores para construir nuestra aplicación Movie Database.
Cuando se crea una nueva aplicación de MVC con Visual Studio, se obtiene una aplicación de muestra. Porque queremos empezar de cero, tenemos que eliminar el contenido de esta aplicación de ejemplo. Es necesario eliminar el siguiente archivo y la carpeta siguiente:

- Controllers\HomeController.CS
- Views\Home

Crear la base de datos

Necesitamos crear una base de datos para mantener nuestros registros de base de datos de la película.Afortunadamente, Visual Studio incluye una base de datos denominado SQL Server Express. Siga estos pasos para crear la base de datos:

1. Haga clic en la carpeta App_Data en la ventana Explorador de soluciones y seleccione la opción de menú**Agregar, nuevo elemento**.
2. Seleccione la categoría de **datos** y seleccione la plantilla de **Base de datos de SQL Server** (ver figura 3).
3. Nombre de su nueva base de datos *MoviesDB.mdf* y haga clic en el botón **Agregar** .

Después de crear la base de datos, puede conectar a la base de datos haciendo doble clic en el archivo MoviesDB.mdf ubicado en la carpeta App_Data. Doble clic en el archivo MoviesDB.mdf se abre la ventana del explorador de servidores.
La ventana del explorador de servidores se llama la ventana del explorador de base de datos en el caso de Visual Web Developer.

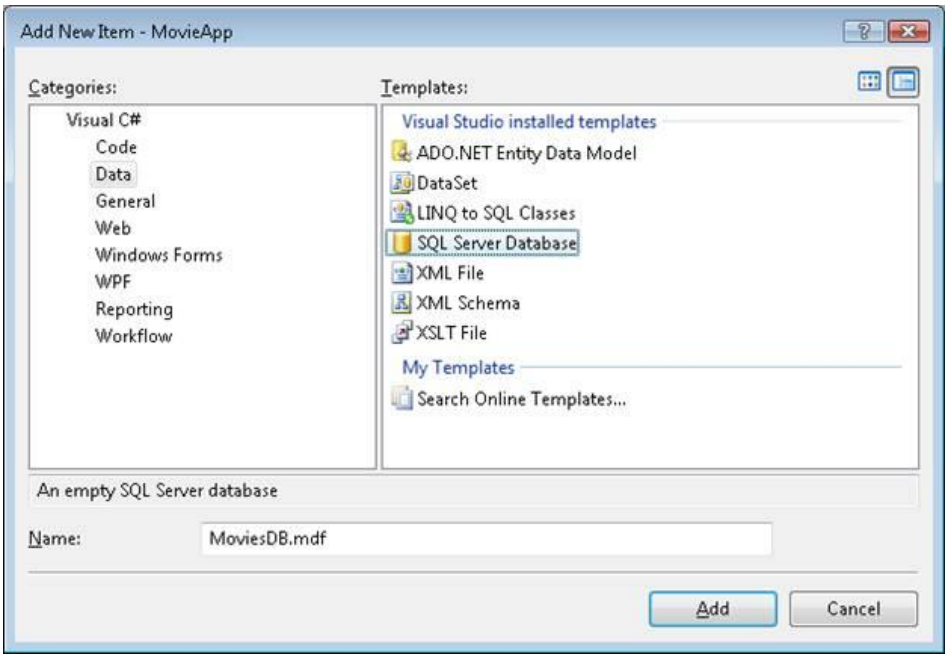

Figura 03: creación de una base de datos de Microsoft SQL Server (haga clic para ver imagen en tamaño completo)
A continuación, tenemos que crear una nueva tabla de base de datos. Desde dentro de la ventana del explorador de servidores, haga clic en la carpeta tablas y seleccione la opción **Agregar nueva tabla**. Al seleccionar esta opción de menú abre el diseñador de tablas de base de datos. Crear las siguientes columnas de base de datos:

Nombre de la columna	Tipo de datos	Permitir valores nulos
ID	Int	Falso
Título	Nvarchar(100)	Falso
Director	Nvarchar(100)	Falso
DateReleased	Fecha y hora	Falso

La primera columna, la columna Id, tiene dos propiedades especiales. En primer lugar, es necesario marcar la columna de Id como la columna de clave principal. Después de seleccionar la columna Id, haga clic en el botón**Establecer clave principal** (es el icono que parece una clave). En segundo lugar, es necesario marcar la columna de Id como una columna de identidad. En la ventana de propiedades de columna, desplácese hacia abajo hasta la sección de especificaciones de identidad y expandirlo. Cambiar la propiedad de **Identidad es** el valor **Sí**. Cuando haya terminado, la tabla debe parecerse a Figura 4.

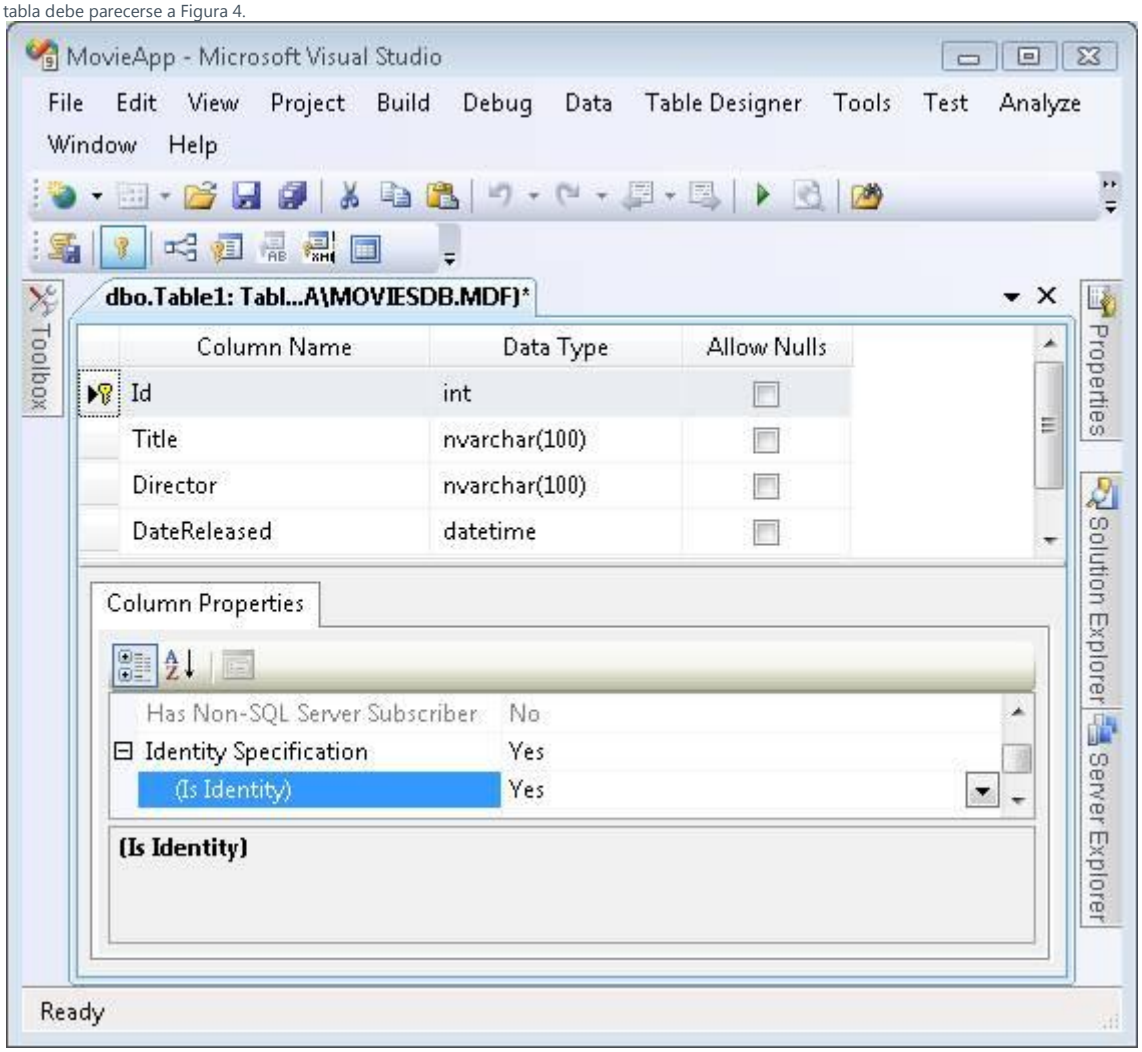

Figura 04: películas de la base de datos de tabla (haga clic para ver imagen en tamaño completo)
El paso final es guardar la nueva tabla. Haga clic en el botón Guardar (el icono del disquete) y dar a la nueva tabla las películas de nombre. Cuando termine de crear la tabla, agregar algunos registros de película a la tabla. Haga clic en la tabla de películas en la ventana del explorador de servidores y seleccione la opción de menú **Mostrar tabla de datos**.Escriba una lista de sus películas favoritas (ver figura 5).

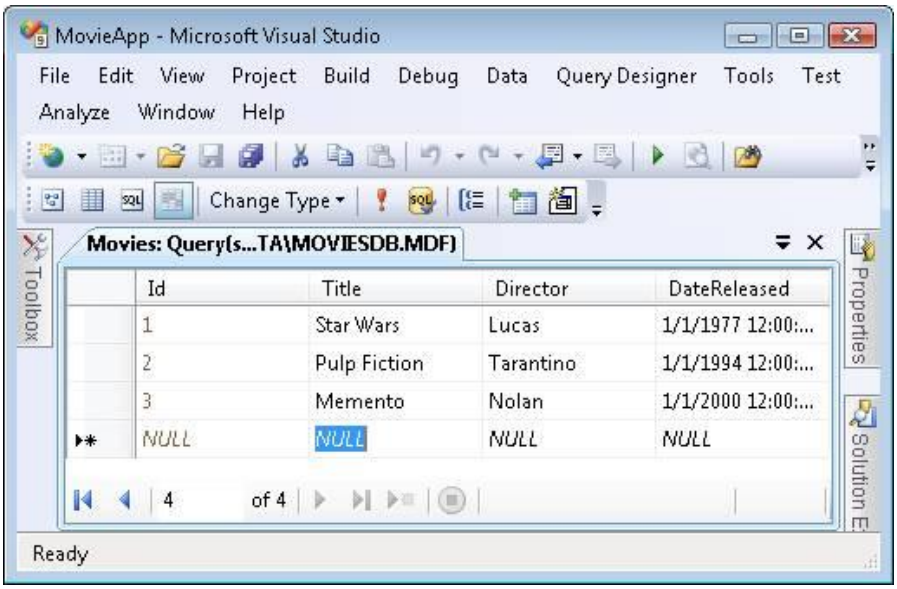

Figura 05: película entrar registros (haga clic para ver imagen en tamaño completo)

Crear el modelo

A continuación tenemos que crear un conjunto de clases para representar a nuestra base de datos. Tenemos que crear un modelo de base de datos. Tomaremos ventaja de Microsoft Entity Framework para generar las clases para nuestro modelo de base de datos automáticamente.
El marco de ASP.NET MVC no está vinculado a la Microsoft Entity Framework. Puede crear la base de datos modelo clases aprovechando las ventajas de una variedad de mapeo relacional de objetos (o / M) herramientas incluyendo LINQ to SQL, subsónico y NHibernate.

Siga estos pasos para iniciar al Asistente de modelo de datos de entidad:

1. Haga clic en la carpeta de modelos en la ventana Explorador de soluciones y la seleccione del menú la opción**Agregar, nuevo elemento**.
2. Seleccione la categoría de **datos** y seleccione la plantilla de **ADO.NET Entity Data Model** .
3. Dar el modelo de datos el nombre *MoviesDBModel.edmx* y haga clic en el botón **Agregar** .

Después de hacer clic en el botón Agregar, Asistente de Entity Data Model aparece (véase figura 6). Siga estos pasos para completar al asistente:

1. En el paso de **Elegir modelo de contenido** , seleccione la opción **generar de base de datos** .
2. En el paso de **Elegir la conexión de datos** , utilice la conexión de datos de *MoviesDB.mdf* y el nombre de*MoviesDBEntities* para la configuración de la conexión. Haga clic en el botón **siguiente** .
3. En el paso **Elija los objetos de base de datos** , expanda el nodo tablas, seleccione la tabla de películas.Entrar en el espacio de nombres *MovieApp.Models* y haga clic en el botón **Finalizar** .

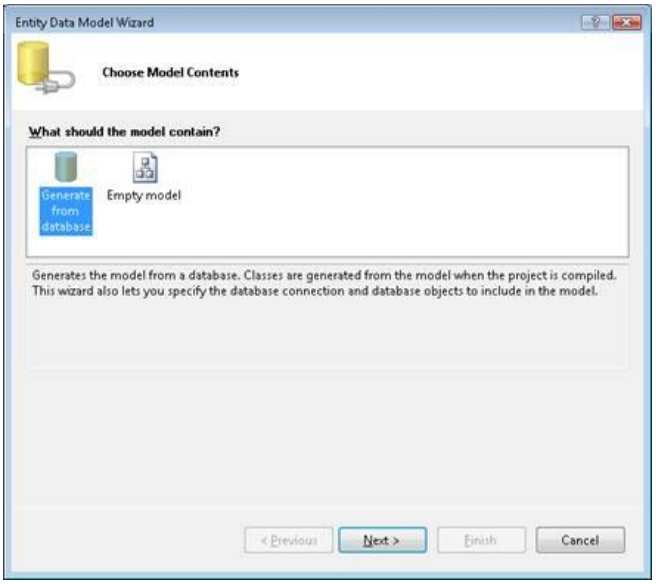

Figura 06: generar un modelo de base de datos con el Asistente de Entity Data Model (haga clic para ver imagen en tamaño completo)
Después de completar al Asistente para modelo de datos de entidad, se abre el Entity Data Model Designer. El diseñador debe mostrar la tabla de base de datos de películas (ver figura 7).

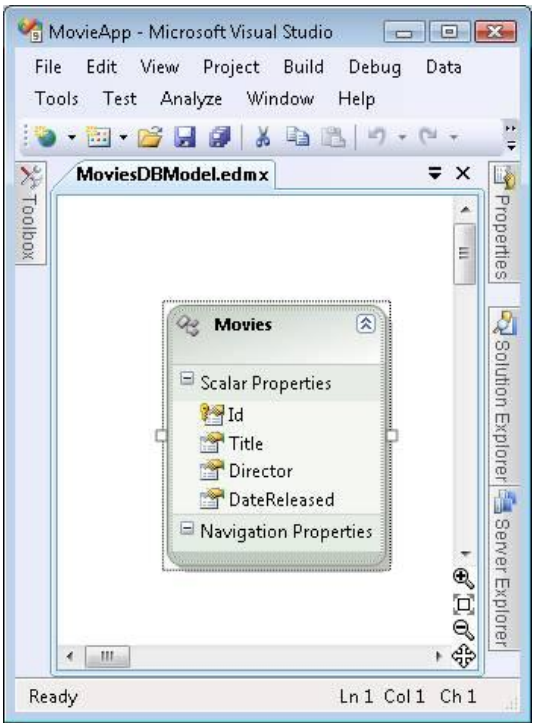

Figura 07: el Entity Data Model Designer (haga clic para ver imagen en tamaño completo)

Tenemos que hacer un cambio antes de continuar. El Asistente para datos de entidad genera una clase de modelo denominada *películas* que representa la tabla de base de datos de películas. Porque vamos a usar la clase de películas para representar una película especial, debemos modificar el nombre de la clase como *película* en lugar de *películas* (singular en lugar de plural).

Haga doble clic en el nombre de la clase en la superficie del diseñador y cambiar el nombre de la clase de películas a la película. Después de realizar este cambio, haga clic en el botón **Guardar** (el icono del disquete) para generar la clase de película.

Crear el controlador de ASP.NET MVC

El siguiente paso es crear el controlador de ASP.NET MVC. Un controlador es responsable de controlar cómo un usuario interactúa con una aplicación ASP.NET MVC.

Siga estos pasos:

1. En la ventana Explorador de soluciones, haga clic en la carpeta de controladores y seleccione la opción de menú **Agregar, regulador**.
2. En el cuadro de diálogo del controlador de agregar, escriba el nombre de *HomeController* y compruebe la casilla de verificación **agregar métodos de acción para crear, actualizar y detalles de escenarios** (ver figura 8).
3. Haga clic en el botón **Agregar** para añadir el nuevo controlador a su proyecto.

Después de completar estos pasos, se crea el controlador en la lista 1. Observe que contiene métodos denominados índice, detalles, crear y editar. En las secciones siguientes, añadiremos el código necesario para obtener estos métodos para trabajar.

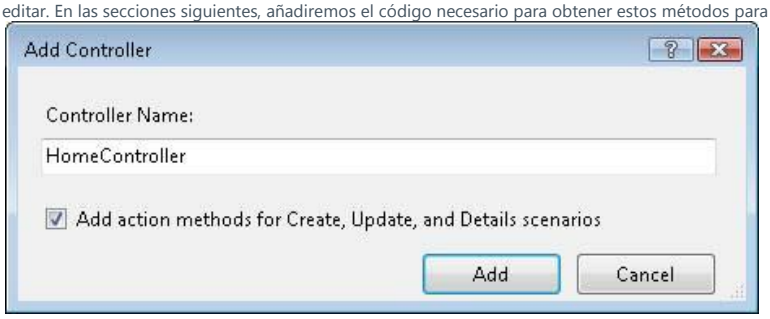

Figura 08: agregar un nuevo de ASP.NET MVC controlador (haga clic para ver imagen en tamaño completo)

Listado 1 – Controllers\HomeController.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Mvc.Ajax;

namespace MovieApp.Controllers
```

```
{
    public class HomeController : Controller
    {
        //
        // GET: /Home/
        public ActionResult Index()
        {
            return View();
        }
        //
        // GET: /Home/Details/5
        public ActionResult Details(int id)
        {
            return View();
        }
        //
        // GET: /Home/Create
        public ActionResult Create()
        {
            return View();
        }
        //
        // POST: /Home/Create
        [AcceptVerbs(HttpVerbs.Post)]
        public ActionResult Create(FormCollection collection)
        {
            try
            {
                // TODO: Add insert logic here
                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
        //
        // GET: /Home/Edit/5
        public ActionResult Edit(int id)
        {
```

```
            return View();
        }
        //
        // POST: /Home/Edit/5
        [AcceptVerbs(HttpVerbs.Post)]
        public ActionResult Edit(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add update logic here
                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
    }
}
```

Listado de registros de base de datos

El método Index() de la controladora de inicio es el método predeterminado para una aplicación ASP.NET MVC.Cuando se ejecuta una aplicación ASP.NET MVC, el método Index() es el primer método de controlador que se llama.
Utilizaremos el método Index() para mostrar la lista de registros de la tabla de base de datos de películas. Clases de modelo que hemos creado anteriormente para recuperar los registros de base de datos de la película con el método Index() tomaremos ventaja de la base de datos.
Que he modificado la clase HomeController en el listado 2, por lo que contiene un campo privado nuevo llamado _db. La clase MoviesDBEntities representa nuestro modelo de base de datos y vamos a usar esta clase para comunicarse con nuestra base de datos.
Que también he modificado el método Index() en el listado 2. El método Index() utiliza la clase MoviesDBEntities para recuperar todos los registros de la película de la tabla de base de datos de películas. La expresión *_db.MovieSet.ToList()* devuelve una lista de todos los registros de la película de la tabla de base de datos de películas.
La lista de películas se pasa a la vista. Cualquier cosa que se pasa al método View() se pasa a la vista como ver datos.
Listado de 2 – Controllers/HomeController.cs (método de índice modificado)

```
using System.Linq;
using System.Web.Mvc;
using MovieApp.Models;
namespace MovieApp.Controllers
{
    public class HomeController : Controller
    {
        private MoviesDBEntities _db = new MoviesDBEntities();
        public ActionResult Index()
        {
            return View(_db.MovieSet.ToList());
        }
    }
```

```
    }
```

El método Index() devuelve una vista denominada índice. Tenemos que crear esta vista para mostrar la lista de registros de base de datos de la película. Siga estos pasos:

Debe construir su proyecto (seleccione la opción de menú **construir, generar solución**) antes de abrir el cuadro de diálogo **Agregar vista** o no clases aparecerá en la lista desplegable de **clase de datos de la vista** .

1. Haga clic en el método Index() en el editor de código y seleccione la opción de menú **Añadir vista** (ver Figura 9).
2. En el cuadro de diálogo Agregar vista, compruebe que está activada la casilla de verificación **crear una inflexible vista** .
3. En la lista desplegable **Ver contenido** , seleccione el valor de la *lista*.
4. En la lista desplegable de la **clase de datos de la vista** , seleccione el valor *MovieApp.Models.Movie*.
5. Haga clic en el botón Agregar para crear el nuevo ver (ver Figura 10).

Después de completar estos pasos, una vista nueva denominada Index.aspx se agrega a la carpeta Views\Home. El contenido de la vista de índice se incluye en el listado 3.

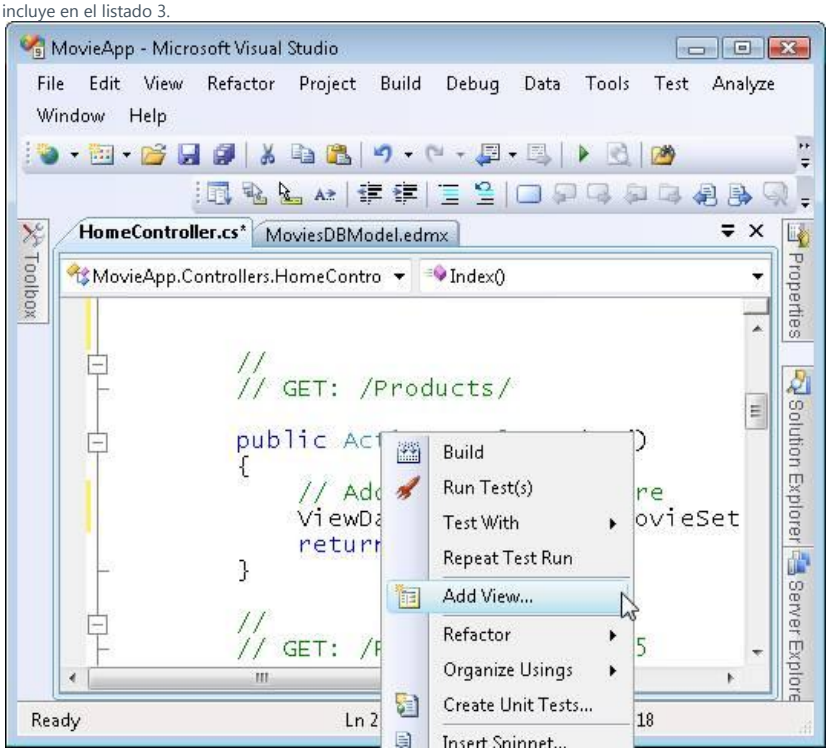

Figura 09: adición de una vista de una acción del controlador (haga clic para ver imagen en tamaño completo)

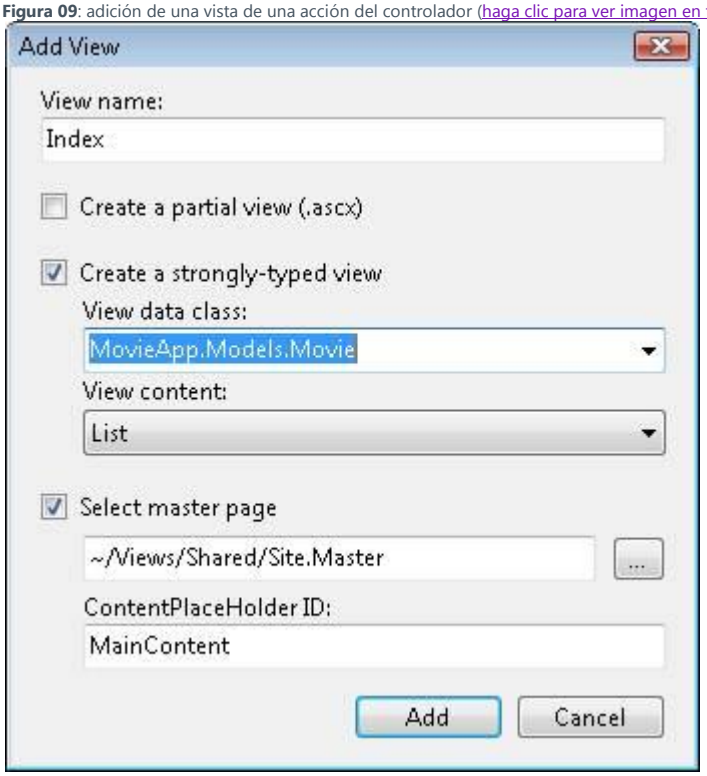

Figura 10: crear una nueva vista con el cuadro de diálogo Agregar vista (haga clic para ver imagen en tamaño completo)

Listado 3: Views\Home\Index.aspx

```
    <%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master"
Inherits="System.Web.Mvc.ViewPage<IEnumerable<MovieApp.Models.Movie>>" %>

    <asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

       Index

    </asp:Content>

    <asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

        <h2>Index</h2>

        <table>

            <tr>

                <th></th>

                <th>

                    Id

                </th>

                <th>

                    Title

                </th>

                <th>

                    Director

                </th>

                <th>

                    DateReleased

                </th>

            </tr>

        <% foreach (var item in Model) { %>


            <tr>

                <td>

                    <%= Html.ActionLink("Edit", "Edit", new { id=item.Id }) %> |

                    <%= Html.ActionLink("Details", "Details", new { id=item.Id })%>

                </td>

                <td>

                    <%= Html.Encode(item.Id) %>

                </td>

                <td>

                    <%= Html.Encode(item.Title) %>

                </td>

                <td>

                    <%= Html.Encode(item.Director) %>

                </td>

                <td>
```

```
                <%= Html.Encode(String.Format("{0:g}", item.DateReleased)) %>

            </td>

        </tr>


    <% } %>

    </table>

    <p>

        <%= Html.ActionLink("Create New", "Create") %>

    </p>

</asp:Content>
```

La vista de índice muestra todos los registros de la película de la tabla de base de datos de películas dentro de una tabla HTML. La vista contiene un bucle foreach que recorre cada película representado por la propiedad ViewData.Model. Si ejecuta la aplicación pulsando la tecla F5, verá la página web en la figura 11.

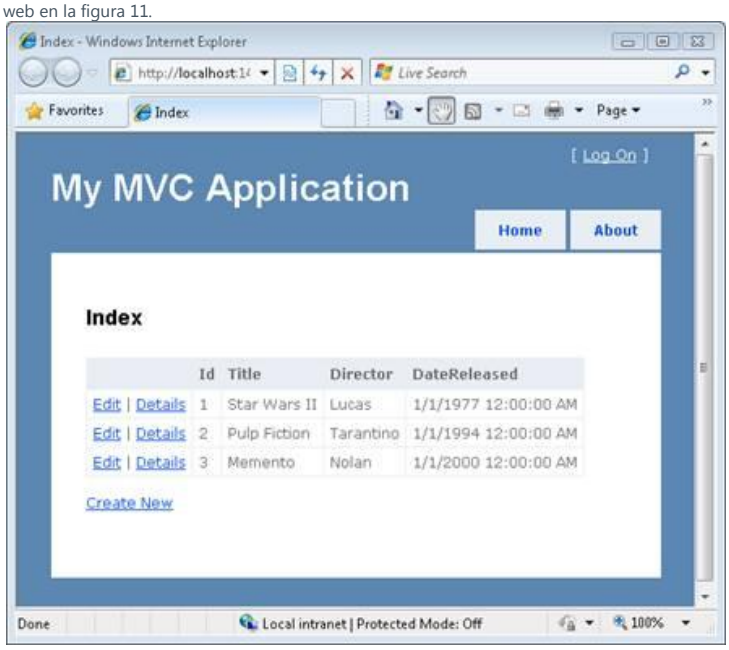

Figura 11: vista del índice (haga clic para ver imagen en tamaño completo)

Creación de nuevos registros de base de datos

La vista de índice que hemos creado en la sección anterior incluye un enlace para crear nuevos registros de base de datos. Sigamos adelante y aplicar la lógica y crear la vista necesaria para crear nuevos registros de base de datos de la película.
El controlador de inicio contiene dos métodos denominados Create. El primer método Create no tiene parámetros.Esta sobrecarga del método Create se utiliza para mostrar el formulario HTML para crear un nuevo registro de base de datos de la película.
El segundo método Create tiene un parámetro FormCollection. Esta sobrecarga del método Create es llamada cuando el formulario HTML para la creación de una nueva película se registra en el servidor. Observe que este segundo método Create tiene un atributo de AcceptVerbs que impide que el método que se llama a menos que se realiza una operación POST de HTTP.
Se ha modificado este segundo método Create en la clase HomeController actualizada en el listado 4. La nueva versión del método Create acepta un parámetro de la película y contiene la lógica para insertar una nueva película en la tabla de base de datos de películas.
Observe el atributo de enlace. Porque no queremos actualizar la propiedad Id de la película de formulario HTML, debemos excluir explícitamente esta propiedad.

Listado de 4 – Controllers\HomeController.cs (método de crear modificado)

```
//

        // GET: /Home/Create

        public ActionResult Create()

        {

            return View();
```

```
        }

        //

        // POST: /Home/Create

        [AcceptVerbs(HttpVerbs.Post)]

        public ActionResult Create([Bind(Exclude="Id")] Movie movieToCreate)

        {

             if (!ModelState.IsValid)

                return View();

            _db.AddToMovieSet(movieToCreate);

            _db.SaveChanges();

            return RedirectToAction("Index");

        }
```

Visual Studio hace que sea fácil crear el formulario para crear una nueva base de datos de película grabar (ver figura 12). Siga estos pasos:

1. Haga clic en el método Create en el editor de código y seleccione la opción de menú **Añadir vista**.
2. Compruebe que está activada la casilla de verificación **crear una inflexible vista** .
3. En la lista desplegable **Ver contenido** , seleccione el valor de *crear*.
4. En la lista desplegable de la **clase de datos de la vista** , seleccione el valor *MovieApp.Models.Movie*.
5. Haga clic en el botón **Agregar** para crear la nueva vista.

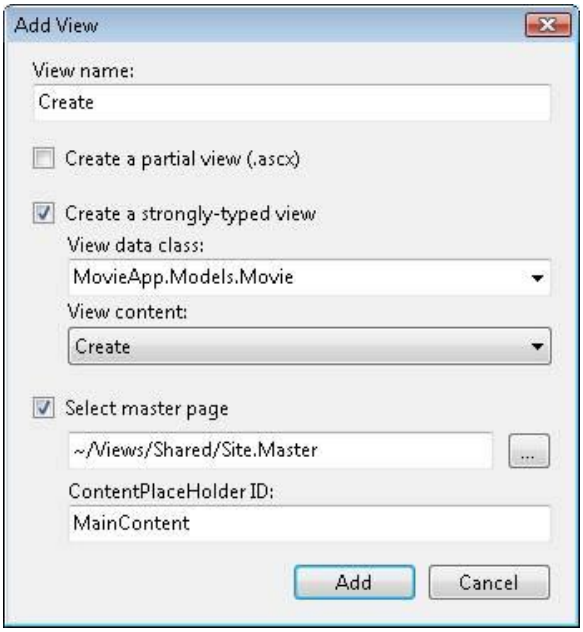

Figura 12: agregar a la vista de crear (haga clic para ver imagen en tamaño completo)

Visual Studio genera automáticamente la vista en el listado 5. Esta vista contiene un formulario HTML que incluye los campos que corresponden a cada una de las propiedades de la clase de película.

Listado de 5 – Views\Home\Create.aspx

```
    <%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master"
Inherits="System.Web.Mvc.ViewPage<MovieApp.Models.Movie>" %>

    <asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">

       Create

    </asp:Content>

    <asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

        <h2>Create</h2>

        <%= Html.ValidationSummary("Create was unsuccessful. Please correct the errors and try again.") %>

        <% using (Html.BeginForm()) {%>
```

```
        <fieldset>
            <legend>Fields</legend>
            <p>
                <label for="Id">Id:</label>
                <%= Html.TextBox("Id") %>
                <%= Html.ValidationMessage("Id", "*") %>
            </p>
            <p>
                <label for="Title">Title:</label>
                <%= Html.TextBox("Title") %>
                <%= Html.ValidationMessage("Title", "*") %>
            </p>
            <p>
                <label for="Director">Director:</label>
                <%= Html.TextBox("Director") %>
                <%= Html.ValidationMessage("Director", "*") %>
            </p>
            <p>
                <label for="DateReleased">DateReleased:</label>
                <%= Html.TextBox("DateReleased") %>
                <%= Html.ValidationMessage("DateReleased", "*") %>
            </p>
            <p>
                <input type="submit" value="Create" />
            </p>
        </fieldset>
    <% } %>
    <div>
        <%=Html.ActionLink("Back to List", "Index") %>
    </div>
</asp:Content>
```

El formulario HTML generado por el cuadro de diálogo Agregar vista genera un campo de formulario de identificación. Porque la columna de Id es una columna de identidad, no necesitamos de este campo del formulario y se puede eliminar con seguridad.

Después de agregar la vista de crear, puede agregar nuevos registros de película a la base de datos. Ejecute la aplicación pulsando la tecla F5 y haga clic en el vínculo crear nuevo para ver la forma en la figura 13. Si completar y enviar el formulario, se crea un nuevo registro de base de datos de la película.
Aviso que obtendrá automáticamente validación de formularios. Si descuide a introducir una fecha de lanzamiento para una película, o introducir una fecha de lanzamiento no es válido, entonces se vuelve a mostrar el formulario y se resalte el campo de fecha de lanzamiento.

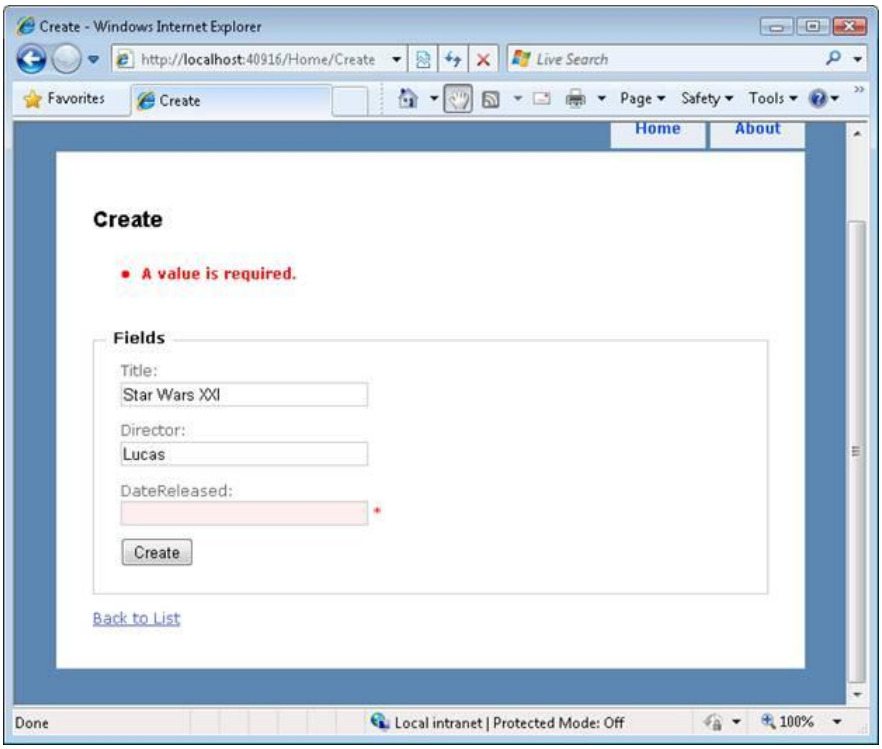

Figura 13: crear un nuevo registro de base de datos de película (haga clic para ver imagen en tamaño completo)

Edición de registros de base de datos existentes

En las secciones anteriores, hablamos de cómo puede una lista y crear nuevos registros de base de datos. En esta última sección, discutimos cómo puede editar registros existentes de la base de datos.

En primer lugar, tenemos que generar el formulario de edición. Este paso es fácil, ya que Visual Studio se generará automáticamente el formulario de edición para nosotros. Abrir la clase HomeController.cs en el editor de código de Visual Studio y siga estos pasos:

1. Haga clic en el método Edit() en el editor de código y seleccione la opción de menú **Añadir vista** (ver figura 14).
2. Compruebe la casilla de verificación **crear una vista inflexible**.
3. En la lista desplegable **Ver contenido** , seleccione el valor *Editar*.
4. En la lista desplegable de la **clase de datos de la vista** , seleccione el valor *MovieApp.Models.Movie*.
5. Haga clic en el botón **Agregar** para crear la nueva vista.

Completar estos pasos, añade una nueva vista con el nombre Edit.aspx a la carpeta Views\Home. Esta vista contiene un formulario HTML para editar un registro de la película.

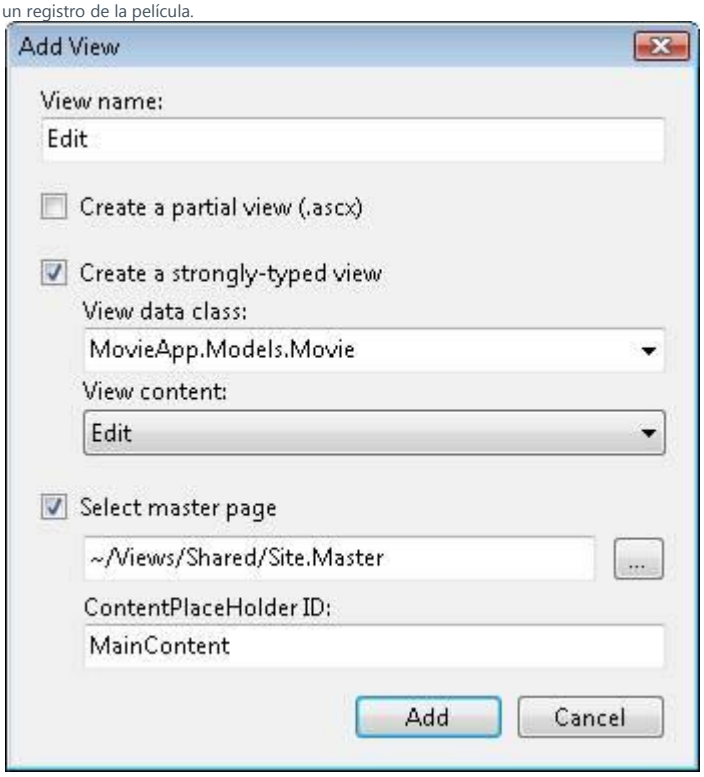

Figura 14: agregar la vista Editar (haga clic para ver imagen en tamaño completo)
La vista Editar contiene un campo de formulario HTML que corresponde a la propiedad Id de la película. Porque no desea editar el valor de la propiedad Id de personas, se debe eliminar este campo del formulario.

Por último, necesitamos modificar el controlador de la casa para que soporta la edición de un registro de base de datos. La clase HomeController actualizada figura en el listado 6.
Listado de 6 – Controllers\HomeController.cs (métodos de edición)

```
    //
    // GET: /Home/Edit/5
    public ActionResult Edit(int id)
    {
        var movieToEdit = (from m in _db.MovieSet
                           where m.Id == id
                           select m).First();
        return View(movieToEdit);
    }
    //
    // POST: /Home/Edit/5
    [AcceptVerbs(HttpVerbs.Post)]
    public ActionResult Edit(Movie movieToEdit)
    {
        var originalMovie = (from m in _db.MovieSet
                             where m.Id == movieToEdit.Id
                             select m).First();
        if (!ModelState.IsValid)
            return View(originalMovie);
            _db.ApplyPropertyChanges(originalMovie.EntityKey.EntitySetName, movieToEdit);
            _db.SaveChanges();
            return RedirectToAction("Index");
    }
```

En el listado 6, he agregado lógica adicional para dos sobrecargas del método Edit(). El primer método Edit() devuelve el registro de base de datos de la película que corresponde al Id parámetro pasado al método. La segunda sobrecarga realiza las actualizaciones a un registro de la película en la base de datos.
Observe que debe recuperar la película original y, a continuación, llame al ApplyPropertyChanges(), para actualizar la película existente en la base de datos.