

## MVC usando EntityFramework sin

**Nuget** => Microsoft.EntityFrameworkCore.SqlSever  
Microsoft.EntitiFrameworkCore.Tools

create database prueba

Use prueba

Create table idols(

Id int identity (1,1) not null primary key,

Nombre varchar(40),

Genero

Música

Discos

)

**Herramientas** =>Administrador de paquetes Nuget =>Consola del administrador de paquetes

```
Scaffold-DbContext "server=localhost; database=producto;  
Trusted_connection=true; TrustServerCertificate=true;"  
Microsoft.EntityFrameworkCore.SqlServer -OutPutDir Models
```

```
Scaffold-DbContext "server=localhost; database=prueba; integrated security =  
true;" Microsoft.EntityFrameworkCore.Sql -OutPutDir Models -Force
```

Appsettings.json=>

```
"ConnectionString":{
```

```
"conexión": "server=localhost; database=prueba; integrated security = true;"
```

```
}
```

Program =>

```
Using Microsoft.entityFrameworkCore;
```

Using NOMBREPROYECTO.Models

```
using Microsoft.EntityFrameworkCore;  
using PruebaEntity.Models;
```

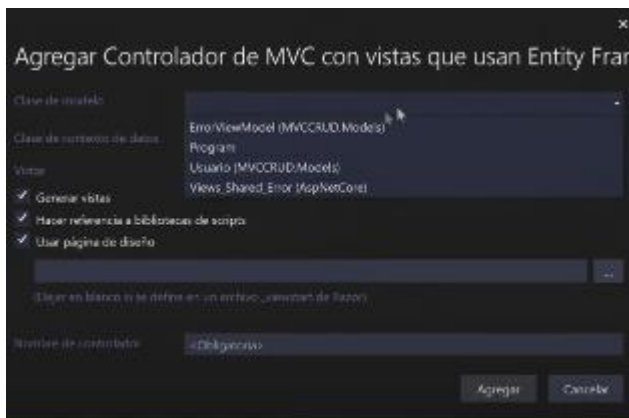
Builder..

```
builder.Services.AddDbContext<ProductoContext>(options =>  
options.UseSqlServer(builder.Configuration.GetConnectionString("conexion")));
```

clic derecho carpeta

Controllers=>

Agregar=>controlador=>Controlador de MVC con vistas que usa Entity Framework



por cada tabla añadir en clase y la de aabajo

View=>shared=>\_Layout.CShtml  
cambiar el

<div>

pagina nueva

</div>

Class="btn btn-warning"

Class="btn btn-danger"

classs="btn btn-primary"

OnClick = "return messeg("asdasd")"

<script>

</script>

MVC sqlClient

```
using Microsoft.AspNetCore.Mvc;
using System.Data;
using System.Data.SqlClient;
using Productos.Models;

namespace Productos.Controllers
{
    public class productoController : Controller
    {
        private string connectionString =
"Server=localhost;Database=producto;Integrated Security=true;";

        // Método para mostrar la lista de usuarios
        public ActionResult Index()
        {
            DataTable dtProductos = new DataTable();
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                con.Open();
                SqlDataAdapter adapter = new SqlDataAdapter("sp_listar", con);
                adapter.SelectCommand.CommandType =
CommandType.StoredProcedure;
                adapter.Fill(dtProductos);
            }
            return View(dtProductos);
        }

        // Método para crear un nuevo usuario
        public ActionResult Crear()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Crear(ProductosModel producto)
        {
            if(!ModelState.IsValid)
                return View();
            try
            {
                using (SqlConnection con = new SqlConnection(connectionString))
                {
                    con.Open();
```

```

        SqlCommand cmd = new SqlCommand("sp_guardar", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Nombre", producto.nombre);
        cmd.Parameters.AddWithValue("@Categoria",
producto.categoria);
        cmd.Parameters.AddWithValue("@Precio", producto.precio);
        cmd.Parameters.AddWithValue("@Fecha", producto.fecha);
        cmd.ExecuteNonQuery();
    }

    }
    catch (Exception ex)
    {

    }
    return RedirectToAction("Index");
}

// Método para editar un usuario
public ActionResult Editar(int id)
{
    ProductosModel producto = new ProductosModel();
    using (SqlConnection con = new SqlConnection(connectionString))
    {
        con.Open();
        SqlDataAdapter adapter = new SqlDataAdapter("sp_obtener", con);
        adapter.SelectCommand.CommandType =
CommandType.StoredProcedure;
        adapter.SelectCommand.Parameters.AddWithValue("@Id", id);
        DataTable dt = new DataTable();
        adapter.Fill(dt);
        if (dt.Rows.Count == 1)
        {
            producto.id = Convert.ToInt32(dt.Rows[0]["Id"]);
            producto.nombre = dt.Rows[0]["nombre"].ToString();
            producto.categoria = dt.Rows[0]["categoria"].ToString();
            producto.precio = Convert.ToInt32(dt.Rows[0]["precio"]);
            producto.fecha = Convert.ToDateTime(dt.Rows[0]["fecha"]);
            return View(producto);
        }
        else
        {
            return RedirectToAction("Index");
        }
    }
}

[HttpPost]
public ActionResult Editar(ProductosModel producto)
{
    if (!ModelState.IsValid)
        return View();
    using (SqlConnection con = new SqlConnection(connectionString))
    {
        con.Open();

```

```

        SqlCommand cmd = new SqlCommand("sp_editar", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Id", producto.id);
        cmd.Parameters.AddWithValue("@Nombre", producto.nombre);
        cmd.Parameters.AddWithValue("@Categoria", producto.categoria);
        cmd.Parameters.AddWithValue("@Precio", producto.precio);
        cmd.Parameters.AddWithValue("@Fecha", producto.fecha);
        cmd.ExecuteNonQuery();
    }
    return RedirectToAction("Index");
}

// Método para eliminar un usuario
public ActionResult Eliminar(int id)
{
    using (SqlConnection con = new SqlConnection(connectionString))
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("sp_eliminar", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Id", id);
        cmd.ExecuteNonQuery();
    }
    return RedirectToAction("Index");
}
}

INSERT INTO dbo.DLC Usuarios (NombreUsuario, Apellidos, FechaNacimiento, FechaAlta)
SELECT @NombreUsuario, @Apellidos, @FechaNacimiento, GETDATE()

DECLARE @IdUsuario AS INT = (SELECT MAX(IdUsuario) FROM dbo.DLC Usuarios)
DECLARE @IdGrupo AS INT = (SELECT IdGrupo FROM [dbo].DLC GruposUsuarios WHERE NombreGrupo=
NombreGrupo)

INSERT INTO [dbo].[DLC Usuarios Grupos] (IdUsuario, IdGrupo, FechaAlta, CodBaja)
SELECT @IdUsuario, @IdGrupo, GETDATE(), 0

I

--SELECT *|

```

## Index

@using System.Data

@model System.Data.DataTable

[Crear Nuevo Producto](@Url.Action()

```

<table class="table">
    <thead>
        <tr>
            <th>Id</th>

```

```

        <th>Nombre</th>
        <th>Apellido</th>
        <th>Correo</th>
        <th>Acciones</th>
    </tr>
</thead>
<tbody>
    @foreach (DataRow row in Model.Rows)
    {
        <tr>
            <td>@row["Id"]</td>
            <td>@row["Nombre"]</td>
            <td>@row["Categoria"]</td>
            <td>@row["Precio"]</td>
            <td>@row["Fecha"]</td>
            <td>
                <a href="@Url.Action("editar", "producto", new { id =
row["Id"] } )" class="btn btn-warning">Editar</a>
                <a href="@Url.Action("eliminar", "producto", new { id =
row["id"] } )" class="btn btn-danger" onclick="return
confirmarEliminacion();">Eliminar</a>
            </td>
        </tr>
    }
</tbody>
</table>
<script>
    // Función para confirmar la eliminación
    function confirmarEliminacion() {
        return confirm("¿Está seguro de que desea eliminar este producto?");
    }
</script>

```

## Crear

```

@model ProductosModel

@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)

    <div class="form-group">
        @Html.LabelFor(model => model.nombre)
        @Html.TextBoxFor(model => model.nombre, new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.nombre)
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.categoria)
        @Html.TextBoxFor(model => model.categoria, new { @class = "form-control" })
    </div>

```

```

        @Html.ValidationMessageFor(model => model.categoria)

    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.precio)
        @Html.TextBoxFor(model => model.precio, new { @class = "form-control" })
    }
    @Html.ValidationMessageFor(model => model.precio)
</div>

<div class="form-group">
    @Html.LabelFor(model => model.fecha)
    @Html.TextBoxFor(model => model.fecha, "{0:yyyy-MM-dd}", new { @class =
"form-control", type = "date" })
    @Html.ValidationMessageFor(model => model.fecha)
</div>

<button type="submit" class="btn btn-primary">Guardar</button>
}

```

Editar

@model ProductosModel

```

@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)

    @Html.HiddenFor(model => model.id)

    <div class="form-group">
        @Html.LabelFor(model => model.nombre)
        @Html.TextBoxFor(model => model.nombre, new { @class = "form-control" })
    }
    @Html.ValidationMessageFor(model => model.nombre)
</div>

<div class="form-group">
    @Html.LabelFor(model => model.categoria)
    @Html.TextBoxFor(model => model.categoria, new { @class = "form-
control" })
    @Html.ValidationMessageFor(model => model.categoria)
</div>

<div class="form-group">
    @Html.LabelFor(model => model.precio)
    @Html.TextBoxFor(model => model.precio, new { @class = "form-control" })
}
    @Html.ValidationMessageFor(model => model.precio)
</div>

<div class="form-group">
    @Html.LabelFor(model => model.fecha)
    @Html.TextBoxFor(model => model.fecha, "{0:yyyy-MM-dd}", new { @class =
"form-control", type = "date" })
    @Html.ValidationMessageFor(model => model.fecha)
}

```

```
        </div>

        <button type="submit" class="btn btn-primary">Guardar Cambios</button>
    }
}
```

Eliminar

@model int

```
@using (Html.BeginForm())
{
```

```
    <h2>Eliminar Usuario</h2>

    <p>¿Estás seguro de que quieres eliminar este usuario?</p>
    @Html.HiddenFor(model => model)

    <button type="submit" class="btn btn-danger">Eliminar</button>
}
}
```