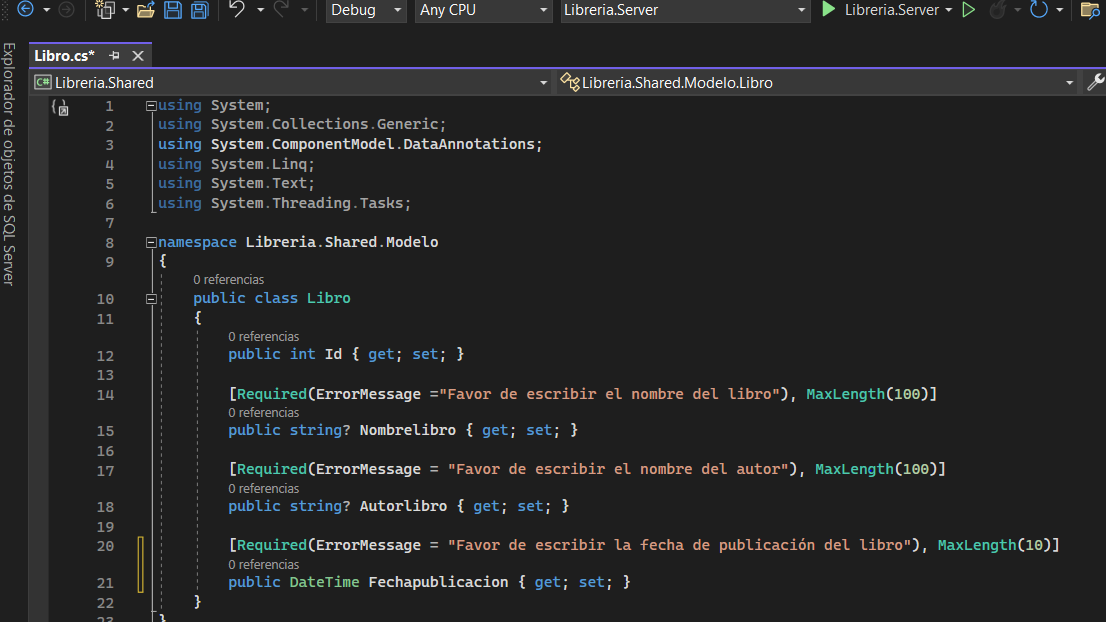
Crear un folder y después una clase

@ Libreria. Shared 
Dependencias 
Modelo 
ca Libro.cs 
WeatherForecast.cs 

Llenar los datos de mi clase



public class Libro

{

public int Id { get; set; }

[Required(ErrorMessage ="Favor de escribir el nombre del libro"), MaxLength(100)]

public string? Nombrelibro { get; set; }

[Required(ErrorMessage = "Favor de escribir el nombre del autor"), MaxLength(100)]

public string? Autorlibro { get; set; }

[Required(ErrorMessage = "Favor de escribir la fecha de publicación del libro"), MaxLength(10)]

public string? Fechapublicacion { get; set; }

}

Crear mi carpeta DATA en la parte de servidor

Program.cs 
LibreriaServer 
c. Connected Services 
•8 Dependencias 
Properties 
Controllers 
Data 
Pages 
aPpsettingsjson 
cz Program.cs 
Libreria_Shared 

Descargo los paquetes de tools y sqlserver

Librería.Server - Administrar paquetes NuGet

> ca Program.cs 
Connected Servi ces 
Dependencias 
Analizadores 
Marcos de trabajo 
•e Paquetes 
Microsoft.AspNetCore.Components.WebAssembly. 
Microsoft.EntityFrameworkCore.SqlSetver (7.0.5) 
MicrosoftEntityFrameworkCore.Tools (705) 

Después hago un clase llamada contexto

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Lleno de datos el LibroContexto

Captura de pantalla de computadora

Descripción generada automáticamente

public class LibroContexto:DbContext

{

public LibroContexto(DbContextOptions<LibroContexto> options) : base(options)

{

}

public DbSet<Libro> Libros { get; set; }

}

Me meto a program.cs para levantar el servidor

Texto

Descripción generada automáticamente

Y se agrega esto

Texto

Descripción generada automáticamente

builder.Services.AddDbContext<LibroContexto>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

Abrimos

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Texto

Descripción generada automáticamente

"ConnectionStrings": {

"DefaultConnection": ""

}

Antes de escribir ahí nos metemos a explorador

Texto

Descripción generada automáticamente

Doy click derecho en la primer bd, propiedades y copio todo el link

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Y lo pongo ahí entre comillas

Captura de pantalla de un celular

Descripción generada automáticamente

PERO ese master lo cambio por Lbros

Interfaz de usuario gráfica, Sitio web

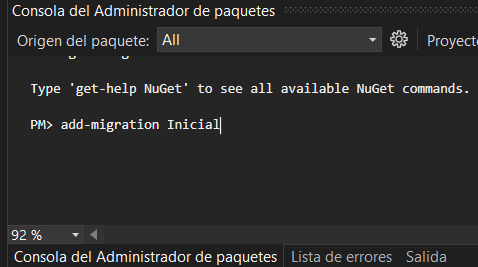
Descripción generada automáticamente con confianza media Texto

Descripción generada automáticamente

Luego

Herramientas – Administrador de paquetes NuGet – Consola adminsitrador…

Escribirmos esto en la consola

 Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Crear el controlador

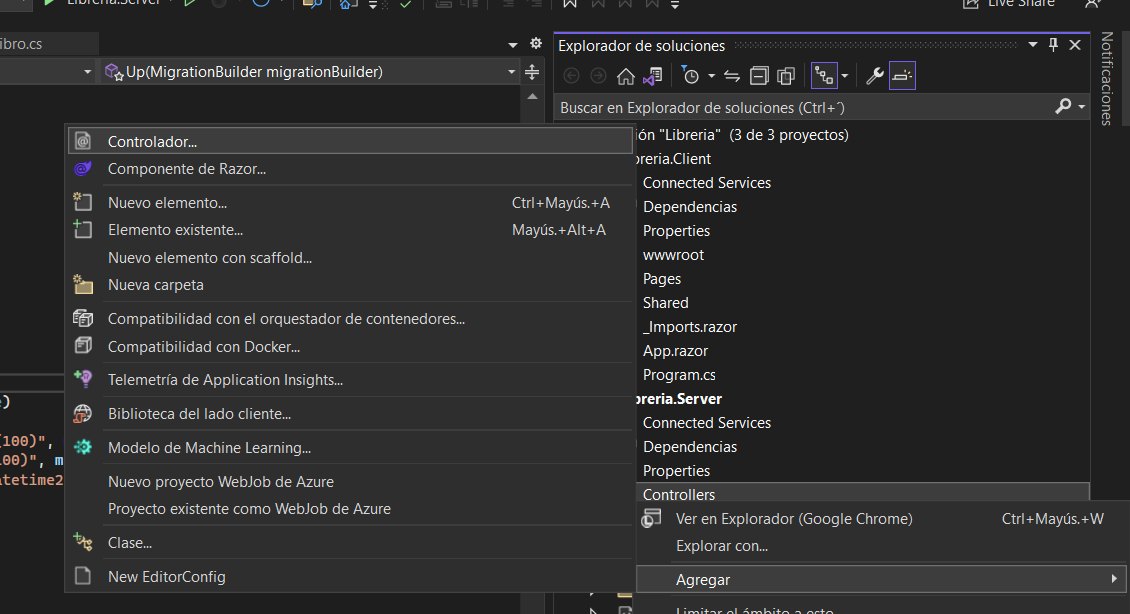


Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza media

Se seleccionan esos pero en vez de persona es libro

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

después hago una carpeta donde meteré mis páginas

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Hago la primera página

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

ENLAZARLO EN NAVMENU

Captura de pantalla de computadora

Descripción generada automáticamente

INDEX

@page "/libros"

@using Libreria.Shared.Modelo;

@inject HttpClient http

@inject NavigationManager navegar

<h3>Formulario</h3>

<button class="btn btn-primary" @onclick="Agregar"><span class="oi oi-plus"></span>Agregar libro</button>

@if (losLibros == null)

{

<p>Cragando datos...</p>

}

else if (losLibros.Count == 0)

{

<p>No existen personas registradas</p>

}

else

{

<table class="table">

<thead>

<tr>

<th>Libro</th>

<th>Autor</th>

<th>Fecha</th>

<th>Acciones</th>

</tr>

</thead>

<tbody>

@foreach (var elLibro in losLibros)

{

<tr>

<td>@elLibro.Nombrelibro</td>

<td>@elLibro.Autorlibro</td>

<td>@elLibro.Fechapublicacion</td>

<td>

<button class="btn btn-danger" @onclick="()=>Modificar(elLibro.Id)">Modificar</button>

<button class="btn btn-warning" @onclick="()=>Eliminar(elLibro)">Eliminar</button>

</td>

</tr>

}

</tbody>

</table>

}

@code {

private List<Libro> losLibros;

protected override async Task OnInitializedAsync()

{

losLibros = await http.GetFromJsonAsync<List<Libro>>("api/libroes");

}

private void Agregar()

{

navegar.NavigateTo("/libros/nueva");

}

private void Modificar(int id)

{

navegar.NavigateTo($"/libros/modificar/{id}");

}

private async Task Eliminar(Libro unLibro)

{

await http.DeleteAsync($"/libros/modificar/{unLibro.Id}");

losLibros.Remove(unLibro);

}

}

MODIFICAR

@page "/libros/modificar/{Id:int}"

@using Libreria.Shared.Modelo;

@inject HttpClient http

@inject NavigationManager navegar

<h3>Modificar</h3>

<**EditForm** **Model**="elLibro" **OnValidSubmit**="Guardar">

<**DataAnnotationsValidator** />

<div class="row mb-3">

<label for="txtLibro" class="col-2 col-form-label">NOMBRE DEL LIBRO</label>

<**InputText** class="form-control" id="txtLibro" @bind-Value="elLibro.Nombrelibro" />

<**ValidationMessage** **For**="(()=>elLibro.Nombrelibro)" />

</div>

<div class="row mb-3">

<label for="txtAutor" class="col-2 col-form-label">AUTOR DEL LIBRO</label>

<**InputText** class="form-control" id="txtAutor" @bind-Value="elLibro.Autorlibro" />

<**ValidationMessage** **For**="(()=>elLibro.Autorlibro)" />

</div>

<div class="row mb-3">

<label for="txtFecha" class="col-2 col-form-label">FECHA DE PUBLICACIÓN DEL LIBRO</label>

<**InputText** class="form-control" id="txtFecha" @bind-Value="elLibro.Fechapublicacion" />

<**ValidationMessage** **For**="(()=>elLibro.Fechapublicacion)" />

</div>

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

</**EditForm**>

@code {

[Parameter]

public int Id { get; set; }

private Libro elLibro = new Libro();

protected override async Task OnInitializedAsync()

{

elLibro = await http.GetFromJsonAsync<Libro>($"api/libroes/{Id}");

}

private async void Guardar()

{

await http.PutAsJsonAsync($"api/libroes/{Id}", elLibro);

navegar.NavigateTo("/libros");

}

}

NUEVA

@page "/libros/nueva"

@using Libreria.Shared.Modelo;

@inject HttpClient http

@inject NavigationManager navegar

<h3>Libro nuevo</h3>

<**EditForm** **Model**="elLibro" **OnValidSubmit**="Guardar">

<**DataAnnotationsValidator** />

<div class="row mb-3">

<label for="txtLibro" class="col-2 col-form-label">NOMBRE DEL LIBRO</label>

<**InputText** class="form-control" id="txtLibro" @bind-Value="elLibro.Nombrelibro" />

<**ValidationMessage** **For**="(()=>elLibro.Nombrelibro)" />

</div>

<div class="row mb-3">

<label for="txtAutor" class="col-2 col-form-label">AUTOR DEL LIBRO</label>

<**InputText** class="form-control" id="txtAutor" @bind-Value="elLibro.Autorlibro" />

<**ValidationMessage** **For**="(()=>elLibro.Autorlibro)" />

</div>

<div class="row mb-3">

<label for="txtFecha" class="col-2 col-form-label">FECHA</label>

<**InputText** class="form-control" id="txtFecha" @bind-Value="elLibro.Fechapublicacion" />

<**ValidationMessage** **For**="(()=>elLibro.Fechapublicacion)" />

</div>

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

</**EditForm**>

@code {

private Libro elLibro = new Libro();

private async void Guardar()

{

await http.PostAsJsonAsync("api/libroes", elLibro);

navegar.NavigateTo("/libros");

}

}

@page "/prestamos/modificar/{Id:int}"

@using Libreria.Shared.Modelo;

@inject HttpClient http

@inject NavigationManager navegar

<h3>Modificar</h3>

<**EditForm** **Model**="elPrestamo" **OnValidSubmit**="Guardar">

<**DataAnnotationsValidator** />

<div class="row mb-3">

<label for="txtInicio" class="col-2 col-form-label">FECHA DE INICIO DE PRESTAMO</label>

<**InputDate** class="form-control" id="txtInicio" @bind-Value="elPrestamo.Inicioprestamo" />

<**ValidationMessage** **For**="(()=>elPrestamo.Inicioprestamo)" />

</div>

<div class="row mb-3">

<label for="txtVencimiento" class="col-2 col-form-label">FECHA DE VENCIMIENTO DE PRESTAMO</label>

<**InputDate** class="form-control" id="txtVencimiento" @bind-Value="elPrestamo.Vencimientoprestamo" />

<**ValidationMessage** **For**="(()=>elPrestamo.Vencimientoprestamo)" />

</div>

<div class="row mb-3">

<label for="txtDevolucion" class="col-2 col-form-label">FECHA DE VENCIMIENTO DEL PRESTAMO</label>

<**InputDate** class="form-control" id="txtDevolucion" @bind-Value="elPrestamo.Fechadevolucion" />

<**ValidationMessage** **For**="(()=>elPrestamo.Fechadevolucion)" />

</div>

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

</**EditForm**>

@code {

[Parameter]

public int Id { get; set; }

private Prestamos elPrestamo = new Prestamos();

protected override async Task OnInitializedAsync()

{

elPrestamo = await http.GetFromJsonAsync<Prestamos>($"api/prestamos/{Id}");

}

private async void Guardar()

{

await http.PutAsJsonAsync($"api/prestamos/{Id}", elPrestamo);

navegar.NavigateTo("/prestamos");

}

}