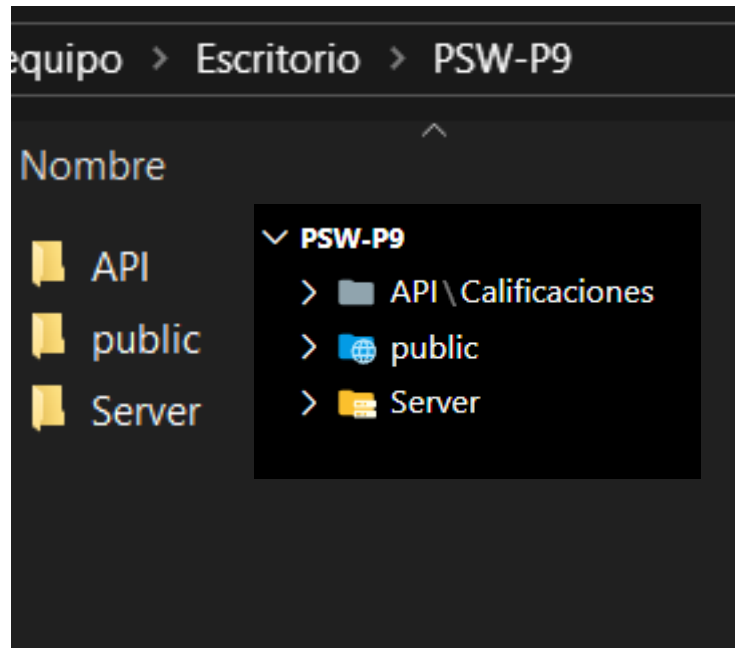
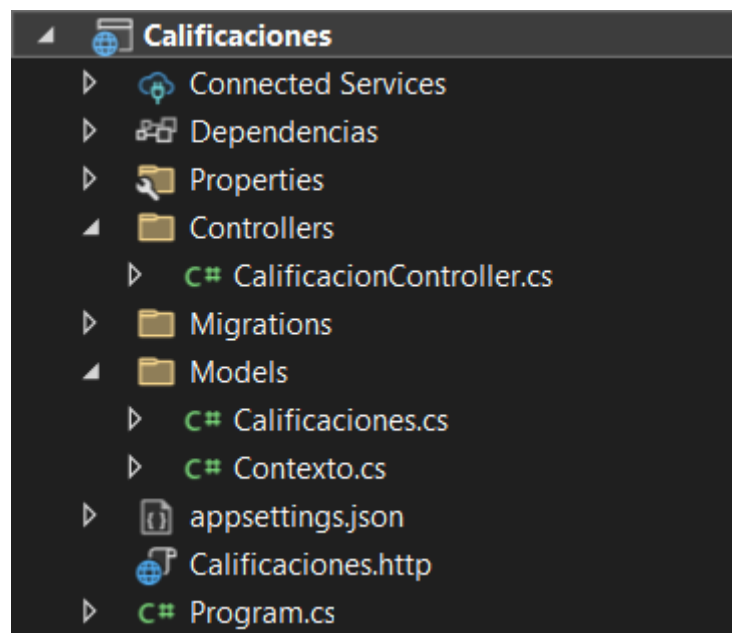


Estructura de mi proyecto

0.Raíz



1.API



1.1 Calificaciones.cs

```
Calificaciones.cs  X
Calificaciones  Calificaciones.Models.Calificacion  nombre

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using static System.Net.Mime.MediaTypeNames;
using static System.Runtime.InteropServices.JavaScript.JSType;

namespace Calificaciones.Models
{
    [Table("TablaCalificaciones")]

    public class Calificacion
    {
        [Key]

        public int id { get; set; }

        [MaxLength(50)]
        [Required]
        0 referencias
        public string nombre { get; set; }

        [MaxLength(50)]
        [Required]
        0 referencias
        public string materia { get; set; }

        0 referencias
        public double calificacion { get; set; }
    }
}
```

1.2 Contexto.cs

```
Contexto.cs  X  Calificaciones.cs
Calificaciones  Calificaciones.Models.Contexto  TablaCalificaciones

using Microsoft.EntityFrameworkCore;

namespace Calificaciones.Models
{
    9 referencias
    public class Contexto : DbContext
    {
        6 referencias
        public DbSet<Calificacion> TablaCalificaciones { get; set; }

        0 referencias
        public Contexto()
        {
        }

        0 referencias
        protected override void OnConfiguring(DbContextOptionsBuilder options)
            => options.UseSqlite($"Data Source=d:\\Datos\\Escritorio\\PSW-P9\\Server\\TablaCalificaciones.db");
    }
}
```

1.3 CalificacionController

Controlador generado por entity framework (genérico)

```
CalificacionController.cs Contexto.cs Calificaciones.cs
Calificaciones
    Calificaciones.Controllers.CalificacionContrc GetCalificacion(int id)
{
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Threading.Tasks;
    using Microsoft.AspNetCore.Http;
    using Microsoft.AspNetCore.Mvc;
    using Microsoft.EntityFrameworkCore;
    using Calificaciones.Models;

    namespace Calificaciones.Controllers
    {
        [Route("api/[controller]")]
        [ApiController]
        public class CalificacionController : ControllerBase
        {
            private readonly Contexto _context;

            0 referencias
            public CalificacionController(Contexto context)
            {
                _context = context;
            }

            // GET: api/Calificacion
            [HttpGet]
            0 referencias
            public async Task<ActionResult<IEnumerable<Calificacion>>> GetTablaCalificaciones()
            {
                return await _context.TablaCalificaciones.ToListAsync();
            }
        }
    }
}
```

1.4 Program.cs

La diferencia fue habilitar CORS para poderse comunicar con el frontend

```
// Agrega y configura la política de CORS
builder.Services.AddCors(options =>
{
    options.AddPolicy("AllowAll", policyBuilder =>
    {
        policyBuilder.AllowAnyOrigin()
            .AllowAnyMethod()
            .AllowAnyHeader();
    });
});
```

2.Server

Nombre	Tipo	Esquema
Tablas (3)		
TablaCalificaciones		CREATE TABLE "TablaCalificaciones" ("id" INTEGER NOT NULL CONSTRAINT "PK_TablaCalificaciones"
id	INTEGER	"id" INTEGER NOT NULL
calificacion	REAL	"calificacion" REAL NOT NULL
materia	TEXT	"materia" TEXT NOT NULL
nombre	TEXT	"nombre" TEXT NOT NULL
> __EFMigrationsHistory		
CREATE TABLE "__EFMigrationsHistory" ("MigrationId" TEXT NOT NULL CONSTRAINT "PK__EFMigr		
> sqlite_sequence		
CREATE TABLE sqlite_sequence(name,seq)		
Índices (0)		
Vistas (0)		
Disparadores (0)		

Server

ArchivoInicioCompartirVista

Anclar al acceso rápidoCopiarPegarPortapapeles

Mover aEliminarCopiar aCambiar nombreOrganizar

Nueva carpetaPropiedadesNuevoAbrir

Seleccionar todoNo seleccionar ningunoInvertir selecciónSeleccionar

Este equipoEscritorioPSW-P9Server

Buscar en Server

Nombre	Fecha de modificación	Tipo	Tamaño
TablaCalificaciones.db	25/10/2024 0:57	Data Base File	24 Ki
TablaCalificaciones.db-shm	25/10/2024 1:05	Archivo DB-SHM	32 Ki
TablaCalificaciones.db-wal	25/10/2024 1:26	Archivo DB-WAL	81 Ki

3 elementos

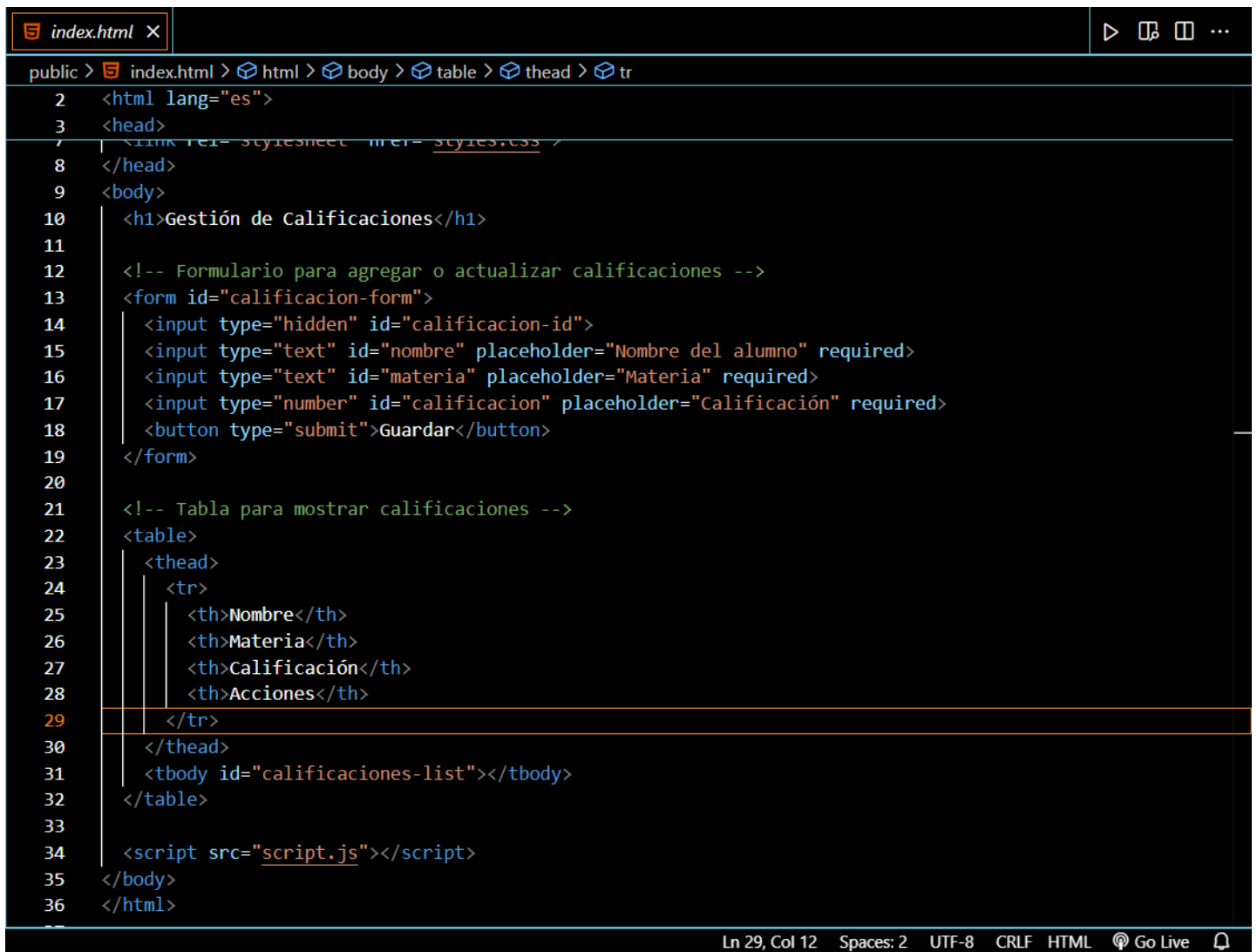
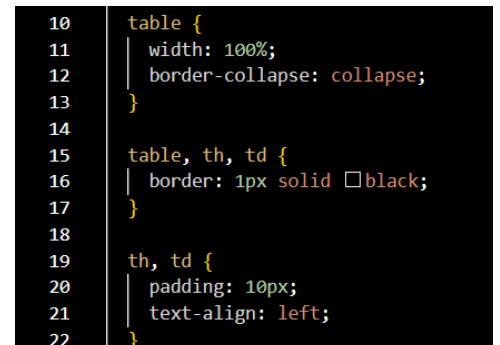
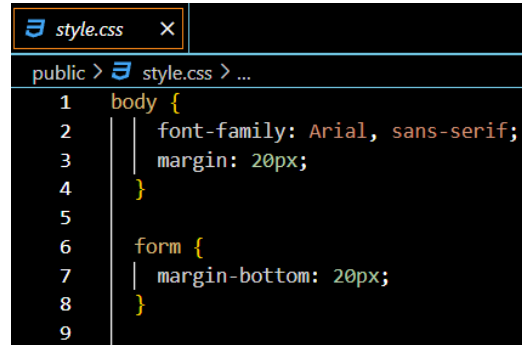
Tabla:

TablaCalificaciones

	id	calificacion	materia	nombre
	Filtro	Filtro	Filtro	Filtro
1	6	10.0	Ingles	Noe

	id	calificacion	materia	nombre
	Filtro	Filtro	Filtro	Filtro
1	6	10.0	Ingles	Noe

3. Public



```
script.js X
public > script.js > obtenerCalificaciones > data.forEach() callback
1 const form = document.getElementById('calificacion-form');
2 const calificacionesList = document.getElementById('calificaciones-list');
3 const apiUrl = 'http://localhost:5161/api/calificacion';
4
5 // Obtener todas las calificaciones al cargar la página
6 async function obtenerCalificaciones() {
7   try {
8     const res = await fetch(apiUrl);
9     if (!res.ok) {
10      throw new Error(`Error: ${res.status} ${res.statusText}`);
11    }
12    const data = await res.json();
13
14    calificacionesList.innerHTML = '';
15    data.forEach(calificacion => {
16      const row = document.createElement('tr');
17      row.innerHTML = `
18        <td>${calificacion.nombre}</td>
19        <td>${calificacion.materia}</td>
20        <td>${calificacion.calificacion}</td>
21        <td>
22          <button onclick="editarCalificacion(${calificacion.id}, '${calificacion.nombre}',
23            '${calificacion.materia}', ${calificacion.calificacion})">Editar</button>
24          <button onclick="eliminarCalificacion(${calificacion.id})">Eliminar</button>
25        </td>
26      `;
27      calificacionesList.appendChild(row);
28    });
29  } catch (error) {
30    console.error("Error al obtener las calificaciones:", error);
31    alert("No se pudieron cargar las calificaciones. Verifica la conexión con la API.");
32  }
33
34 // Crear o actualizar una calificación
35 form.addEventListener('submit', async (e) => {
36   e.preventDefault();
37
38   const id = document.getElementById('calificacion-id').value;
39   const nombre = document.getElementById('nombre').value;
40   const materia = document.getElementById('materia').value;
41   const calificacion = parseFloat(document.getElementById('calificacion').value);
42
43   const method = id ? 'PUT' : 'POST';
44   const url = id ? `${apiUrl}/${id}` : apiUrl;
45   const body = id
46     ? { id: parseInt(id), nombre, materia, calificacion } // Para actualizar incluye el ID
47     : { nombre, materia, calificacion }; // Para crear, no incluye el ID
48
49   try {
50     const res = await fetch(url, {
51       method,
52       headers: { 'Content-Type': 'application/json' },
53       body: JSON.stringify(body)
54     });
55   }
```

En pocas palabras,
consumimos nuestra API
a partir de su despliegue
local(url) utilizando el
recurso "fetch"

JS script.js X

public > JS script.js > form.addEventListener('submit') callback

```
36 form.addEventListener('submit', async (e) => {
55   });
56
57   if (res.ok || res.status === 204) { // Maneja códigos 200 y 204 como éxito
58     form.reset();
59     obtenerCalificaciones();
60   } else {
61     throw new Error(`Error: ${res.status} ${res.statusText}`);
62   }
63 } catch (error) {
64   console.error("Error al guardar la calificación:", error);
65   alert("No se pudo guardar la calificación. Verifica la conexión con la API.");
66 }
67 });
68
69 // Editar una calificación
70 function editarCalificacion(id, nombre, materia, calificacion) {
71   document.getElementById('calificacion-id').value = id;
72   document.getElementById('nombre').value = nombre;
73   document.getElementById('materia').value = materia;
74   document.getElementById('calificacion').value = calificacion;
75 }
76
77 // Eliminar una calificación
78 async function eliminarCalificacion(id) {
79   try {
80     const res = await fetch(`${apiUrl}/${id}`, { method: 'DELETE' });
81     if (!res.ok) {
82       throw new Error(`Error: ${res.status} ${res.statusText}`);
83     }
84     obtenerCalificaciones();
85   } catch (error) {
86     console.error("Error al eliminar la calificación:", error);
87     alert("No se pudo eliminar la calificación. Verifica la conexión con la API.");
88   }
89 }
90
91 // Cargar calificaciones al inicio
92 obtenerCalificaciones();
93
```

4. Pruebas

- Vacío

The screenshot shows two browser windows. The left window, titled 'Gestión de Calificaciones', has a form with three input fields: 'Nombre del alumno' (containing 'Fidel'), 'Materia' (containing 'Ingles'), and 'Calificación' (containing '10'). Below the fields is a 'Guardar' button. The right window, titled 'Swagger UI', shows a GET request to the endpoint '/api/Calificacion'. The 'Parameters' section is empty. The 'Responses' section shows a 200 status code with an empty response body '[]'.

- Crear

This screenshot is identical to the one above, showing the 'Gestión de Calificaciones' application with the 'Fidel', 'Ingles', and '10' data, and the Swagger UI showing a successful GET request to '/api/Calificacion' with an empty response body.

- Creado

Gestión de Calificaciones

Nombre del alumno: Materia: Calificación:

Nombre	Materia	Calificación	Acciones
Fidel	Ingles	10	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Swagger UI

GET /api/Calificacion

Parameters: No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5161/api/Calificacion' \
  -H 'accept: text/plain'
```

Request URL

http://localhost:5161/api/Calificacion

Server response

Code: 200

Details

Response body

```
{
  "id": 7,
  "nombre": "Fidel",
  "materia": "Ingles",
  "calificacion": 10
}
```

Response headers

- Editar

Gestión de Calificaciones

Nombre del alumno: Materia: Calificación:

Nombre	Materia	Calificación	Acciones
Fidel	Ingles	10	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Swagger UI

GET /api/Calificacion

Parameters: No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5161/api/Calificacion' \
  -H 'accept: text/plain'
```

Request URL

http://localhost:5161/api/Calificacion

Server response

Code: 200

Details

Response body

```
{
  "id": 7,
  "nombre": "Fidel",
  "materia": "Ingles",
  "calificacion": 10
}
```

Response headers

- Los datos se llenaron solos

The screenshot shows two browser windows. The left window, titled 'Gestión de Calificaciones', displays a form with input fields for 'Nombre' (Fidel), 'Materia' (Ingles), and 'Calificación' (10). Below the form is a 'Guardar' button. A table below the form shows the saved data: Fidel, Ingles, 10, with 'Editar' and 'Eliminar' buttons. The right window shows the Swagger UI for the endpoint `/api/Calificacion`. The 'Parameters' section is empty. The 'Responses' section shows a 200 status code with a JSON response body: `{ "id": 7, "nombre": "Fidel", "materia": "Ingles", "calificacion": 10 }`.

- Modificar y guardar

The screenshot shows the same two browser windows. In the 'Gestión de Calificaciones' application, the 'Nombre' field now contains 'Noe' instead of 'Fidel'. The 'Guardar' button is highlighted with a mouse cursor. The table below still shows the previous entry: Fidel, Ingles, 10. The Swagger UI on the right remains the same, showing the 200 response with the JSON body for 'Fidel'.

- Datos actualizados

Gestión de Calificaciones

Nombre del alumno: Materia: Calificación:

Nombre	Materia	Calificación	Acciones
Noe	Inglés	10	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Swagger UI

GET /api/Calificacion

Parameters: No parameters

Execute Clear

Responses

Curl:

```
curl -X 'GET' \
'http://localhost:5161/api/Calificacion' \
-H 'accept: text/plain'
```

Request URL:

```
http://localhost:5161/api/Calificacion
```

Server response

Code: 200

Details: Response body:

```
{
  "id": 7,
  "nombre": "Noe",
  "materia": "Inglés",
  "calificacion": 10
}
```

- Eliminar

Gestión de Calificaciones

Nombre del alumno: Materia: Calificación:

Nombre	Materia	Calificación	Acciones
Noe	Inglés	10	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Swagger UI

GET /api/Calificacion

Parameters: No parameters

Execute Clear

Responses

Curl:

```
curl -X 'GET' \
'http://localhost:5161/api/Calificacion' \
-H 'accept: text/plain'
```

Request URL:

```
http://localhost:5161/api/Calificacion
```

Server response

Code: 200

Details: Response body:

```
{
  "id": 7,
  "nombre": "Noe",
  "materia": "Inglés",
  "calificacion": 10
}
```

- Eliminado

Gestión de Calificaciones

Nombre	Materia	Calificación	Acciones
--------	---------	--------------	----------

Swagger UI

GET /api/Calificacion

No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5161/api/Calificacion' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5161/api/Calificacion
```

Server response

Code	Details
200	<div><div>Response body</div><pre>[]</pre><div><input type="button" value="Download"/></div></div> <div><div>Response headers</div><pre>content-type: application/json; charset=utf-8 date: Fri, 25 Oct 2024 09:20:12 GMT</pre></div>