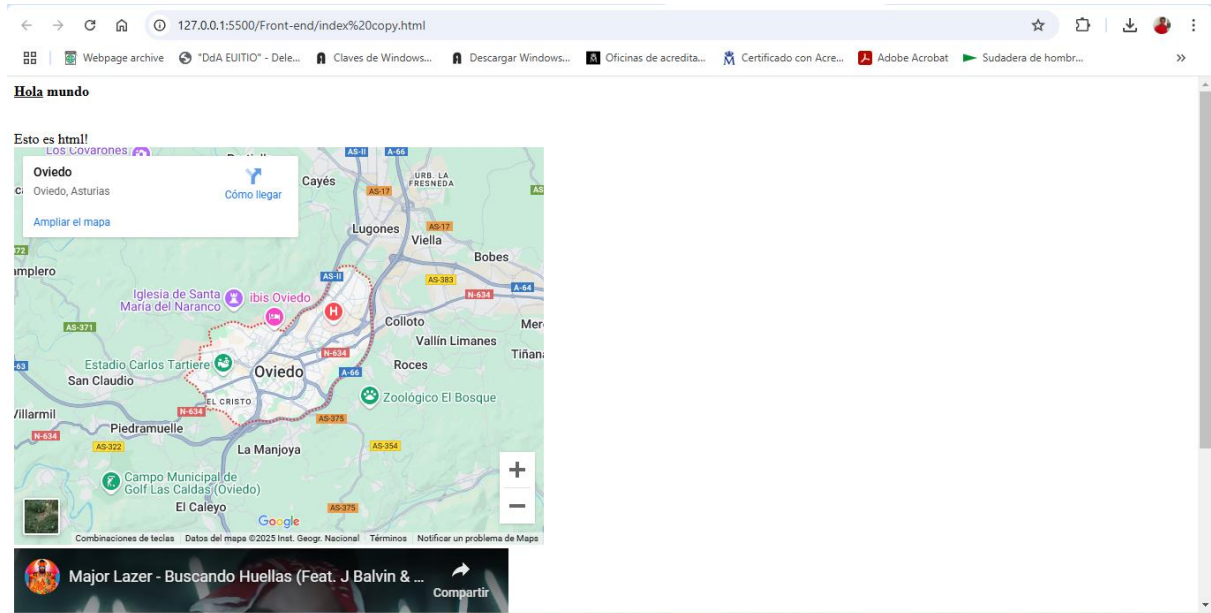


.NET y SWAGGER (2022)

Servidor ¿Who is? _____	2
Introducción Front (HTML) _____	2
Tablas _____	3
Botones _____	3
Menú y elemento modal _____	6
Servicios Web _____	9
SOAP _____	9
REST _____	10
Estructura de una URL _____	10
Métodos HTTP _____	11
JSON y XML _____	11
MVC y REST _____	12
MVC _____	12
REST _____	13
Creación de un API Rest en .Net _____	13
Pasos _____	13
Validaciones _____	17
Status code _____	17
Xampp _____	19
Conexión a la BBDD _____	20
Herramienta para probar nuestra API _____	22
Herramienta Documentación _____	22
Solución a la creación de la API _____	22
Conexión entre Back and Front-end _____	28
1. CORS (Cross-Origin Resource Sharing) _____	28
Cómo habilitar CORS en .NET _____	28
Creación de nuestro Script _____	30
Vista final del proyecto _____	35

Servidor ¿Who is?

Introducción Front (HTML)



Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>
  <b><u>Hola</u> mundo</b> <!-- b: negrita | u: subrayado -->
  <br>
  <br/> <!-- br: salto de línea -->
  <br/> <!-- br: salto de línea -->
  Esto es html!
  <br>
  <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d46405.72502209
565!2d-
5.889857618056057!3d43.36954185097312!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f
13.1!3m3!1m2!1s0xd368c9a60ac1c67%3A0x3134440ecc5e6224!2sOviedo%2C%20Astur
ias!5e0!3m2!1ses!2ses!4v1738657338169!5m2!1ses!2ses" width="600"
height="450" style="border:0;" allowfullscreen="" loading="lazy"
referrerpolicy="no-referrer-when-downgrade"></iframe>
```

```

    <br>
    <iframe width="560" height="315"
src="https://www.youtube.com/embed/pjg_09tDltY?si=GUTZ3j0L-aIAn8AQ"
title="YouTube video player" frameborder="0" allow="accelerometer;
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-
picture; web-share" referrerpolicy="strict-origin-when-cross-origin"
allowfullscreen></iframe>
</body>
</html>

```

Tablas

- Buscar en Google: style tables css tool
- Página de referencia (ejemplo): https://www.w3schools.com/css/css_table.asp

Botones

- Buscar en Google: style button tool
- Página de ejemplo: <https://www.bestcssbuttongenerator.com/#/13>

Pinchas en los primeros enlaces (infinidad de herramientas)

NOTA: Siempre que se trabaja en front-end, se usa nada más que HTML, CSS y JavaScript

Index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Index</title>
</head>

<body>
  <h1>Gestión de Clientes</h1>

  <table id="customers">
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Email</th>
      <th>Teléfono</th>
      <th>Acciones</th>
    </tr>
    <tr>
      <td>Germán</td>
      <td>Iglesias</td>

```

```

        <td>german@terra.es</td>
        <td>666999000</td>
        <td>
            <a href="#" class="myButton">Editar</a>
            <a href="#" class="myButtonDelete">Eliminar</a>
        </td>
    </tr>
    <tr>
        <td>José</td>
        <td>Mourinho</td>
        <td>jose@terra.es</td>
        <td>666111222</td>
        <td>
            <a href="#" class="myButton">Editar</a>
            <a href="#" class="myButtonDelete">Eliminar</a>
        </td>
    </tr>
</table>

<style>
    #customers {
        font-family: Arial, Helvetica, sans-serif;
        border-collapse: collapse;
        width: 100%;
    }

    #customers td,
    #customers th {
        border: 1px solid #ddd;
        padding: 8px;
    }

    #customers tr:nth-child(even) {
        background-color: #f2f2f2;
    }

    #customers tr:hover {
        background-color: #ddd;
    }

    #customers th {
        padding-top: 12px;
        padding-bottom: 12px;
        text-align: left;
        background-color: #04AA6D;
        color: white;
    }

    .myButton {

```

```

        box-shadow: 0px 1px 0px 0px #f0f7fa;
        background: linear-gradient(to bottom, #33bdef 5%, #019ad2
100%);
        background-color: #33bdef;
        border-radius: 6px;
        border: 1px solid #057fd0;
        display: inline-block;
        cursor: pointer;
        color: #ffffff;
        font-family: Arial;
        font-size: 15px;
        font-weight: bold;
        padding: 6px 24px;
        text-decoration: none;
        text-shadow: 0px -1px 0px #5b6178;
    }

    .myButton:hover {
        background: linear-gradient(to bottom, #019ad2 5%, #33bdef
100%);
        background-color: #019ad2;
    }

    .myButton:active {
        position: relative;
        top: 1px;
    }

    .myButtonDelete {
        box-shadow: 0px 1px 0px 0px #f0f7fa;
        background: linear-gradient(to bottom, #ed3434 5%, #d10214 100%);
        background-color: #ed3434;
        border-radius: 6px;
        border: 1px solid #057fd0;
        display: inline-block;
        cursor: pointer;
        color: #ffffff;
        font-family: Arial;
        font-size: 15px;
        font-weight: bold;
        padding: 6px 24px;
        text-decoration: none;
        text-shadow: 0px -1px 0px #5b6178;
    }

    .myButtonDelete:hover {
        background: linear-gradient(to bottom, #d10214 5%, #ed3434 100%);
        background-color: #d10214;
    }

    .myButtonDelete:active {

```

```

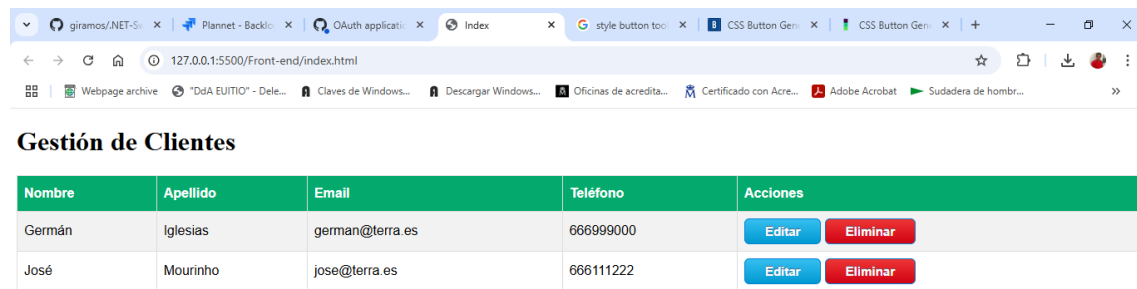
    position:relative;
    top:1px;
}

</style>
</body>

</html>

```

Página web



Nombre	Apellido	Email	Teléfono	Acciones
Germán	Iglesias	german@terra.es	666999000	Editar Eliminar
José	Mourinho	jose@terra.es	666111222	Editar Eliminar

NOTA: *Página de Referencia:* <https://codepen.io/>

Menú y elemento modal

Index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="style.css"> <!-- Enlaza el archivo
style.css -->
  <link rel="stylesheet" href="menu.css">
  <title>Index</title>
</head>
<nav id='menu'>
  <input type='checkbox' id='responsive-menu'
onclick='updatemenu()'><label></label>
  <ul>
    <li><a href='http://>Home</a></li>
    <li><a class='dropdown-arrow' href='http://>Products</a>
      <ul class='sub-menus'>
        <li><a href='http://>Products 1</a></li>
        <li><a href='http://>Products 2</a></li>
        <li><a href='http://>Products 3</a></li>
        <li><a href='http://>Products 4</a></li>
      </ul>
    </li>
  </ul>

```

```

    </li>
    <li><a href='http://'>About</a></li>
    <li><a class='dropdown-arrow' href='http://'>Services</a>
      <ul class='sub-menus'>
        <li><a href='http://'>Services 1</a></li>
        <li><a href='http://'>Services 2</a></li>
        <li><a href='http://'>Services 3</a></li>
      </ul>
    </li>
    <li><a href='http://'>Contact Us</a></li>
  </ul>
</nav>

<body>
  <h1>Gestión de Clientes</h1>
  <a onclick="agregarCliente()" href="#" class="myButton">Agregar
Cliente</a>
  <table id="customers">
    <tr>
      <th>Nombre</th>
      <th>Apellido</th>
      <th>Email</th>
      <th>Teléfono</th>
      <th>Acciones</th>
    </tr>
    <tr>
      <td>Germán</td>
      <td>Iglesias</td>
      <td>german@terra.es</td>
      <td>666999000</td>
      <td>
        <a href="#" class="myButton">Editar</a>
        <a href="#" class="myButtonDelete">Eliminar</a>
      </td>
    </tr>
    <tr>
      <td>José</td>
      <td>Mourinho</td>
      <td>jose@terra.es</td>
      <td>666111222</td>
      <td>
        <a href="#" class="myButton">Editar</a>
        <a href="#" class="myButtonDelete">Eliminar</a>
      </td>
    </tr>
  </table>
  <!-- Modal -->
  <div id="modal" class="modale" aria-hidden="true">
    <div class="modal-dialog">

```

```

        <div class="modal-header">
            <h2>Datos del Cliente</h2>
            <a onclick="cerrarCliente()" href="#" class="btn-close
closemodale" aria-hidden="true">&times;</a>
        </div>

        <div class="modal-body">
            <input type="text" name="u" placeholder="Nombre"
size="20" /><br/>
            <input type="text" name="u" placeholder="Apellido"
size="20" /><br/>
            <input type="text" name="u" placeholder="Teléfono"
size="20" /><br/>
            <input type="text" name="u" placeholder="Email" size="20"
/><br/>
        </div>

        <div class="modal-footer">

            <a href="#" class="btn" id="btn_ingresar">Guardar</a>
        </div>
    </div>
</div>
<!-- /Modal -->

</body>

<script>
    //alert(1);//Muestra un mensaje de alerta
    function agregarCliente() {
        //alert("Agregar Cliente");//Muestra un mensaje de alerta
        htmlModal = document.getElementById("modal"); //Obtiene el
elemento con el id modal
        htmlModal.setAttribute("class", "modale opened"); //Agrega la
clase open al elemento modal
    }

    function cerrarCliente() {
        //alert("Agregar Cliente");//Muestra un mensaje de alerta
        htmlModal = document.getElementById("modal"); //Obtiene el
elemento con el id modal
        htmlModal.setAttribute("class", "modale"); //Agrega la clase open
al elemento modal
    }
</script>

</html>

```


Vista.html

← → ↺ 🏠 🔍 127.0.0.1:5500/Front-end/index.html#

Webpage archive "DdA EUITIO" - Dele... Claves de Windows... Descargar Windows... Oficinas de acredita... Certificado con Acre... Adobe Acrobat Sudadera de hombr...

Home Products ▾ About Services ▾ Contact Us

Gestión de Clientes

Agregar Cliente

Nombre	Apellido	Email	Teléfono	Acciones
Germán	Iglesias	german@terra.es	666999000	<div>Editar Eliminar</div>
José	Mourinho	jose@terra.es	666111222	<div>Editar Eliminar</div>

← → ↺ 🏠 🔍 127.0.0.1:5500/Front-end/index.html#

Webpage archive "DdA EUITIO" - Dele... Claves de Windows... Descargar Windows... Oficinas de acredita... Certificado con Acre... Adobe Acrobat Sudadera de hombr...

Home Products ▾ About Services ▾ Contact Us

Gestión de Clientes

Agregar Cliente

Datos del Cliente

Nombre

Apellido

Teléfono

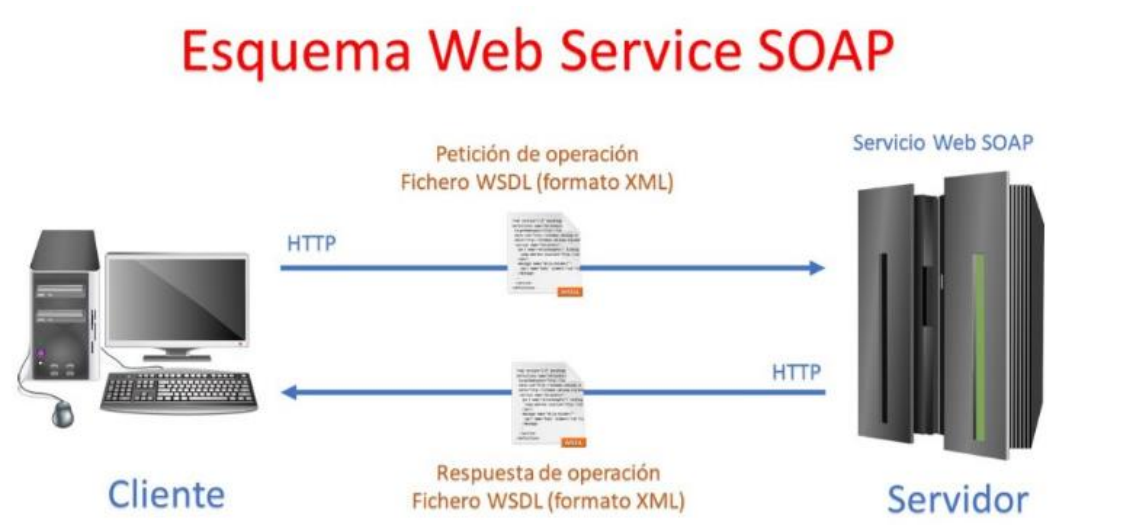
Email

Guardar

Nombre	Apellido	Email	Acciones
Germán	Iglesias	german@terra.es	<div>Editar Eliminar</div>
José	Mourinho	jose@terra.es	<div>Editar Eliminar</div>

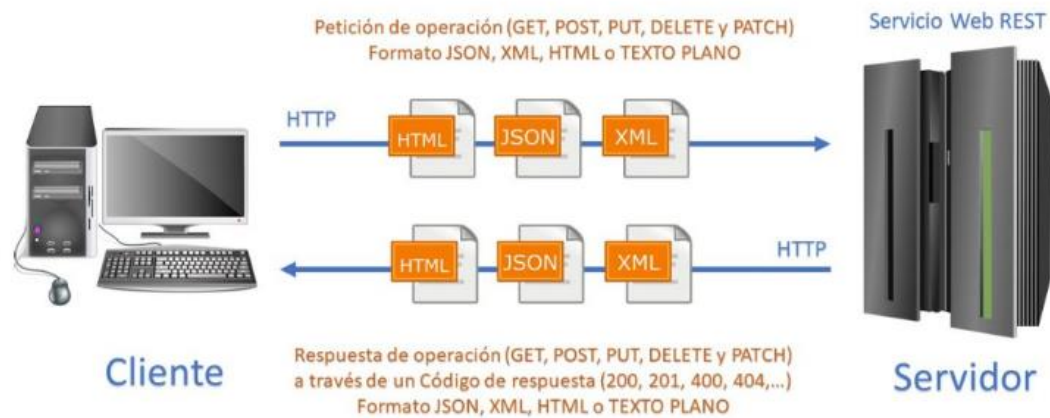
Servicios Web

SOAP



REST

Esquema Web Service REST

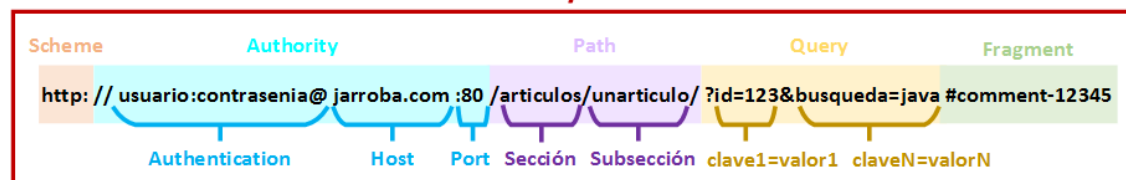


Estructura de una URL



URI

URI / URL



Enlace → <https://jarroba.com/cliente-servidor-peticion-del-cliente/>

Métodos HTTP

Operation	SQL	HTTP	DDS
Create	INSERT	PUT / POST	write
Read (Retrieve)	SELECT	GET	read / take
Update (Modify)	UPDATE	PUT / PATCH	write
Delete (Destroy)	DELETE	DELETE	dispose

JSON y XML

Comunicación entre Servicios web

XML

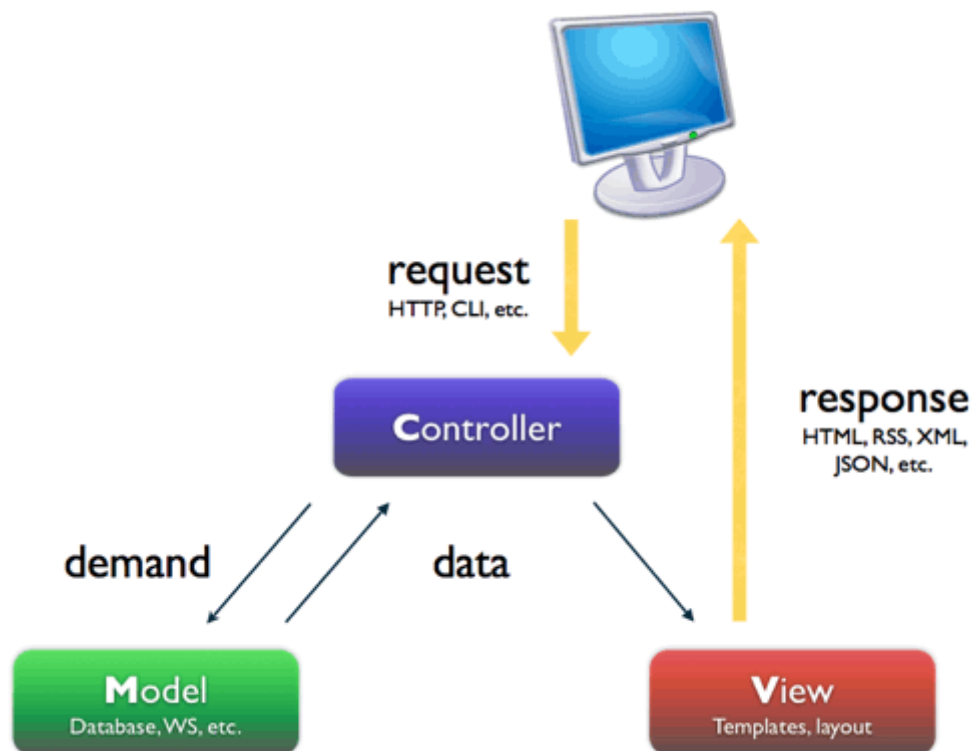
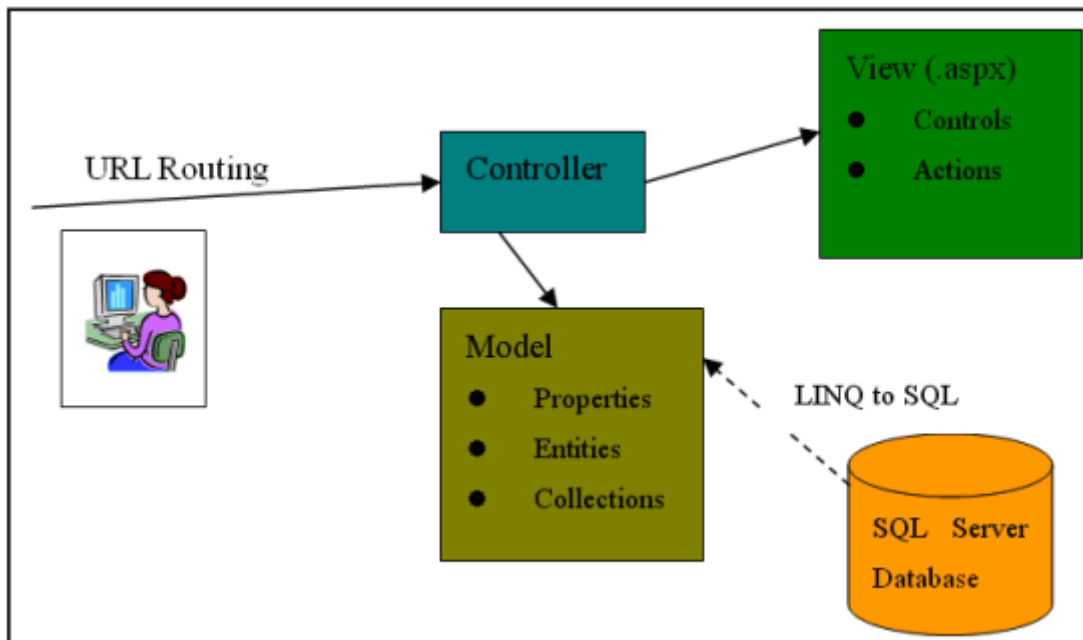
```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

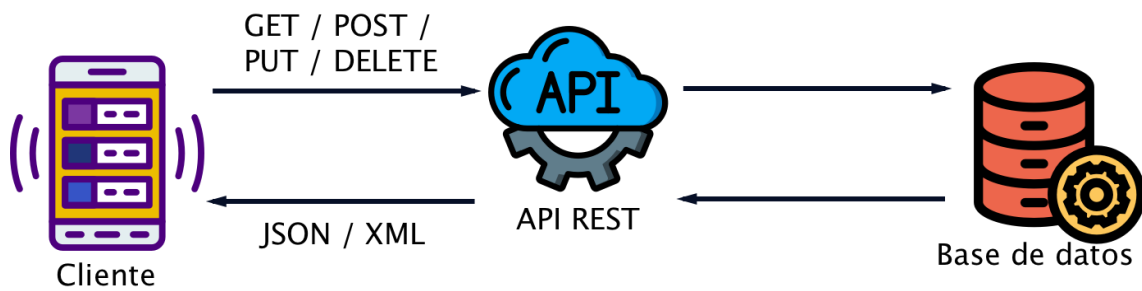
MVC y REST

MVC



NOTA: uso de Ajax (envío de emails), cientos de miles de proyectos, etc

REST



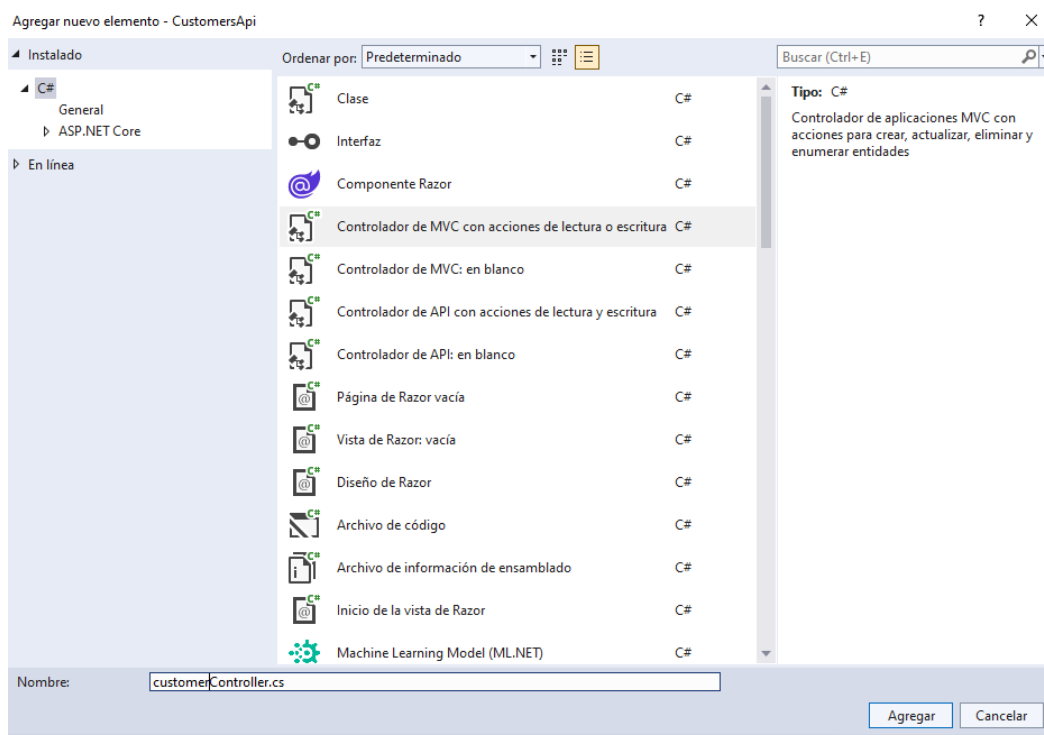
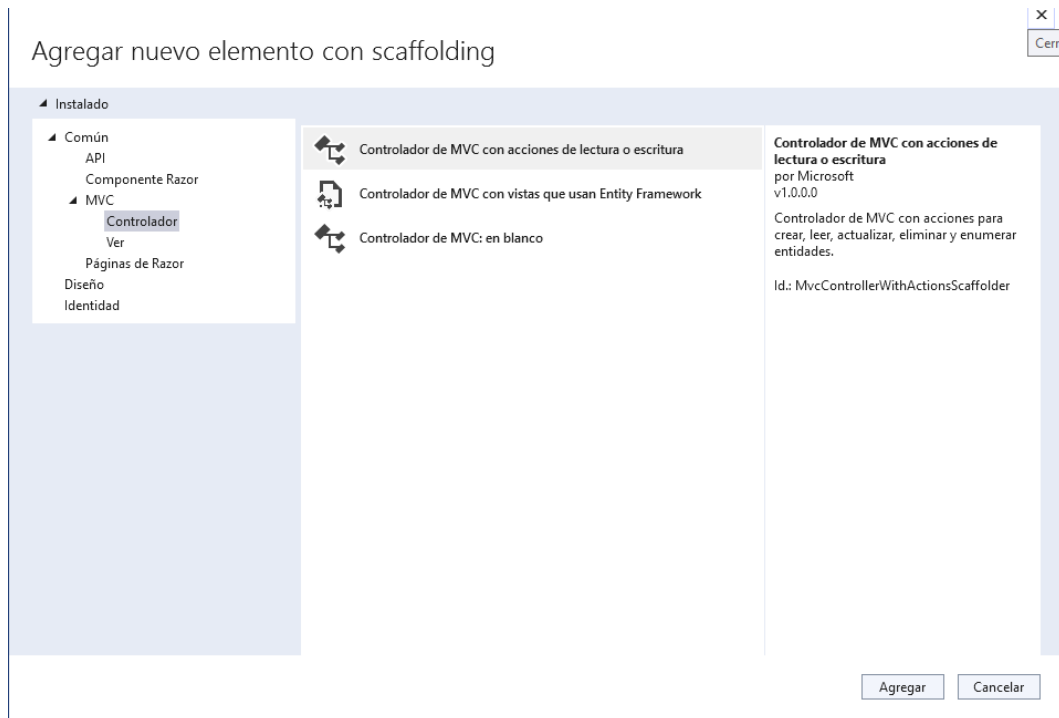
Creación de un API Rest en .Net

Pasos

- 1- Creas un proyecto API core
- 2- versión 6
- 3- Habilita los controladores y OpenIA

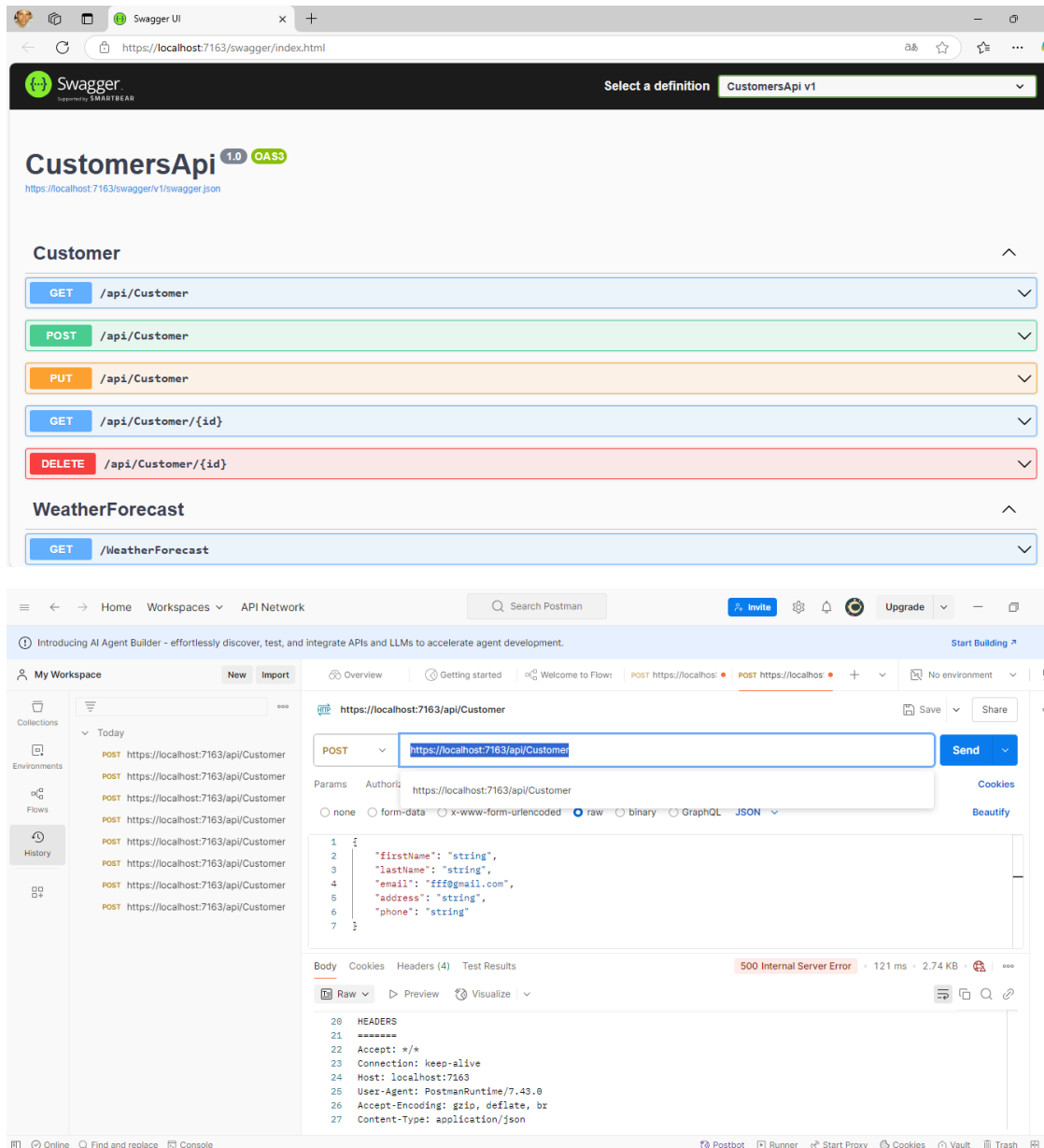


- 4- Agrega un nuevo controlador

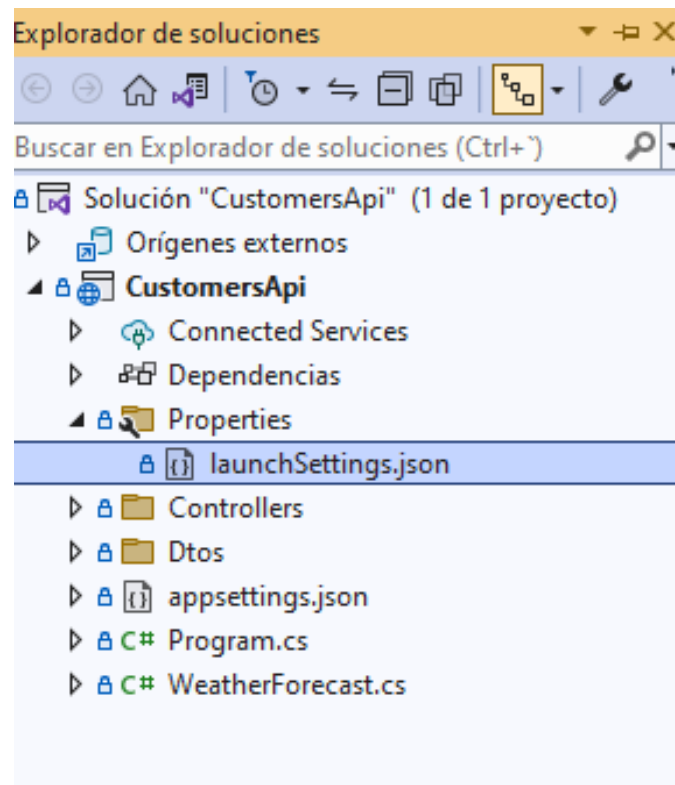


Ejecutando el código donde se construye el swagger...

NOTA: ¡pon de explorador a **Microsoft Edge**, con Google tendrás problemas!



Para definir o modificar y especificar otro número del puerto en tu proyecto ve a launchsettings.json de la carpeta Properties



```
{
  "$schema": "https://json.schemastore.org/launchsettings.json",
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:43668",
      "sslPort": 44304
    }
  },
  "profiles": {
    "CustomersApi": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "launchUrl": "swagger",
      "applicationUrl": "https://localhost:7163;http://localhost:5168",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "launchUrl": "swagger",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```


Validaciones

CreateCustomerDto.cs

```
using System.ComponentModel.DataAnnotations;

namespace CustomersApi.Dtos
{
    public class CreateCustomerDto
    {
        [Required (ErrorMessage = "El nombre propio tiene que estar especificado")]
        public string FirstName { get; set; }

        [Required(ErrorMessage = "El apellido propio tiene que estar especificado")]
        public string LastName { get; set; }
        [RegularExpression(@"^[^@\s]+@[^@\s]+\.[^@\s]+$", ErrorMessage = "El formato del email no es válido.")]
        public string Email { get; set; }
        public string Address { get; set; }
        public string Phone { get; set; }
    }
}
```

Status code

HTTP response status codes

HTTP response status codes indicate whether a specific [HTTP](#) request has been successfully completed. Responses are grouped in five classes:

1. [Informational responses](#) (100 – 199)
2. [Successful responses](#) (200 – 299)
3. [Redirection messages](#) (300 – 399)
4. [Client error responses](#) (400 – 499)
5. [Server error responses](#) (500 – 599)

- **200** OK
- **400** mala respuesta
- **401** no autorización
- **403** el cliente no posee los permisos. El cliente si está identificado, pero no tiene permisos
- **404** el recurso no lo pudo encontrar
- **500** el servidor se encuentra con un problema
- **503** el servidor no esta listo para manejar la petición. El servidor esta sobrecargado o caído

CustomerController.cs

```

using CustomersApi.Dtos;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

namespace CustomersApi.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class CustomerController : Controller
    {
        //api/customer/
        [HttpGet]
        [ProducesResponseType(StatusCodes.Status200OK, Type =
typeof(CustomerDto))]

        public async Task<IActionResult> GetCustomers()
        {
            throw new NotImplementedException();
        }
        //api/customer/{id}
        [HttpGet("{id}")]
        [ProducesResponseType(StatusCodes.Status200OK, Type =
typeof(CustomerDto))]
        [ProducesResponseType(StatusCodes.Status404NotFound)]

        public async Task<IActionResult> GetCustomer(long id)
        {
            var vacio = new CustomerDto();
            return new OkObjectResult(vacio);
        }
        [HttpDelete("{id}")]
        [ProducesResponseType(StatusCodes.Status200OK, Type = typeof(bool))]

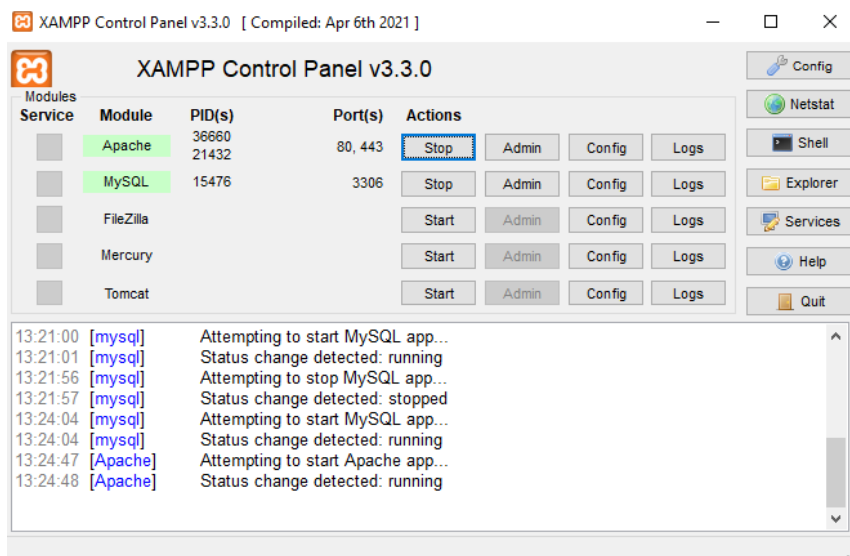
        public async Task<IActionResult> DeleteCustomer(long id)
        {
            throw new NotImplementedException();
        }
        [HttpPost]
        [ProducesResponseType(StatusCodes.Status201Created, Type =
typeof(CustomerDto))]

        public async Task<IActionResult> CreateCustomer(CreateCustomerDto
customer)
        {
            var vacio = new CustomerDto();
            return new
CreatedResult($"https://localhost:7163/api/customer/{vacio.Id}", null);
        }
        [HttpPut]
        [ProducesResponseType(StatusCodes.Status200OK, Type =
typeof(CustomerDto))]
        [ProducesResponseType(StatusCodes.Status404NotFound)]
        public async Task<IActionResult> UpdateCustomer(CustomerDto
customer)
        {
            throw new NotImplementedException();
        }
    }
}

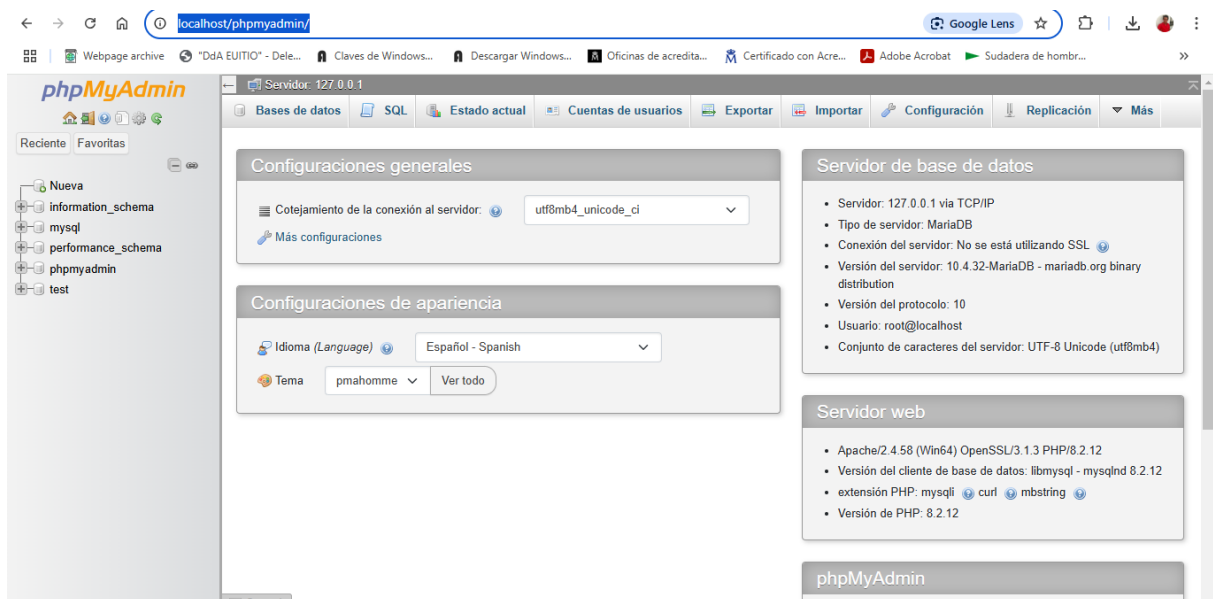
```

Xampp

Bien corre tanto php(Apache) como MySql



Ve al navegador y pon: <http://localhost/phpmyadmin/>

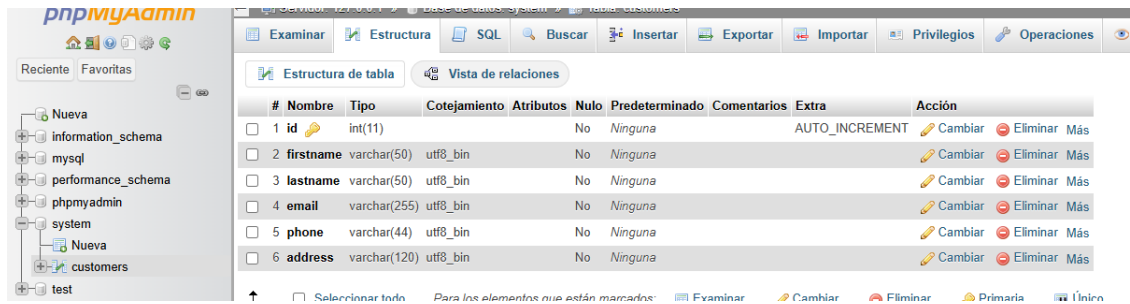


Crear BBDD

Nombre y UTF8_bin

Crear la tabla

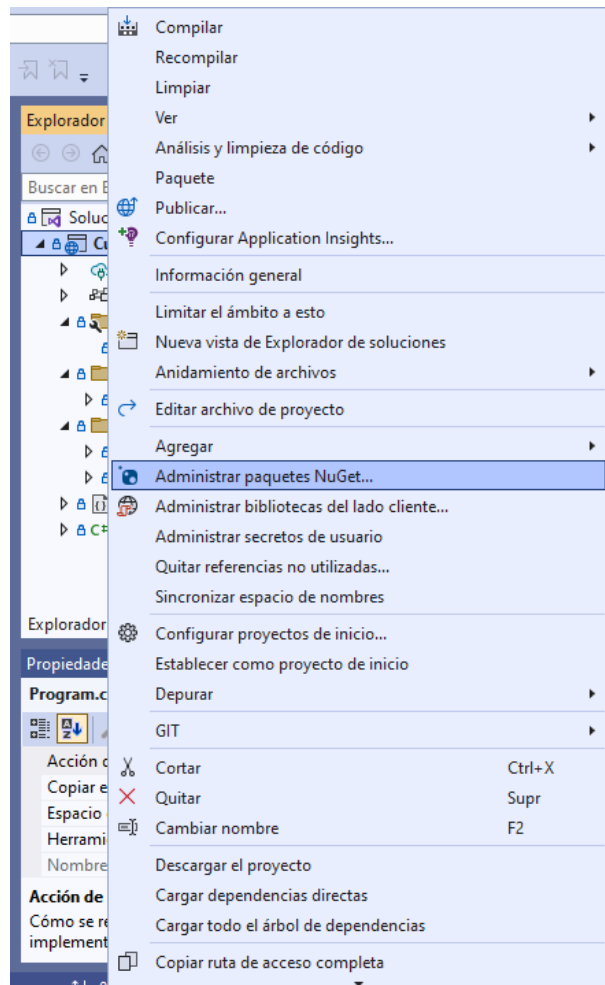
Nota: acuérdate en el id de poner autoIncrementar



Inserta valores

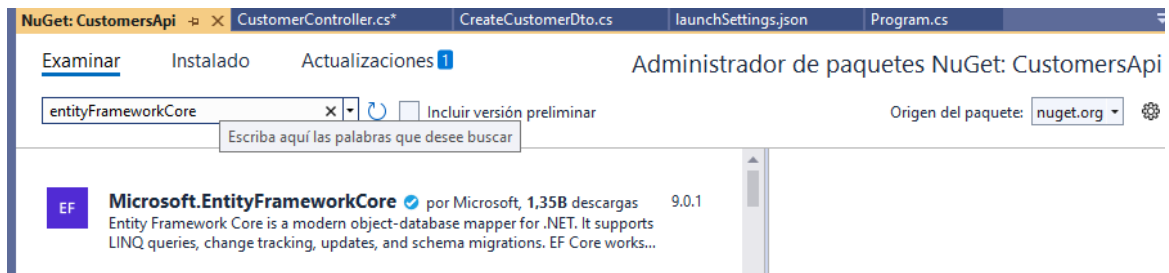
Conexión a la BBDD

EntityFramework: librería dentro de **Nugget**



Instala entityFrameworkCore

NOTA: ¡cuidado con la versión!, no tiene por qué ser la última, mira que sea lo mas cercana y compatible a tu versión inicial (en tu caso fue la seis)



Y creas una clase nuevo “DataContext”

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace CustomersApi.Repositories
{
    public class CustomerDatabaseContext : DbContext
    {
    }
}
```

Importante añadir una clase parecida al CustomerDto, pero con otro nombre. Nos resulta útil, aunque sean iguales para que se puede acceder de dos sitios (**Acuérdete de RI, dtos para bussines y para persistence**)

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace CustomersApi.Repositories
{
    public class CustomerDatabaseContext : DbContext
    {
    }

    public class CustomerEntity
    {
        public long Id { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Email { get; set; }
        public string Address { get; set; }
        public string Phone { get; set; }
    }
}
```

Accede a la BBDD y establece una conexión asíncrona (await) → Buenas practicas

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace CustomersApi.Repositories
{
    public class CustomerDatabaseContext : DbContext
    {
    }
}
```

```

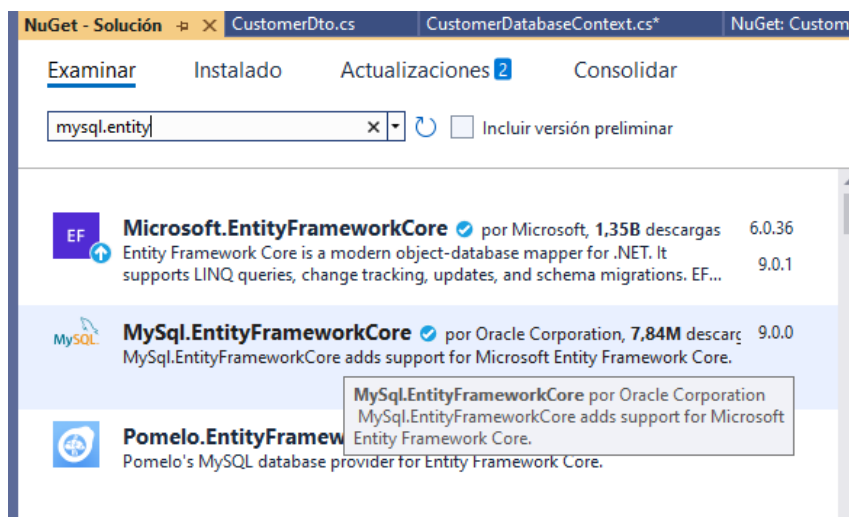
public DbSet<CustomerEntity> Customer { get; set; } // acceso a la
BBDD

public async Task<CustomerEntity> Get(long id)
{
    return await Customer.FirstAsync(x => x.Id == id); // comunica
asincrona
}

public class CustomerEntity
{
    public long Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Email { get; set; }
    public string Address { get; set; }
    public string Phone { get; set; }
}
}

```

Necesitaras volver a Nuget e instalar otro paquete en este caso mysql.entityCore



Herramienta para probar nuestra API

Postman

Herramienta Documentación

Swagger es un conjunto de herramientas y especificaciones que facilita la creación, documentación, diseño, consumo y pruebas de APIs RESTful. Es ampliamente utilizado en el desarrollo de servicios web para comunicar de manera clara cómo funcionan las APIs y cómo los clientes pueden interactuar con ellas.

Solución a la creación de la API

Program.cs

```

using CustomersApi.CasosDeUso;

```

```

using CustomersApi.Repositories;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers(); // añade el controlador que hemos creado
// Learn more about configuring Swagger/OpenAPI at
https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddRouting(routing => routing.LowercaseUrls = true); //
// minúscula en la ruta . De https://localhost:7163/api/Customer a
// https://localhost:7163/api/customer
builder.Services.AddDbContext<CustomerDatabaseContext>(mysqlBuilder =>
{
    //builder.UseMySQL("Server=localhost;Port=3306;Database=system;Uid=root;pwd=
    //"); // conexión string a nuestra bbdd

    mysqlBuilder.UseMySQL(builder.Configuration.GetConnectionString("Connection1
    "));
});

builder.Services.AddScoped<IUpdateCustomerUseCase, UpdateCustomerUseCase>();

var app = builder.Build(); // construye una app

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment()) // swagger solo desarrollo
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

// ESTO es lógica. No se recomienda ponerlo aquí
//app.MapGet("/customer/{id}", (long id) =>
//{
//    return "net 6";
//});

app.MapControllers();

app.Run();

```

appsettings.json

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "ConnectionStrings": {

```

```

        "Connection1":
        "Server=localhost;Port=3306;Database=system;Uid=root;pwd="
    },

    "AllowedHosts": "*"
}

```

CustomerDatabaseContext.cs

```

using CustomersApi.Dtos;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.ChangeTracking;

namespace CustomersApi.Repositories
{
    public class CustomerDatabaseContext : DbContext
    {
        public
        CustomerDatabaseContext(DbContextOptions<CustomerDatabaseContext> options) :
        base(options)
        {
        }

        public DbSet<CustomerEntity> Customer { get; set; } // acceso a la
        BBDD

        public async Task<CustomerEntity?> Get(long id)
        {
            return await Customer.FirstOrDefaultAsync(x => x.Id == id); //
            comunica asincrona
        }

        /// <summary>
        /// Añade un customer a la BBDD
        /// </summary>
        /// <param name="customerDto"></param>
        /// <returns></returns>
        /// <exception cref="Exception"></exception>
        public async Task<CustomerEntity> Add(CreateCustomerDto customerDto)
        {
            CustomerEntity entity = new CustomerEntity()
            {
                Id = null,
                Address = customerDto.Address,
                Email = customerDto.Email,
                FirstName = customerDto.FirstName,
                LastName = customerDto.LastName,
                Phone = customerDto.Phone
            };
            EntityEntry<CustomerEntity> response = await
            Customer.AddAsync(entity);
            await SaveChangesAsync();
            return await Get(response.Entity.Id ?? throw new Exception("no
            se ha podido añadir"));
        }

        /// <summary>
        /// Borra un customer
        /// </summary>
        /// <param name="id">Ident del customer</param>
        /// <returns></returns>
    }
}

```



```

        public async Task<bool> Delete(long id)
        {
            var entity = await Get(id);
            Customer.Remove(entity);
            SaveChanges();
            return true;
        }

        public async Task<bool> Actualizar(CustomerEntity customerEntity)
        {
            Customer.Update(customerEntity);
            await SaveChangesAsync();
            return true;
        }
    }

    public class CustomerEntity
    {
        public long? Id { get; set; } // puede ser null
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Email { get; set; }
        public string Address { get; set; }
        public string Phone { get; set; }

        public CustomerDto ToDto()
        {
            return new CustomerDto
            {
                Address = Address,
                Email = Email,
                FirstName = FirstName,
                LastName = LastName,
                Phone = Phone,
                Id = Id ?? throw new Exception("El id no puede ser null")
            };
        }
    }
}

```

CustomerDto.cs

```

namespace CustomersApi.Dtos
{
    public class CustomerDto
    {
        public long Id { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Email { get; set; }
        public string Address { get; set; }
        public string Phone { get; set; }
    }
}

```

CreateCustomerDto.cs

```

using System.ComponentModel.DataAnnotations;

namespace CustomersApi.Dtos

```

```

{
    public class CreateCustomerDto
    {
        [Required (ErrorMessage = "El nombre propio tiene que estar
especificado")]
        public string FirstName { get; set; }

        [Required(ErrorMessage = "El apellido propio tiene que estar
especificado")]
        public string LastName { get; set; }
        [RegularExpression(@"^[^@\s]+@[^@\s]+\.[^@\s]+$", ErrorMessage = "El
formato del email no es válido.")]
        public string Email { get; set; }
        public string Address { get; set; }
        public string Phone { get; set; }
    }
}

```

CustomerController.cs

```

using CustomersApi.CasosDeUso;
using CustomersApi.Dtos;
using CustomersApi.Repositories;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;

namespace CustomersApi.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class CustomerController : Controller
    {
        private readonly CustomerDatabaseContext _customerDatabaseContext;
        private readonly IUpdateCustomerUseCase _updateCustomerUseCase;

        public CustomerController(CustomerDatabaseContext
customerDatabaseContext,
            IUpdateCustomerUseCase updateCustomerUseCase)
        {
            this._customerDatabaseContext = customerDatabaseContext;
            this._updateCustomerUseCase = updateCustomerUseCase;
        }

        //api/customer/
        [HttpGet]
        [ProducesResponseType(StatusCodes.Status200OK, Type =
typeof(List<CustomerDto>))]
        public async Task<IActionResult> GetCustomers()
        {
            var result = _customerDatabaseContext.Customer.Select(c =>
c.ToDto()).ToList();
            return new OkObjectResult(result);
        }

        //api/customer/{id}
        [HttpGet("{id}")]
        [ProducesResponseType(StatusCodes.Status200OK, Type =
typeof(CustomerDto))]
        [ProducesResponseType(StatusCodes.Status404NotFound)]
        public async Task<IActionResult> GetCustomer(long id)

```

```

    {
        //var vacio = new CustomerDto();
        CustomerEntity result = await _customerDatabaseContext.Get(id);
        return new OkObjectResult(result.ToDto());
    }
    [HttpDelete("{id}")]
    [ProducesResponseType(StatusCodes.Status200OK, Type = typeof(bool))]

    public async Task<IActionResult> DeleteCustomer(long id)
    {
        var result = await _customerDatabaseContext.Delete(id);
        return new OkObjectResult(result);
    }
    [HttpPost]
    [ProducesResponseType(StatusCodes.Status201Created, Type =
typeof(CustomerDto))]

    public async Task<IActionResult> CreateCustomer(CreateCustomerDto
customer)
    {
        CustomerEntity result = await
_customerDatabaseContext.Add(customer);
        //var vacio = new CustomerDto();
        return new
CreatedResult($"https://localhost:7163/api/customer/{result.Id}", null);
    }

    [HttpPut]
    [ProducesResponseType(StatusCodes.Status200OK, Type =
typeof(CustomerDto))]
    [ProducesResponseType(StatusCodes.Status404NotFound)]
    public async Task<IActionResult> UpdateCustomer(CustomerDto
customer)
    {
        CustomerDto? result = await
_updateCustomerUseCase.Execute(customer);
        if (result == null)
        {
            return new NotFoundResult();
        }
        return new OkObjectResult(result);
    }
}
}

```

UpdateCustomerUseCase.cs

```

using CustomersApi.Dtos;
using CustomersApi.Repositories;

namespace CustomersApi.CasosDeUso
{
    public interface IUpdateCustomerUseCase
    {
        Task<CustomerDto> Execute(CustomerDto customer);
    }
    public class UpdateCustomerUseCase: IUpdateCustomerUseCase
    {
        private readonly CustomerDatabaseContext _customerDatabaseContext;
    }
}

```

```

        public UpdateCustomerUseCase(CustomerDatabaseContext
customerDatabaseContext)
        {
            _customerDatabaseContext = customerDatabaseContext;
        }

        public async Task<CustomerDto> Execute(CustomerDto customer)
        {
            //llamar a la BBDD
            var entity = await _customerDatabaseContext.Get(customer.Id);

            if (entity == null)
            {
                return null;
            }

            entity.FirstName = customer.FirstName;
            entity.LastName = customer.LastName;
            entity.Address = customer.Address;
            entity.Phone = customer.Phone;
            entity.Email = customer.Email;

            await _customerDatabaseContext.Actualizar(entity);
            return entity.ToDto();
        }
    }
}

```

Conexión entre Back and Front-end

´Pasos:

- 1- Crear .js
- 2- Del servidor vamos a recibir la info para convertir en HTML
- 3- El código HTML debemos de ponerlo en el index

NOTA: Mucho cuidado con la url, no te salía!. Tuviste que modificar el back añadiendo CORS

¿cómo queda ahora program.cs BackEnd API?

1. CORS (Cross-Origin Resource Sharing)

Dado que tu frontend está en <http://127.0.0.1:5500> y el backend en <http://localhost:5168>, se considera una solicitud de **origen cruzado**, y el navegador la puede bloquear. Necesitas asegurarte de que el backend permita CORS.

Cómo habilitar CORS en .NET

En Program.cs o Startup.cs, agrega algo así:

```

csharp
CopiarEditar
app.UseCors(builder =>

```

```
builder.WithOrigins("http://127.0.0.1:5500")
    .AllowAnyHeader()
    .AllowAnyMethod();
```

También debes asegurarte de registrar CORS en los servicios:

```
csharp
CopiarEditar
builder.Services.AddCors();
```

Si deseas una política más abierta durante el desarrollo, puedes permitir cualquier origen:

```
csharp
CopiarEditar
app.UseCors(builder =>
    builder.AllowAnyOrigin()
        .AllowAnyHeader()
        .AllowAnyMethod());
```

Enlace → <https://learn.microsoft.com/es-es/aspnet/core/security/cors?view=aspnetcore-9.0>

Program.cs

```
using CustomersApi.CasosDeUso;
using CustomersApi.Repositories;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers(); // añade el controlador que hemos creado
// Learn more about configuring Swagger/OpenAPI at
// https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddRouting(routing => routing.LowercaseUrls = true); //
// minúscula en la ruta . De https://localhost:7163/api/Customer a
// https://localhost:7163/api/customer
builder.Services.AddDbContext<CustomerDatabaseContext>(mysqlBuilder =>
{
    //builder.UseMySQL("Server=localhost;Port=3306;Database=system;Uid=root;pwd=
    //"); // conexión string a nuestra bbdd

    mysqlBuilder.UseMySQL(builder.Configuration.GetConnectionString("Connection1
    "));
});

builder.Services.AddScoped<IUpdateCustomerUseCase, UpdateCustomerUseCase>();
builder.Services.AddCors(); // añadido CORSSSSSSSS

var app = builder.Build(); // construye una app

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment()) // swagger solo desarrollo
{
```

```

        app.UseSwagger();
        app.UseSwaggerUI();
    }

    app.UseHttpsRedirection();

    app.UseAuthorization();

    // ESTO es lógica. No se recomienda ponerlo aqui
    //app.MapGet("/customer/{id}", (long id) =>
    //{
    //    return "net 6";
    //});

    // Añadida instruccion para que admita y sea mas flexible en el origen url
    app.UseCors(builder =>
        builder.AllowAnyOrigin()
            .AllowAnyHeader()
            .AllowAnyMethod());

    app.MapControllers();

    app.Run();

```

Creación de nuestro Script

Script.js

```

document.addEventListener("DOMContentLoaded", init); // Cuando se cargue
la página, se ejecuta la función search
const URL_API = 'https://localhost:7163/api/customer' // URL de la API
var customers = [] // Array de objetos con los datos de los clientes

function init() { search() } // Función que se ejecuta al cargar la
página

//alert(1); //Muestra un mensaje de alerta
function abrirFormulario() {
    //alert("Agregar Cliente"); //Muestra un mensaje de alerta
    htmlModal = document.getElementById("modal"); //Obtiene el elemento
con el id modal
    htmlModal.setAttribute("class", "modale opened"); //Agrega la clase
open al elemento modal
}

function cerrarCliente() {
    //alert("Agregar Cliente"); //Muestra un mensaje de alerta
    htmlModal = document.getElementById("modal"); //Obtiene el elemento
con el id modal
    htmlModal.setAttribute("class", "modale"); //Agrega la clase open al
elemento modal
}

function clean() { // Función para limpiar el formulario

```

```

        document.getElementById('txtId').value = ''
        document.getElementById('txtFirstName').value = ''
        document.getElementById('txtLastName').value = ''
        document.getElementById('txtPhone').value = ''
        document.getElementById('txtEmail').value = ''
        document.getElementById('txtAddress').value = ''
    }

    function agregar() { // Función para agregar un cliente
        clean() // Se limpia el formulario
        abrirFormulario() // Se abre el formulario
    }

    // Función que se ejecuta al cargar la página
    async function search() {
        var url = URL_API // URL de la API
        var response = await fetch(url, {
            "method": 'GET',
            "headers": {
                "Content-Type": 'application/json'
            }
        })
        customers = await response.json();

        var html = ''
        for (customer of customers) {
            var row = `<tr>
                <td>${customer.firstName}</td>
                <td>${customer.lastName}</td>
                <td>${customer.email}</td>
                <td>${customer.phone}</td>
                <td>
                    <a href="#" onclick="edit(${customer.id})"
class="myButton">Editar</a>
                    <a href="#" onclick="remove(${customer.id})"
class="myButtonDelete">Eliminar</a>
                </td>
            </tr>`
            html = html + row;
        }
        // Array de objetos con los datos de los clientes

        // Se añade la tabla al HTML
        document.querySelector("#customers > tbody").innerHTML = html
    }

    // Función para eliminar un cliente
    async function remove(id) {

```

```

    // Se muestra un mensaje de confirmación
    var respuesta = confirm("¿Estás seguro de que quieres eliminar este cliente?")
    // Si la respuesta es afirmativa
    if (respuesta) {
        var url = URL_API + '/' + id // URL de la API
        await fetch(url, {
            method: 'DELETE',
            headers: {
                "Content-Type": 'application/json'
            }
        })

        // 1 version. Se actualiza
        window.location.reload() // Se recarga la página

        // 2 version. Se actualiza

        // .then(response => response.json()) // Se convierte la
        // respuesta a JSON
        // .then(data => { // Se ejecuta la función search
        //     search()
        // })

        alert("Cliente eliminado") // Se muestra un mensaje de
        confirmación
    }
}

async function save() {
    // Se obtienen los datos del formulario
    var data = {
        "firstName": document.getElementById('txtFirstName').value,
        "lastName": document.getElementById('txtLastName').value,
        "phone": document.getElementById('txtPhone').value,
        "email": document.getElementById('txtEmail').value,
        "address": document.getElementById('txtAddress').value
    }

    var id = document.getElementById('txtId').value // Se obtiene el id
    del cliente
    // Si el id no está vacío, se añade al objeto
    if (id != '') {
        data.id = id
    }

    var url = URL_API // URL de la API
    await fetch(url, {

```



```

        "method": id != '' ? 'PUT' : 'POST', // Si el id no está vacío,
        se actualiza, si no, se crea
        "body": JSON.stringify(data), // Se convierte el objeto a un
        string
        "headers": {
            "Content-Type": 'application/json' // Se especifica que el
            contenido es JSON
        }
    })

    // 1 version. Se actualiza
    window.location.reload() // Se recarga la página
}

//
function edit(id) { // Función para editar un cliente
    // Se busca el cliente con el id seleccionado
    var customer = customers.find(x => x.id == id)
    // Se muestran los datos del cliente en el formulario
    document.getElementById('txtId').value = customer.id
    document.getElementById('txtFirstName').value = customer.firstName
    document.getElementById('txtLastName').value = customer.lastName
    document.getElementById('txtPhone').value = customer.phone
    document.getElementById('txtEmail').value = customer.email
    document.getElementById('txtAddress').value = customer.address
    // Se abre el formulario
    abrirFormulario()
}

```

De esta “guisa” quedó el **index.html**

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="style.css"> <!-- Enlaza el archivo
style.css -->
    <link rel="stylesheet" href="menu.css">
    <script src="script.js"></script>
    <title>Index</title>
</head>
<nav id='menu'>

```

```

    <input type='checkbox' id='responsive-menu'
onclick='updatemenu()'><label></label>
    <ul>
      <li><a href='http://>Home</a></li>
      <li><a class='dropdown-arrow' href='http://>Products</a>
        <ul class='sub-menus'>
          <li><a href='http://>Products 1</a></li>
          <li><a href='http://>Products 2</a></li>
          <li><a href='http://>Products 3</a></li>
          <li><a href='http://>Products 4</a></li>
        </ul>
      </li>
      <li><a href='http://>About</a></li>
      <li><a class='dropdown-arrow' href='http://>Services</a>
        <ul class='sub-menus'>
          <li><a href='http://>Services 1</a></li>
          <li><a href='http://>Services 2</a></li>
          <li><a href='http://>Services 3</a></li>
        </ul>
      </li>
      <li><a href='http://>Contact Us</a></li>
    </ul>
</nav>

<body>
  <h1>Gestión de Clientes</h1>
  <a onclick="agregar()" href="#" class="myButton">Agregar Cliente</a>
  <table id="customers">
    <thead>
      <tr>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Email</th>
        <th>Teléfono</th>
        <th>Acciones</th>
      </tr>
    </thead>
    <tbody>

  </tbody>
</table>
<!-- Modal -->
<div id="modal" class="modale" aria-hidden="true">
  <div class="modal-dialog">

    <div class="modal-header">
      <h2>Datos del Cliente</h2>
      <a onclick="cerrarCliente()" href="#" class="btn-close
closemodale" aria-hidden="true">&times;</a>

```

```

    </div>

    <div class="modal-body">
        <input id="txtId" type="hidden" placeholder="Id"
size="20" /><br/>
        <input id="txtFirstName"
type="text" placeholder="Nombre" size="20" /><br/>
        <input id="txtLastName"
type="text" placeholder="Apellido" size="20" /><br/>
        <input id="txtPhone" type="text" placeholder="Teléfono"
size="20" /><br/>
        <input id="txtEmail" type="text" placeholder="Email"
size="20" /><br/>
        <input id="txtAddress" type="text" placeholder="Address"
size="20" /><br/>
    </div>

    <div class="modal-footer">
        <a onclick="save()" href="#" class="btn"
id="btn_ingresar">Guardar</a>
    </div>
</div>
</div>
<!-- /Modal -->

</body>

<script>

</script>

</html>

```

Vista final del proyecto

Nombre	Apellido	Email	Teléfono	Acciones
German	Iglesias Ramos	german@altavista.es	985 297 897	Editar Eliminar
Rosa	Alejandría	rosa@eresmas.es	654 756 862	Editar Eliminar
Pepe	Julías	juli@ono.es	902 20 21 22	Editar Eliminar

Swagger UI interface for the Customer API endpoint.

Customer

GET /api/customer

Parameters

No parameters

Execute **Clear**

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7163/api/customer' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7163/api/customer

Server response

Code **Details**

phpMyAdmin interface showing the 'customer' table in the 'system' database.

localhost/phpmyadmin/index.php?route=/sql&db=system&table=customer&pos=0

Interrumpir la carga de esta página

Claves de Windows... Descargar Windows... Oficinas de acredita... Certificado con Acre... Adobe Acrobat Sudadera de hombr...

Servidor: 127.0.0.1 Base de datos: system Tabla: customer

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Más

Opciones extra

	id	firstname	lastname	email	phone	address
<input type="checkbox"/>	1	German	Iglesias Ramos	german@altavista.es	985 297 897	Otero,12
<input type="checkbox"/>	5	Rosa	Alejandria	rosa@eresmas.es	654 756 862	Calvo Sotelo,12
<input type="checkbox"/>	6	Pepe	Julias	juli@ono.es	902 20 21 22	Ramiro I

☐ Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar Exportar

☐ Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista

Guardar esta consulta en favoritos

Etiqueta: ☐ Permitir que todo usuario pueda acceder a este favorito

Guardar esta consulta en favoritos