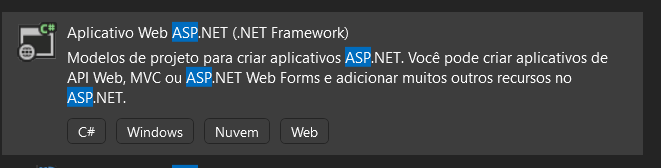
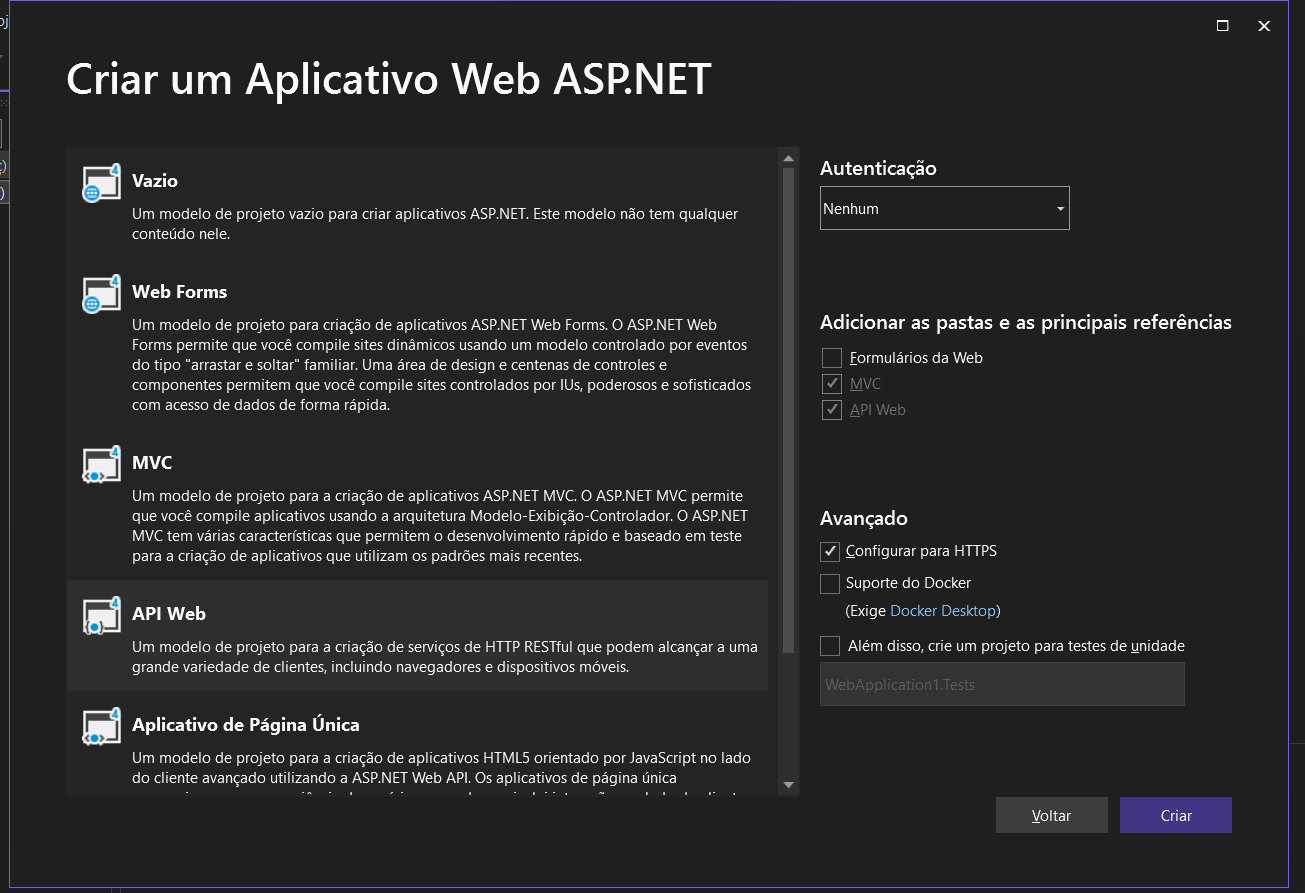
**ASP.NET**

**Criando o projeto Web api:**

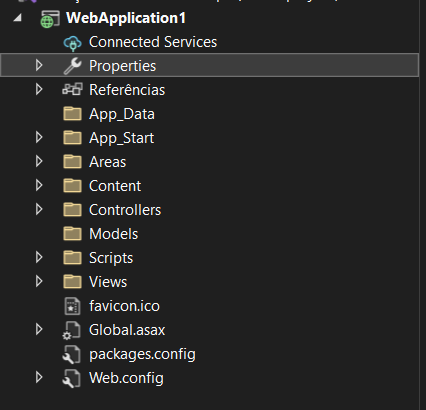
Primeiro criamos uma solução em branco, após isso adicionamos um projeto com o template:



Após isso devemos configurar o template inicial do projeto:



Vamos usar o api web, após isso teremos uma webAplicattion padrão:



**Analisando a estrutura das rotas http:**

Podemos analisar uma classe controller que possuirá métodos públicos REST mapeados:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace WebApplication1.Controllers

{

public class ValuesController : ApiController

{

// GET api/values

public IEnumerable<string> Get()

{

return new string[] { "value1", "value2" };

}

// GET api/values/5

public string Get(int id)

{

return "value";

}

// POST api/values

public void Post([FromBody] string value)

{

}

// PUT api/values/5

public void Put(int id, [FromBody] string value)

{

}

// DELETE api/values/5

public void Delete(int id)

{

}

}

}

Analisando essa classe percebemos que ela extende a classe ApiController, responsável por definir os controllers de Rotas Rest, a url do projeto é mapeada na classe WebApiConfig:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web.Http;

namespace WebApplication1

{

public static class WebApiConfig

{

public static void Register(HttpConfiguration config)

{

// Configuração e serviços de API Web

// Rotas de API Web

config.MapHttpAttributeRoutes();

config.Routes.MapHttpRoute(

name: "DefaultApi",

routeTemplate: "api/{controller}/{id}",

defaults: new { id = RouteParameter.Optional }

);

}

}

}

Isso significa que toda requisição “api/nomeDoController” sera redirecionada para a classe com o nome especifico, exemplo:

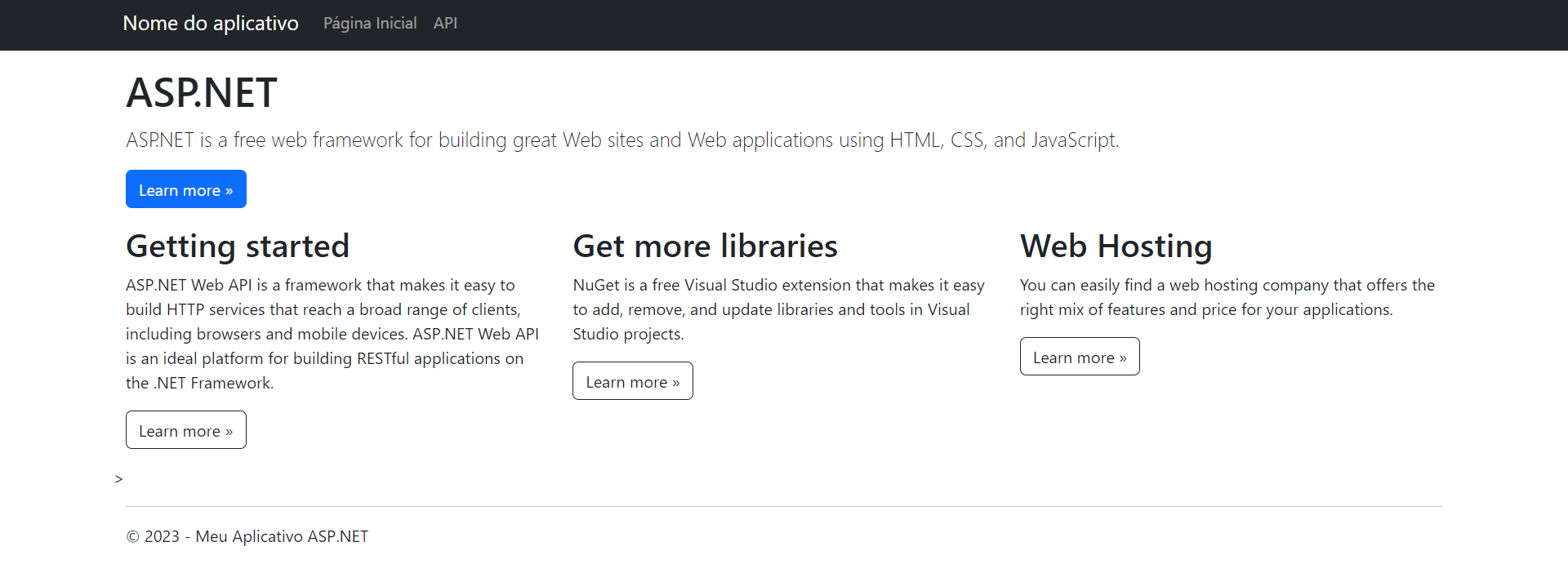
🡪Uma requisição com URL api/values será tratada pelo controlador de nome ValuesController

**IIS EXPRESS**

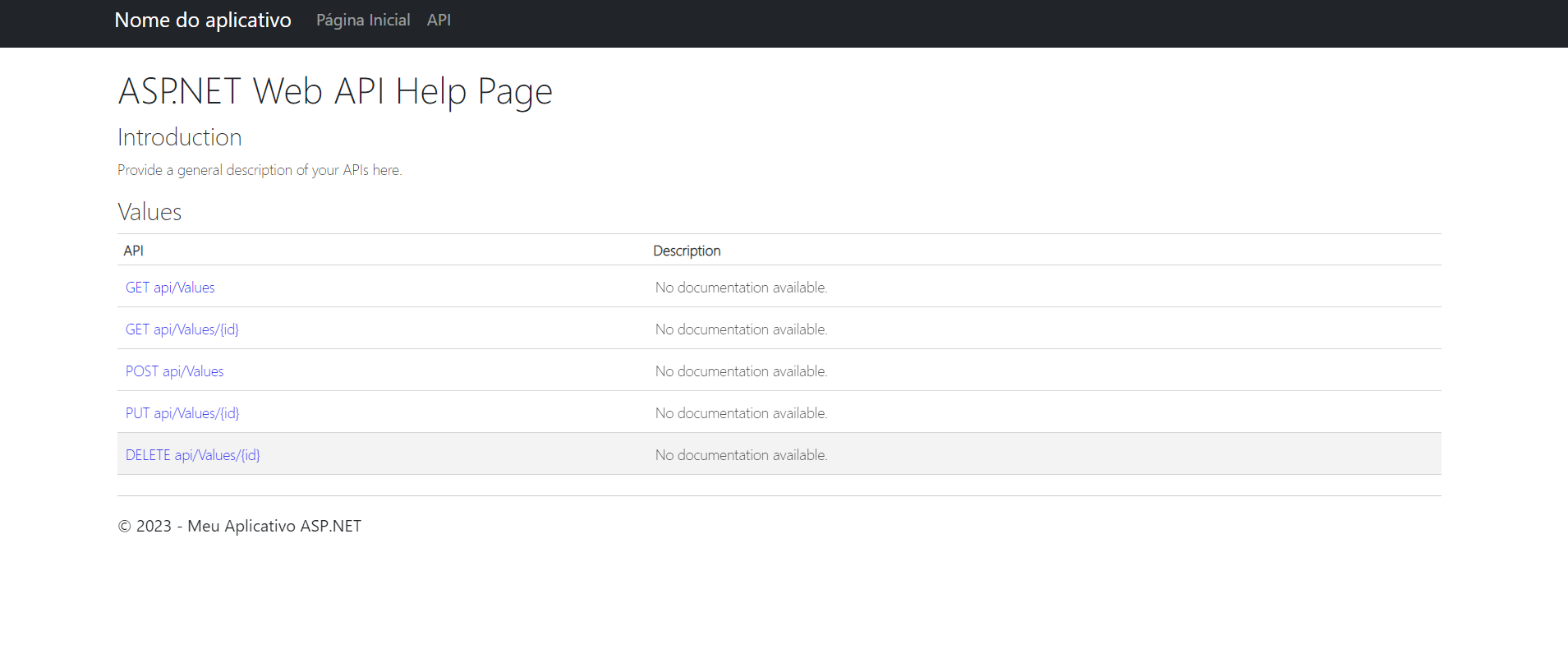
O IIS (Internet Information Service) Express da Microsoft permite que as aplicações Web API sejam executadas e depuradas na máquina local, sem a necessidade de um ambiente de hospedagem terceiro. Para tal, dentro do Visual Studio, basta clicar no nome do navegador padrão, na barra de ferramentas superior, que o projeto abrirá em execução no IIS Express:



Após isso teremos uma tela de index padrão:



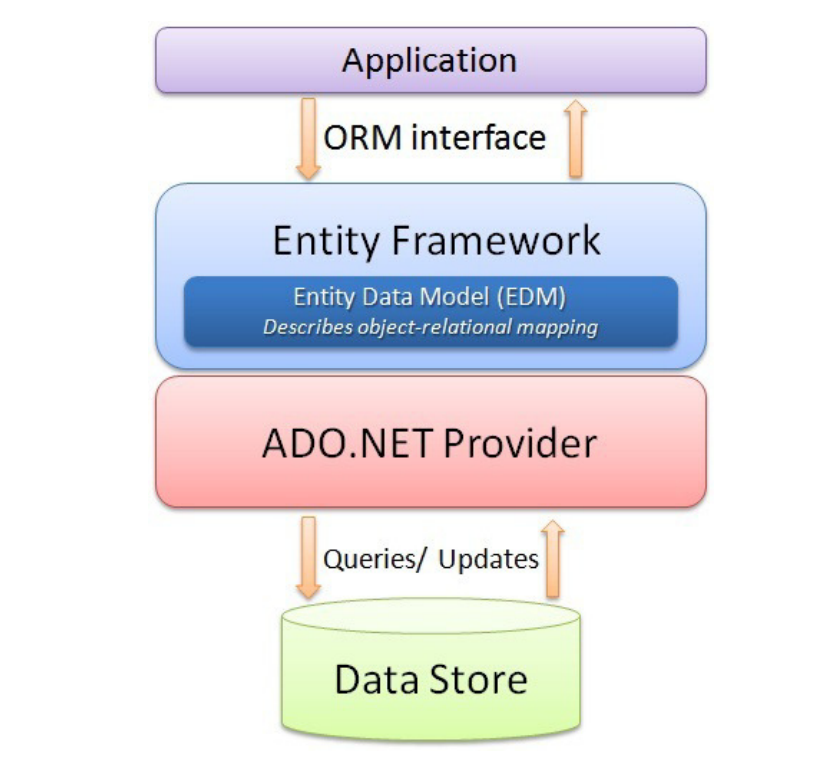
A parte interessante dessa pagina é a parte de API:



Perceba temos uma documentação completa das rotas da api.

**Entity Framework**

O Entity Framework é um poderoso Object Relational Mapper (ORM), que gera objetos de negócios e entidades de acordo com as tabelas do banco de dados. Veja o diagrama de sua estrutura:



O Entity Framework possui 3 linhas de atuação principal:

**.Database First 🡪** Primeiro é feito a database no banco de dados e depois é interligada no Entity Framework.

**.Model First 🡪** Primeiro é feito o modelo no sistema e a partir dele podemos gerar nossa base de dados

**.Code First 🡪** Primeiro é feito o modelo no sistema e depois é de responsabilidade do Entity Framework criar a database.

**Data Annotations:**

Os data Anotations é um recurso que permite que você adicione atributos e métodos em nossas classes para alterar convenções padrão e personalizar alguns comportamentos.

**Prindipais Atributos:**

**🡪 Required:** Significa campo obrigatório**.**

**🡪 RegularExpression:** Valida o campo por expressão regular.

**🡪 Display:** Nome a ser mostrado em todas as interfaces de usuário.

**🡪 StringLength:** Determina a quantidade máxima de caracteres que poderá ser informada.

**🡪 MinLength:** Determina a quantidade mínima de caracteres que poderá ser informada**.**

**🡪DisplayFormat:** Formato a ser exibido nas interfaces de usuário.

**🡪Range:** Define a faixa de dados aceita pela propriedade.

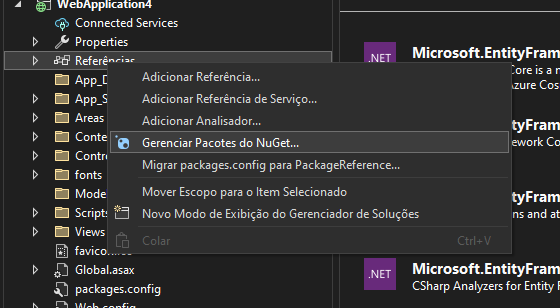
**Migrations:**

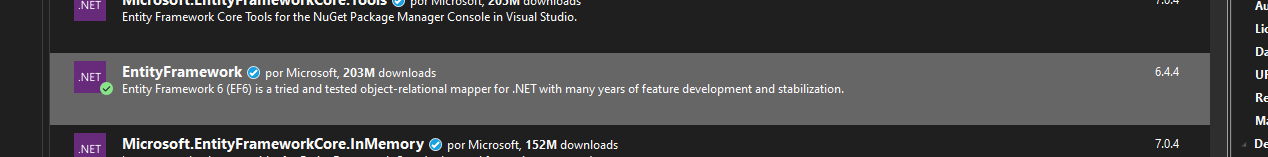
É um recurso que oferece uma maneira de atualizar o banco de dados de forma incremental para manter a sincronia com os modelos de classes.

Também é possível através do Migrations fazer o Downgrade caso você deseje voltar para a versão anterior em que se encontrava.

**Aplicando o Entity Framework ao Projeto:**

Primeiro devemos instalar o Entity Framework:





**Configurando o banco de dados:**

Após adicionarmos o EntityFramework ao projeto nos devemos criar uma classe de contexto onde tera a configuração do banco de dados.

Para isso criarei como exemplo um banco de dados com uma entidade: 