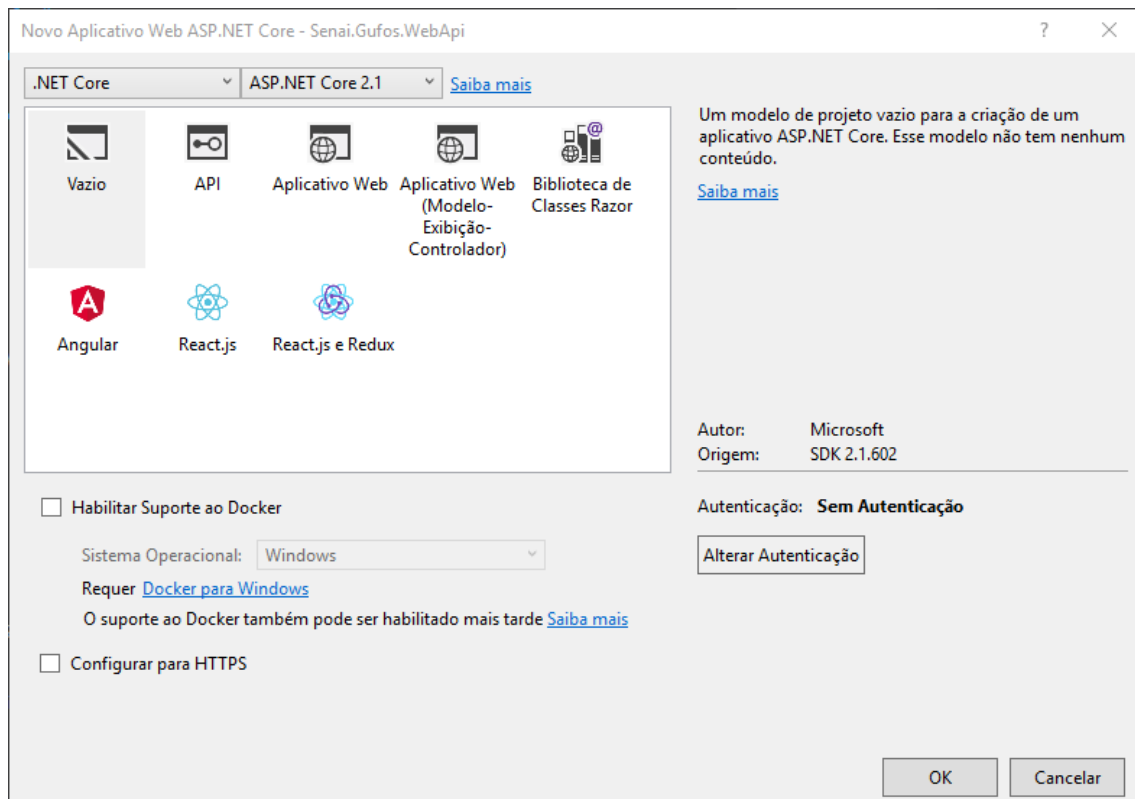
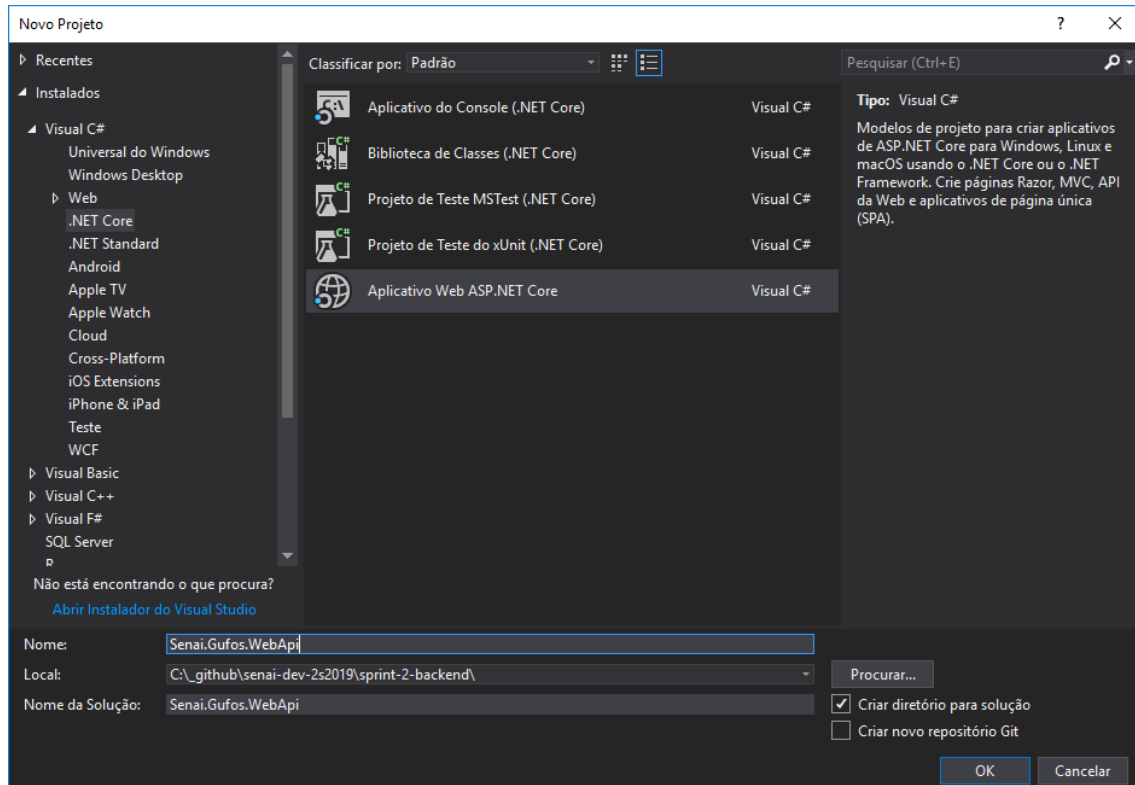


Apresentação EF Core

Criar novo projeto – Senai.Gufos.WebApi

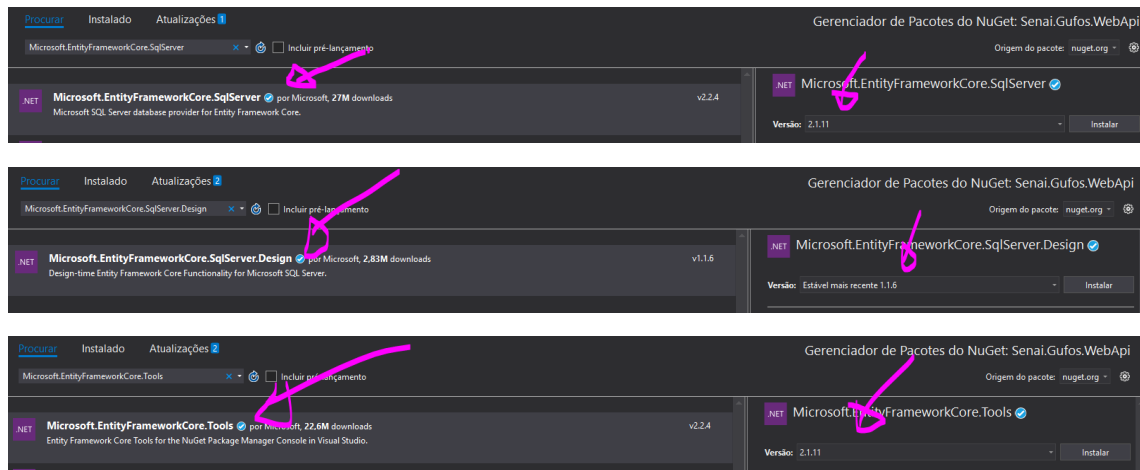


Gerenciar Pacotes do NuGet

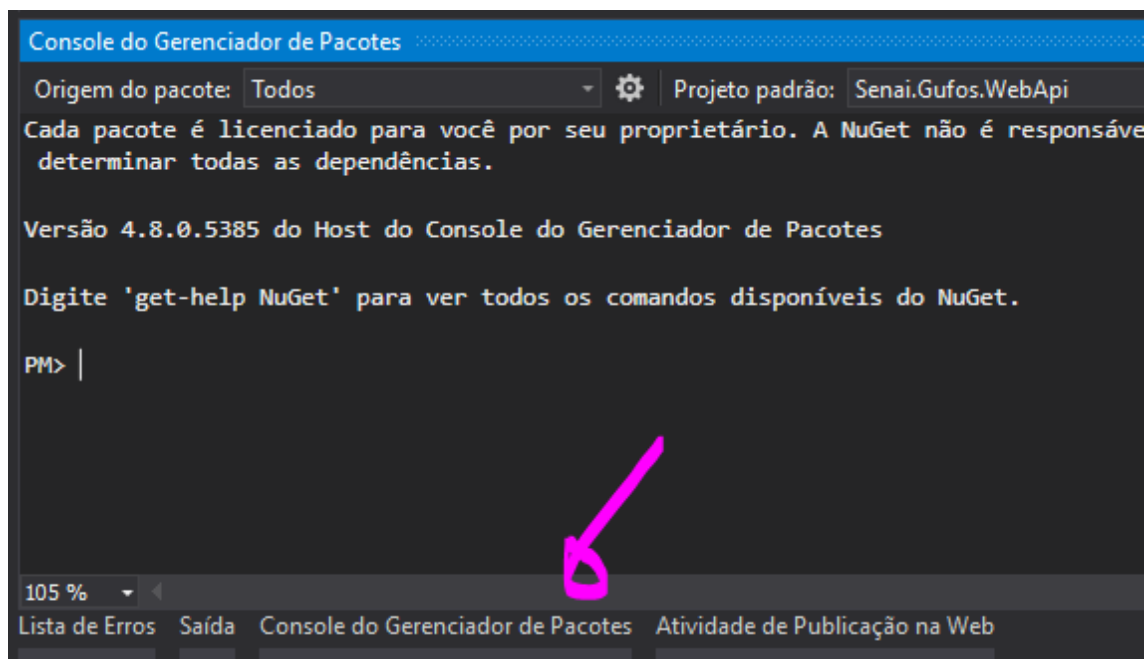
Microsoft.EntityFrameworkCore.SqlServer – versão 2.1.11

Microsoft.EntityFrameworkCore.SqlServer.Design – versão 1.1.6

Microsoft.EntityFrameworkCore.Tools – versão 2.1.11

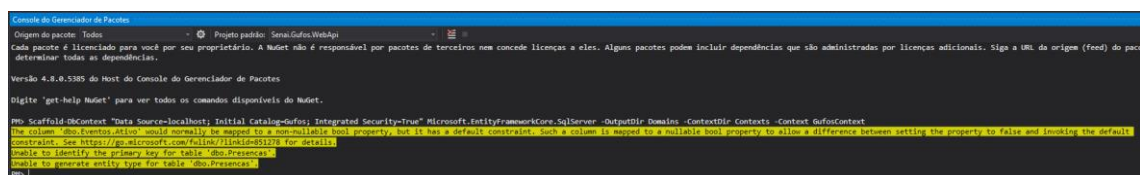


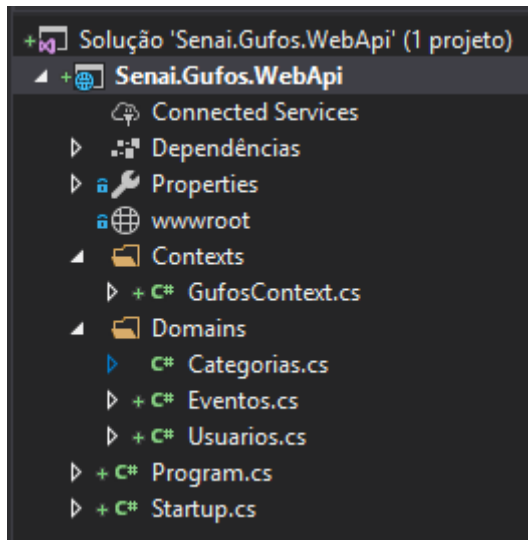
Console de Gerenciador de Pacotes



Scaffold-DbContext "Data Source=localhost; Initial Catalog=Gufos; Integrated Security=True"

Microsoft.EntityFrameworkCore.SqlServer -OutputDir Domains -ContextDir Contexts -Context GufosContext





Configurar Startup

```
public class Startup
{
    // This method gets called by the runtime. Use this method to add services to the container.
    // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
    }

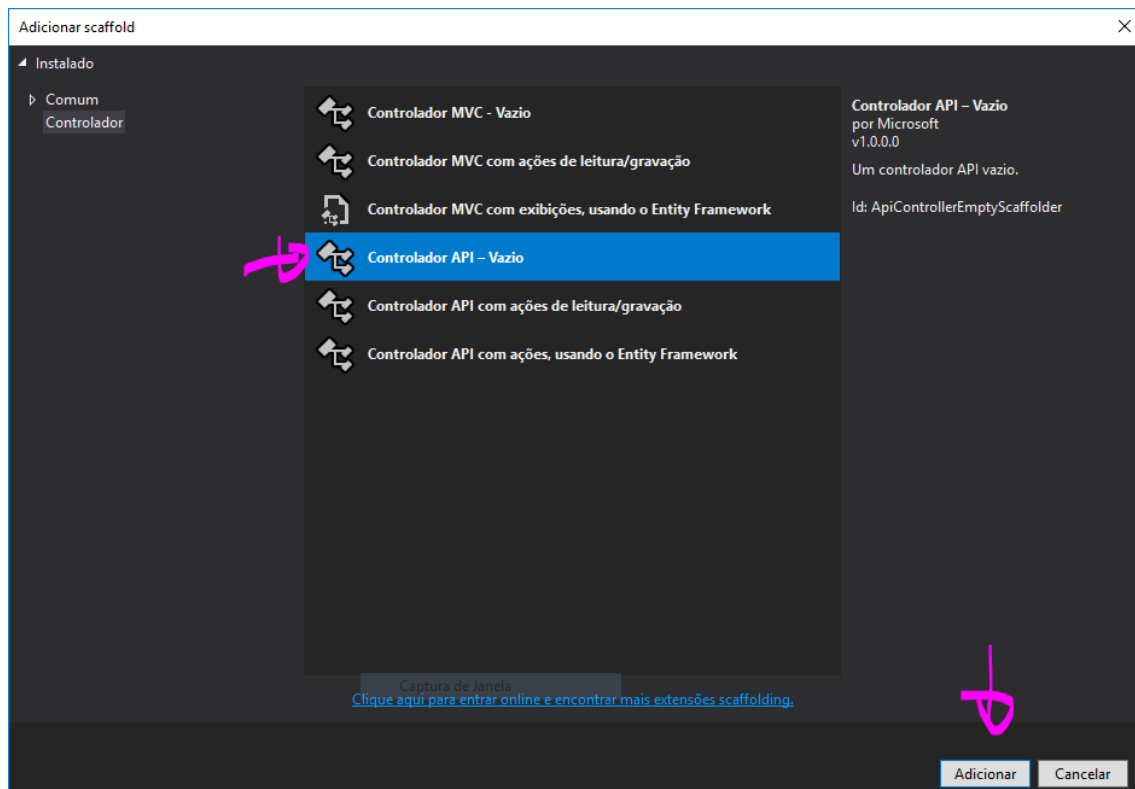
    // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    {
        if (env.IsDevelopment())
        {
            app.UseDeveloperExceptionPage();
        }

        app.UseMvc();

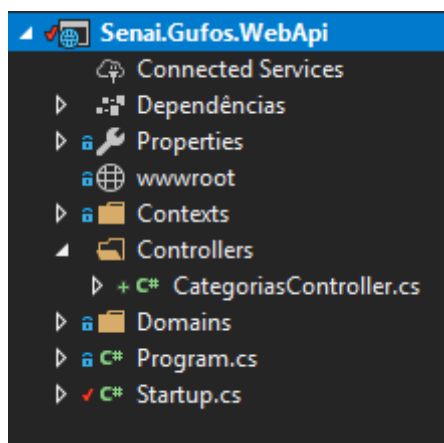
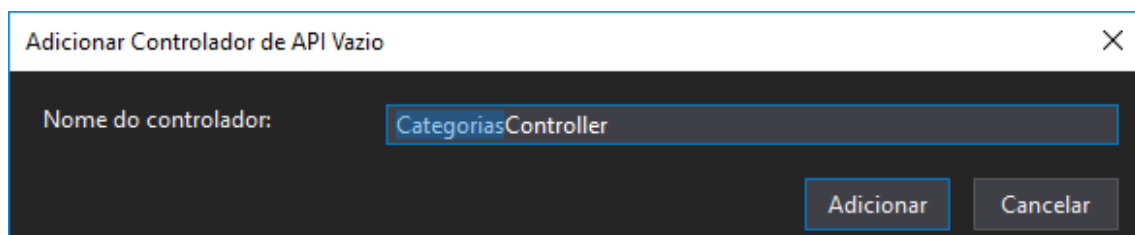
        //app.Run(async (context) =>
        //{
        //    await context.Response.WriteAsync("Hello World!");
        //});
    }
}
```

Criar pasta chamada Controllers.

Adicionar novo controlador – Controlador API – Vazio



CategoriasController.cs



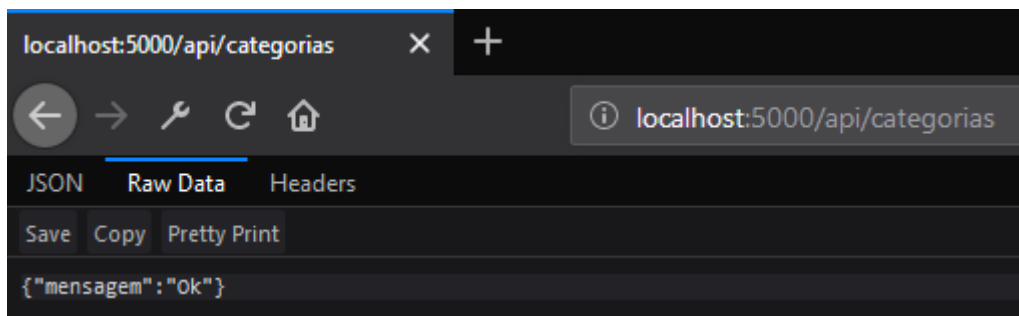
Alterar CategoriasController.cs

```

[Route("api/[controller]")]
[Produces("application/json")]
[ApiController]
public class CategoriasController : ControllerBase
{
    [HttpGet]
    public IActionResult ListarTodos()
    {
        return Ok(new { mensagem = "Ok" });
    }
}

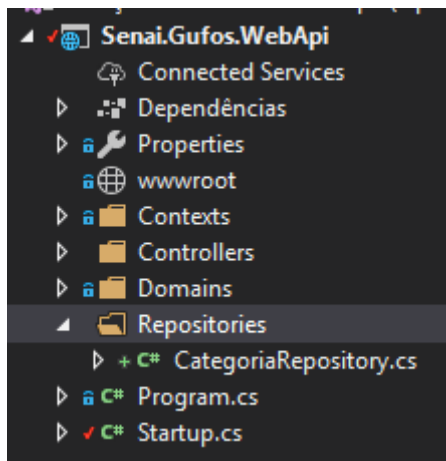
```

localhost:5000/api/categorias



Criar pasta Repositories

Adicionar nova classe CategoriaRepository.cs



Criar método Listar()

```

public class CategoriaRepository
{
    public List<Categorias> Listar()
    {
        using (GufosContext ctx = new GufosContext())
        {
            return ctx.Categorias.ToList();
        }
    }
}

```

Alterar no CategoriasController.cs

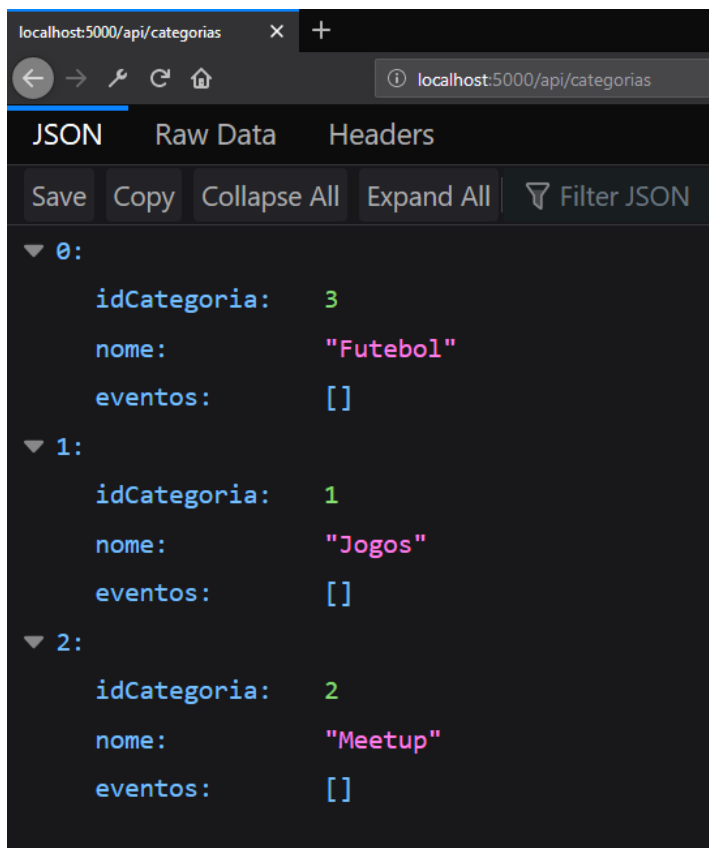
```

CategoriaRepository categoriaRepository = new CategoriaRepository();

[HttpGet]
public IActionResult ListarTodos()
{
    return Ok(categoriaRepository.Listar());
}

```

Visualizar resultado no Browser.



Cadastrar

CategoriaRepository.cs

```

public void Cadastrar(Categorias categoria)
{
    using (GufosContext ctx = new GufosContext())
    {
        ctx.Categorias.Add(categoria);
        ctx.SaveChanges();
    }
}

```

CategoriasController.cs

```

[HttpPost]
public IActionResult Cadastrar(Categorias categoria)
{
    try
    {
        CategoriaRepository.Cadastrar(categoria);
        return Ok();
    }
    catch (Exception)
    {
        return BadRequest();
    }
}

```

► Categorias.Cadastrar

POST	localhost:5000/api/categorias
------	-------------------------------


Params Authorization Headers (9) **Body** Pre-request Script Tests

☐ none
☐ form-data
☐ x-www-form-urlencoded
☒ raw
☐ binary
☐ GraphQL ^{BETA}
☒ JSON (application/json)

```

1 {
2   "nome" : "Ping-Pong"
3 }

```

▶ Categorias.Listar 

GET ▼

localhost:5000/api/categorias

Params

Authorization

Headers (7)

Body

Query Params

	KEY
	Key

Body

Cookies

Headers (4)


Test Results

Pretty

Raw

Preview

JSON ▼



1 ▼

2 ▼

3

4

5

6

7 ▼

8

9

10

11

12 ▼

13

14

15

16

17 ▼

18

19

20

21

22

```
[
  {
    "idCategoria": 3,
    "nome": "Futebol",
    "eventos": []
  },
  {
    "idCategoria": 1,
    "nome": "Jogos",
    "eventos": []
  },
  {
    "idCategoria": 2,
    "nome": "Meetup",
    "eventos": []
  },
  {
    "idCategoria": 4,
    "nome": "Ping-Pong",
    "eventos": []
  }
]
```



```
[HttpGet("{id}")]
public IActionResult BuscarPorId(int id)
{
    Categorias Categoria = CategoriaRepository.BuscarPorId(id);
    if (Categoria == null)
    {
        return NotFound();
    }
    return Ok(Categoria);
}
```

```
public Categorias BuscarPorId(int id)
{
    using (GufosContext ctx = new GufosContext())
    {
        return ctx.Categorias.FirstOrDefault(x => x.IdCategoria == id);
    }
}
```

GET localhost:5000/api/categorias/4

Params Authorization Headers (7) Body

Query Params

KEY
Key

Body Cookies Headers (4) Test Results

Pretty Raw Preview JSON

```
1 {
2   "idCategoria": 4,
3   "nome": "Ping-Pong",
4   "eventos": []
5 }
```

Deletar

CategoriaRepository.cs

```

public void Deletar(int id)
{
    using (GufosContext ctx = new GufosContext())
    {
        Categorias CategoriaBuscada = ctx.Categorias.Find(id);
        ctx.Categorias.Remove(CategoriaBuscada);
        ctx.SaveChanges();
    }
}

```

```

[HttpDelete("{id}")]
public IActionResult Deletar(int id)
{
    CategoriaRepository.Deletar(id);
    return Ok();
}

```

► Categorias.Deletar

DELETE ▼	localhost:5000/api/categorias/5
----------	---------------------------------

Atualizar

```

public void Atualizar(Categorias categoria)
{
    using (GufosContext ctx = new GufosContext())
    {
        Categorias CategoriaAtualizada = ctx.Categorias.FirstOrDefault(x => x.IdCategoria == categoria.IdCategoria);
        CategoriaAtualizada.Nome = categoria.Nome;
        ctx.Categorias.Update(CategoriaAtualizada);
        ctx.SaveChanges();
    }
}

```

```

[HttpPut]
public IActionResult Atualizar(Categorias categoria)
{
    try
    {
        Categorias CategoriaBuscada = CategoriaRepository.BuscarPorId(categoria.IdCategoria);
        if (CategoriaBuscada == null)
            return NotFound();

        CategoriaRepository.Atualizar(categoria);
        return Ok();
    }
    catch (Exception)
    {
        return BadRequest();
    }
}

```

► Categorias.Atualizar

PUT ▼ localhost:5000/api/categorias

Params Authorization Headers (9) Body ● Pre-req

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw

```
1 {  
2   "idCategoria": 6,  
3   "nome": "Tênis de Mesa - O Retorno"  
4 }
```

EVENTOS

EventosController.cs

Adicionar scaffold

Instalado

Comum

Controlador

- Controlador MVC - Vazio
- Controlador MVC com ações de leitura/gravação
- Controlador MVC com exibições, usando o Entity Framework
- Controlador API - Vazio**
- Controlador API com ações de leitura/gravação
- Controlador API com ações, usando o Entity Framework

Controlador API - Vazio
por Microsoft
v1.0.0.0
Um controlador API vazio.
Id: ApiControllerEmptyScaffolder

[Clique aqui para entrar online e encontrar mais extensões scaffolding.](#)

Adicionar Cancelar

Adicionar Controlador de API Vazio

Nome do controlador: EventosController

Adicionar Cancelar

```
[Route("api/[controller]")]
[Produces("application/json")]
[ApiController]
public class EventosController : ControllerBase
{
}
```

```
[HttpGet]
public IActionResult Listar()
{
    return Ok();
}
```

EventoRepository.cs

```
public class EventoRepository
{
    public List<Eventos> Listar()
    {
        using (GufosContext ctx = new GufosContext())
        {
            return ctx.Eventos.ToList();
        }
    }
}
```

EventosController.cs

```
[Route("api/[controller]")]
[Produces("application/json")]
[ApiController]
public class EventosController : ControllerBase
{
    EventoRepository EventoRepository = new EventoRepository();

    [HttpGet]
    public IActionResult Listar()
    {
        return Ok(EventoRepository.Listar());
    }
}
```

Criar pasta no postman.

GET Eventos.Listar

Eventos.Listar

GET localhost:5000/api/eventos

Params Authorization Headers (7) Body Pre-request Script

Query Params

KEY
Key

Body Cookies Headers (4) Test Results

Pretty Raw Preview JSON

```
1 [
2   {
3     "idEvento": 1,
4     "titulo": "Titulo A",
5     "descricao": "Descricao A",
6     "dataEvento": "2019-06-17T14:26:46.513",
7     "ativo": true,
8     "localizacao": "Alameda Barão de Limeira, 539",
9     "idCategoria": 1,
10    "idCategoriaNavigation": null
11  },
12  {
13    "idEvento": 2,
14    "titulo": "Titulo B",
15    "descricao": "Campeonato de Futebol",
16    "dataEvento": "2019-06-17T14:27:08.4",
17    "ativo": true,
18    "localizacao": "Alameda Barão de Limeira, 539",
19    "idCategoria": 3,
20    "idCategoriaNavigation": null
21  }
22 ]
```

Incluir a categoria nos eventos.

EventoRepository.cs

```
public List<Eventos> Listar()
{
    using (GufosContext ctx = new GufosContext())
    {
        return ctx.Eventos.Include(x => x.IdCategoriaNavigation).ToList();
    }
}
```

Could not beautify

Events.Listar

GET localhost:5000/api/eventos

Send Save

Params Authorization Headers (7) Body Pre-request Script Tests

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 2962 ms Size: 391 B Save Download

Pretty Raw Preview JSON

```
[{"IdEvento":1,"titulo":"Título A","descricao":"Descrição A","dataEvento":"2019-06-17T14:26:46.513","ativo":true,"localizacao":"Alameda Barão de Limeira, 539","idCategoria":1,"idCategoriaNavigation":{"idCategoria":1,"name":"Jogos","eventos":[]}]
```

Startup.cs

```
// This method gets called by the runtime. Use this method to add services to the container.
// For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc()
        .AddJsonOptions(
            options => {
                options.SerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;
                options.SerializerSettings.NullValueHandling = Newtonsoft.Json.NullValueHandling.Ignore;
            })
        .SetCompatibilityVersion(Microsoft.AspNetCore.Mvc.CompatibilityVersion.Version_2_1);
}
```

Localhost:5000/api/eventos

► Eventos.Listar

GET localhost:5000/api/eventos

Params Authorization Headers (7) Body Pre-request Scr

Query Params

KEY
Key

Body Cookies Headers (4) Test Results


Pretty Raw Preview JSON

```
1 [
2   {
3     "idEvento": 1,
4     "titulo": "Titulo A",
5     "descricao": "Descricao A",
6     "dataEvento": "2019-06-17T14:26:46.513",
7     "ativo": true,
8     "localizacao": "Alameda Barão de Limeira, 539",
9     "idCategoria": 1,
10    "idCategoriaNavigation": {
11      "idCategoria": 1,
12      "nome": "Jogos",
13      "eventos": []
14    },
15  },
16  {
17    "idEvento": 2,
18    "titulo": "Titulo B",
19    "descricao": "Campeonato de Futebol",
20    "dataEvento": "2019-06-17T14:27:08.4",
21    "ativo": true,
22    "localizacao": "Alameda Barão de Limeira, 539",
23    "idCategoria": 3,
24    "idCategoriaNavigation": {
25      "idCategoria": 3,
26      "nome": "Futebol",
27      "eventos": []
28    },
29  }
30 ]
```


Cadastrar novo evento - POST

EventoRepository.cs

```
public void Cadastrar(Eventos evento)
{
    using (GufosContext ctx = new GufosContext())
    {
        ctx.Eventos.Add(evento);
        ctx.SaveChanges();
    }
}
```



```
[HttpPost]
public IActionResult Cadastrar(Eventos evento)
{
    try
    {
        EventoRepository.Cadastrar(evento);
        return Ok();
    }
    catch (Exception ex)
    {
        return BadRequest(new { mensagem = ex.Message });
    }
}
```



GET Eventos.Listar X POST Eventos.Cadastrar X + ...



▶ Eventos.Cadastrar

POST localhost:5000/api/eventos

Params Authorization Headers (9) Body Pre-request Script Tests

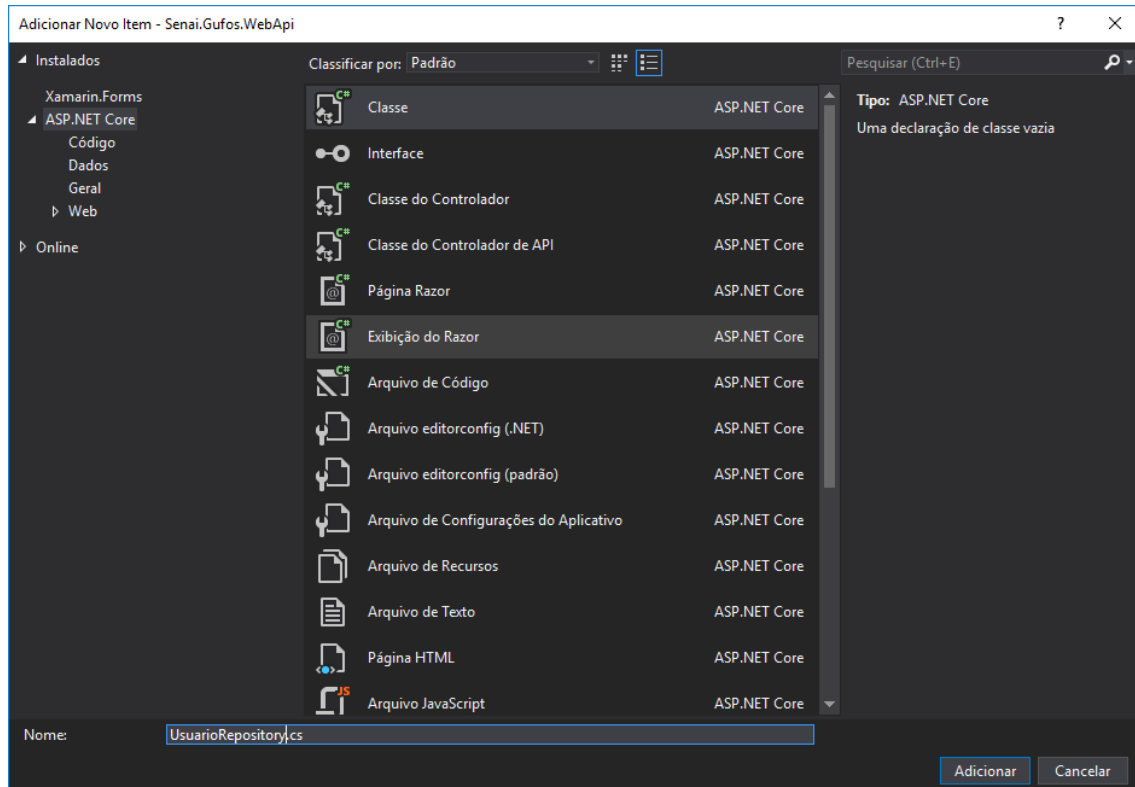
☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL BETA

```
1 {
2   "titulo": "Jogo de Ping-Pong",
3   "descricao": "Descricao Do Jogo",
4   "dataEvento": "2019-06-17T14:26:46.513",
5   "ativo": true,
6   "localizacao": "Alameda Barão de Limeira, 539",
7   "idCategoria": 1
8 }
```



USUARIOS

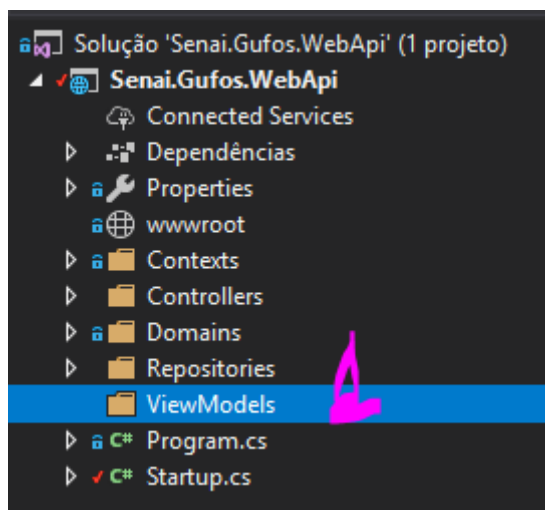
UsuarioRepository.cs

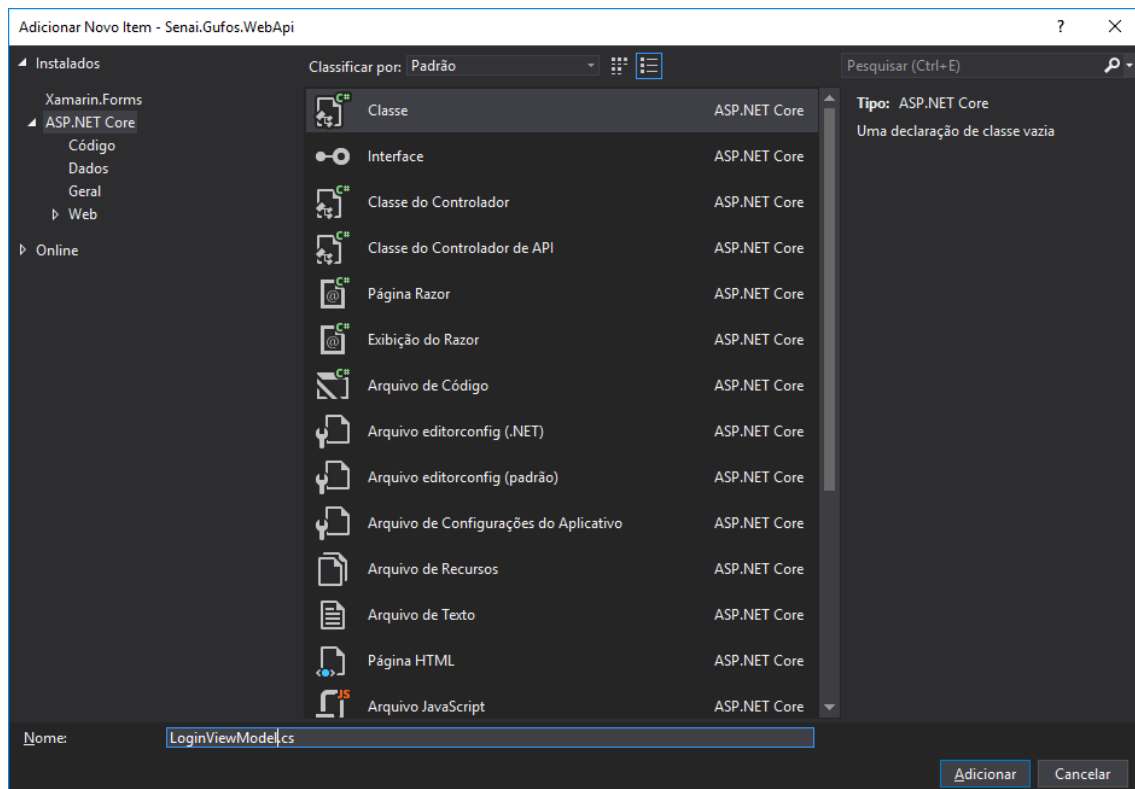


Adicionar o repositório

```
public Usuarios BuscarPorEmailESenha(LoginViewModel loginViewModel)
{
    using (GufosContext ctx = new GufosContext())
    {
        Usuarios usuarioBuscado = ctx.Usuarios.FirstOrDefault(x => x.Email == loginViewModel.Email && x.Senha == loginViewModel.Senha);
        if (usuarioBuscado == null)
            return null;
        return usuarioBuscado;
    }
}
```

Criar nova pasta chamada ViewModels e adicionar a classe LoginViewModel.cs





```
namespace Senai.Gufos.WebApi.ViewModels
{
    public class LoginViewModel
    {
        [Required]
        public string Email { get; set; }
        public string Senha { get; set; }
    }
}
```

Realizar a alteração no UsuarioRepository.cs

```
public Usuarios BuscarPorEmailESenha(LoginViewModel loginViewModel)
{
    using (GufosContext ctx = new GufosContext())
    {
        Usuarios usuarioBuscado = ctx.Usuarios.FirstOrDefault(x => x.Email == loginViewModel.Email && x.Senha == loginViewModel.Senha);
        if (usuarioBuscado == null)
            return null;
        return usuarioBuscado;
    }
}
```

Criar um novo controller chamado LoginController.cs

```
[Route("api/[controller]")]
[Produces("application/json")]
[ApiController]
public class LoginController : ControllerBase
{
    UsuarioRepository UsuarioRepository = new UsuarioRepository();

    [HttpPost]
    public IActionResult Login(LoginViewModel login)
    {
        try
        {
            Usuarios usuarioBuscado = UsuarioRepository.BuscarPorEmailESenha(login);
            if (usuarioBuscado == null)
                return NotFound(new { mensagem = "Eita, deu ruim." });
            return Ok(new { mensagem = "Usuário sucesso." });
        }
        catch (System.Exception ex)
        {
            return BadRequest(new { mensagem = ex.Message });
        }
    }
}
```

1 Usuarios.Login

2 POST localhost:5000/api/login

3 Params Authorization Headers (9) Body Pre-rec

4 none form-data x-www-form-urlencoded raw

```
1 {
2   "email" : "admin@admin.com"
3   , "senha" : "123456"
4 }
```

Body Cookies Headers (4) Test Results

Pretty Raw Preview JSON

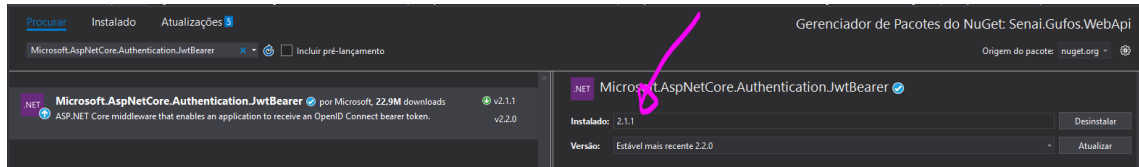
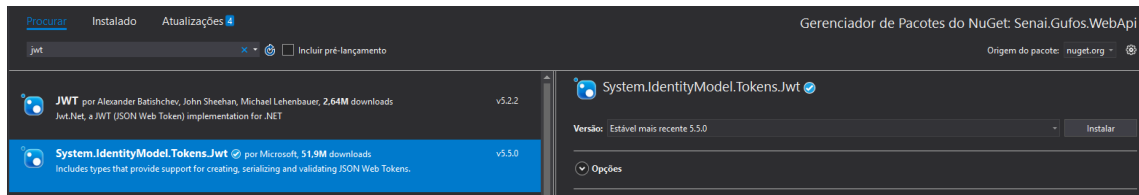
```
1 {
2   "mensagem": "Usuário sucesso."
3 }
```

JWT

Apresentação

System.IdentityModel.Tokens.Jwt – criar e validar o jwt

Microsoft.AspNetCore.Authentication.JwtBearer - Integrar a parte de autenticação



Startup.cs

```
// This method gets called by the runtime. Use this method to add services to the container.
// For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc()
        .AddJsonOptions(
            options => {
                options.SerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;
                options.SerializerSettings.NullValueHandling = Newtonsoft.Json.NullValueHandling.Ignore;
            })
        .SetCompatibilityVersion(Microsoft.AspNetCore.Mvc.CompatibilityVersion.Version_2_1);

    services.AddAuthentication(options =>
    {
        options.DefaultAuthenticateScheme = "JwtBearer";
        options.DefaultChallengeScheme = "JwtBearer";
    }).AddJwtBearer("JwtBearer", options =>
    {
        options.TokenValidationParameters = new TokenValidationParameters
        {
            ValidateIssuer = true,
            ValidateAudience = true,
            ValidateLifetime = true,
            IssuersSigningKey = new SymmetricSecurityKey(System.Text.Encoding.UTF8.GetBytes("gufos-chave-autenticacao")),
            ClockSkew = TimeSpan.FromMinutes(30),
            ValidIssuer = "Gufos.WebApi",
            ValidAudience = "Gufos.WebApi"
        };
    });
}
```

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    app.UseAuthentication();
    app.UseMvc();

    //app.Run(async (context) =>
    //{
    //    await context.Response.WriteAsync("Hello World!");
    //});
}
```


PASTE A TOKEN HERE

EDIT THE PAYLOAD AND SECRET

```
HMACSHA256(  
    base64UrlEncode(header) + "." +  
    base64UrlEncode(payload),  
    your-256-bit-secret  
) ☐ secret base64 encoded
```

Mostrar as categorias que qualquer pessoa continua conseguindo listar.

► Categorias.Listar

GET localhost:5000/api/categorias

Params Authorization Headers (7) Body Pre-request Script

Query Params

KEY
Key

Body Cookies Headers (4) Test Results

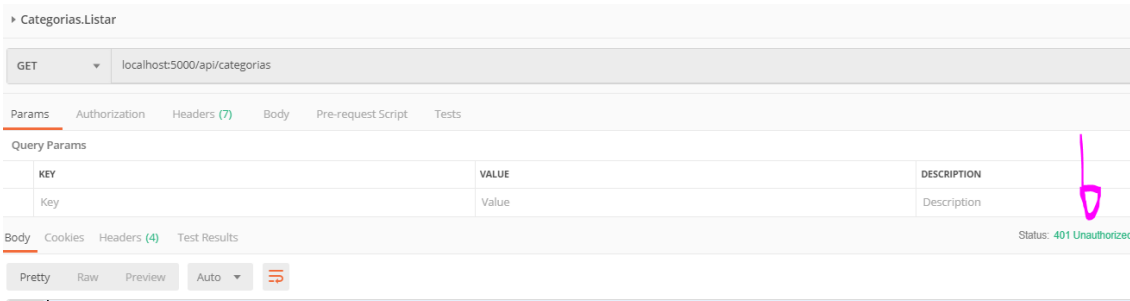
Pretty Raw Preview JSON

```
1 [
2   {
3     "idCategoria": 3,
4     "nome": "Futebol",
5     "eventos": []
6   },
7   {
8     "idCategoria": 1,
9     "nome": "Jogos",
10    "eventos": []
11  },
12  {
13    "idCategoria": 2,
14    "nome": "Meetup",
15    "eventos": []
16  },
17  {
18    "idCategoria": 4,
19    "nome": "Ping-Pong",
20    "eventos": []
21  },
22  {
23    "idCategoria": 6,
24    "nome": "Tênis de Mesa - O Retorno",
25    "eventos": []
26  }
27 ]
```

Colocar permissões nos endpoints

CategoriasController.cs

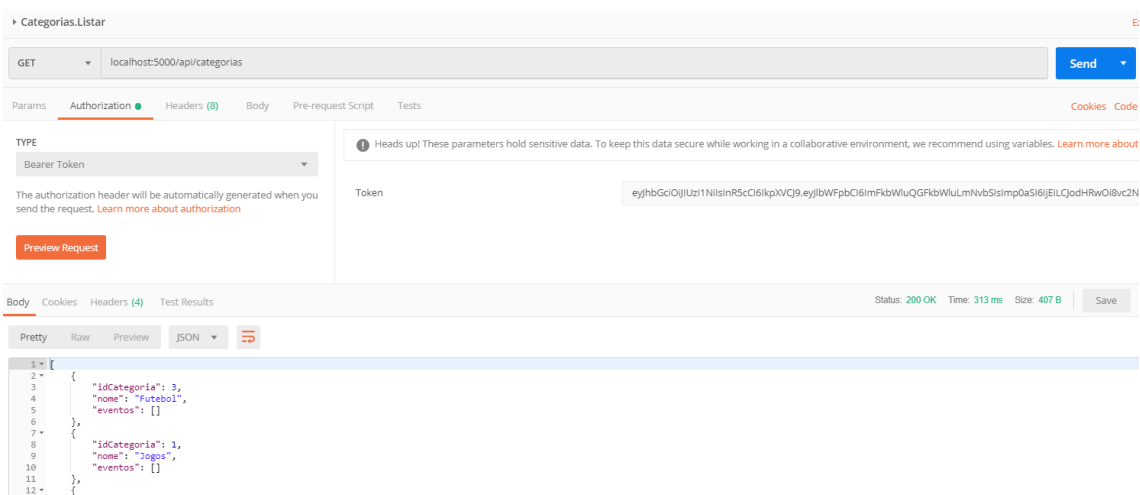

```
[Authorize]
[HttpGet]
public IActionResult ListarTodos()
{
    return Ok(CategoriaRepository.Listar());
}
```



Realizar post

localhost:5000/api/login

Inclur a Authorization – Bearer



Realizar a permissão somente para Administradores

CategoriasController.cs

```
[Authorize(Roles = "ADMINISTRADOR")]
[HttpGet("{id}")]
public IActionResult BuscarPorId(int id)
{
    Categorias Categoria = CategoriaRepository.BuscarPorId(id);
    if (Categoria == null)
    {
        return NotFound();
    }
    return Ok(Categoria);
}
```

[illegible]

► Categorías.BuscarPorId

GET localhost:5000/api/categorias/4


Params Authorization **Headers (8)** Body Pre-request Script Tests

▼ Headers (1)

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFPbCI6ImFkbWluQGFKbWUuL...	
Key	Value	Description

► Temporary Headers (7) ⓘ

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview JSON 

```

1 {
2   "idCategoria": 4,
3   "nome": "Ping-Pong",
4   "eventos": []
5 }
```

Usuarios.Login

POSTlocalhost:5000/api/login

SendSave

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTests

Examples (0)

noneform-datax-www-form-urlencodedrawbinaryGraphQLBETAJSON (application/json)

CookiesCodeComments (0)Beautify

```
1- {
2  "email": "a@auno.com"
3  , "senha": "123456"
4 }
```




Swashbuckle.AspNetCore



ConfigureServices

Configure

Localhost:5000/swagger

 swagger

Select a specGufos API V1

Gufos API

v1

[/swagger/v1/swagger.json](#)

Categorias		▼
GET	/api/Categorias	
PUT	/api/Categorias	
POST	/api/Categorias	
GET	/api/Categorias/{id}	