

Finalizando os controlers da API:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using Projeto.Application.ViewModels;
using Projeto.Application.Contracts;

namespace Projeto.Presentation.Controllers
{
    [RoutePrefix("api/Contato")] //ENDPOINT
    public class ContatoController : ApiController
    {
        //atributo
        private readonly IContatoAppService appService;

        //construtor para injeção de dependência
        public ContatoController(IContatoAppService appService)
        {
            this.appService = appService;
        }

        [HttpPost]
        public HttpResponseMessage Post(ContatoCadastroViewModel model)
        {
            if (ModelState.IsValid)
            {
                try
                {
                    appService.Cadastrar(model);
                    return Request.CreateResponse(
                        HttpStatusCode.OK, "Contato cadastrado com sucesso.");
                }
                catch (Exception e)
                {
                    return Request.CreateResponse(
                        HttpStatusCode.InternalServerError, e.Message);
                }
            }
            else
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        [HttpPut]
        public HttpResponseMessage Put(ContatoEdicaoViewModel model)
        {
            if (ModelState.IsValid)
            {
                try
                {
                    appService.Atualizar(model);
                    return Request.CreateResponse(
                        HttpStatusCode.OK, "Contato atualizado com sucesso.");
                }
            }
        }
    }
}
```

```

        catch (Exception e)
        {
            return Request.CreateResponse
                (HttpStatusCode.InternalServerError, e.Message);
        }
    }
    else
    {
        return Request.CreateResponse(HttpStatusCode.BadRequest);
    }
}

[HttpDelete]
public HttpResponseMessage Delete(int id)
{
    try
    {
        appService.Excluir(id);
        return Request.CreateResponse
            (HttpStatusCode.OK, "Contato excluído com sucesso.");
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}

[HttpGet]
public HttpResponseMessage GetAll()
{
    try
    {
        var model = appService.ConsultarTodos();
        return Request.CreateResponse(HttpStatusCode.OK, model);
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}

[HttpGet]
public HttpResponseMessage GetById(int id)
{
    try
    {
        var model = appService.ConsultarPorId(id);
        return Request.CreateResponse(HttpStatusCode.OK, model);
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}
}
}
}

```

```
using Projeto.Application.Contracts;
using Projeto.Application.ViewModels;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace Projeto.Presentation.Controllers
{
    [RoutePrefix("api/Compromisso")]
    public class CompromissoController : ApiController
    {
        //atributo
        private readonly ICompromissoAppService appService;

        //construtor para injeção de dependência
        public CompromissoController(ICompromissoAppService appService)
        {
            this.appService = appService;
        }

        [HttpPost]
        public HttpResponseMessage Post(CompromissoCadastroViewModel model)
        {
            if (ModelState.IsValid)
            {
                try
                {
                    appService.Cadastrar(model);
                    return Request.CreateResponse
                        (HttpStatusCode.OK, "Compromisso cadastrado
                                         com sucesso.");
                }
                catch (Exception e)
                {
                    return Request.CreateResponse
                        (HttpStatusCode.InternalServerError, e.Message);
                }
            }
            else
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        [HttpPut]
        public HttpResponseMessage Put(CompromissoEdicaoViewModel model)
        {
            if (ModelState.IsValid)
            {
                try
                {
                    appService.Atualizar(model);
                    return Request.CreateResponse
                        (HttpStatusCode.OK, "Compromisso atualizado
                                         com sucesso.");
                }
                catch (Exception e)
                {

```

```

        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}
else
{
    return Request.CreateResponse(HttpStatusCode.BadRequest);
}
}

[HttpDelete]
public HttpResponseMessage Delete(int id)
{
    try
    {
        appService.Excluir(id);
        return Request.CreateResponse
            (HttpStatusCode.OK, "Compromisso excluído com sucesso.");
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}

[HttpGet]
public HttpResponseMessage GetAll()
{
    try
    {
        var model = appService.ConsultarTodos();
        return Request.CreateResponse(HttpStatusCode.OK, model);
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}

[HttpGet]
public HttpResponseMessage GetById(int id)
{
    try
    {
        var model = appService.ConsultarPorId(id);
        return Request.CreateResponse(HttpStatusCode.OK, model);
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}
}
}
}

```

```
using Projeto.Application.Contracts;
using Projeto.Application.ViewModels;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace Projeto.Presentation.Controllers
{
    [RoutePrefix("api/Endereco")]
    public class EnderecoController : ApiController
    {
        //atributo
        private readonly IEnderecoAppService appService;

        //construtor para injeção de dependência
        public EnderecoController(IEnderecoAppService appService)
        {
            this.appService = appService;
        }

        [HttpPost]
        public HttpResponseMessage Post(EnderecoCadastroViewModel model)
        {
            if (ModelState.IsValid)
            {
                try
                {
                    appService.Cadastrar(model);
                    return Request.CreateResponse
                        (HttpStatusCode.OK, "Endereço cadastrado com sucesso.");
                }
                catch (Exception e)
                {
                    return Request.CreateResponse
                        (HttpStatusCode.InternalServerError, e.Message);
                }
            }
            else
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        [HttpPut]
        public HttpResponseMessage Put(EnderecoEdicaoViewModel model)
        {
            if (ModelState.IsValid)
            {
                try
                {
                    appService.Atualizar(model);
                    return Request.CreateResponse
                        (HttpStatusCode.OK, "Endereço atualizado com sucesso.");
                }
                catch (Exception e)
                {
                    return Request.CreateResponse
                        (HttpStatusCode.InternalServerError, e.Message);
                }
            }
        }
    }
}
```

```

    }
}
else
{
    return Request.CreateResponse(HttpStatusCode.BadRequest);
}
}

[HttpDelete]
public HttpResponseMessage Delete(int id)
{
    try
    {
        appService.Excluir(id);
        return Request.CreateResponse
            (HttpStatusCode.OK, "Endereço excluído com sucesso.");
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}

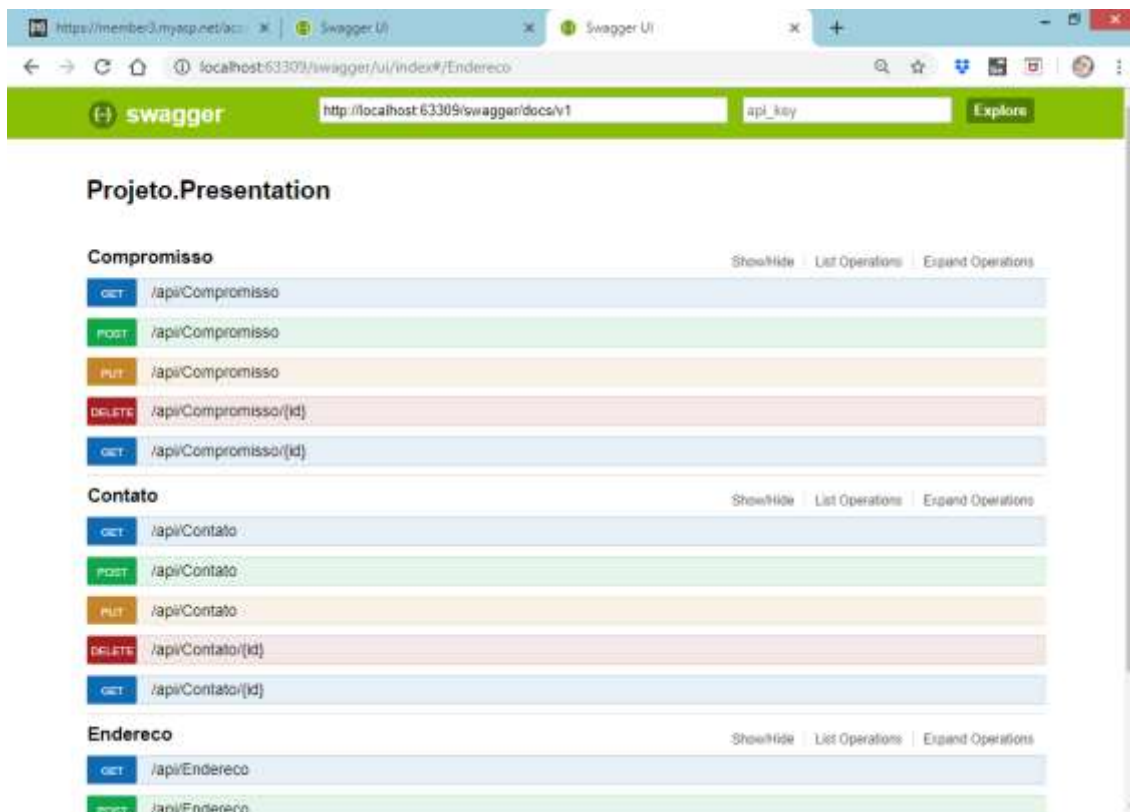
[HttpGet]
public HttpResponseMessage GetAll()
{
    try
    {
        var model = appService.ConsultarTodos();
        return Request.CreateResponse(HttpStatusCode.OK, model);
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}

[HttpGet]
public HttpResponseMessage GetById(int id)
{
    try
    {
        var model = appService.ConsultarPorId(id);
        return Request.CreateResponse(HttpStatusCode.OK, model);
    }
    catch (Exception e)
    {
        return Request.CreateResponse
            (HttpStatusCode.InternalServerError, e.Message);
    }
}
}
}

```

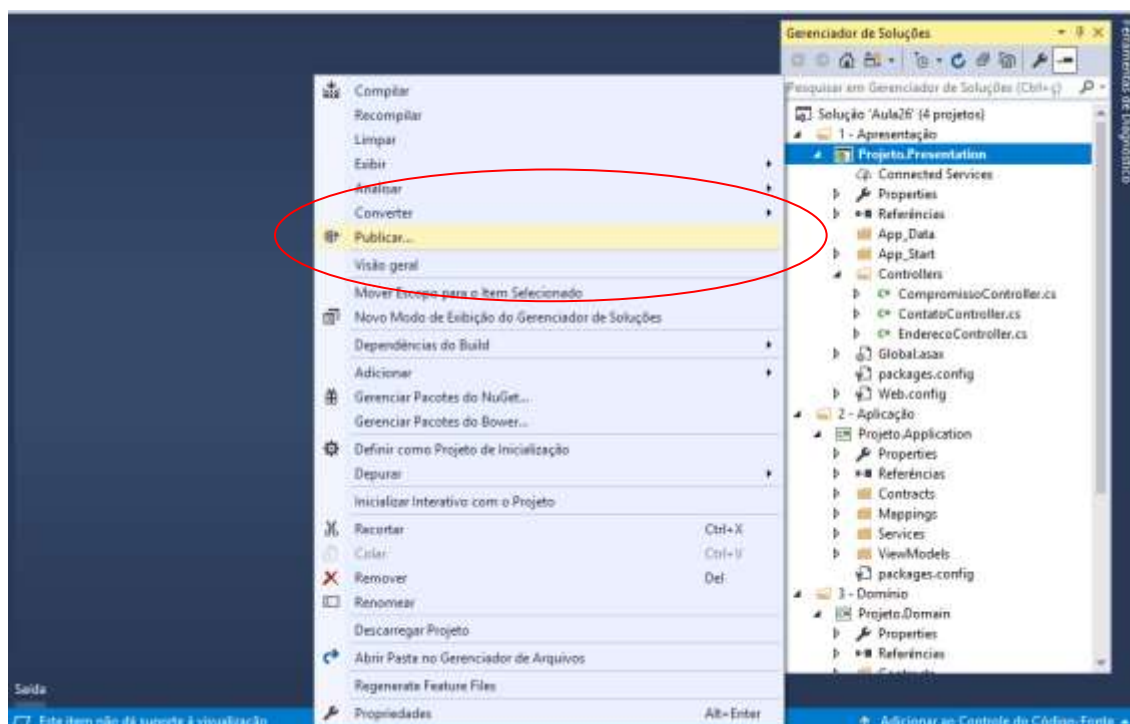
Testando a execução da API:

<http://localhost:63309/swagger/ui/index>

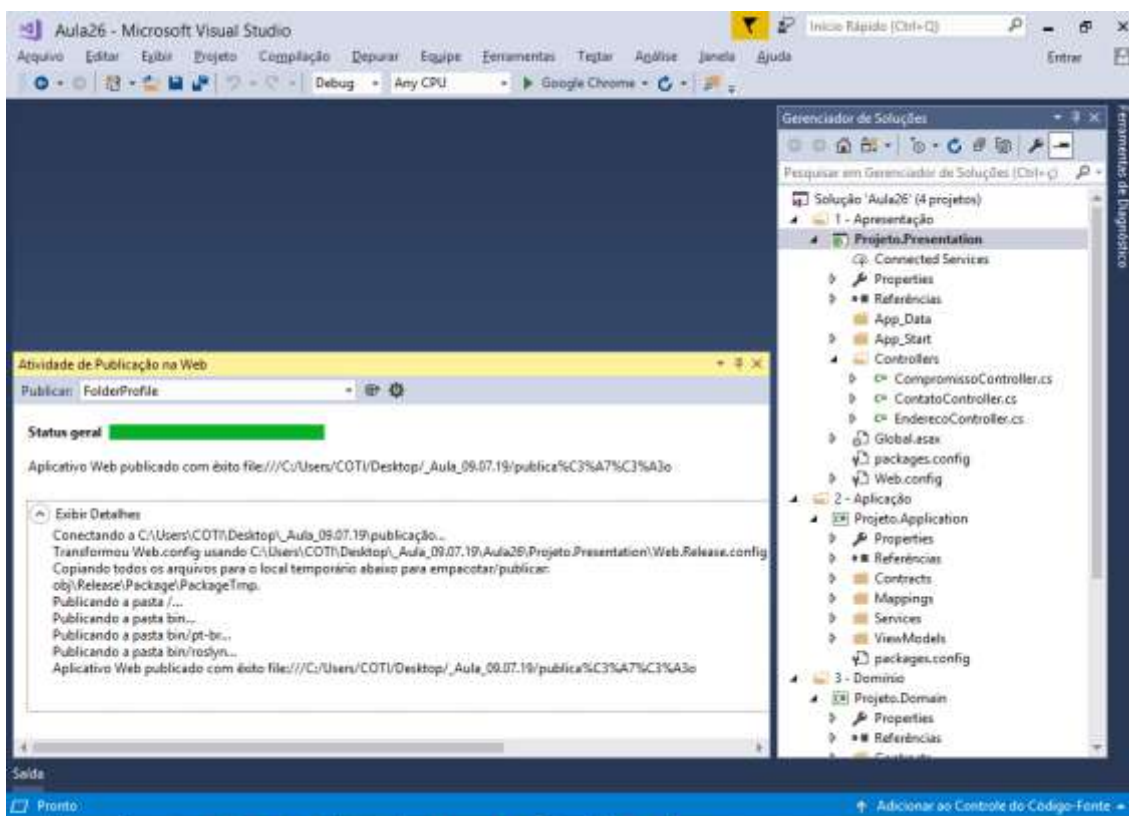
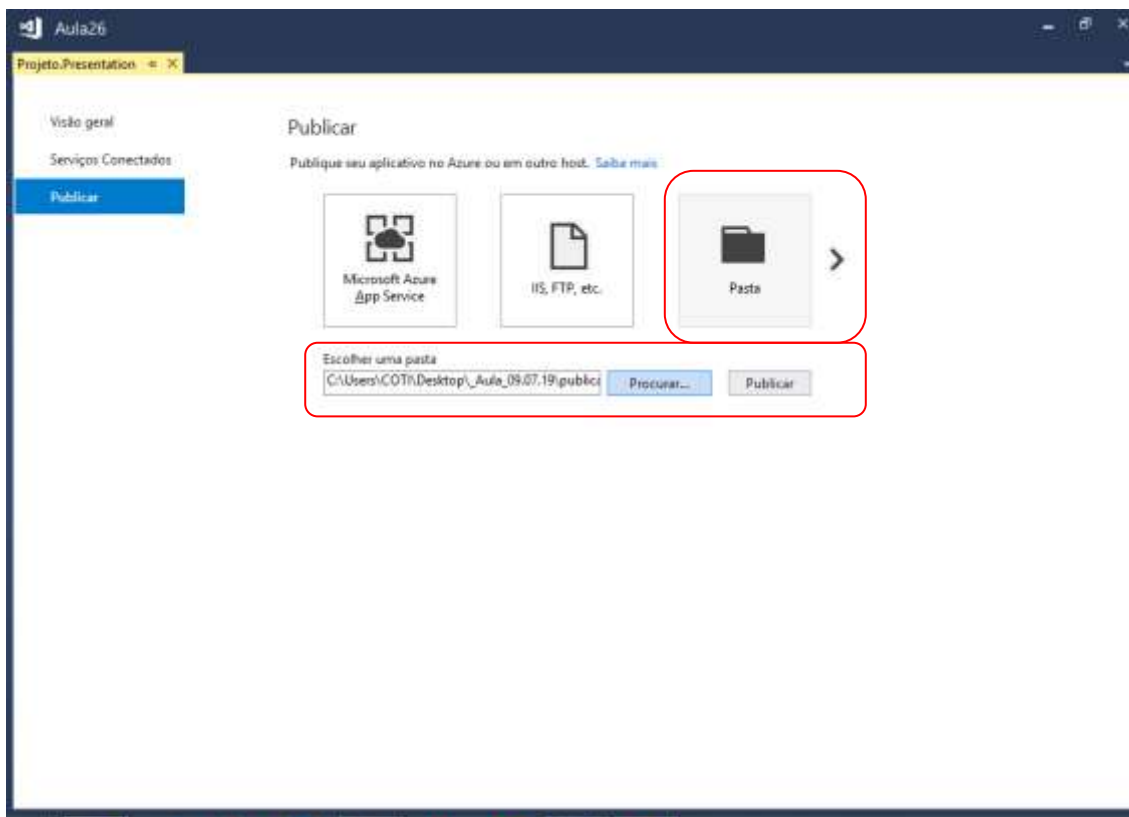


Publicando o projeto:

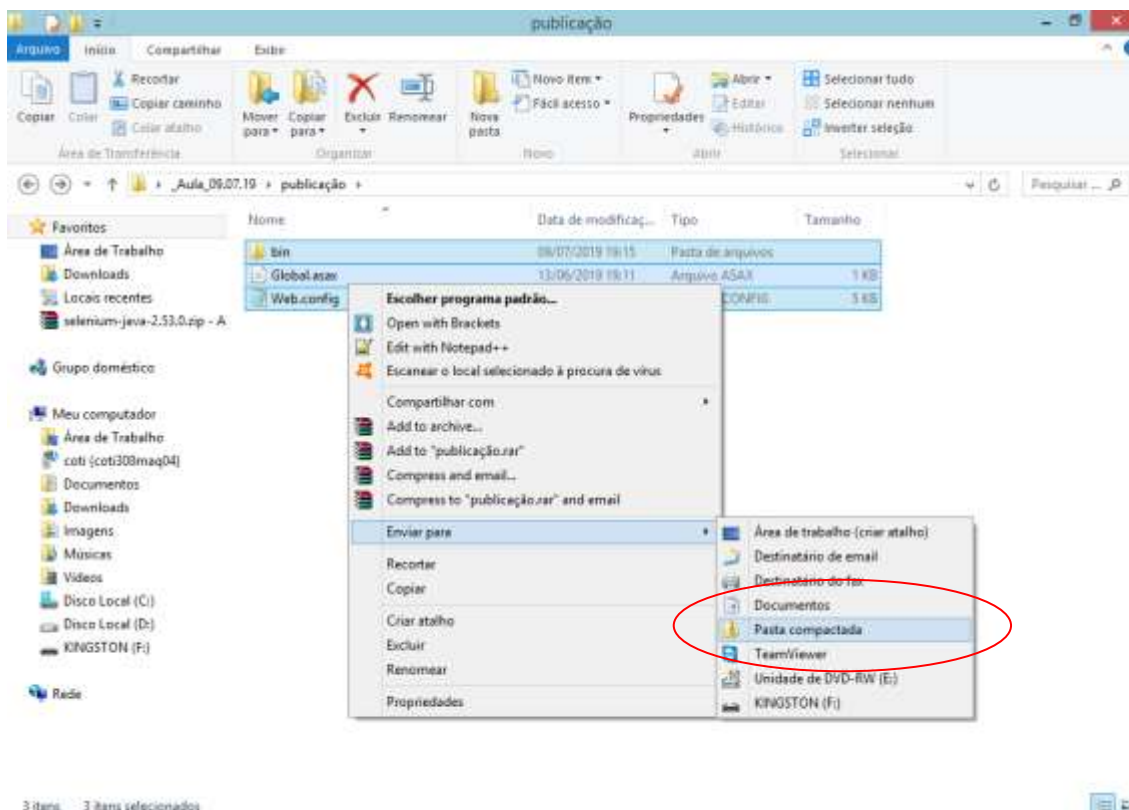
Exportando os arquivos de publicação do projeto para uma pasta local



Selecione "**Pasta**" e escolha o diretório para o qual deseja exportar os arquivos de publicação:



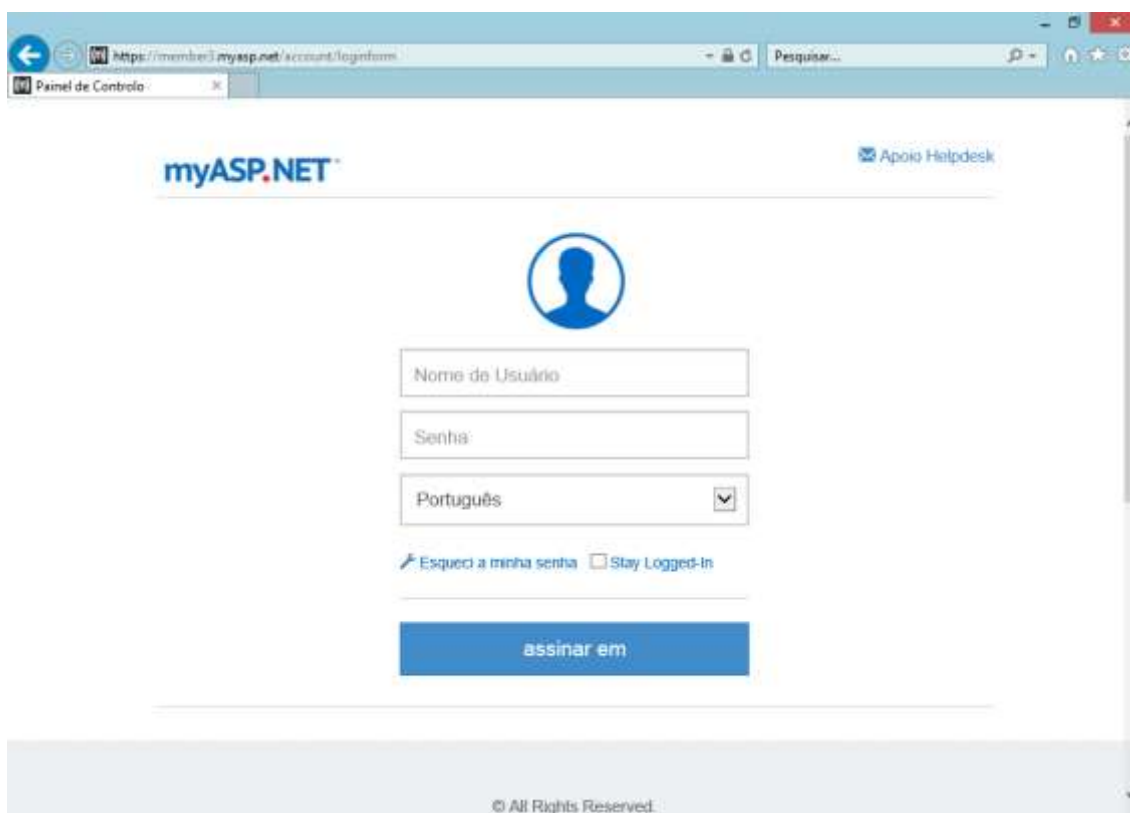
Zipando os arquivos gerados:



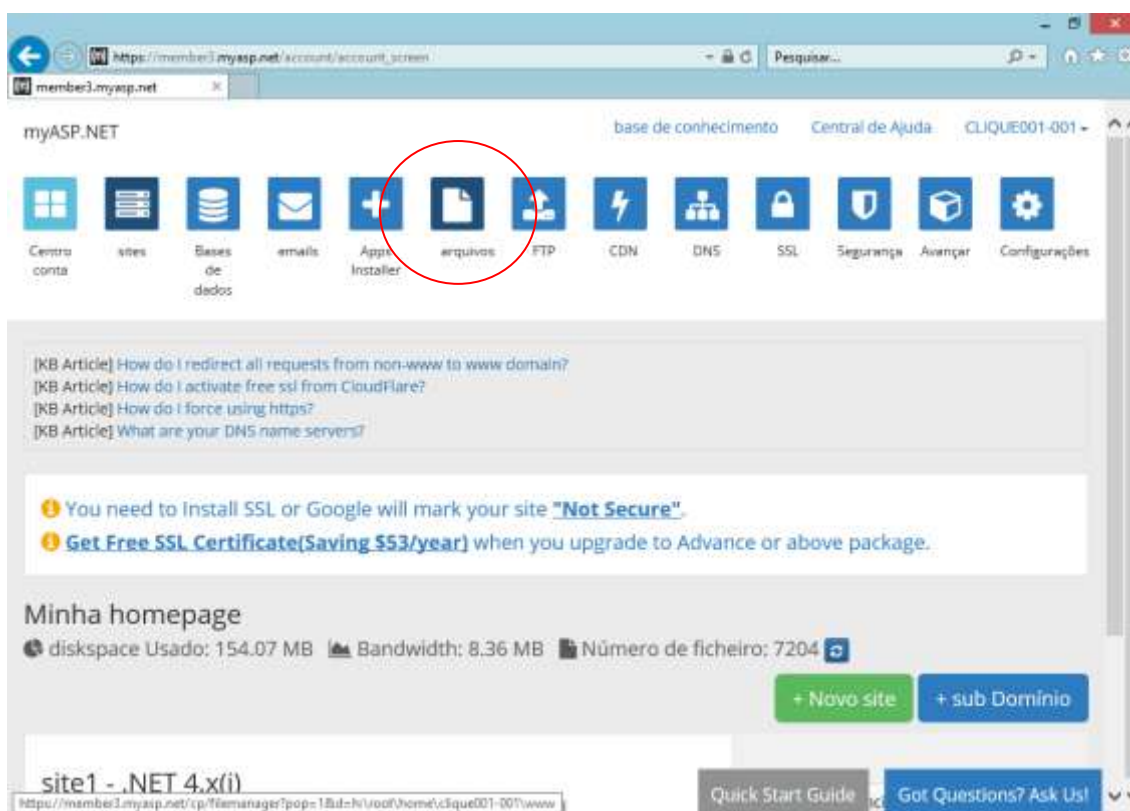
<https://www.myasp.net/>



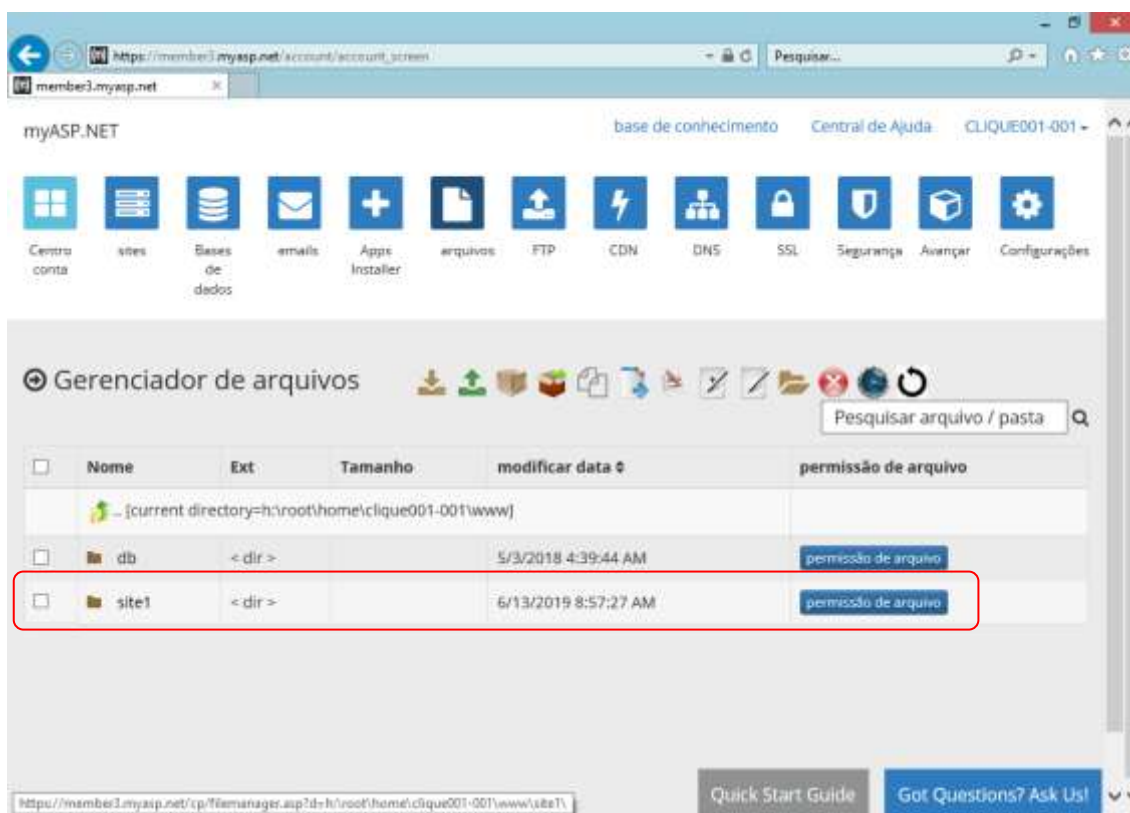
Acessando o serviço de hospedagem:



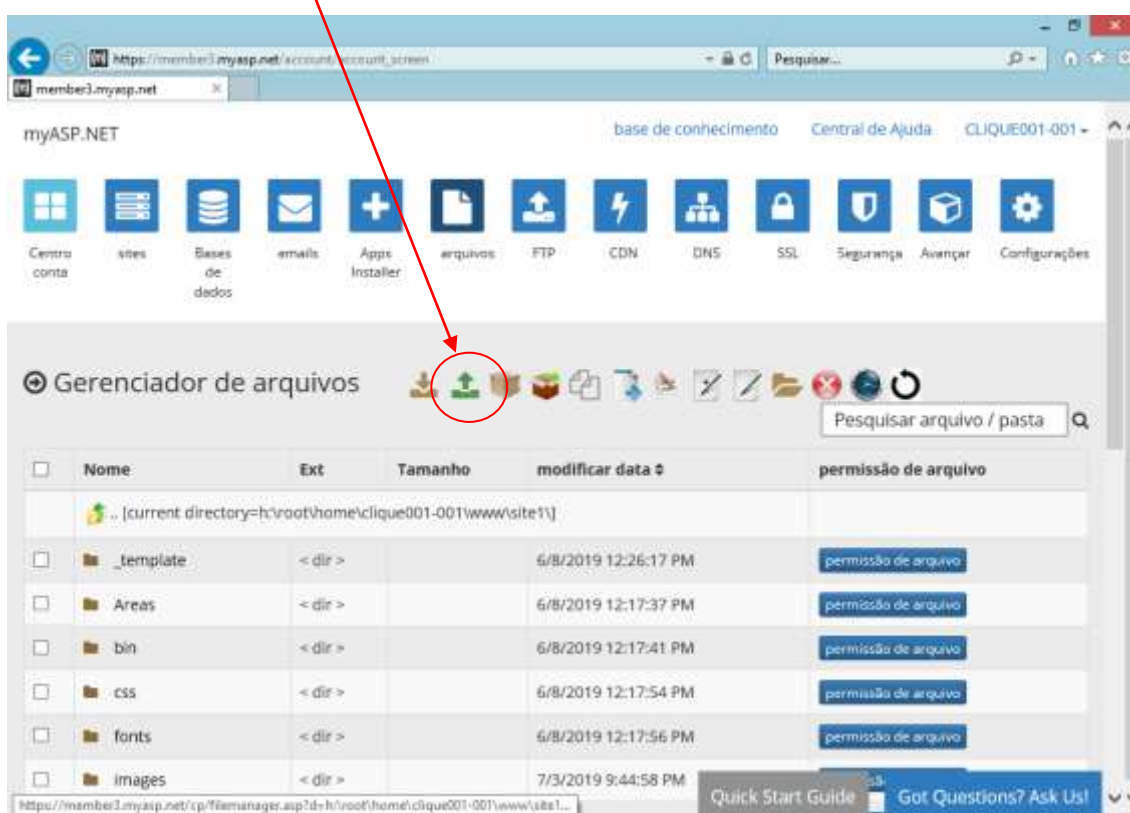
Acesse a aba "Arquivos"



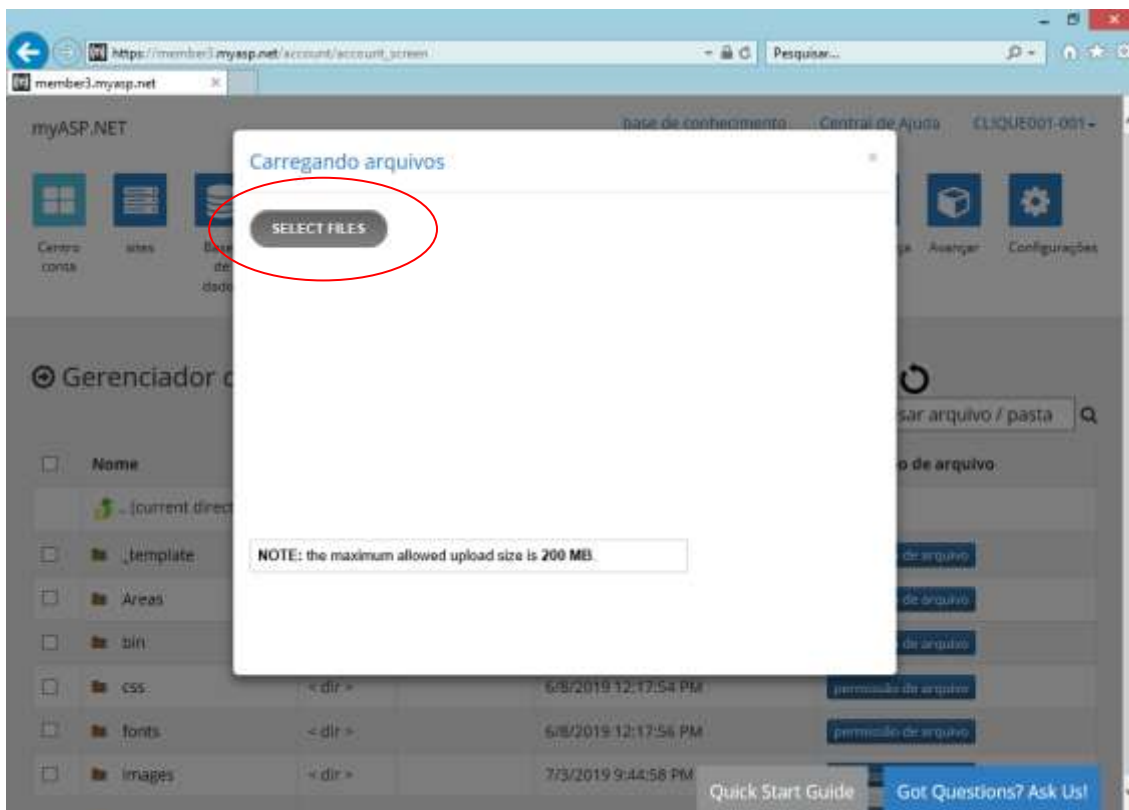
Acesse a pasta: **site1**



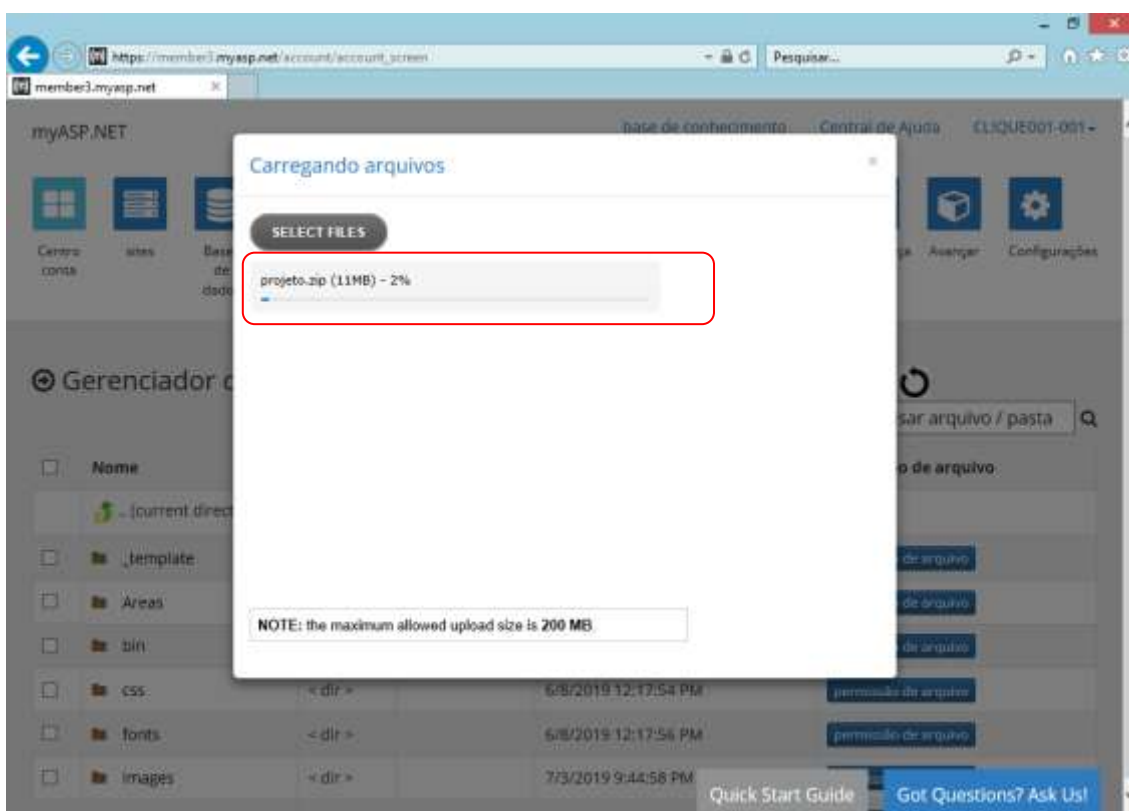
Escolha a opção **"Upload"**



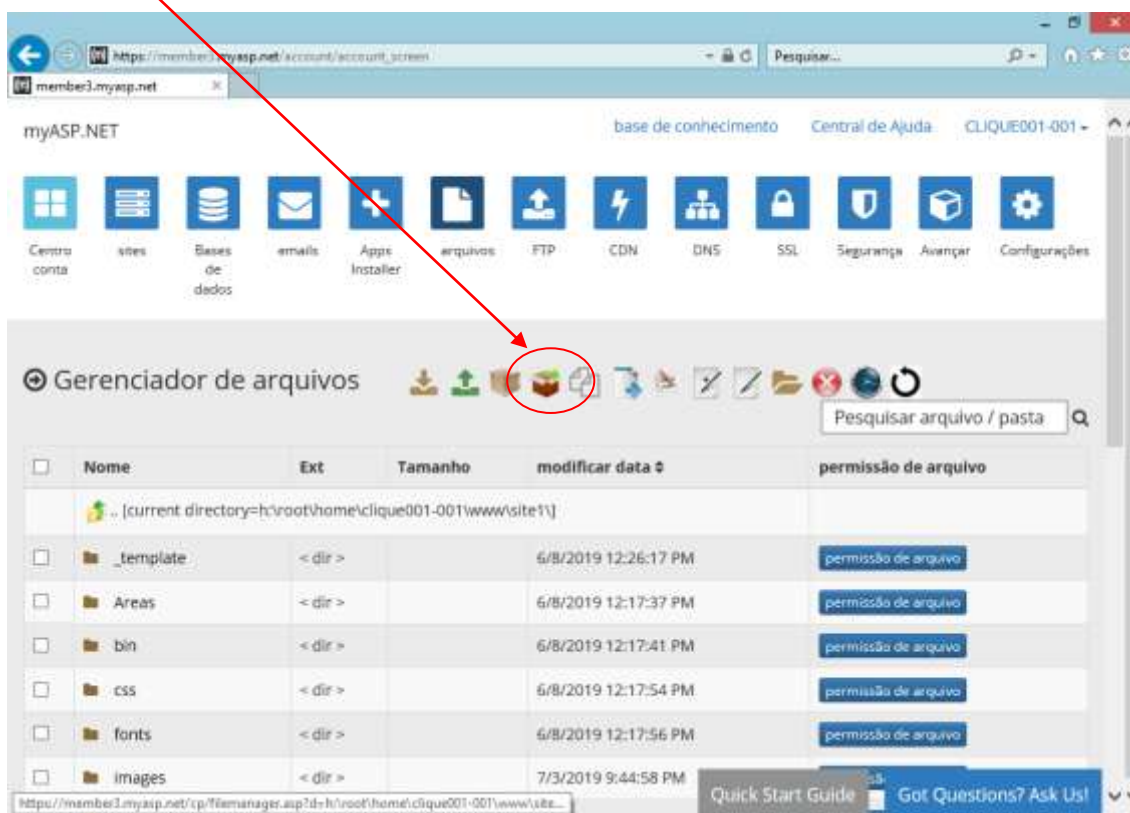
Clique em "**Selecionar Arquivos**"



Selecione e envie o arquivo .zip

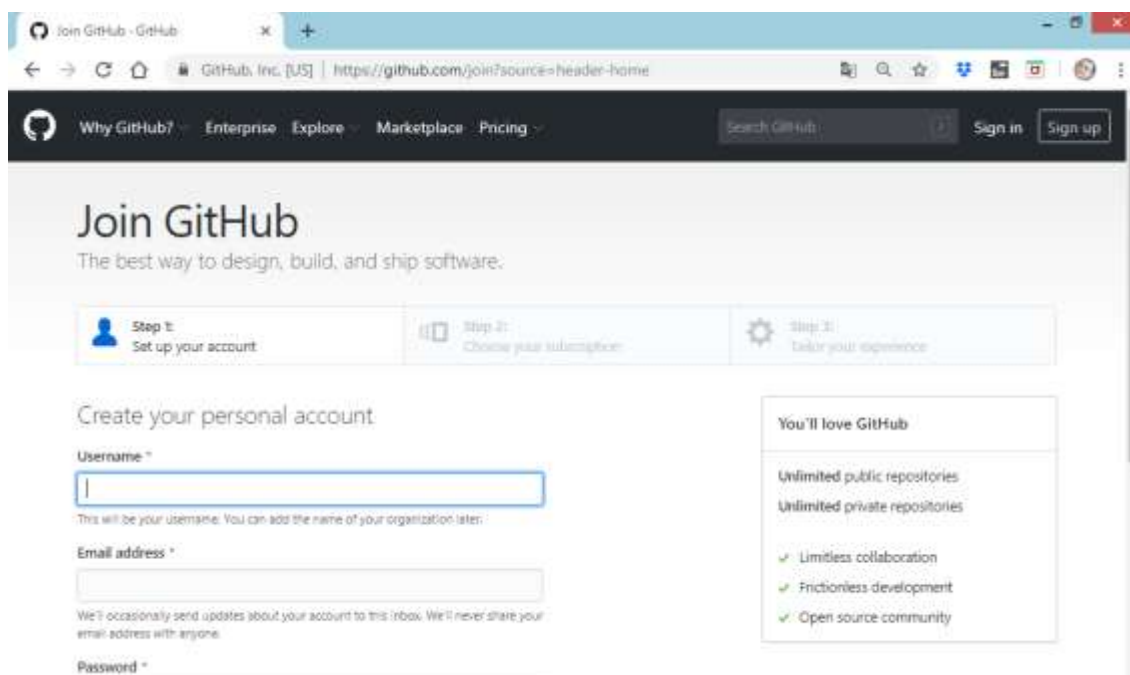


Descompacte o arquivo zip:

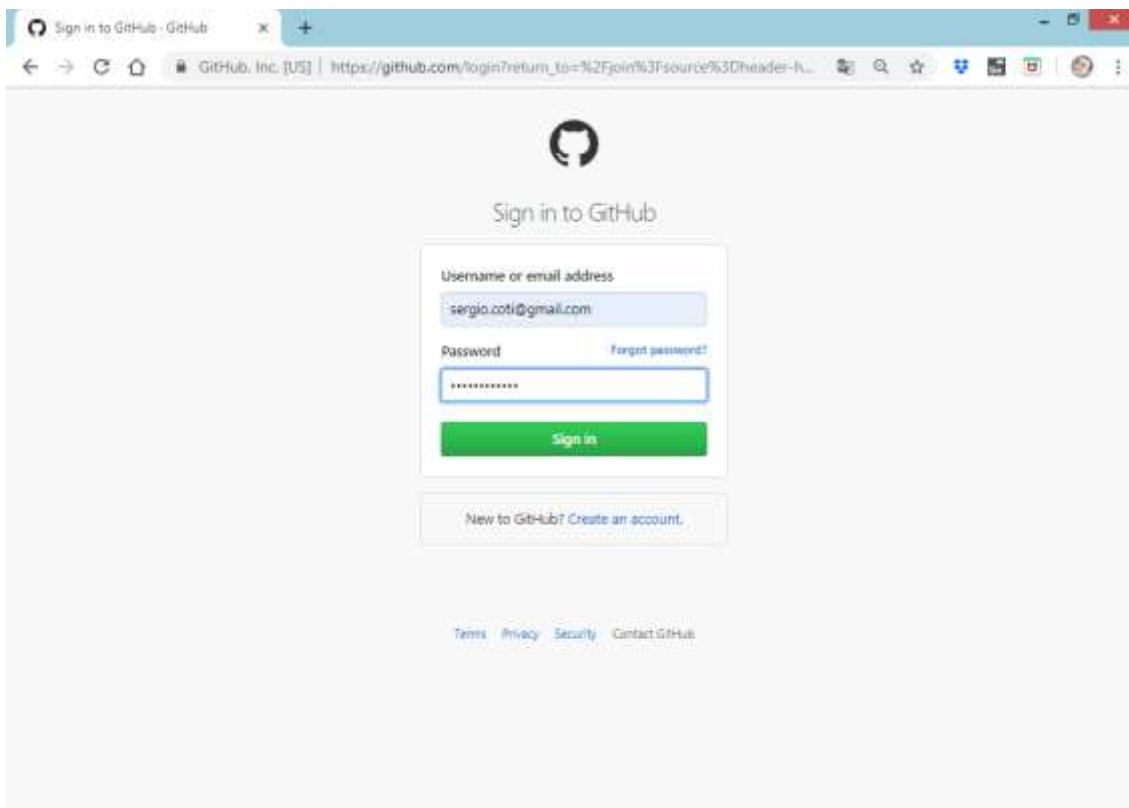


Criando uma conta no GitHub (<https://github.com/>)
Repositório de código (Portfolio de programadores)

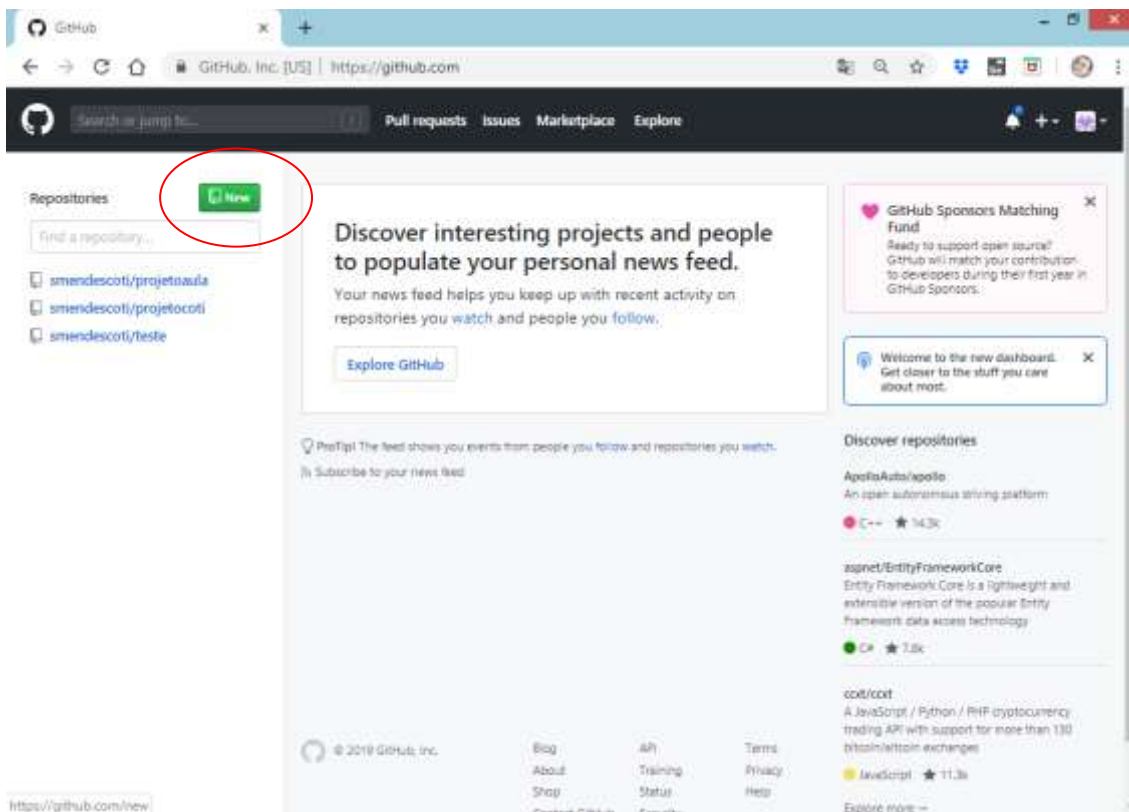
Criando uma nova conta:

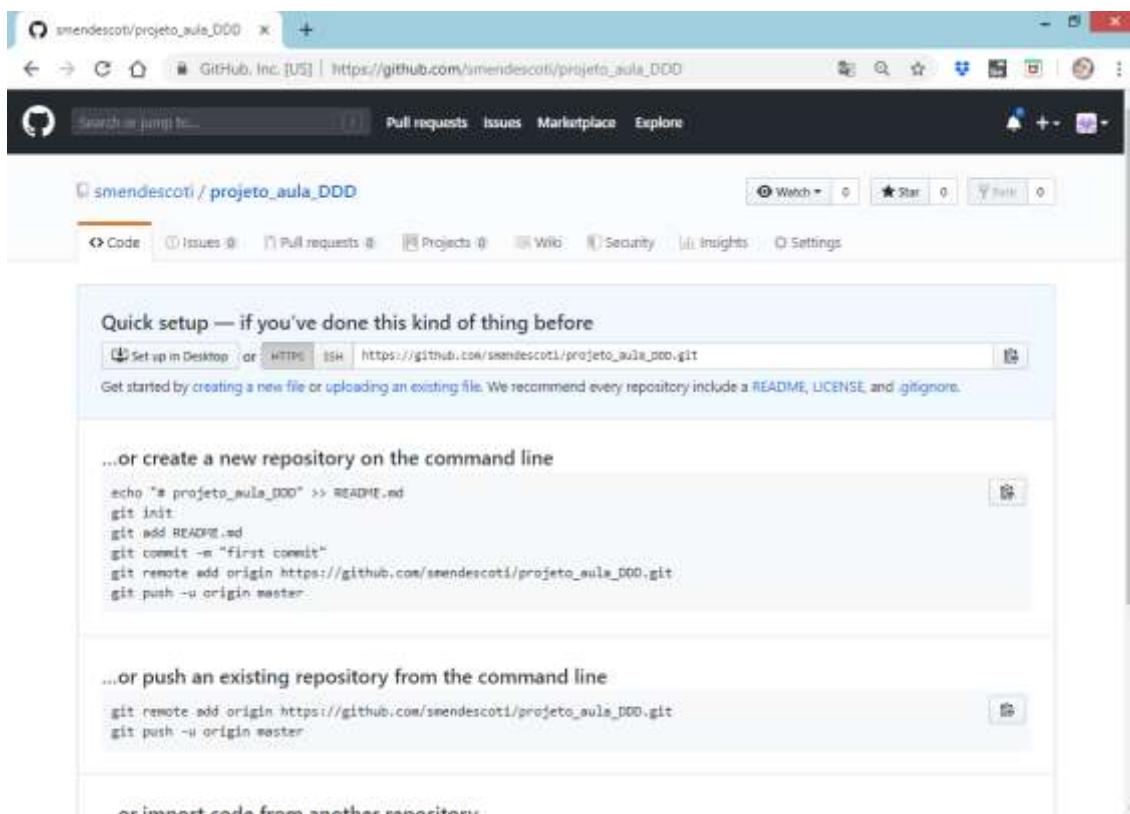
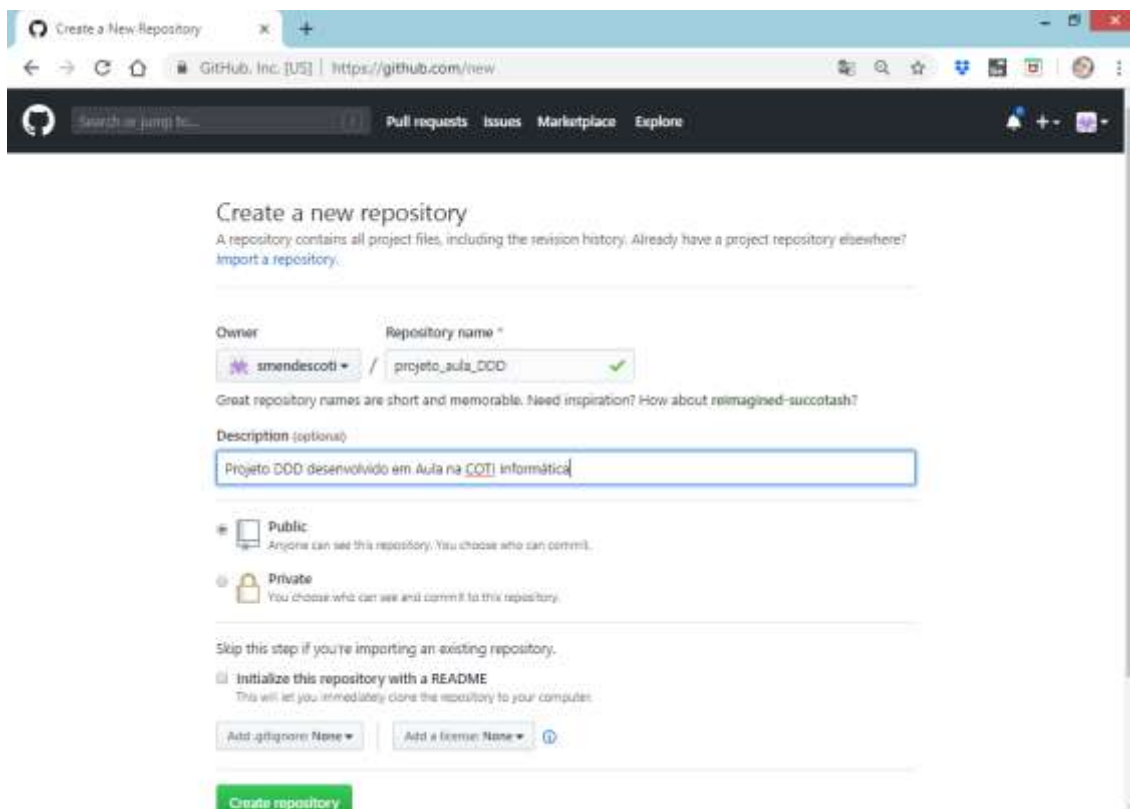


Acessando a conta:



Criando um novo repositório de projeto:



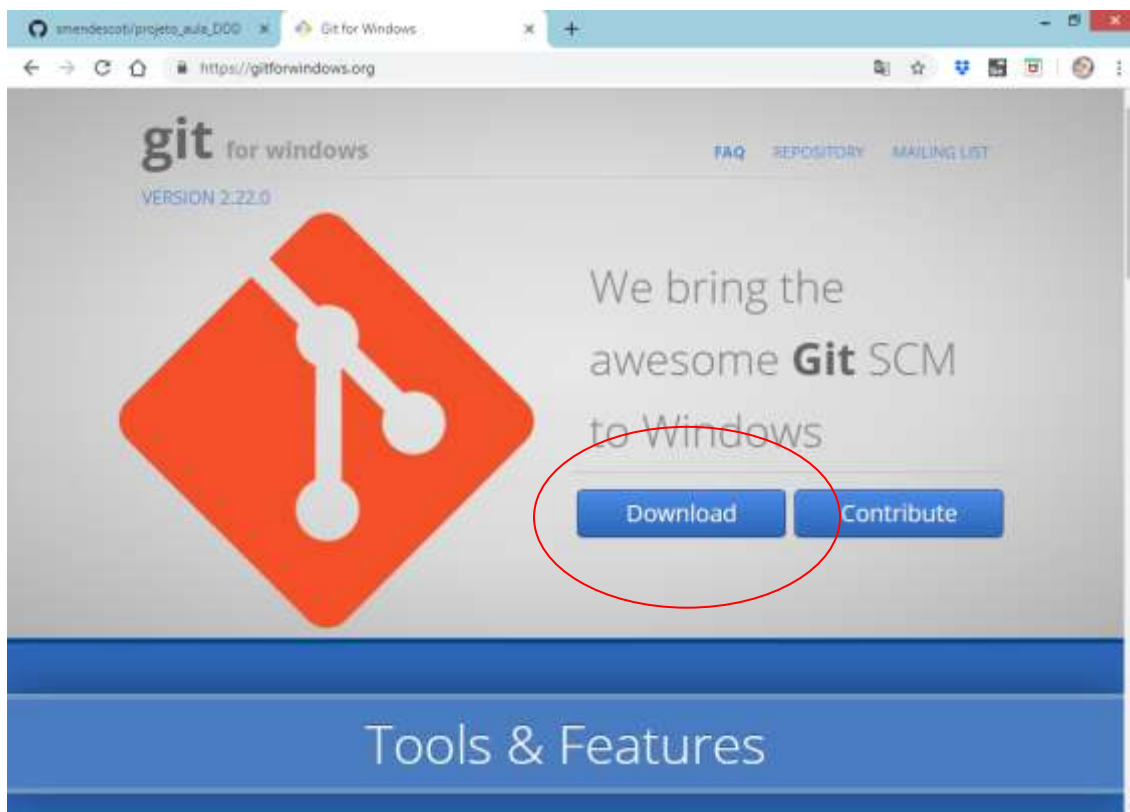


Endereço do repositório:

https://github.com/smendescoti/projeto_aula_DDD.git

Para que possamos fazer o upload do projeto para o GitHub iremos utilizar uma interface de comandos denominada **GitBash**

<https://gitforwindows.org/>



Registrando o usuário do GitHub

Prompt de comandos do GitBash

```
COTI@COTI308MAQ16 MINGW64 ~
```

```
$ cd c:
```

```
COTI@COTI308MAQ16 MINGW64 /c
```

```
$ cd Aula26
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26
```

```
$ git config --global user.name 'smendescoti'
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26
```

```
$ git config --global user.email  
'sergio.coti@gmail.com'
```


Configurando o Repositório e subindo os arquivos

Prompt de comandos do GitBash

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26
```

```
$ git init
```

```
Initialized empty Git repository in  
C:/Aula26/.git/
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
```

```
$ git add Aula26.sln
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
```

```
$ git add Projeto.Presentation
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
```

```
$ git add Projeto.Application
```

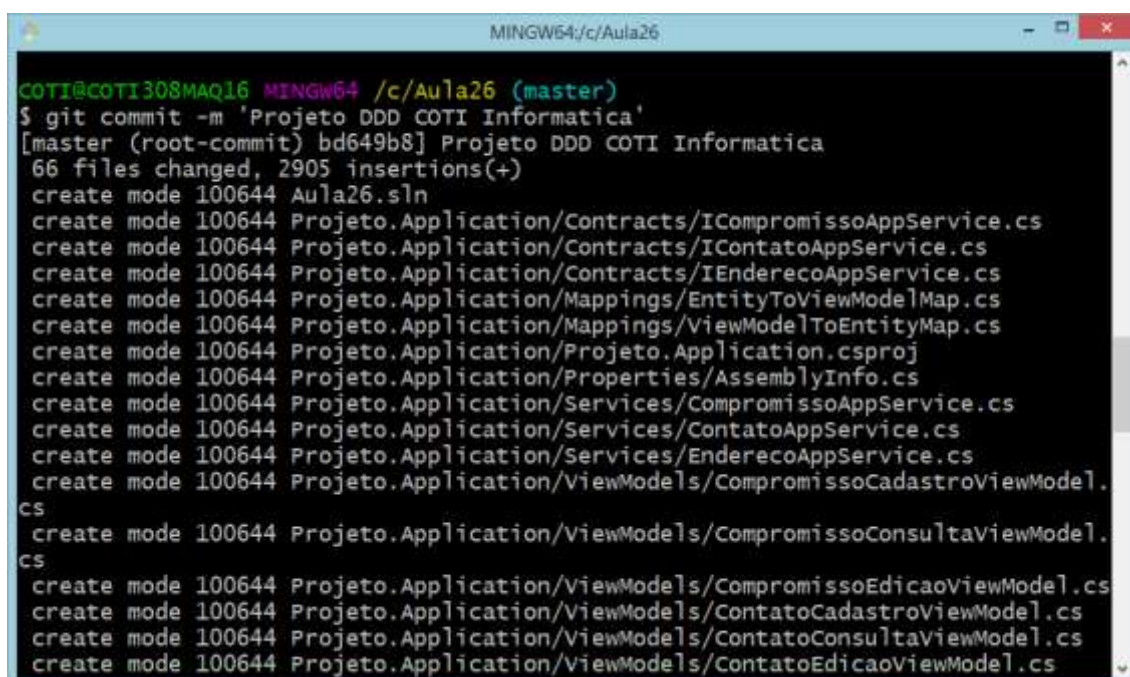
```
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
```

```
$ git add Projeto.Domain
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
```

```
$ git add Projeto.Infra.Data
```

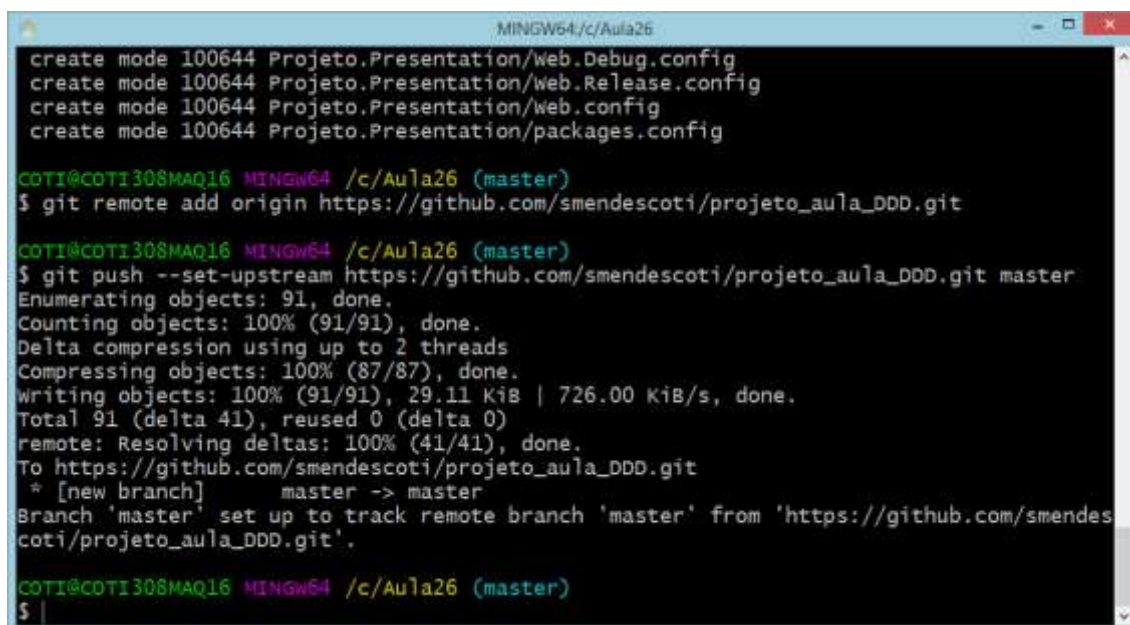
```
-----  
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)  
$ git commit -m 'Projeto DDD COTI Informatica'
```



```
MINGW64/c/Aula26  
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)  
$ git commit -m 'Projeto DDD COTI Informatica'  
[master (root-commit) bd649b8] Projeto DDD COTI Informatica  
66 files changed, 2905 insertions(+)  
create mode 100644 Aula26.sln  
create mode 100644 Projeto.Application/Contracts/ICompromissoAppService.cs  
create mode 100644 Projeto.Application/Contracts/IContatoAppService.cs  
create mode 100644 Projeto.Application/Contracts/IEnderecoAppService.cs  
create mode 100644 Projeto.Application/Mappings/EntityToViewModelMap.cs  
create mode 100644 Projeto.Application/Mappings/ViewModelToEntityMap.cs  
create mode 100644 Projeto.Application/Projeto.Application.csproj  
create mode 100644 Projeto.Application/Properties/AssemblyInfo.cs  
create mode 100644 Projeto.Application/Services/CompromissoAppService.cs  
create mode 100644 Projeto.Application/Services/ContatoAppService.cs  
create mode 100644 Projeto.Application/Services/EnderecoAppService.cs  
create mode 100644 Projeto.Application/ViewModels/CompromissoCadastroViewModel.  
cs  
create mode 100644 Projeto.Application/ViewModels/CompromissoConsultaViewModel.  
cs  
create mode 100644 Projeto.Application/ViewModels/CompromissoEdicaoViewModel.cs  
create mode 100644 Projeto.Application/ViewModels/ContatoCadastroViewModel.cs  
create mode 100644 Projeto.Application/ViewModels/ContatoConsultaViewModel.cs  
create mode 100644 Projeto.Application/ViewModels/ContatoEdicaoViewModel.cs
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
$ git remote add origin
https://github.com/smendescoti/projeto_aula_DDD.git
```

```
COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
$ git push --set-upstream
https://github.com/smendescoti/projeto_aula_DDD.git
master
```



```
MINGW64/c/Aula26
create mode 100644 Projeto.Presentation/Web.Debug.config
create mode 100644 Projeto.Presentation/Web.Release.config
create mode 100644 Projeto.Presentation/Web.config
create mode 100644 Projeto.Presentation/packages.config

COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
$ git remote add origin https://github.com/smendescoti/projeto_aula_DDD.git

COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
$ git push --set-upstream https://github.com/smendescoti/projeto_aula_DDD.git master
Enumerating objects: 91, done.
Counting objects: 100% (91/91), done.
Delta compression using up to 2 threads
Compressing objects: 100% (87/87), done.
Writing objects: 100% (91/91), 29.11 KiB | 726.00 KiB/s, done.
Total 91 (delta 41), reused 0 (delta 0)
remote: Resolving deltas: 100% (41/41), done.
To https://github.com/smendescoti/projeto_aula_DDD.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'https://github.com/smendescoti/projeto_aula_DDD.git'.

COTI@COTI308MAQ16 MINGW64 /c/Aula26 (master)
$
```

Código armazenado no GitHub:

