

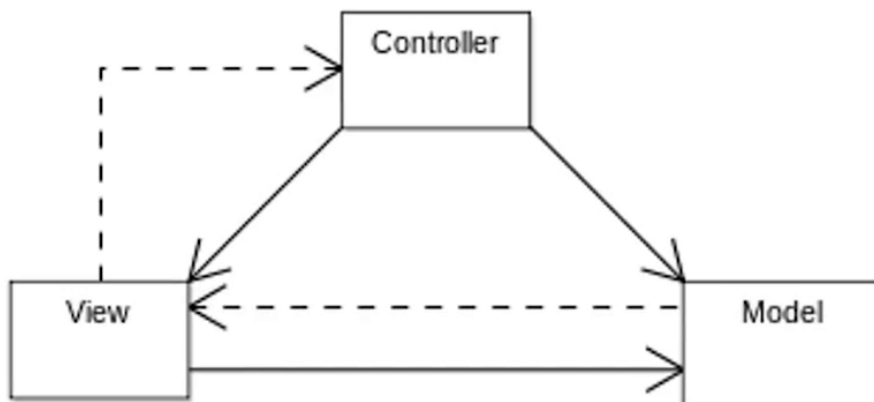
Desenvolvendo uma aplicação web MVC

MVC

O **MVC** é um padrão de arquitetura de software organizado em camadas. Um de seus principais objetivos é separar a interface das regras e do modelo de negócios, com cada camada assumindo responsabilidades no sistema.

Sua aplicação é dividida em três camadas principais:

- **Controller** – responsável pela lógica de negócios do sistema. É uma espécie de “maestro” do sistema;
- **Model** – responsável pela implementação (tipos, tamanhos, comportamentos etc.), acesso e manipulação da fonte de dados;
- **View** – responsável pela interface apresentada ao usuário.



Importante

Verifique a versão do .NET no computador, pois podem ocorrer mudanças de uma versão para outra. Para isso, abra o prompt de comando do Windows (cmd) e digite --version. Nessa atividade, estamos trabalhando com a versão 6.0.



Selecionar Prompt de Comando

```
C:\C-SHARP\MVC>dotnet --version  
6.0.306  
C:\C-SHARP\MVC>
```

Criação do projeto

1. Abra o cmd (prompt de comando do Windows) e crie um projeto MVC com o seguinte comando:

```
dotnet new mvc -o LHPet
```

2. Acesse a pasta do projeto com o comando cd, desta forma:

```
cd LHPet
```

3. Teste a aplicação para verificar se está funcionando com o comando:

```
dotnet run
```

4. Acesse o link gerado para conferir. Para isso, copie o link e cole-o no navegador de internet.

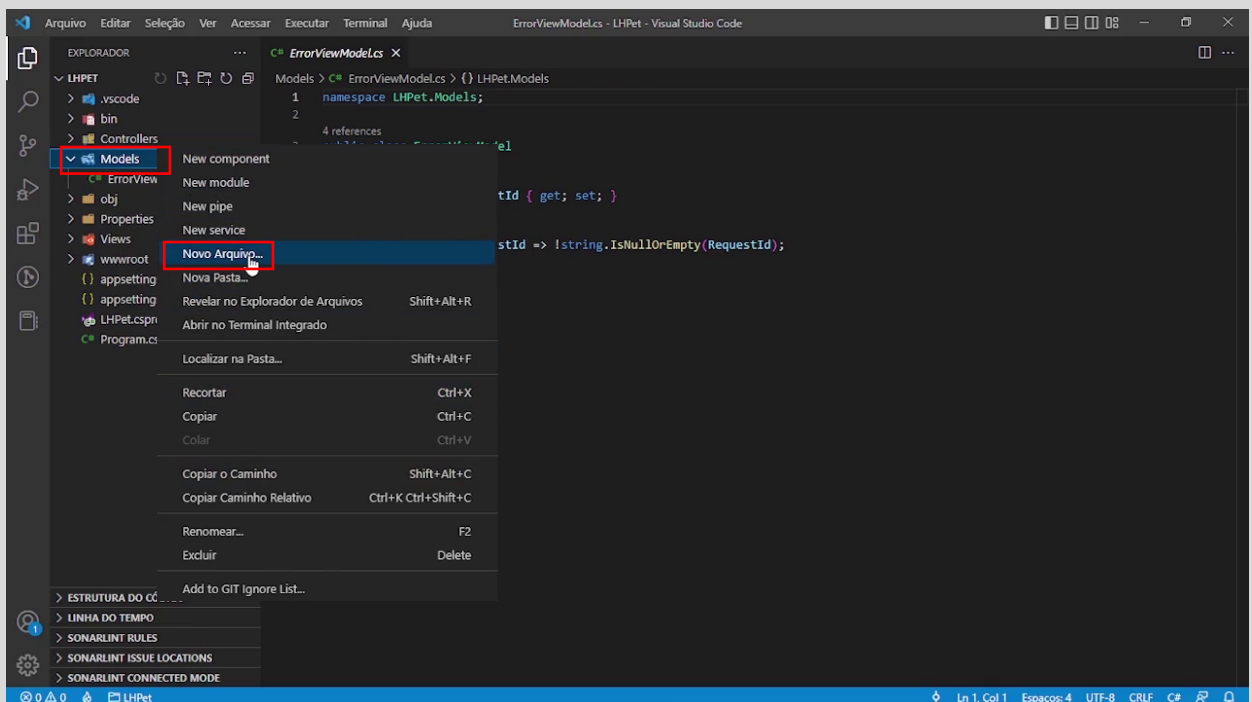
```
C:\C-SHARP\MVC\LHPet>dotnet run
Compilando...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7072
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5103
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\C-SHARP\MVC\LHPet\
```

5. Use o comando **Ctrl + C** para parar a aplicação. Abra a pasta do seu projeto no VsCode, por meio do prompt de comando, utilizando o código:

```
code .
```

Implementação dos Models

1. Após aberto o VsCode, localize a pasta **Models** na lateral esquerda. Clique com o botão direito nela e depois em **Novo Arquivo...**. Nomeie este arquivo como **Cliente.cs**.



2. No arquivo **Cliente.cs**, digite o código a seguir para incluir os atributos com getters e setters (int Id, string Nome, string Cpf, string Email e string Paciente) e o construtor da classe.

```
namespace LHPet.Models;

public class Cliente
{
    /* No trecho de código a seguir são criados os
    atributos de Cliente e seus getters e setters
    correspondentes. */
    public int Id { get; set; }
    public string Nome { get; set; }
    public string Cpf { get; set; }
    public string Email { get; set; }
    public string Paciente { get; set; }

    /* Nesse trecho do código, é criado o Construtor do
    model Cliente e determinado quais parâmetros são
    necessários na criação das instâncias. */
    public Cliente(int id, string nome, string cpf,
    string email, string paciente)
    {
        this.Id = id;
        this.Nome = nome;
        this.Cpf = cpf;
        this.Email = email;
        this.Paciente = paciente;
    }
}
```

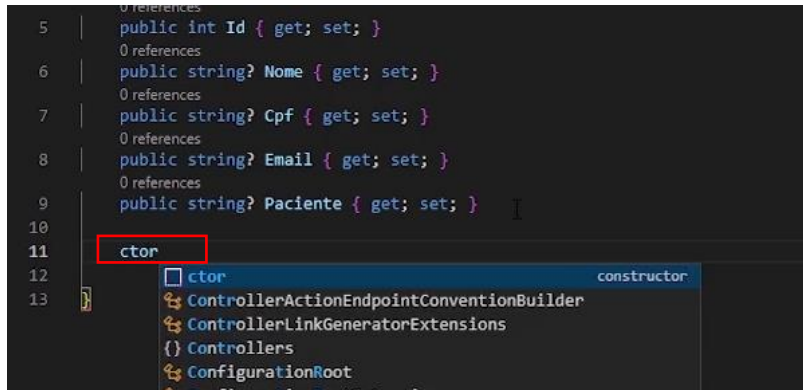
Dica!

Para iniciar o construtor, digite o comando **ctor** + **Enter**.

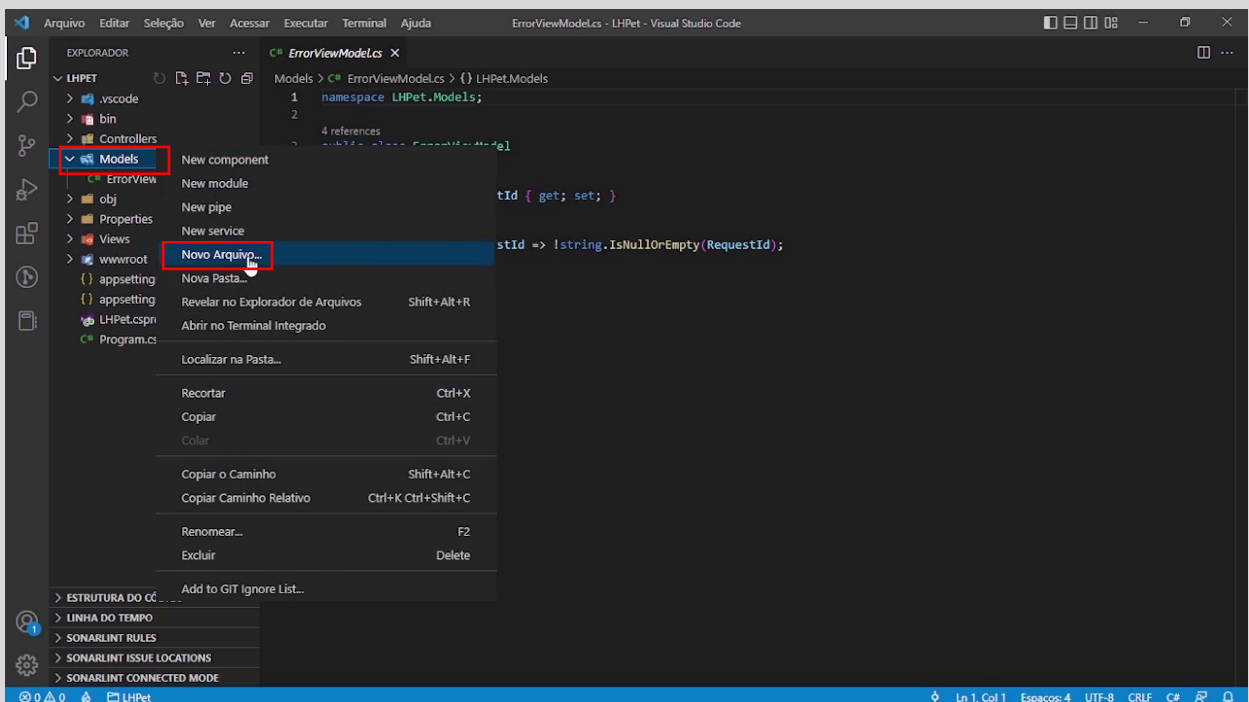
```

5 | public int Id { get; set; }
6 | public string? Nome { get; set; }
7 | public string? Cpf { get; set; }
8 | public string? Email { get; set; }
9 | public string? Paciente { get; set; }
10 |
11 | ctor
12 |
13 |

```




4. Vamos criar a classe fornecedor. Você irá novamente na pasta **Models**. Clique com o botão direito nele para criar um **Novo Arquivo** e o nomeie como **Fornecedor.cs**.



5. Vamos digitar o namespace e a classe, conforme a seguir:

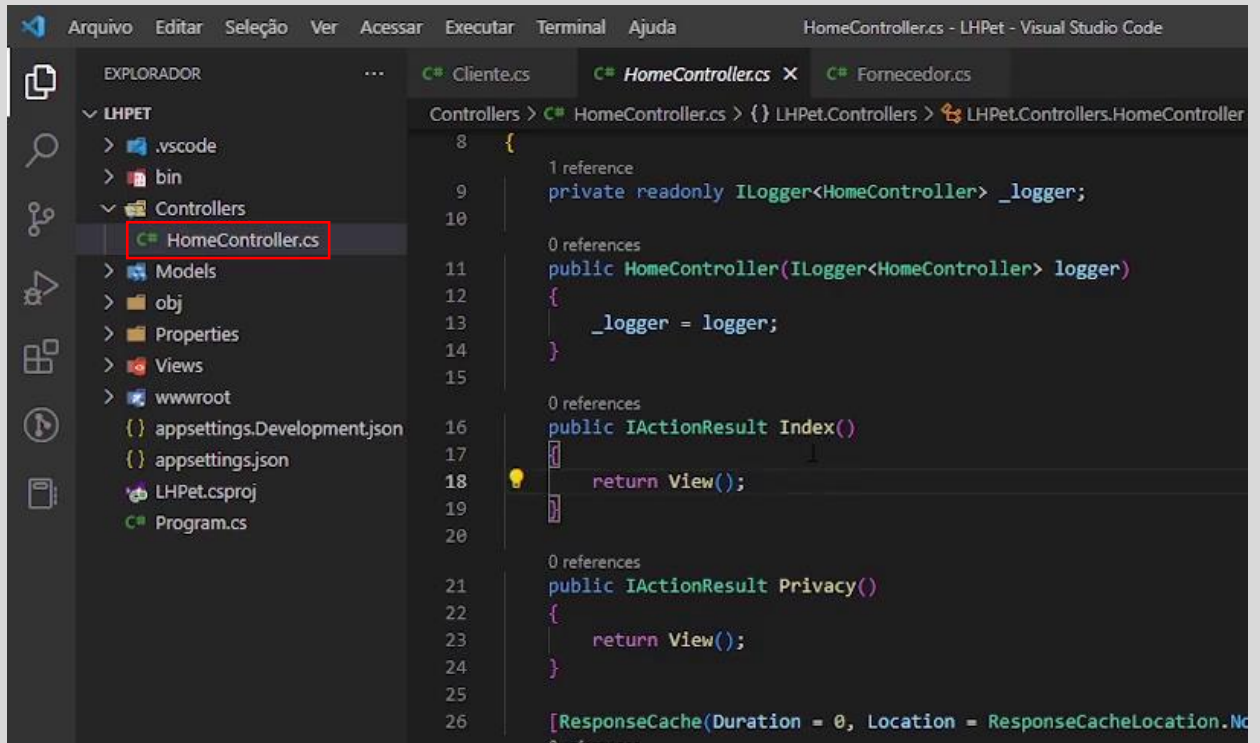
```
namespace LHPet.Models;  
  
public class Fornecedor
```

6. Digite o código a seguir para incluir os atributos com getters e setters (int Id, string Nome, string Cnpj e string Email) e o construtor da classe.

```
{  
    public int Id { get; set; }  
    public string Nome { get; set; }  
    public string Cnpj { get; set; }  
    public string Email { get; set; }  
  
    public Fornecedor(int id, string nome, string cnpj,  
string email)  
    {  
        this.Id = id;  
        this.Nome = nome;  
        this.Cnpj = cnpj;  
        this.Email = email;  
    }  
}
```

Instanciando objetos, clientes e fornecedores, adicionando em listas e enviando para a view

1. No menu à esquerda, localize a pasta **Controllers** e clique no arquivo **HomeController.cs**.



2. Com o arquivo **HomeController.cs** aberto, digite as instâncias do tipo cliente, conforme trecho destacado a seguir:

```
using System.Diagnostics;
using Microsoft.AspNetCore.Mvc;
using LHPet.Models;

namespace LHPet.Controllers;

public class HomeController : Controller
{
    private readonly ILogger<HomeController> _logger;

    public HomeController(ILogger<HomeController> logger)
    {
        _logger = logger;
    }

    public IActionResult Index()
    {
        Cliente cliente1 = new Cliente(01, "Arthur A.
Ferreira", "857.032.950-41", "arthur.antunes@sp.senai.br",
"Madruga");
        Cliente cliente2 = new Cliente(02, "William Henry
Gates III", "039.618.250-09", "bill@microsoft.com", "Bug");
        Cliente cliente3 = new Cliente(03, "Ada Lovelace",
"800.777.920-50", "ada@ada.language.com", "Byron");
        Cliente cliente4 = new Cliente(04, "Linus
Torvalds", "933.622.400-03", "torvalds@osdl.org",
"Pinguim");
        Cliente cliente5 = new Cliente(05, "Grace Hopper",
"911.702.988-00", "grace.hopper@cobol.com", "Loboc");
    }
}
```

Importante

Nesse trecho do código foram criadas as instâncias de clientes, respeitando os parâmetros solicitados no construtor que está no model Cliente.



3. Agora vamos criar as instâncias de fornecedores. Digite o código a seguir.

```
Fornecedor fornecedor1 = new Fornecedor(01, "C#  
PET S/A", "14.182.102/0001-80", "c-sharp@pet.org");  
  
Fornecedor fornecedor2 = new Fornecedor(02, "Ctrl  
Alt Dog", "15.836.698/0001-57", "ctrl@alt.dog.br");  
  
Fornecedor fornecedor3 = new Fornecedor(03,  
"BootsPet INC", "40.810.224/0001-83",  
"boots.pet@gatomania.us");  
  
Fornecedor fornecedor4 = new Fornecedor(04, "Tik  
Tok Dogs", "87.945.350/0001-09",  
"noisnamidia@tiktokdogs.uk");  
  
Fornecedor fornecedor5 = new Fornecedor(05,  
"Bifinho Forever", "18.760.614/0001-37",  
"contato@bff.us");
```

4. É necessário criar a lista **listaClientes** e adicionar os Clientes. Para isso, copie o código a seguir logo após as instâncias do tipo cliente.

```
List<Cliente> listaClientes = new List<Cliente>();  
listaClientes.Add(cliente1);  
listaClientes.Add(cliente2);  
listaClientes.Add(cliente3);  
listaClientes.Add(cliente4);  
listaClientes.Add(cliente5);
```

5. Vamos criar uma lista chamada **listaFornecedores**, para os objetos fornecedores, e adicionar todos os fornecedores criados anteriormente nessa lista. Digite o código a seguir, logo após as instâncias do tipo fornecedor.

```
List<Fornecedor> listaFornecedores = new  
List<Fornecedor>();  
listaFornecedores.Add(fornecedor1);  
listaFornecedores.Add(fornecedor2);  
listaFornecedores.Add(fornecedor3);  
listaFornecedores.Add(fornecedor4);  
listaFornecedores.Add(fornecedor5);
```

6. Para enviar essas listas para as views, utilize o código a seguir, logo após a lista de clientes:

```
ViewBag.listaClientes = listaClientes;
```

7. Digite o código da **listaFornecedores** para que ela seja usada na View que vai exibir os dados.

```
ViewBag.listaFornecedores = listaFornecedores;
```

8. Insira o código a seguir para retornar para a view com todas as informações que enviamos.

```
return View();
```

9. Confira a seguir todo o código digitado nessa parte:

```
public IActionResult Index()
{
    Cliente cliente1 = new Cliente(01, "Arthur A. Ferreira", "857.032.950-41",
    "arthur.antunes@sp.senai.br", "Madruga");
    Cliente cliente2 = new Cliente(02, "William Henry Gates III",
    "039.618.250-09", "bill@microsoft.com", "Bug");
    Cliente cliente3 = new Cliente(03, "Ada Lovelace", "800.777.920-50",
    "ada@ada.language.com", "Byron");
    Cliente cliente4 = new Cliente(04, "Linus Torvalds", "933.622.400-03",
    "torvalds@osdl.org", "Pinguim");
    Cliente cliente5 = new Cliente(05, "Grace Hopper", "911.702.988-00",
    "grace.hopper@cobol.com", "Loboc");

    List<Cliente> listaClientes = new List<Cliente>();
    listaClientes.Add(cliente1);
    listaClientes.Add(cliente2);
    listaClientes.Add(cliente3);
    listaClientes.Add(cliente4);
    listaClientes.Add(cliente5);

    ViewBag.listaClientes = listaClientes;

    Fornecedor fornecedor1 = new Fornecedor(01, "C# PET S/A",
    "14.182.102/0001-80", "c-sharp@pet.org");
    Fornecedor fornecedor2 = new Fornecedor(02, "Ctrl Alt Dog",
    "15.836.698/0001-57", "ctrl@alt.dog.br");
    Fornecedor fornecedor3 = new Fornecedor(03, "BootsPet INC",
    "40.810.224/0001-83", "boots.pet@gatomania.us");
    Fornecedor fornecedor4 = new Fornecedor(04, "Tik Tok Dogs",
    "87.945.350/0001-09", "noisnamidia@tiktokdogs.uk");
    Fornecedor fornecedor5 = new Fornecedor(05, "Bifinho Forever",
    "18.760.614/0001-37", "contato@bff.us");

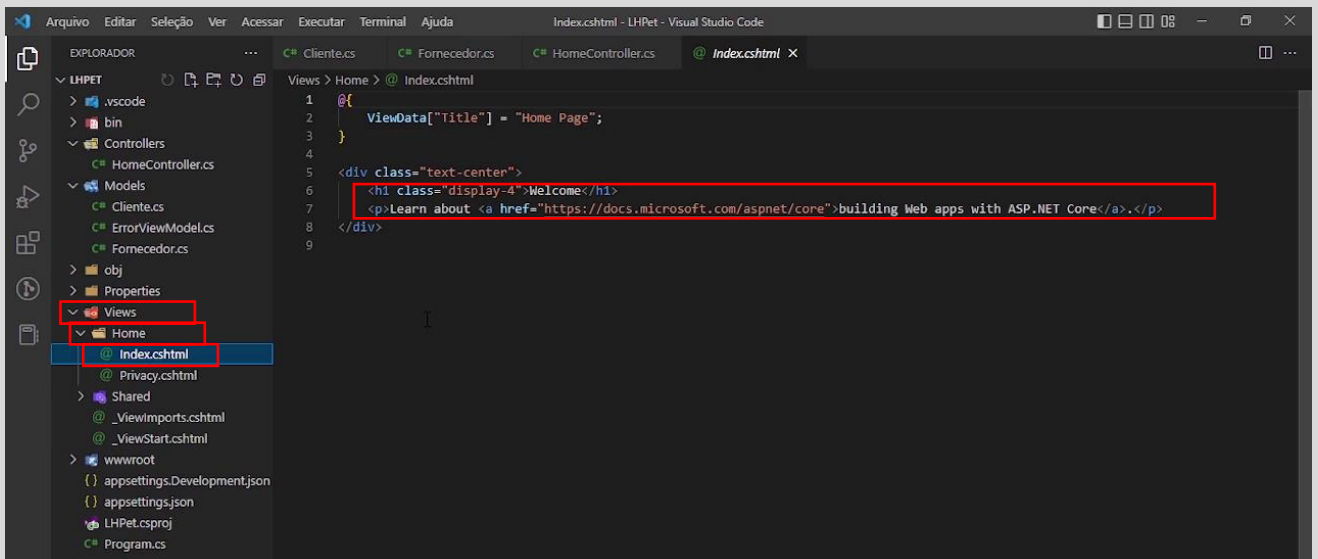
    List<Fornecedor> listaFornecedores = new List<Fornecedor>();
    listaFornecedores.Add(fornecedor1);
    listaFornecedores.Add(fornecedor2);
    listaFornecedores.Add(fornecedor3);
    listaFornecedores.Add(fornecedor4);
    listaFornecedores.Add(fornecedor5);

    ViewBag.listaFornecedores = listaFornecedores;

    return View();
}
```

Implementando e exibindo as listas de clientes e fornecedores na View.

1. No menu à esquerda, localize a pasta **Views > Home**, abra o arquivo **index.cshtml** e exclua o trecho destacado na imagem.



Dica!

Para conferir o resultado de cada etapa, digite no Terminal o comando **dotnet run** e acesse o link exibido para testar a aplicação. Use o comando **Ctrl + C** para parar a aplicação.



2. Digite o código a seguir dentro da tag div. Isso fará com que as listas Cliente e Fornecedores sejam percorridas e seus atributos exibidos.

```
@foreach (var item in @ViewBag.listaClientes)
{
    <tr>
        <th scope="row">@item.Id</th>
        <td>@item.Nome</td>
        <td>@item.Cpf</td>
        <td>@item.Email</td>
        <td>@item.Paciente</td>
    </tr>
}

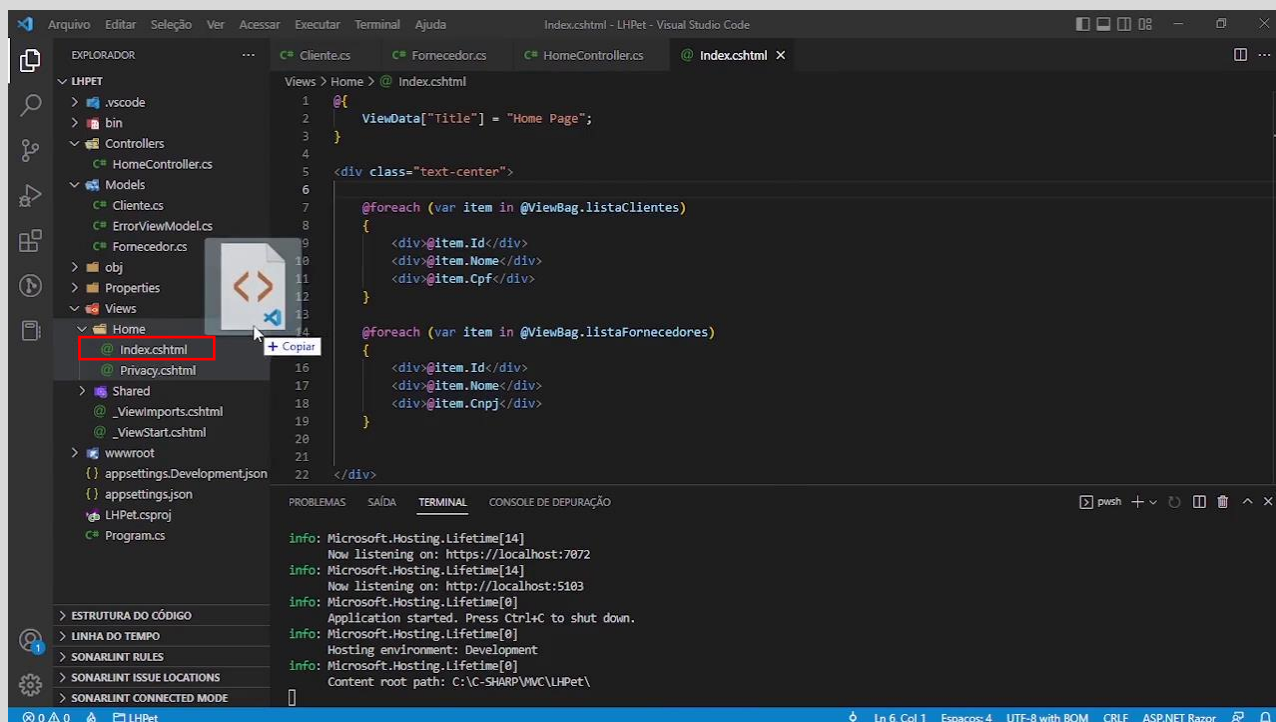
@foreach (var item in @ViewBag.listaFornecedores)
{
    <tr>
        <th scope="row">@item.Id</th>
        <td>@item.Nome</td>
        <td>@item.Cnpj</td>
        <td>@item.Email</td>
    </tr>
}
```

Importante

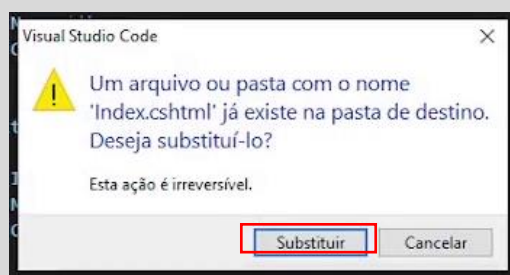
Vamos utilizar o arquivo ***index.cshtml***, que foi disponibilizado na atividade, e contém um template do bootstrap com a página para exibir as listas. Baixe a pasta .zip e extraia dentro da atividade no local indicado.



3. Localize o arquivo citado anteriormente, **index.cshtml**, nos arquivos da atividade, baixe, clique nele e arraste-o para o VSCode na árvore de arquivos, na linha indicada a seguir.



4. Na caixa de diálogo, clique em **Substituir** para confirmar a troca de arquivo.



5. O código será substituído. Nesse novo código, localize a **Lista de clientes** e digite o trecho de código destacado a seguir:

```
<table class="table table-hover table-sm table-bordered table-sm caption-top">
  <caption>Lista de clientes</caption>
  <thead class="table-dark">
    <tr>
      <th scope="col">Id</th>
      <th scope="col">Nome</th>
      <th scope="col">CPF</th>
      <th scope="col">E-mail</th>
      <th scope="col">Paciente (pet)</th>
    </tr>
  </thead>
  <tbody>
    1 <@foreach (var item in @ViewBag.listaClientes)
      {
        <tr>
          2 <th scope="row">@item.Id</th>
            <td>@item.Nome</td>
            <td>@item.Cpf</td>
            <td>@item.Email</td>
            <td>@item.Paciente</td>
          </tr>
        3 }
      </tbody>
    </table>
```

6. Agora editaremos o trecho **Lista Fornecedores**. Digite o trecho de código destacado a seguir:

```
<table class="table table-hover table-sm table-bordered table-sm caption-top">
  <caption>Lista de fornecedores</caption>
  <thead class="table-dark">
    <tr>
      <th scope="col">Id</th>
      <th scope="col">Nome</th>
      <th scope="col">CNPJ</th>
      <th scope="col">E-mail</th>
    </tr>
  </thead>
  <tbody>
    1  <@foreach (var item in @ViewBag.listaFornecedores)
      {
        <tr>
          2  <th scope="row">@item.Id</th>
            <td>@item.Nome</td>
            <td>@item.Cnpj</td>
            <td>@item.Email</td>
          </tr>
        3  }
      </tbody>
    </table>
```