

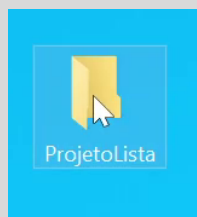
Projeto em listas

Iniciando o projeto listas

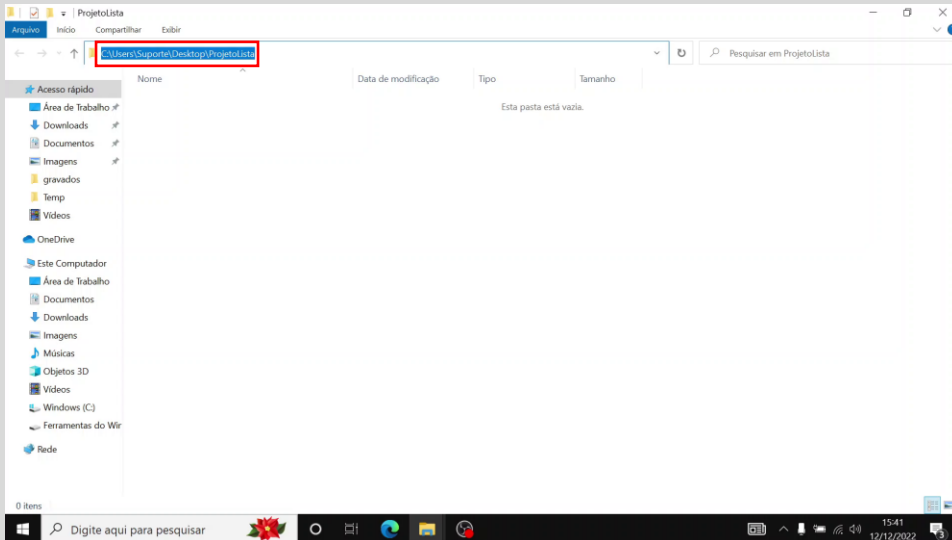
Uma das principais funções da lista é que ela interage com diversos elementos, sem limites de inserções, com o objetivo de realizar impressão, cálculo ou algum outro tipo de operação dentro do programa.

Siga o conteúdo para iniciar um projeto em listas.

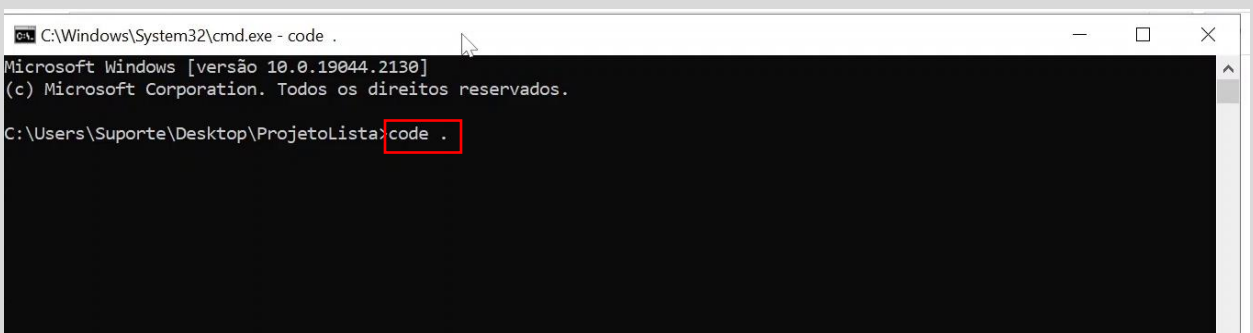
1. Crie uma pasta e nomeie de **ProjetoLista**.



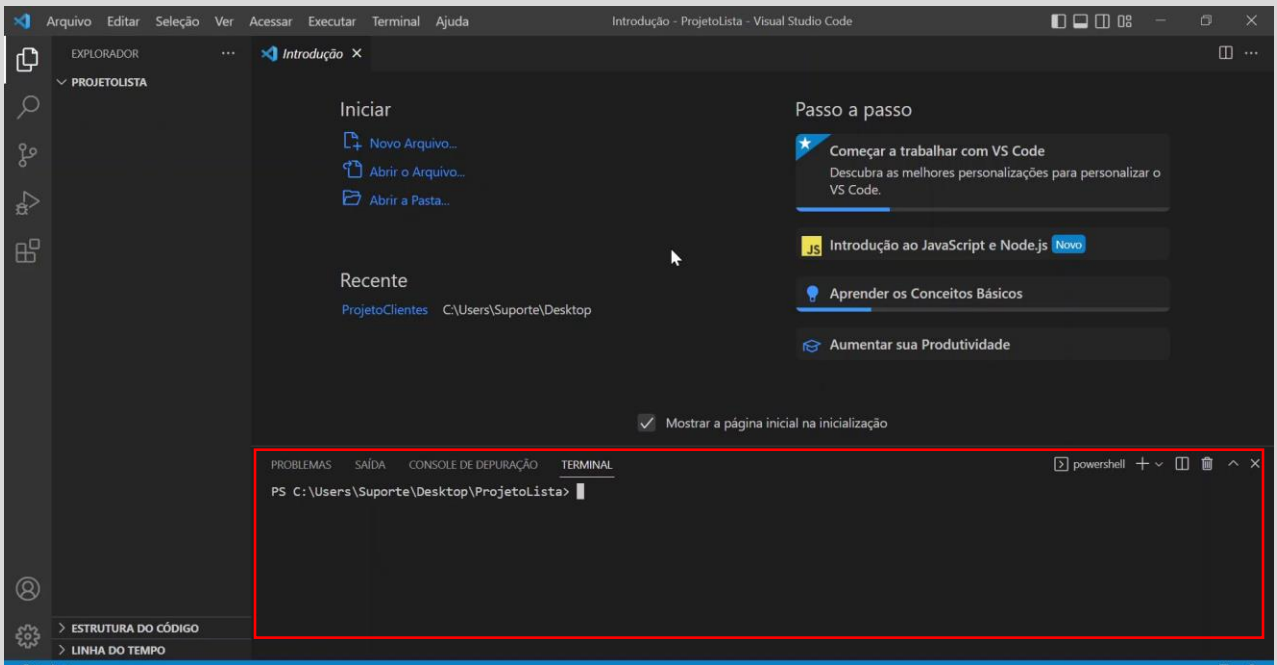
2. Abra essa pasta e selecione o texto na **Barra de Endereços**, conforme indicado a seguir:



3. Apague o texto da **Barra de Endereços**, digite **cmd** e pressione **Enter**. Na janela, digite o comando **code .**, conforme indicado a seguir (**atenção**: tudo em minúsculo e atenção para o **espaço** antes do ponto).



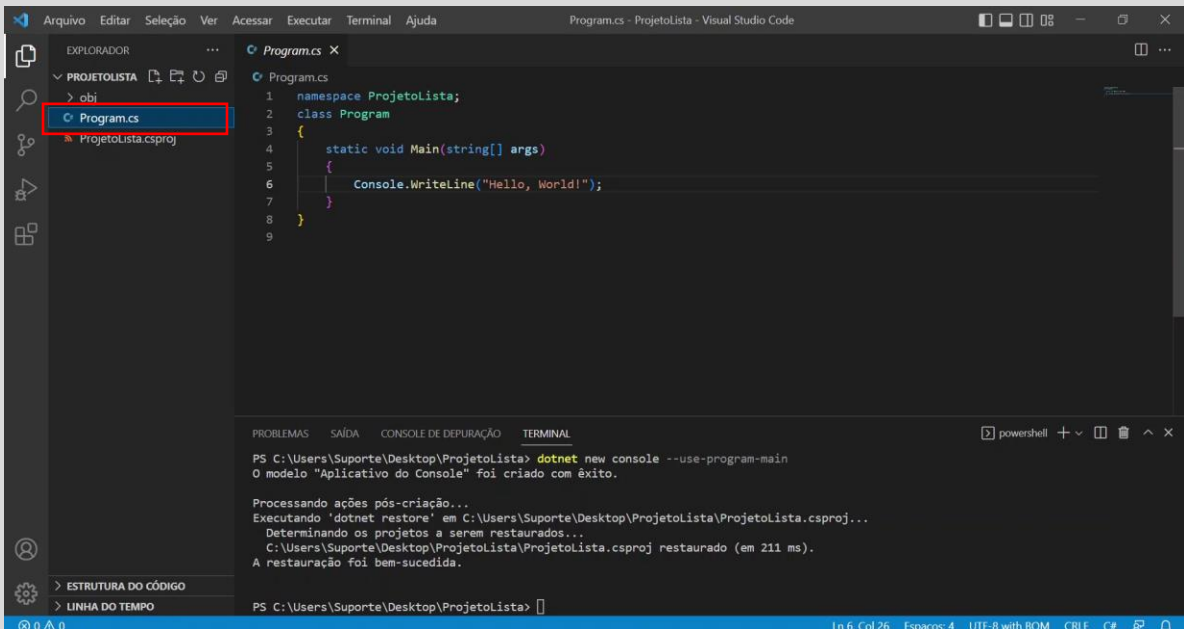
4. O VSCode irá iniciar com o arquivo **PROJETOLISTA** aberto. Utilize o atalho **Ctrl + Apóstrofe** para abrir o Terminal na parte inferior do programa, como indicado a seguir:



5. Digite no Terminal o código a seguir:

```
dotnet new console --use-program-main
```

6. Selecione **Program.cs** no menu à esquerda.



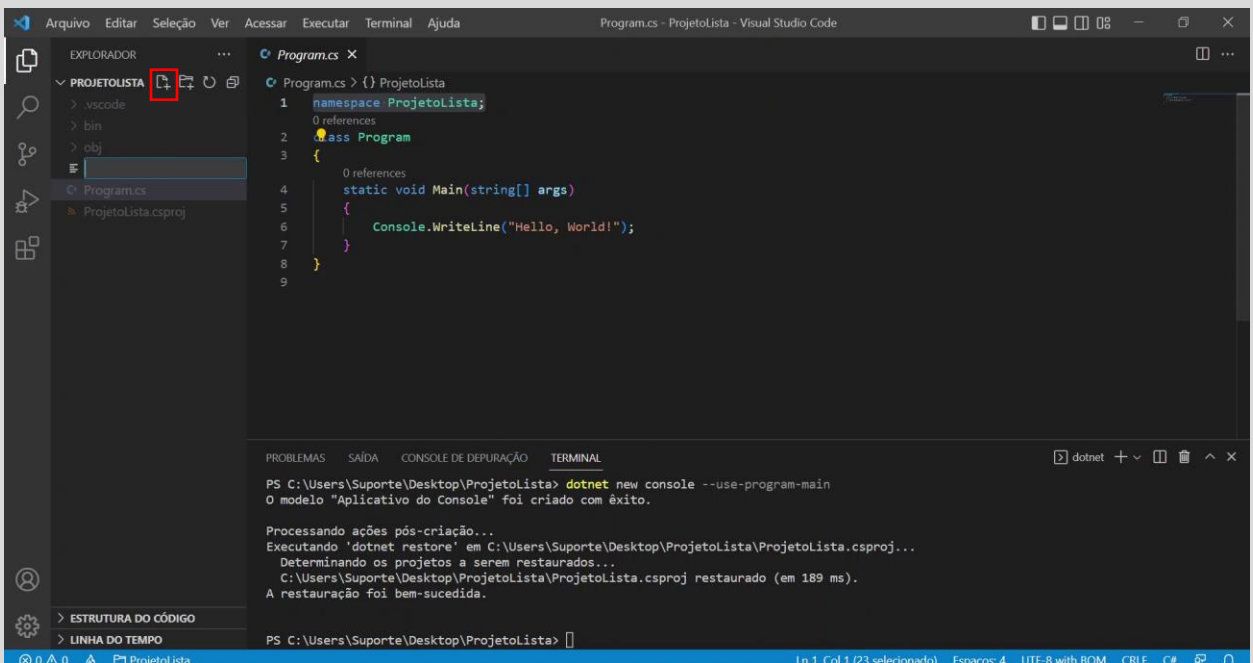
Você sabia

Muitas vezes não é possível prever a quantidade de cadastros que o programa irá armazenar. Nesses casos, usamos um objeto chamado List, que permite receber uma quantidade flexível de elementos sem interferir no processamento do programa.

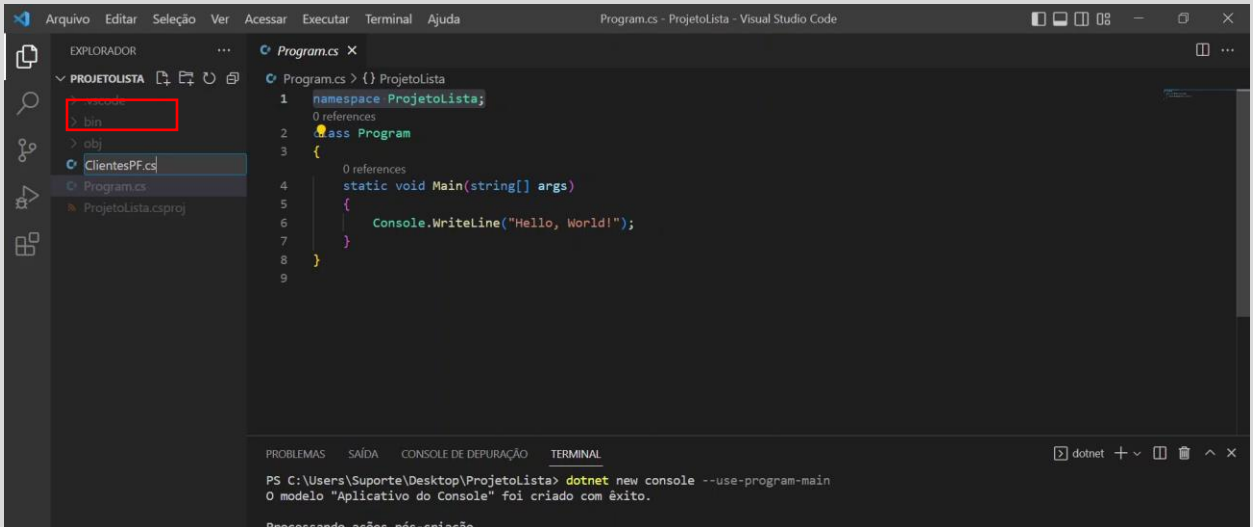


Classe Clientes

1. No menu à esquerda, clique no botão **Novo Arquivo...** como indicado a seguir.



2. No nome do arquivo, digite **ClientesPF.cs**.



3. Com **ClientesPF** criado, agora vamos digitar o código. Copie e cole conforme a seguir:

```
class ClientesPF{  
  
    public int id {set; get;}  
    public string? nome {set; get;}  
  
}
```

4. No menu à esquerda, clique no arquivo **Program.cs**. Copie a parte do código destacada na estrutura a seguir:

```
namespace ProjetoLista;
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello, World!");

        ClientesPF cl1= new ClientesPF();
        cl1.id=1;
        cl1.nome="Ana";

        List<ClientesPF> lista = new List<ClientesPF>();

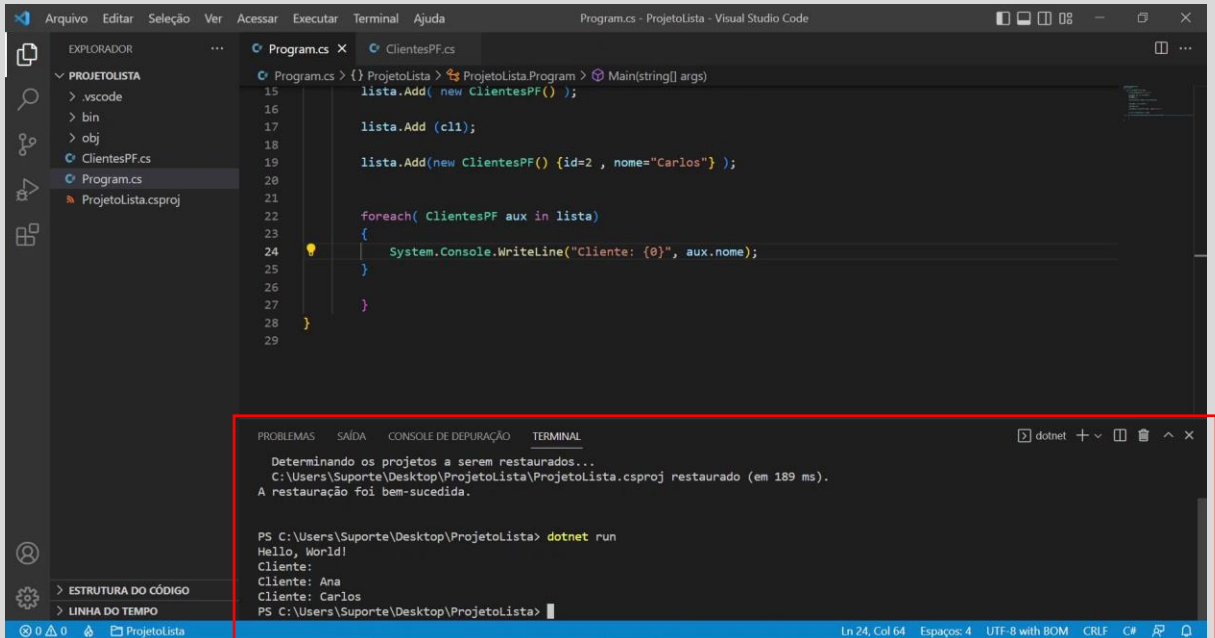
        lista.Add( new ClientesPF() );

        lista.Add (cl1);

        lista.Add(new ClientesPF() {id=2,nome="Carlos"} );

        foreach( ClientesPF aux in lista)
        {
            System.Console.WriteLine("Cliente: {0}",
aux.nome);
        }
    }
}
```

5. No terminal, digite **dotnet run** para verificar.



The screenshot shows the Visual Studio Code interface with a C# project named 'ProjetoLista'. The file explorer on the left shows the project structure. The main editor displays the 'Program.cs' file with the following code:

```
15 lista.Add( new ClientesPF() );
16
17 lista.Add( c11);
18
19 lista.Add(new ClientesPF() {id=2 , nome="Carlos"} );
20
21
22 foreach( ClientesPF aux in lista)
23 {
24     System.Console.WriteLine("Cliente: {0}", aux.nome);
25 }
26
27
28
29 }
```

The terminal at the bottom shows the output of the command `dotnet run`:

```
dotnet
Determinando os projetos a serem restaurados...
C:\Users\Suporte\Desktop\ProjetoLista\ProjetoLista.csproj restaurado (em 189 ms).
A restauração foi bem-sucedida.

PS C:\Users\Suporte\Desktop\ProjetoLista> dotnet run
Hello, World!
Cliente:
Cliente: Ana
Cliente: Carlos
PS C:\Users\Suporte\Desktop\ProjetoLista>
```

Importante

O comando **Foreach** instancia um contador auxiliar e em seguida executa toda a lista.

Em nosso exemplo, o **Foreach** vai percorrer a lista e, a cada índice dela, será atribuída a variável **aux**; o comando **System.Console.WriteLine** imprime o atributo "nome do índice" da lista que estiver naquele momento na variável **aux**. Exemplo: se o índice for 0, a variável **aux** está com 0 e será exibido o atributo "nome" do elemento que estiver na posição 0 da lista.

