



# **SW-I**

# **Sistemas Web I**

Prof. Anderson Vanin

---

# Ambiente de Desenvolvimento – VS Code

Para Executar C# podemos utilizar o Visual Studio Code, mas é necessário a instalação de um plugin e verificar se a plataforma .NET 6.0 está presente em sua máquina.

<https://dotnet.microsoft.com/en-us/download>

Free. Cross-platform. Open source.

## Download .NET

For Windows

### .NET 7.0

Standard Term Support Recommended

.NET SDK x64

Version 7.0.2, released January 10, 2023

[All .NET 7.0 downloads](#) [All .NET versions](#)

### .NET 6.0

Long Term Support

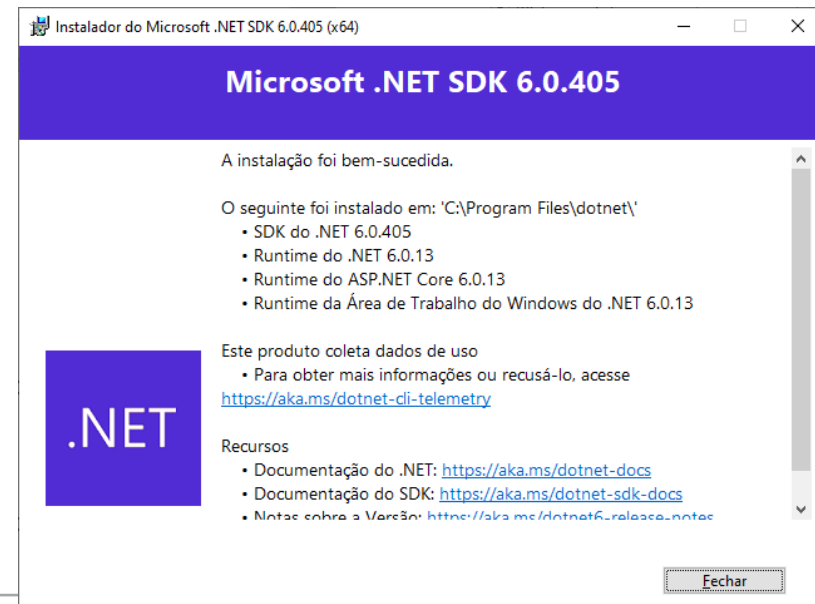
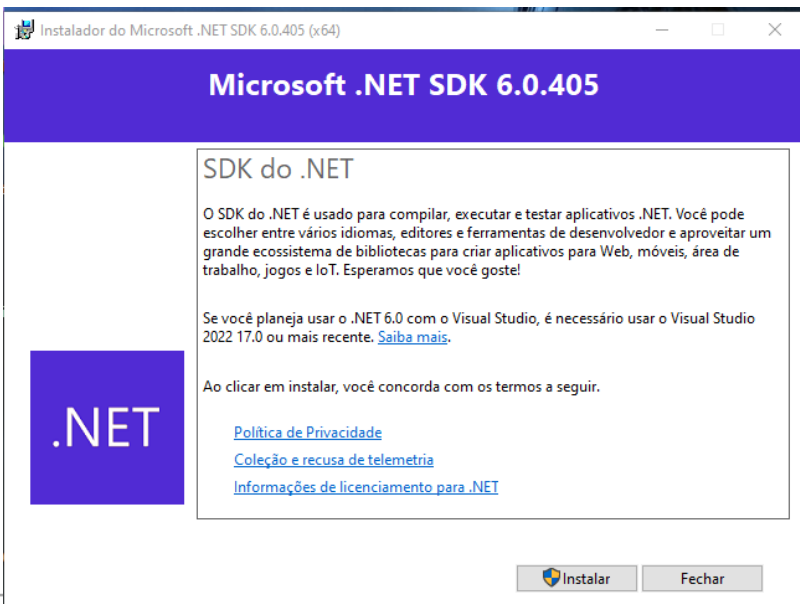
.NET SDK x64

Version 6.0.13, released January 10, 2023

[All .NET 6.0 downloads](#)

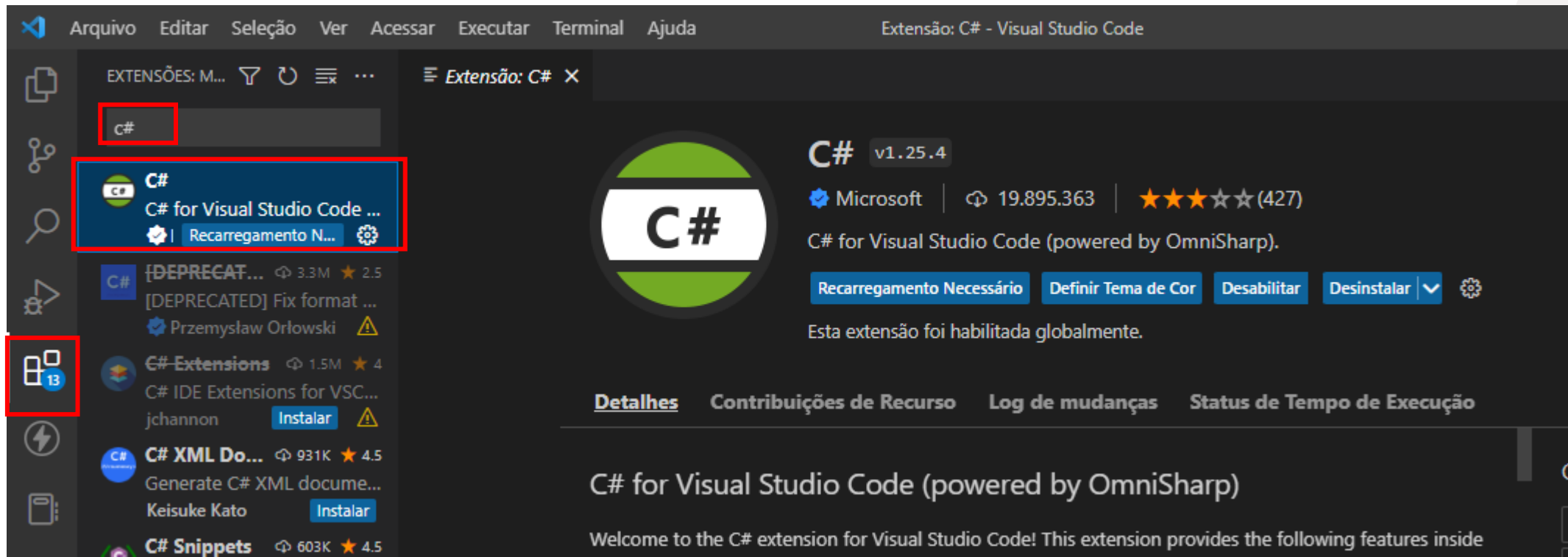
# Plataforma .Net

Após o download faça a instalação padrão seguindo todas as informações do instalado.

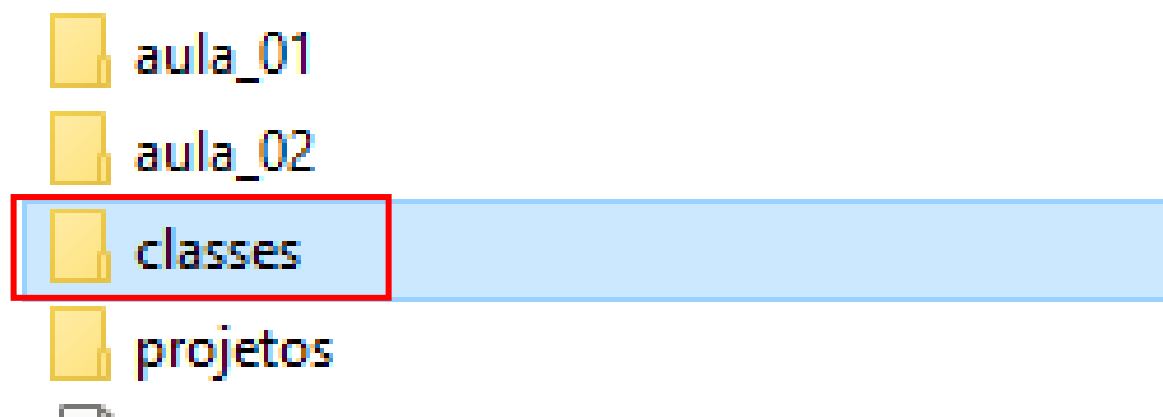


# VS Code

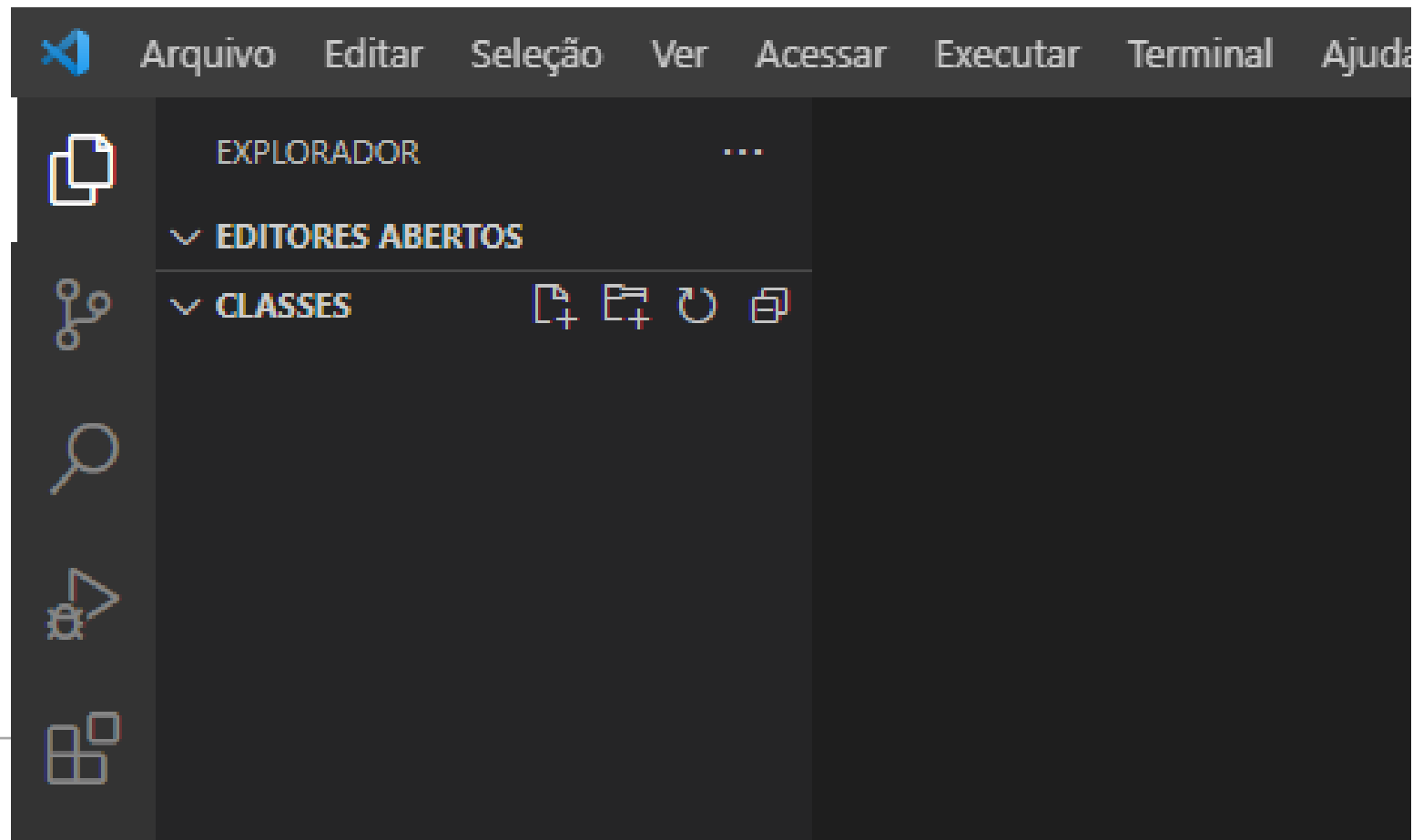
Instale o plugin C# no Visual Studio Code



Crie um diretório chamado classes em um local de sua preferência.



Abra o VS Code no diretório criado



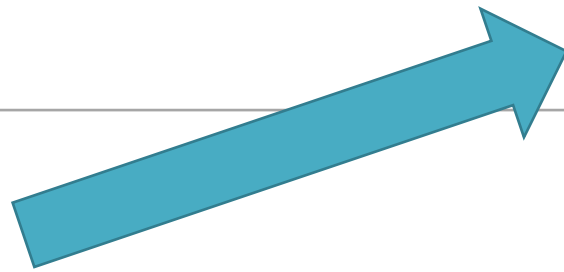
Abra um novo terminal e digite o comando:  
***dotnet new console***

```
PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL  JUPYTER  
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes>dotnet new console
```

***Para gerar no modelo antigo .Net 5.0 use:***

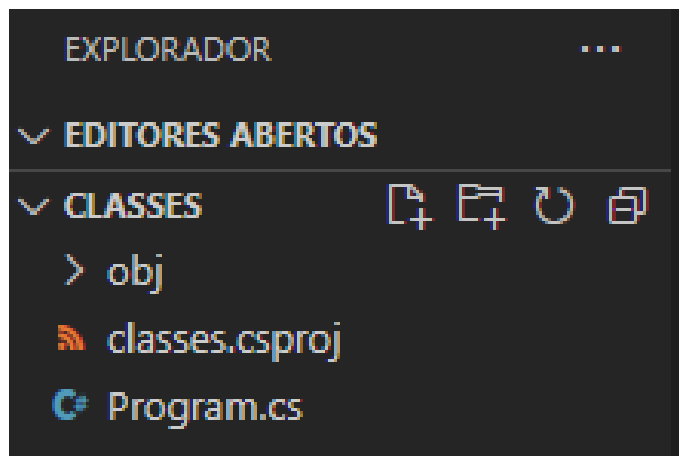
***dotnet new console --use-program-main***

Nesta aula estarei usando o modelo antigo.



# Estrutura de pastas

Ao final da execução do comando você deve ter a seguinte estrutura de pastas:



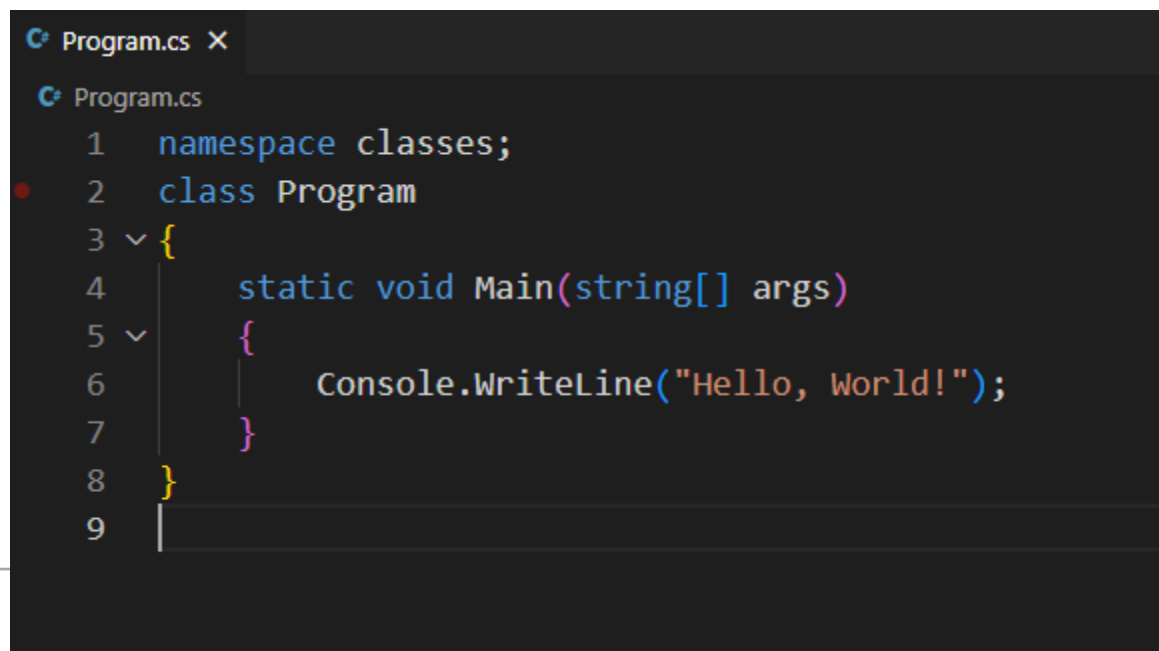
```
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes>dotnet new console --use-program-main
O modelo "Aplicativo do Console" foi criado com êxito.

Processando ações pós-criação...
Executando 'dotnet restore' em D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes\classes.csproj...
Determinando os projetos a serem restaurados...
```



# Program.cs

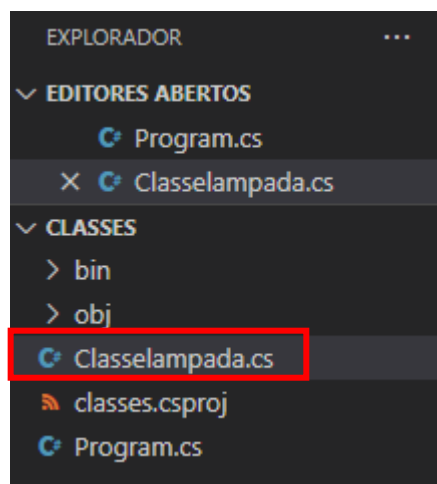
O arquivo **Program.cs** é o nosso arquivo principal. Através dele é que todas as classes criadas em outros arquivos serão instanciadas nele, gerando os novos objetos das classes referenciadas nele.

A screenshot of a code editor window titled 'Program.cs'. The code is written in C# and defines a namespace 'classes' and a class 'Program'. The 'Program' class has a static method 'Main' that takes an array of strings 'args' and writes 'Hello, World!' to the console. The code is as follows:

```
1 namespace classes;
2 class Program
3 {
4     static void Main(string[] args)
5     {
6         Console.WriteLine("Hello, World!");
7     }
8 }
9
```

# Classe: Lampada

Agora vamos criar a classe **Classelampada**. Dentro do diretório **classes**, crie um novo arquivo chamado **Classelampada.cs**



```
Program.cs  Classelampada.cs X
Classelampada.cs > ...
1  namespace classes;
   0 references
2  class Classelampada
3  {
   3 references
4      public bool ligada;
   0 references
5      public double potencia;
6
   0 references
7      public void ligar(){
8          ligada = true;
9      }
   0 references
10     public void desligar(){
11         ligada = false;
12     }
   0 references
13     public bool estaLigada(){
14         return ligada;
15     }
16 }
```

# Instanciando um novo objeto

Agora que temos a classe ***Classelampada.cs*** criada (o nosso “molde”) voltamos ao programa principal ***Program.cs***, onde vamos instanciar um novo objeto dessa classe chamado teste1 e acessar seus atributos métodos.

---



# Instanciando um novo objeto

Program.cs > ...

```
1 namespace classes;
  0 references
2 class Program
3 {
  0 references
4 static void Main(string[] args)
5 {
6     Classelampada teste1 = new Classelampada();
7
8     //Vamos verificar o estado da lampada teste1 acessando o método estaLigada()
9     teste1.estaLigada();
10
11 }
12 }
```

objeto

Nova instância chamada teste1 da classe **Classelampada()**

método

Chamada do método estaLigada() na classe **Classelampada()**

Para rodar este programa digite no prompt de comando:  
**dotnet run**

# Instanciando um novo objeto

```
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes>dotnet run
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes\Classelampada.cs(5,19): warning CS0649: Campo "Classelampada.potencia" nunca é atribuído e sempre terá seu valor padrão 0 [D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes\classes.csproj]
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes>
```

Repare que a não foi mostrado nada na saída, pois não especificamos o comando para escrever a saída na tela. Vamos alterar o código para visualizar esta saída, o retorno da função chamada.

```
Program.cs > ...
1 namespace classes;
  0 references
2 class Program
3 {
  0 references
4 static void Main(string[] args)
5 {
6     Classelampada teste1 = new Classelampada();
7
8     //Vamos verificar o estado da lampada teste1 acessando o método estaLigada()
9     Console.WriteLine(teste1.estaLigada());
10
11 }
12 }
```

```
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes>dotnet run
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes\Classelampada.c
02-2021\Etec 2023\2C E 2F - SW-I\classes\classes.csproj]
False
```

# Instanciando um novo objeto

Vamos melhorar um pouco a apresentação do resultado

```
Program.cs > ...
1 namespace classes;
  0 references
2 class Program
3 {
  0 references
4     static void Main(string[] args)
5     {
6         Classelampada teste1 = new Classelampada();
7
8         //Vamos verificar o estado da lampada teste1 acessando o método estaligada()
9         Console.WriteLine("A lâmpada está ligada? => " + teste1.estaligada());
10    }
11 }
12 }
```

```
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes>dotnet run
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes\Classelampada.cs(5,19): v
02-2021\Etec 2023\2C E 2F - SW-I\classes\classes.csproj]
A lâmpada está ligada? => False
```

# Instanciando um novo objeto

Vamos alterar o estado da lâmpada e depois chamar novamente o método que verifica o estado dela.

```
Program.cs > ...
1 namespace classes;
  0 references
2 class Program
3 {
  0 references
4     static void Main(string[] args)
5     {
6         Classelampada teste1 = new Classelampada();
7
8         teste1.ligar();
9
10        //Vamos verificar o estado da lampada teste1 acessando o método estaLigada()
11        Console.WriteLine("A lâmpada está ligada? => " + teste1.estaLigada());
12
13    }
14 }
```

```
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes>dotnet run
D:\bkp 13-02-2021\Etec 2023\2C E 2F - SW-I\classes\Classelampada
02-2021\Etec 2023\2C E 2F - SW-I\classes\classes.csproj]
A lâmpada está ligada? => True
```