# Enumeração – enum 23/12/2020

Enumerações:

É um tipo especial que serve para especificar de forma literal um conjunto de constantes relacionadas

Palavra chave em C#: enum

**Nota:** enum é um tipo valor

Vantagem:

* Melhor semântica:
  + Exemplo básico. Ao representar os dias da semana, não será necessário usar 1 para domingo, 2 para segunda, ... Será possível utilizar os próprios nomes. Domingo, Segunda, ...
* Código mais legível e auxiliado pelo compilador.
  + Ou seja, o compilador não irá permitir inserir valores diferente dos pré-estipulados.

Exemplo: Ciclo de vida de um **pedido**

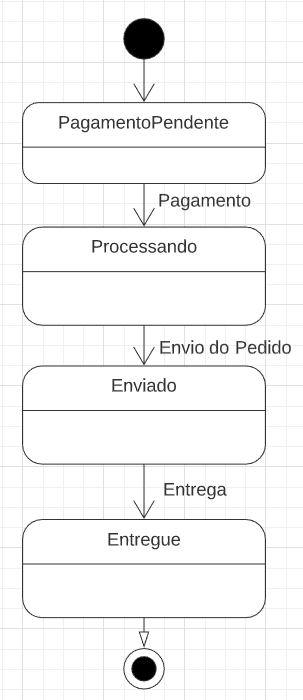
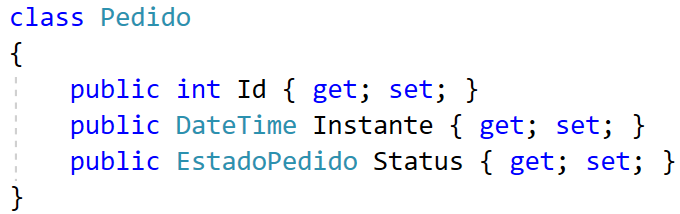


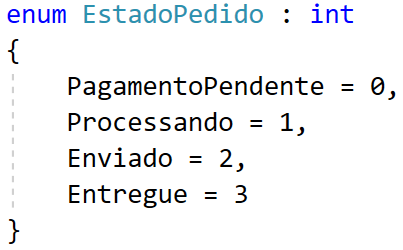
Figura 1. Diagrama de máquina de estados

O Diagrama de máquina de estado, representa o ciclo de vida de alguma coisa. Quais os estados que ela passa.

As classes serão estas:

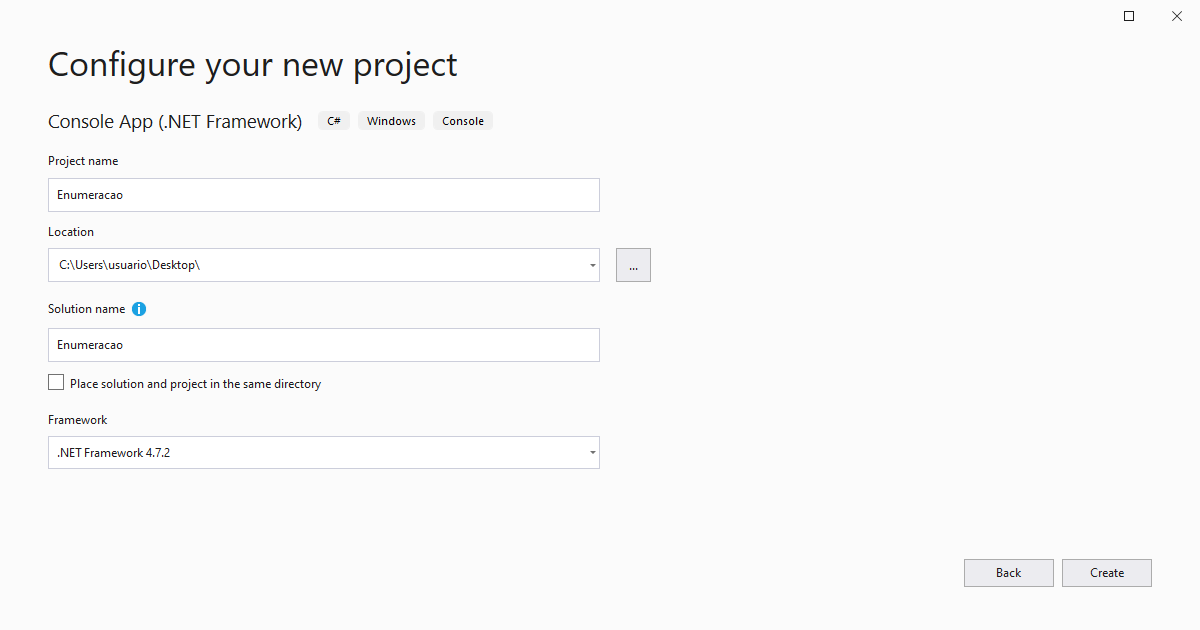


e

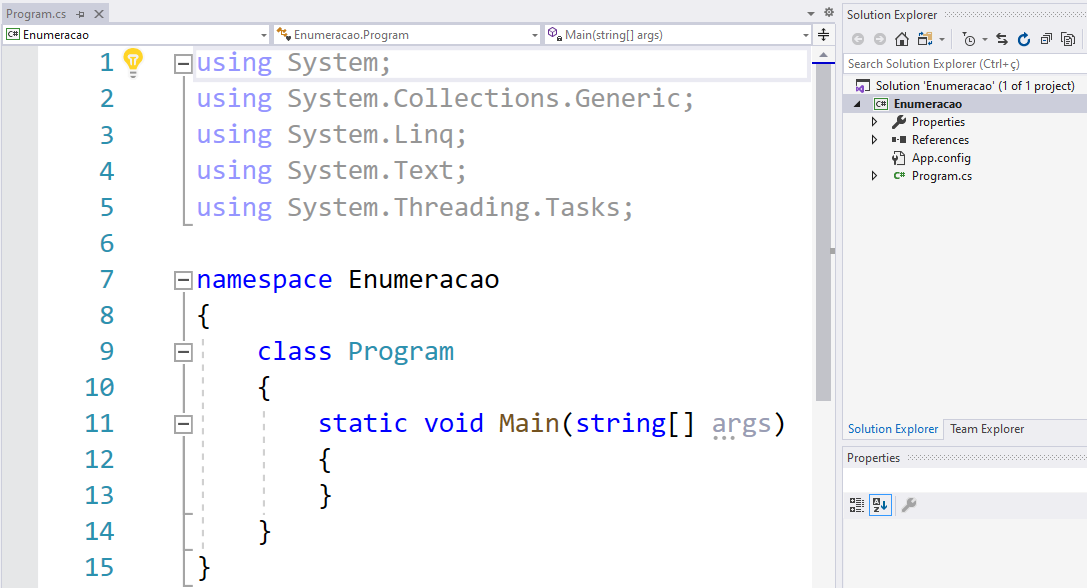


# Prática:

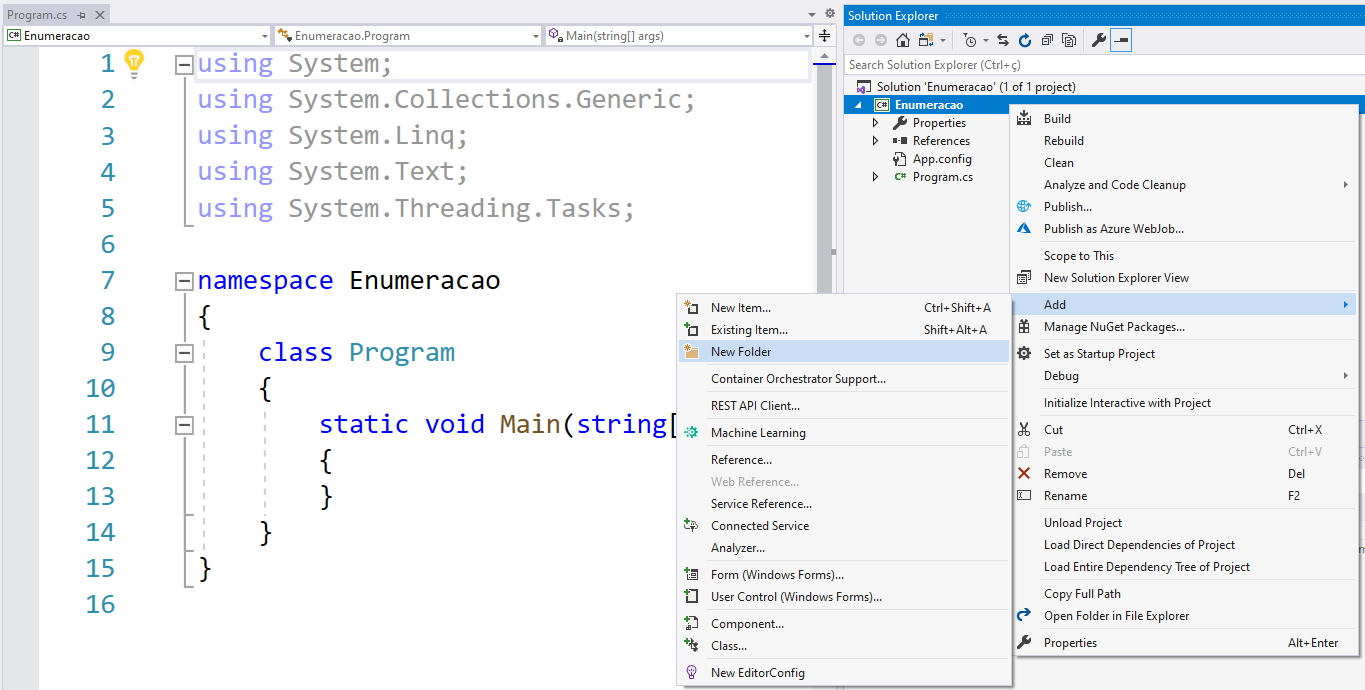
Crie um projeto chamado **Enumeracao**



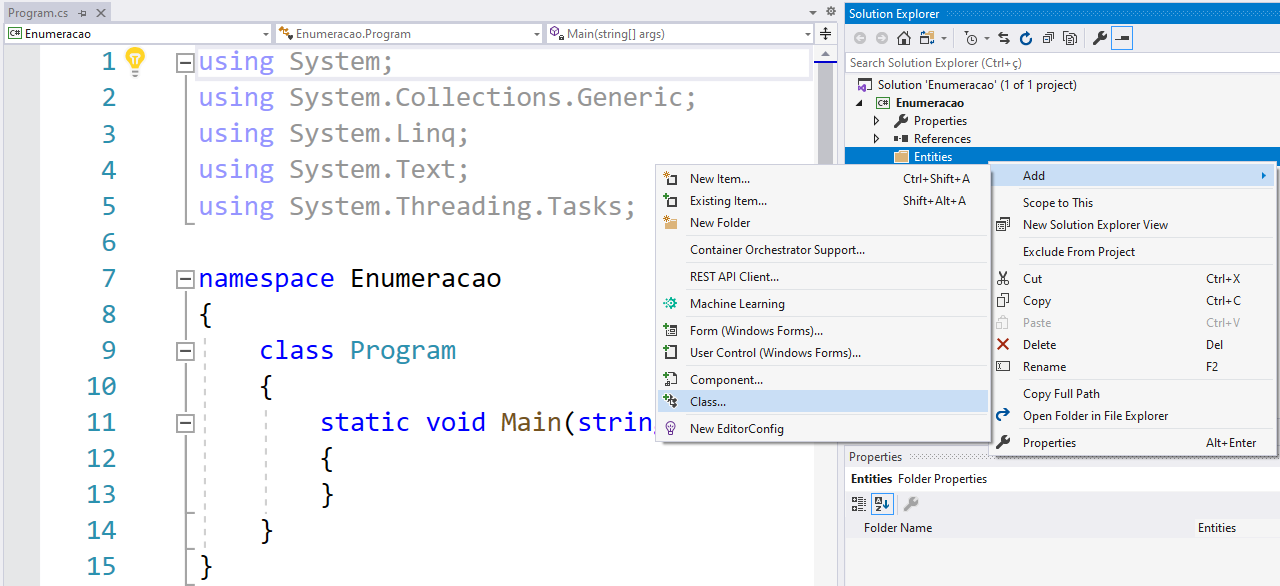
Visualizando



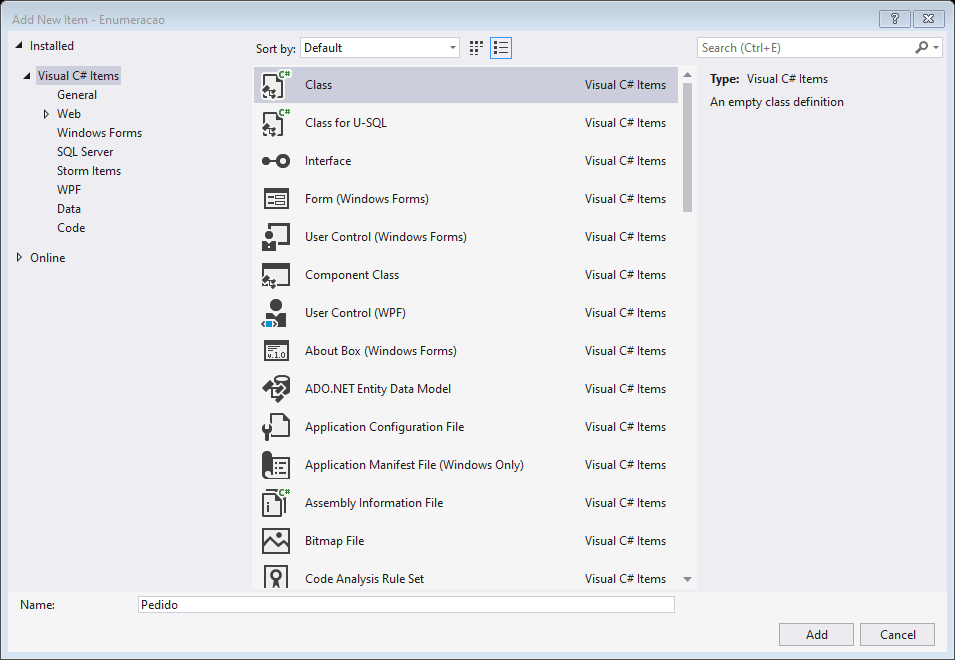
Para ficar mais organizado o projeto, crie uma pasta chamada Entities (endidades).



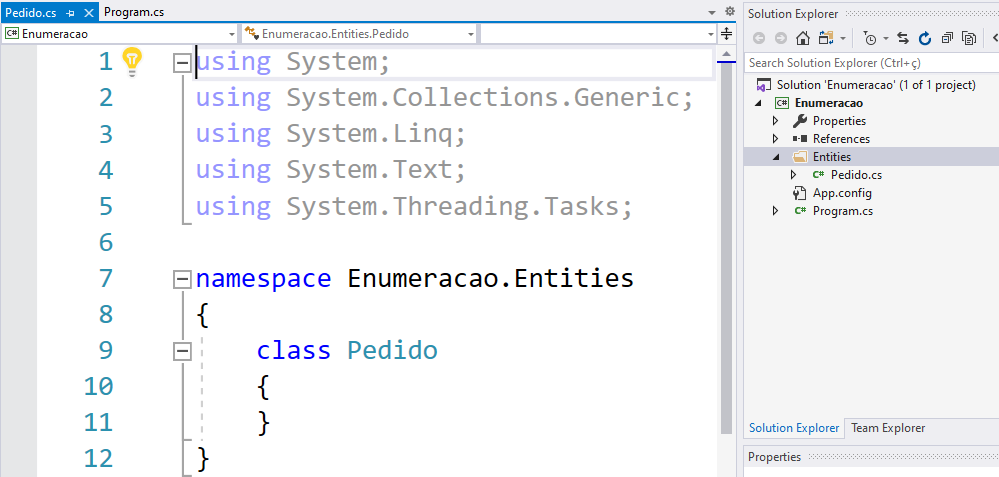
Dentro da pasta Entities, vamos criar uma classe chamada Pedido



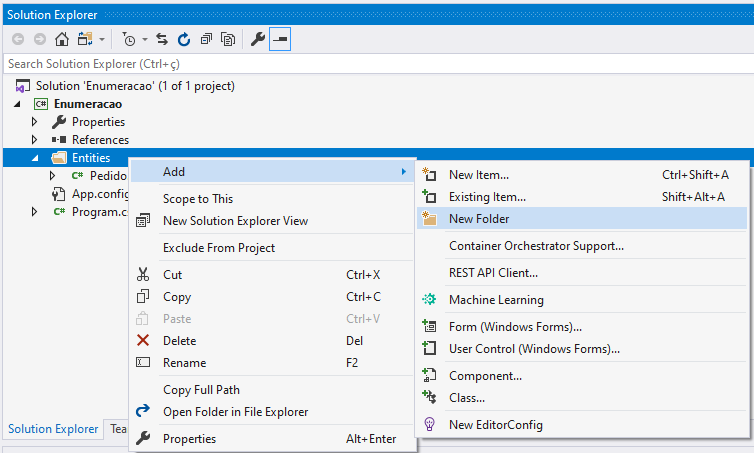
Nomeando



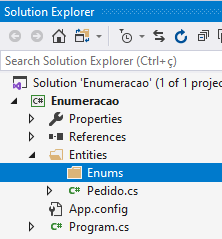
Ficará assim:



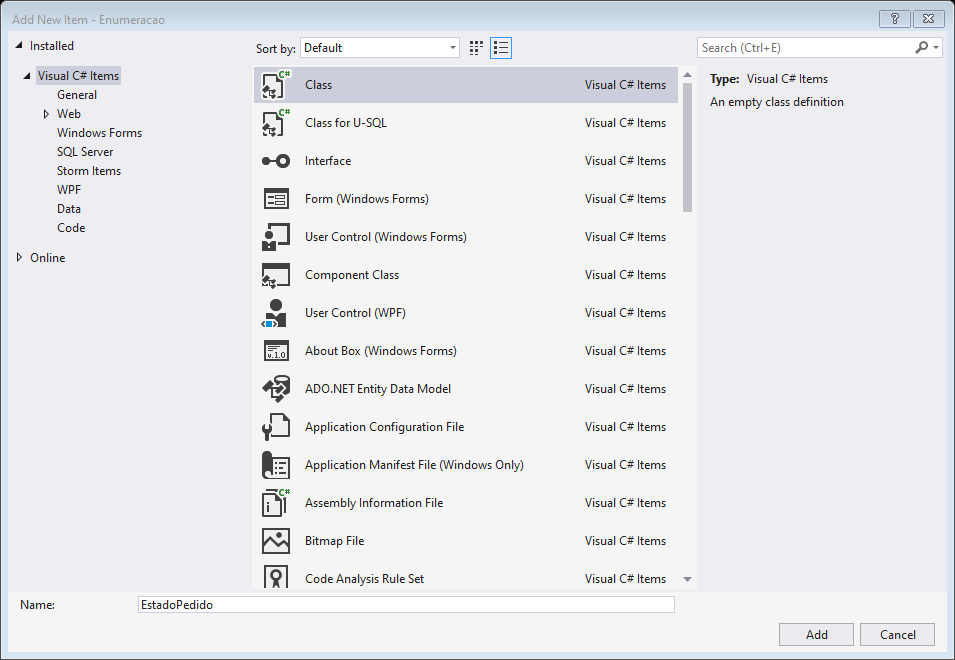
Uma opção para melhor organização do projeto é dentro da pasta Entities, criar uma subpasta chamada Enums (enumeração)



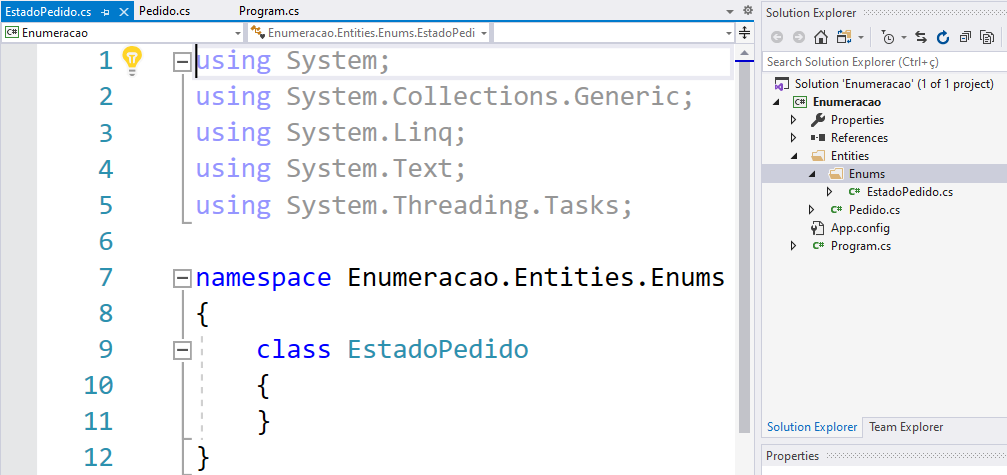
Folder criado



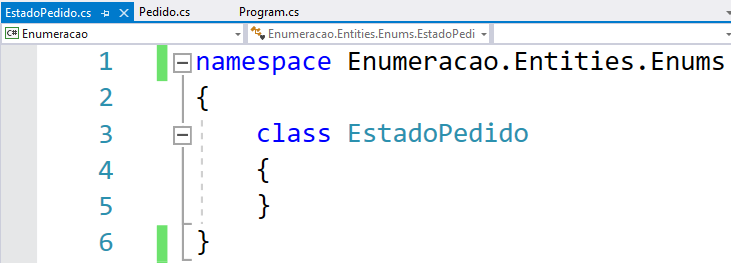
Dentro da pasta Enums, iremos criar uma classe chamada EstadoPedido.



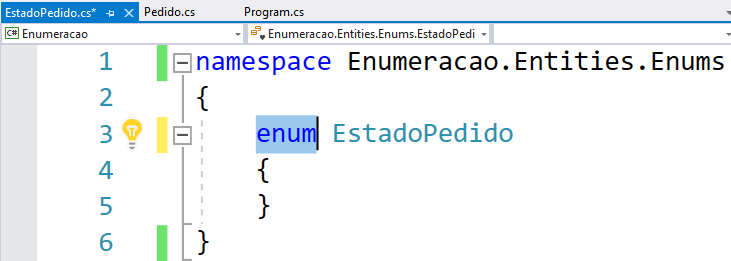
Ficará assim:



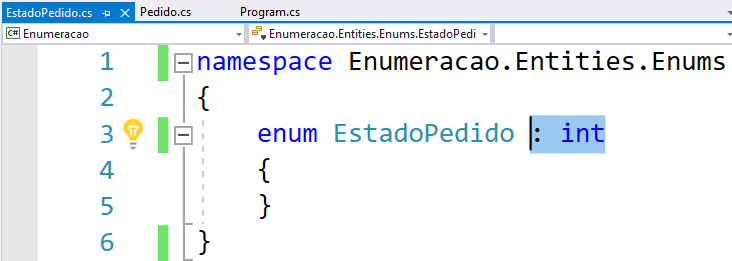
Apague as dependências desnecessárias para deixar o código mais limpo.



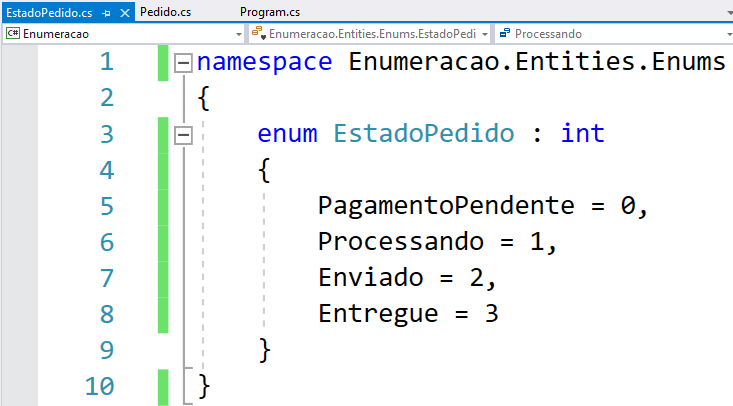
Altere class por enum



Como nossa enumeração será do tipo numérica. Vamos fazer o enum EstadoPedido herdar as propriedades de int .

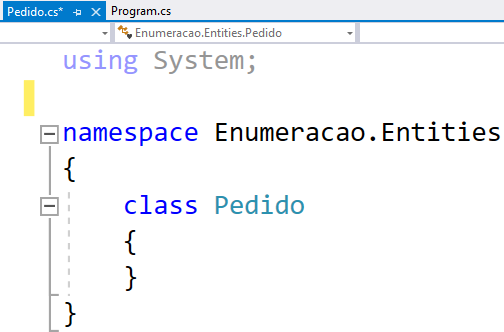


Vamos criar os estados da enumeração. Nessa situação, os estados são separados por , (vírgula).

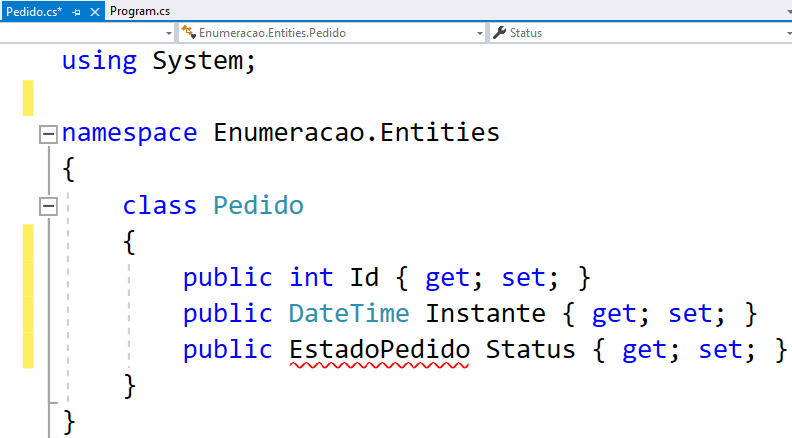


Vamos criar os atributos da classe pedido.

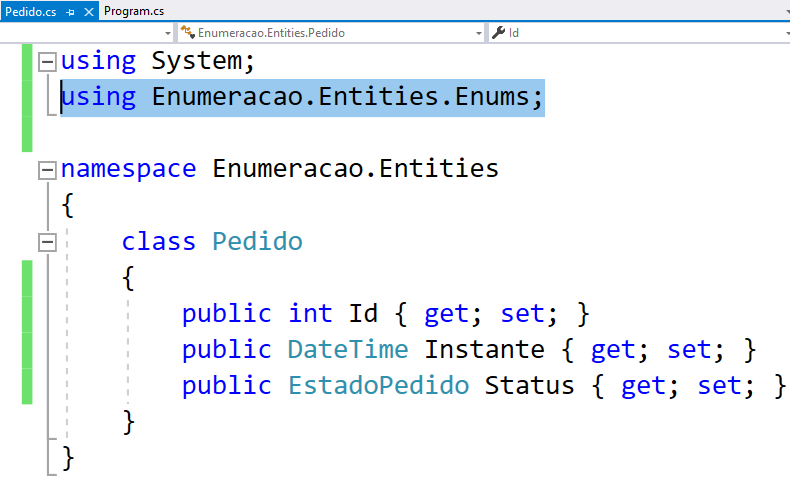
Apague as dependências desnecessárias. A System será necessária para utilizarmos o DateTime.



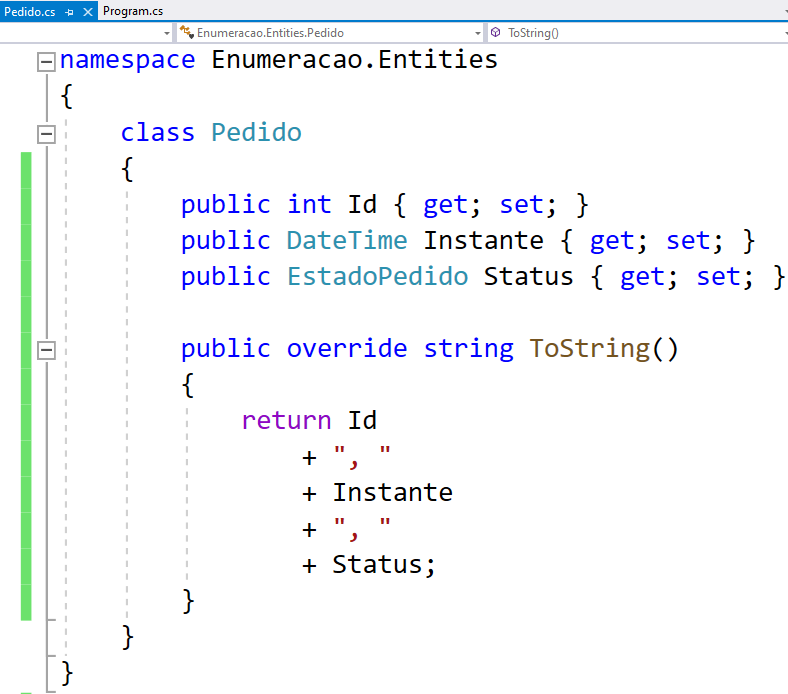
Observe o erro:



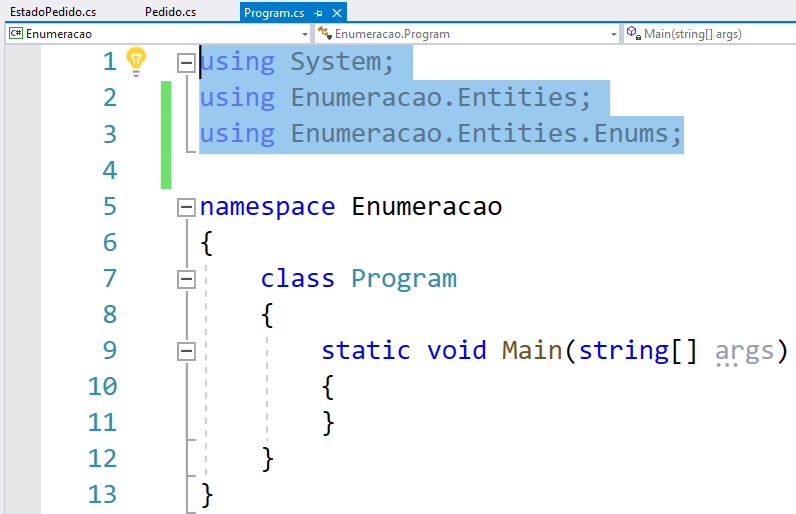
Precisamos referenciar as pastas.



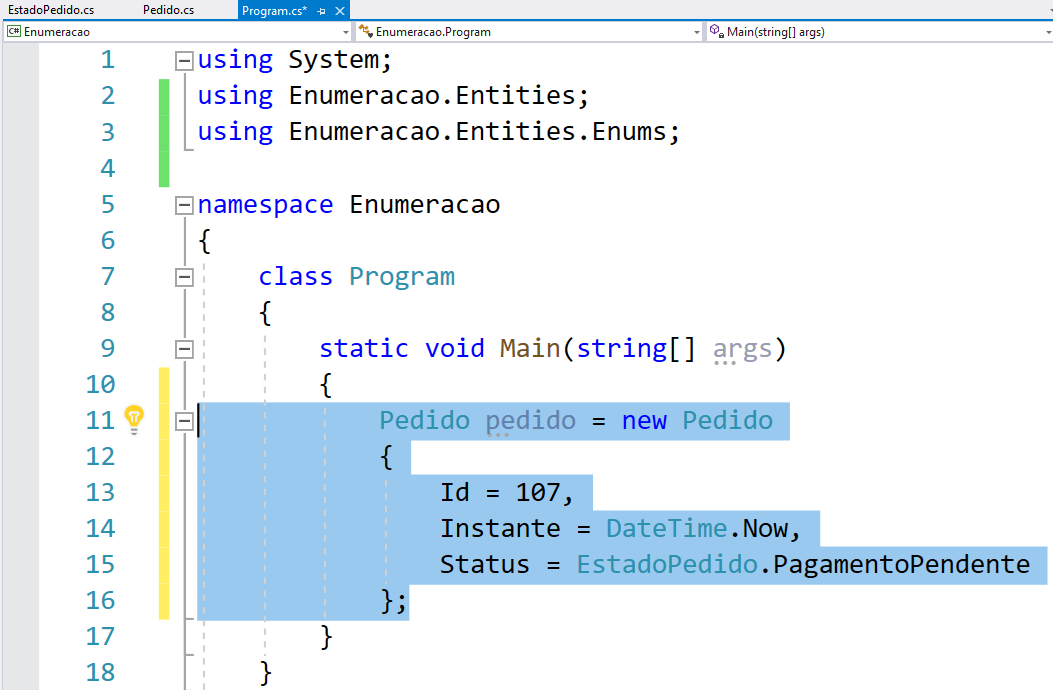
Vamos implementar com um override.



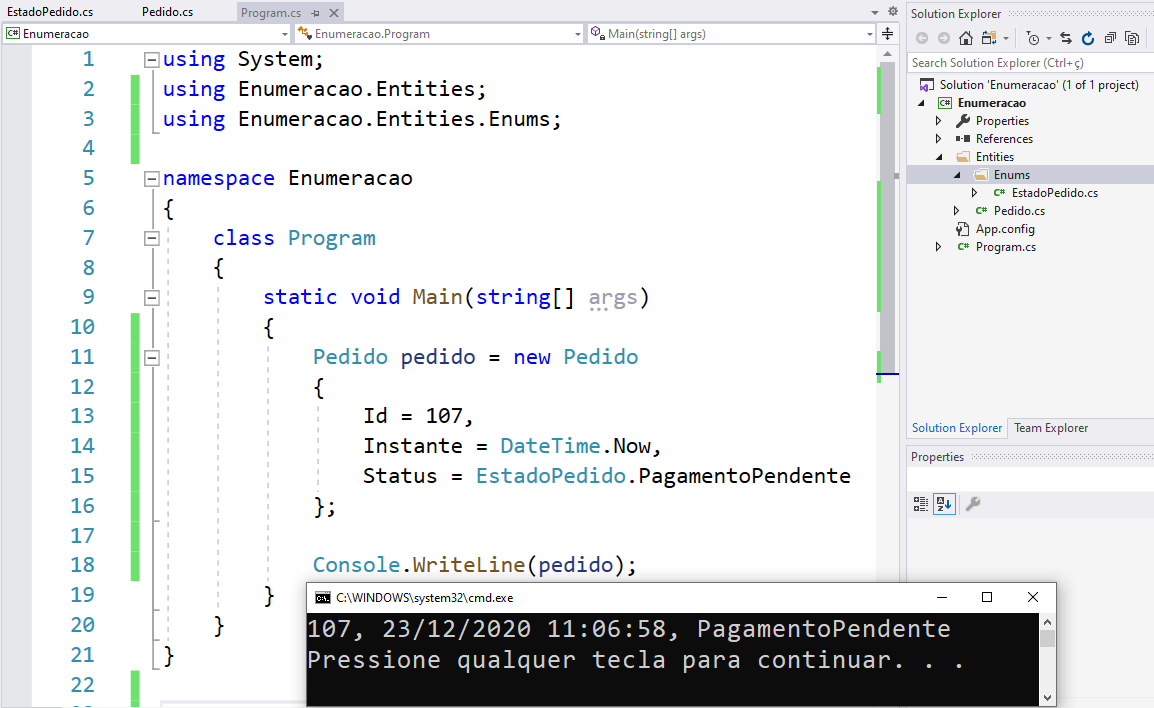
No programa principal, vamos referenciar as pastas e apagar as desnecessárias.



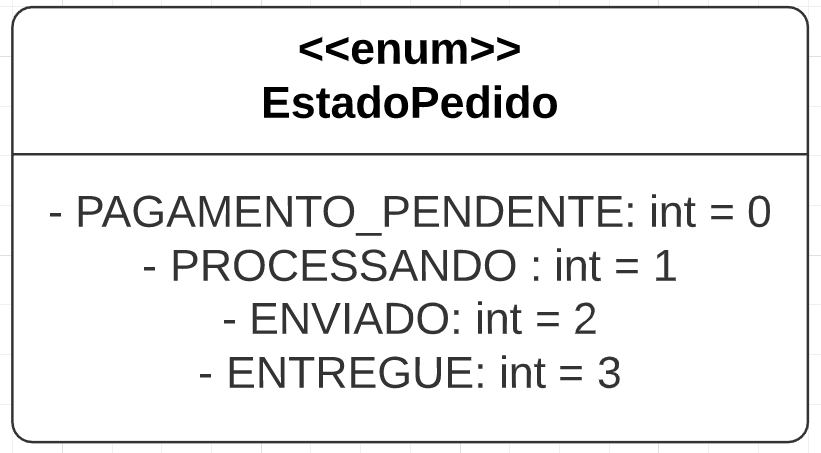
Vamos instanciar um pedido utilizando a sintaxe das chaves.

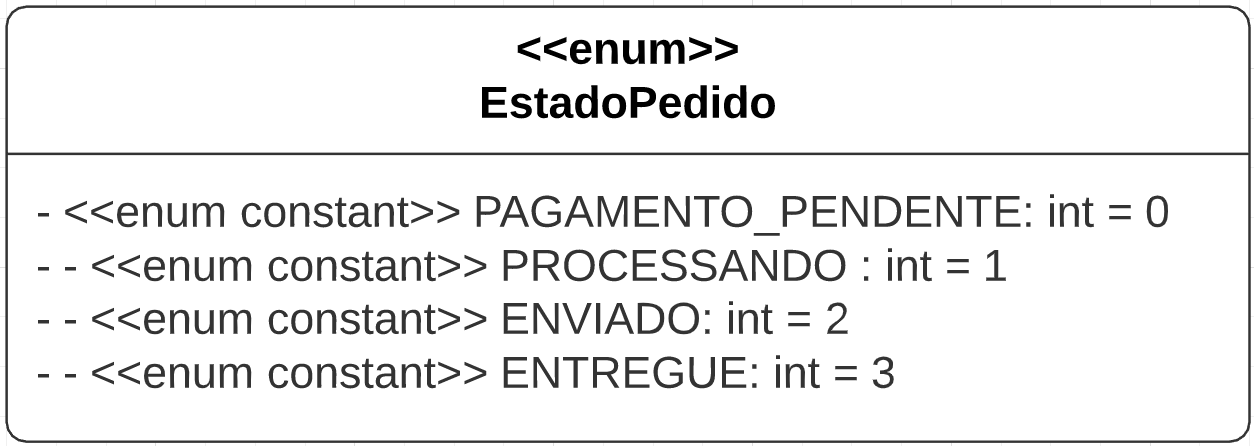


Vamos mandar imprimir.



Notação UML





Código do enum EstadoPedido

namespace Enumeracao.Entities.Enums

{

enum EstadoPedido : int

{

PagamentoPendente = 0,

Processando = 1,

Enviado = 2,

Entregue = 3

}

}

Código da classe Pedido:

using System;

using Enumeracao.Entities.Enums;

namespace Enumeracao.Entities

{

class Pedido

{

public int Id { get; set; }

public DateTime Instante { get; set; }

public EstadoPedido Status { get; set; }

public override string ToString()

{

return Id

+ ", "

+ Instante

+ ", "

+ Status;

}

}

}

Código da classe principal:

using System;

using Enumeracao.Entities;

using Enumeracao.Entities.Enums;

namespace Enumeracao

{

class Program

{

static void Main(string[] args)

{

Pedido pedido = new Pedido

{

Id = 107,

Instante = DateTime.Now,

Status = EstadoPedido.PagamentoPendente

};

Console.WriteLine(pedido);

}

}

}

# GitHub

O código está na subpasta \Aula\_08\_Classe\ **Enumeracao**

https://github.com/endroni/ProgramacaoDeAplicativos.git

# Referência:

Docs Microsoft – Tipos de enumeração (referência C#) – < https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/builtin-types/enum > Acessado dia 23 de dezembro de 2020.

Tutorial W3Schools – C# Enum - < https://www.w3schools.com/cs/cs\_enums.asp > Acessado dia 23 de dezembro de 2020.