

Linguagem de Programação Visual

5º Semestre – Sistemas de Informação

C# e .NET

- C# é uma linguagem de programação (regras sintáticas).
- .NET (2002) é uma plataforma de desenvolvimento para se criar diversos tipos de aplicações, podendo usar várias linguagens de programação (VB, F#).

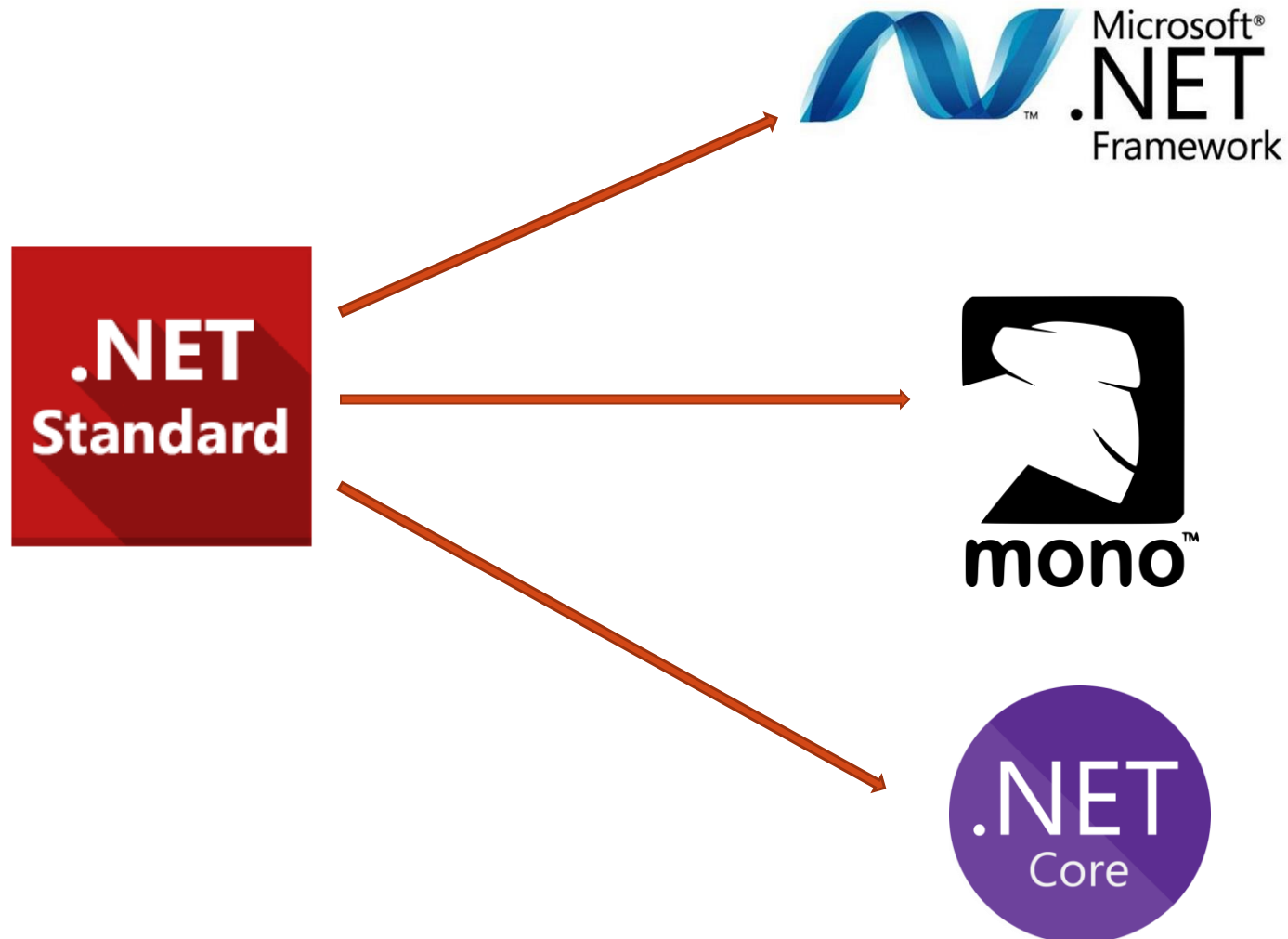
.NET

- BCL – Base Class Library
 - Conjunto de classes, funcionalidades que o .NET disponibiliza.
- CLR – Common Language Runtime (Máquina Virtual)
 - Executa os programas feitos no .NET
 - JVM do Java

Implementações do .NET

- O .NET é apenas uma especificação chamada .NET Standard.
- .NET Framework (2002), windows.
- Mono (2004) multiplataforma, código aberto.
 - Usam implementação do Mono como Xamarin, Unity.
- .NET Core (2016) participação da Microsoft e comunidade, código aberto, multiplataforma.
 - Rápido e performático.

Implementações do .NET



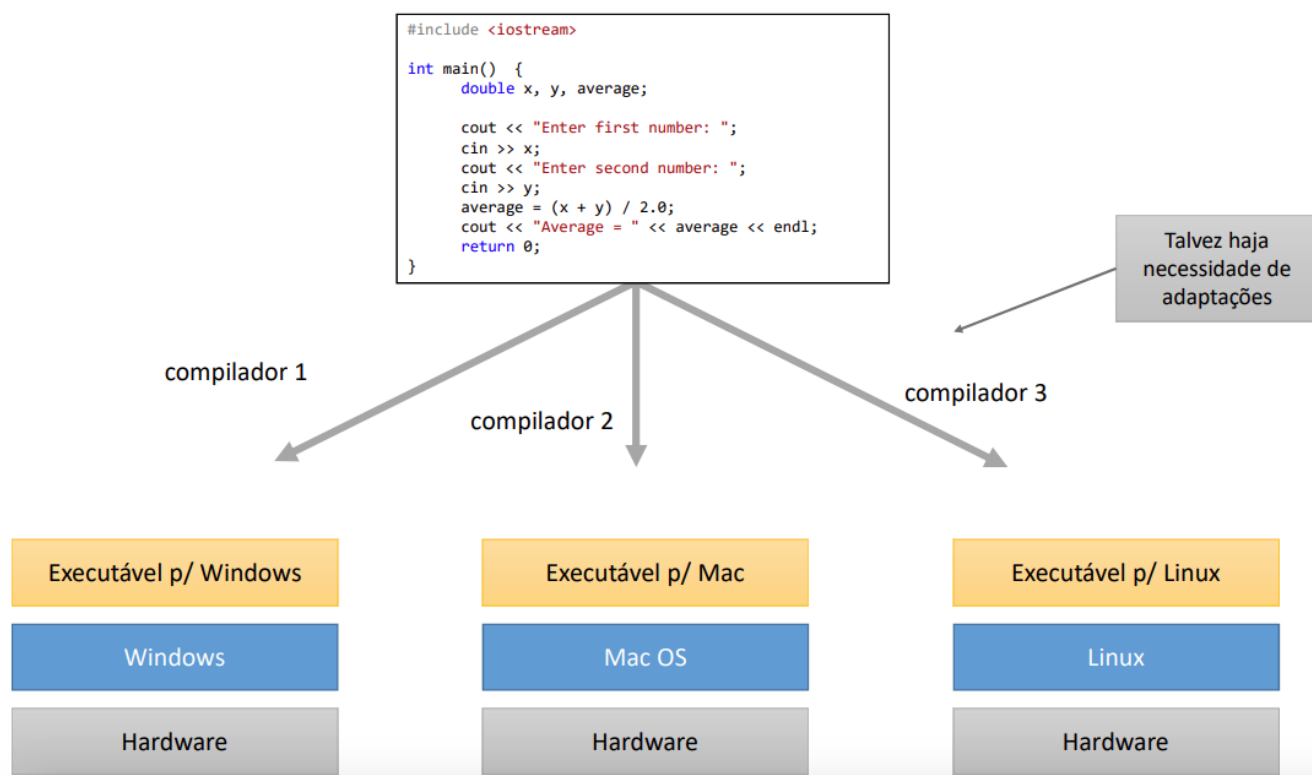
Implementações do .NET

	SO	Open Source	Propósito
.NET Standard	-	Sim	Especificação do .NET
.NET Framework	Windows	Não	Usado para criar aplicativos Windows desktop (console e gráfico) e aplicativos web ASP.NET rodando sobre o IIS
Mono	Vários	Sim	Usado para criar aplicativos e jogos multiplataforma
Xamarin	iOS Android Mac OS	Sim	Usado para criar aplicativos móveis híbridos para iOS e Android, e também para aplicações desktop para Mac OS
.NET Core	Windows Linux Mac OS	Sim	Usado para criar aplicativos modo console multiplataforma, e também para criar aplicativos web ASP.NET Core e serviços para nuvem

Compilação e Interpretação

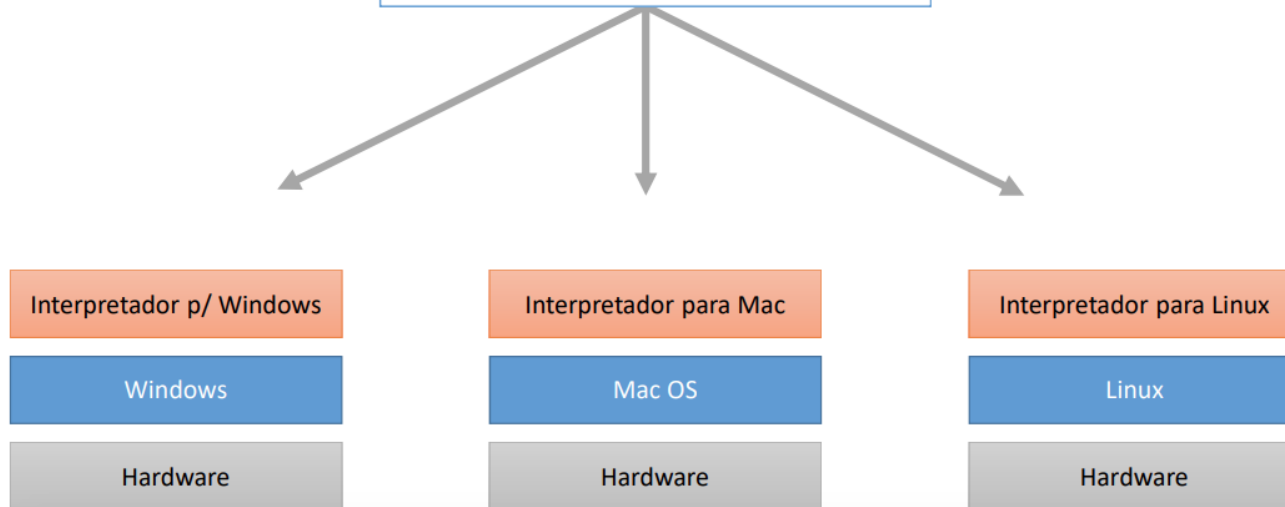
- Escrevemos código fonte usando uma linguagem de programação.
- Não é compreendido pela máquina, tem que ser transformado em código executável (linguagem de máquina).
- Dois processos: compilação e interpretação.
- Linguagens compiladas: C, C++.
- Linguagens interpretadas: PHP, JavaScript.
- Linguagens pré-compiladas que executam em uma máquina virtual: Java, C#.

Compilação e Interpretação



Compilação e Interpretação

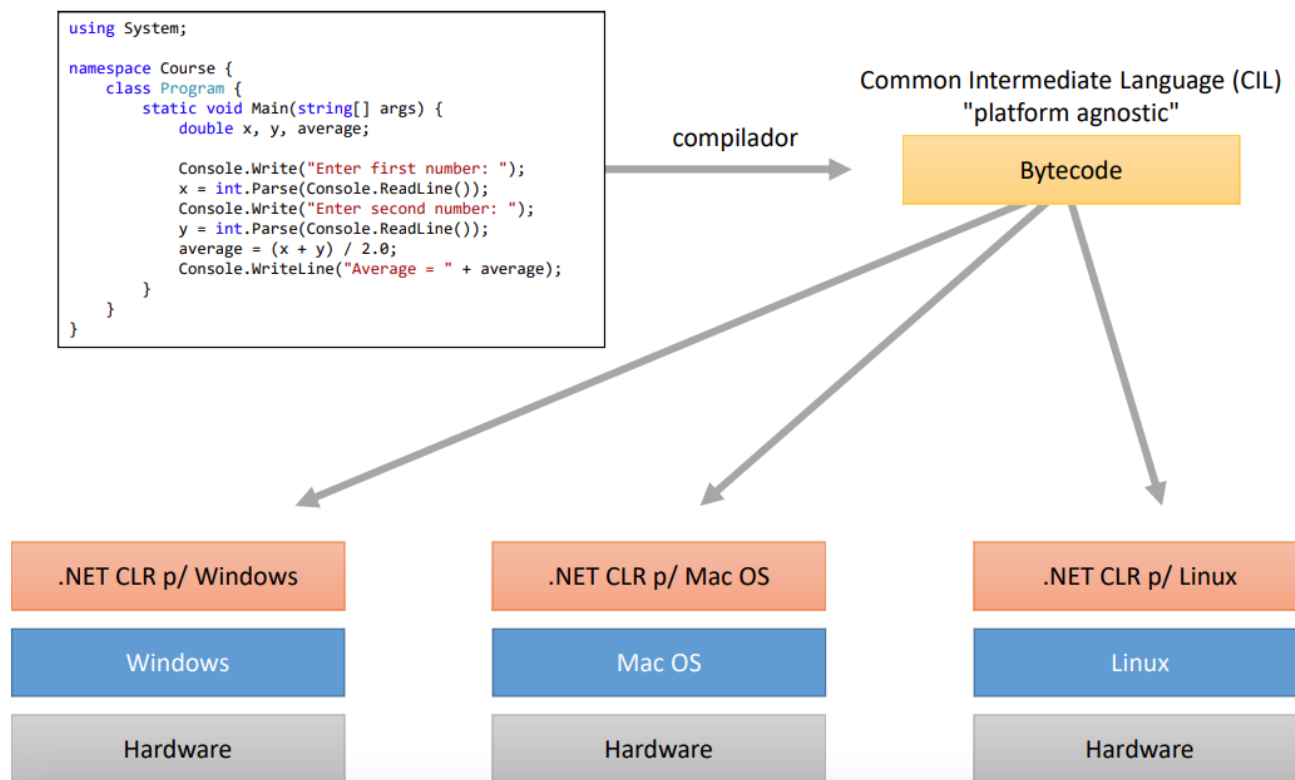
```
<?php
print "Enter first number: ";
$x = trim(fgets(STDIN));
print "Enter second number: ";
$y = trim(fgets(STDIN));
$average = ($x + $y) / 2;
print "Average = $average";
?>
```



Compilação e Interpretação

- O código fonte passa por um processo de pré-compilação que gera um código intermediário bytecode (linguagem intermediária).
- Vantagem: bytecode passou pelo processo de compilação, garanti que está correto sintaticamente.
- Bytecode será executado pela máquina virtual específica de cada plataforma.
- O processo de interpretação pela máquina virtual é chamado de just-in-time (JIT).

Compilação e Interpretação



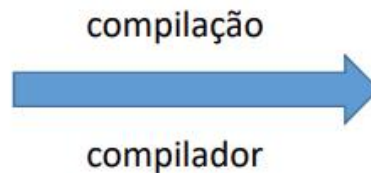
Modelo de execução

```
using System;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.WriteLine("Hello World");

        }
    }
}
```



Common Intermediate
Language (CIL)

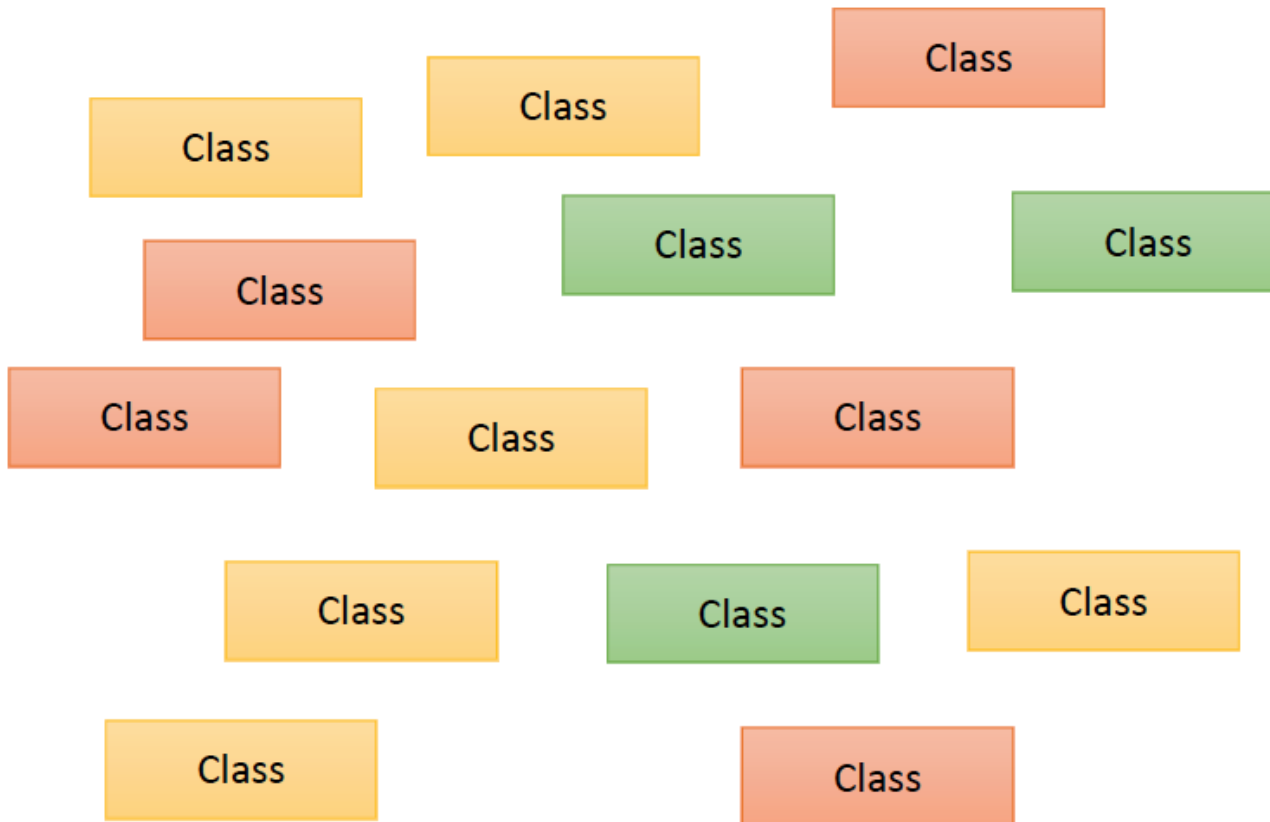
*Compilação just-in-time
(JIT)
Muito mais rápido que
a interpretação*

*.NET
Common Language Runtime
(CLR) - específica ao SO*

Código de máquina

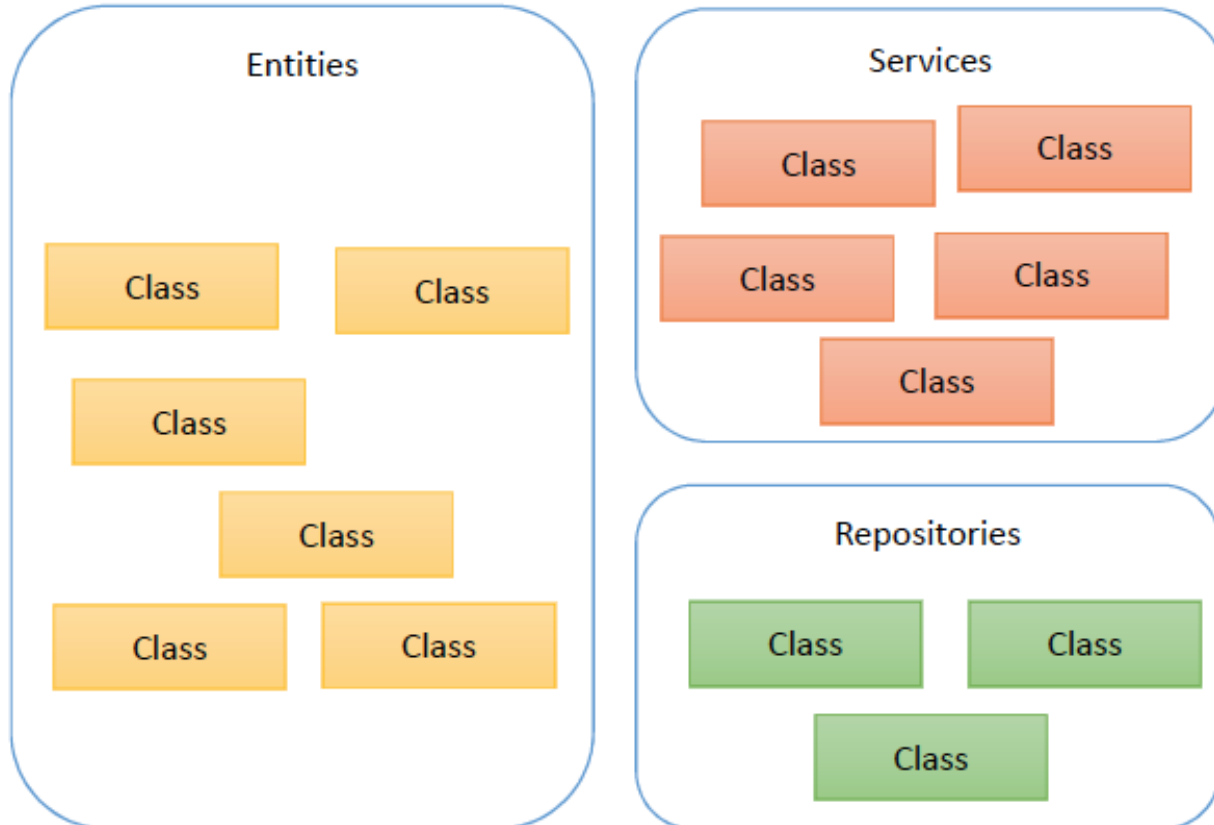
Estrutura de uma aplicação C# .NET

- Uma aplicação é composta por classes.



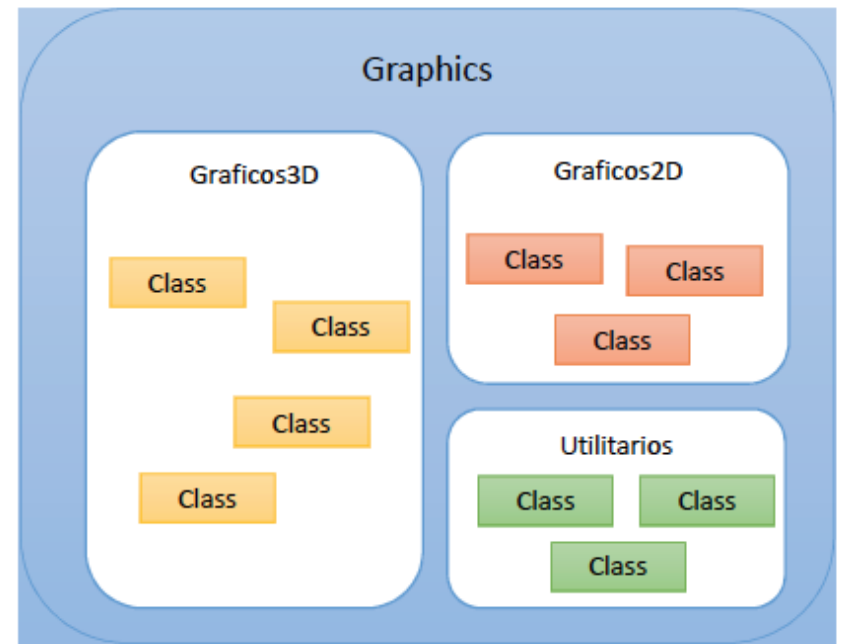
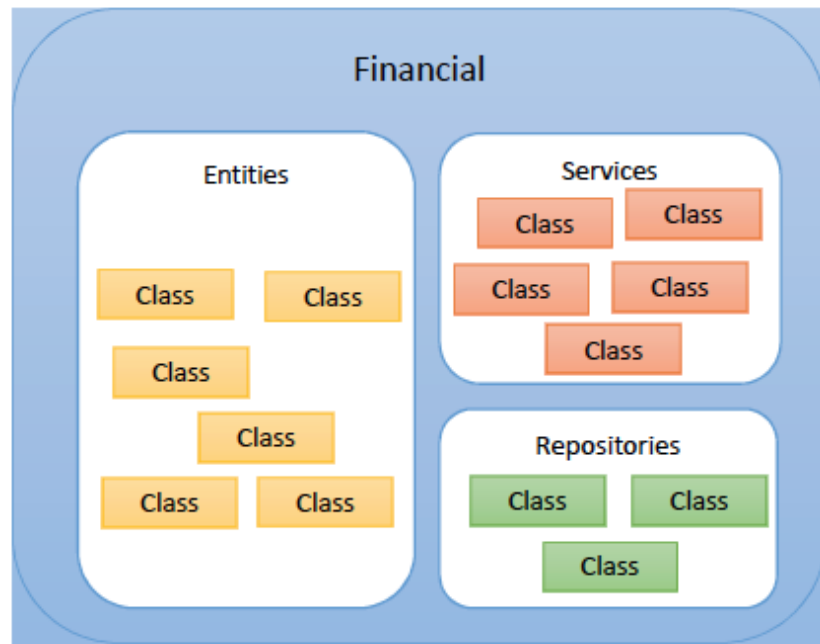
Estrutura de uma aplicação C# .NET

- Namespace é um agrupamento lógico de classes relacionadas.



Estrutura de uma aplicação C# .NET

- Assembly (dll ou exe) é um agrupamento físico de classes relacionadas (build).



Estrutura de uma aplicação C# .NET

- Aplicação é um agrupamento de assemblies relacionados. (Aplicação = Solution e Assembly = Project).

