

# Noções básicas de



integração contínua e boas práticas de programação em geral

Matheus Neder – Arquiteto de Software BHS/Olé Consignado

<https://github.com/matheusneder> | <http://linkedin.com/in/matheus-neder-66b16a16/>



# C# (C Sharp)

- Anders Hejlsberg (Turbo Pascal e Delphi)
- Microsoft .NET Framework
- ECMA 334 - ISO/IEC 23270
  - Mono
  - dotGNU
  - Portable.NET

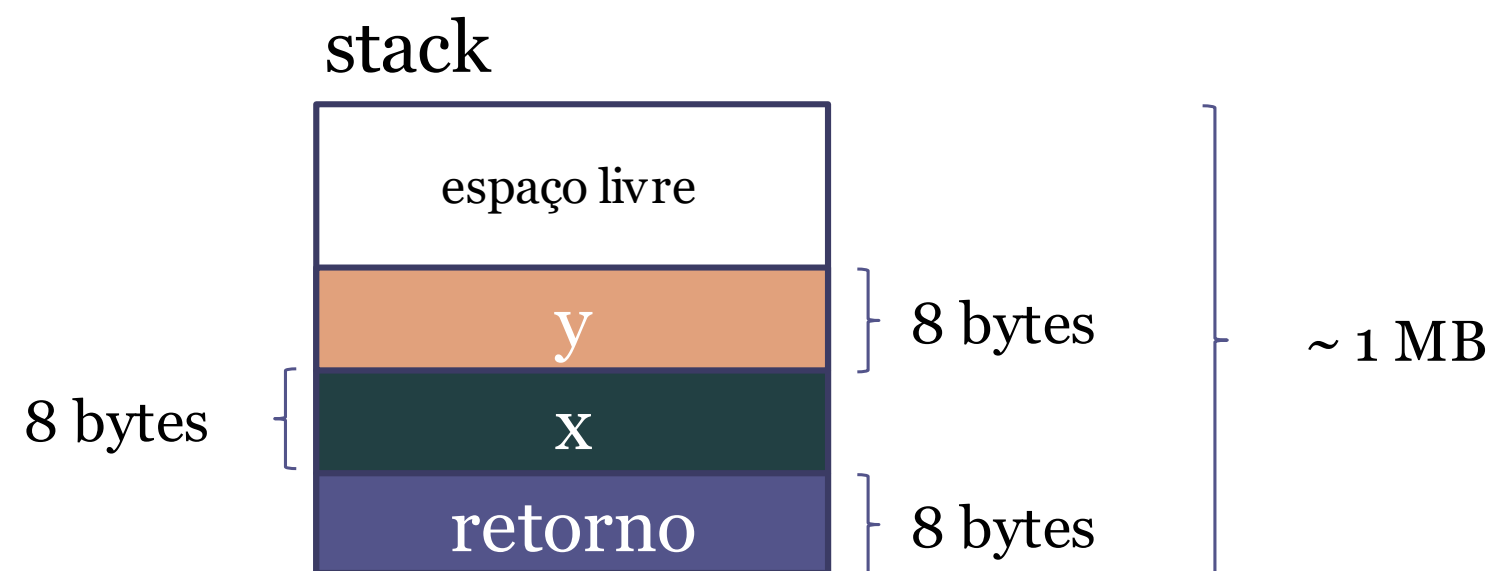


# Revisão relâmpago

- Tipos de valor e tipos de referencia e gerenciamento de memória
- Classes e estruturas e seus membros
  - Campos (atributos)
  - Métodos
  - Propriedades
- Herança e polimorfismo
- Interfaces

# Gerenciamento de memória

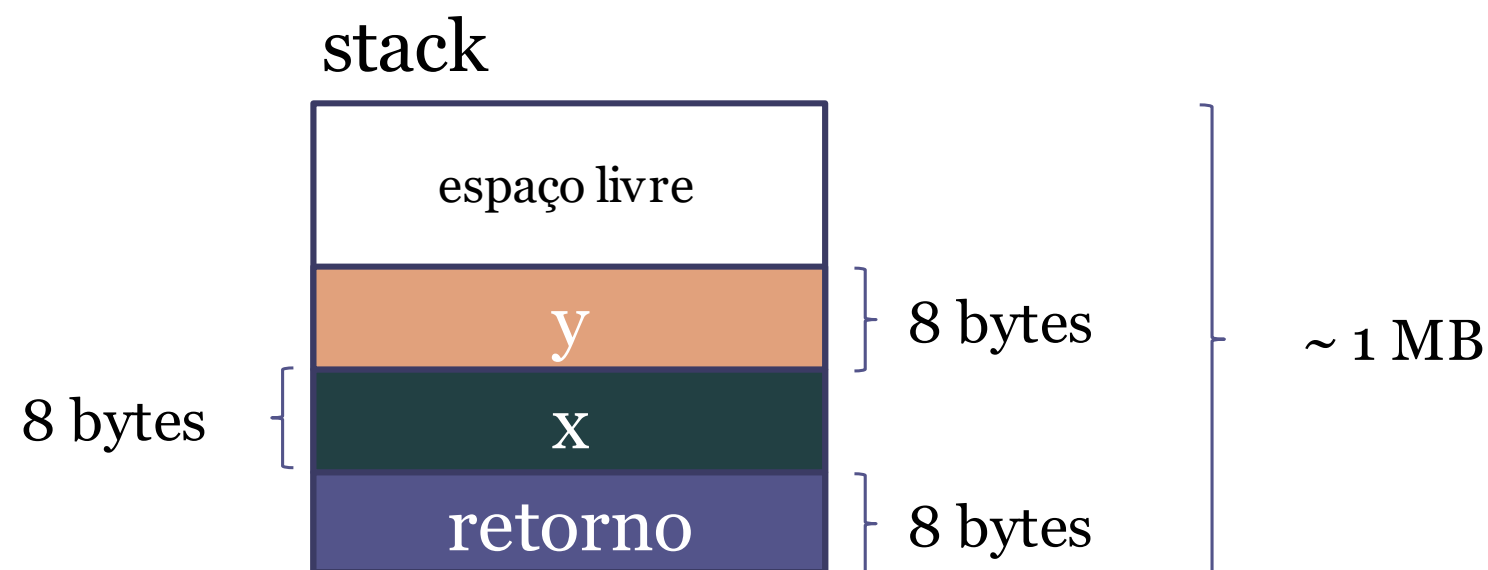
```
public long Soma(long x, long y)
{
    return x + y;
}
```



Observação: 8 bytes = sizeof(long)

# Gerenciamento de memória

Tipos de valor (value types) são armazenados na pilha (stack).

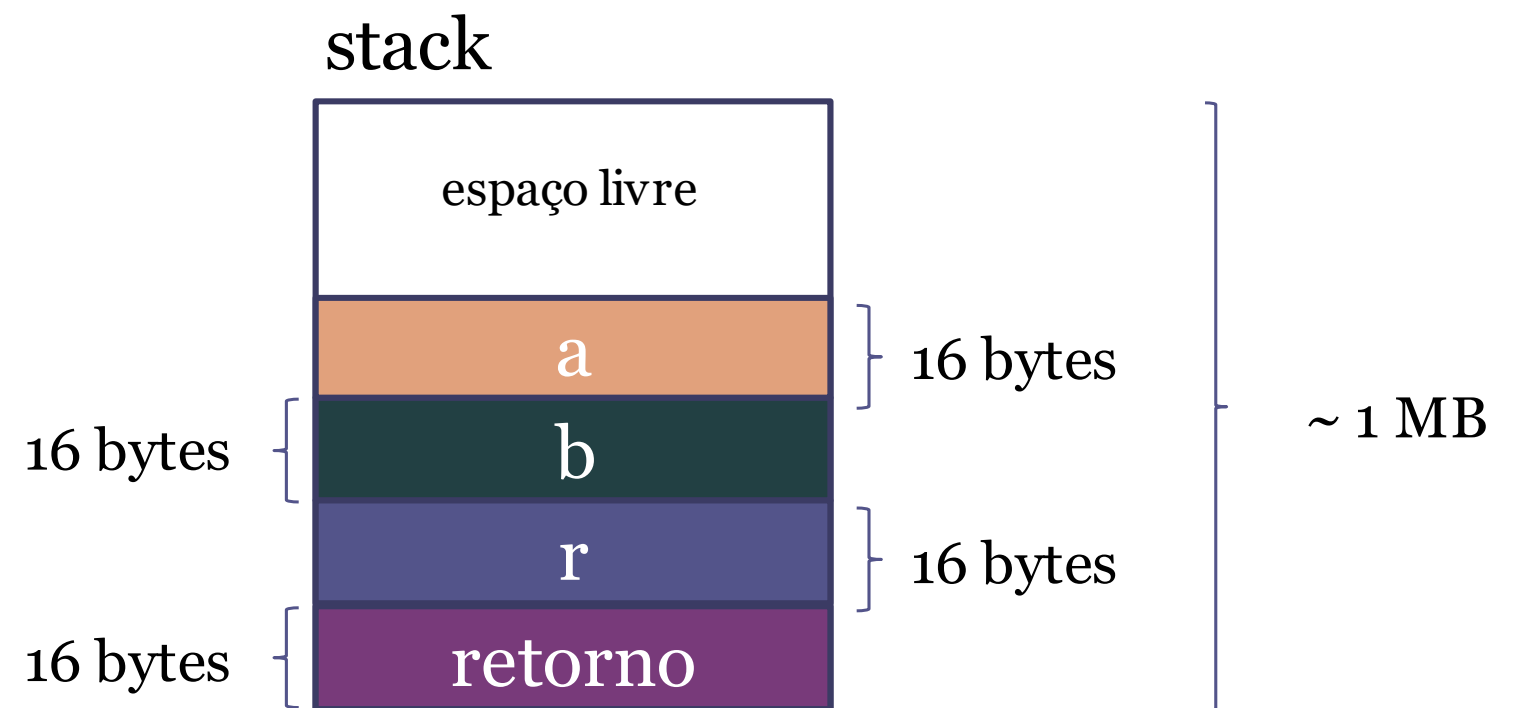


Observação: 8 bytes = sizeof(long)

# Gerenciamento de memória

```
public Coord Midpoint(Coord a, Coord b)
{
    Coord r = new Coord();
    ...
    return r;
}
```

```
struct Coord
{
    public double x;
    public double y;
}
```



Observação: 16 bytes = sizeof(Coord)

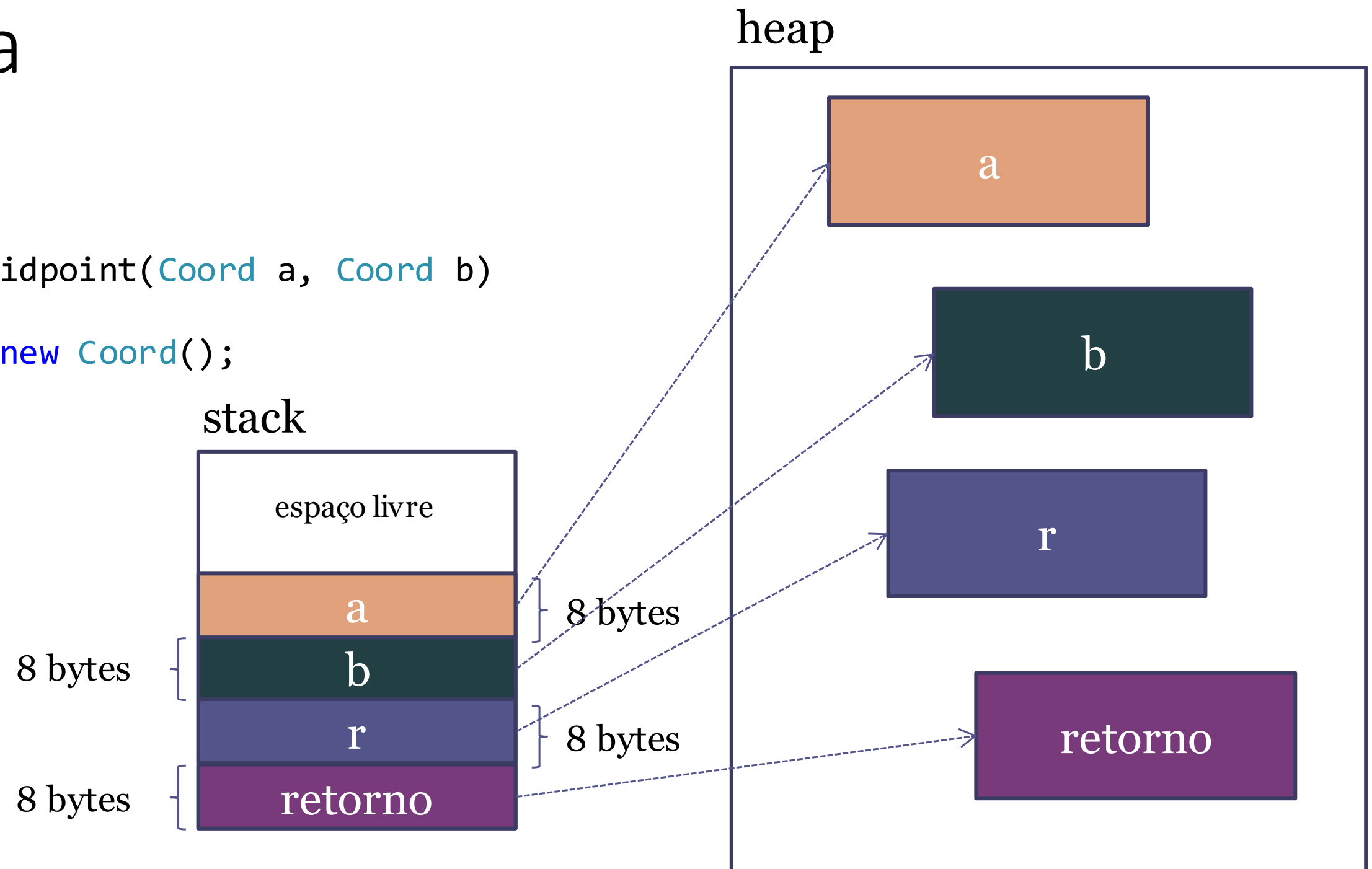
# Gerenciamento de memória

Considere uma nova versão do tipo *Coord*:

```
class Coord
{
    public double x;
    public double y;
}
```

# Memória

```
public Coord Midpoint(Coord a, Coord b)
{
    Coord r = new Coord();
    ...
    return r;
}
```

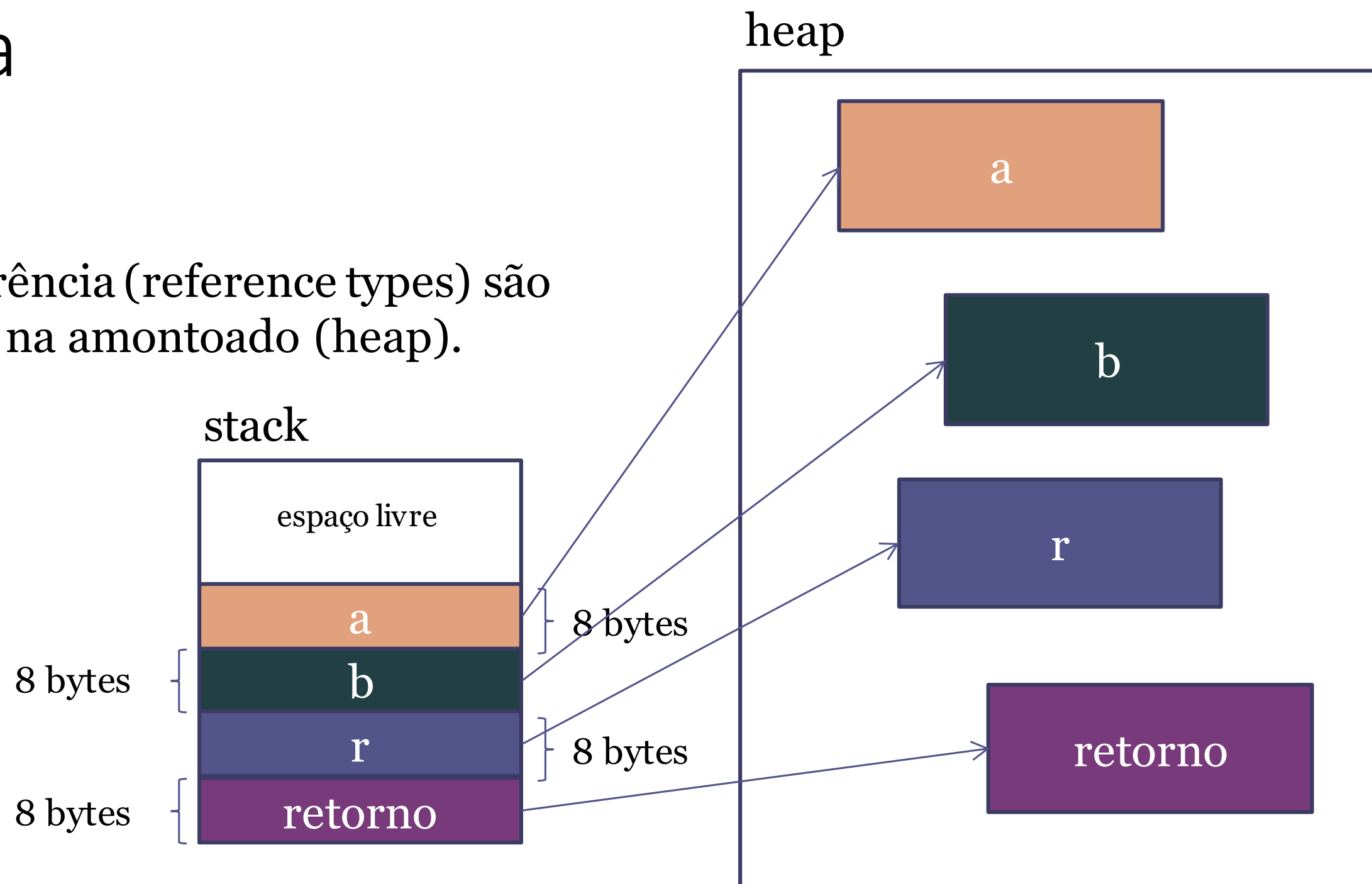


Observação: 8 bytes = tamanho do ponteiro



# Memória

Tipos de referência (reference types) são armazenados na amontoado (heap).



Observação: 8 bytes = tamanho do ponteiro

# Gerenciamento de memória

```
struct Coord
{
    public double x;
    public double y;
}

public void Swap(Coord coord)
{
    double aux = coord.x;
    coord.x = coord.y;
    coord.y = aux;
}
```

```
Coord coord = new Coord();
coord.x = 10.0;
coord.y = 5.0;

Swap(coord);

Console.WriteLine($"x: {coord.x}");
Console.WriteLine($"y: {coord.y}");
```

?

# Gerenciamento de memória

```
class Coord
{
    public double x;
    public double y;
}

public void Swap(Coord coord)
{
    double aux = coord.x;
    coord.x = coord.y;
    coord.y = aux;
}
```

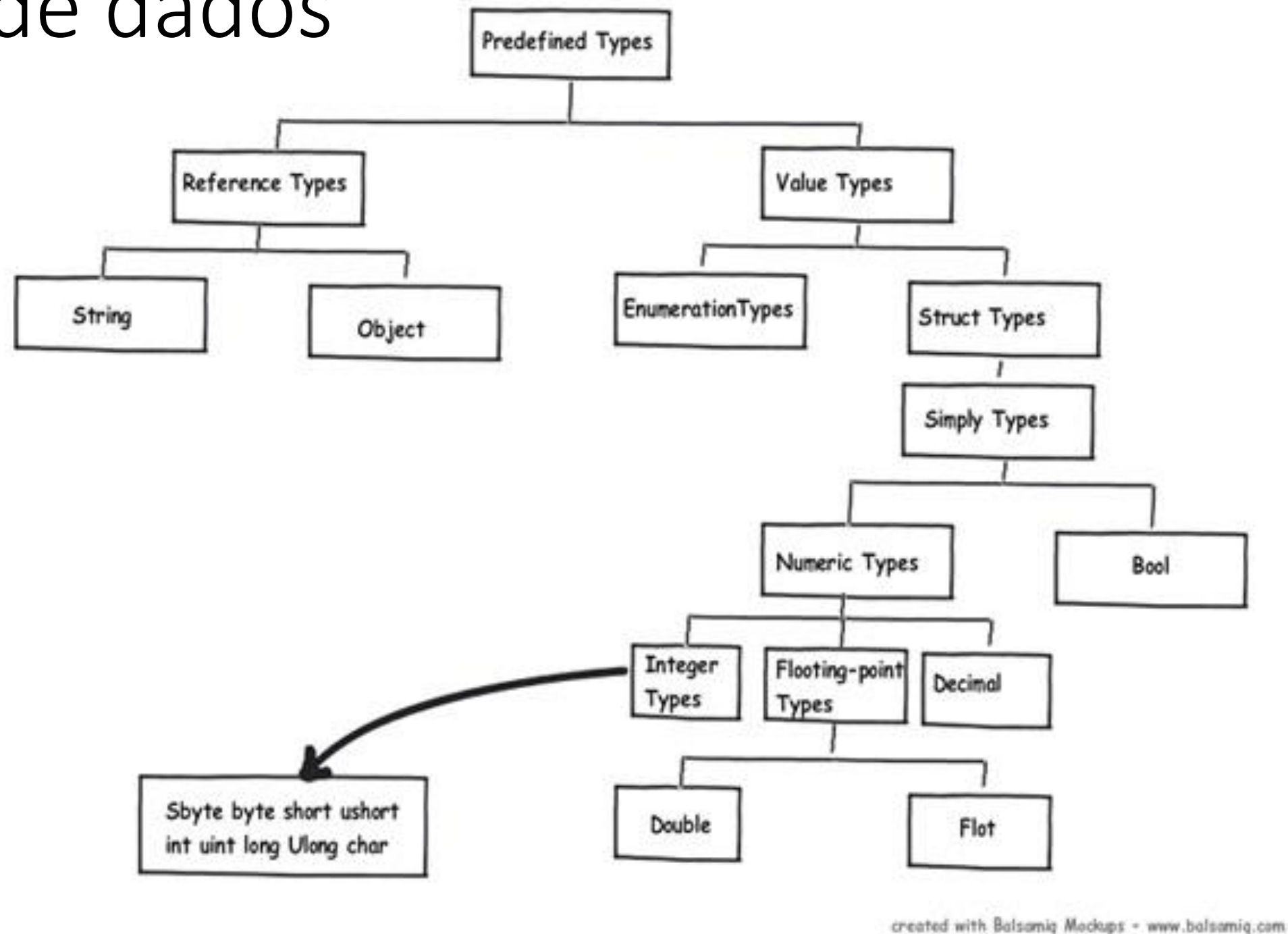
```
Coord coord = new Coord();
coord.x = 10.0;
coord.y = 5.0;

Swap(coord);

Console.WriteLine($"x: {coord.x}");
Console.WriteLine($"y: {coord.y}");
```

?

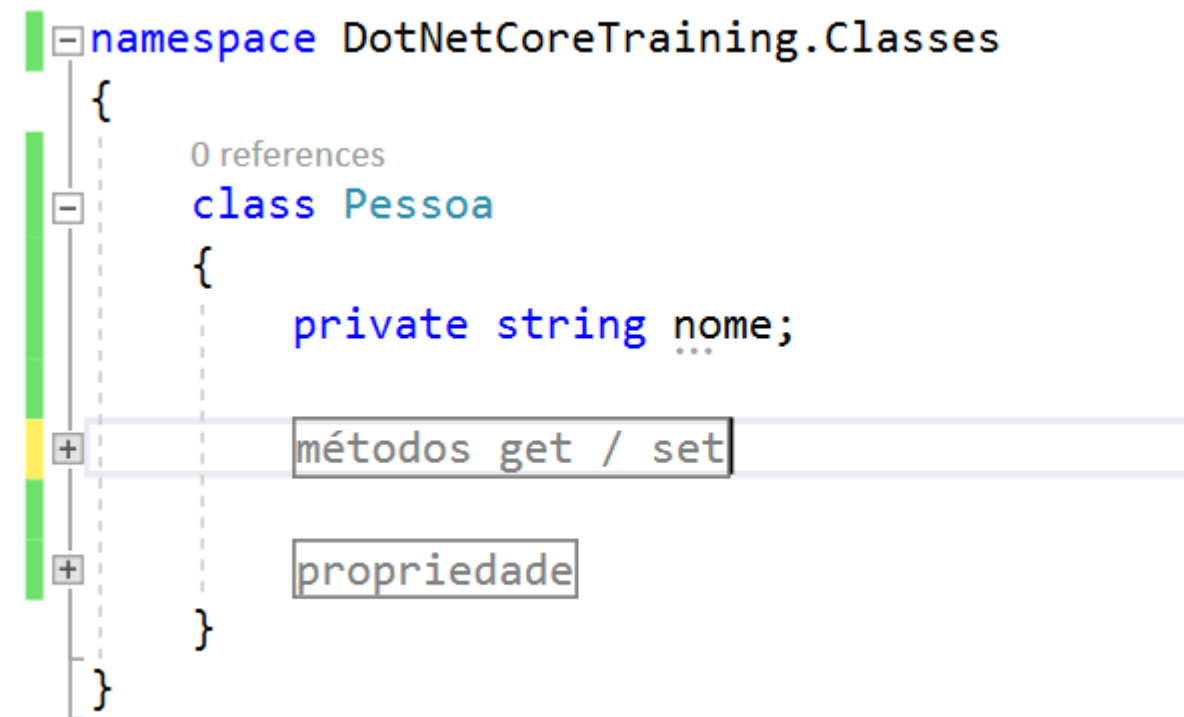
# Tipos de dados



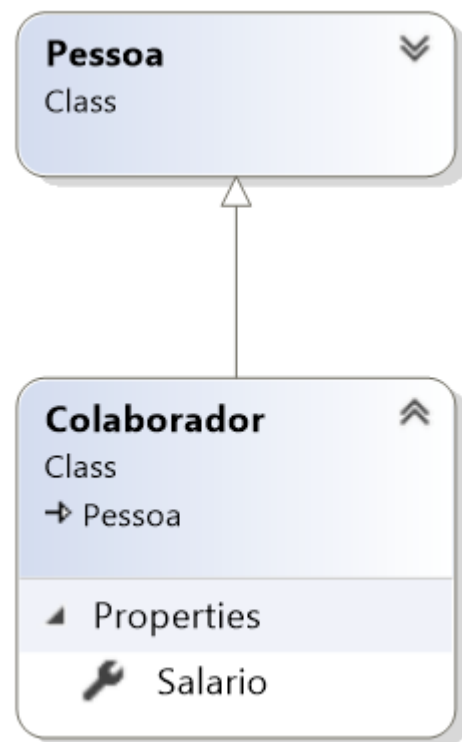
Fonte: <https://www.codeproject.com/Articles/76153/Six-important-NET-concepts-Stack-heap-value-types>

# Classes e estruturas

- Campos (atributos)
- Métodos
- Propriedades



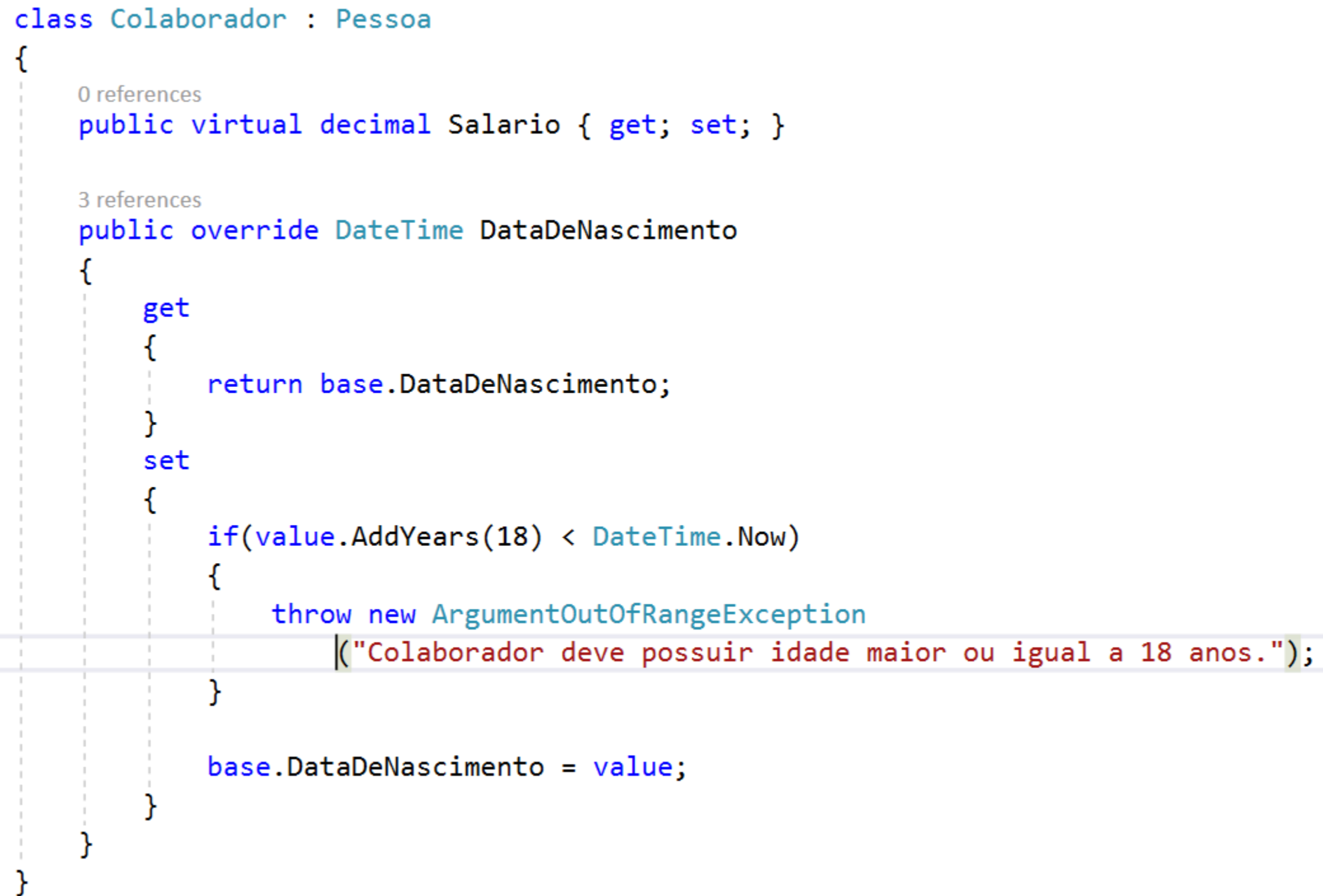
# Herança e polimorfismo



```
namespace DotNetCoreTraining.Classes
{
    0 references
    class Colaborador : Pessoa
    {
        0 references
        public decimal Salario { get; set; }
    }
}
```

Obs.: Herança é possível apenas para tipos de referencia

# Herança e polimorfismo



```
class Colaborador : Pessoa
{
    0 references
    public virtual decimal Salario { get; set; }

    3 references
    public override DateTime DataDeNascimento
    {
        get
        {
            return base.DataDeNascimento;
        }
        set
        {
            if(value.AddYears(18) < DateTime.Now)
            {
                throw new ArgumentOutOfRangeException
                    |("Colaborador deve possuir idade maior ou igual a 18 anos.");
            }

            base.DataDeNascimento = value;
        }
    }
}
```

# Herança e polimorfismo

1 reference

```
public void CorrigirDataDeNascimento(Pessoa pessoa, DateTime novaDataDeNascimento)
{
    pessoa.DataDeNascimento = novaDataDeNascimento;
}
```

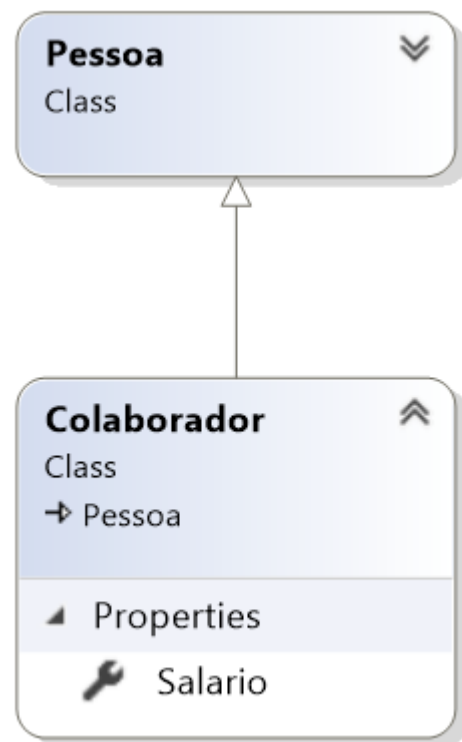
2 references

```
public void Executar()
{
    try
    {
        var joseDaSilva = new Colaborador()
        {
            Nome = "José da Silva",
            DataDeNascimento = DateTime.Parse("1980-04-30"),
            Salario = 500
        };

        CorrigirDataDeNascimento(joseDaSilva, DateTime.Parse("2000-04-30"));
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
        Console.ReadLine();
    }
}
```



# Herança e polimorfismo



```
namespace DotNetCoreTraining.Operacoes
{
    0 references
    class Exemplo
    {
        0 references
        public void Adicionar(Pessoa pessoa)
        {
            // logica para adicionar

            pessoa.Salario;
        }
    }
}
```

# Interfaces

2 references

class RepositorioPessoas

{

0 references

public Pessoa[] ObterPessoas()

{

var query = @"

select top 100 nome

, salario

, data\_de\_nascimento

, tipo

from pessoas";

return Db.Query<Pessoa>(query);

}

1 reference

public void Atualizar(Pessoa pessoa)

{

var query = "update pessoas set ...";

Db.Query(query);

}

}

# Interfaces

1 reference

class Exemplo02

```
{  
    private RepositorioPessoas repositorioDePessoas;  
  
    0 references  
    public Exemplo02(RepositorioPessoas repositorioDePessoas)  
    {  
        this.repositorioDePessoas = repositorioDePessoas;  
    }  
  
    0 references  
    public void CorrigirDataDeNascimento(Pessoa pessoa, DateTime novaDataDeNascimento)  
    {  
        pessoa.DataDeNascimento = novaDataDeNascimento;  
  
        repositorioDePessoas.Atualizar(pessoa);  
    }  
}
```

**RepositorioPessoas**

Class

Methods

Atualizar

ObterPessoas

# Interfaces

1 reference

```
interface IRepositoryPessoas
{
    2 references
    void Atualizar(Pessoa pessoa);
    1 reference
    Pessoa[] ObterPessoas();
}
```

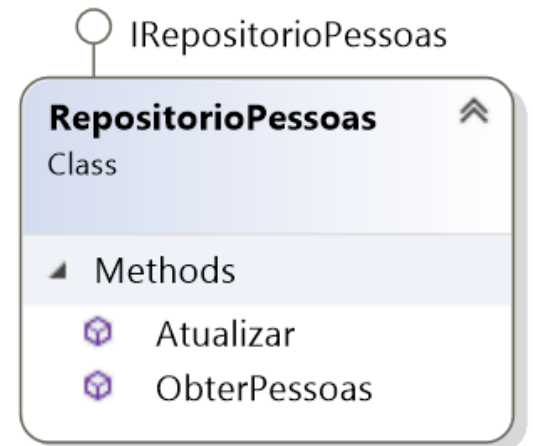
2 references

```
class RepositorioPessoas : IRepositoryPessoas
{
    1 reference
    public Pessoa[] ObterPessoas()
    {
        var query = @"
        select top 100 nome
            , salario
            , data_de_nascimento
            , tipo
        from pessoas";

        return Db.Query<Pessoa>(query);
    }

    2 references
    public void Atualizar(Pessoa pessoa)
    {
        var query = "update pessoas set ...";

        Db.Query(query);
    }
}
```



# Interfaces

```
1 reference
class Exemplo02
{
    private IRepositoryPessoas repositorioDePessoas;

    0 references
    public Exemplo02(IRepositoryPessoas repositorioDePessoas)
    {
        this.repositorioDePessoas = repositorioDePessoas;
    }

    0 references
    public void CorrigirDataDeNascimento(Pessoa pessoa, DateTime novaDataDeNascimento)
    {
        pessoa.DataDeNascimento = novaDataDeNascimento;

        repositorioDePessoas.Atualizar(pessoa);
    }
}
```

## IRepositoryPessoas

Interface

### Methods



Atualizar



ObterPessoas



<https://goo.gl/forms/6UP0BiX2INTol2xx2>



Mono

# Cross platform, open source .NET framework



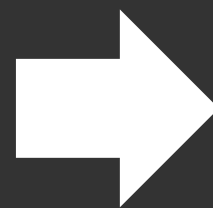
European Computer Manufacturers Association

Standard ECMA-335 (CLI) / ECMA-334 (C#)

Ximian / Novell



Mono



# Cross platform, open source .NET framework



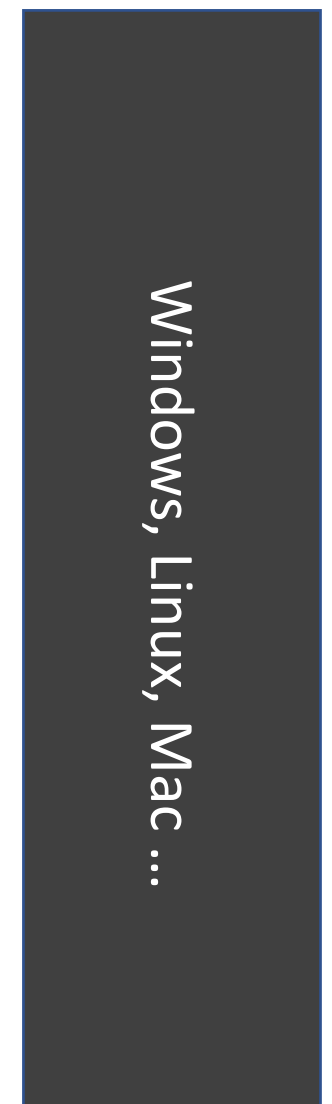
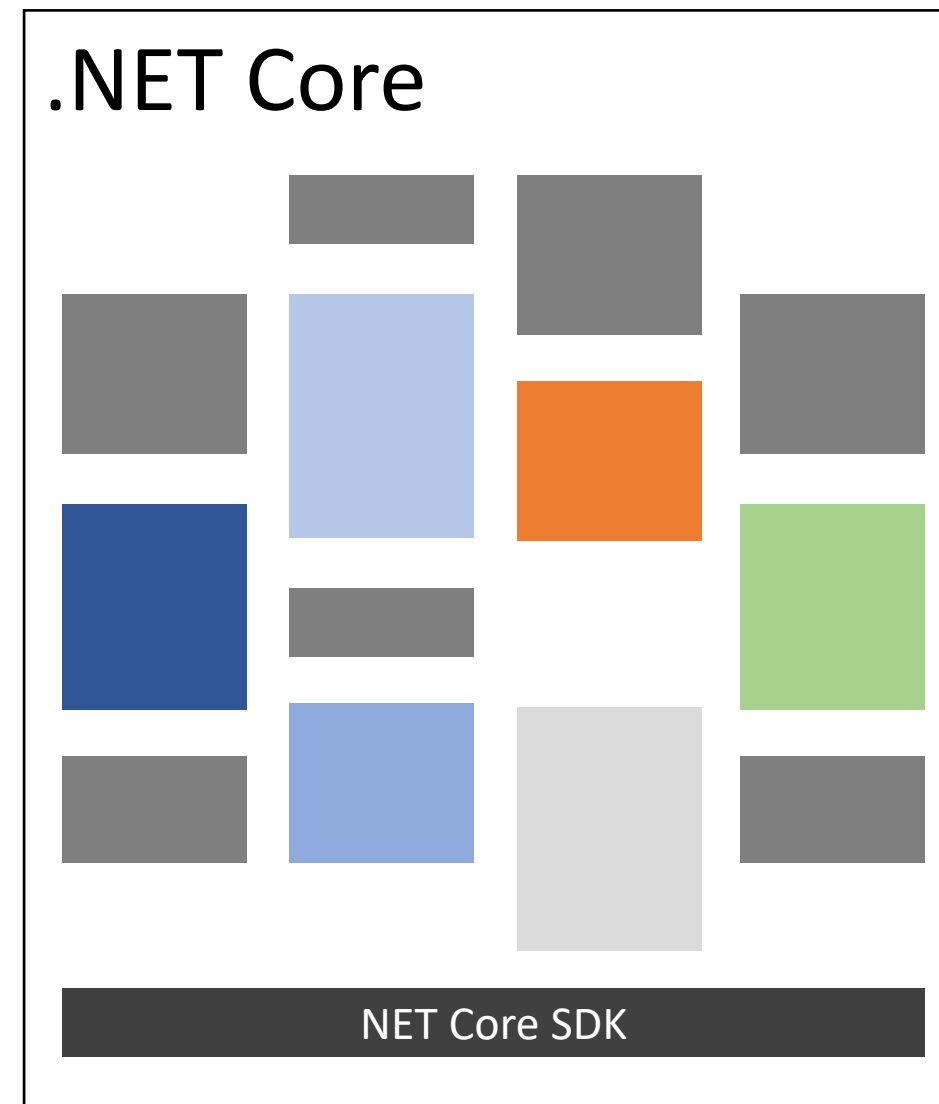
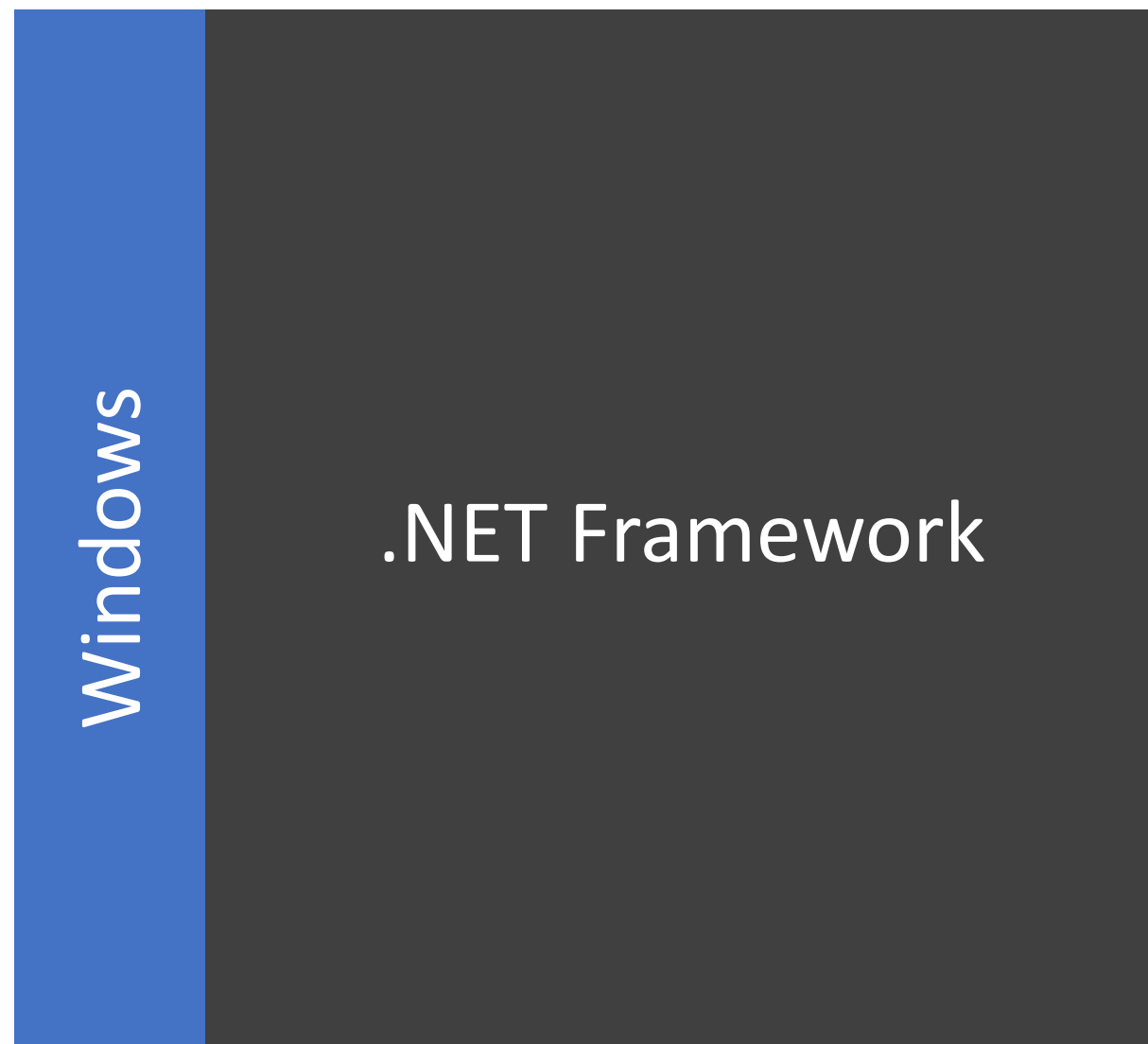
European Computer Manufacturers Association

Standard ECMA-335 (CLI) / ECMA-334 (C#)

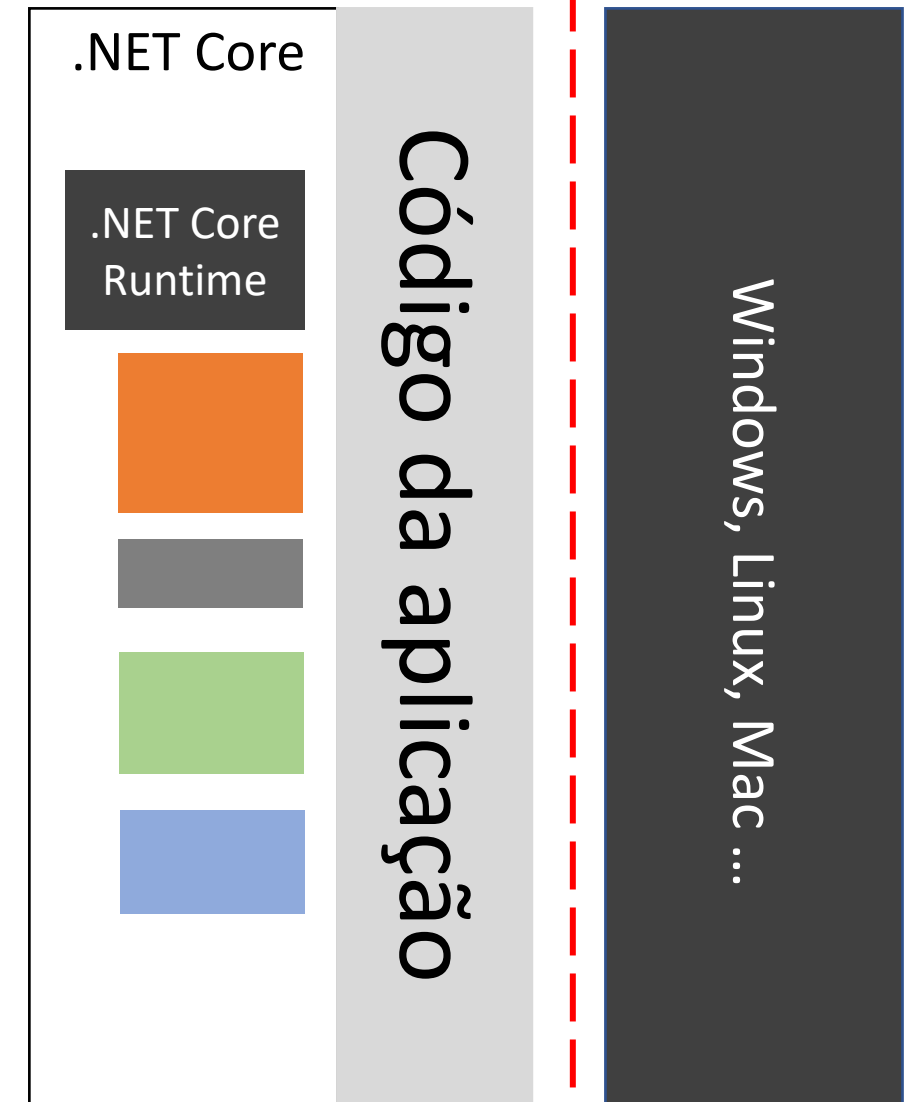
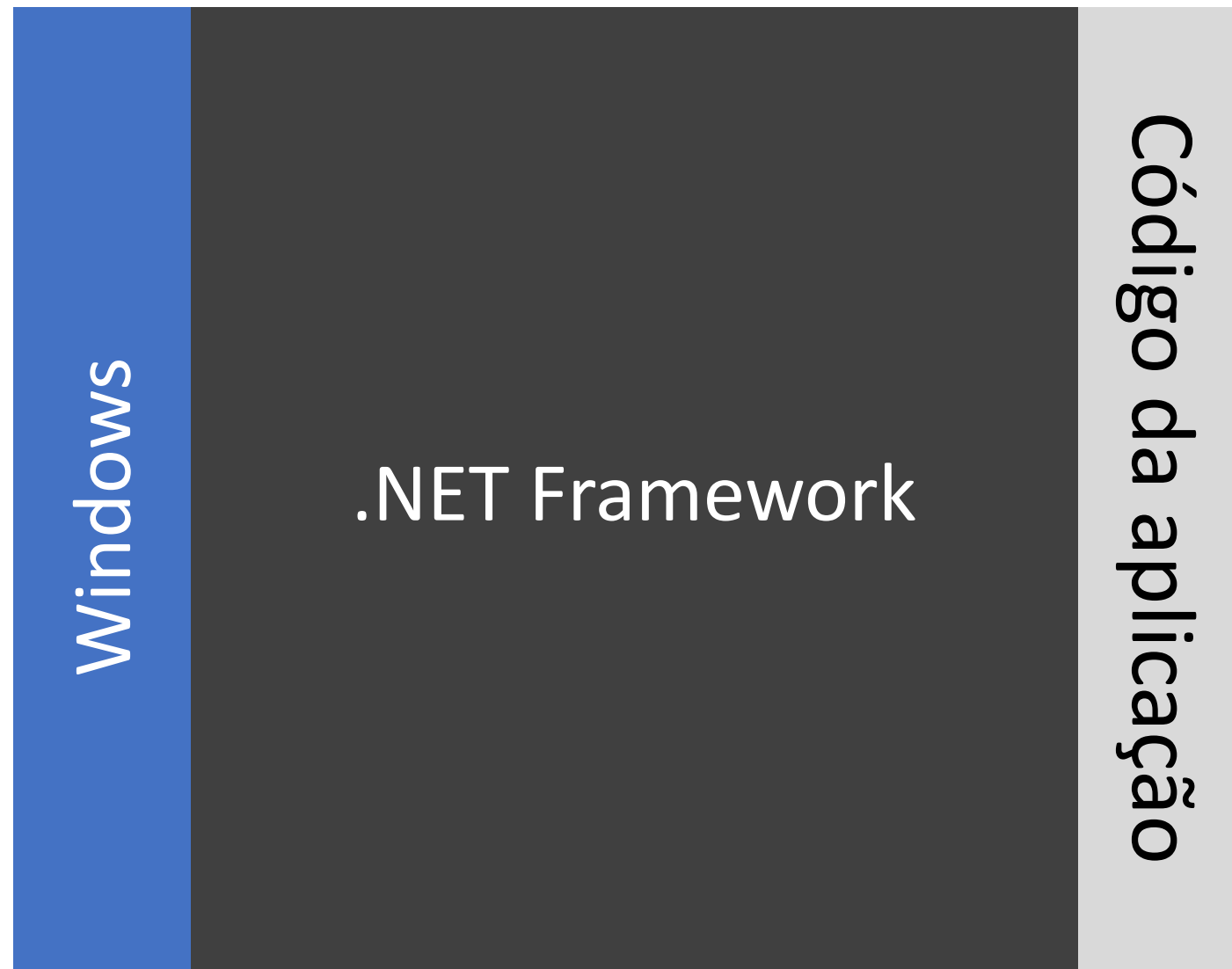
Ximian / Novell



# .NET Core vs .NET Framework



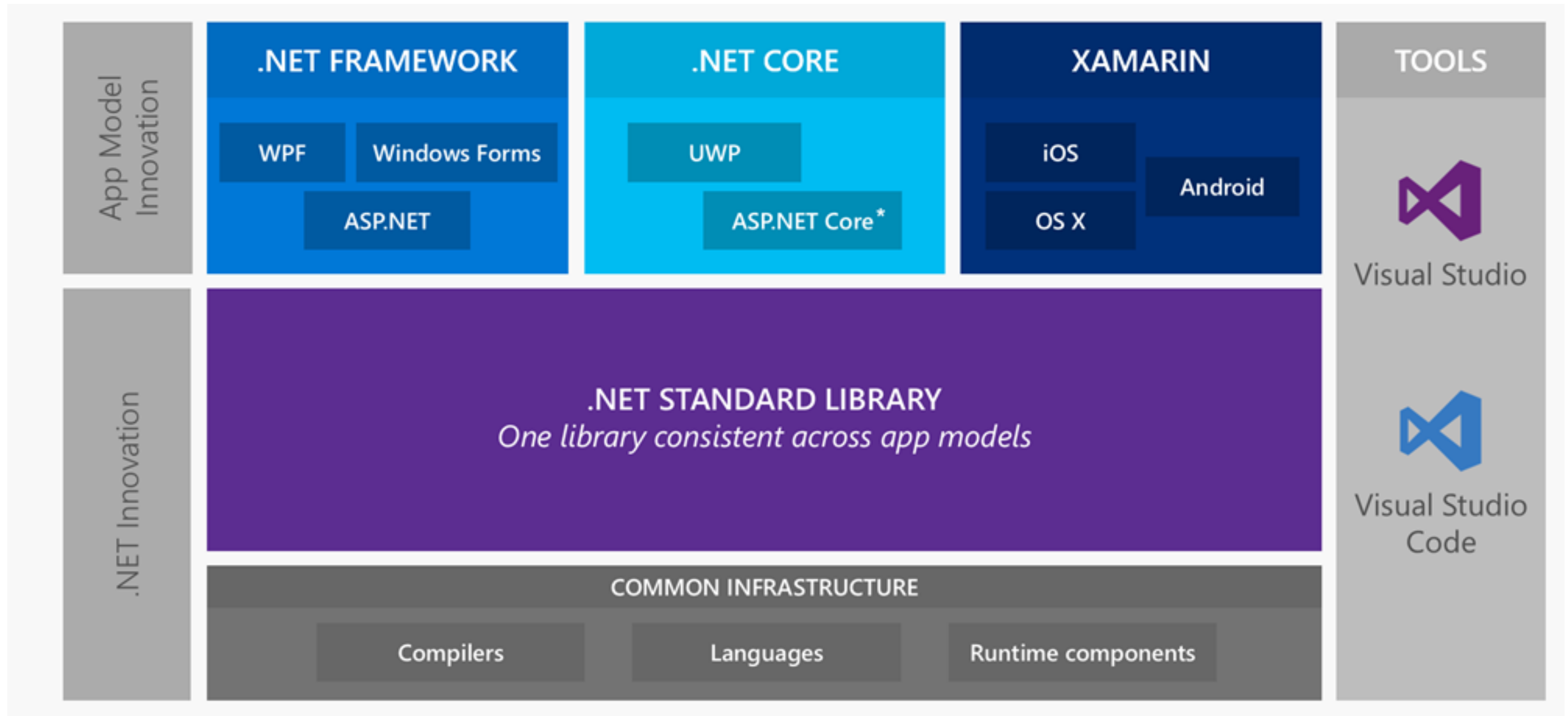
# .NET Core vs .NET Framework



# .NET Standard



# .NET Standard



<https://www.slideshare.net/dotnet18/moving-forward-with-aspnet-core>

# .NET Standard

| <b>.NET Standard</b>   | <a href="#"><u>1.0</u></a> | <a href="#"><u>1.1</u></a> | <a href="#"><u>1.2</u></a> | <a href="#"><u>1.3</u></a> | <a href="#"><u>1.4</u></a> | <a href="#"><u>1.5</u></a> | <a href="#"><u>1.6</u></a> | <a href="#"><u>2.0</u></a> |
|------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| <b>.NET Core</b>       | 1.0                        | 1.0                        | 1.0                        | 1.0                        | 1.0                        | 1.0                        | 1.0                        | 2.0                        |
| <b>.NET Framework</b>  | 4.5                        | 4.5                        | 4.5.1                      | 4.6                        | 4.6.1                      | 4.6.1                      | 4.6.1                      | 4.6.1                      |
| <b>Mono</b>            | 4.6                        | 4.6                        | 4.6                        | 4.6                        | 4.6                        | 4.6                        | 4.6                        | 5.4                        |
| <b>Xamarin.iOS</b>     | 10.0                       | 10.0                       | 10.0                       | 10.0                       | 10.0                       | 10.0                       | 10.0                       | 10.14                      |
| <b>Xamarin.Mac</b>     | 3.0                        | 3.0                        | 3.0                        | 3.0                        | 3.0                        | 3.0                        | 3.0                        | 3.8                        |
| <b>Xamarin.Android</b> | 7.0                        | 7.0                        | 7.0                        | 7.0                        | 7.0                        | 7.0                        | 7.0                        | 8.0                        |
| <b>UWP</b>             | 10.0                       | 10.0                       | 10.0                       | 10.0                       | 10.0                       | 10.0.16                    | 10.0.16                    | 10.0.16                    |
| <b>Windows</b>         | 8.0                        | 8.0                        | 8.1                        |                            |                            |                            |                            |                            |

<https://github.com/dotnet/standard/blob/master/docs/versions.md>

# .NET Core

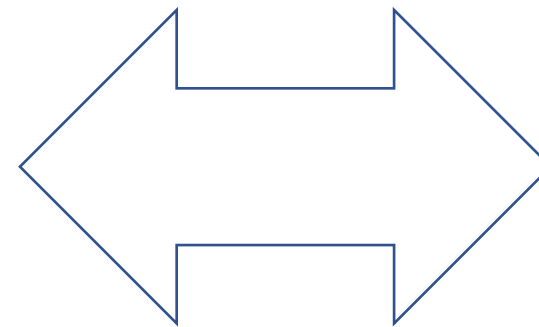
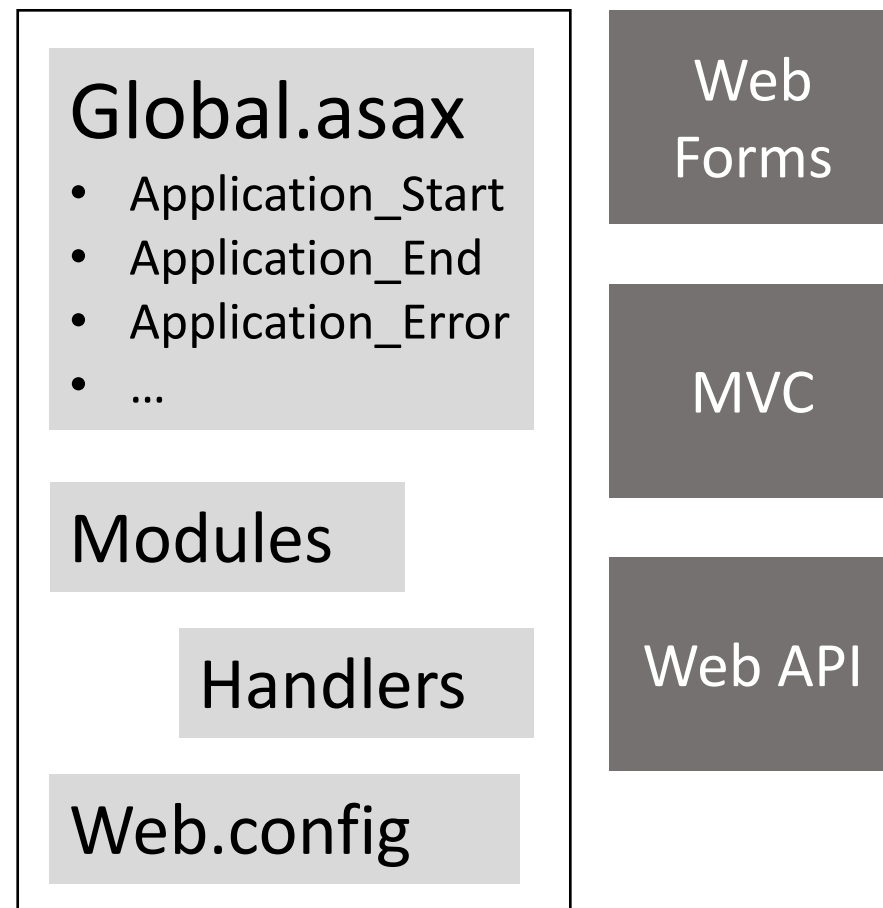
- Plataforma de desenvolvimento de propósitos gerais
- Executa sobre Windows, Linux, macOS, nuvem ou dispositivos embarcados
- Open source
- Compatível com .NET Framework, Xamarin e Mono via .NET Standard.

# ASP.NET Core



O que mudou?

# ASP.NET Core



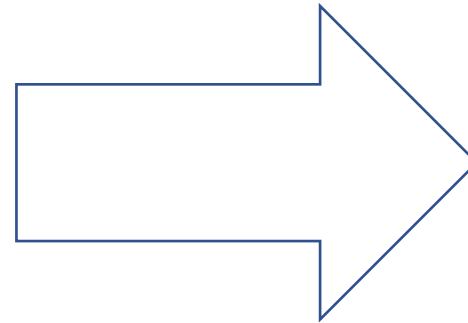


# ASP.NET Core

ASP  
.NET

MVC

Web API 2

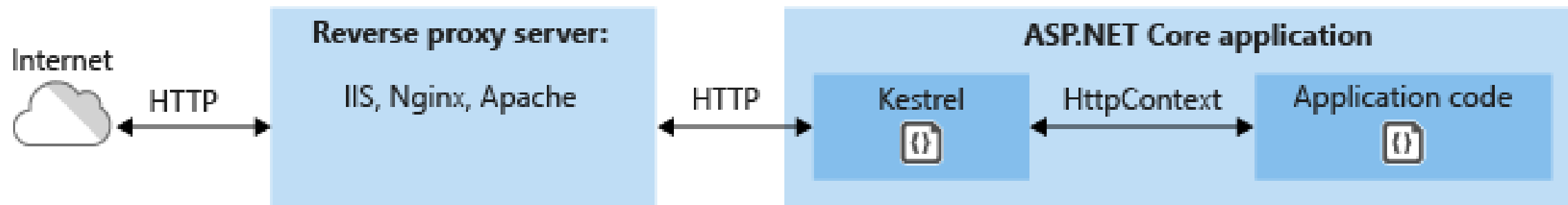


ASP.NET  
Core

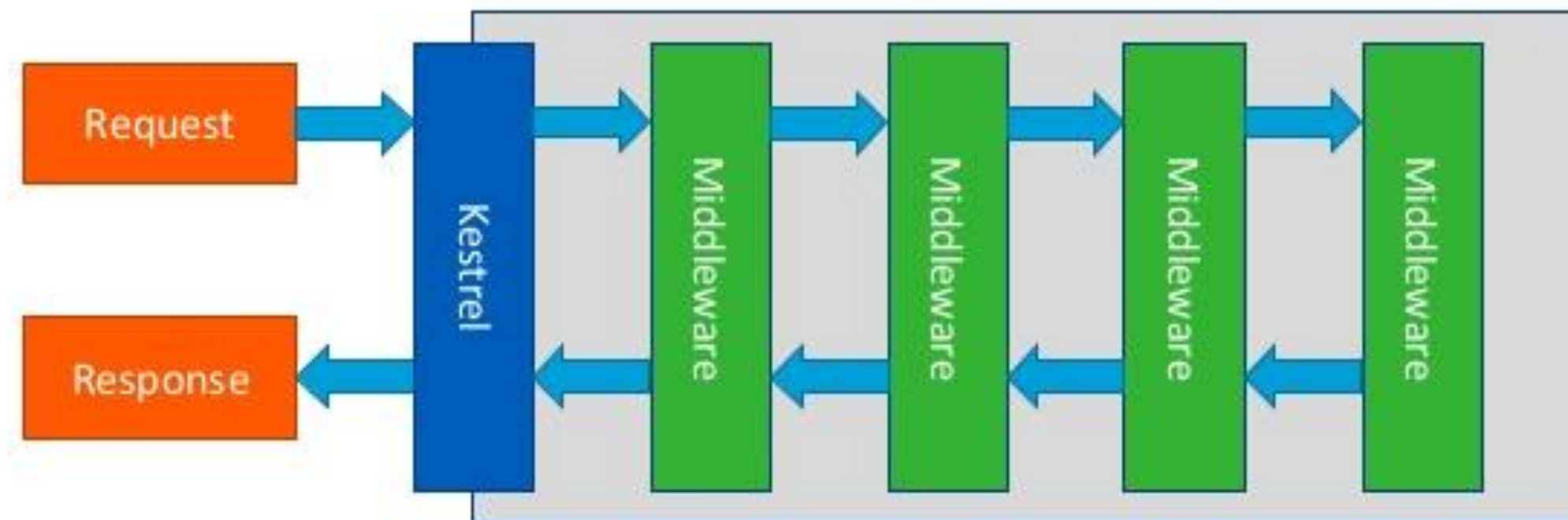
MVC

Escolha ruim para o nome da biblioteca ou estaria a Microsoft tentando ressuscitar o ASP.NET MVC?

# ASP.NET Core



<https://imasters.com.br/dotnet/configuracao-e-deploy-de-aplicativos-asp-net-core-2-0-no-iis>



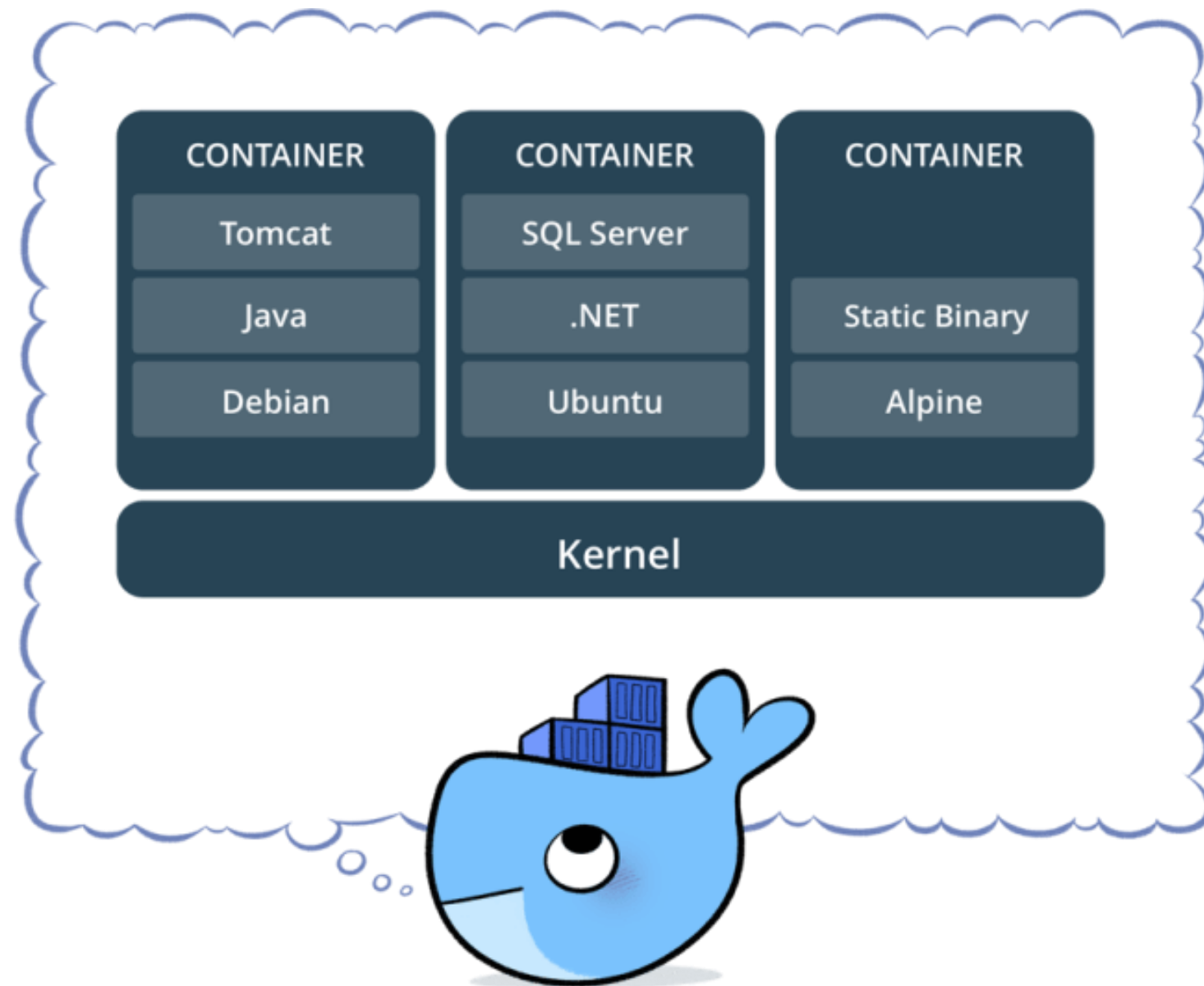
<https://www.slideshare.net/Avanade-Nederland/introduction-to-aspnet-core>

# hands-on



<http://espacojacyra.com.br/mao-na-massa-colaboradoras-participam-de-treinamento-na-chocolandia/>

# Docker

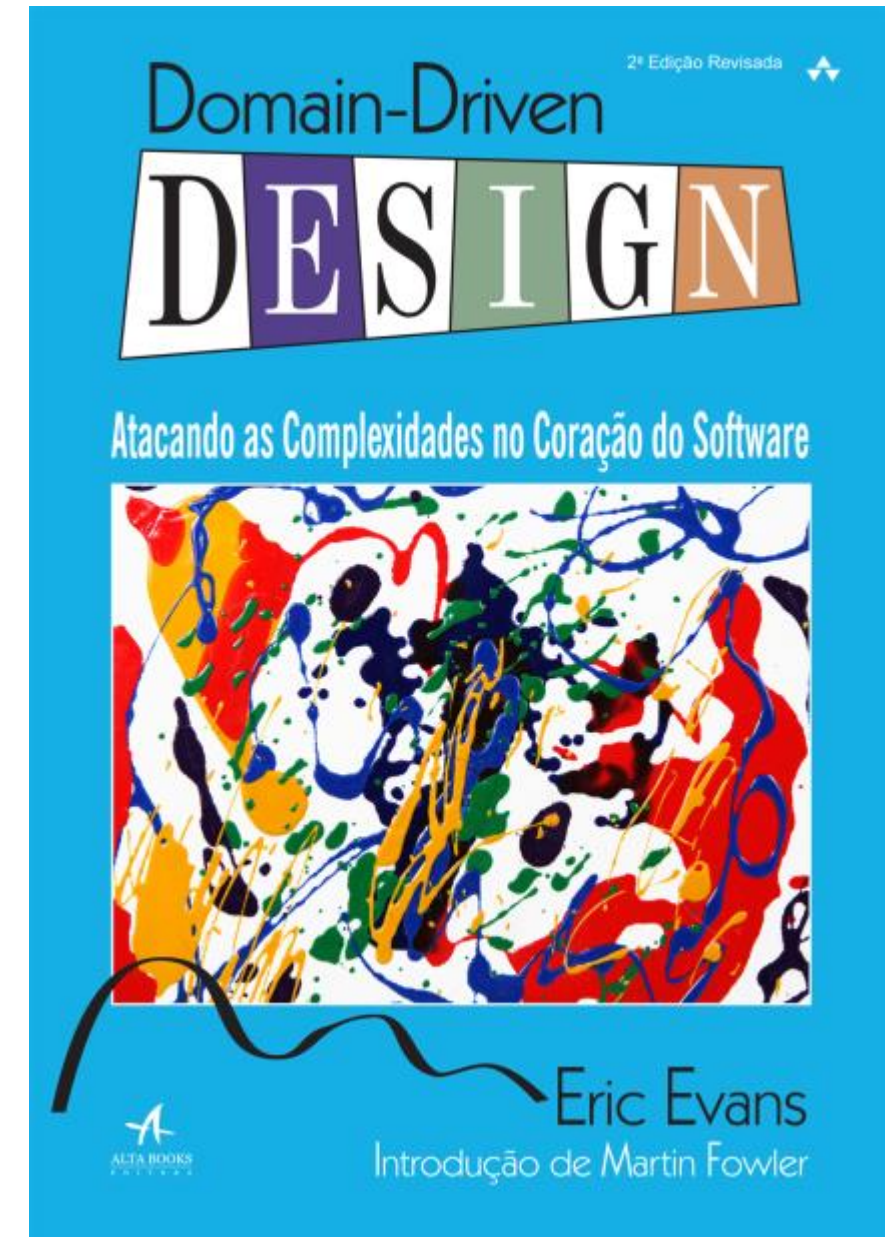


<https://blog.docker.com/2017/08/docker-101-introduction-docker-webinar-recap/>

# DDD – Domain-Driven Design

Eric Evans - 2003

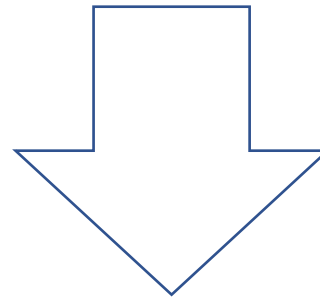
- Definir um modelo de domínio nos termos do negócio;
- Incorporar a terminologia de domínio no código;
- Proteger o conhecimento do domínio contra corrupção por outros domínios, subdomínios técnicos, etc.





# DDD – Domain-Driven Design

Projeto Orientado ao Domínio



Projeto de software orientado ao **negócio**

# DDD – Domain-Driven Design

*"Qualquer tolo consegue escrever código que um computador entenda.*

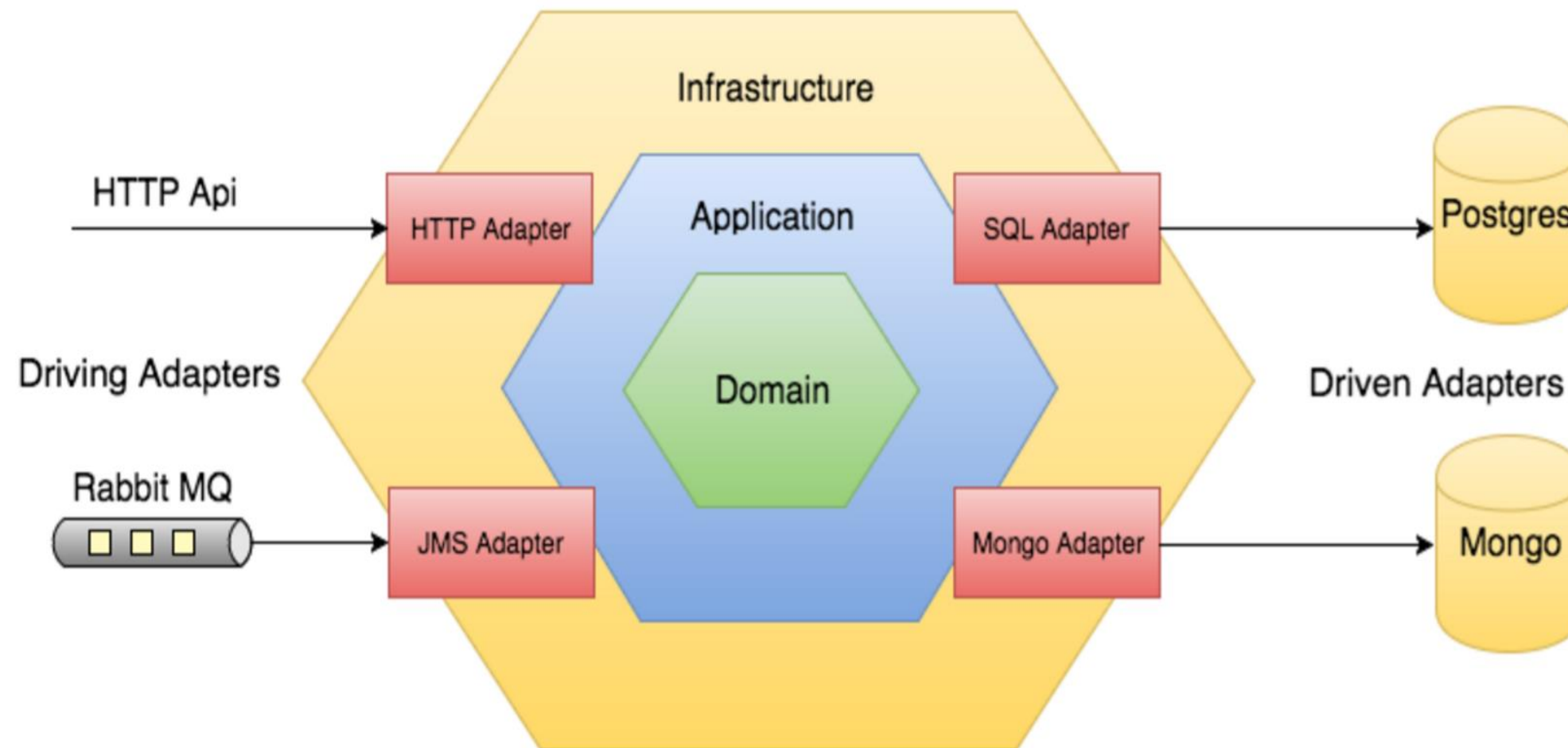
*Bons programadores escrevem código que humanos possam entender."*

(Martin Fowler)

# Arquitetura Hexagonal

Ports and Adapters | Clean Architecture | Onion Architecture

Alistair Cockburn - 2005

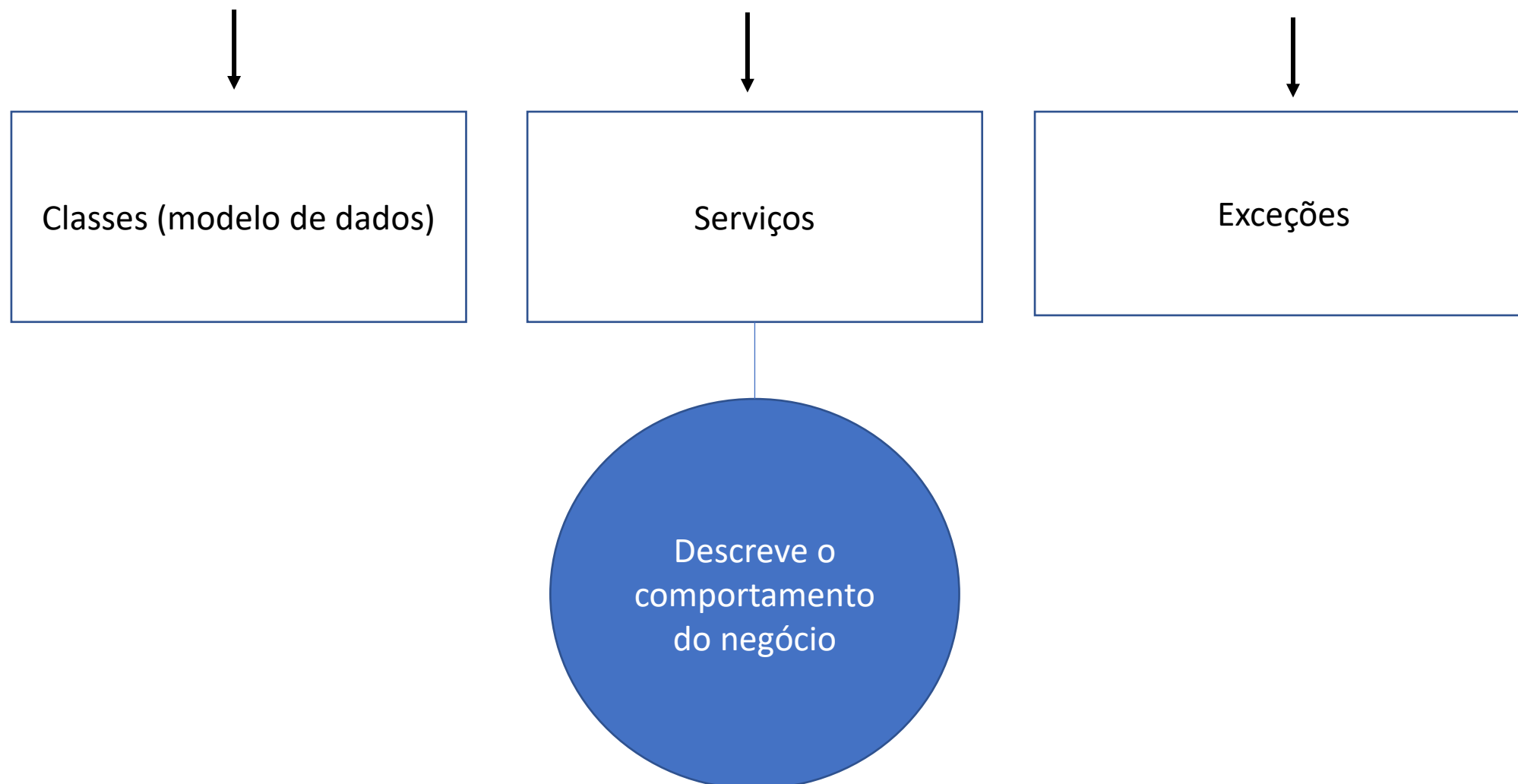


<https://gumtreeuk.github.io/presentations/gumtree-tech-talks/microengines-241116/index.html>



# Domínio

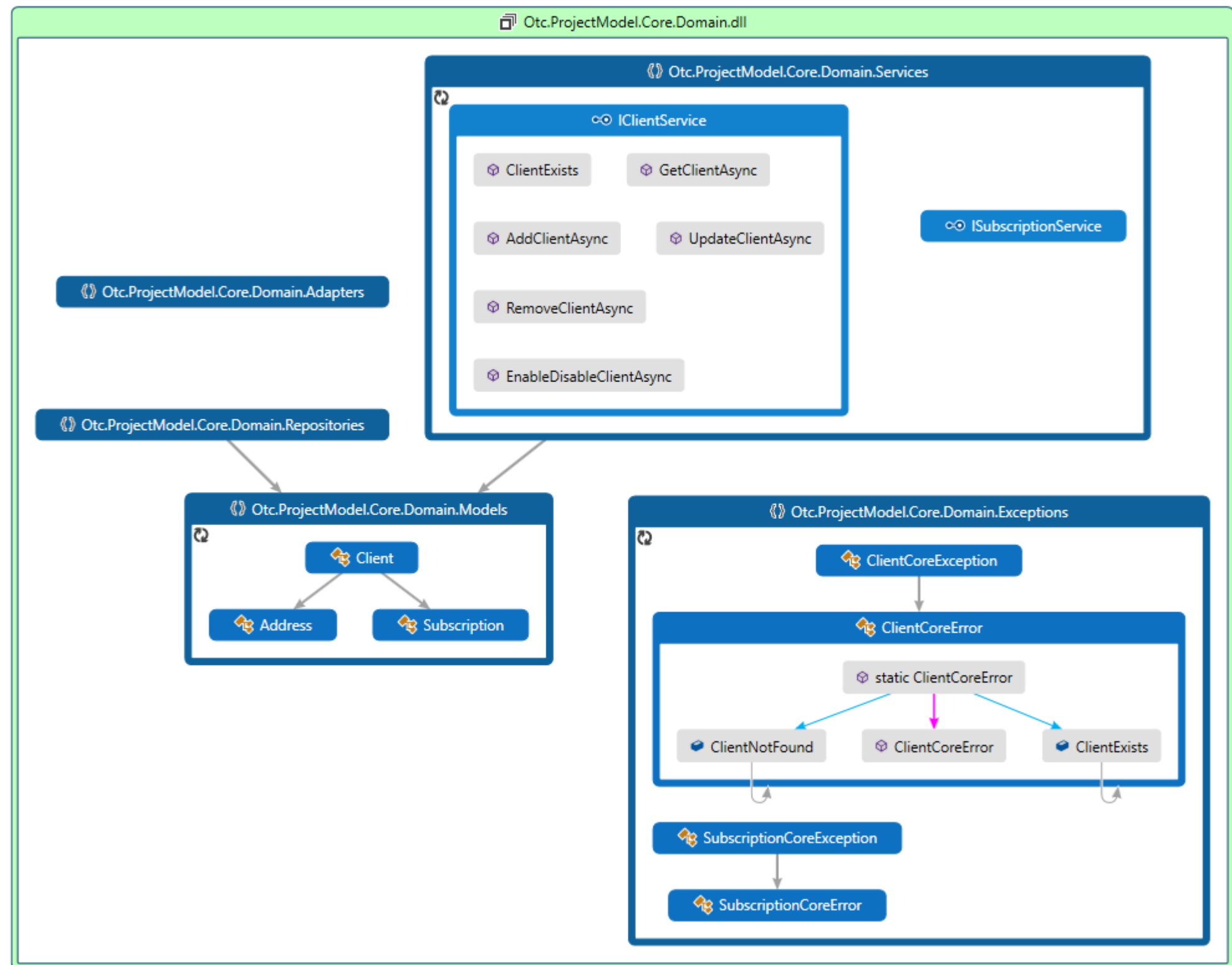
- Definir linguagem ubíqua (onipresente)
  - terminologias utilizados no domínio do negócio



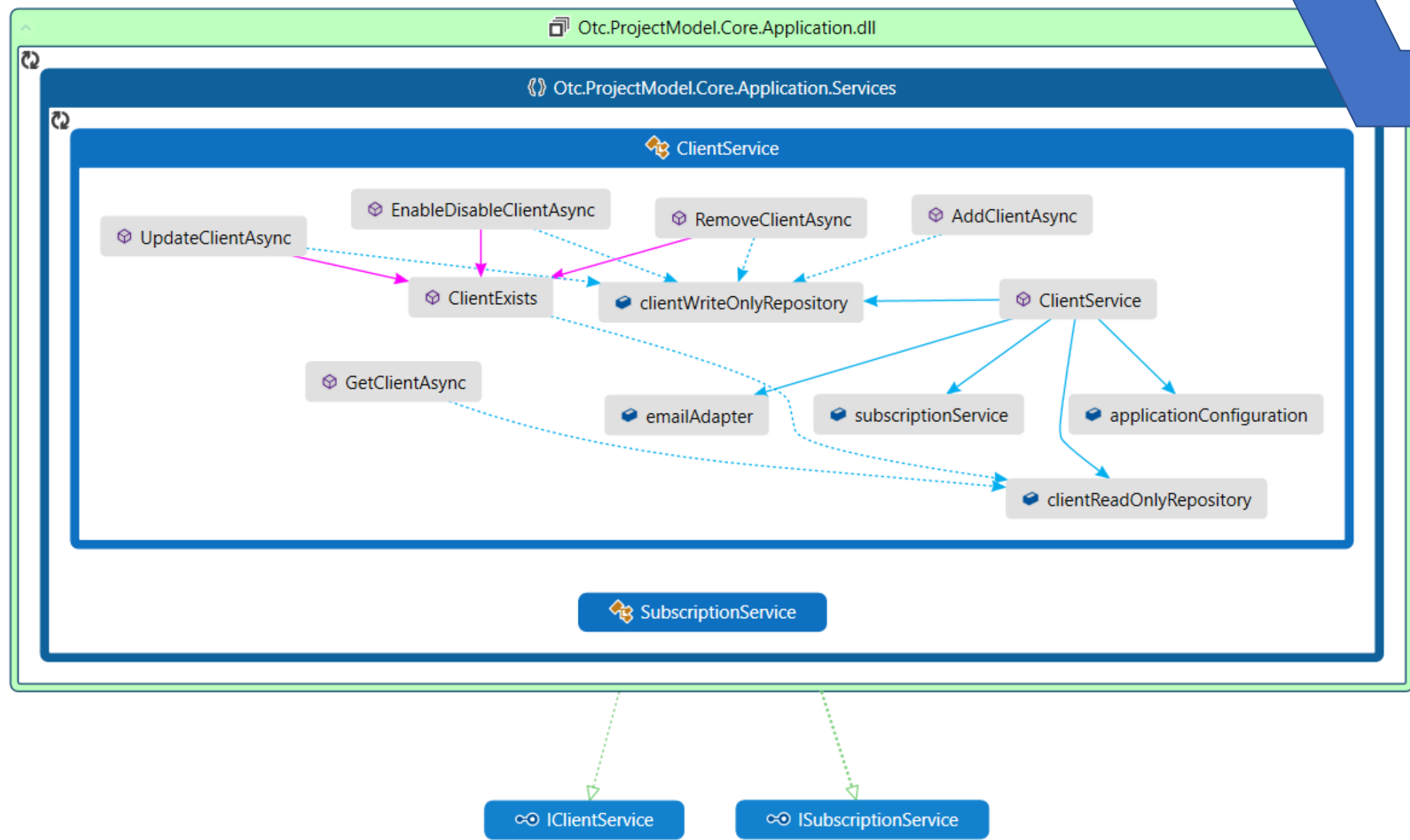
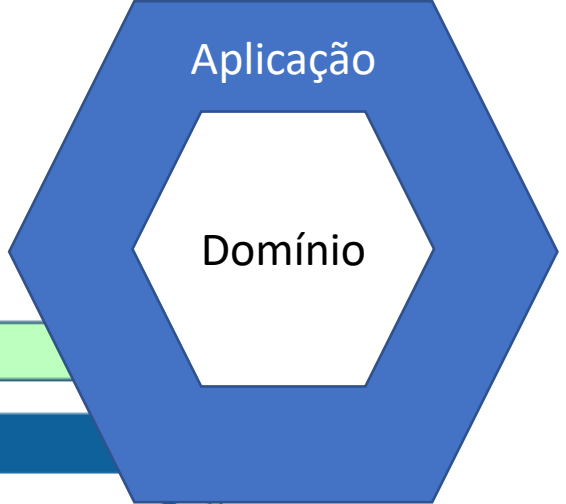
# Domínio

- O domínio deve ser agnóstico à tecnologia
- O domínio não deve depender das demais camadas do sistema.

# Domínio



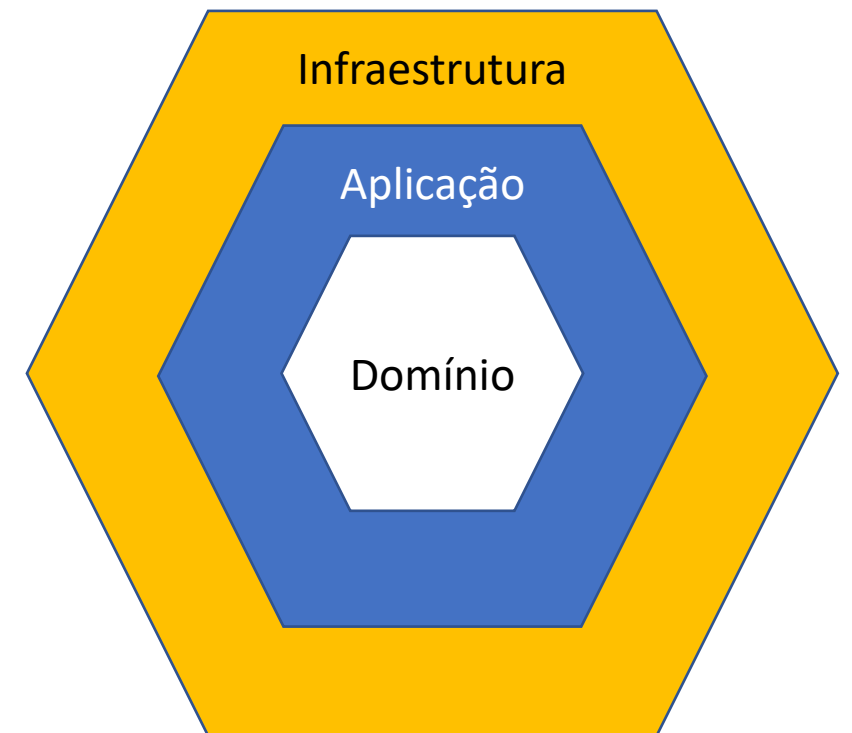
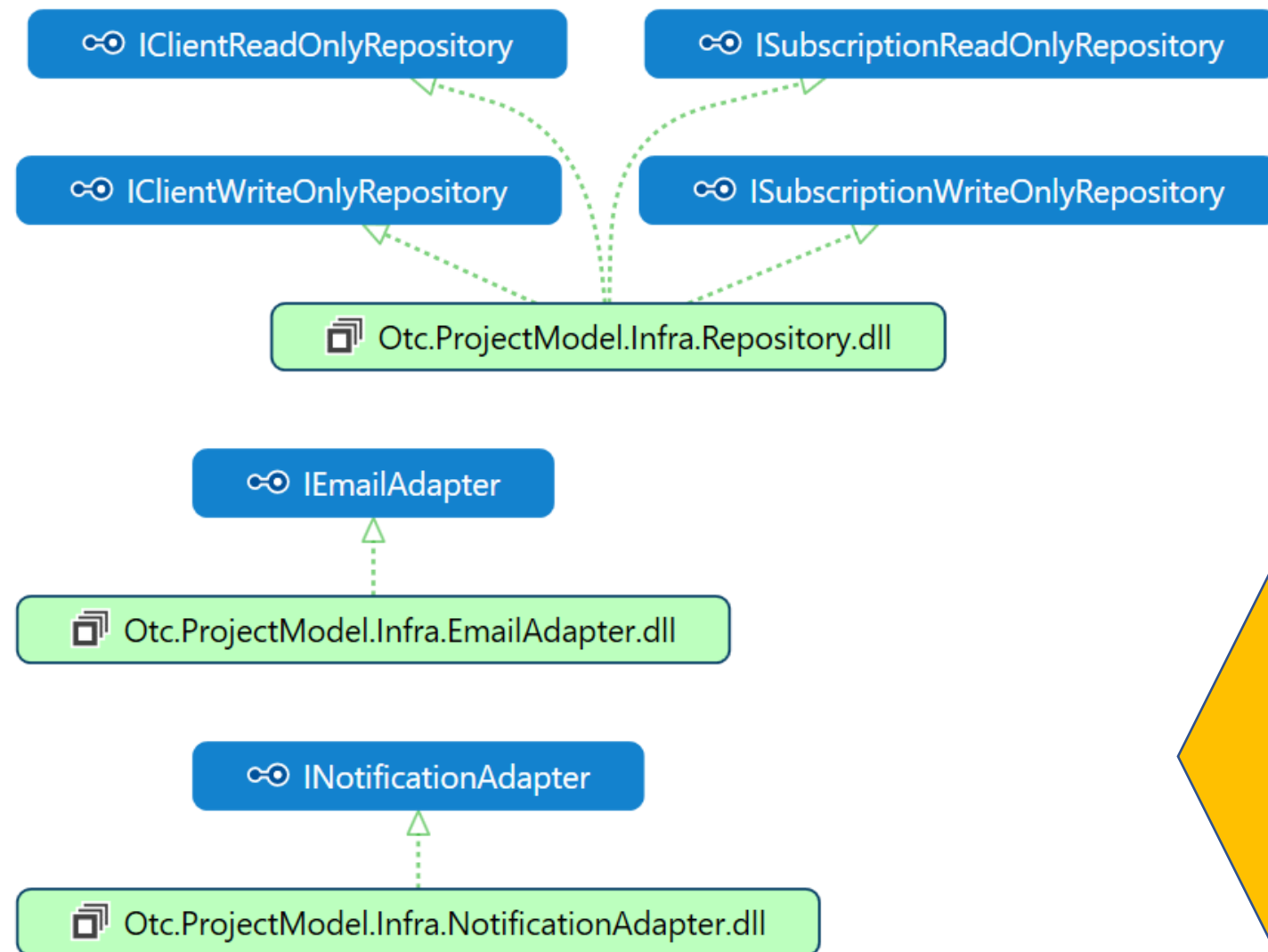
# Aplicação



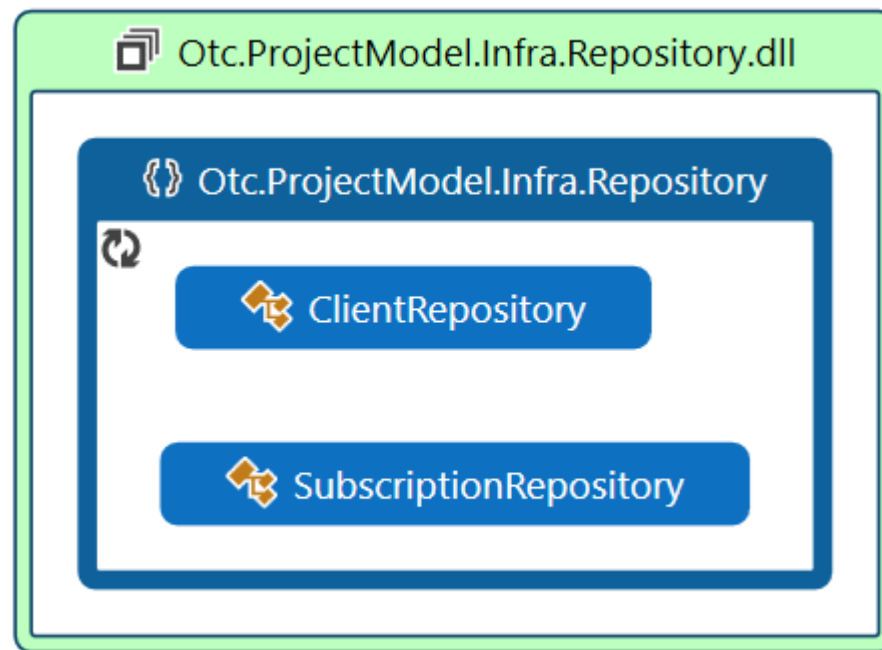
# Aplicação

- Assim como o domínio, a camada de aplicação também deve ser agnóstica à tecnologia.
- A única dependência da camada de Aplicação deve ser a camada de Domínio.

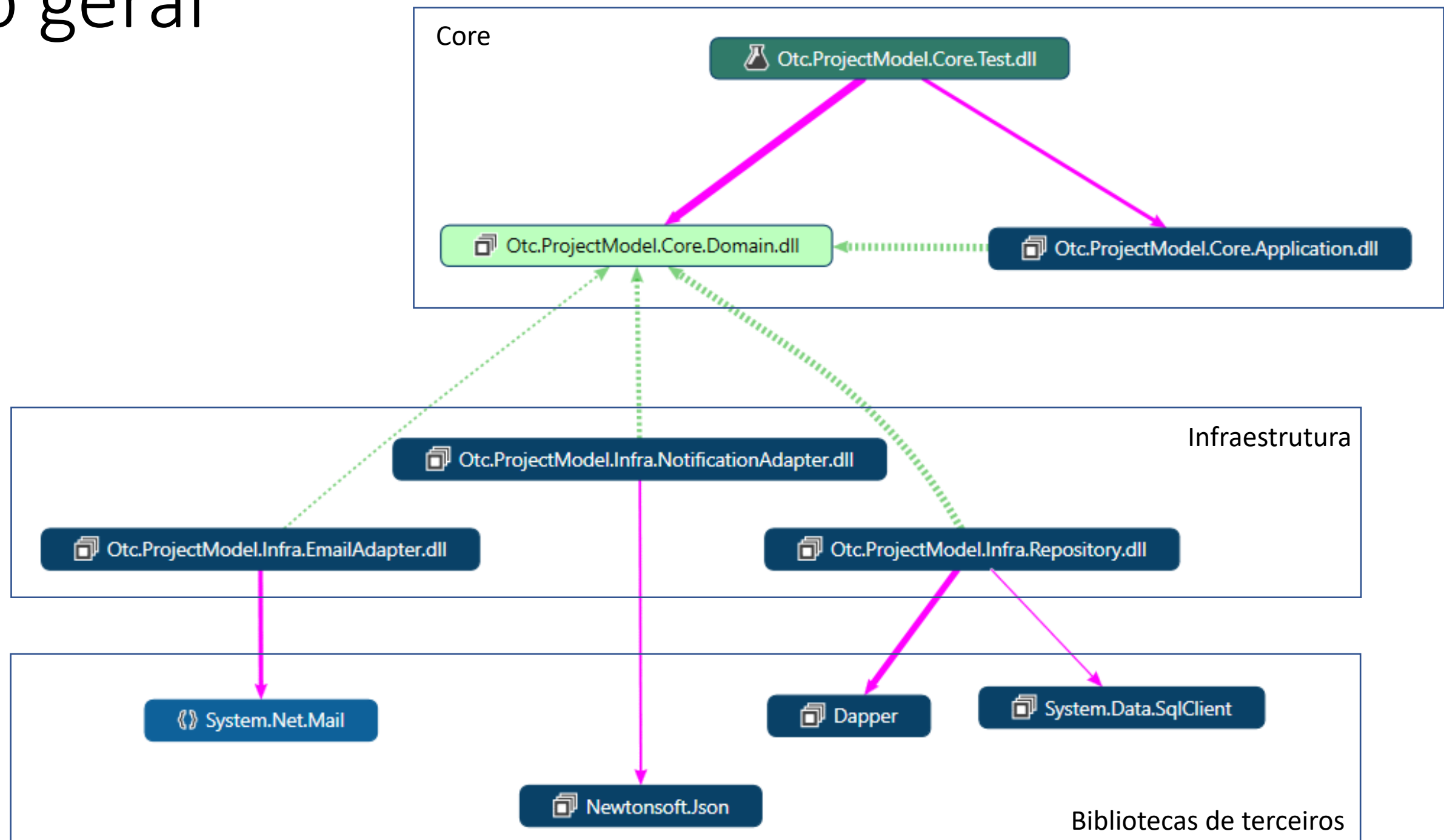
# Infraestrutura / Portas e Adaptadores



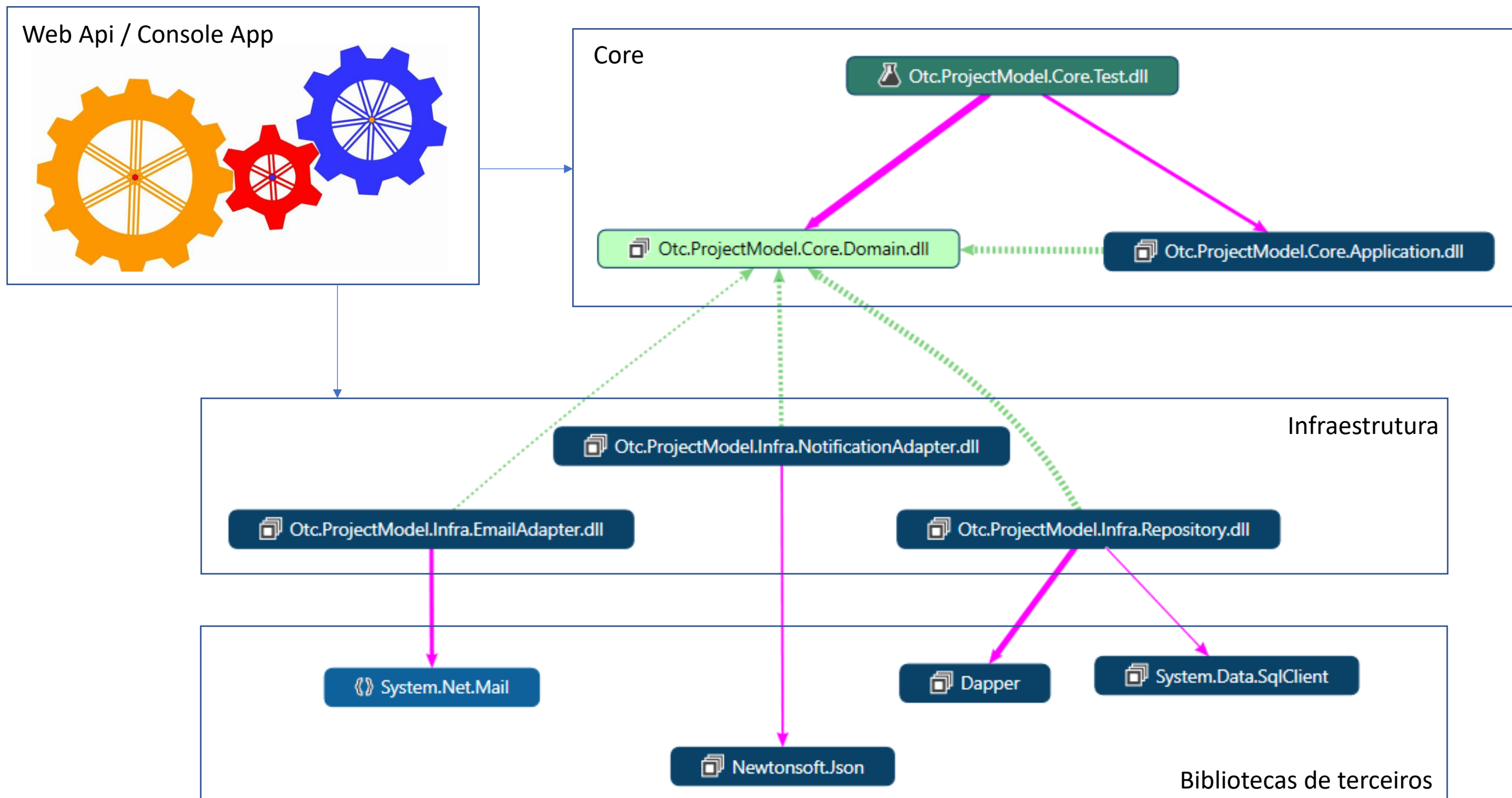
# Infraestrutura



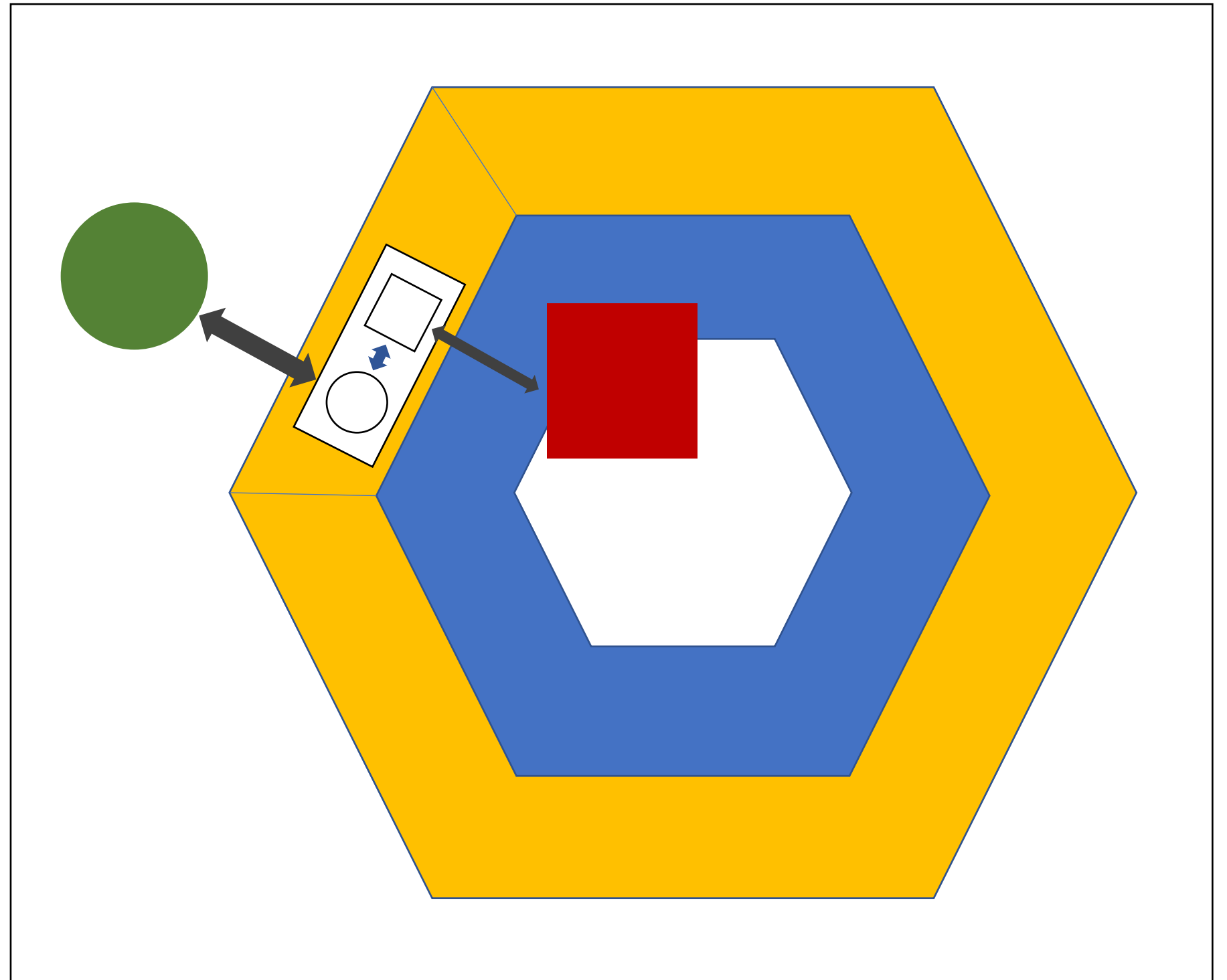
# Visão geral



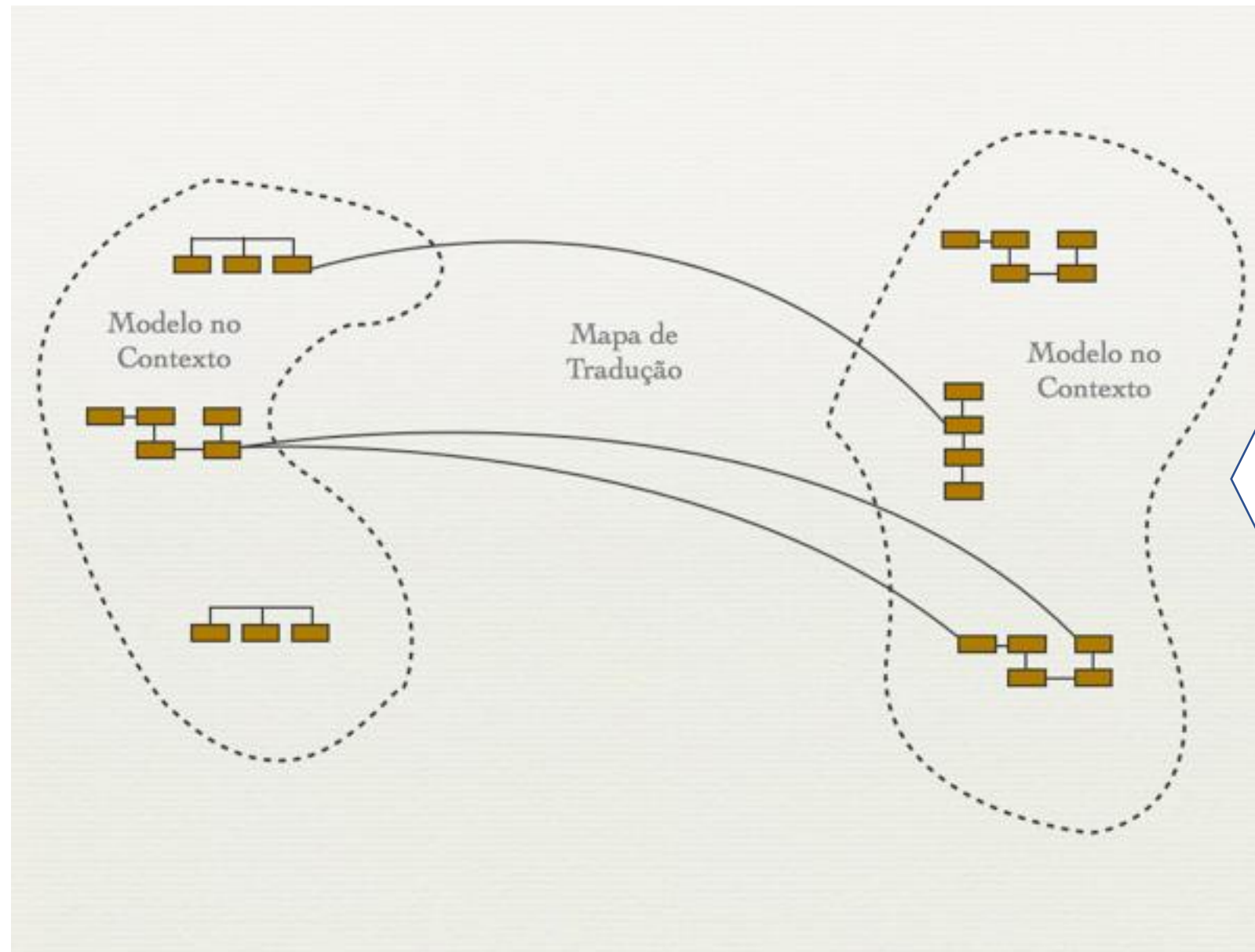




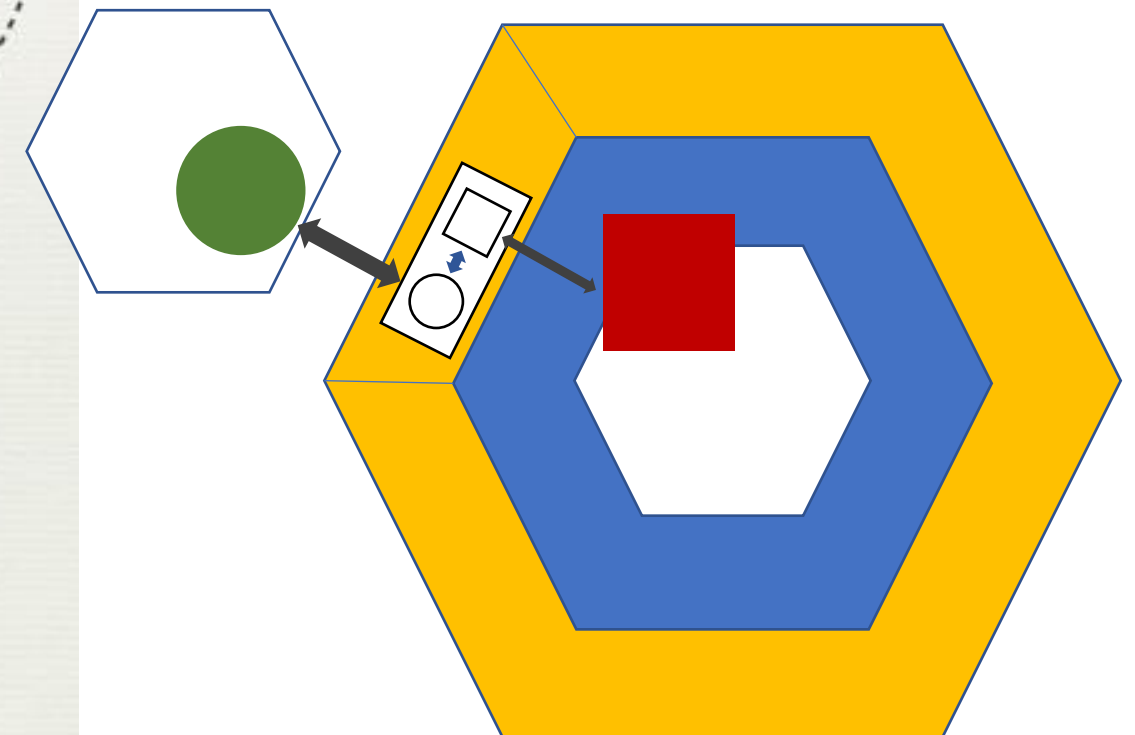
# Adaptadores



# Bounded Context / Context Mapping

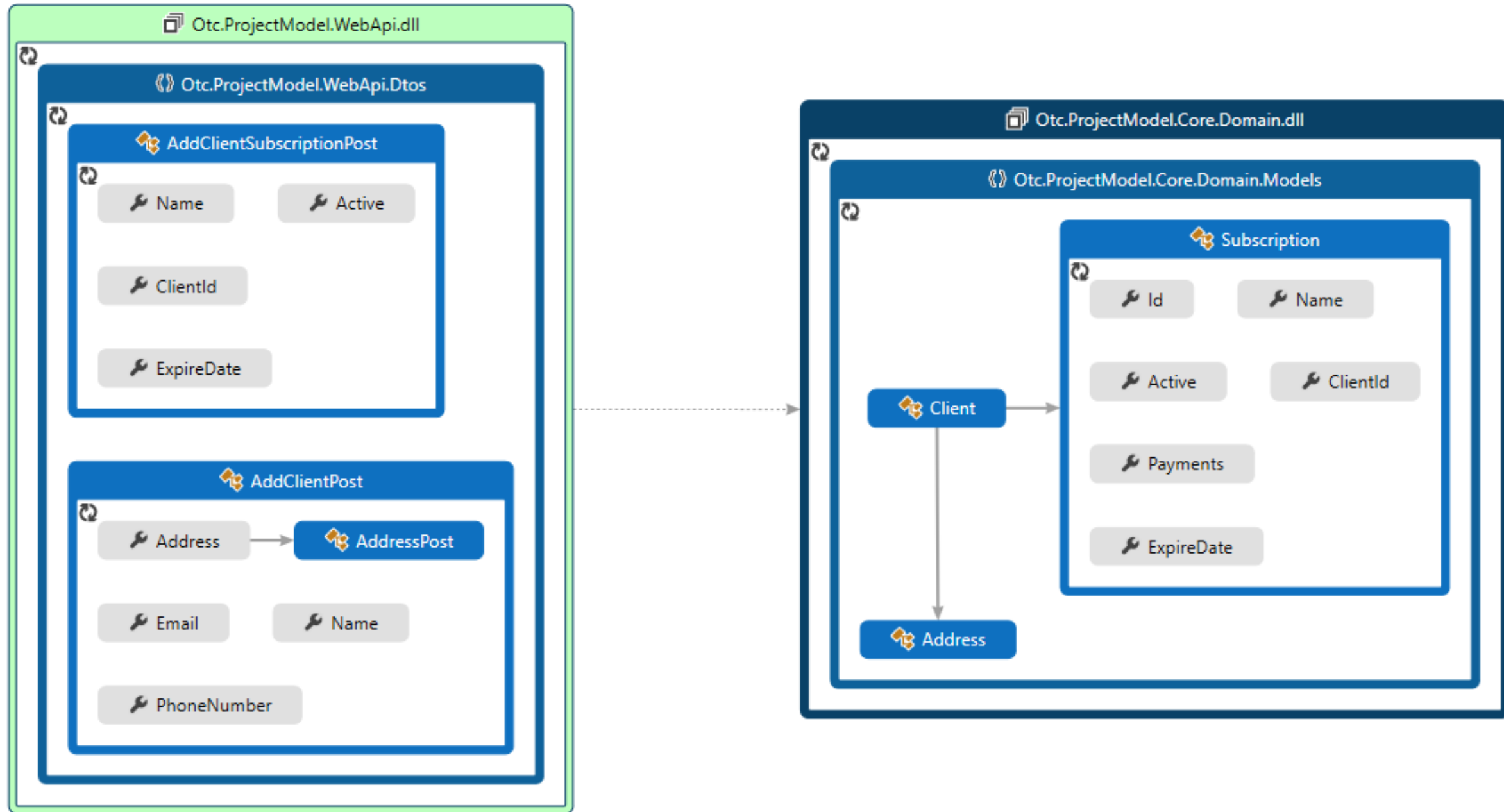


Proteger o conhecimento do domínio contra corrupção por outros domínios, subdomínios técnicos, etc.



<http://www.agileandart.com/2010/07/16/ddd-introducao-a-domain-driven-design/>

# Adaptadores / Context Mapping





<https://goo.gl/forms/CUCHBQPGTE1S2wgC3>

# Princípios e boas práticas



# Princípios e boas práticas

```
string Weekday1(int day)
{
    switch (day)
    {
        case 1:
            return "Monday";
        case 2:
            return "Tuesday";
        case 3:
            return "Wednesday";
        case 4:
            return "Thursday";
        case 5:
            return "Friday";
        case 6:
            return "Saturday";
        case 7:
            return "Sunday";
        default:
            throw new InvalidOperationException("day must be in range 1 to 7");
    }
}
```

```
string Weekday2(int day)
{
    if ((day < 1) || (day > 7)) throw new InvalidOperationException("day must be in range 1 to 7");
    string[] days = { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday" };
    return days[day - 1];
}
```



# Princípios e boas práticas

```
string Weekday1(int day)
{
    switch (day)
    {
        case 1:
            return "Monday";
        case 2:
            return "Tuesday";
        case 3:
            return "Wednesday";
        case 4:
            return "Thursday";
        case 5:
            return "Friday";
        case 6:
            return "Saturday";
        case 7:
            return "Sunday";
        default:
            throw new InvalidOperationException("day must be in range 1 to 7");
    }
}
```



```
string Weekday2(int day)
{
    if ((day < 1) || (day > 7)) throw new InvalidOperationException("day must be in range 1 to 7");
    string[] days = { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday" };
    return days[day - 1];
}
```





# Princípios e boas práticas

**DRY**  
**DON'T**  
**REPEAT**  
**YOURSELF**

```
/// <summary>  
/// Inclui um novo Cliente  
/// </summary>  
/// <param name="addClientRequest">Cliente</param>  
/// <returns>Client</returns>  
[HttpPost]  
[ProducesResponseType(typeof(ClientCoreException), 400)]  
[ProducesResponseType(typeof(AddClientPost), 200)]  
0 references | Luciano Lima, 9 days ago | 1 author, 1 change | 3 work items | 0 requests | 0 exceptions  
public async Task<IActionResult> AddClientAsync([FromBody] AddClientPost addClientRequest)  
{  
    try  
    {  
        var client = Mapper.Map<Client>(addClientRequest);  
  
        await clientService.AddClientAsync(client);  
  
        return Ok(client);  
    }  
    catch (CoreException e)  
    {  
        return BadRequest(e.Message);  
    }  
}
```

# Princípios e boas práticas

**DRY**  
**DON'T**  
**REPEAT**  
**YOURSELF**

```
/// <summary>
/// Inclui um novo Cliente
/// </summary>
/// <param name="addClientRequest">Cliente</param>
/// <returns>Client</returns>
[HttpPost]
[ProducesResponseType(typeof(ClientCoreException), 400)]
[ProducesResponseType(typeof(AddClientPost), 200)]
0 references | Luciano Lima, 9 days ago | 1 author, 1 change | 3 work items | 0 requests | 0 exceptions
public async Task<IActionResult> AddClientAsync([FromBody] AddClientPost addClientRequest)
{
    var client = Mapper.Map<Client>(addClientRequest);

    await clientService.AddClientAsync(client);

    return Ok(client);
}
```

# Princípios e boas práticas

- S** Single Responsibility Principle (SRP)
- O** Open/Closed Principle (OCP)
- L** Liskov Substitution Principle (LSP)
- I** Interface Segregation Principle (ISP)
- D** Dependency Inversion Principle (DIP)

Robert C. Martin (Uncle Bob) – Design Principles and Design Patterns – 2000

# Single Responsibility Principle (SRP)

- Uma classe deve ter uma única responsabilidade;
- Uma classe deve ter apenas uma razão para mudança;
- A mudança em uma classe deve afetar apenas um ator (Clean Architecture)

# Single Responsibility Principle (SRP)

```
1 public class DebitoContaCorrente
2 {
3     public void ValidarSaldo(int valor) { }
4     public void DebitarConta(int valor) { }
5     public void EmitirComprovante() { }
6 }
7
8 }
```

Eduardo Pires - <http://www.eduardopires.net.br/2013/05/single-responsibility-principle-srp/>

# Single Responsibility Principle (SRP)

```
1 public class DebitoContaCorrente
2 {
3     public void DebitarConta(int valor) { }
4 }
5
6 public class SaldoContaCorrente
7 {
8     public void ValidarSaldo(int valor) { }
9 }
10
11 public class ComprovanteContaCorrente
12 {
13     public void EmitirComprovante() { }
14 }
```

Eduardo Pires - <http://www.eduardopires.net.br/2013/05/single-responsibility-principle-srp/>

# Open/Closed Principle (OCP)

- Fechado para modificações porém aberto para extensibilidade;
- Você deve ser capaz de estender o comportamento das classes sem precisar modificá-las.

# Open/Closed Principle (OCP)

```
1 public enum TipoDebito { ContaCorrente, Poupanca }
2
3 public class Debito
4 {
5     public void Debitar(int valor, TipoDebito tipo)
6     {
7         if (tipo == TipoDebito.Poupanca)
8         {
9             // Debita Poupanca
10        }
11        if (tipo == TipoDebito.ContaCorrente)
12        {
13            // Debita ContaCorrente
14        }
15    }
16 }
```

Eduardo Pires - <http://www.eduardopires.net.br/2013/05/open-closed-principle-ocp/>



# Open/Closed Principle (OCP)

```
1 public abstract class Debito
2 {
3     public abstract void Debitar(int valor);
4 }
5
6 public class DebitoContaCorrente : Debito
7 {
8     public override void Debitar(int valor)
9     {
10         // Debita Conta Corrente
11     }
12 }
13
14 public class DebitoContaPoupanca : Debito
15 {
16     public override void Debitar(int valor)
17     {
18         // Debita Conta Poupança
19     }
20 }
21
22 public class DebitoContaInvestimento : Debito
23 {
24     public override void Debitar(int valor)
25     {
26         // Debita Conta Investimento
27     }
28 }
```

Eduardo Pires - <http://www.eduardopires.net.br/2013/05/open-closed-principle-ocp/>

# Liskov Substitution Principle (LSP)

Se  $q(x)$  é uma propriedade demonstrável dos objetos  $x$  de tipo  $T$ .

Então  $q(y)$  deve ser verdadeiro para objetos  $y$  de tipo  $S$  onde  $S$  é um subtipo de  $T$ .

Barbara Liskov, 1993

# Liskov Substitution Principle (LSP)

Se  $q(x)$  é uma propriedade demonstrável dos objetos  $x$  de tipo  $T$ .  
Então  $q(y)$  deve ser verdadeiro para objetos  $y$  de tipo  $S$  onde  $S$  é um subtipo de  $T$ .

Barbara Liskov, 1993

Classes derivadas devem ser substituíveis por suas classes bases e garantir o comportamento esperado na classe base.

Uncle Bob, 2000

# Liskov Substitution Principle (LSP)

2 references | 0 changes | 0 authors, 0 changes

```
class Retangulo
{
    6 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public virtual double Altura { get; set; }
    6 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public virtual double Comprimento { get; set; }
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public double Area { get { return Altura * Comprimento; } }
}
```

1 reference | 0 changes | 0 authors, 0 changes

```
class Quadrado : Retangulo
{
    6 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public override double Altura { set { base.Altura = base.Comprimento = value; } }
    6 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public override double Comprimento { set { base.Altura = base.Comprimento = value; } }
}
```

# Liskov Substitution Principle (LSP)

0 references | 0 changes | 0 authors, 0 changes

```
class Operacoes
```

```
{
```

0 references | 0 changes | 0 authors, 0 changes | 0 exceptions

```
public void Crescer(Retangulo retangulo)
```

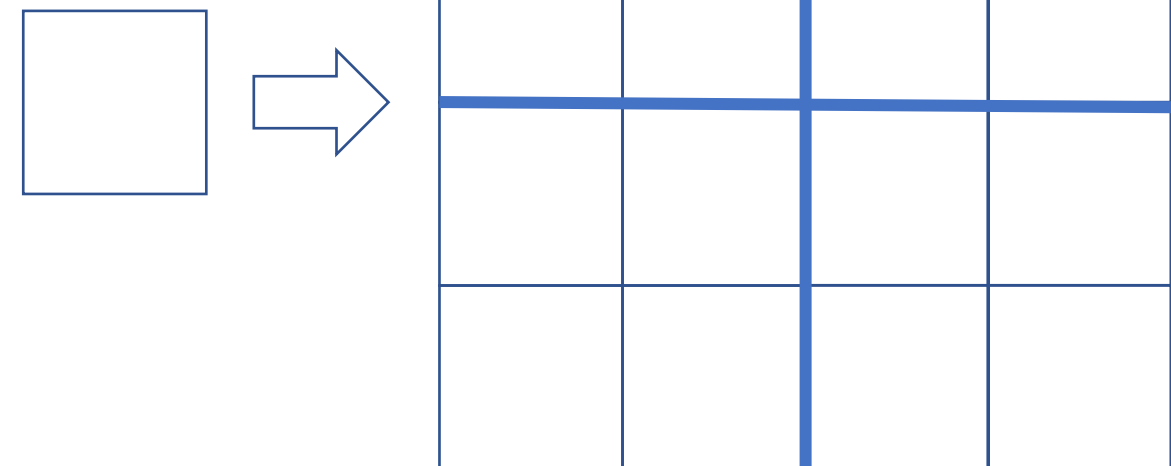
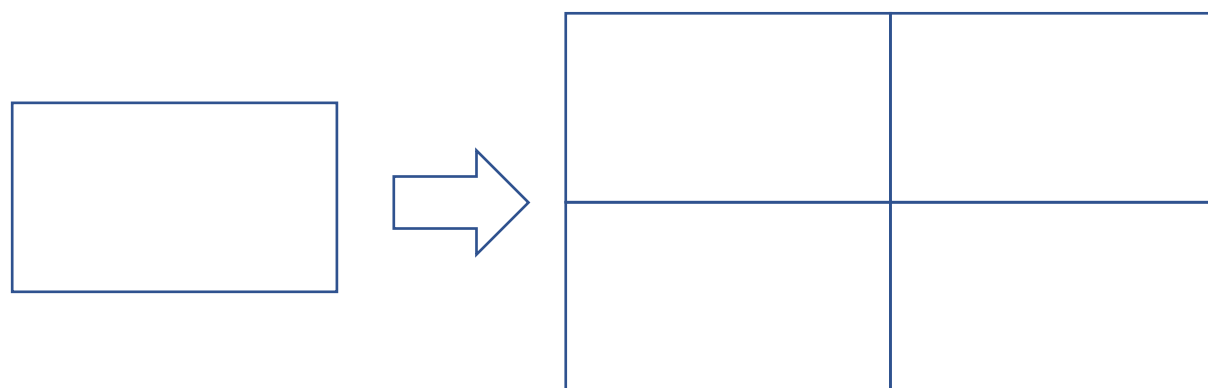
```
{
```

```
    retangulo.Altura *= 2;
```

```
    retangulo.Comprimento *= 2;
```

```
}
```

```
}
```



# Interface Segregation Principle (ISP)

Construa interfaces com granularidade fina que sejam específicas para o cliente.

Uncle Bob, 2000

# Interface Segregation Principle (ISP)

```
0 references | 0 changes | 0 authors, 0 changes
public interface IBaseRepository<TEntity> where TEntity : class
{
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    void Gravar(TEntity obj);
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    void Atualizar(TEntity obj);
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    IEnumerable<TEntity> ObterTodos();
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    TEntity ObterPorId(int id);
}
```

# Interface Segregation Principle (ISP)

```
0 references | 0 changes | 0 authors, 0 changes
public class ClienteRepository : IBaseRepository<Cliente>
{
    1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions
    public void Gravar(Cliente obj)
    {
        //ação de gravar
    }

    1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions
    public void Atualizar(Cliente obj)
    {
        //ação de atualizar
    }

    1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions
    public IEnumerable<Cliente> ObterTodos()
    {
        //ação de listar todos
        return Enumerable.Empty<Cliente>();
    }

    1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions
    public Cliente ObterPorId(int id)
    {
        //ação de listar por id
        return default(Cliente);
    }
}
```



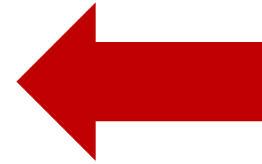
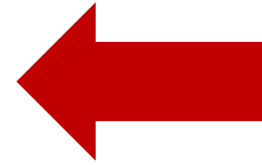
# Interface Segregation Principle (ISP)

```
0 references | 0 changes | 0 authors, 0 changes
public class CidadeRepository : IBaseRepository<Cidade>
{
    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public void Gravar(Cidade obj)
    {
        throw new NotImplementedException();
    }

    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public void Atualizar(Cidade obj)
    {
        throw new NotImplementedException();
    }

    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public IEnumerable<Cidade> ObterTodos()
    {
        //ação para obter todas as cidades
        return Enumerable.Empty<Cidade>();
    }

    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public Cidade ObterPorId(int id)
    {
        //ação para obter uma cidade por id
        return default(Cidade);
    }
}
```



# Interface Segregation Principle (ISP)

```
0 references | 0 changes | 0 authors, 0 changes
public interface IBaseEscritaRepository where TEntity : class
{
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    void Gravar(TEntity obj);
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    void Atualizar(TEntity obj);
}
```

```
0 references | 0 changes | 0 authors, 0 changes
public interface IBaseLeituraRepository where TEntity : class
{
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    IEnumerable ObterTodos();
    0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    TEntity ObterPorId(int id);
}
```

# Interface Segregation Principle (ISP)

0 references | 0 changes | 0 authors, 0 changes

```
public class ClienteRepository : IBaseEscritaRepository<Cliente>, IBaseLeituraRepository<Cliente>
```

```
{
```

1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions

```
public void Gravar(Cliente obj)
```

```
{
```

```
    //ação de gravar
```

```
}
```

1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions

```
public void Atualizar(Cliente obj)
```

```
{
```

```
    //ação de atualizar
```

```
}
```

1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions

```
public IEnumerable<Cliente> ObterTodos()
```

```
{
```

```
    //ação de listar todos
```

```
    return Enumerable.Empty<Cliente>();
```

```
}
```

1 reference | 0 changes | 0 authors, 0 changes | 0 exceptions

```
public Cliente ObterPorId(int id)
```

```
{
```

```
    //ação de listar por id
```

```
    return default(Cliente);
```

```
}
```

```
}
```

# Interface Segregation Principle (ISP)

```
0 references | 0 changes | 0 authors, 0 changes
public class CidadeRepository : IBaseLeituraRepository<Cidade>
{
    2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
    public IEnumerable<Cidade> ObterTodos()
    {
        //ação para obter todas as cidades
        return Enumerable.Empty<Cidade>();
    }
}

2 references | 0 changes | 0 authors, 0 changes | 0 exceptions
public Cidade ObterPorId(int id)
{
    //ação para obter uma cidade por id
    return default(Cidade);
}
}
```

# Dependency Inversion Principle (DIP)

Dependa de abstrações (interfaces) ao invés  
de classes concretas.

Uncle Bob, 2000

# Interfaces

2 references

`class RepositorioPessoas`

{

0 references

`public Pessoa[] ObterPessoas()`

{

`var query = @"`

`select top 100 nome`

`, salario`

`, data_de_nascimento`

`, tipo`

`from pessoas";`

`return Db.Query<Pessoa>(query);`

}

1 reference

`public void Atualizar(Pessoa pessoa)`

{

`var query = "update pessoas set ...";`

`Db.Query(query);`

}

}

Revisão de exemplo  
apresentado anteriormente.

# Dependency Inversion Principle (DIP)

1 reference

```
class Exemplo02
```

```
{  
    private RepositorioPessoas repositorioDePessoas;  
  
    0 references  
    public Exemplo02(RepositorioPessoas repositorioDePessoas)  
    {  
        this.repositorioDePessoas = repositorioDePessoas;  
    }  
  
    0 references  
    public void CorrigirDataDeNascimento(Pessoa pessoa, DateTime novaDataDeNascimento)  
    {  
        pessoa.DataDeNascimento = novaDataDeNascimento;  
  
        repositorioDePessoas.Atualizar(pessoa);  
    }  
}
```

**RepositorioPessoas**

Class

Methods

Atualizar

ObterPessoas

# Dependency Inversion Principle (DIP)

1 reference

```
interface IRepositoryPessoas
{
    2 references
    void Atualizar(Pessoa pessoa);
    1 reference
    Pessoa[] ObterPessoas();
}
```

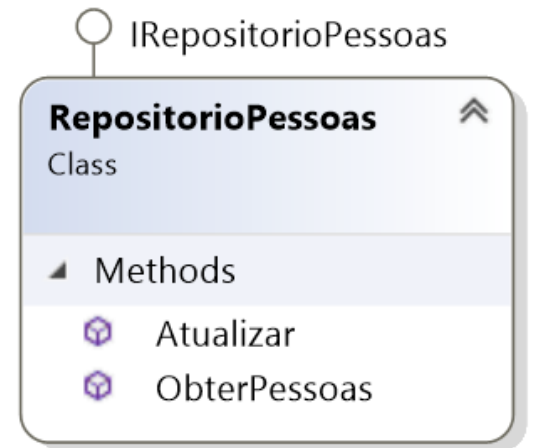
2 references

```
class RepositorioPessoas : IRepositoryPessoas
{
    1 reference
    public Pessoa[] ObterPessoas()
    {
        var query = @"
            select top 100 nome
                , salario
                , data_de_nascimento
                , tipo
            from pessoas";

        return Db.Query<Pessoa>(query);
    }

    2 references
    public void Atualizar(Pessoa pessoa)
    {
        var query = "update pessoas set ...";

        Db.Query(query);
    }
}
```





# Dependency Inversion Principle (DIP)

```
1 reference
class Exemplo02
{
    private IRepositoryPessoas repositorioDePessoas;

    0 references
    public Exemplo02(IRepositoryPessoas repositorioDePessoas)
    {
        this.repositorioDePessoas = repositorioDePessoas;
    }

    0 references
    public void CorrigirDataDeNascimento(Pessoa pessoa, DateTime novaDataDeNascimento)
    {
        pessoa.DataDeNascimento = novaDataDeNascimento;

        repositorioDePessoas.Atualizar(pessoa);
    }
}
```

## IRepositoryPessoas

Interface

### Methods



Atualizar



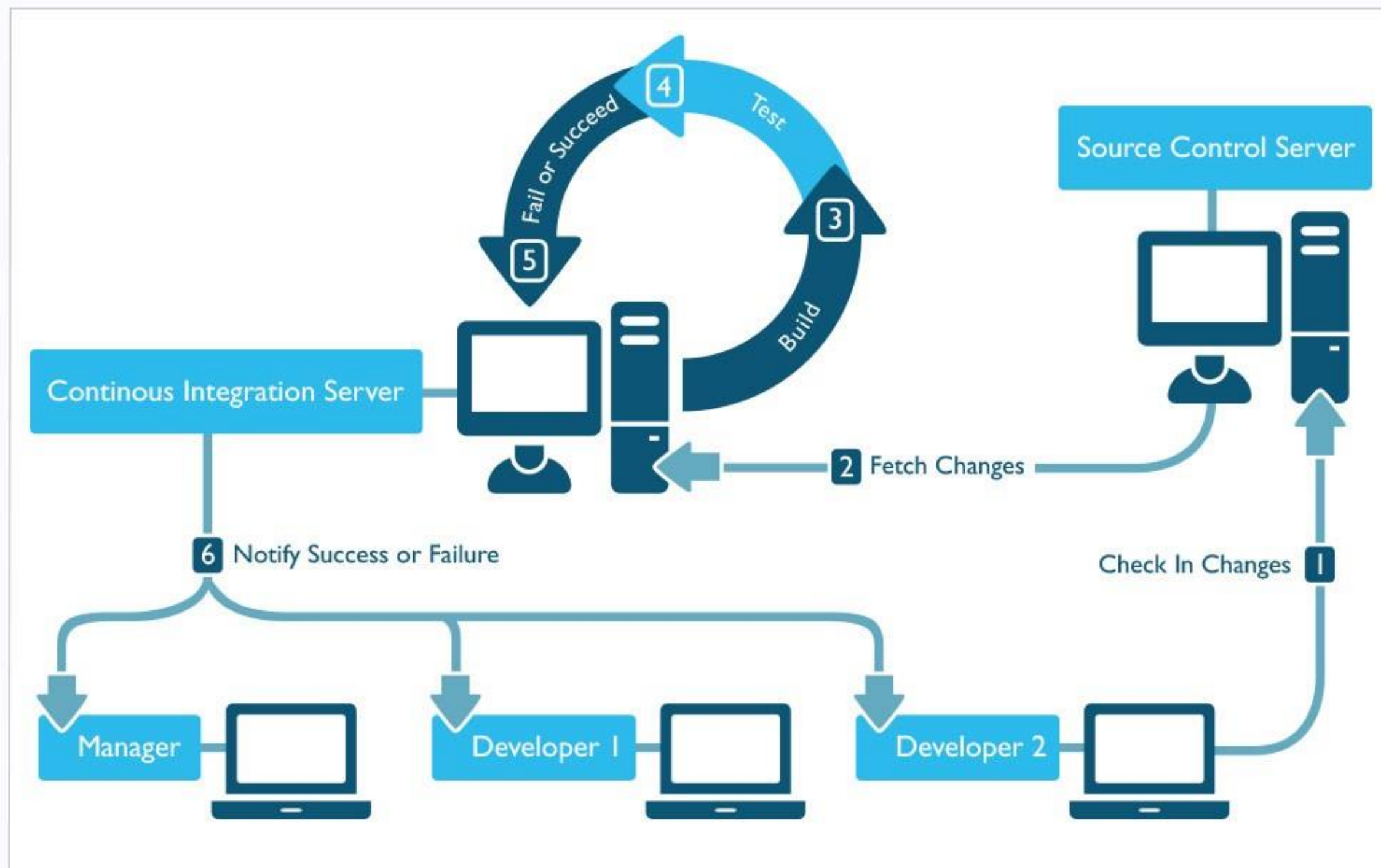
ObterPessoas



<https://goo.gl/forms/xqTzlynPITmdr9CF3>

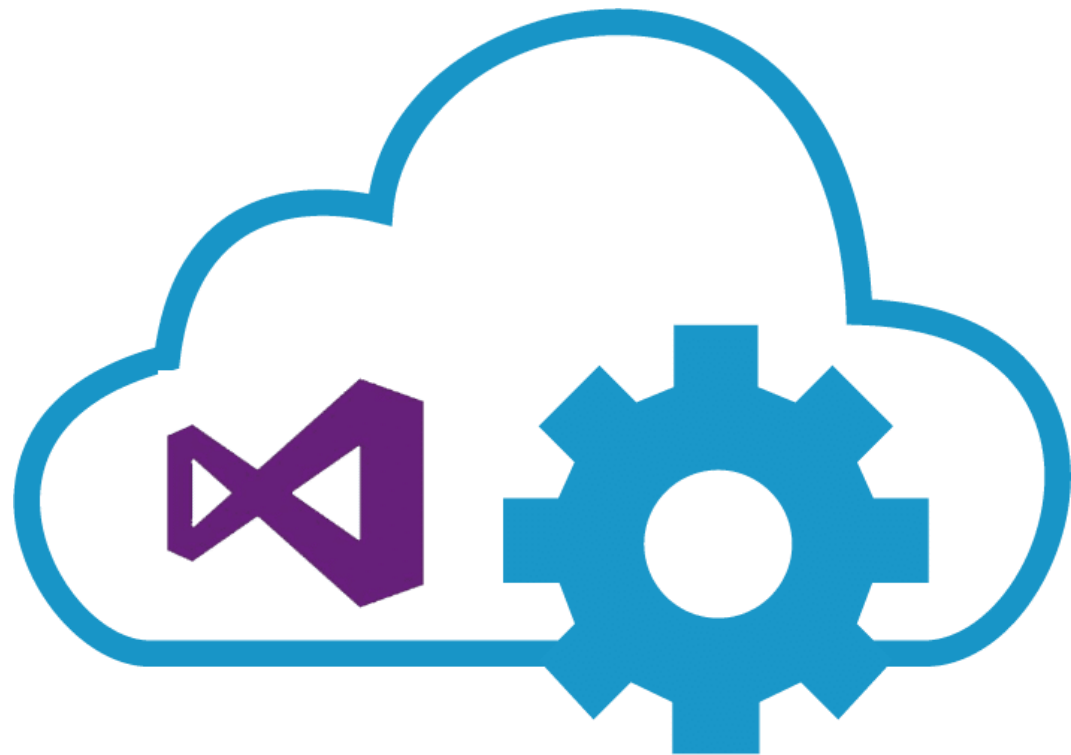
# Integração Contínua





Fonte: C. Aaron Cois, **DevOps** Blog - <https://insights.sei.cmu.edu/devops/2015/01/continuous-integration-in-devops-1.html>

# Integração Contínua



Visual Studio Team Services

+



Git

# Integração Contínua

## Terminologia (TFS vs. Git)

| TFS                                     | Git                                      |
|---|--|
| Create workspace and get latest version | Clone                                    |
| Switch workspace branch                 | Checkout                                 |
| Checkin                                 | Commit / Push                            |
| Pending changes                         | Status ou Changes no VS Team Explorer UI |
| Shelve                                  | Stash                                    |
| Get latest version                      | Pull                                     |
| Label                                   | Tag / Release                            |
| Include changes                         | Stage                                    |
| Exclude changes                         | Unstage                                  |
| Branch                                  | Branch                                   |
| Merge                                   | Merge / Rebase                           |

# Integração Contínua

The screenshot shows the Azure DevOps interface for the 'cilab-vsts-git' team project. The top navigation bar includes 'Dashboards', 'Code', 'Work', 'Build and Release', and a search bar. The 'Builds' tab is highlighted in the top bar and the left sidebar. The 'Build Definitions' page is displayed, showing a message: 'The cilab-vsts-git team project doesn't have any build definitions.' Below this message is a '+ New definition' button. A red arrow points from this button to the 'ASP.NET (PREVIEW)' template in the 'Select a template' dialog. The dialog also shows other templates like '.NET Desktop', 'ASP.NET Core', 'ASP.NET Core (.NET Framework)', 'Azure Web App', and 'Universal Windows Platform'.

Build Definitions

Mine All Definitions Queued

You can use a build definition to automate your builds.

The cilab-vsts-git team project doesn't have any build definitions.

+ New definition ? Get started

Select a template

Or start with an [Empty process](#)

Featured

- .NET Desktop  
Build and run tests for .NET Desktop or Windows Classic Desktop solutions. This template requires that Visual Studio be installed on the build agent.
- ASP.NET (PREVIEW)  
Build ASP.NET web applications [Apply](#)
- ASP.NET Core  
Build ASP.NET Core web applications
- ASP.NET Core (.NET Framework)  
Build ASP.NET Core web applications targeting the full .NET Framework
- Azure Web App  
Build, package, test and deploy your Azure Web App.
- Universal Windows Platform  
Build Universal Windows Platform applications using Visual Studio

# Integração Contínua

The screenshot displays the Azure DevOps web interface for a project named 'cilab-vsts-git'. The top navigation bar includes 'Dashboards', 'Build and Release', and a search bar. Below this, the 'Builds' tab is selected, showing a list of build definitions. A specific build, 'cilab-pull-requests-ci : #20170924.1', is highlighted, indicating it was requested by 'Matheus Neder' just now. The build's status is 'in progress', which is highlighted with a red box. A red arrow points from this status box to the build's log output on the right. The log shows the build process, including file uploads, artifact publishing, and post-job cleanup. The build is triggered by an 'Initial commit' with the commit hash 'cfc903'.

Build Definitions

Build ID or build number  [+ New](#) [+ Import](#)

Mine All Definitions Queued

Recently built

|   | Status             | Triggered by                     |
|---|--------------------|----------------------------------|
| <b>cilab-pull-requests-ci : #20170924.1</b><br>Matheus Neder requested just now | <b>in progress</b> | Initial commit.<br>cfc903 in ... |

Build Log:

```
Uploading 5 files
File upload succeed.
Upload 'd:\a\1\1\1' to file container: '#/1175662/cilab-pull-requests-ci.20170924.1'
Associated artifact 12 with build 25
*****
Async Command End: Upload Artifact
*****
Finishing: Publish Artifact
*****
Starting: Post Job Cleanup
*****
Cleaning any cached credential from repository: cilab-vsts-git (Git)
git remote set-url origin https://matheusneder.visualstudio.com/_git/cilab-vsts-git
git remote set-url --push origin https://matheusneder.visualstudio.com/_git/cilab-vsts-git
*****
Finishing: Post Job Cleanup
*****
Finishing: Build
*****
```



cilab-vsts-git

Dashboards

Code

Work

Build and Release

Test

Wiki\*

Search v

Overview

Work

Security

Version Control

Policies

Agent Queues

Notifications

Service Hooks

Services

Test

Release

Policies for: cilab-vsts-git > cilab-vsts-git > master

Save changes

Discard changes

☒ Protect this branch

- Code changes must be submitted via pull request
- This branch cannot be deleted
- Manage permissions for this branch on the [Security page](#)

☐ Require a minimum number of reviewers

Require approval from a specified number of reviewers on pull requests.

☐ Check for linked work items

Encourage traceability by checking for linked work items on pull requests.

☐ Check for comment resolution

Check to see that all comments have been resolved on pull requests.

☐ Enforce a merge strategy

Require a specific type of merge when pull requests are completed.

Build validation

Validate code by pre-merging and building pull request changes

+ Add build policy

Require approval from external services

Require third party services to post successful status to complete pull requests.

+ Add service

Automatically include code reviewers

Include specific users or groups in the code review based on which files changed.

+ Add automatic reviewers

Add build policy

Build definition \*

cilab-pull-requests-ci

☒ Automatic (whenever the source branch is updated)

☐ Manual

Policy requirement

☒ Required

Build must succeed in order to complete pull requests.

☐ Optional

Build failure will not block completion of pull requests.

Build expiration

☐ Immediately when master is updated

☒ After 

12

 hours if master has been updated

☐ Never

Display name

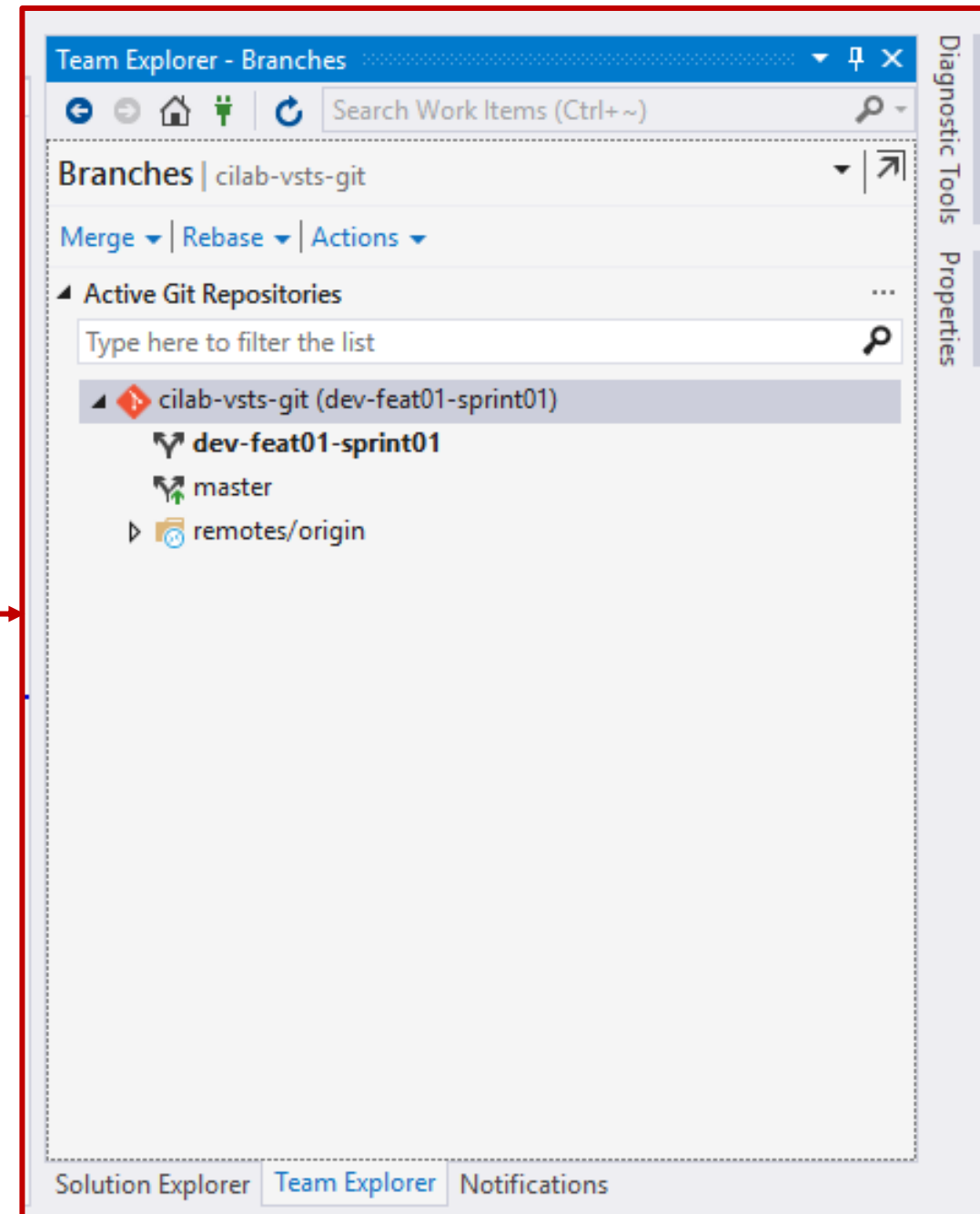
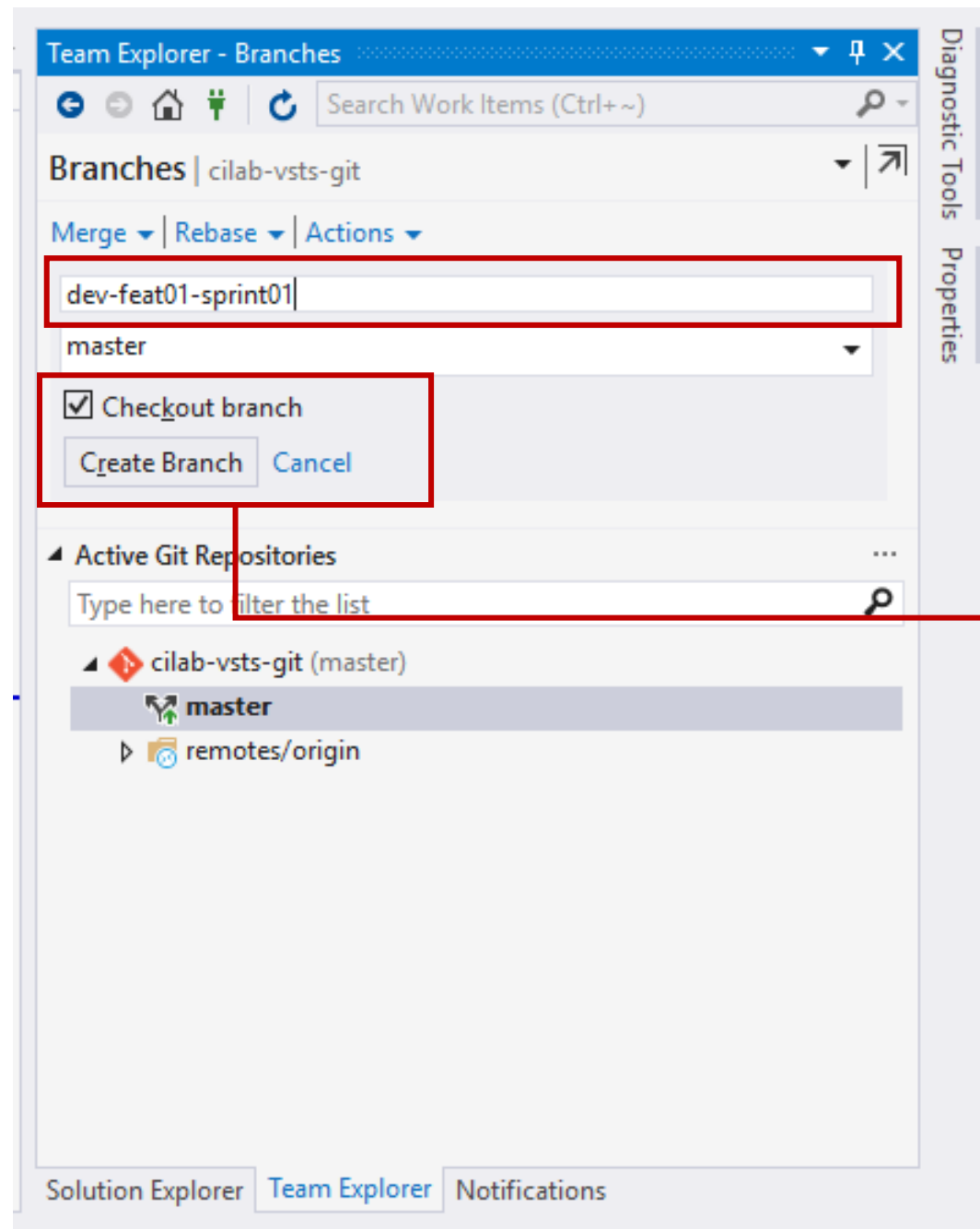
Save

Cancel

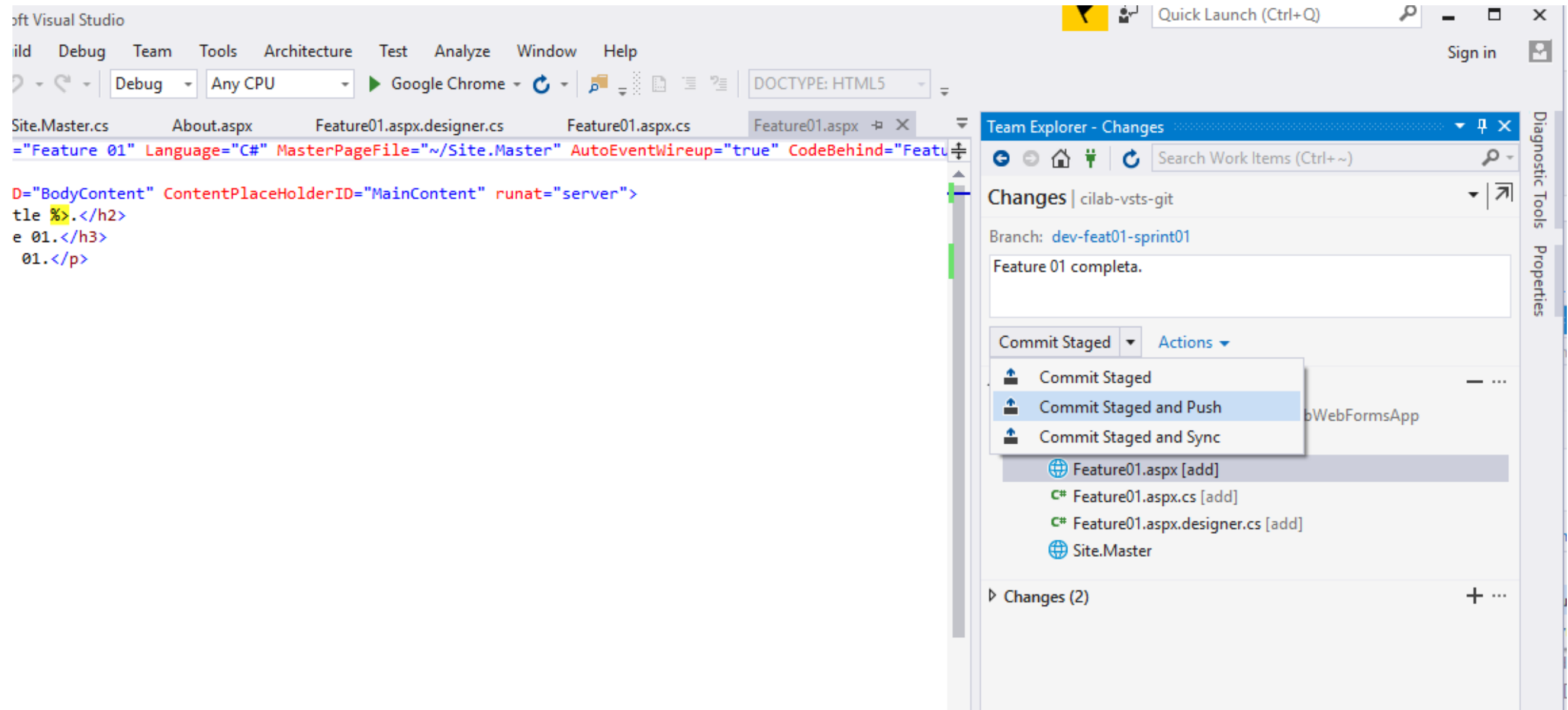
04/08/2018

Olé Tecnologia

# Integração Contínua



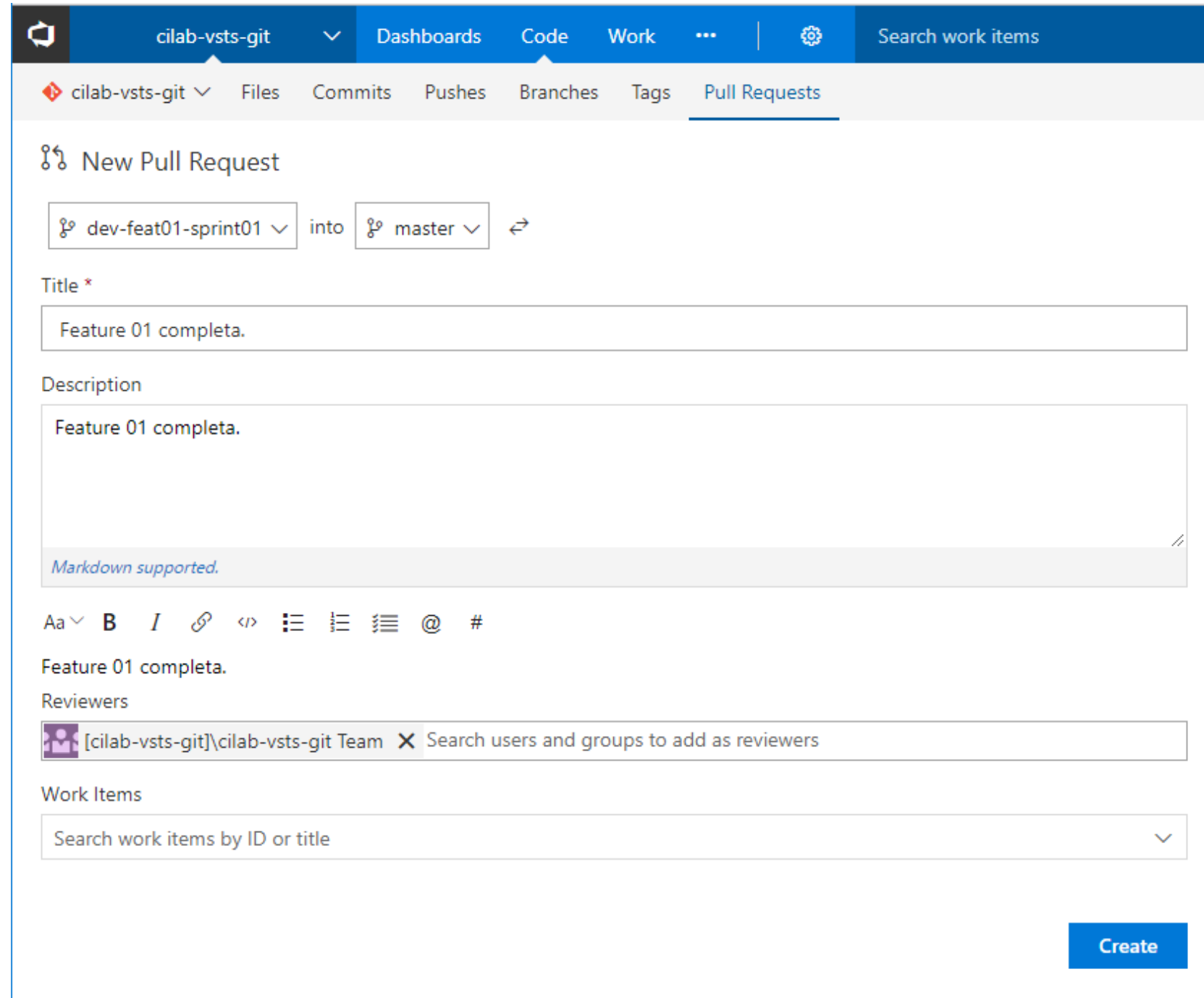
# Integração Contínua



# Integração Contínua

The image shows a screenshot of the GitHub web interface for a repository named 'cilab-vsts-git'. The top navigation bar includes 'Dashboards', 'Code', 'Work', and a settings gear. The left sidebar shows the repository name and a list of options: 'Files', 'Commits', 'Pushes', 'Branches', 'Tags', 'Pull Requests' (highlighted with a red box), 'New pull request', and 'Manage repositories'. The main content area is titled 'Pull Requests' and features a 'New pull request' button (highlighted with a red box). Below this, there is a message: 'You updated dev-feat01-sprint01 3 minutes ago — Create a pull request'. At the bottom, there is a section with the text 'Let the rest of your team know your code is ready to review' and 'Currently, no pull requests need your attention.' Below this text is another 'New pull request' button (highlighted with a red box) and a link 'Learn more about Pull Requests'. A red arrow points from the 'Pull Requests' option in the sidebar to the 'New pull request' button in the main content area.


# Integração Contínua





The screenshot shows the 'New Pull Request' page in Azure DevOps. The top navigation bar includes 'cilab-vsts-git', 'Dashboards', 'Code', 'Work', and a search bar. Below this, a secondary bar shows repository-specific options: 'Files', 'Commits', 'Pushes', 'Branches', 'Tags', and 'Pull Requests' (which is selected). The main form area is titled 'New Pull Request' and contains several sections: a branch selection dropdown set to 'dev-feat01-sprint01' into 'master'; a 'Title' field with the text 'Feature 01 completa.'; a 'Description' field with the same text and a 'Markdown supported.' note; a rich text editor toolbar; a 'Reviewers' section with a dropdown menu showing '[cilab-vsts-git]\cilab-vsts-git Team' and a search prompt; and a 'Work Items' section with a search bar. A blue 'Create' button is located at the bottom right of the form.

**cilab-vsts-git** ▾ Dashboards Code Work ... | Search work items

**cilab-vsts-git** ▾ Files Commits Pushes Branches Tags Pull Requests

 New Pull Request

 dev-feat01-sprint01 ▾ into  master ▾ ↔






Title \*

Feature 01 completa.

Description


Feature 01 completa.

*Markdown supported.*

Aa ▾ **B** *I*      @ #

Feature 01 completa.

Reviewers

 [cilab-vsts-git]\cilab-vsts-git Team ✕ Search users and groups to add as reviewers

Work Items

Search work items by ID or title ▾

**Create**

cilab-vsts-git ▾ Dashboards Code Work ... | Search work items

cilab-vsts-git ▾ Files Commits Pushes Branches Tags Pull Requests Fork Clone

🔗 10 **ACTIVE** Feature 01 completa.

MN Matheus Neder 🔗 dev-feat01-sprint01 into 🔗 master MN Approve ▾ Set auto-complete ▾ ...

Overview Files Updates Commits

Policies

Required  
▶ Build in progress

Work Items ✕ +  
No related work items

Reviewers ✉ +  
cilab-vsts-git Team

Description  
Feature 01 completa.

Show everything ▾ 🔍

MN Add a comment...

Created by MN Matheus Neder just now

- 🔗 Test Assemblies
- 🔗 Publish symbols path
- 🔗 Publish Artifact
- 🔗 Post Job Cleanup

Task Groups Deployment Groups\*

cilab-pull-requests-ci / Build 20170925.1 / Build

Edit build definition Cancel Queue new build... Download all logs as zip

**Build Started**

Build 🔗  
Running for 19 seconds (Hosted Agent)

Console Timeline Code coverage\* Tests

```
Feeds used:
  https://api.nuget.org/v3/index.json
Installed:
  25 package(s) to packages.config projects
*****
Finishing: NuGet restore
*****
Starting: Build solution
*****
Task       : Visual Studio Build
Description: Build with MSBuild and set the Visual Studio version property
Version    : 1.120.0
Author     : Microsoft Corporation
Help       : [More Information](https://go.microsoft.com/fwlink/?LinkID=613727)
*****
"D:\a\_tasks\VSBuild_71a9a2d3-a98a-4caa-96ab-affca411ecda\1.120.0\ps_modules\MSBuild\
version [15.0,16.0) -latest -format json
"D:\a\_tasks\VSBuild_71a9a2d3-a98a-4caa-96ab-affca411ecda\1.120.0\ps_modules\MSBuild\
version [15.0,16.0) -products Microsoft.VisualStudio.Product.BuildTools -latest -form
"C:\Program Files (x86)\MSBuild\14.0\bin\msbuild.exe" "d:\a\1\s\source\CiLabWebForms\
/d1:CentralLogger,"D:\a\_tasks\VSBuild_71a9a2d3-a98a-4caa-96ab-
affca411ecda\1.120.0\ps_modules\MSBuildHelpers\Microsoft.TeamFoundation.DistributedT
```

# Integração Contínua

The screenshot displays the 'Pull Requests' page for a repository named 'cilab-vsts-git'. The top navigation bar includes 'Dashboards', 'Code', 'Work', and a search bar. Below the navigation bar, the pull request details are shown: '10 ACTIVE Feature 01 completa.' by 'Matheus Neder' (MN) merging 'dev-feat01-sprint01' into 'master'. A red box highlights the 'Complete' button in the top right corner. On the left sidebar, the 'Policies' section is highlighted with a red box, showing a 'Required' policy with a green checkmark and the text 'Build succeeded'. Below this, the 'Work Items' section shows 'No related work items', and the 'Reviewers' section lists 'cilab-vsts-git Team'. The main content area shows the 'Description' as 'Feature 01 completa.', a comment input field, and a 'Created by' field listing 'Matheus Neder'.

**Navigation Bar:** cilab-vsts-git, Dashboards, Code, Work, Search work items

**Repository:** cilab-vsts-git

**Files:** Files, Commits, Pushes, Branches, Tags, Pull Requests

**Pull Request Details:** 10 ACTIVE Feature 01 completa. Matheus Neder (MN) dev-feat01-sprint01 into master

**Buttons:** Approve, Complete (highlighted), ...

**Policies:** Required, Build succeeded (highlighted)

**Work Items:** No related work items

**Reviewers:** cilab-vsts-git Team

**Description:** Feature 01 completa.

**Comments:** Add a comment...

**Created by:** Matheus Neder (MN) just now

**Description**  
Feature 01 completa.

Show everything ▾

MN Add a comment...

MN **Matheus Neder** 2 minutes ago Active ▾  
Necessário adicionar testes unitários.

MN Write a reply... Resolve

2 MN **Matheus Neder** pushed 1 commit creating update 2 just now

7ac744ae Conteúdo corrigido! matheusneder@g... just now

MN **Matheus Neder** 20 minutes ago Resolved ▾  
Necessário corrigir o conteúdo da página da Feature 01. Gentileza rever a especificação da feature para maiores detalhes.

```
[TestMethod]
0 references | 0 changes | 0 authors, 0 changes | 0 exceptions
public void TestMethod1()
{
    Assert.Equals(1, 2);
}
```

Help

designer.cs Feature01.aspx.cs

TestMethod1()

**Team Explorer - Changes**

Search Work Items (Ctrl+~)

**Changes** | cilab-vsts-git

Branch: dev-feat01-sprint01

Testes unitários adicionados.

Commit Staged ▾ Actions ▾

- Commit Staged
- Commit Staged and Push**
- Commit Staged and Sync

C# AssemblyInfo.cs [add]  
C# CiLabWebFormsApp.Tests.csproj [add]  
packages.config [add]  
C# UnitTest1.cs [add]

Changes (5)

- C:\temp\cilab-vsts-git\source
  - CiLabWebFormsApp
    - C# Feature01.aspx.cs \*



# Integração Contínua

The screenshot displays a GitHub Pull Request (PR) for the repository 'cilab-vsts-git'. The PR is titled 'Feature 01 completa.' and is in the 'ACTIVE' state. It was created by Matheus Neder, pushing from the 'dev-feat01-sprint01' branch to the 'master' branch. The PR is marked as 'All resolved' and includes buttons for 'Approve', 'Set auto-complete', and a dropdown menu.

On the left sidebar, the 'Policies' section is highlighted with a red box. It shows a 'Required' policy named 'Build in progress' with a play button icon. Below this, the 'Work Items' section shows 'No related work items'. The 'Reviewers' section lists 'cilab-vsts-git Team'.

The main content area shows the PR description 'Feature 01 completa.' and a list of comments. A red box highlights a comment from Matheus Neder stating 'Matheus Neder pushed 1 commit creating update 3' with a commit hash 'f2b78da7' and the message 'Testes unitários adicionados.' The comment is dated 'just now'.

Below the highlighted comment, there is a comment from Matheus Neder dated '7 minutes ago' stating 'Necessário adicionar testes unitários.' This comment is marked as 'Resolved'. Below this, another comment from Matheus Neder dated 'just now' states 'Correção realizada no commit f2b78da7'. A 'Reactivate' button is visible at the bottom of the comments section.

# Integração Contínua

The screenshot displays the Azure DevOps interface for a build named 'cilab-pull-requests-ci / Build 20170925.4'. The build status is 'Build failed'. The left sidebar shows the build steps, with 'Test Assemblies' highlighted in red. The main panel shows the build summary, including a 'Build failed' banner, a progress bar, and a table of test results. The test results table shows 1 failed test, 'TestMethod1', which failed on 'FACTORYVM-AZ177'. The error message is displayed in a red box, stating: 'Assert.Fail failed. Assert.Equals should not be used for Assertions. Please use Assert.AreEqual & overloads instead.' The stack trace is also visible, showing the test method 'TestMethod1' in 'CiLabWebFormsApp.Tests.UnitTest1'.

**Build failed**

Build 20170925.4 %  
Ran for 71 seconds (Hosted), completed 102 seconds ago

Summary Timeline Artifacts Code coverage\* Tests

Total tests 1 (+1) Failed tests 1 (+1) Pass percentage 0% (+0%) Run duration 316ms (+316ms) Test failures Test duration

Test

| Test  | Failing since | Failing build | Duration    |
|---|---------------|---------------|-------------|
| 0/1 Passed - VSTest Test Run release any... |               |               | 0:00:00.096 |
| ✖ TestMethod1 New                           | 2 minutes ago | Current build | 0:00:00.096 |

**TestMethod1**  
Failed on FACTORYVM-AZ177  
Duration 0:00:00.096, 2 minutes ago  
[View History](#)

**Error message**

Assert.Fail failed. Assert.Equals should not be used for Assertions. Please use Assert.AreEqual & overloads instead.

**Stack trace**

at CiLabWebFormsApp.Tests.UnitTest1.TestMethod1() in

**Attachments (0)** **Bugs (0)**

No attachments No linked bugs

# Integração Contínua

The screenshot displays the Azure DevOps interface for a pull request. The top navigation bar includes tabs for 'Dashboards', 'Code', 'Work', 'Build and Release', and a search bar. The pull request is titled '10 ACTIVE Feature 01 completa.' and is from 'dev-feat01-sprint01' to 'master'. The 'Policies' section on the left shows a 'Required' policy with a 'Build succeeded' status. The 'Description' section contains the text 'Feature 01 completa.' and a list of comments and approvals. A dropdown menu is open next to the 'Complete' button, showing options: 'Complete', 'Set auto-complete', and 'Abandon'. A red box highlights the 'Complete' option. A dialog box titled 'Complete pull request' is open, showing the merged PR text 'Merged PR 10: Feature 01 completa.' and the description 'Feature 01 completa.'. The dialog includes checkboxes for 'Complete linked work items after merging' (unchecked), 'Delete dev-feat01-sprint01 after merging' (checked), and 'Squash changes when merging' (unchecked). A red box highlights the 'Delete dev-feat01-sprint01 after merging' checkbox. The 'Complete merge' button is highlighted with a red box.

cilab-vsts-git ▾ Dashboards Code Work Build and Release ... Search work items

cilab-vsts-git ▾ Files Commits Pushes Branches Tags Pull Requests Fork Clone

10 **ACTIVE** Feature 01 completa.

MN Matheus Neder dev-feat01-sprint01 into master

All resolved MN MN Approve Complete ...

Overview Files Updates Commits

**Policies**

Required  
✓ Build succeeded

**Work Items** ✕ +  
No related work items

**Reviewers** ✉ +

cilab-vsts-git Team via Matheus Neder ✓  
MN Matheus Neder Approved ✓

**Description**  
Feature 01 completa.

Show everything ▾

MN Add a comment...

✓ Approved by MN Matheus Neder

MN Matheus Neder joined as a reviewer

MN Matheus Neder pushed 1 commit

3ac38c54 Teste corrigido.

MN Matheus Neder pushed 1 commit

**Complete pull request** ✕

Merged PR 10: Feature 01 completa.

Feature 01 completa.

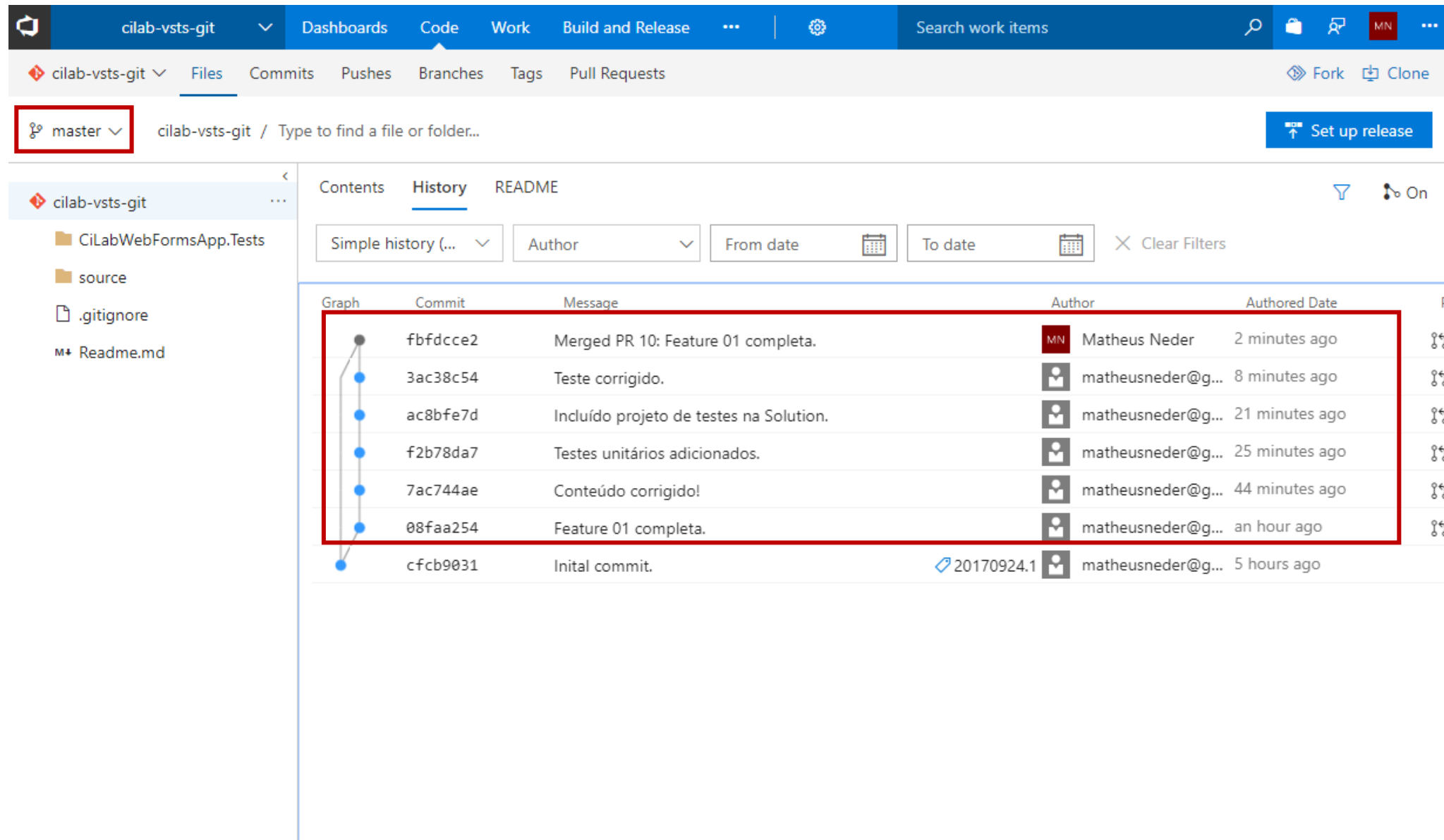
☐ Complete linked work items after merging ⓘ

☒ Delete dev-feat01-sprint01 after merging








☐ Squash changes when merging [Learn more](#)

**Complete merge** Cancel

# Integração Contínua



The screenshot shows the Azure DevOps interface for the repository 'cilab-vsts-git'. The 'master' branch is selected. The commit history is displayed in a table with columns: Graph, Commit, Message, Author, and Authored Date. A red box highlights the last six commits, which are all by 'Matheus Neder'.

| Graph   | Commit   | Message                                 | Author            | Authored Date  |
|---|----------|---|-------------------|----------------|
|    | fbfdcce2 | Merged PR 10: Feature 01 completa.      | Matheus Neder     | 2 minutes ago  |
|    | 3ac38c54 | Teste corrigido.                        | matheusneder@g... | 8 minutes ago  |
|   | ac8bfe7d | Incluído projeto de testes na Solution. | matheusneder@g... | 21 minutes ago |
|  | f2b78da7 | Testes unitários adicionados.           | matheusneder@g... | 25 minutes ago |
|  | 7ac744ae | Conteúdo corrigido!                     | matheusneder@g... | 44 minutes ago |
|  | 08faa254 | Feature 01 completa.                    | matheusneder@g... | an hour ago    |
|  | cfc9031  | Initial commit.                         | matheusneder@g... | 5 hours ago    |

# Dúvidas?

- *Dúvidas*
- *Críticas*
- *Sugestões ...*



- ✓ [matheusneder@gmail.com](mailto:matheusneder@gmail.com)
- ✓ [github.com/matheusneder](https://github.com/matheusneder)
- ✓ [linkedin.com/in/matheus-neder-66b16a16/](https://linkedin.com/in/matheus-neder-66b16a16/)