

Implementando
testes de
integração após
10 anos



Caliel Lima da Costa

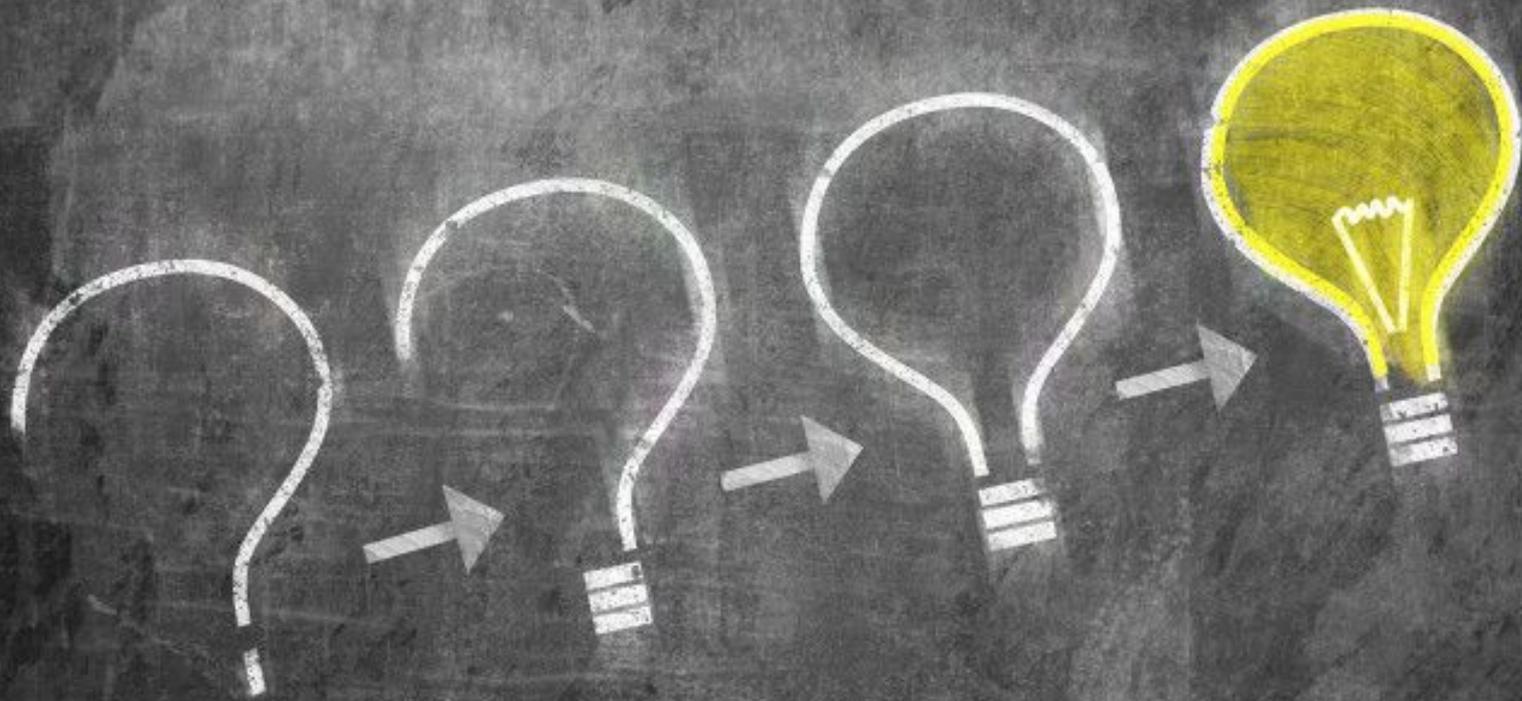
Desenvolvedor .NET com mais de 10 anos de experiência.

Trabalhou em vários projetos, alguns desde a sua concepção.

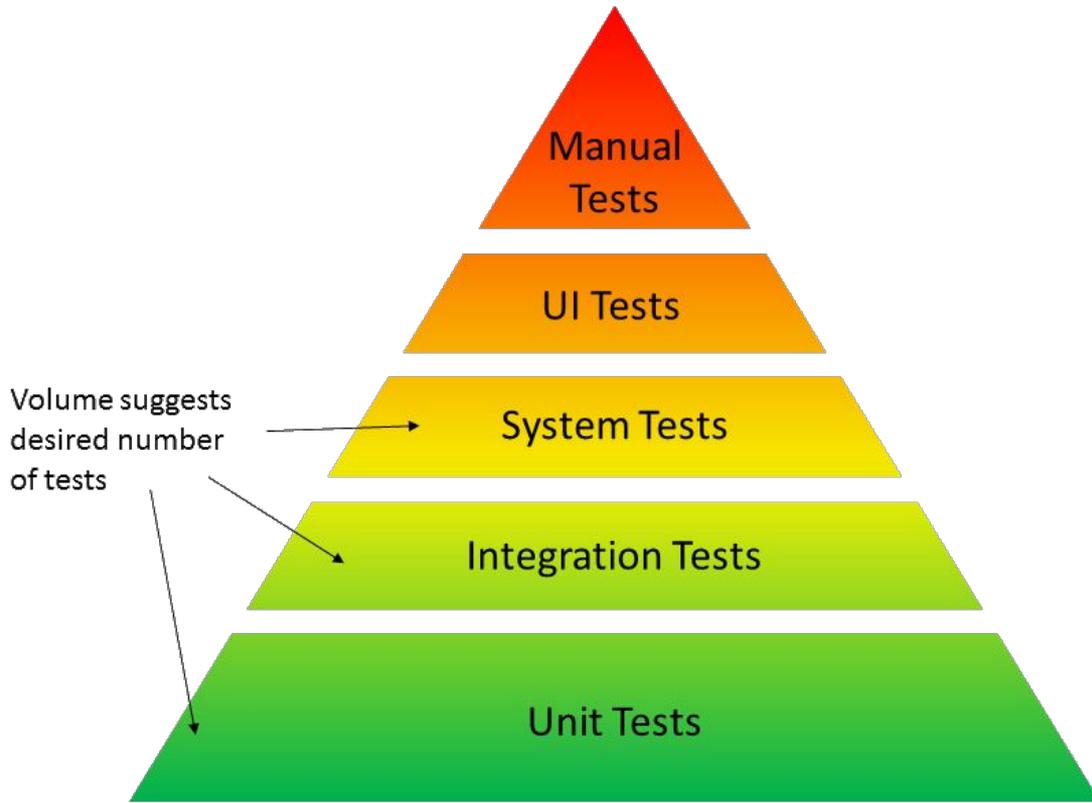
Atualmente presta seus serviços na WAES.



can't talk. coding.



Você já deve saber





Vou fazer uma
confissão

“

Até 2014 eu desenvolvi software
sem testes automatizados.

Sem testes de unidade.

Sem testes integração.

NADA. NOTHING.



“

Hoje não consigo dormir feliz quando coloco algum código em produção que não tenha cobertura de testes.

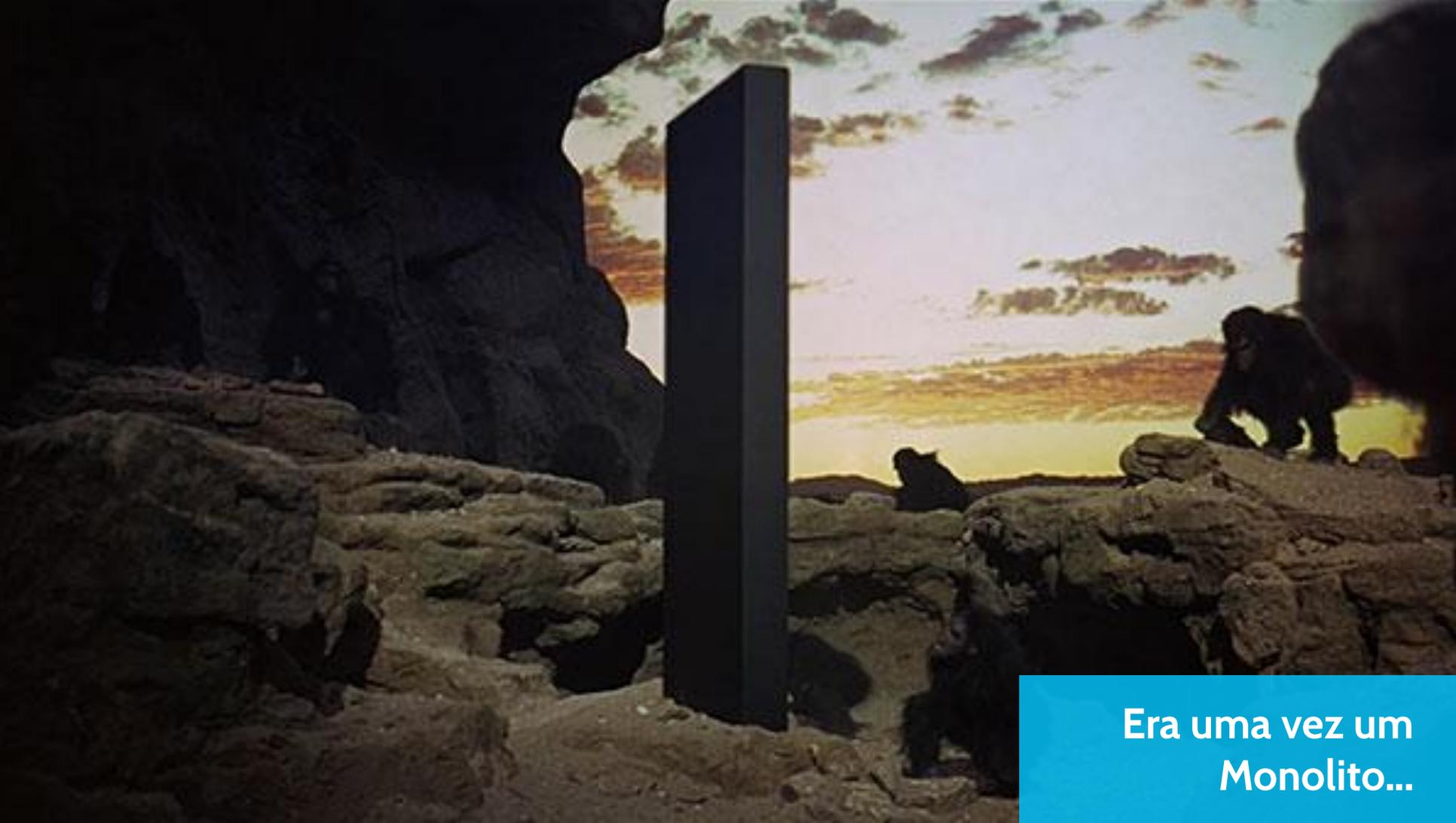
“

Mas acredito que existam muitos sistemas ainda em desenvolvimento que não possuam testes.

gifbin.com



**Então vou contar a
minha história...**



Era uma vez um
Monolito...



Alta dependência

Dependency Injection in .NET

Mark Seemann
FOREWORD BY GLENN BLOCK

 MANNING



Sem injeção de
dependência

```
public static class ConfiguracoesDoEquipamento
{
    public static DadosDoEquipamento Carregue()...
    public static void Salve(DadosDoEquipamento dados)...
```

```
public interface IConfiguracoesDoServidor
{
    DadosDoServidor Carregue();
    void Salve(DadosDoServidor dados);
}

public class ServicoDeProcessamentoDeDadosParaExportacao
{
    public static IConfiguracoesDoServidor Servidor { get; set; }
    = new ConfiguracoesDoServidor();
}
```

The
Pragmatic
Programmers

2nd Edition
Covers NUnit 2.4

Pragmatic Unit Testing

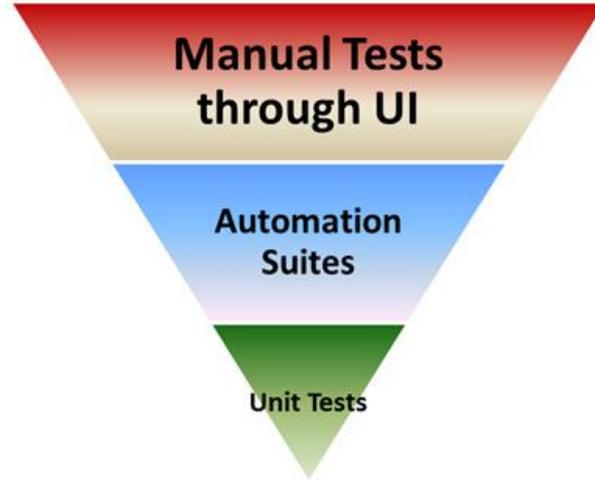
In C# with NUnit

The Pragmatic Starter Kit—Volume II



*Andrew Hunt David Thomas
with Matt Hargett*

Começo dos testes
de unidade



Traditional
(find bugs)

Baseado em testes
manuais



DIRETO GOL

CMR BRA

HYUNDAI

Replay de
execução



www.33.am

Tenta arrumar uma
coisa...



Visita do Luciano

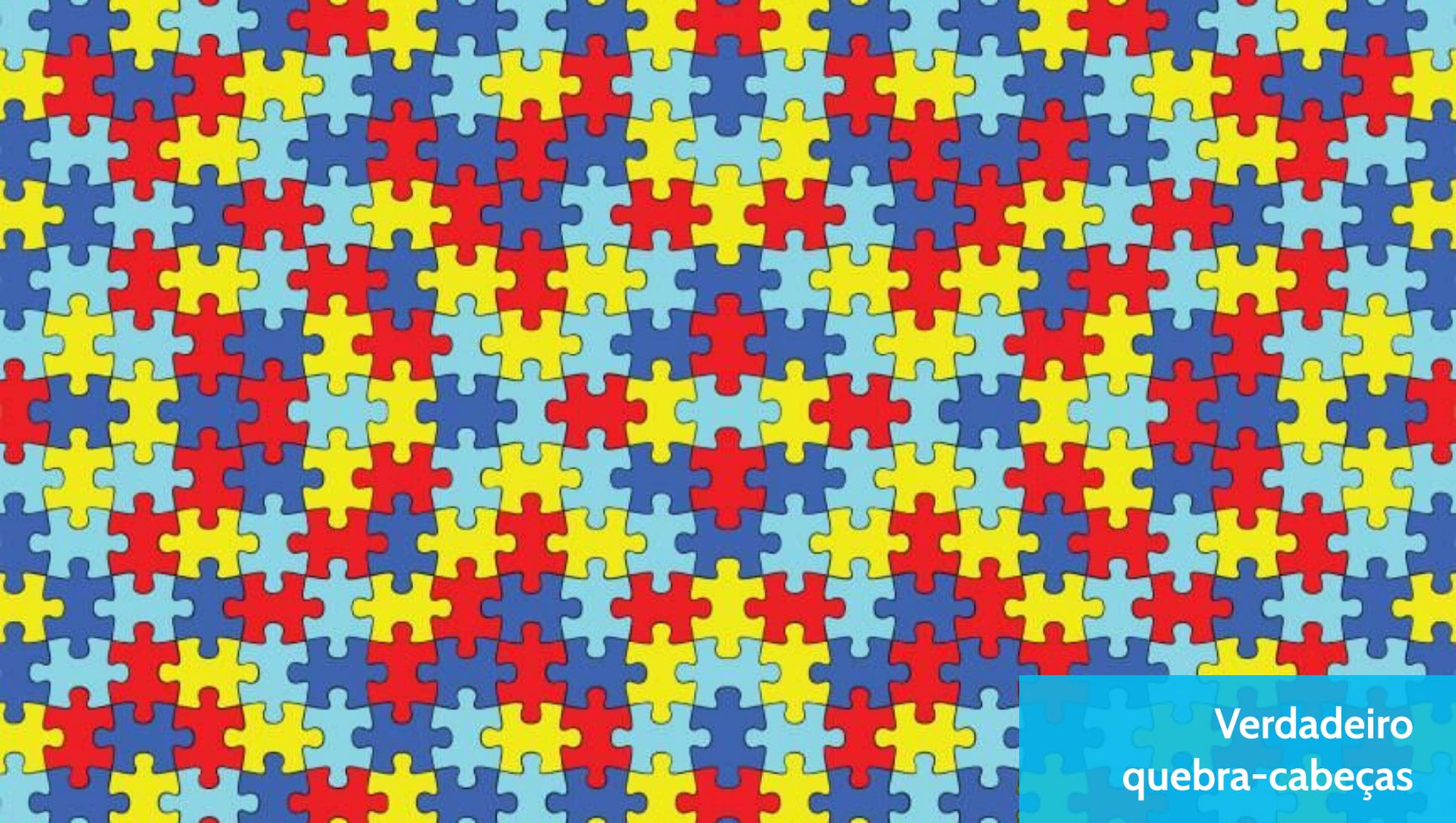
```
public class ServicoDeProcessamentoDeDadosParaExportacao
{
    public ServicoDeProcessamentoDeDadosParaExportacao(IConfigurationDoServidor servidor)...
```

```
public class ComponenteXYModulo : Module
{
    private readonly IDadosDoServidor dados;

    public ComponenteXYModulo(IDadosDoServidor dados) ...

    protected override void Load(ContainerBuilder builder)
    {
        base.Load(builder);
        if (dados.SituacaoA)
        {
            builder.RegisterType<ABC>().As<IABC>();
        }
        else
        {
            builder.RegisterType<ABC2>().As<IABC>();
        }

        if (dados.SituacaoB > 10)
        {
            builder.RegisterType<DEF>().As<IDEF>();
        }
        else
        {
            builder.RegisterType<DEF2>().As<IDEF>();
        }
    }
}
```



Verdadeiro
quebra-cabeças



2 UNIT TESTS, 0 INTEGRATION TESTS

via reddit.com/r/programmerhumor



“

Nós deveríamos ter mais testes de integração

A man with short brown hair, wearing a dark suit, a light blue striped shirt, and a dark tie, is looking upwards and to the left with a thoughtful expression. He is holding the knot of his tie with his right hand. The background features a window with a grid of small, textured glass panes. To the left, a lamp with a glowing yellow shade is mounted on a dark wooden frame. The overall lighting is warm and indoor.

Challenge Accepted!



Aprendendo sobre
o sistema

GOAL

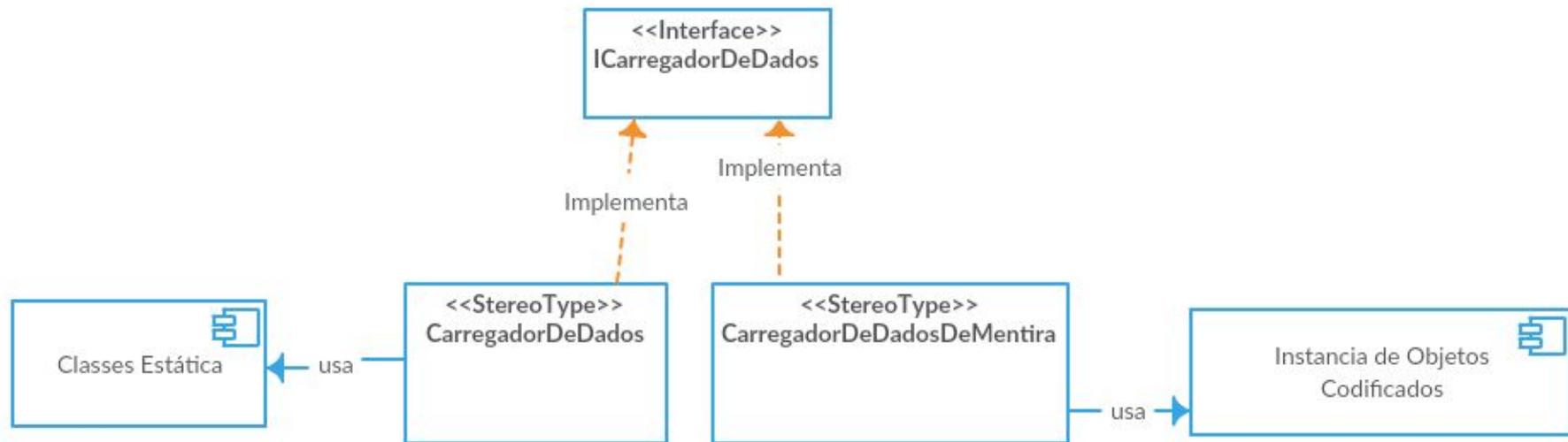
The image features the word "GOAL" in a bold, red, sans-serif font. The letter "O" is replaced by a target symbol with three concentric white circles and a red bullseye. A black dart with a silver barrel and a yellow tip is shown in a 3D perspective, appearing to have just struck the bullseye. The dart is positioned diagonally, with its head pointing towards the center of the target.

```
public class ComponenteXYModulo : Module
{
    private readonly IDadosDoServidor dados;

    public ComponenteXYModulo(IDadosDoServidor dados) ...

    protected override void Load(ContainerBuilder builder)
    {
        base.Load(builder);
        if (dados.SituacaoA)
        {
            builder.RegisterType<ABC>().As<IABC>();
        }
        else
        {
            builder.RegisterType<ABC2>().As<IABC>();
        }

        if (dados.SituacaoB > 10)
        {
            builder.RegisterType<DEF>().As<IDEF>();
        }
        else
        {
            builder.RegisterType<DEF2>().As<IDEF>();
        }
    }
}
```



Reescrevendo
código antigo

```
class ServicoXYZ;
```

```
{  
    public double CalculeAlgo()  
    {  
        var dadosDoEquipamento = CarregadorDeDadosEstatico.Leia<DadosDeVelocidadeDoEquipamento>();  
        return dadosDoEquipamento.Propriedade1 * dadosDoEquipamento.Propriedade2;  
    }  
}
```

```
public class DadosDeVelocidadeDoEquipamento
{
    public double Propriedade1 { get; set; }
    public int Propriedade2 { get; set; }
    public int Propriedade3 { get; set; }
    public int Propriedade4 { get; set; }
    public int Propriedade5 { get; set; }
    public int Propriedade6 { get; set; }
    public int Propriedade7 { get; set; }
    public int Propriedade8 { get; set; }
    public int Propriedade9 { get; set; }
    public int Propriedade10 { get; set; }
    public double Propriedade11 { get; set; }
    public double Propriedade12 { get; set; }
    public double Propriedade13 { get; set; }
    public double Propriedade14 { get; set; }
    public double Propriedade15 { get; set; }
    public double Propriedade16 { get; set; }
    public double Propriedade17 { get; set; }
    public double Propriedade18 { get; set; }
    public double Propriedade19 { get; set; }
    public double Propriedade20 { get; set; }
    public double Propriedade21 { get; set; }
}
```



**Trabalho duro
começou**

```
class ServicoXYZ
{
    private readonly IDadosDeVelocidadeMediaDoEquipamento dadosDeVelocidadeMedia;

    public ServicoXYZ(IDadosDeVelocidadeMediaDoEquipamento dadosDeVelocidadeMedia)
    {
        this.dadosDeVelocidadeMedia = dadosDeVelocidadeMedia;
    }

    public double CalculeAlgo()
    {
        return dadosDeVelocidadeMedia.Propriedade1 * dadosDeVelocidadeMedia.Propriedade2;
    }
}
```

```
public interface IDadosDeVelocidadeMediaDoEquipamento
{
    double Propriedade1 { get; }
    int Propriedade2 { get; }
}

public class DadosDeVelocidadeDoEquipamento : IDadosDeVelocidadeMedi
{
    public double Propriedade1 { get; set; }
    public int Propriedade2 { get; set; }
    public int Propriedade3 { get; set; }
    public int Propriedade4 { get; set; }
    public int Propriedade5 { get; set; }
    public int Propriedade6 { get; set; }
    public int Propriedade7 { get; set; }
    public int Propriedade8 { get; set; }
    public int Propriedade9 { get; set; }
    public int Propriedade10 { get; set; }
    public double Propriedade11 { get; set; }
    public double Propriedade12 { get; set; }
    public double Propriedade13 { get; set; }
    public double Propriedade14 { get; set; }
    public double Propriedade15 { get; set; }
    public double Propriedade16 { get; set; }
    public double Propriedade17 { get; set; }
}
```



Altera um lugar, passa pro próximo...
Altera o próximo, passa pro seguinte...



**Criando testes e
pagando débitos técnicos**



Código pronto, só falta
aprovar o Pull Request



Cada componente
é um feudo



Quando tudo
estava pronto...

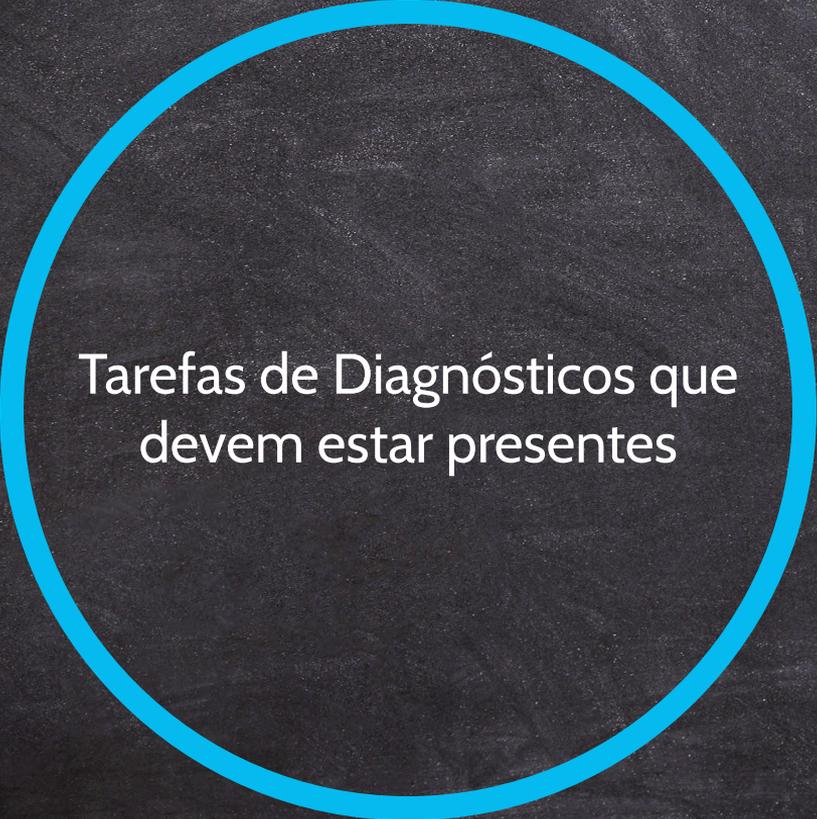


Muito tempo se
passou...

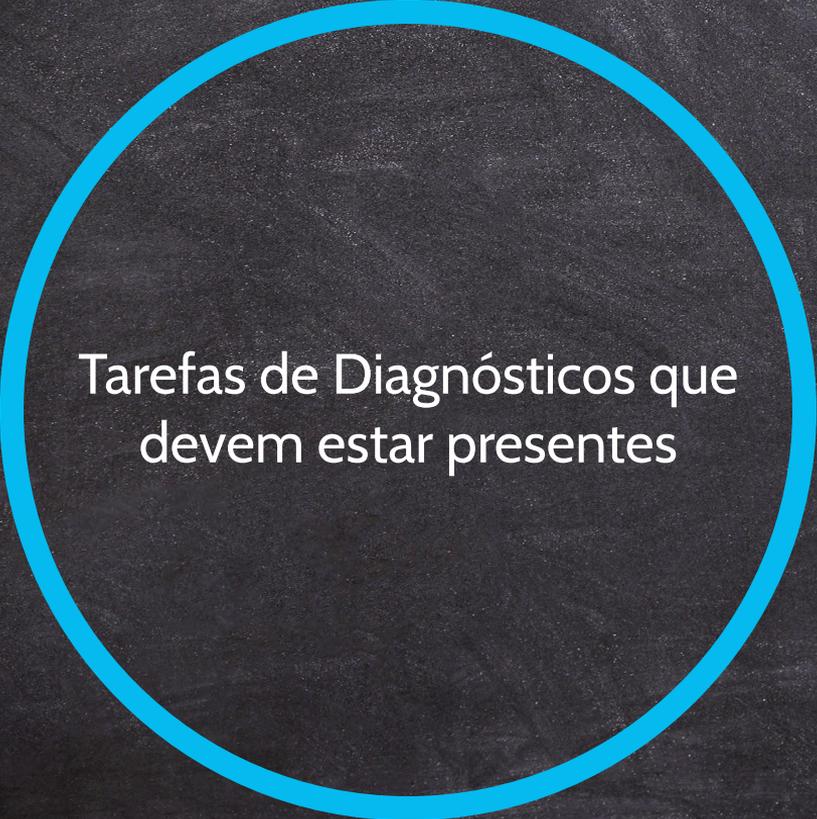


Só a ponta do
iceberg

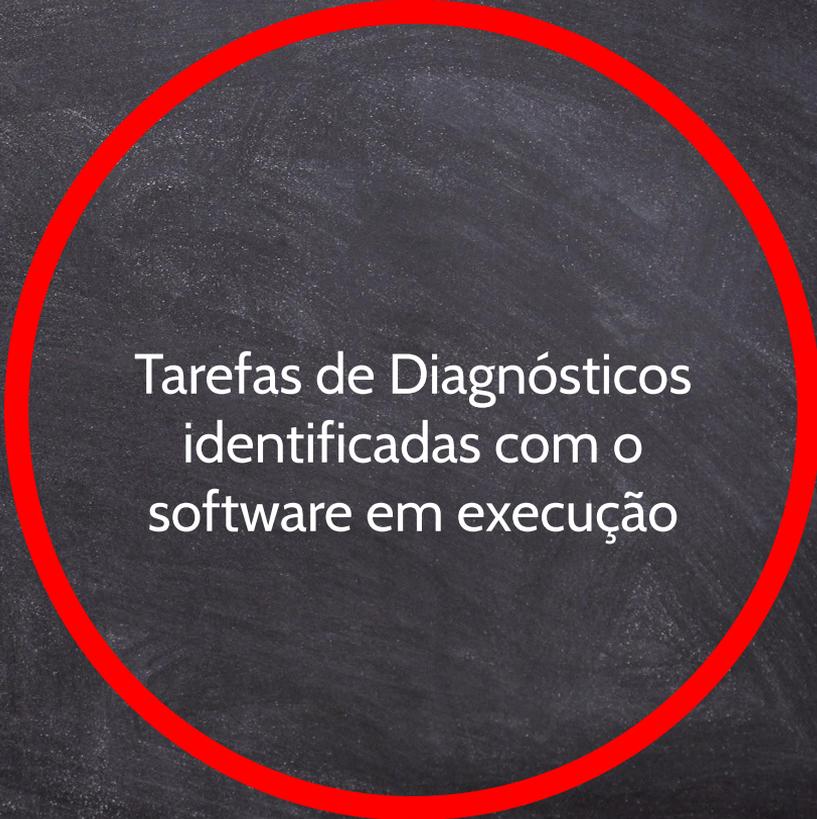




Tarefas de Diagnósticos que
devem estar presentes

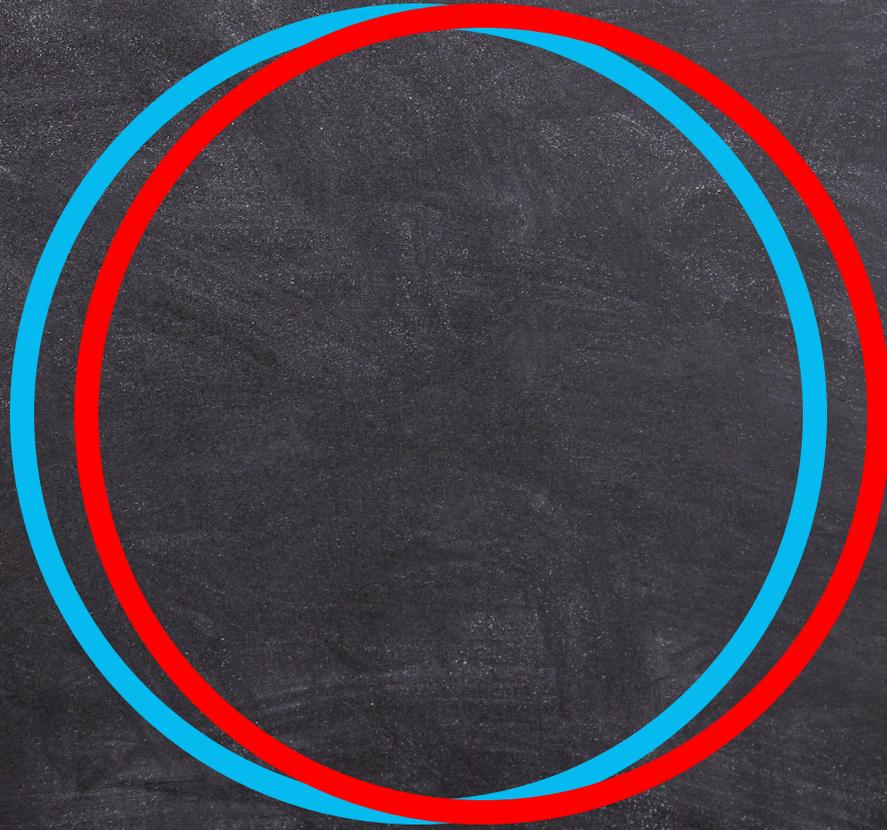


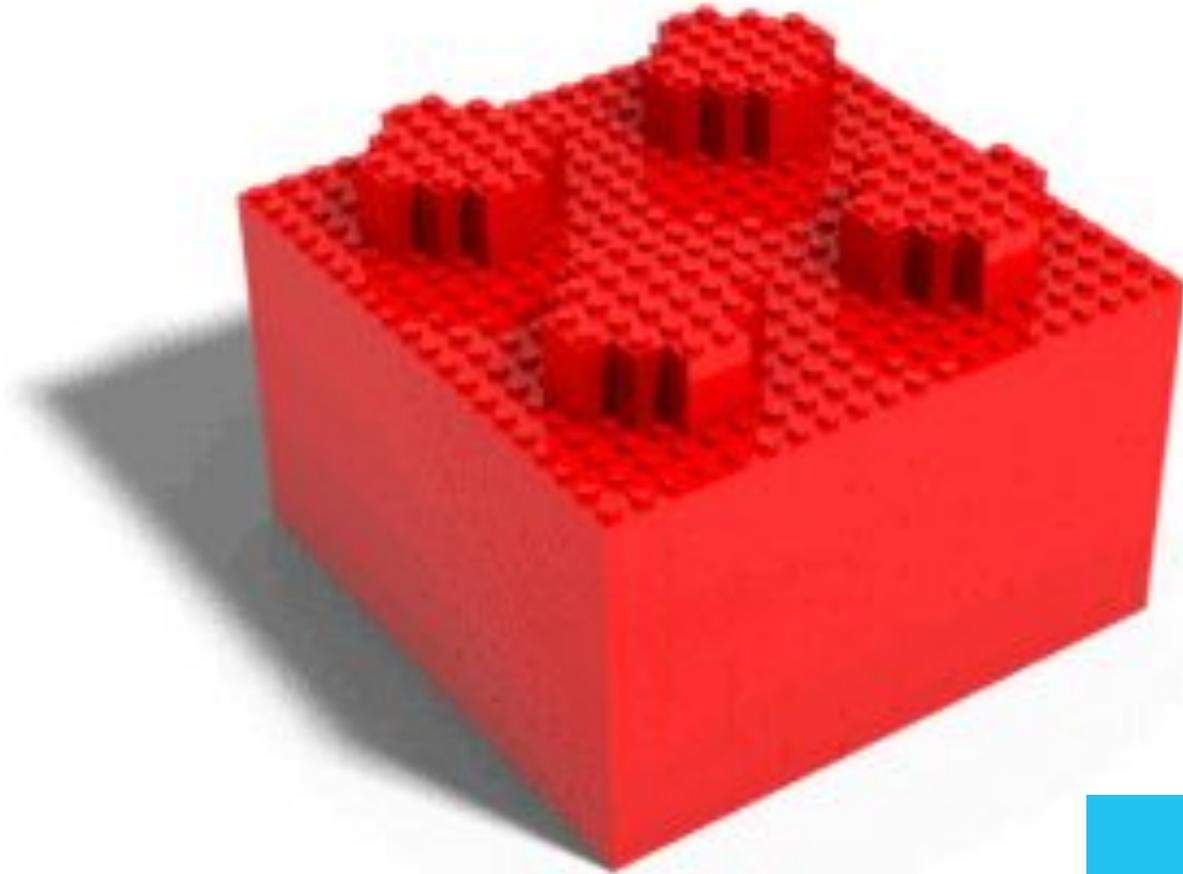
Tarefas de Diagnósticos que
devem estar presentes



Tarefas de Diagnósticos
identificadas com o
software em execução

Intersecção entre as duas listas. Se algo ficar de fora, teste falhou





Verificando cada
componente

Resultados

Encontramos 4 bugs de código já em produção

Prevenimos que outros 3 bugs fossem para produção

A signpost with two yellow directional signs. The top sign is a yellow arrow pointing to the right with the word "RIGHT" written in bold black capital letters. The bottom sign is a yellow arrow pointing to the left with the word "WRONG" written in bold black capital letters. The signpost is a silver metal pole. The background is a bright blue sky with white clouds and a bright sun in the upper right corner.

RIGHT

WRONG



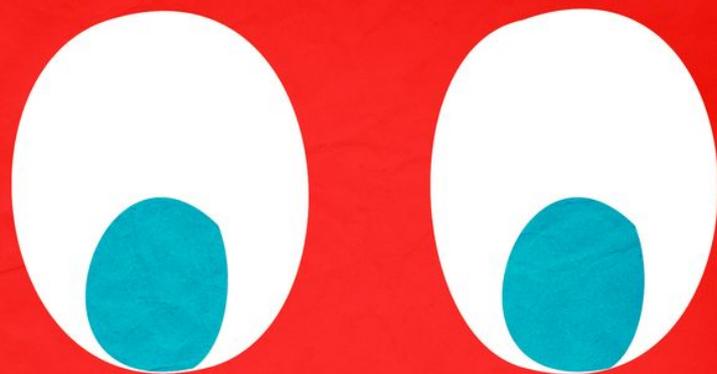
Não seja notado



SPONSORSHIP



Chame outros



THE ENEMY WILL FIND YOU AND EAT YOU
BE VIGILANT

A close-up shot of a diverse crowd of people. In the center foreground, a man with dark hair and a mustache wears a wide-brimmed straw hat and a dark shirt. He has a look of intense focus or concern. Behind him, several other people are visible, including a man with a goatee and a woman with blonde hair. The lighting is somewhat dim, suggesting an indoor or nighttime setting.

YOU CAN DO IT

Implementando
testes de
integração após
10 anos

QUESTIONS?



Caliel Lima da Costa

<https://www.linkedin.com/in/caliel/>

<https://github.com/calielc>