



07 de Outubro de 2025

XUnit

Teste de Software – Projeto de Testes com xUnit

Juliana Parreiras, Pedro Henrique Marques, Gabriel Henrique e Amanda Bicalho



O que é XUnit?

- Framework de testes unitários para .NET
- Suporta .NET Core, .NET 5/6/7/8
- Baseado em atributos [Fact] e [Theory]
- Permite testar métodos isoladamente
- Integrável com Visual Studio, VS Code e CI/CD



Por que usar?

- Garante correção da lógica do código
- Detecta erros cedo no desenvolvimento
- Facilita refatoração segura
- Permite automação de testes
- Compatível com boas práticas de TDD (Test Driven Development)

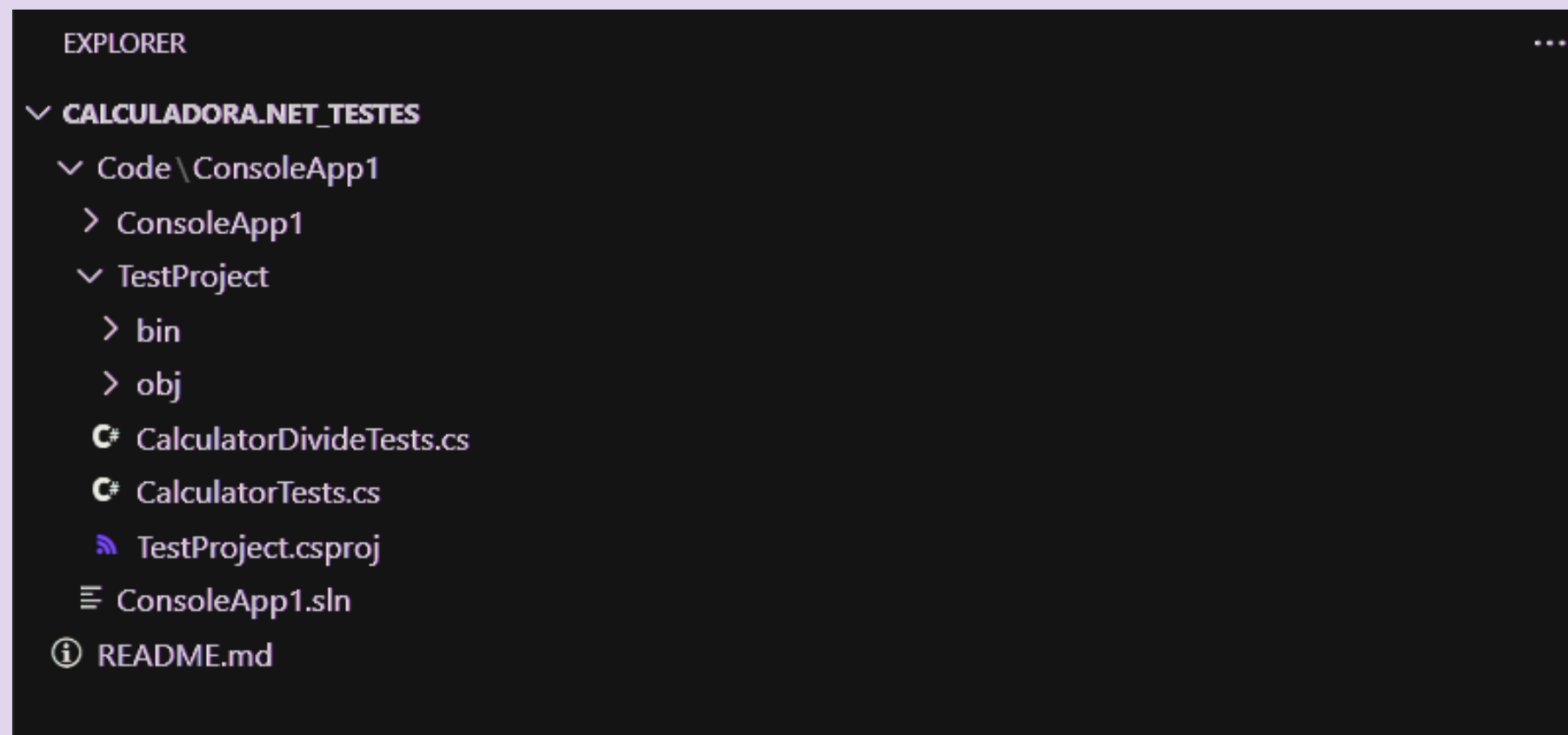


Exemplo Prático



Estrutura de testes no projeto

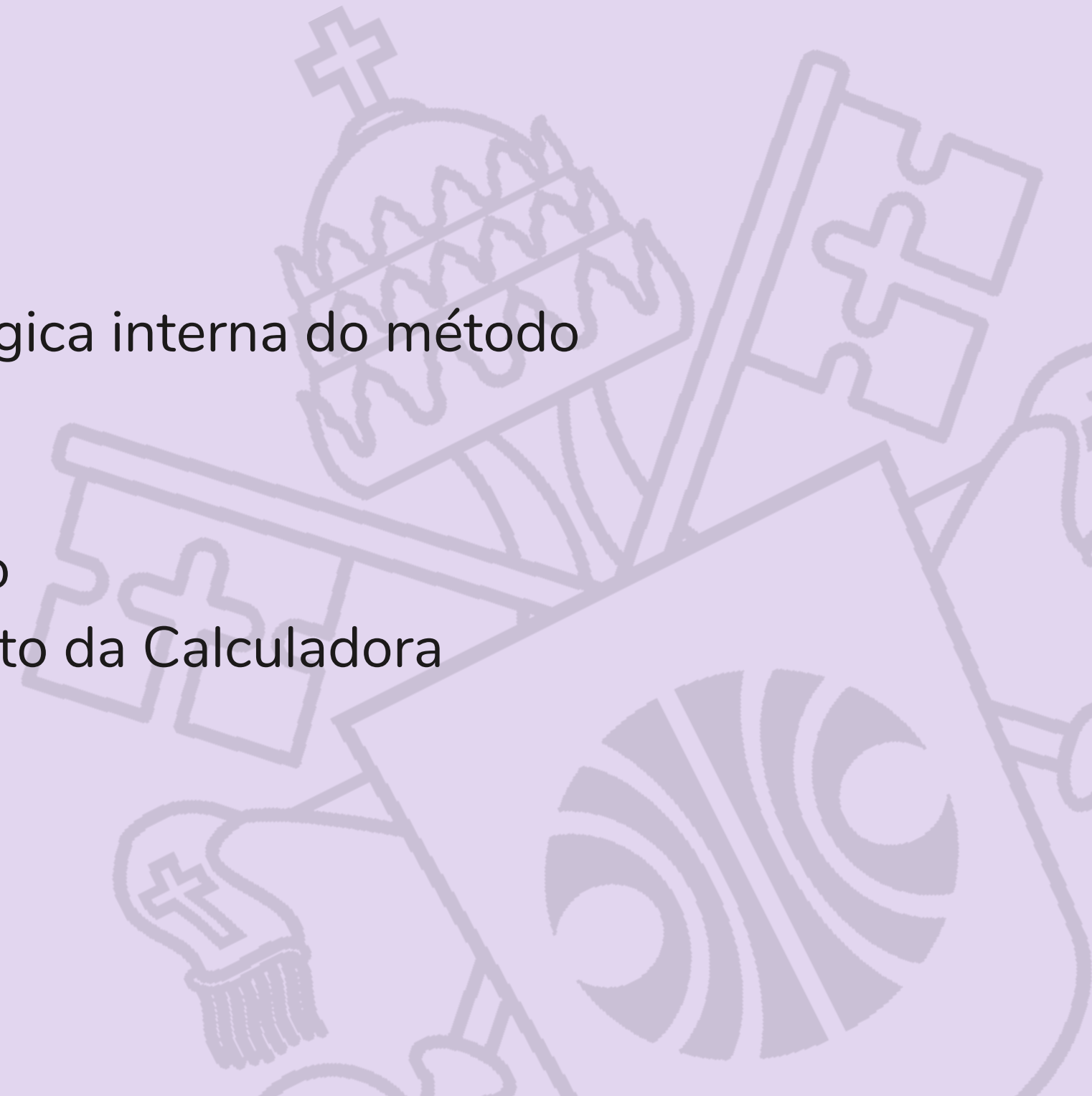
- Projeto: CalculatorTests
- Cada operação da calculadora tem classe de teste própria:
- CalculatorMultiplyTests
- CalculatorDivideTests
- Cada teste é um método [Fact]
- Acesso aos métodos private static via Reflection



Técnicas de Teste

Técnicas utilizadas pelo xUnit:

- **Caixa preta (Black-box)**
 - Valida os resultados esperados sem considerar a lógica interna do método
 - Ex.: `Multiply(2, 3) → 6`
- **Caixa branca (White-box)**
 - Permite testar fluxos internos e caminhos do código
 - Ex.: acessar métodos private via Reflection no projeto da Calculadora



Níveis de Teste

Níveis aplicáveis com xUnit:

- **Teste Unitário (principal)**
 - Cada método (Add, Subtract, Multiply, Divide) testado isoladamente
- **Teste de Integração**
 - Testa interação entre múltiplos métodos da calculadora.
 - Ex.: sequência (Add → Multiply → Divide → Subtract) → resultado final esperado.
 - Possível porque os métodos são public static, acessíveis diretamente nos testes.

Cobertura de Testes

Exemplo: Multiplicação (Multiply)

- Casos básicos: positivo, negativo, zero
- Decimais e fracionários
- Valores grandes (overflow)
- Valores especiais (NaN, Infinity, underflow)

Exemplo: Divisão (Divide)

- Casos básicos: positivo, negativo, zero
- Divisão por zero → lança exceção
- Valores especiais: NaN, Infinity, underflow



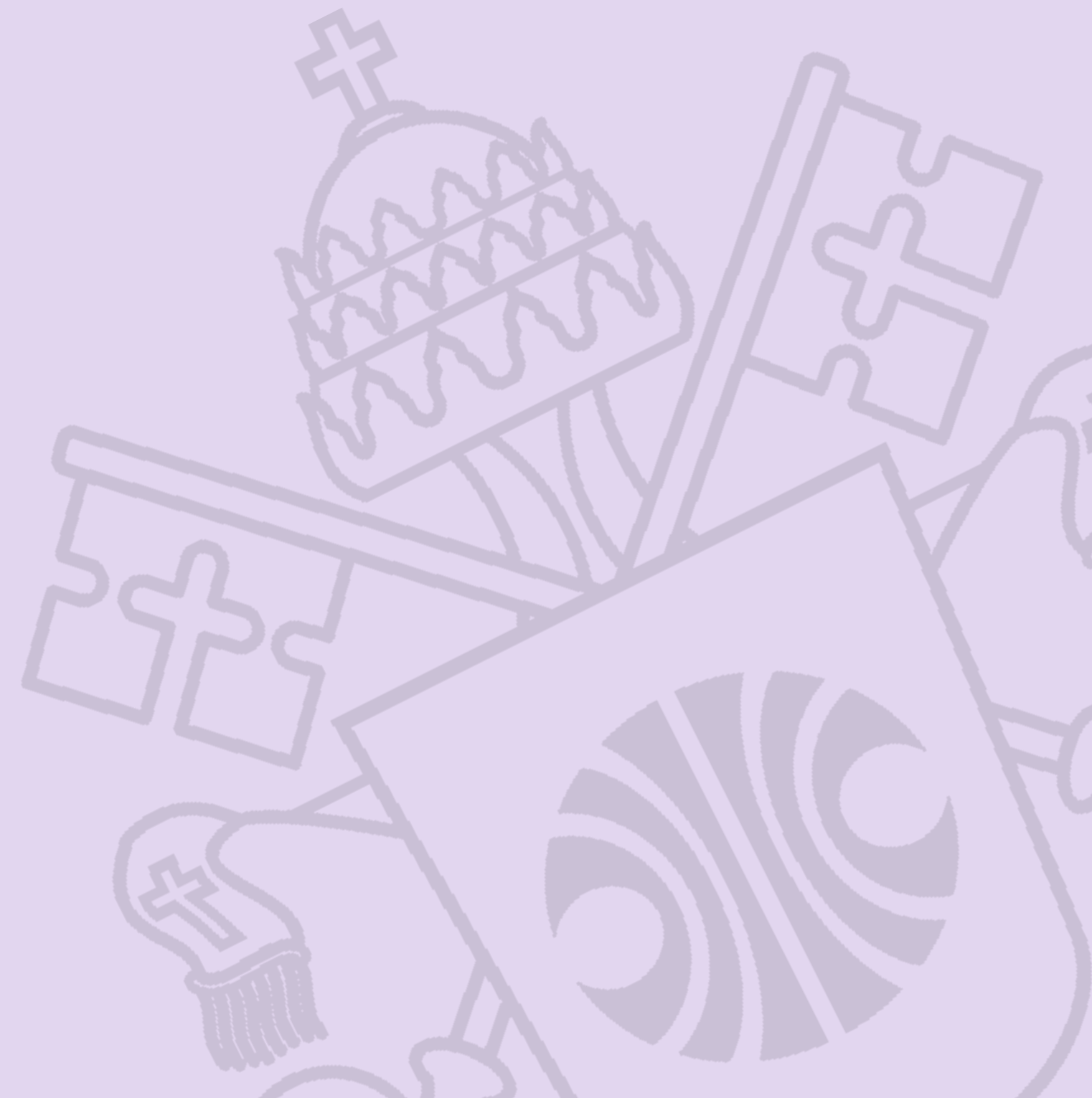
Como rodar os testes?

Pelo terminal:

```
cd TestProject  
dotnet test
```

Pelo Visual Studio / VS Code:

- Menu Test > Run All Tests
- Painel Test Explorer mostra resultados detalhados



Benefícios identificados

- Garantia de resultados corretos para operações matemáticas
- Testes de cenários extremos (overflow, divisão por zero)
- Permite alterar a calculadora com segurança
- Documenta comportamento esperado de cada método



Conclusão

- xUnit é leve, moderno e integrado ao .NET
- Testes unitários aumentam qualidade e confiabilidade do código
- No projeto Calculadora:
- Cobertura completa do Multiply e Divide
- Base sólida para adicionar testes de Add e Subtract
- Facilita manutenção e evolução do software





Obrigado! =)

