



THE
DEVELOPER'S
CONFERENCE

Testes automatizados



RAD: DELPHI C++

Leonardo Gregianin

github.com/leogregianin

Vamos falar sobre...



THE
DEVELOPER'S
CONFERENCE

- Visão geral sobre TDD
- Frameworks para testes Delphi e C++Builder
- Exemplos de unittest e integration tests
- Links sobre conceitos e ferramentas

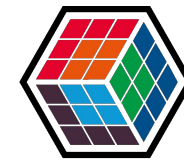
Nossos problemas...



THE
DEVELOPER'S
CONFERENCE

- Ciclo de desenvolvimento:
 1. No final do ciclo: **Testar manualmente**
 2. Se encontrar bugs: Volta para o desenvolvimento
 3. Terminado a correção dos bugs: **Testar tudo novamente**
- Novo ciclo de desenvolvimento:
 - **Testar as alterações deste ciclo de desenvolvimento**
 - **Testar todos os outros recursos do software**
- Se tudo tem que ser testado manualmente e repetidamente porque não automatizar?

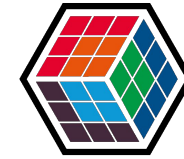
Tipos de testes



THE
DEVELOPER'S
CONFERENCE

- Teste de Unidade
- Teste de Integração
- Teste Funcional
- Teste de Aceitação
- Teste de Regressão
- Teste de Interface
- Teste de Estresse
- Teste de Usabilidade
- Teste de Compatibilidade
- Teste de Segurança
- Teste de Fumaça
- ∞
- Um tipo de teste pode ser integrado a outro(s)
 - Por exemplo: **Teste de Unidade** com **Teste de Integração** junto com **Teste de Carga**.

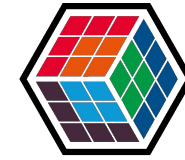
O que são testes unitários?



THE
DEVELOPER'S
CONFERENCE

- Teste de unidade é qualquer parte do código que pode ser isolado e testado (normalmente os métodos de uma classe).
- Testes unitários são parte da metodologia ágil do eXtreme Programming (XP).
- Todas as linguagens tem suporte e é extremamente utilizada.

Como?



THE
DEVELOPER'S
CONFERENCE

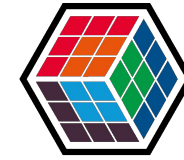
1. Escrever o teste
2. Rodar o teste
3. Ver o teste **falhar** (Lógico! Ainda não escrevi o código)
4. Escrever o código
5. Rodar o teste
6. Ver o teste **passar**

ESCREVER O TESTE



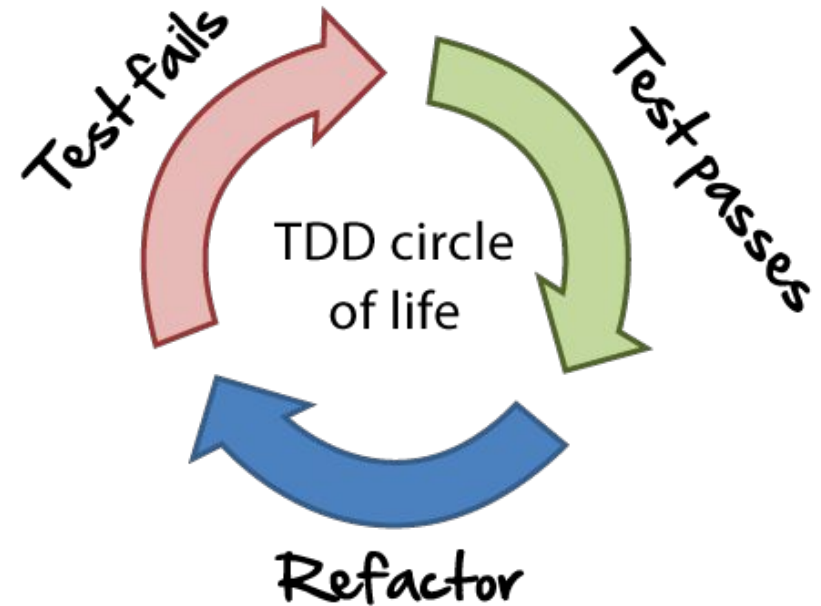
ANTES DO CÓDIGO?

Test-Driven Development

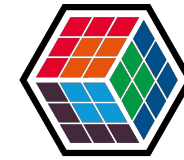


THE
DEVELOPER'S
CONFERENCE

- Escrever o teste antes do código é o conceito do **desenvolvimento orientado a testes (TDD)**
- Mudando a maneira de desenvolver



Porque?



THE
DEVELOPER'S
CONFERENCE

- **Garantia** que o código já nasce testado
- Encontrar problemas rapidamente
- **Qualidade** do código
- **Segurança** para refatorar o código, pois as funcionalidades antigas serão testadas automaticamente
- Se tudo tem que ser testado manualmente, porque não automatizar?

Frameworks



THE
DEVELOPER'S
CONFERENCE

DUnit

- Framework open source padrão de 2004 até 2015
- Projeto descontinuado a vários anos (Último commit em 2009)

DUnitX

- Framework open source padrão a partir de 2015
- Mantido pela VSoft Technologies (Final Builder/ContinuaCI)
- Qualquer classe pode conter testes
- Testes baseados em Decorators ou Published
- Framework em constante evolução (FMX, RegExp, Generics)

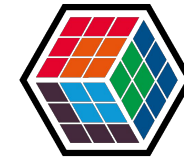
Implementação simples!



THE
DEVELOPER'S
CONFERENCE

DUnit	DUnitX
Check(Boolean)	Assert.Pass – Fail
CheckTrue(Boolean)	Assert.AreEqual – NotAreEqual
CheckFalse(Boolean)	Assert.Contains – NotContains
CheckEquals(Esperado, Atual)	Assert.IsTrue – IsFalse
CheckNotEquals	Assert.IsEmpty – IsNotEmpty
CheckNull	Assert.IsNull – IsNotNull
CheckNotNull	Assert.IsMatch para RegExp
CheckSame	Assert.WillRaise / Assert.StartsWith
Mais...	Mais...

Exemplo

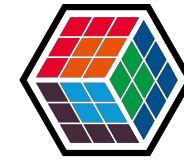


THE
DEVELOPER'S
CONFERENCE

```
class function Adicao(Valor1, Valor2: Real): Real;  
begin  
    Result := Valor1 + Valor2;  
end;
```

```
procedure TForm1.ButtonSomaClick(Sender: TObject);  
begin  
    Result.Value := Adicao(Number1.Value, Number2.Value);  
end;
```

Exemplo com DUnit



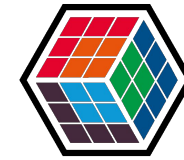
THE
DEVELOPER'S
CONFERENCE

```
TestTCalc = class(TTestCase)
public
    procedure SetUp; override;
    procedure TearDown; override;
published
    procedure TestSoma1;
    procedure TestSubtracao1;
end;

procedure TestTCalc.TestSoma1;
begin
    Result := Adicao(5,5);
    CheckEquals(10, Result, 'Erro Soma1');

    Result := Adicao(10,5);
    CheckEquals(11, Result, 'Erro Soma2');
end;
```

Exemplo com DUnit




THE
DEVELOPER'S
CONFERENCE


DUnit: An Xtreme testing framework

File Test Tree Options Actions

Test Hierarchy:

- [-] ☒ UnittestExample.exe
 - [-] ☒ TExampleTest
 - [-] ☒ TestSoma1
 - [-] ☒ TestSubtracao1

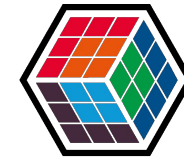
Progress:  50%

Score:  50%

Tests	Run	Failures	Errors	Overrides	Test Time	Total Time
2	2	1	0	0	0:00:00.012	0:00:00.012

Test Name	Failure Type	Message	Location
TestSoma1	ETestFailure	Erro Soma2, esperado: <11> mas está: <15>	\$004FB8D1

Exemplo com DUnitX



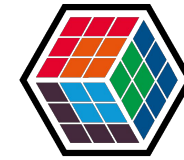
THE
DEVELOPER'S
CONFERENCE

```
TestTCalc = class(TObject) // Qualquer classe base
public
    procedure SetUp;
    procedure TearDown;

    [TestCase('TesteSoma1', '8, 2, 10')]
    [TestCase('TesteSoma2', '5, 5, 10')]
    [TestCase('TesteSoma3', '4, 2, 5')]
    [TestCase('TesteSoma4', '8000, 2, 8002')]
    [TestCase('TesteSoma5', '1, 10000, 10001')]
    procedure TesteSoma(Value1, Value2, _Result: Real);

procedure TestTCalc.TesteSoma(Value1, Value2, _Result: Real);
begin
    R := TCalc.Adicao(Value1, Value2);
    Assert.AreEqual(R, _Result);
end;
```

Exemplo com DUnitX



THE
DEVELOPER'S
CONFERENCE

```
*****
*          DUnitX - (c) 2015 Vincent Parrett & Contributors          *
*          vincent@finalbuilder.com                                   *
*                                                                    *
*          License - http://www.apache.org/licenses/LICENSE-2.0 *
*****

DUnitX - [DUnitX_Test1.exe] - Starting Tests.

.....F.....

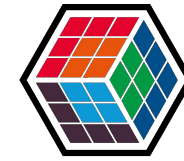
Tests Found      : 9
Tests Ignored    : 0
Tests Passed     : 8
Tests Leaked     : 0
Tests Failed     : 1
Tests Errored    : 0

Failing Tests

  TestCalcUnit.TestTCalc.TesteSoma.TesteSoma3
  Message: Expected 6 but got 5

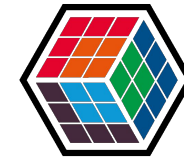
Done.. press <Enter> key to quit.
```


Demonstração



THE
DEVELOPER'S
CONFERENCE

Conclusão



THE
DEVELOPER'S
CONFERENCE

- Você pode escrever testes depois que seu código já está pronto.
- Se o seu código está muito difícil de fase testes de unidades, faça testes de integração.
- Um bug report é uma boa oportunidade para escrever um teste de unidade.
- Próximo passo: <https://github.com/leogregianin/delphi-unittest>

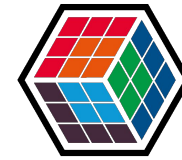
Mais...



THE
DEVELOPER'S
CONFERENCE

- TDD simples e prático: <https://goo.gl/fGRtzm>
- Porque usar TDD: <https://goo.gl/ECZEeC>
- Nick Hodges Unit Testing Video: <https://goo.gl/ZHxhTW>
- Overview Unit Testing in Delphi: <https://goo.gl/4dvpjf>
- DUnit: <http://dunit.sourceforge.net/>
- DUnitX: <https://github.com/VSoftTechnologies/DUnitX>
- Mocks: <https://github.com/VSoftTechnologies/Delphi-Mocks>
- Code coverage:
<https://github.com/VSoftTechnologies/DelphiCodeCoverageExample>
- Spec (BDD): <https://github.com/RomanYankovsky/DelphiSpec>

Muito obrigado!



THE
DEVELOPER'S
CONFERENCE

Leonardo Gregianin
leogregianin@gmail.com
github.com/leogregianin