

# Testes de Unidade com C++

## Agile Brazil 2011

Ramiro Batista da Luz

DAINF - Departamento Acadêmico de Informática  
Programa de Pós-Graduação em Computação Aplicada  
Universidade Tecnológica Federal do Paraná

26 Junho 2011

# Quem?

Ramiro Batista da Luz, programador, debian-pr [GUD-PR(2006)], grupy-pr [Grupy-PR(2007)], dojo-pr [Dojo-PR(2010)].

- Mestrando pela Universidade Tecnológica Federal do Paraná.

# Quem?

Ramiro Batista da Luz, programador, debian-pr [GUD-PR(2006)], grupy-pr [Grupy-PR(2007)], dojo-pr [Dojo-PR(2010)].

- Mestrando pela Universidade Tecnológica Federal do Paraná.
- Programador da Câmara Municipal de Curitiba.

# Onde?

- Nascido em Florianópolis-SC

# Onde?

- Nascido em Florianópolis-SC
- Aos 3 anos foi para Chapecó-SC.

# Onde?

- Nascido em Florianópolis-SC
- Aos 3 anos foi para Chapecó-SC.
- Aos 6 anos mudança para Curitiba-PR.

# Onde?

- Nascido em Florianópolis-SC
- Aos 3 anos foi para Chapecó-SC.
- Aos 6 anos mudança para Curitiba-PR.
- Estudou no Positivo, até a graduação, Bacharelado em Informática.

# Onde?

- Nascido em Florianópolis-SC
- Aos 3 anos foi para Chapecó-SC.
- Aos 6 anos mudança para Curitiba-PR.
- Estudou no Positivo, até a graduação, Bacharelado em Informática.
- Ingressou no mestrado profissional em computação aplicada da UTFPR.



# Como?

- Palestra JUnit Adolfo, [Adolfo Neto(2010)]

# Como?

- Palestra JUnit Adolfo, [Adolfo Neto(2010)]
- Sugestão de Coding Dojo na UTFPR

- Palestra JUnit Adolfo, [Adolfo Neto(2010)]
- Sugestão de Coding Dojo na UTFPR
- Seleção do Mestrado - Engenharia de Software - Métodos Ágeis - Coding Dojo [CodingDojo(2004)].

# Porque?

- Python unittest [Foundation(2001)]

# Porque?

- Python unittest [Foundation(2001)]
- Dissertação - TDD [Beck(2002)] - Coding Dojo [CodingDojo(2004)]

# Porque?

- Python Unittest [Foundation(2001)]
- Dissertação - TDD [Beck(2002)] - Coding Dojo [CodingDojo(2004)]
- Algoritmos [DAINF - UTFPR(2011a)] - Testes simples

# Porque?

- Python Unittest [Foundation(2001)]
- Dissertação - TDD [Beck(2002)] - Coding Dojo [CodingDojo(2004)]
- Algoritmos [DAINF - UTFPR(2011a)] - Testes simples
- Programação Avançada [DAINF - UTFPR(2011b)] - Foco em orientação a objetos - Testes mais avançados.

- TDD - Test Driven Development [Beck(2002)]



- TDD - Test Driven Development [Beck(2002)]
- SUnit - Original para Smalltalk. [Wikipedia(1999b)] e [Wikipedia(1972)]. Por Kent Beck. [Beck(1999)]

- TDD - Test Driven Development [Beck(2002)]
- SUnit - Original para Smalltalk. [Wikipedia(1999b)] e [Wikipedia(1972)]. Por Kent Beck. [Beck(1999)]
- xUnit - Nomenclatura usada para generalizar o padrão, muitas linguagens desenvolveram bibliotecas similares, JUnit, NUnit, PyUnit, CPPUnit, ... [Wikipedia(1999a)]

- TDD - Test Driven Development [Beck(2002)]
- SUnit - Original para Smalltalk. [Wikipedia(1999b)] e [Wikipedia(1972)]. Por Kent Beck. [Beck(1999)]
- xUnit - Nomenclatura usada para generalizar o padrão, muitas linguagens desenvolveram bibliotecas similares, JUnit, NUnit, PyUnit, CPPUnit, ... [Wikipedia(1999a)]
- Bowling Kata in C por Olve Maudal [Maudal(2007)] inspirado por Robert C. Martin [Martin(2005)]

## CPPUnit

- Portado do JUnit para C++ por Michael Feathers [Feathers(2002a)]

## CPPUnit

- Portado do JUnit para C++ por Michael Feathers [Feathers(2002a)]
- Para instalar obter a biblioteca no site [Feathers(2002b)] ou ...

## CPPUnit

- Portado do JUnit para C++ por Michael Feathers [Feathers(2002a)]
- Para instalar obter a biblioteca no site [Feathers(2002b)] ou ...
- instalar via gerenciador de pacotes(linux/debian)  
# aptitude install libcppunit-1.12-1 libcppunit-dev libcppunit-doc

## CPPUnit

- Portado do JUnit para C++ por Michael Feathers [Feathers(2002a)]
- Para instalar obter a biblioteca no site [Feathers(2002b)] ou ...
- instalar via gerenciador de pacotes(linux/debian)  
# aptitude install libcppunit-1.12-1 libcppunit-dev libcppunit-doc
- Utilizada na Disciplina de Programação Avançada [DAINF - UTFPR(2011a)]

## Características [Feathers(2002c)]

- Saída XML



## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE

## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE
- Macros auxiliares para facilitar a declaração da suíte de testes

## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE
- Macros auxiliares para facilitar a declaração da suíte de testes
- Suporte a preparação de testes hierárquica

## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE
- Macros auxiliares para facilitar a declaração da suíte de testes
- Suporte a preparação de testes hierárquica
- Registro de testes para reduzir a recompilação

## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE
- Macros auxiliares para facilitar a declaração da suíte de testes
- Suporte a preparação de testes hierárquica
- Registro de testes para reduzir a recompilação
- Plugin de teste para agilizar o ciclo de compilação/teste

## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE
- Macros auxiliares para facilitar a declaração da suíte de testes
- Suporte a preparação de testes hierárquica
- Registro de testes para reduzir a recompilação
- Plugin de teste para agilizar o ciclo de compilação/teste
- Protetor para encapsular a execução do teste

## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE
- Macros auxiliares para facilitar a declaração da suíte de testes
- Suporte a preparação de testes hierárquica
- Registro de testes para reduzir a recompilação
- Plugin de teste para agilizar o ciclo de compilação/teste
- Protetor para encapsular a execução do teste
- Executor de teste MFC (`MfcTestRunner` *MfcTestRunner*) por Baptiste Lepilleur [Baptiste Lepilleur(2002b)]

## Características [Feathers(2002c)]

- Saída XML
- Saída de texto similar a compilador para integrar com IDE
- Macros auxiliares para facilitar a declaração da suíte de testes
- Suporte a preparação de testes hierárquica
- Registro de testes para reduzir a recompilação
- Plugin de teste para agilizar o ciclo de compilação/teste
- Protetor para encapsular a execução do teste
- Executor de teste MFC (`MfcTestRunner` *MfcTestRunner*) por Baptiste Lepilleur [Baptiste Lepilleur(2002b)]
- Executor de teste gráfico baseado em QT (`QtTestRunner` *QtTestRunner*) por Baptiste Lepilleur [Baptiste Lepilleur(2002a)]



# CPPUnit - Includes

```
#include <fstream>
#include <cppunit/portability/Stream.h>
#include <cppunit/CompilerOutputter.h>
#include <cppunit/extensions/TestFactoryRegistry.h>
#include <cppunit/XmlOutputter.h>
#include <cppunit/ui/text/TestRunner.h>
```

# CPPUnit - Saída XML

```
bool xml_runner()
{
    // Get the top level suite from the registry
    CppUnit::Test *suite =
        CppUnit::TestFactoryRegistry::getRegistry().makeTest();

    CppUnit::TextUi::TestRunner runner;

    runner.addTest( suite );

    std::ofstream file( "result.xml" );
    CppUnit::XmlOutputter* outputter =
        new CppUnit::XmlOutputter(&runner.result(), file);
    runner.setOutputter(outputter);

    // use the suite factory, then execute all tests
    return runner.run( "", false );
}
```

# CPPUnit - Saída Compiler

```
bool text_runner()
{
    // Get the top level suite from the registry
    CppUnit::Test *suite =
        CppUnit::TestFactoryRegistry::getRegistry().makeTest();

    // Adds the test to the list of test to run
    CppUnit::TextUi::TestRunner runner;

    runner.addTest( suite );

    // Change the default outputter to a compiler error format outputter
    runner.setOutputter( new CppUnit::CompilerOutputter( &runner.result(),
                                                         std::cerr ) );

    // Run the tests.
    return runner.run();
}
```

```
class TestData : public CppUnit::TestFixture
{
    CPPUNIT_TEST_SUITE( TestData );
    CPPUNIT_TEST( testConstructor );
    CPPUNIT_TEST( testDiferencaAnos_MesMenor );
    CPPUNIT_TEST( testDiferencaAnos_MesMaior );
    CPPUNIT_TEST( testDiferencaAnos_MesmoMesDiaMenor );
    CPPUNIT_TEST( testDiferencaAnos_MesmoMesDiaMaior );
    CPPUNIT_TEST( testDiferencaAnos_MesmosMesDia );
    CPPUNIT_TEST( testDiferencaAnos_AniversarioSimao );
    CPPUNIT_TEST( testDiferencaAnos_AniversarioEistein );
    CPPUNIT_TEST( testDiferencaAnos_AniversarioNewton );
    CPPUNIT_TEST( testDiferencaDias_BissextosMesMenor );
    CPPUNIT_TEST( testDiferencaDias_BissextosMesMaior );
    CPPUNIT_TEST( testDiferencaDias_BissextosMesmoMesDiaMenor );
    CPPUNIT_TEST( testDiferencaDias_BissextosMesmoMesDiaMaior );
    CPPUNIT_TEST( testDiferencaDias_BissextosMesmosMesDia );
}
```

# Exemplo

- Ver TestAcademico.cpp e TestData.\*

## Google Test

- Utilizado para projetos internos na Google [Google(2008b)]

## Google Test

- Utilizado para projetos internos na Google [Google(2008b)]
- Para instalar baixar do projeto pelo site [Google(2008a)] ou ...

## Google Test

- Utilizado para projetos internos na Google [Google(2008b)]
- Para instalar baixar do projeto pelo site [Google(2008a)] ou ...
- Instalar via gerenciador de pacotes(linux/debian)  
# aptitude install libgtest-dev libgtest0



## Google Test

- Utilizado para projetos internos na Google [Google(2008b)]
- Para instalar baixar do projeto pelo site [Google(2008a)] ou ...
- Instalar via gerenciador de pacotes(linux/debian)  
# aptitude install libgtest-dev libgtest0
- Utilizada para Algoritmos [DAINF - UTFPR(2011a)] / CodingDojo [Dojo-PR(2010)]

Cartilha ou Como começar? [Google(2008c)]

- Permite rodar testes isoladamente

Cartilha ou Como começar? [Google(2008c)]

- Permite rodar testes isoladamente
- Agrupa testes relacionados em casos de teste (Test Cases)

Cartilha ou Como começar? [Google(2008c)]

- Permite rodar testes isoladamente
- Agrupa testes relacionados em casos de teste (Test Cases)
- Funciona em vários sistemas operacionais e com vários compiladores

Cartilha ou Como começar? [Google(2008c)]

- Permite rodar testes isoladamente
- Agrupa testes relacionados em casos de teste (Test Cases)
- Funciona em vários sistemas operacionais e com vários compiladores
- Não é interrompido nas falhas, permitindo corrigir vários bugs/testes num único ciclo de execução-edição-compilação

Cartilha ou Como começar? [Google(2008c)]

- Permite rodar testes isoladamente
- Agrupa testes relacionados em casos de teste (Test Cases)
- Funciona em vários sistemas operacionais e com vários compiladores
- Não é interrompido nas falhas, permitindo corrigir vários bugs/testes num único ciclo de execução-edição-compilação
- Automaticamente mantém registro dos testes definidos

Cartilha ou Como começar? [Google(2008c)]

- Permite rodar testes isoladamente
- Agrupa testes relacionados em casos de teste (Test Cases)
- Funciona em vários sistemas operacionais e com vários compiladores
- Não é interrompido nas falhas, permitindo corrigir vários bugs/testes num único ciclo de execução-edição-compilação
- Automaticamente mantém registro dos testes definidos
- Permite o reuso de recursos compartilhados entre os testes

# Google Test - Rodando testes isolados

```
// Google GTest Framework Example
// To COMPILE IT
// g++ test_arvore.cpp -lgtest_main -pthread -o test_arvore
// TO RUN
// ./test_arvore
//
// TO FILTER TEST EXECUTION
// ./test_arvore --gtest-filter=ArvoreTest.Insere
//

#include <gtest/gtest.h>
#include "arvore.cpp"

...

TEST_F (ArvoreTest, Insere) {
```



# Google Test - Test Case

```
// ./test_arvore --gtest-filter=ArvoreTest.Insere
//

#include <gtest/gtest.h>
#include "arvore.cpp"

namespace {

// The fixture for testing class Arvore.
class ArvoreTest : public ::testing::Test {

...

TEST_F (ArvoreTest, RemoveRaizSemFilhos) {
    ostream impressao;

    impressao << "Arvore vazia, nada para imprimir" << endl;

    arvore_insere (5);
    arvore_remove (5);
    ASSERT_EQ (impressao.str(), arvore_texto());
}
```

# Google Test - Não é interrompido

```
[=====] Running 17 tests from 1 test case.  
[-----] Global test environment set-up.  
[-----] 17 tests from ArvoreTest  
[ RUN      ] ArvoreTest.Insere  
test_arvore.cpp:81: Failure  
...  
[ FAILED   ] ArvoreTest.Insere (7 ms)  
[ RUN      ] ArvoreTest.CriaArvoreAltura3  
[          OK ] ArvoreTest.CriaArvoreAltura3 (0 ms)  
[ RUN      ] ArvoreTest.RemoveNo2FilhosEsquerda  
...
```

# Google Test - Registro automático

```
int main(int argc, char **argv) {  
    ::testing::InitGoogleTest(&argc, argv);  
    return RUN_ALL_TESTS();  
}
```

```
ArvoreTest() {  
    // You can do set-up work for each test here.  
}  
  
...  
  
virtual void SetUp() {  
    // Code here will be called immediately after the constructor (right  
    // before each test).  
}
```

## Básicos

- `ASSERT_TRUE(condição); EXPECT_TRUE(condição);`
- `ASSERT_FALSE(condição); EXPECT_FALSE(condição);`

- `== ASSERT_EQ(esperado, atual); EXPECT_EQ(esperado, atual);`
- `!= ASSERT_NE(val1, val2); EXPECT_NE(val1, val2);`
- `< ASSERT_LT(val1, val2); EXPECT_LT(val1, val2);`
- `<= ASSERT_LE(val1, val2); EXPECT_LE(val1, val2);`
- `> ASSERT_GT(val1, val2); EXPECT_GT(val1, val2);`
- `>= ASSERT_GE(val1, val2); EXPECT_GE(val1, val2);`

- `ASSERT_STREQ(str_esperada, str_atual); EXPECT_STREQ(str_esperada, str_atual);`
- `ASSERT_STRNE(str1, str2); EXPECT_STRNE(str1, str2);`
- `ASSERT_STRCASEEQ(str_esperada, str_atual);`  
`EXPECT_STRCASEEQ(str_esperada, str_atual);`
- `ASSERT_STRCASENE(str1, str2); EXPECT_STRCASENE(str1, str2);`

- `ASSERT_THROW(comando, tipo_exceção); EXPECT_THROW(comando, tipo_exceção);`
- `ASSERT_ANY_THROW(comando); EXPECT_ANY_THROW(comando);`
- `ASSERT_NO_THROW(comando); EXPECT_NO_THROW(comando);`



- `ASSERT_FLOAT_EQ(esperado, atual); EXPECT_FLOAT_EQ(esperado, atual);`
- `ASSERT_DOUBLE_EQ(esperado, atual); EXPECT_DOUBLE_EQ(esperado, atual);`

# Exemplo

- Ver `arvore.cpp` e `test_arvore.cpp`

- MiniDojo

# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.

# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.
- Adolfo Gustavo Serra Seca Neto - Orientador

# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.
- Adolfo Gustavo Serra Seca Neto - Orientador
- Tania Mezzadri - Algoritmos

# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.
- Adolfo Gustavo Serra Seca Neto - Orientador
- Tania Mezzadri - Algoritmos
- Jean Simão e João Alberto Fabro - Programação Avançada

# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.
- Adolfo Gustavo Serra Seca Neto - Orientador
- Tania Mezzadri - Algoritmos
- Jean Simão e João Alberto Fabro - Programação Avançada
- Henrique Pereira(@ikkebr) e Gabriel Oliveira(@GpaOliveira)



# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.
- Adolfo Gustavo Serra Seca Neto - Orientador
- Tania Mezzadri - Algoritmos
- Jean Simão e João Alberto Fabro - Programação Avançada
- Henrique Pereira(@ikkebr) e Gabriel Oliveira(@GpaOliveira)
- Arthur Furlan(@afurlan) Autor do <http://va.mu>

# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.
- Adolfo Gustavo Serra Seca Neto - Orientador
- Tania Mezzadri - Algoritmos
- Jean Simão e João Alberto Fabro - Programação Avançada
- Henrique Pereira(@ikkebr) e Gabriel Oliveira(@GpaOliveira)
- Arthur Furlan(@afurlan) Autor do <http://va.mu>
- Organizadores Agile Brazil

# Agradecimentos

- Carlos Niemeyer - Diretor de Informática e João Claudio Derosso - Presidente da Câmara Municipal de Curitiba.
- Adolfo Gustavo Serra Seca Neto - Orientador
- Tania Mezzadri - Algoritmos
- Jean Simão e João Alberto Fabro - Programação Avançada
- Henrique Pereira(@ikkebr) e Gabriel Oliveira(@GpaOliveira)
- Arthur Furlan(@afurlan) Autor do <http://va.mu>
- Organizadores Agile Brazil
- Ao público presente.

## Referencias I



Adolfo Neto.

Desenvolvimento dirigido por testes com junit.

<http://va.mu/AJ50>, May 2010.

URL <http://va.mu/AJ50>.



Baptiste Lepilleur.

SourceForge.net: QtTestRunner - cppunit.

<http://va.mu/oDe>, 2002a.

URL <http://va.mu/oDe>.

[Online; accessed 13-June-2011].



Baptiste Lepilleur.

SourceForge.net: MfcTestRunner - cppunit.

<http://va.mu/oDN>, 2002b.

URL <http://va.mu/oDN>.

[Online; accessed 13-June-2011].



Beck.

*Test driven development: by example.*

Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

ISBN 0321146530.



K. Beck.

*Kent Beck's guide to better Smalltalk.*

Cambridge University Press, Cambridge, U.K. New York, 1999.

ISBN 9780521644372.



CodingDojo.

Welcome to the coding dojo.

<http://va.mu/3e3>, 2004.

URL <http://va.mu/3e3>.

[Online; accessed 13-June-2011].



## DAINF - UTFPR.

Ementários das disciplinas - DAINF-Departamento acadêmico de informática  
- Algoritmos e Estruturas de Dados (CABS-001).

<http://va.mu/oLH>, 2011a.

URL <http://va.mu/oLH>.

[Online; accessed 13-June-2011].



## DAINF - UTFPR.

Ementários das disciplinas - DAINF-Departamento acadêmico de informática  
- Programação Avançada (CABS-002).

<http://va.mu/oNA>, 2011b.

URL <http://va.mu/oNA>.

[Online; accessed 13-June-2011].



## Dojo-PR.

Dojo-parana's profile - GitHub.

<https://github.com/dojo-parana>, 28 May 2010.

URL <https://github.com/dojo-parana>.

[Online; accessed 13-June-2011].



## M. Feathers.

SourceForge.net: cppunit.

<http://va.mu/oNB>, 2002a.

URL <http://va.mu/oNB>.

[Online; accessed 13-June-2011].



## M. Feathers.

SourceForge.net: cppunit download.

<http://va.mu/oND>, 2002b.

URL <http://va.mu/oND>.

[Online; accessed 13-June-2011].



M. Feathers.

SourceForge.net: cppunit what is.

<http://va.mu/oNE>, 2002c.

URL <http://va.mu/oNE>.

[Online; accessed 13-June-2011].



P. S. Foundation.

25.3. unittest - unit testing framework - python v2.7.1 documentation.

<http://va.mu/oMf>, 2001.

URL <http://va.mu/oMf>.

[Online; accessed 13-June-2011].



Google.

Downloads - googletest - google c++ testing framework - Downloads.

<http://va.mu/oMg>, 2008a.

URL <http://va.mu/oMg>.

[Online; accessed 13-June-2011].





Google.

Googletest - google c++ testing framework - home page.

<http://va.mu/oMi>, 2008b.

URL <http://va.mu/oMi>.

[Online; accessed 13-June-2011].



Google.

V1\_5\_Primer - googletest - getting started with google c++ testing framework.

<http://va.mu/oNF>, 2008c.

URL <http://va.mu/oNF>.

[Online; accessed 13-June-2011].



Grupy-PR.

GruPy-PR grupos do google.

<http://va.mu/oMk>, 2007.

URL <http://va.mu/oMk>.

[Online; accessed 13-June-2011].



GUD-PR.

GUD-BR-PR.

<http://va.mu/3ez>, 2006.

URL <http://va.mu/3ez>.

[Online; accessed 13-June-2011].



R. C. Martin.

ArticleS.UncleBob.TheBowlingGameKata.

<http://va.mu/oMl>, 2005.

URL <http://va.mu/oMl>.

[Online; accessed 13-June-2011].



O. Maudal.

Test-Driven development in c - geektalk.

<http://va.mu/oMm>, 27 Nov. 2007.

URL <http://va.mu/oMm>.

[Online; accessed 13-June-2011].

## Referencias VIII



Wikipedia.

Sunit - wikipedia, the free encyclopedia.

<http://va.mu/oNI>, 1999a.

URL <http://va.mu/oNI>.

[Online; accessed 13-June-2011].

Wikipedia.

<http://va.mu/AKJL>, 1999b.

[Online; accessed 20-June-2011].