UNIVEM – Centro Universitário Eurípides de Marília
Programa de Pós-Graduação em Ciência da Computação

FATEsC – Ferramenta de Apoio ao Teste Estrutural de Componentes

Vânia Somaio Teixeira
vania@fgp.com.br

Conteúdo

- Motivação e Caracterização do Problema
- Objetivo
- Ferramenta
- Referências

Motivação e Caracterização do Problema

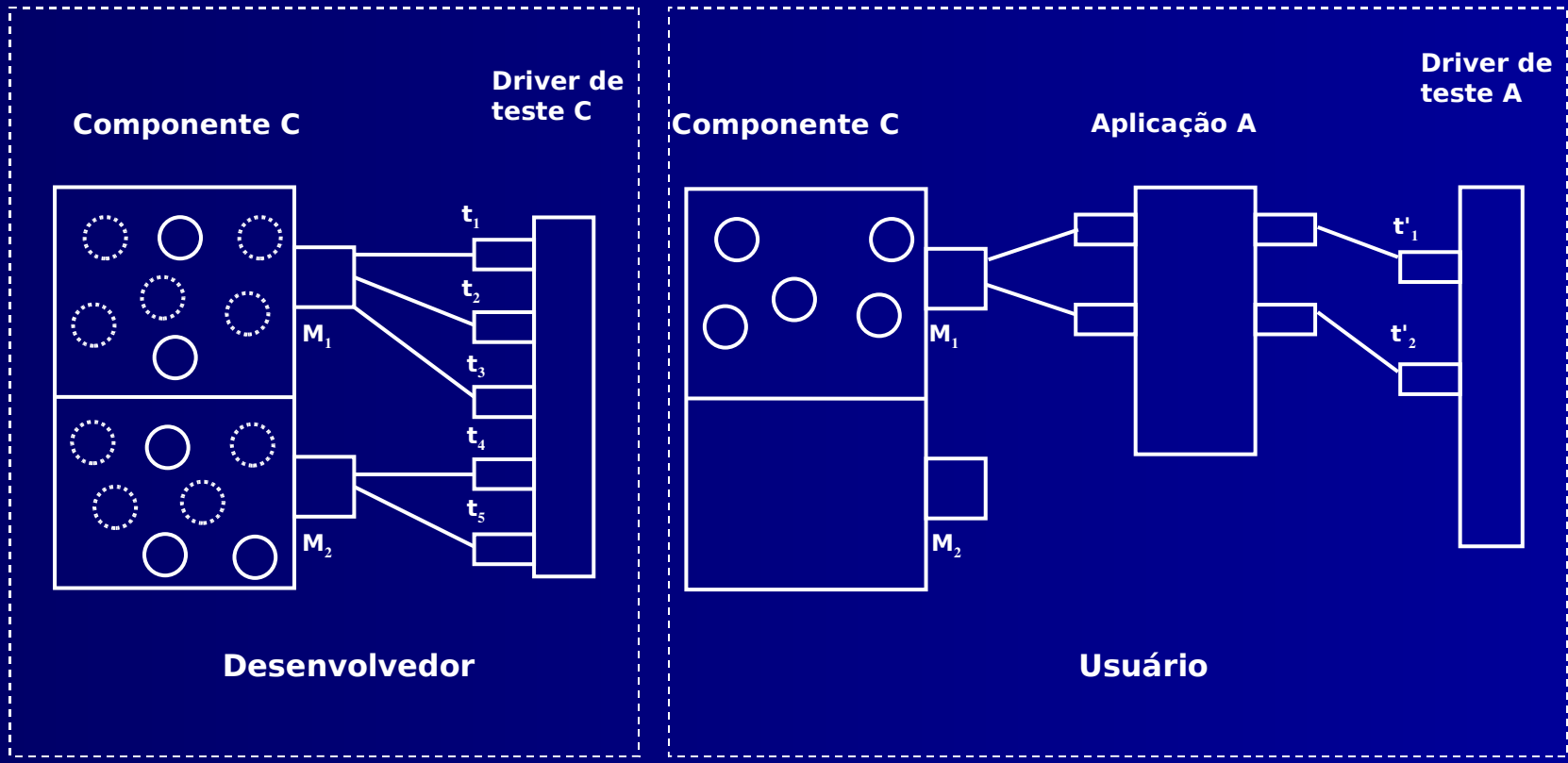
- Desenvolvimento baseado em componentes;

- Falta de Informação (Testes)
  - Desenvolvedor do componente;
  - Usuário do Componente.

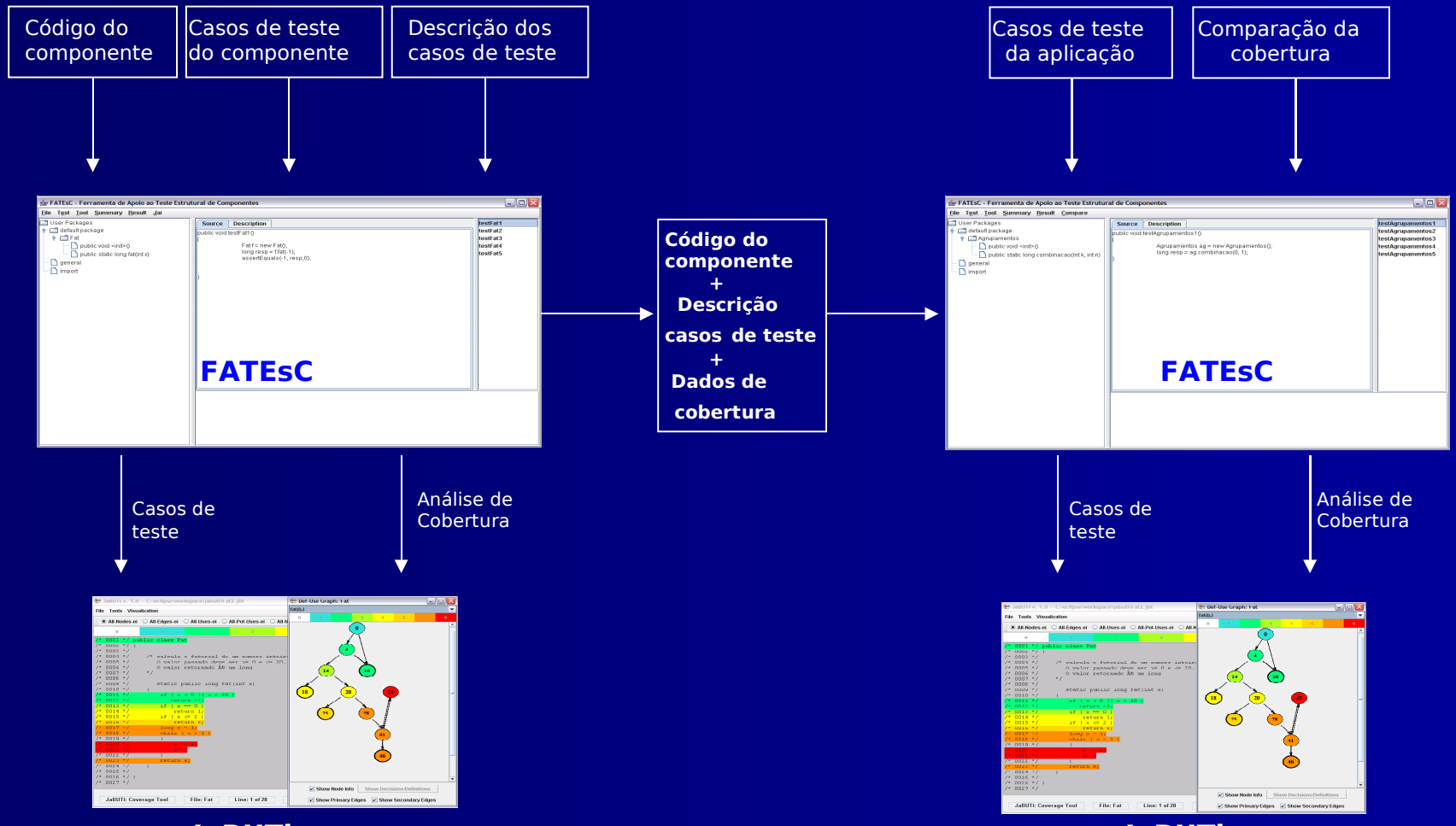
Objetivo

- Propostas Metadados;
- Disponibilizar:
  - Informações de cobertura dos requisitos de teste;
  - Descrição informal de cada casos de teste.

Premissa


Componente C
Driver de teste C
t_1
t_2
t_3
t_4
t_5
M_1
M_2
Desenvolvedor
Componente C
Aplicação A
Driver de teste A
M_1
M_2
t'_1
t'_2
Usuário

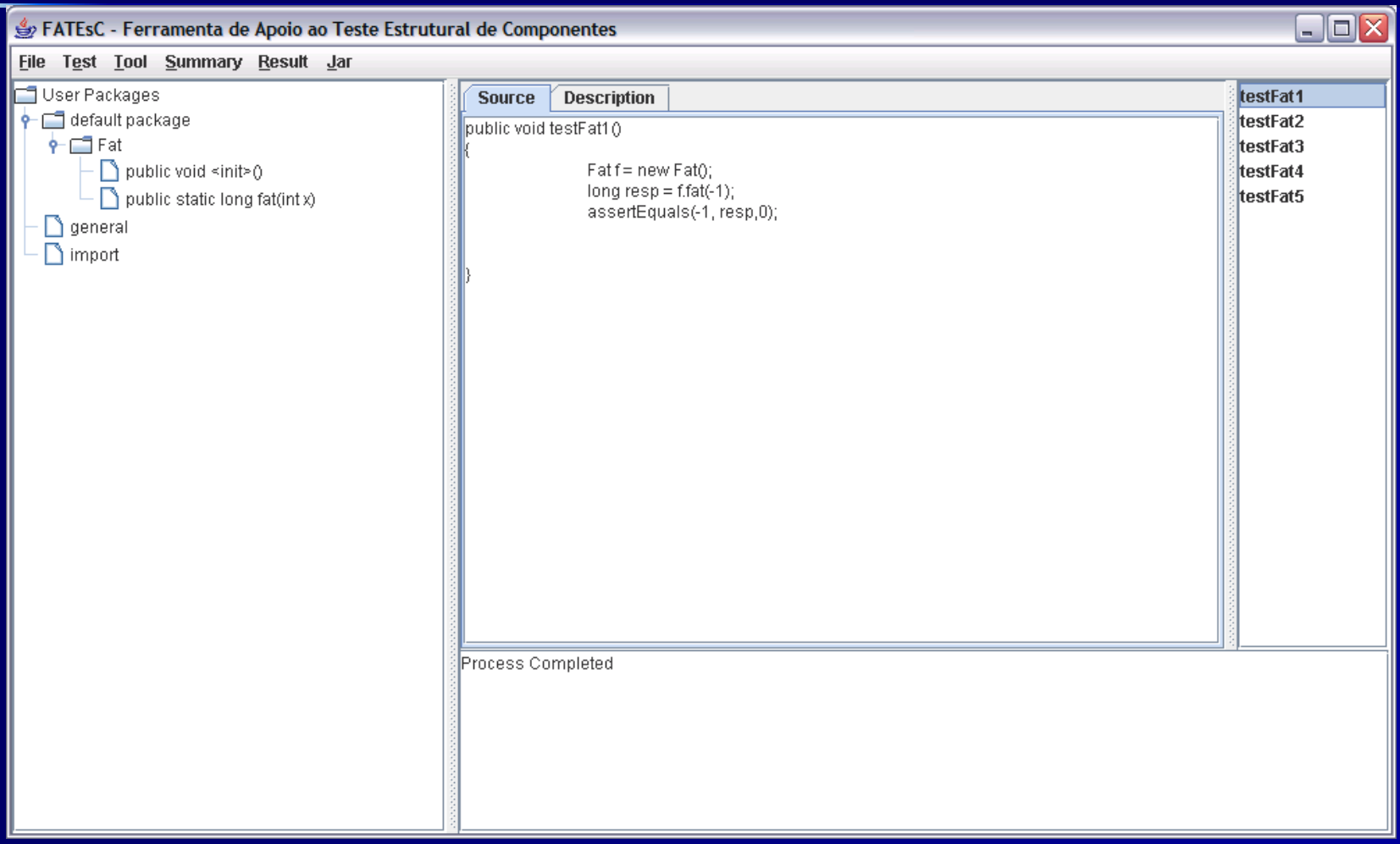
Descrição do Processo



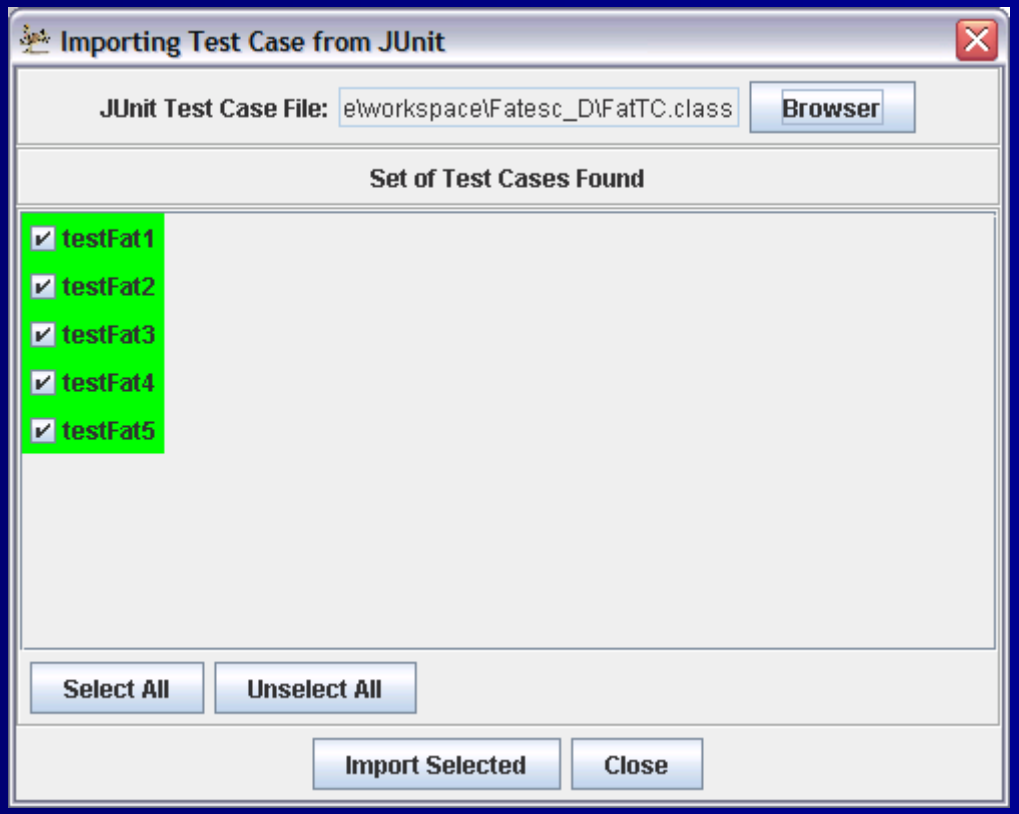
Componente Fat

```
public class Fat
{
    public long fat(int x)
    {
        if ( x < 0 || x > 20 )
         return -1;
        if ( x == 0 )
            return 1;
        if ( x <= 2 )
            return x;
        long s = 1;
        while ( x > 1 )
        {
            s *= x;
            x--;
        }
}
```

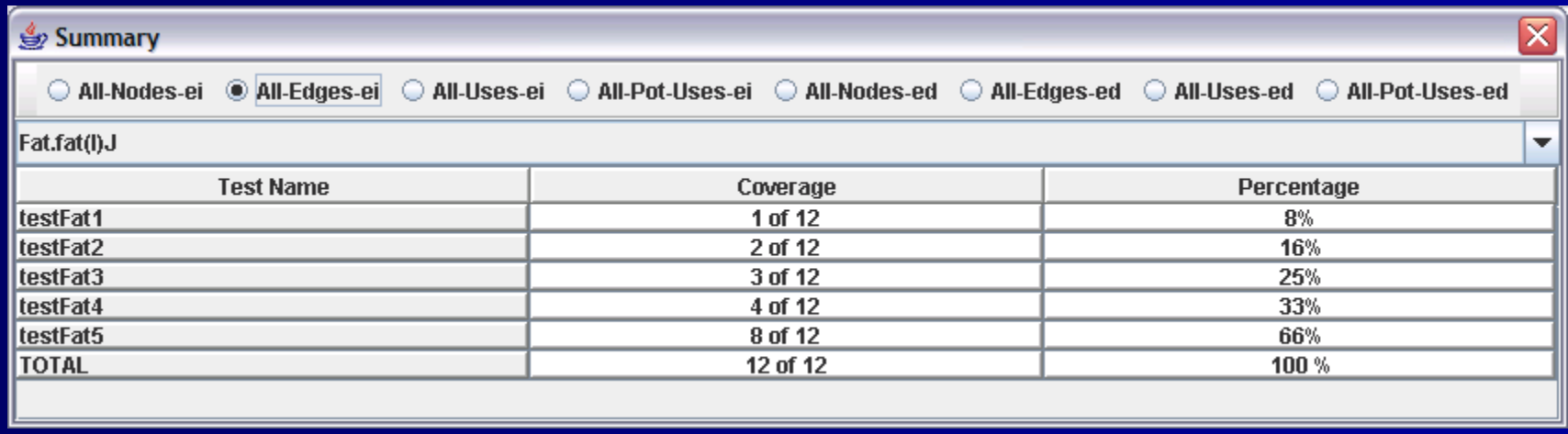
FATEsC_D


FATEsC - Ferramenta de Apoio ao Teste Estrutural de Componentes
File Test Tool Summary Result Jar
User Packages
default package
Fat
public void <init>()
public static long fat(int x)
general
import
Source
Description
public void testFat1()
{
Fat f = new Fat();
long resp = f.fat(-1);
assertEquals(-1, resp,0);
}
testFat1
testFat2
testFat3
testFat4
testFat5
Process Completed

Execução Casos de teste na JaBUTI



Resultados Obtidos



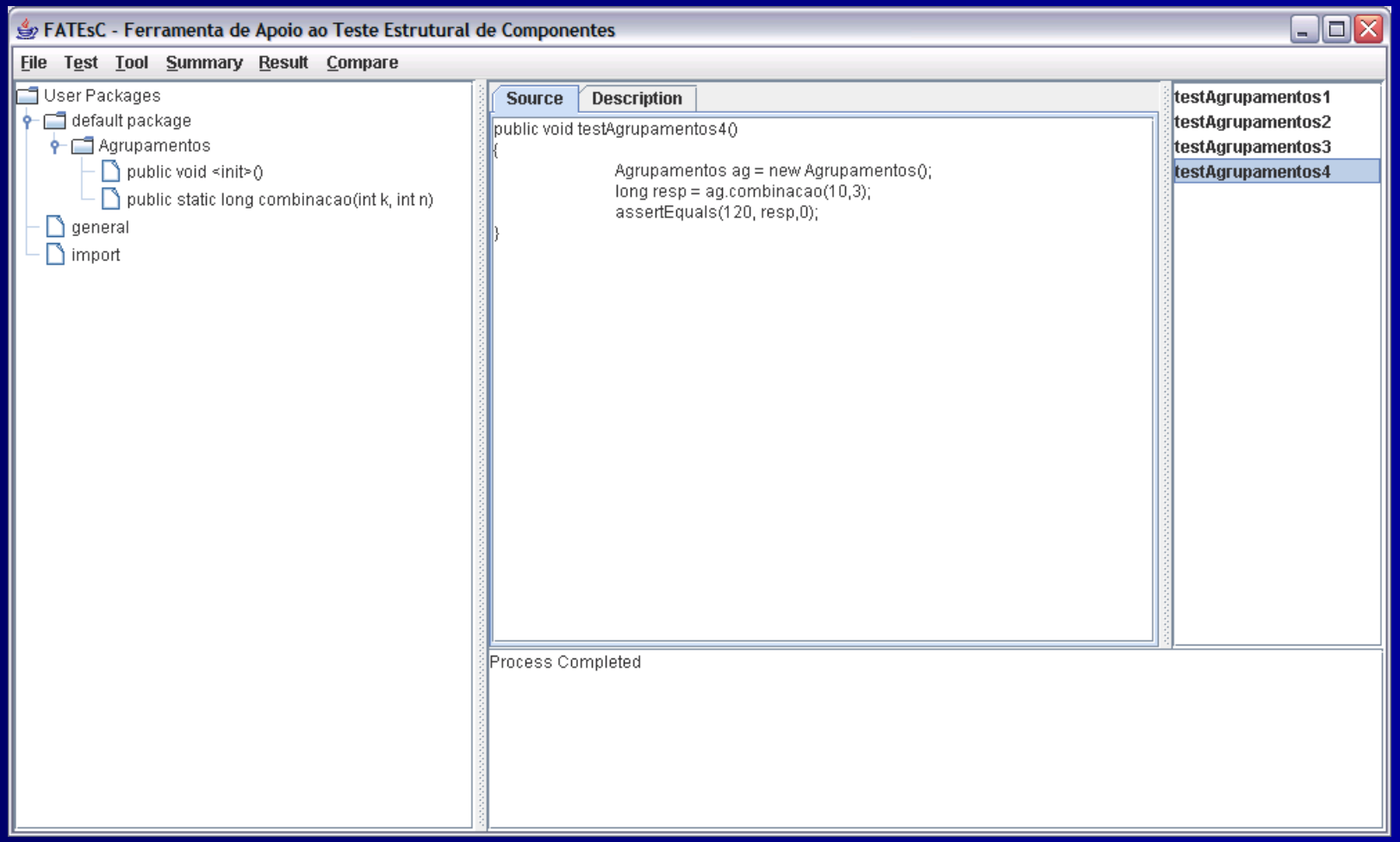
Test Name	Coverage	Percentage
testFat1	1 of 12	8%
testFat2	2 of 12	16%
testFat3	3 of 12	25%
testFat4	4 of 12	33%
testFat5	8 of 12	66%
TOTAL	12 of 12	100 %

Aplicação Agrupamentos

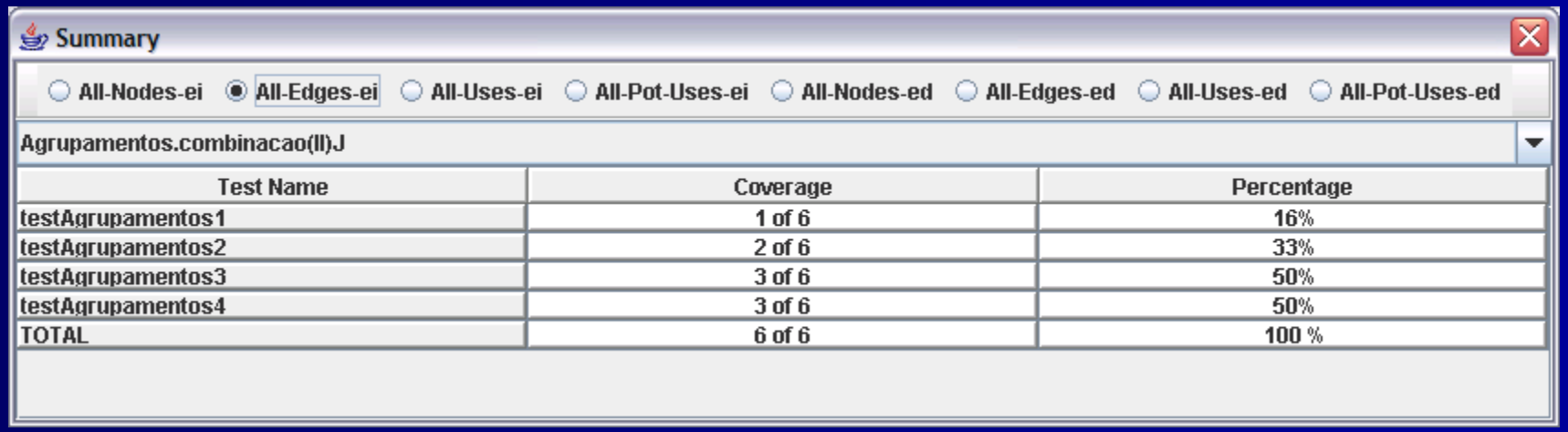
```
public class Agrupamentos {

    public long combinacao(int k, int n)
    {
        Fat f = new Fat();
            if (k < 1 || n < 1 || k < n )
          throw new IllegalArgumentException("Argumento
invalido");
        long s = f.fat(k);
        s /= f.fat(n);
        s /= f.fat(k-n);
        return s;
    }
}
```

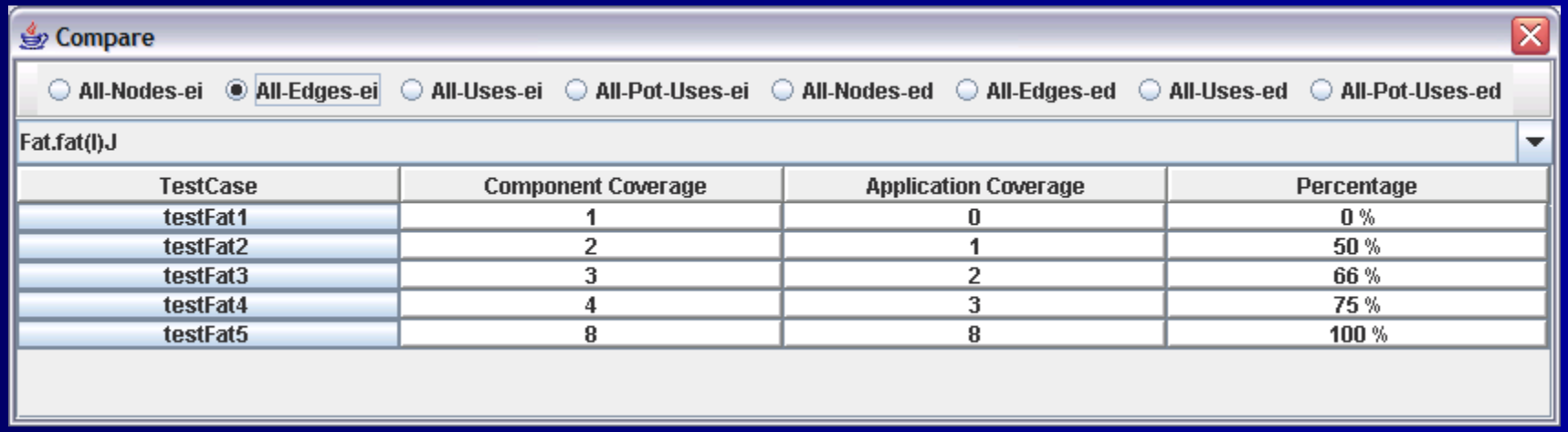
FATEsC_U


FATEsC - Ferramenta de Apoio ao Teste Estrutural de Componentes
File Test Tool Summary Result Compare
User Packages
default package
Agrupamentos
public void <init>()
public static long combinacao(int k, int n)
general
import
Source
Description
public void testAgrupamentos4()
{
Agrupamentos ag = new Agrupamentos();
long resp = ag.combinacao(10,3);
assertEquals(120, resp,0);
}
testAgrupamentos1
testAgrupamentos2
testAgrupamentos3
testAgrupamentos4
Process Completed

Resultados Obtidos

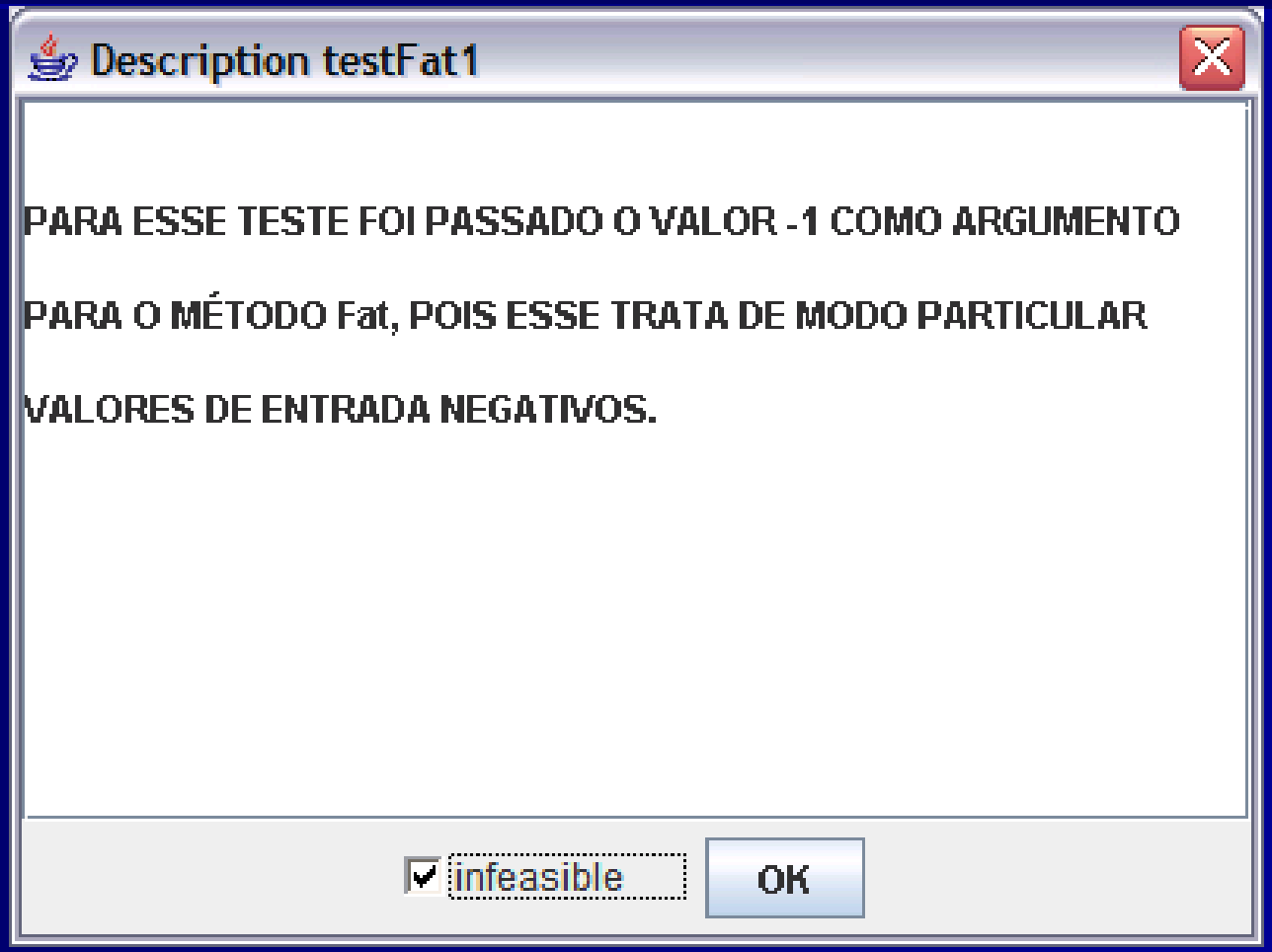


Comparação

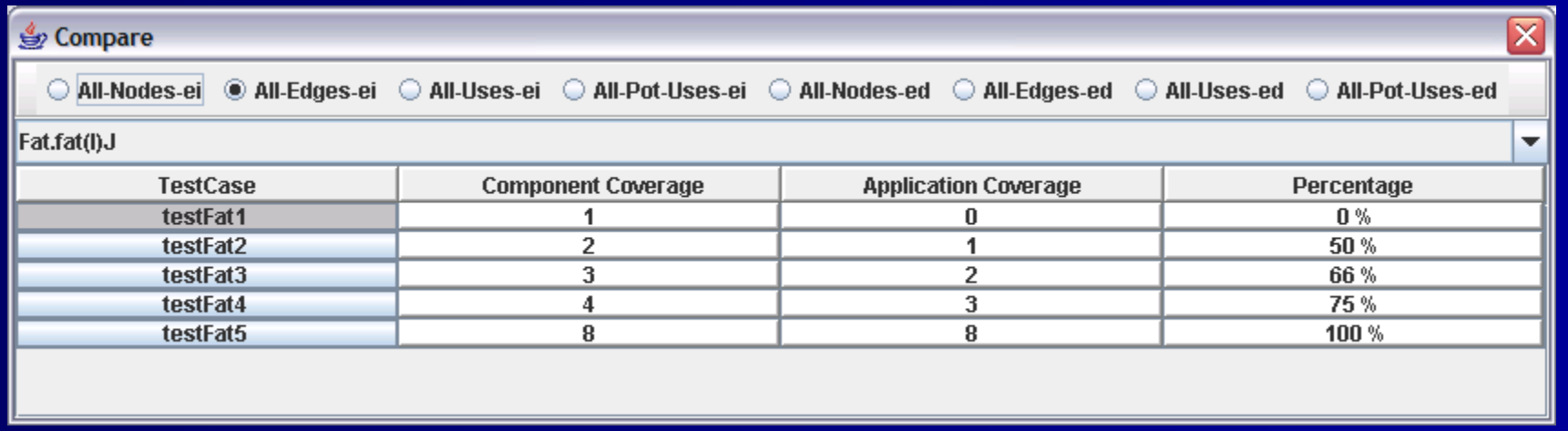


TestCase	Component Coverage	Application Coverage	Percentage
testFat1	1	0	0 %
testFat2	2	1	50 %
testFat3	3	2	66 %
testFat4	4	3	75 %
testFat5	8	8	100 %

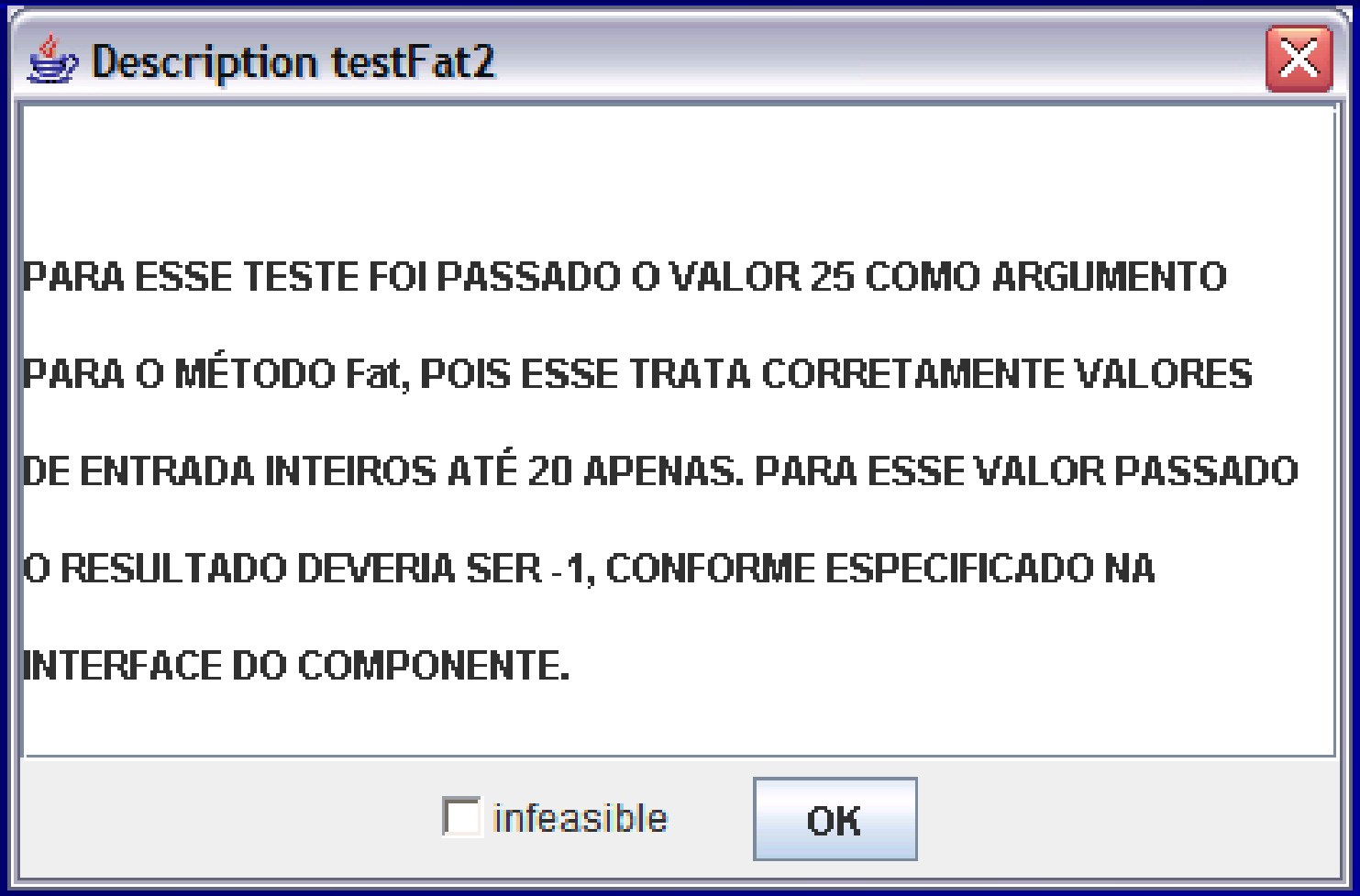
Análises


Description testFat1
PARA ESSE TESTE FOI PASSADO O VALOR -1 COMO ARGUMENTO
PARA O MÉTODO Fat, POIS ESSE TRATA DE MODO PARTICULAR
VALORES DE ENTRADA NEGATIVOS.
infeasible
OK

Análises


Compare
All-Nodes-ei
All-Edges-ei
All-Uses-ei
All-Pot-Uses-ei
All-Nodes-ed
All-Edges-ed
All-Uses-ed
All-Pot-Uses-ed
Fat.fat(I)J
TestCase
Component Coverage
Application Coverage
Percentage
testFat1 1 0 0 %
testFat2 2 1 50 %
testFat3 3 2 66 %
testFat4 4 3 75 %
testFat5 8 8 100 %

Análises


Description testFat2
PARA ESSE TESTE FOI PASSADO O VALOR 25 COMO ARGUMENTO
PARA O MÉTODO Fat, POIS ESSE TRATA CORRETAMENTE VALORES
DE ENTRADA INTEIROS ATÉ 20 APENAS. PARA ESSE VALOR PASSADO
O RESULTADO DEVERIA SER -1, CONFORME ESPECIFICADO NA
INTERFACE DO COMPONENTE.
infeasible
OK

Referências

- **BARROCA**, E.; **GIMENES**, I.; **HUZITA**, E.: **Desenvolvimento baseado em componentes**. Editora Ciência Moderna, 2005.
- **BEYDEDA**, S.; **GRUHN**, V.: **State of the art in testing components**. In: International Conference on Quality Software (QSIC), IEE Computer Society Press, 2003.
- **BUNDELL**, G. et al: **A software component verification tool** (SMT), IEEE Computer Society Press, 2000.
- **BURNSTEIN**, I.: **Pratical Software Testing**, Springer-Verlag, 2002.
- **EDWARDS, S.: Toward reflective metadata wrappers for formally specified software components. In OOPSLA Workshop Specification and Verification of Component-Based Systems, 2001.**

Referências

- LIU, C.; RICHARDSON, D.: Software component with retrospectors. In International Workshop on the Role of Software Architecture in Testing and Analysis, 1998.
- MALDONADO, J. et al: Introdução ao Teste de Software, Notas Didáticas do ICMC, São Carlos 2004.
- ORSO, A.; HARROLD, M.; ROSENBLUM, D.: Component metadata for software engineering task. In International Workshop on Engineering Distributed Object (EDO), London, UK, Springer-Verlag, 2000.
- PRESSMAN, R.: Engenharia de Software. Editora McGraw-Hill, 2002.
- VINCENZI, A. et al: Software baseado em componente: uma revisão sobre teste. In: Desenvolvimento baseado em componente. Editora Ciência Moderna, 2005.