

CURSO DE TESTES AUTOMATIZADOS COBERTURA

Wagner Costa
wcaquino@gmail.com

CONTEXTO

- Os bons desenvolvedores já se convenceram que têm que desenvolver testes unitários para as classes que desenvolvem.
 - Eles estão verificando todas as partes do código?

COBERTURA

- Cobertura de código é uma métrica aplicada nos testes de software que descreve o percentual de código que foi testado em um programa.
- Técnica aplicada em princípios de testes caixa branca, porém é possível utilizá-la em testes caixa cinza.

STATEMENT X DECISION COVERAGE

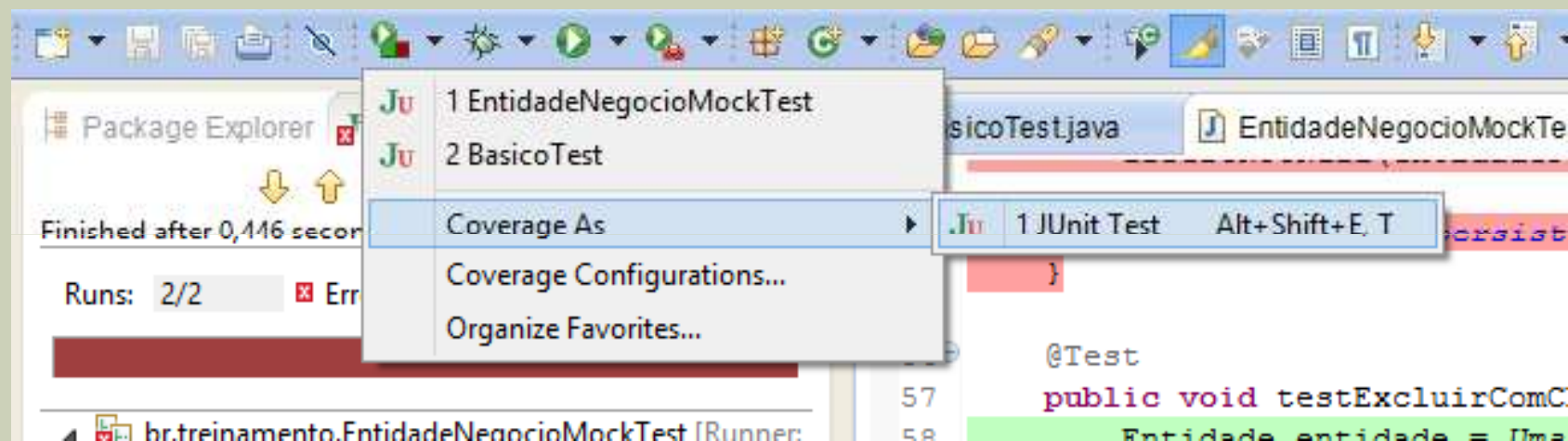
- **Statement:** Cada linha é contada uma única vez.
 - 100 mil linhas de código com 80 mil linhas testadas = 80% de cobertura

- **Decision:** Linhas de código que possibilitem um ou mais caminhos devem ser completamente testados
 - Estruturas lógicas e de repetições devem ser testados completamente
 - If (valor > 100)

ECLEMMMA

- Utiliza o JaCoCo
- Não invasivo
- Download gratuito em eclemma.org
- Instalável também pelo Eclipse Marketplace Client. Eclipse 3.6+

COMO RODAR OS TESTES?



RELATÓRIO DE EXECUÇÃO

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
Testes	12,1 %	226	1.644	1.870
src	11,5 %	147	1.132	1.279
br.treinamento	11,5 %	147	1.132	1.279
EntidadeDAO.java	0,5 %	3	580	583
Entidade.java	19,6 %	51	209	260
EntidadeNegocio.java	29,6 %	63	150	213
BancoUtils.java	0,0 %	0	126	126
ConnectionFactory.java	0,0 %	0	35	35
Basico.java	0,0 %	0	16	16
EntidadeBuilder.java	65,2 %	30	16	46
test	13,4 %	79	512	591
br.treinamento	13,4 %	79	512	591
EntidadeDAOJUnitRefact.java	0,0 %	0	217	217
EntidadeDAOJUnitTest.java	0,0 %	0	161	161
BasicoTest.java	0,0 %	0	84	84
EntidadeNegocioMockTest.java	61,2 %	79	50	129

CDD

```
26
27 private void validarRegras(Entidade entidade) throws Exception{
28     if(entidade.getNome().length() <= 4)
29         throw new Exception("O nome não pode ter menos que 5 caracteres");
30     if(entidade.getNome().length() >= 30)
31         throw new Exception("O nome não pode ter mais que 30 caracteres");
32     if(entidade.getDataInicial() != null && entidade.getDataInicial().compareTo(new Date()) < 0)
33         throw new Exception("A data inicial não pode ser menor que a data atual");
34     if(entidade.getDataInicial() != null && entidade.getDataFinal().compareTo(entidade.getDataInicial()) < 0)
35         throw new Exception("A data final não pode ser menor que a data inicial");
36     if(entidade.getTipoDocumento() != 1 && entidade.getTipoDocumento() != 2)
37         throw new Exception("Tipo de documento inválido");
38     if(entidade.getEmail() != null && !(entidade.getEmail().contains("@") || entidade.getEmail().contains(".")))
39         throw new Exception("Endereço de email inválido");
40 }
41
```

- Linha **verde** = Linha de código **TOTALMENTE** coberta
- Linha **amarela** = Linha de código **PARCIALMENTE** coberta
- Linha **Vermelha** = Linha de código **NÃO** coberta

GRANULARIDADE DE VERIFICAÇÕES

- Instructions
- Branchs
- Lines
- Methods
- Types
- Complexity

A BUSCA PELO 100%



- Muitos tentam chegar ao 100% de cobertura, poucos conseguem.
- É necessário?

100% É O LIMITE?

- 100% de código coberto não significa 100% de pontos de falhas cobertos.

```
public int somar(String a, String b){  
    return Integer.parseInt(a) + Integer.parseInt(b);  
}
```

COMO ESCOLHER O QUE TESTAR?

Priorizar

- Métodos de negócio.
- Métodos destinados a reuso. (utilitários)
- Métodos que serão integrados com outros serviços.
- Métodos que costumam apresentar defeitos.

Final da fila

- Códigos simples:
 - Exemplo: Gets e Sets
- Código gerado por alguma ferramenta.

MÃOS À OBRA

- Instalar o plugin EcEmma no Eclipse
- Rodar os testes existentes e verificar o percentual de cobertura no momento