



Contatos:

Rafael Espinha

rafael.espinha@primeup.com.br

+55 21 9470-9289

www.primeup.com.br

+55 21 2512-6005

DBUnit: Aplicando Ferramentas Open-Source para Testes
de Software

Rafael Espinha

Quem Somos



Tornar nossos clientes mais produtivos no desenvolvimento de software

- ▶ *Quadro de consultores altamente qualificados*
 - ▶ Mais de 15 anos de experiência em mercado.
 - ▶ Mestres, Doutores e Pós-graduados em Engenharia de Software
 - ▶ Forte colaboração com o Laboratório de Engenharia de Software (LES) da PUC-Rio.
- ▶ *Vencedora do Prêmio Solução Rio Info 2007*

Atuação da PrimeUp

- ▶ *Para quem compra ou subcontrata desenvolvimento de software*
 - ▶ Definição de critérios técnicos de subcontratação de desenvolvimento via RFPs
 - ▶ Desenvolvimento de software sob encomenda
 - ▶ Garantia da qualidade do software desenvolvido por terceiros
 - ▶ *Normas, controles, prazos, processos*

- ▶ *Para quem desenvolve software*
 - ▶ Reestruturação da produção de software
 - ▶ *Aumento de produtividade, automação de processos*
 - ▶ Capacitação de equipes

- ▶ *Para quem precisa manter software em funcionamento*
 - ▶ Inovação Tecnológica
 - ▶ Simplificação de Arquitetura
 - ▶ Racionalização de Investimentos



Agenda

- ▶ *Teste de Software*
- ▶ *DBUnit*
- ▶ *Estudo de Caso*
- ▶ *Considerações Finais*

Testes de Software

- ▶ *Atividades para encontrar e expor pontos fracos do sistema*
- ▶ *Envolve ações que vão do levantamento de requisitos até a execução dos testes propriamente ditos*
- ▶ *Validar as funcionalidades do sistema conforme projetadas*
 - ▶ Os requisitos foram implementados de forma adequada?
 - ▶ Validar as suposições feitas nas especificações de projeto e requisito através do código executável
- ▶ *Localizar e documentar defeitos*
- ▶ *Maximizar a quantidade potencial de defeitos que podem ser detectados antes da entrega do produto ao cliente*

Níveis de Teste

- ▶ *Aplicado aos diversos estágios do desenvolvimento*

- ▶ *Diferentes tipos de teste (organizador por “granularidade”)*
 - ▶ **Testes de Unidade**
 - ▶ Testes de Integração
 - ▶ **Teste de Regressão**
 - ▶ Testes de Sistema
 - ▶ Testes de Aceitação
 - ▶ Teste de Carga e Desempenho
 - ▶ Testes de Vulnerabilidade

Técnicas de Testes

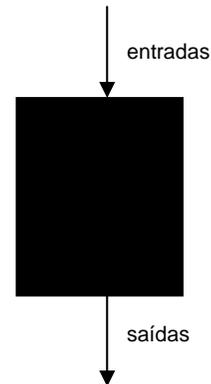
Técnicas de testes

Visão

Fontes de informação

Métodos

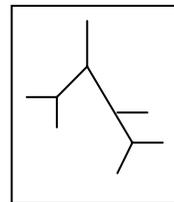
Caixa preta



Domínio do problema
Requisitos
Especificações de projeto
Dados de análise de defeitos

Particionamento de classe
de equivalência
Análise de valores limites
Diagrama de estados
Grafos de causa e efeito

Caixa branca



Diagramas de projeto
Detalhes de projeto
Código fonte
Grafos de fluxo de controle
Complexidade ciclomática

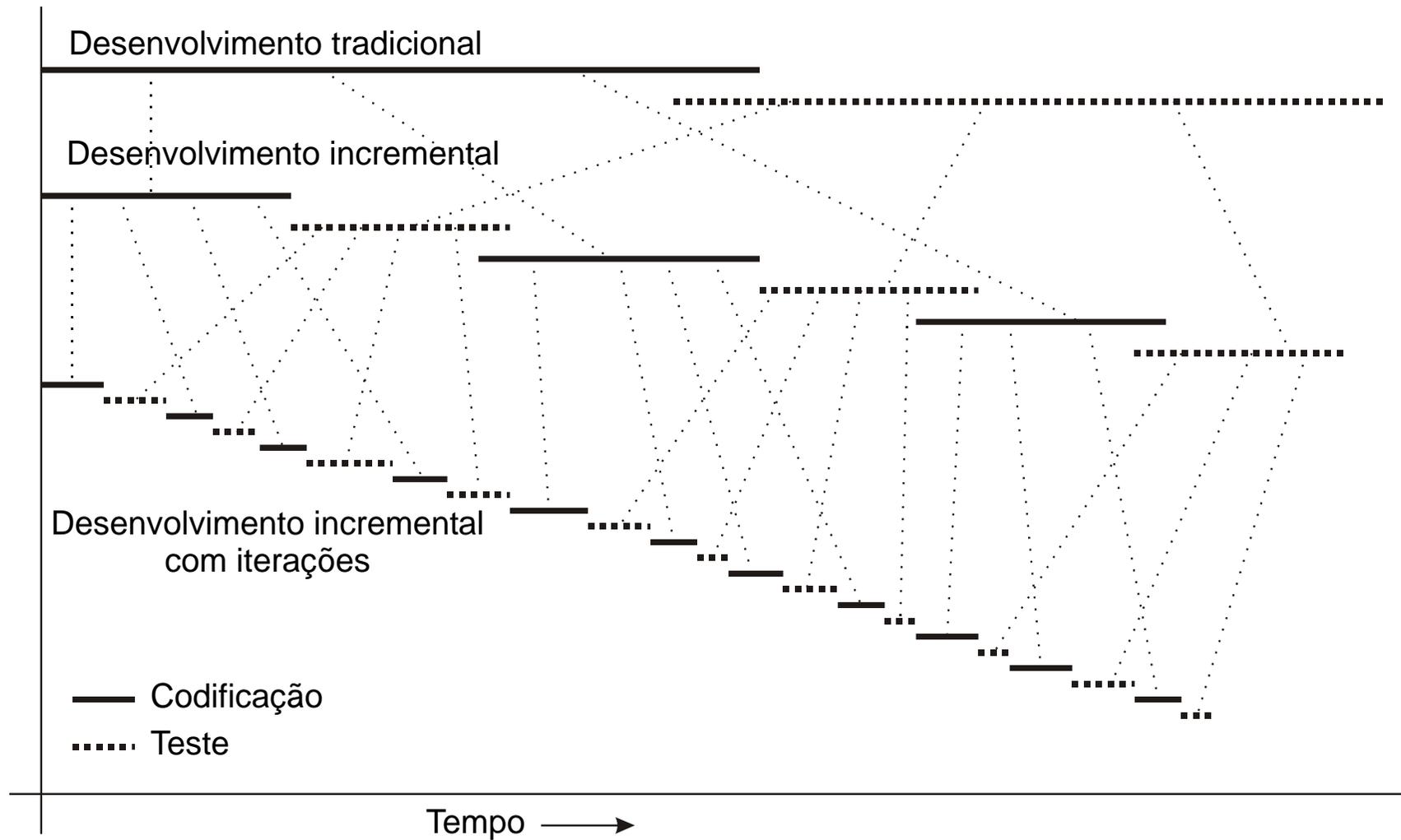
Statement testing
Branch testing
Path testing
Data flow testing
Mutation testing
Loop testing

Testes de Software

- ▶ Testes devem ser realizados no decorrer do ciclo de desenvolvimento
 - ▶ *Não é uma boa prática realizar após o desenvolvimento*
- ▶ Integrar testes no decorrer do desenvolvimento
 - ▶ Na Abordagem XP testes são realizados antes da escrita do código de negócio



Ciclo de Desenvolvimento de Software



Testes, quem precisa?

- ▶ *São conhecidos falhas de software “glamourosas”*
 - ▶ Aeroporto de Denver
 - ▶ Sistema de ambulâncias de Londres
 - ▶ Morte súbita do sistema de telefonia interurbana na região de Washington
 - ▶ Queda de Airbus em voo de teste
 - ▶ Falha de sonda enviada para Marte (Mars orbiter)

- ▶ *ver: ACM Software Engineering Notes para mais exemplos*

Qual o impacto da ausência de testes em Software?

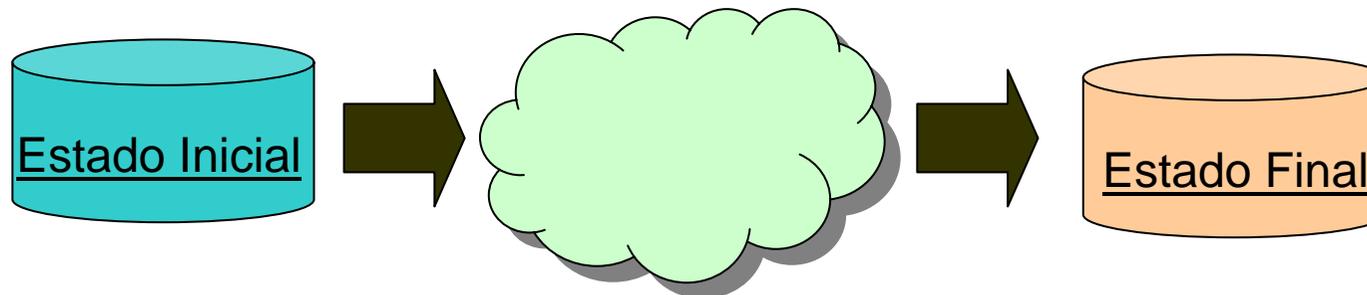
- ▶ *Criticidade de sistemas*
 - ▶ Prejuízo para a imagem da empresa
 - ▶ Prejuízos Sociais
 - ▶ Prejuízos Ambientais
 - ▶ Prejuízos Financeiros

- ▶ *Cerca de 40% a 50% dos programas postos em uso contém faltas não triviais*
 - ▶ Como reduzir este percentual?
 - ▶ Como reduzir os possíveis prejuízos?
 - ▶ Como evitar a propagação de danos?
 - ▶ Como reduzir o tempo médio para corrigir e recuperar?

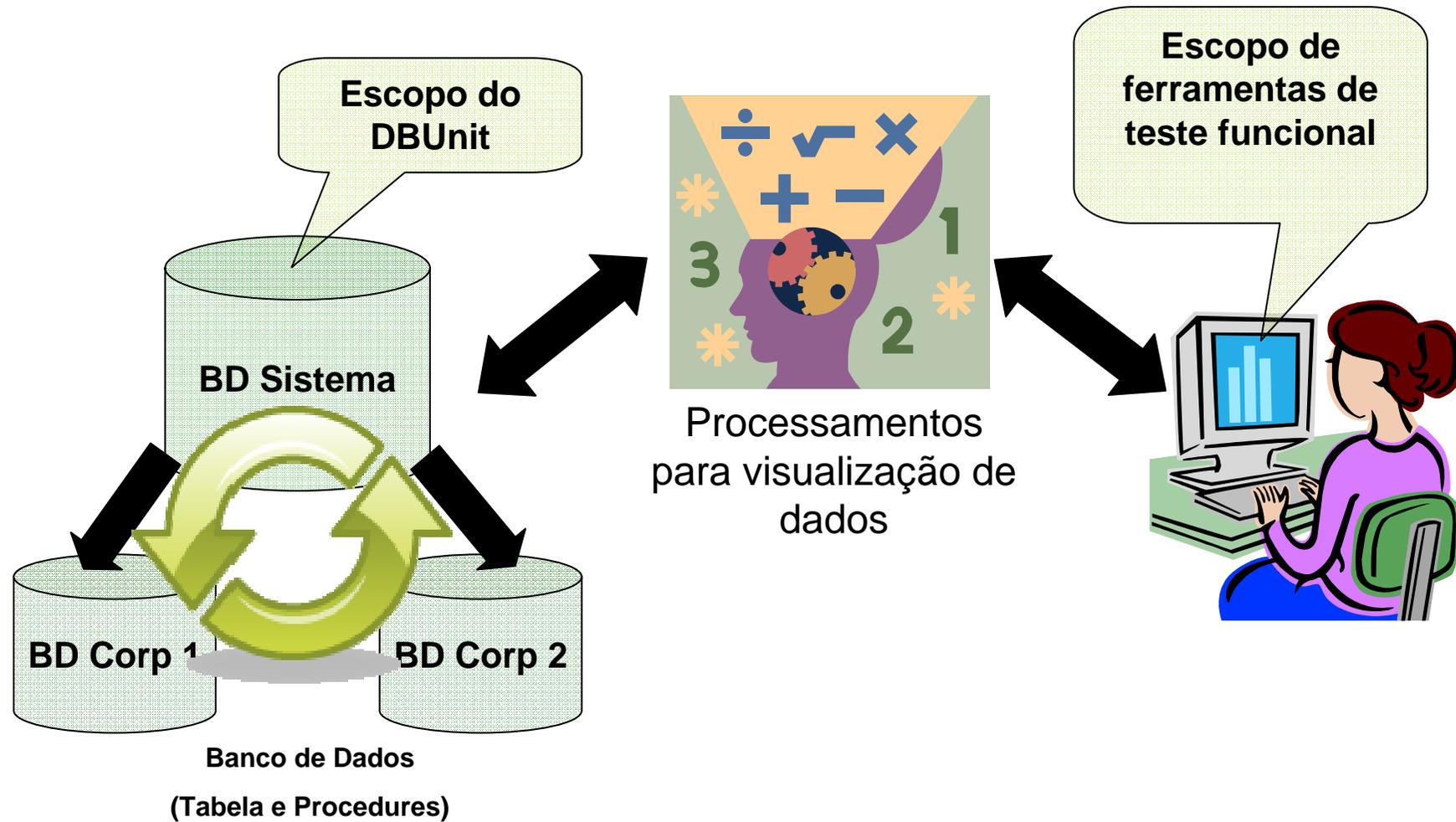
- ▶ *Boehm, B.W.; Basili, V.R.; "Software Defect Reduction Top 10 List"; IEEE Computer*

Sistemas Baseados em BD

- ▶ *Fortemente baseado em regras de negócio*
- ▶ *Utilização abundante de stored procedures, functions, triggers e constraints*
- ▶ *Processamentos ocorrem internamente no BD*



Sistemas Baseados em BD



DBUnit

- ▶ *Extensão do JUnit para testar aplicações orientadas a banco de dados*

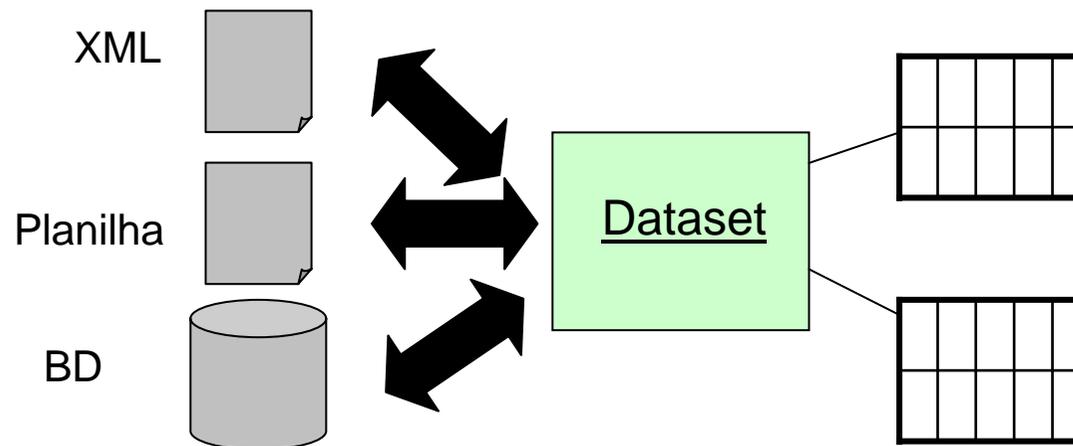
- ▶ *Casos e suites de teste estendem JUnit*
 - ▶ Possibilidade de integração com testes unitários
 - ▶ Possibilidade de integração com outros testes automatizados

- ▶ *Oferece funcionalidades de:*
 - ▶ Leitura e Carga de dados na base
 - ▶ Consulta a tabelas
 - ▶ Comparação (assert) entre tabelas ou entre tabelas e massa de dados (XML ou planilha)
 - ▶ Funcionalidades do JUnit

DBUnit

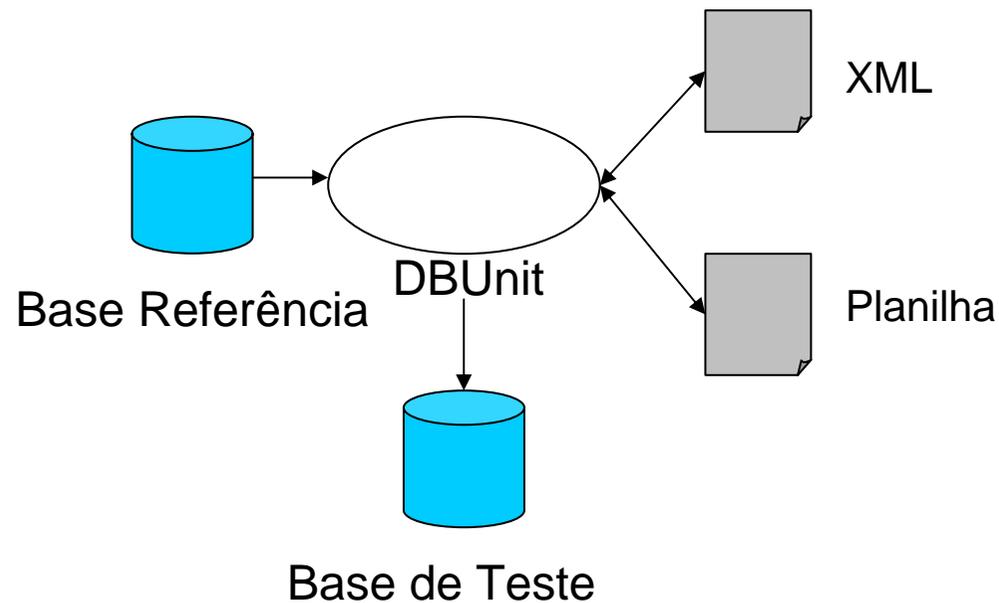
▶ *DataSet*

- ▶ Principal componente do DBUnit
- ▶ Realiza as operações de leitura e escrita de dados
 - ▶ *carga da massa de teste no banco*
 - ▶ *Leitura de resultados esperados*
 - ▶ *Leitura de resultados obtidos*
- ▶ Armazena um conjunto de tabelas
- ▶ Utilizados na comparação de dados (resultado obtido e resultado esperado)



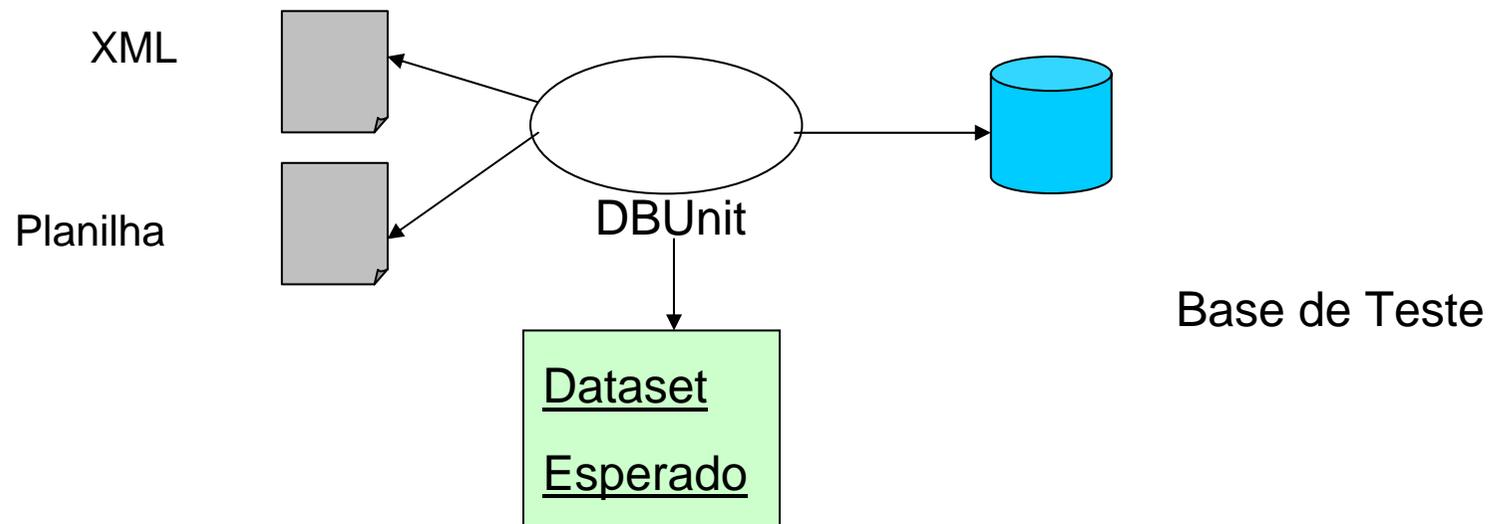
DBUnit em um Projeto

- ▶ *Fase 1: Criação da massa de teste*
 - ▶ Leitura dos dados do BD de referência ou criação direta (automática ou não) dos arquivos
 - ▶ Carregamento de dados na base de teste



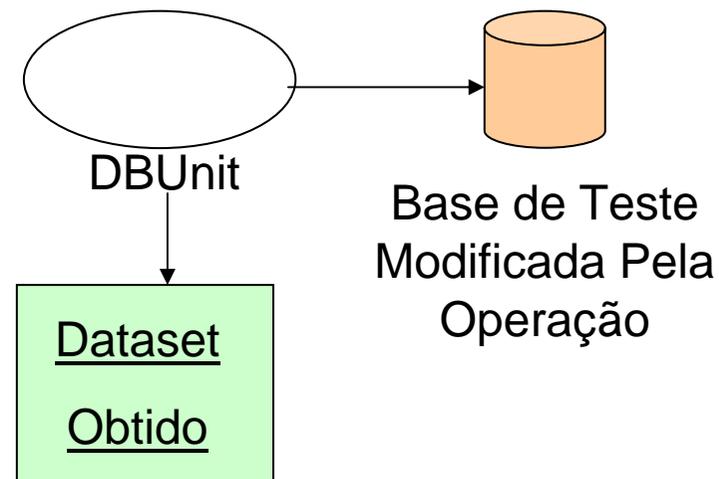
DBUnit em um Projeto

- ▶ *Fase 2: Leitura dos valores esperados*



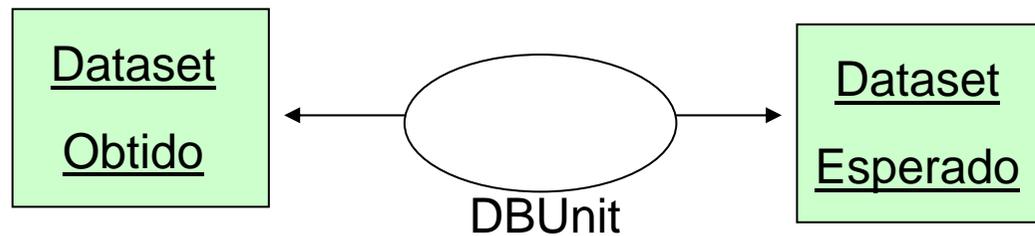
DBUnit em um Projeto

- ▶ *Fase 3: Execução da operação e Leitura dos valores obtidos*



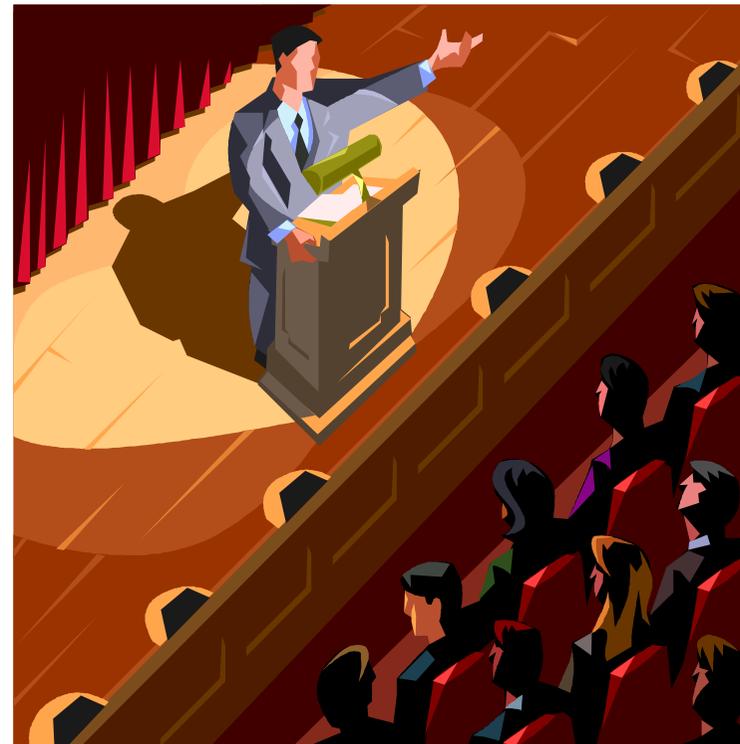
DBUnit em um Projeto

- ▶ *Fase 4: Comparação dos valores*



Apresentação de Resultados

▶ *Demo!!*



Conclusão

- ▶ *A utilização dos testes automatizados com DBUnit facilita a homologação das entregas:*

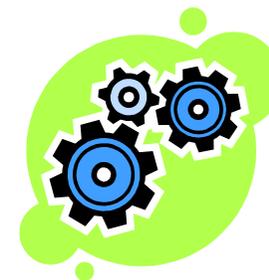
Antes:

- Verificação de dados complexa (vários bancos e tabelas envolvidos)
- Realização de testes manuais e susceptíveis a falha
- Regressão difícil e custosa
- Homologação demorada



Depois:

- Verificação de dados simplificada (automática)
- Execução de testes mais confiáveis e consistentes
- Regressão facilitada
- Homologação ágil





Contatos:

Rafael Espinha

rafael.espinha@primeup.com.br

+55 21 9470-9289

www.primeup.com.br

+55 21 2512-6005

Qualidade : Testes Automatizados com DBUnit

Rafael Espinha