

FERRAMENTAS PARA AUTOMATIZAÇÃO DE TESTES

TIPOS E EXEMPLOS

"A ATIVIDADE DE TESTAR É O PROCESSO DE EXECUTAR UM PROGRAMA COM A INTENÇÃO DE DESCOBRIR UM ERRO. UM BOM CASO DE TESTE É AQUELE QUE TEM UMA ELEVADA PROBABILIDADE DE REVELAR UM ERRO AINDA NÃO DESCOBERTO. UM TESTE BEM-SUCEDIDO É AQUELE QUE REVELA UM ERRO AINDA NÃO DESCOBERTO."


(Glen Myers)



TESTE DE SOFTWARE

Vimos que o teste de software é a investigação do software a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar. Isso inclui o processo de utilizar o produto para encontrar seus defeitos.

IMPACTOS DOS ERROS

- 50% do tempo dos desenvolvedores é gasto para encontrar e corrigir erros;
 - 80% dos custos de desenvolvimento são destinados para identificar e corrigir erros;
 - 56% dos erros encontrados depois do projeto pronto são originados na fase de requisitos;
 - A cada 10 linhas de código escritas, pelo menos 1 erro é gerado;
 - São gastas, em média, 12 horas para corrigir cada erro de um código;
 - São encontrados entre 20 e 30 *bugs* para cada 1000 linhas de código.
- 

QUALIDADE DE SOFTWARE

Conforme a ISO 9126 utilizamos os seguintes atributos para definir a qualidade de software:

Funcionalidade;

Confiabilidade;

Usabilidade;

Eficiência;

Fácil manutenção;

Portabilidade;



TÉCNICAS

Existem muitas maneiras de se testar um software. Mesmo assim, existem as técnicas que sempre foram muito utilizadas

Apesar de os paradigmas de desenvolvimento ser completamente diferentes, o objetivo principal destas técnicas é encontrar falhas no software.



TÉCNICA : CAIXA BRANCA

Conhecido também como teste estrutural ou orientado à lógica

Essa técnica avalia o comportamento interno do componente de software.

Trabalha diretamente sobre o código fonte.



TÉCNICA: CAIXA PRETA

Também chamada de teste funcional, orientado a dado ou orientado a entrada e saída

A técnica de caixa-preta avalia o comportamento externo do componente de software.



TÉCNICA: CAIXA CINZA

A técnica de teste de caixa-cinza é um mesclado do uso das técnicas de caixa-preta e de caixa-branca.

- . A caixa-cinza pode incluir também o uso de engenharia reversa para determinar, por exemplo, os limites superiores e inferiores das classes, além de mensagens de erro.

TÉCNICA: TESTE DE REGRESSÃO

Essa é uma técnica de teste aplicável a uma nova versão de software ou à necessidade de se executar um novo ciclo de teste durante o processo de desenvolvimento

- . Para efeito de aumento de produtividade e de viabilidade dos testes, é recomendada a utilização de ferramentas de automação de teste, de forma que, sobre a nova versão ou ciclo de teste, todos os testes anteriores possam ser executados novamente com maior agilidade.


TIPOS DE TESTES

Uma prática comum é testar o software após uma funcionalidade ser desenvolvida, antes de ser implantada no cliente.

Essa prática pode resultar em compensar atrasos do projeto, comprometendo o tempo devotado ao teste.

Outra prática é começar o teste no mesmo momento que o projeto, num processo contínuo até o fim do projeto.

TIPOS DE TESTES

- Teste de unidade;
 - Teste de integração;
 - Teste de aceitação ;
 - Teste de sistema;
 - Teste de operação;
 - Teste orientados a objetos.
- 

CICLO DE VIDA DOS TESTES

O Ciclo de Vida dos Testes é composto de 5 fases:

- Planejamento: Elaborar a Estratégia de Teste e o Plano de Teste
- Preparação: Preparar o Ambiente de Teste
- Especificação: Elaborar/ Revisar casos de testes e roteiros de testes.
- Execução: Executar e registrar os resultados obtidos
- Entrega: Finalização do projeto e toda documentação é finalizada e arquivada.

CICLO DE VIDA DOS TESTES: PAPEIS E RESPONSÁVEIS

Gerente de Teste	Gerente de vários projetos de teste ou responsável pela área de teste na empresa;
Líder do Projeto de Testes	Técnico responsável pela liderança de um projeto de teste, normalmente relacionado a um projeto de desenvolvimento de sistema.
Arquiteto de Teste	Técnico responsável pela montagem da infraestrutura de teste, configurando o ambiente de teste, escolhendo as ferramentas de teste e preparando a equipe para executar o seu trabalho neste ambiente.
Analista de Teste	Técnico responsável pela modelagem e elaboração dos casos de teste e pelos <i>scripts</i> de teste.
Testador	Técnico responsável pela execução dos casos e <i>scripts</i> de teste.

AUTOMAÇÃO DE TESTES

Devido a complexidade dos softwares atuais, é indispensável a automação dos testes;

A automação de teste proporciona maior qualidade nos teste e no software final, pois traz de benefícios:

- Qualidade,
- Maior desempenho,
- Mais eficiência e eficácia,
- Em menor tempo, economizando custo.

AUTOMAÇÃO DE TESTE: CUSTOS

O custo benefício da automação de teste pode ser verificado sob alguns aspectos:

Ganho na avaliação da qualidade do software, pois permite realizar maior quantidade de testes em menos tempo;

Maior cobertura de testes;

Mais tempo para que a equipe possa realizar outras atividades do processo de teste, como, detalhar e elaborar casos de testes mais complexos e com mais cuidado.



TIPOS DE FERRAMENTAS

Os tipos de automação são normalmente agrupados de acordo com a forma como os testes automatizados interagem com a aplicação. Em geral, os tipos de automação são agrupados em dois paradigmas (mas não são limitados a esses):

- Baseado na interface gráfica,
 - Baseados na lógica de Negócios.
- 

Paradigma	Vantagens	Desvantagens
<p>Baseado na Interface Gráfica</p>	<p>Não requer modificações na aplicação para criar os testes automatizados. Também não é necessário tornar a aplicação mais fácil de testar (testabilidade) porque os testes se baseiam na mesma interface utilizada pelos usuários.</p>	<p>Existe uma forte dependência da estabilidade da interface gráfica. Se a interface gráfica mudar, os testes falham. Baixo desempenho para testes automatizados que exigem centenas de milhares de repetições, testes de funcionalidades que realizam cálculos complexos, integração entre sistemas diferentes e assim por diante.</p>
<p>Baseado na Lógica de Negócio</p>	<p>Foco na camada onde existe maior probabilidade de existir erros. Independência das mudanças da interface gráfica. Alto desempenho para testes automatizados que exigem centenas de milhares de repetições, testes de funcionalidades que realizam cálculos complexos, integração entre sistemas diferentes e assim por diante.</p>	<p>Requer grandes modificações na aplicação para expor as funcionalidades ao mundo exterior. Exige profissionais especializados em programação para criar os testes automatizados. Existem poucas ferramentas/<i>frameworks</i> que suportam essa abordagem (normalmente é necessário criar soluções caseiras).</p>

TIPOS DE AUTOMAÇÃO

Dirigidos a dados (Data-Driven):

- Representam um refinamento dos testes baseados na interface gráfica. Basicamente, nesta abordagem, é utilizado um mecanismo para auxiliar a execução de testes que executam as mesmas ações repetidamente, porém com dados diferentes.

TIPOS DE AUTOMAÇÃO









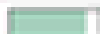







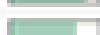







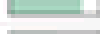







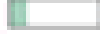





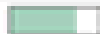

Dirigidos à palavra-chave (Keyword-Driven):

- São realizados por meio da interface gráfica da aplicação. No entanto, os testes são baseados em palavras-chaves (*keywords*). Normalmente a ferramenta de automação oferece um conjunto pré-definido de palavras-chaves para permitir a criação dos testes.

Escolha as colunas a serem exibidas: Exibir/ocultar colunas Tipo de correspondência Ampla 

Palavras-chave

Concorrência do
anunciante Volume aproximado de
pesquisa: junho Volume médio de pesquisa
aproximado Tipo de correspondência Ampla Palavras-chave relacionadas ao(s) termo(s) inserido(s) - classificado por relevância 

Palavras-chave	Concorrência do anunciante 	Volume aproximado de pesquisa: junho 	Volume médio de pesquisa aproximado 	Tipo de correspondência 
camos		2.740.000	3.350.000	Adicionar 
camos usados		201.000	246.000	Adicionar 
venda de camos		49.500	60.500	Adicionar 
camos novos		49.500	74.000	Adicionar 
camos antigos		110.000	110.000	Adicionar 
fotos de camos		110.000	135.000	Adicionar 
camos importados		22.200	22.200	Adicionar 
camos a venda		22.200	33.100	Adicionar 
camos semi novos		18.100	27.100	Adicionar 
aluguel de camos		40.500	49.500	Adicionar 
preço de camos		14.800	18.100	Adicionar 
camos tuning		90.500	74.000	Adicionar 
camos tunados		135.000	135.000	Adicionar 
jogos de camos		246.000	301.000	Adicionar 
camos usados venda		12.100	14.800	Adicionar 
camos baratos		3.600	5.400	Adicionar 
camos rebaixados		49.500	49.500	Adicionar 
comprar camos		12.100	18.100	Adicionar 

TIPOS AUTOMAÇÃO

Baseados na linha de comando (Command LineInterface - CLI):

- Fornece um mecanismo no qual o usuário pode interagir com a aplicação por meio de um *prompt* ou *shell* do sistema operacional

TIPOS DE AUTOMAÇÃO

Baseados em API (Application Programming Interface):

- Representa um conjunto de operações expostas por uma aplicação a fim de permitir que outras aplicações possam acessar ou consumir as suas funcionalidades

Test Harness :

- Baseado na Lógica de Negócio que prega o uso racional e inteligente da automação, mas não pode ter interação com a interface gráfica.

FERRAMENTAS DE TESTE



FERRAMENTAS DE TESTE

No mercado existem vários tipos de ferramentas para testes de softwares, o objetivo principal de todas é o mesmo: **garantir a qualidade do software detectando o maior número de erros possíveis.**

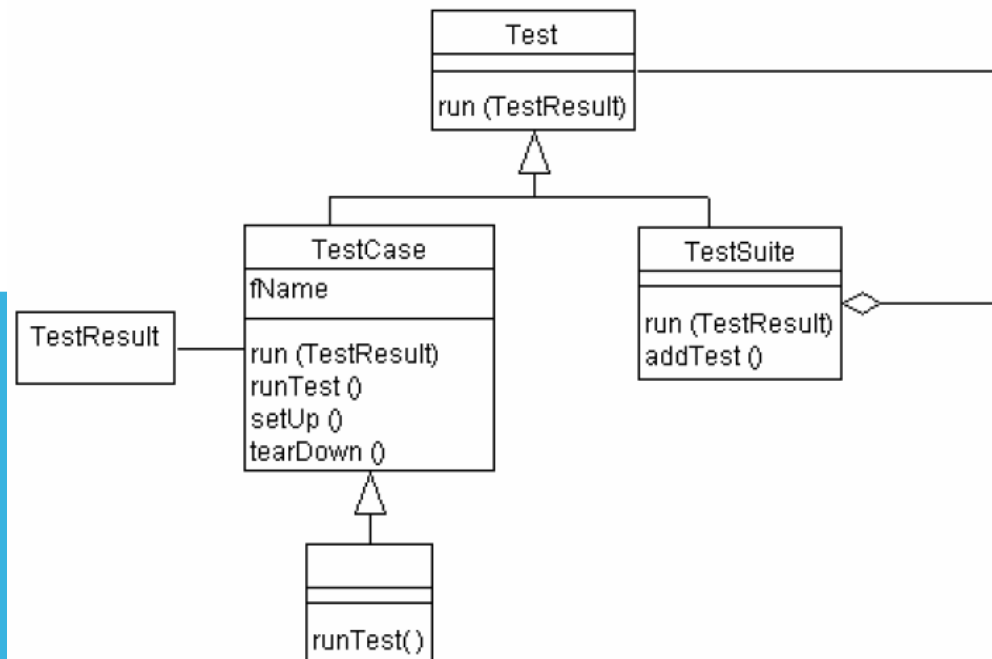
Mas elas possuem algumas características diferentes



FERRAMENTAS ORIENTADAS A OBJETOS - JUNIT

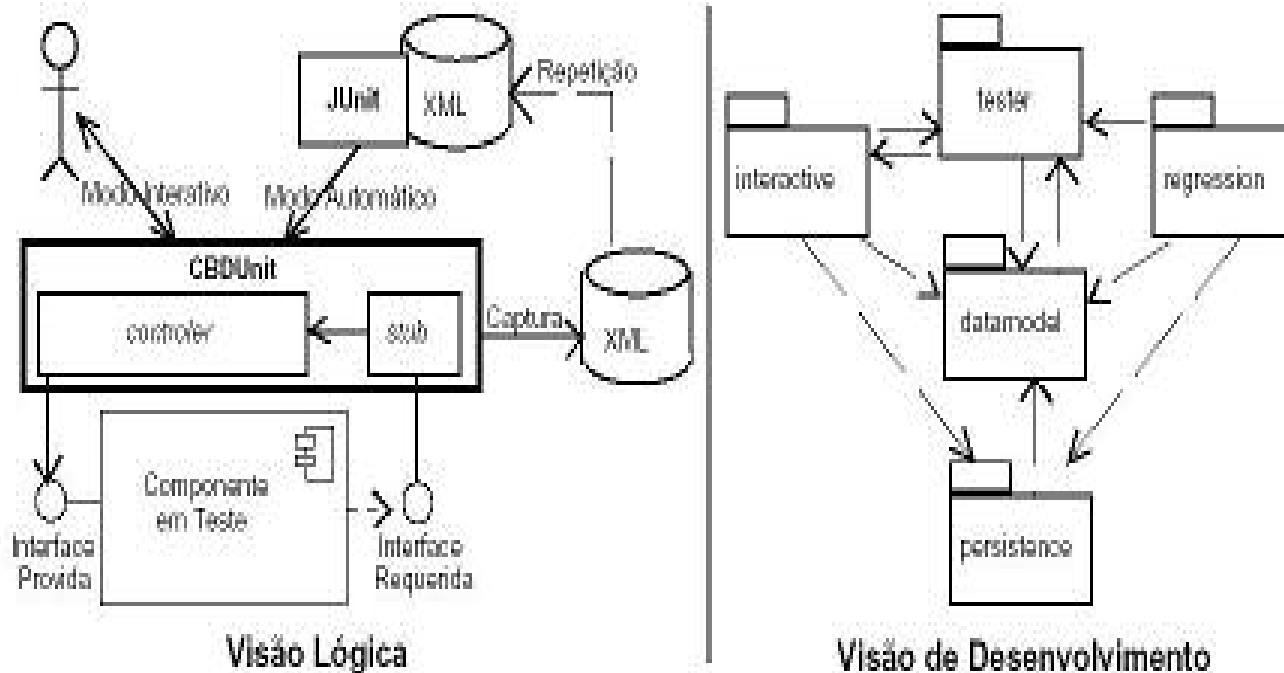
JUnit é um framework que auxilia a criação e execução de testes unitários em classes Java.

É composto por uma API (Application Programming Interface) composta por classes e interfaces, que permitem executar os casos de teste mapeados.



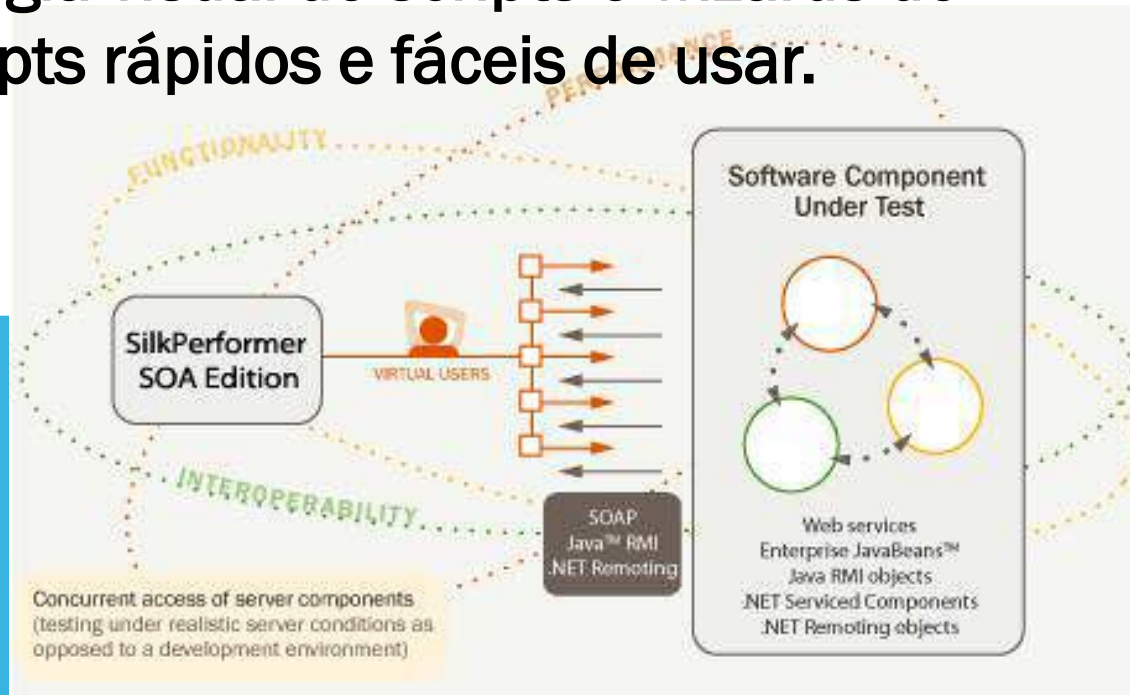
FERRAMENTAS ORIENTADAS A OBJETOS - CBDUNIT

A CBDUnit é uma Ferramenta de teste unitário que oferece facilidades em automatizar tarefas de geração de casos de teste. Essas são obtidas através de modos de operação interativos, onde o usuário especifica os casos de teste.



FERRAMENTAS ORIENTADAS A OBJETOS - SILKPERFORMANCE

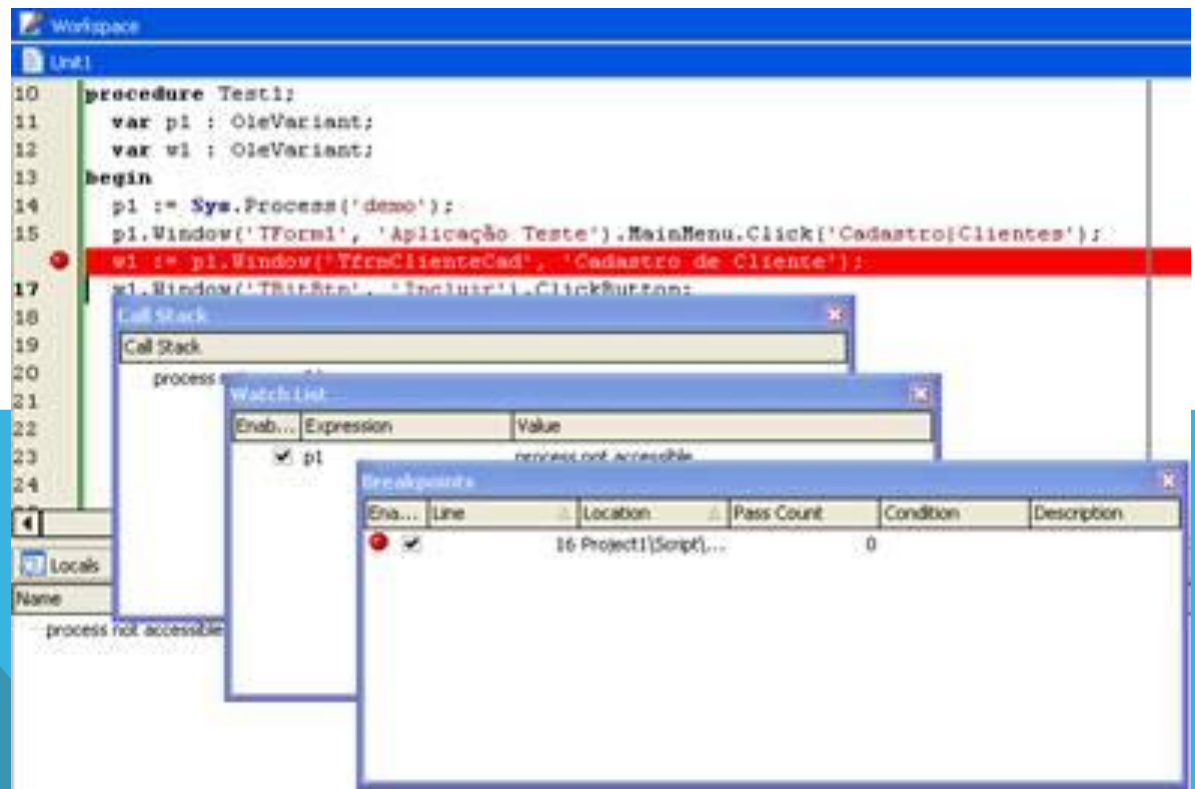
A ferramenta SilkPerformance SOA Edition, da Borland, é uma ferramenta para teste unitário e de componentes de software, tem como objetivo otimizar aspectos como funcionalidade, performance e interoperabilidade. Utiliza servidores, tecnologia visual de scripts e wizards de importação de scripts rápidos e fáceis de usar.



FERRAMENTAS DE TESTE PARA DELPHI: TESTCOMPLETE

As ações dos usuários em um script que pode ser reproduzido (Playback) posteriormente.

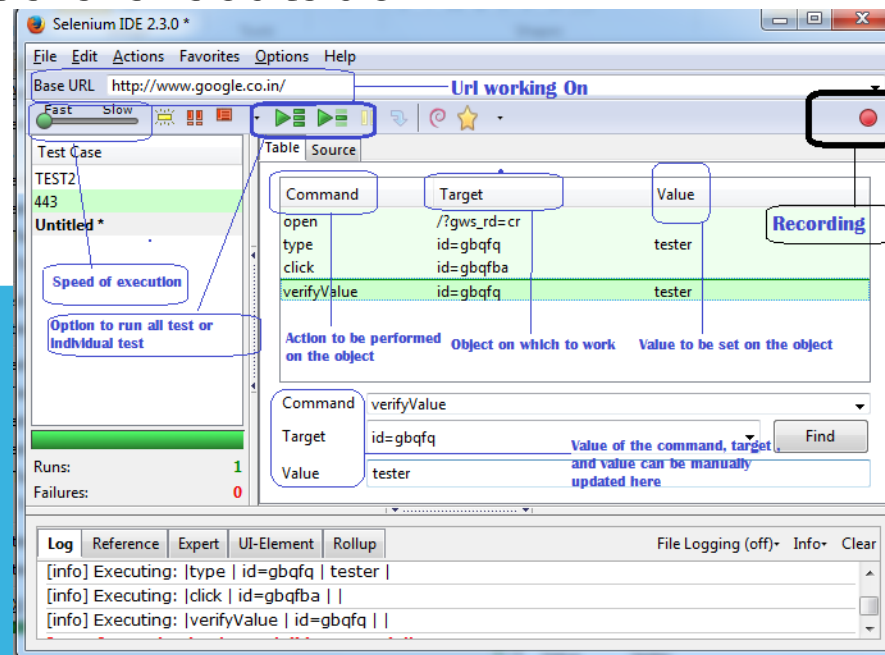
A arquitetura é baseada em plug-ins. Estes plug-ins são chamados de Itens de Projeto.



Ferramentas de teste para WEB: Selenium

Selenium IDE é um ambiente integrado de desenvolvimento para scripts de testes.

É uma extensão do Firefox e permite gravar, editar e depurar os testes. “O Selenium IDE inclui o Selenium Core, permitindo que você facilmente e rapidamente possa gravar e reproduzir os testes no ambiente real que será executado.”



Ferramentas de teste para de desempenho: **JMeter**

JMeter é uma ferramenta utilizada para testes de carga em serviços oferecidos por sistemas computacionais.

JMeter disponibiliza diversos tipos de requisições, além de controladores lógicos como loops(ciclos) e controles condicionais

O JMeter disponibiliza também um controle de threads, no qual é possível configurar a quantidade de vezes que cada thread será executado, o que ajuda a realizar os testes de stress.

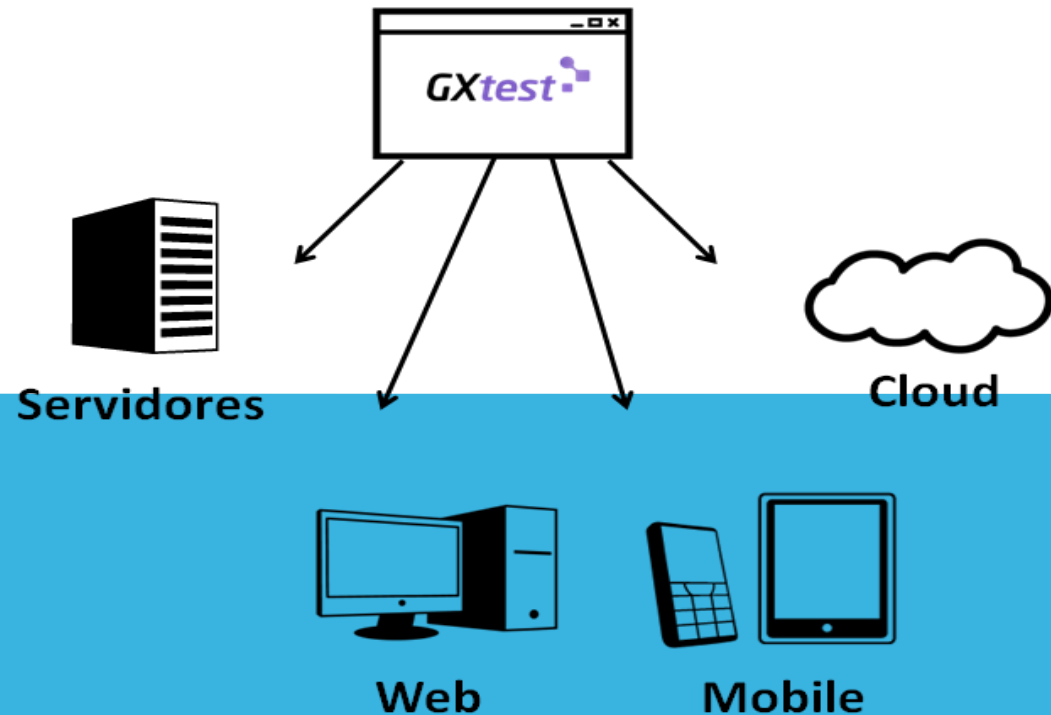
Existem diversos listeners, que se baseando nos resultados das requisições, podem ser usados para gerar gráficos e tabelas.

FERRAMENTA DE TESTE PARA LINGUAGEM GENEXUS

Gxtest é uma ferramenta de teste funcional, criada para testar aplicações realizadas com genexus.

O Gxtest permite quatro tipos de testes:

- Regressão
- Aceitação
- Integração
- Sistema.



COMPARATIVOS ENTRES AS DIVERSAS FERRAMENTAS



Ferramenta	Tipo	Foco	O que ela faz	Linguagem
FATUS	Teste de Unidade	Métodos, atributos e construtores	Gerar planos de teste	Java
JUnit	Teste de Unidade	Métodos das classes	Verifica o funcionamento dos métodos das classes.	Java
CBDUnit	Teste de Unidade	Automatizar geração de casos de teste	Testa componentes do código.	Java, C, C++
Jmeter	Teste Funcional	Desempenho de aplicações WEB	Simula fluxos de aplicações WEB, simulando acessos múltiplos.	
Xpediter		<i>Procedures e Triggers</i> em Banco de Dados	Detecta execuções incorretas.	
NuMega		Componentes das aplicações	Faz diagnóstico de erros e problemas de desempenho.	Visual Basic, C++, Delphi e Java
QACenter	Teste Funcional	Aplicações WEB	Detecta problemas e erros em páginas WEB. Na versão mainframe simula múltiplos usuários.	
Suíte TestStudio		Estrutura de gerenciamento de testes	Configura e compartilha o tempo entre as fases, considerando o ciclo de vida do projeto.	
Robot		Processo de teste	Gerenciamento: cria, modifica e automatiza os processos de teste.	HTML, Java, Visual Basic, C++, Delphi, etc.
VisualTest	Teste Funcional	Sistemas Operacionais	Auxilia os testadores a criar testes para aplicações Windows.	

Fonte: PILLAT (2007), VEROTTI (2004)

Ferramenta	Tipo	Foco	O que ela faz	Linguagem
Logest	Teste estrutural	Estrutura do código	Gera casos de teste, baseado na análise estrutural.	Pascal
JTest	Teste funcional, estrutural e regressão	Classes	Testa uma classe ou conjunto de classes, previne que erros sejam inseridos no código.	Java
WebTestTools	Teste Funcional	Aplicações WEB	Simula carga em servidores web, valida HTML e verifica links.	
SilkPerformance	Teste de unidade e componentes	Servidores .NET, web services e EJBs	Simula acesso simultâneo de usuários.	Java e .NET
Nocaut	Teste de unidade	Aplicações java	Testa códigos dentro ou fora de um servidor WEB.	Java
PureCoverage		Áreas exercitadas ou não do código	Apresenta opções que auxiliam o testador a identificar as áreas do código que foram ou não testadas.	C++ e Java
C++Test	Teste de Unidade, funcional, estrutural e regressão	Classes	Testa classes imediatamente escritas usando <i>drivers</i> e <i>stubs</i> .	C, C++

Fonte: PILLAT (2007), VEROTTI (2004)

TRABALHO

- Dividirem-se em grupos de no máximo 4 pessoas;
- Cada grupo irá escolher uma ferramenta de testes (não podem escolher a mesma. Preferência para quem escolheu primeiro caso haja repetição);
- Realizar uma pesquisa sobre a ferramenta. Apresentar um seminário explicando e mostrando (pode ser com imagens, ou caso de uso) como funciona a ferramenta.
- Abordem questões com: tipo de licença, história da ferramenta, preço (caso seja paga), clientes que usam (caso de uso), linguagens que aborda (c, java, etc), plataformas (windows, linux, etc), versões, instalação, o que precisa para fazer o teste, que tipo de testes podem ser feitos, curiosidades, etc.

TRABALHO

- Apresentação no dia 15/06 .
 - No máximo 20 minutos cada apresentação.
 - Preparar slides para apresentação do seminário;
 - Valor: 10 pontos;
 - Grupos com nomes completos dos alunos e com o software escolhido devem ser enviados por e-mail até as 23:59:59 de 10/06.
 - Podem ser escolhidos outros softwares que não foram apresentados aqui.
- 