

## **SIEP-TEST: UM PROCESSO DE DESENVOLVIMENTO DE TESTE DE SOFTWARE**

**Luiz Carlos CHAVES; Carlos QUIRINO; Ygor CARVALHO; Herbert VASCONCELOS;  
Edemberg SILVA; Heremita LIRA; Thiago MOURA; Crishane FREIRE**

Centro Federal de Educação Tecnológica da Paraíba (CEFET-PB), Av. 1º de Maio - 700, Jaguaribe, João Pessoa, Paraíba, Brasil, e-mail: {lucachaves, diegoquirino, ygor.oliveria.carvalho, herbert.anderson}@gmail.com, {edemberg, heremita, tmoura, crishane}@cefetpb.edu.br

### **RESUMO**

No cenário de desenvolvimento de software, a cada dia, vem se destacando uma variável muito importante no resultado de um projeto, que é a qualidade de software obtida através de técnicas de validação e verificação, como é o caso dos testes de software. É devido a esse fato que muitas instituições tentam esboçar processos para testes de software que colaboram na qualidade dos produtos desenvolvidos. Então, sendo levado por essa necessidade o projeto de pesquisa e desenvolvimento do SIEP (Sistema de Informação da Educação Profissional e Tecnológica) planejou, inicialmente, um levantamento de dados sobre Testes de Software, que fornecesse como resultado final um processo padronizado de teste, intitulado SIEP-Test, que fosse aplicado aos softwares desenvolvidos nos diversos núcleos do SIEP. E, para garantir a consolidação do *SIEP-Test*, o mesmo foi aplicado a um dos softwares desenvolvidos pelo SIEP, e observou-se que o processo serviu como uma ótima ferramenta de agregação de qualidade, já que a rotina de teste evitou principalmente que o software fosse disponibilizado com erros que pudessem ocasionar problemas e desconforto ao usuário final.

**Palavras-chave:** engenharia de software, qualidade de software e teste de software.

### **1 INTRODUÇÃO**

Em qualquer ambiente de programação, a atividade de teste é de extrema importância para gerar produtos de qualidade. Seu princípio básico consiste em fazer uma revisão geral do sistema para assegurar que o software cumpra com suas especificações e atenda às necessidades dos usuários, diminuindo, na medida do possível, a ocorrência de erros no produto entregue ao usuário final. O sucesso dos testes melhora a percepção de qualidade do software, aumentando a satisfação do usuário com o produto e garante maior credibilidade à equipe responsável pelo desenvolvimento.

Para se ter uma idéia sobre a importância dos testes, Bastos et al. (2007) afirma que os defeitos existentes nos softwares, na maior parte das vezes, constituem-se em riscos tanto para o negócio quanto para a imagem da empresa. Diante dessa consideração, podemos afirmar que o sucesso de um software depende bastante do processo de teste ao qual foi submetido, tornando-o parte essencial do processo de desenvolvimento.

Entretanto, na prática, não se consegue testar completamente um programa, nem garantir que ele estará livre de defeitos, pois é quase impossível testar todas as possibilidades e condições criadas pela lógica do programador. Outra problemática que envolve o desenvolvimento de software é que não se pode garantir que todos os programas funcionariam corretamente, sem a presença de erros humanos, visto que os mesmos muitas vezes possuem um grande número de estados com fórmulas, atividades e algoritmos complexos. Principalmente nos casos em que o tamanho do projeto a ser desenvolvido e a quantidade de pessoas envolvidas no processo aumentam ainda mais a complexidade.

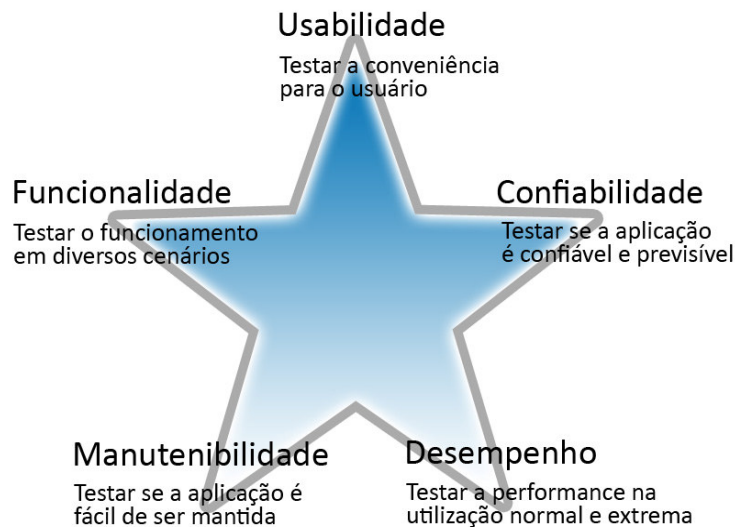
A melhor forma de contornar essas dificuldades seria através de recursos e processos de testes eficientes Delemaro et al. (2007). Porém, bons resultados de teste dependem muito da forma como é aplicado ao software, pois o sucesso de um teste depende diretamente de seu processo de execução. Assim, quando se segue um processo para o desenvolvimento de um produto de software espera-se um produto final que melhor atenda as necessidades tanto aos clientes quanto ao próprio fornecedor do software.

Então, na busca de qualidade dos softwares desenvolvidos pelo SIEP, o núcleo do CEFET-PB elaborou um processo de teste, intitulado SIEP-Test (SIEP-PB, 2008), que serve como guia para orientar a condução dos testes, desde sua concepção até a sua execução, de forma padronizada.

Ciente do prévio conhecimento teórico sobre testes e os seus tipos, o núcleo do CEFET-PB desenvolveu um guia de referência sobre testes de software (SIEP-PB, 2008) que dará subsídios para a construção de alguns dos artefatos gerados pelo SIEP-Test. Nesse guia encontram-se as características dos vários tipos de testes disponíveis, de forma mais simples, avaliando suas capacidades, potencialidades e limitações, buscando mostrar as situações em que se aplicam e que ferramentas poderão ser empregadas.

## 2 QUALIDADE DE SOFTWARE

A busca da qualidade é uma das grandes preocupações da Engenharia de Software, e para isso, se faz necessária a verificação e validação para confirmar se as atividades estão sendo bem feitas e se é aquilo que deveria ser feito (SOMMERVILLE 2007). É através do processo de teste que se pode realizar este procedimento de verificação e validação, buscando saber se o produto está sendo bem construído e se condiz com o que foi planejado inicialmente. Então o teste de software pode ser visto como uma parte do processo de qualidade de software, principalmente quando se tem um processo definido.



**Figura 1 - Atributos qualitativos do software**

A qualidade da aplicação pode, e normalmente, varia significativamente de sistema para sistema, mas os atributos qualitativos previstos na norma ISO 9126 (ISO 2000) (usabilidade, funcionalidade, manutenibilidade, desempenho e confiabilidade) ficam bem subsidiados com o auxílio dos testes, como é ilustrado pela Figura 1. Ou seja, a execução dos testes pode subsidiar no estabelecimento das diretrizes de qualidade previstas, como ocorre com a aplicação de alguns testes:

- **Usabilidade:** os testes de usabilidade buscam garantir um software usável, ou seja, que possa ser facilmente compreendido e operado pelos seus usuários. Softwares com alta qualidade em termos de usabilidade tendem a ter grande sucesso no mercado, pois os usuários buscam sempre aquilo que é mais simples e intuitivo de operar.
- **Confiabilidade:** os testes de confiabilidade buscam garantir um produto de software robusto e estável. Este é o segundo aspecto mais importante do ponto de vista dos usuários finais, visto que não adianta procurar um software que seja fácil de usar, mas que, no momento de sua utilização se comporta de maneira não desejada, com travamentos e erros.
- **Desempenho:** testar o desempenho de um software garantirá que ele funcionará corretamente nos mais diversos cenários, sendo eficiente no consumo de recursos e sendo compatível com outros softwares que possivelmente serão usados juntamente com ele. Usuários finais tendem a não procurar saber sobre o

desempenho de softwares antes de sua aquisição, mas o relato de softwares com baixo desempenho, ou seja, tempo de resposta muito longo poderá afastar os usuários.

- *Manutenibilidade*: o teste de manutenibilidade serve para garantir uma qualidade que é mais do interesse da empresa criadora do software do que de seus usuários finais, e isto é algo fácil de ser compreendido: este teste verifica se o produto terá uma manutenção fácil de ser realizada, garantindo que seu desenvolvimento possa ter continuidade e que ele estará sempre evoluindo.
- *Funcionalidade*: o teste de funcionalidade verifica um dos pontos mais críticos sob o ponto de vista da qualidade do software: verifica se o produto concreto é aquele que havia sido planejado.

Estes cinco aspectos, que juntos formam o conceito de qualidade do software, só podem ser quantificados e constatados por meio de testes, e é neste ponto que, o processo de testes é um dos grandes contribuintes para um software de qualidade.

### 3 TESTES DE SOFTWARE

Segundo Myers et al. (2004) o teste de software deve seguir três fundamentos básicos: todo teste deve apoiar-se num processo de funcionalidade do programa com o intuito de encontrar erros; um bom caso de teste é aquele que tem alta probabilidade de encontrar falhas ainda não encontradas; e os casos de teste bem sucedidos são aqueles que encontram erros desconhecidos.

De uma forma simples, o teste de software significa a verificação controlada do que ocorre no sistema em relação ao especificado. E o objetivo principal desta tarefa é encontrar o número máximo de erros dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos.

Existem várias técnicas de testes que podem ser executadas para se buscar obtenção de melhores resultados num processo de teste. Porém, a utilização de uma técnica depende do tipo de teste a qual ela se destina. Para isso se faz necessária a classificação dos tipos de testes, levando em consideração a sua semântica e sua categorização sobre o conceito teste de caixa branca (com domínio sobre o código a ser testado) ou preta (sem domínio sobre o código a ser testado), a fim de combiná-los e obter melhores resultados. Sabendo disso, o núcleo do CEFET-PB realizou essa classificação a fim de melhor subsidiar os projetistas a elaborarem os testes (SIEP-PB, 2008).

Considerando os aspectos semânticos, temos os tipos de teste:

- *Acessibilidade*: a inclusão de recurso para auxílio a deficiências físicas.
- *Auditoria*: o exame cuidadoso, sistemático e independente com o objetivo de averiguar se a execução, por exemplo, a implementação de um sistema, está de acordo com as disposições planejadas ou estabelecidas.
- *Disponibilidade*: a averiguação sobre a viabilidade de acesso a um sistema.
- *Padronização de Código*: a averiguação sobre o que foi escrito no código fonte de um programa.
- *Compatibilidade*: o exame quanto à adequação do produto atual, quanto a produtos mais antigos e à interação com outros produtos.
- *Conteúdo*: a análise sobre o teor da informação referenciada.
- *Funcionalidade*: o exame sobre o uso prático ou sobre a operação (ou operações) que serão efetivamente realizadas.
- *Navegação*: o exame sobre o planejamento e a execução de passos para se atingir um objetivo ou encontrar a informação em um sistema.
- *Desempenho*: o exame de desenvoltura de um sistema, levando em consideração o seu rendimento quanto à execução de determinada(s) tarefa(s).
- *Confiabilidade*: a probabilidade de um item desempenhar uma função, sob condições específicas, de forma adequada, como previsto no projeto, durante um período pré-determinado.
- *Escalabilidade*: a averiguação de um sistema em termo de sua habilidade de manipular uma porção crescente de trabalho de maneira uniforme, estando preparado para o próprio crescimento.

- *Segurança*: o exame que garanta condição de um sistema estar protegido do perigo, ameaça ou perda.
- *Usabilidade*: o fato de demonstrar que um sistema possua as facilidades com que as pessoas possam empregar no uso de suas ferramentas, realizando tarefas com o menor esforço, ou de maneira mais natural.

A Tabela (1) ilustra a classificação semântica dos testes, agrupados de acordo com os testes de caixa branca e preta.

**Tabela1 - Tipos de teste de software**

Caixa Branca	Caixa Preta			
	Funcionalidade	Confiabilidade	Desempenho	Suportabilidade
Caminho	Navegabilidade	Recuperação de Falhas	Desempenho	Configuração
Unidade	Aceitação	Regressão		Instalação
Regressão	Acessibilidade	Integridade		
Classe	Usabilidade	Segurança		
Padronização de Código	Validação de Entrada			

Da mesma forma que se aplica padronização para a execução, estratégia e processo de teste, também ocorre padronização para a documentação dos testes. E um dos padrões existentes usados para definir documentação de teste é a norma IEEE 829-1998, que descreve como planejar, executar e analisar os documentos de testes de software.

De acordo com a norma para se realizar a documentação deve-se aplicar oito documentos, que cobrem as tarefas de planejamento, especificação e relato de testes. E esses documentos são:

- *Plano de Teste* – Apresenta o planejamento para execução do teste, incluindo a abrangência, abordagem, recursos e cronograma das atividades de teste.
- *Especificação de Projeto de Teste* – Refina a abordagem apresentada no Plano de Teste e identifica as funcionalidades e características a serem testadas pelo projeto e por seus testes associados.
- *Especificação de Caso de Teste* – Define os casos de teste, incluindo dados de entrada, resultados esperados, ações e condições gerais para a execução do teste.
- *Especificação de Procedimento de Teste* – Especifica os passos para executar nos casos de teste.
- *Diário de Teste* - Apresenta registros cronológicos dos detalhes relevantes relacionados com a execução dos testes.
- *Relatório de Incidente de Teste* - Documenta qualquer evento que ocorra durante a atividade de teste e que requeira análise posterior.
- *Relatório Resumo de Teste* – Apresenta de forma resumida os resultados das atividades de teste associadas com uma ou mais especificações de projeto de teste e provê avaliações baseadas nesses resultados
- *Relatório de Encaminhamento de Item de Teste* – Identifica os itens encaminhados para teste no caso de equipes distintas serem responsáveis pelas tarefas de desenvolvimento e de teste.

A norma, diante da sua estrutura, pode ser adaptativa conforme o cenário e propósito dos interessados em aplicar os testes de software. Além disso, ela serve como parâmetro para a elaboração de processos de teste, já que ela garante subsídio para o planejamento, execução e relato de testes. Sua correta utilização auxiliará a

gerência da equipe desenvolvedora a pensar em testes antes de concluir o desenvolvimento do software, evitando a perigosa armadilha de só começar a pensar em testes após a conclusão da fase de codificação.

#### 4 SIEP-TEST

Como uma forma de agrupar e adaptar todos os conceitos e normas de documentação de testes de software em um processo de teste, o núcleo do CEFET-PB elaborou o SIEP-Test. Logo, o processo representa uma proposta de padronização de estruturação das fases e artefatos da aplicação dos testes de software, tomando como base a norma IEEE 829 - 1998, levando em consideração alguns dos seus itens. A seguir serão detalhadas as etapas que compõem o processo de teste representado pela Figura 2.

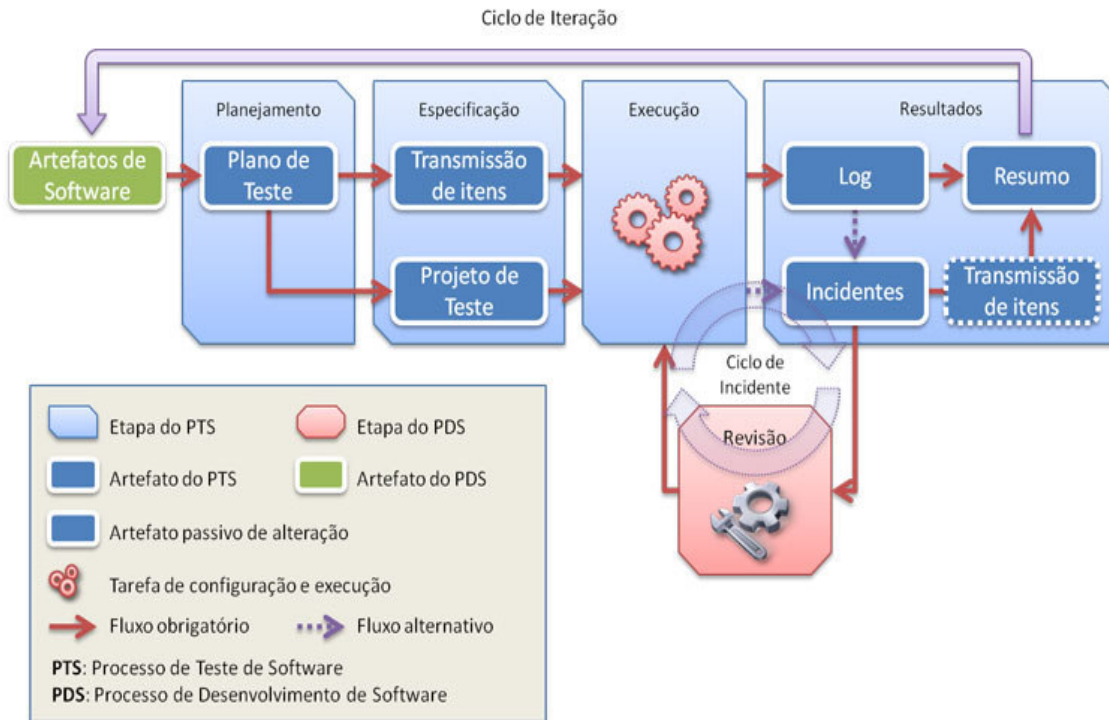


Figura 2: Diagrama do processo de teste do SIEP-Test

##### 4.1 Planejamento

Muitas vezes chamada por fase de concepção, traduz o momento em que a proposta de testes, baseada nos artefatos de entrada e nas expectativas do cliente, é apresentada diante da equipe responsável pelo projeto. Nesse momento, serão estabelecidos prazos, custos e estipulada a qualidade esperada, possibilitando dimensionar a equipe e estabelecer um esforço de acordo com as necessidades apontadas.

Uma das etapas iniciais dessa fase no processo elaborado inicia-se com a obtenção de artefatos gerados em paralelo com o desenvolvimento e nas outras fases do ciclo de desenvolvimento do software. A partir dos artefatos será possível a montagem de toda uma investigação, montagem de estratégias e entendimento do sistema como um todo. Os artefatos que garantem maior subsídio a aplicação estão representado pela Figura 3.

Nesta etapa um único artefato é gerado:

- *O documento de plano de testes*, que deve ser encarado como a base de sustentação para a realização dos testes. Ele descreve o escopo, a abordagem, os recursos, a composição das tarefas e o cronograma relacionado às atividades de teste a serem executadas. Sua montagem é de responsabilidade do Líder do Projeto de Teste.



**Figura 3: Artefatos coletados no fase de planejamento**

## 4.2 Especificação

Trata da identificação dos casos de testes que deverão ser construídos ou modificados em função das mudanças solicitadas pelo cliente, bem como pelo próprio aperfeiçoamento do Processo de Testes (ampliação da cobertura). Definidos os casos de testes, e tendo equipes disponíveis para o processo como um todo, é possível alocar devidamente os recursos para posterior execução dos testes.

Nesta etapa dois artefatos são gerados:

- *Documento de Transmissão de Itens*, que visa identificar os itens de teste que estão sendo repassados aos testadores. Sua criação é de responsabilidade do Líder do Projeto de Teste.
- *Documento de Projeto de Teste*, que deve ser encarado como o documento dos refinamentos sobre o que foi planejado e para identificar as tarefas a serem executadas com maior precisão através da construção dos Casos de Teste. Sua montagem é realizada pelo Analista (ou Projetista) de Testes, responsável pela criação dos Casos de Teste e dos seus *scripts*. Este documento é o único que aglutinou mais de um artefato dos propostos pela norma do IEEE 829-1998, que foram: o de especificação de projeto, de caso de teste e de procedimento de teste. Essa aglutinação se deu ao fato de melhor compor o documento eliminando redundâncias desnecessárias (quando separados) e garantindo a integridade entre eles.

## 4.3 Execução

A execução das atividades de teste não requer documentação, uma vez que se situa na parte prática do SIEP-Test em questão. É necessário, no entanto, pensar sobre: disponibilização do ambiente dos testes e execução dos casos de testes.

## 4.4 Resultados

A partir do que foi executado, é possível extrair os resultados do que aconteceu durante os testes, quais falharam e quais passaram (*pass/fail*), registro dos eventos normais e incidentes. Para compô-los, é

necessário verificar os registros disponíveis e os incidentes acontecidos, a fim de definir: confirmação dos resultados progressivos; confirmação dos resultados regressivos; formalização dos defeitos detectado.

Para fins de avaliação e de homologação dos resultados obtidos, quando todos os itens tiverem sido contemplados e o funcionamento já for algo certo, é necessário produzir um Documento de Resumo de Testes, que encerra o processo. Neste caso, deve-se ter em mente a revisão dos resultados, a avaliação final e a melhoria do processo.

Os artefatos gerados por esta etapa são:

- Documento de Log de Teste, como artefato da etapa de Resultados, dentro do Processo de Desenvolvimento e Execução de Testes da Equipe do SIEP, visa passar uma idéia mais precisa das ocorrências da execução do teste. Sua construção é de responsabilidade dos testadores.
- Documento de Incidentes de Teste, como artefato da fase de Resultados, dentro do Processo de Desenvolvimento e Execução de Testes da Equipe do SIEP, vem destacar a ocorrência de erros e falhas na execução das tarefas de testes quanto ao que foi previsto no Projeto. Sua construção é de responsabilidade dos testadores.
- Documento de Resumo de Teste, como artefato final, dentro do Processo de Desenvolvimento e Execução de Testes da Equipe do SIEP, visa sintetizar os resultados obtidos com a execução dos testes, assim como fazer uma análise final, com vistas nos objetivos gerais da realização dos testes. Em suma, proporcionar a visualização de um resumo dos resultados das atividades de teste que foram designadas, provendo avaliações baseadas nesses resultados. A montagem do documento de Resumo de Teste é de responsabilidade do Líder do Projeto de Teste.
- Documento de Transmissão de Itens, apesar de sua elaboração ser na fase de especificação, esse documento poderá ser alterado nesta fase de resultados caso ocorra incidentes na execução dos testes. Essa alteração visa registrar o item de teste que se encontra em incidência.

Dois são os ciclos definidos pelo SIEP-Test, a fim de minimizar esforços e atender aos requisitos básicos mediante alterações mínimas, com menores custos, sobretudo na alocação de recursos para planejamentos. Deve-se analisar a necessidade de novo ciclo, avaliando-se:

- A necessidade de um novo ciclo de testes diante da quantidade e severidade dos defeitos;
- Os impactos de outros projetos diante da necessidade de um novo ciclo de testes;
- O nível de cobertura dos testes a ser considerado (ciclo completo ou reduzido);
- A possibilidade de aplicar os testes em paralelo ao processo de homologação/implementação;

#### **4.4.1 Ciclo de Incidentes**

Arelado a ocorrência de eventos inesperados, sobretudo quanto as instabilidades externas ao software ou módulos associados, não diretamente ligados ao item do software que está sendo testado, procura corrigir tais problemas pela re-execução dos testes em conformidade com o que já está especificado.

#### **4.4.2 Ciclo de Iteração**

Somente executada após a avaliação final, via Documento de Resumo de Testes, de um processo em sua totalidade, a fim de começar um novo (tendo em vista as demais funcionalidades do software que necessitarão do teste) ou a reconstrução dos testes atuais.

### **4.5 Manutenção**

Pertencente ao Processo de Desenvolvimento, a etapa de manutenção, conhecida como a revisão, serve para dar manutenção aos artefatos que apresentaram problemas durante a execução dos testes, corrigindo os problemas e preparando tais artefatos para que sejam re-testados.

## **5 ESTUDO DE CASO**

Um dos propósitos do SIEP é que todos os seus projetos desenvolvidos devem ser amparados pela aprovação de um processo de teste para que validem todos os artefatos gerados no desenvolvimento. Baseando-se nesse propósito o SIEP-Test foi aplicado ao projeto do Gerenciamento Remoto do Atualizador

(GRA) do SIEP, um sistema que irá servir de acesso as funcionalidades de atualização de dados que serão extraídos das entidades de Ensino Profissionais e Tecnológicos (EPT) para SETEC/MEC.

O SIEP-Test foi utilizado como forma de garantir a liberação do produto, com robustez e segurança dos requisitos desenvolvidos. Porém a aplicação do SIEP-Test no GRA constatou que muitas funcionalidades ainda estavam permitindo brechas e falhas que poderiam comprometer a utilização dos usuários e prejudicar alguns princípios qualidade de software, como funcionalidade e usabilidade.

A aplicação do SIEP-Test, no GRA, começou através da sua fase de planejamento gerando como artefato o documento de plano de teste, a partir dos documentos de visão, requisitos, análise e projeto do sistema. Esse serviu para definir o escopo do quê e como seria testado o sistema. Nessa fase foram focados os casos de uso, especificados no documento de análise, que estavam liberados na primeira versão do GRA.

Após o planejamento dos testes, começou-se a estruturar a fase de especificação, que se iniciou com o documento de projeto de teste. Esse documento detalhou como os casos de uso seriam testados, modelando os passos de execução prática, com detalhamento dos campos de entrada e saída, e os resultados esperados. Por exemplo, para o caso de teste de login, se espera que a entrada seja o nome e senha do usuário resultando no acesso da página inicial da aplicação, e que apenas esses valores validem a autenticação do usuário no sistema. Ainda na fase de especificação, sobre os itens a serem testados, registrou-se no documento de transmissão de itens, a sua localização e quem seria o responsável pelo seu teste.

Com o planejamento e especificação foi possível a execução dos casos de testes, que se basearam em duas ferramentas de teste Web: o JSFUnit (JSFUNIT, 2008) e o Selenium IDE (SELENIUM, 2008). Ambas garantiram que a execução dos casos de teste simulassem a situação real da aplicação, ou seja, os testes foram aplicados em cima das requisições e respostas do container Web da aplicação. A Figura 4 mostra a aplicação dos casos de teste de autenticação, que executou com êxito todos os casos através dos testes unitários da ferramenta JSFUnit.

## Unit Test Results

Designed for use with [Cactus](#).

### Summary

Tests	Failures	Errors	Success rate	Time
8	0	0	100.00%	8.738

Note: *failures* are anticipated and checked for with assertions while *errors* are unanticipated.

### TestCase mec.siep.atualizador.gerenciador.test.jsfunit.TCUC1

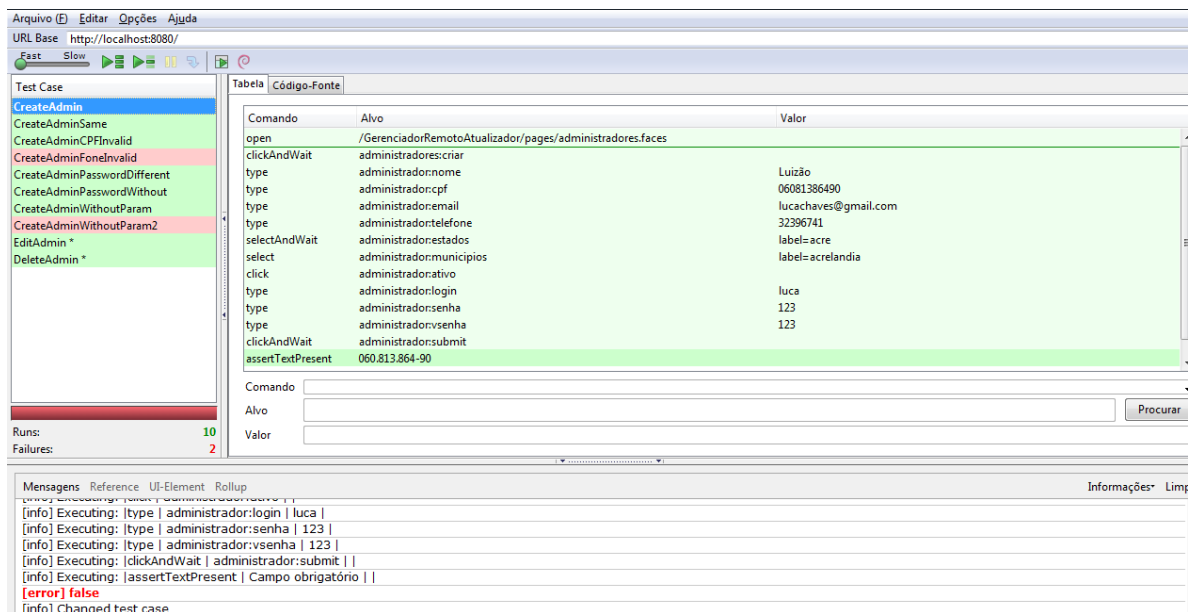
Name	Status	Type	Time(s)
testLoginWithNameInvalid	Success		7.364
testLoginWithPasswordInvalid	Success		0.216
testLoginWithoutPasswordAndName	Success		0.089
testLoginWithoutName	Success		0.108
testLoginWithoutPassword	Success		0.085
testLoginAvaliableActive	Success		0.669
testLoginAvaliableNotActive	Success		0.089
testAccessPageWithoutLogin	Success		0.040

[Back to top](#)

Figura 4: Resultado dos casos de teste de autenticação aplicados no JSFUnit



Já a Figura 5 mostra a aplicação dos casos de teste de gerência de administrador nos *scripts* de execução da ferramenta Selenium IDE. Além dos testes unitários também foram aplicados alguns teste manuais aplicados nos testes de usabilidade.



**Figura 5: Resultado dos casos de teste de administrador aplicados no Selenium IDE**

Depois da execução, a fase de resultados foi alcançada. Nessa fase, a execução dos casos de teste foi registrada, em detalhes, no documento de log, e as falhas dos casos de testes foram descritas no documento de incidentes. No final, o documento de resultado dos testes mostrou todo o percentual de sucessos e falhas do GRA. Esse percentual permitiu mensurar o nível da qualidade do software, almejados em todo o processo, no tocante a funcionalidade e confiabilidade do sistema.

Todos esses artefatos gerados, após a execução dos testes, serviram de subsidio para auxiliar a equipe de desenvolvimento na correção dos erros identificados.

## 6 CONCLUSÃO

Várias ferramentas e técnicas podem auxiliar no desenvolvimento de um produto com boa qualidade. Com a execução de teste, em um processo bem definido, podemos chegar a uma conclusão de que os requisitos de um produto realmente atendem corretamente a sua especificação de forma eficiente e sem erros. Através do SIEP-Test, os produtos obtiveram um melhor êxito na aplicação de técnicas de teste, pois todo o processo serviu de guia de referência para as atividades de teste desenvolvidas pelo SIEP.

O processo de teste também serviu como uma ótima ferramenta para controle e evidência da situação do software. Como foi constado no caso do GRA onde os testes garantiram uma melhor estabilidade e qualidade das funcionalidades testadas, além da identificação de falhas e vulnerabilidades. Considerando também que a confiabilidade do sistema aumentou consideravelmente, pois além do software ter passado pelo crivo da equipe desenvolvimento, também passou pela equipe de teste.

## REFERÊNCIA

BASTOS, A. et al. **Base de conhecimento em teste de software**. São Paulo, Martins, 2007.

CRESPO et al. **Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo**. UNICAMP, Campinas SP, 2005.

DELAMARO, M. E. et al. **Introdução ao Teste de Software**, Rio de Janeiro, Campus, 2007.

JSFUNIT, **Ferramenta de teste unitários para web**, disponível em: <http://labs.jboss.com/jsfunit/>, acessado em agosto de 2008.

IEEE Computer Society; **IEEE Std 829**: Standard for Software Test Documentation; setembro, 1998.

MYERS, G. J.; BADGETT, T.; SANDLER, C.; THOMAS, M. T. **The Art of Software Testing**, ltd, John Wiley & Sons, 2004.

SELENIUM, **Ferramenta de teste unitários para web**, disponível em: <http://selenium-ide.openqa.org/>, acessado em agosto de 2008.

SOMMERVILLE, I. **Engenharia de Software**. 8ª Ed. Tradução: MELNIKOFF, S. S. S.; ARAKAKI, R.; BARBOSA, E. A. São Paulo, Pearson Addison-Wesley, 2007.

PRESSMAN, R. S. **Software engineering**: a practitioner's approach. 4. ed. New York: McGraw-Hill, 1997.

SCALET et al.: **ISO/IEC 9126**, Qualidade de Produtos no contexto de engenharia de software, 2000.

SIEP-PB. **SIEP Teste**, disponível em: <http://siep.cefetpb.edu.br/guiateste/>, acessado em julho de 2008.