

# Definição do Processo de Teste de Software para o Sistema de Informação da Educação Profissional

**Luiz Carlos Rodrigues Chaves**

**Ygor de Oliveira de Carvalho**

**Carlos Diego Quirino Lima**

**Herbert Anderson de Vasconcelos Dantas**

**Edemberg Rocha da Silva**

**Heremita Brasileiro Lira**

**Thiago Moura**

**Crishane Azevedo Freire**

Centro Federal de Educação Tecnológica da Paraíba

lucachaves@gmail.com

**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 SIEP (Sistema de Informação da Educação Tecnológica) definiu, 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 SIEP.

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

**ABSTRACT:** In the scenario of software development, each day, has been highlighting a very important variable in the outcome of a project, which is the quality of software obtained through the validation and verification techniques, such as the testing of software. Due to this fact that many institutions try to outline procedures for testing of software that work in the quality of the products developed. Then, being led by this need the SIEP (Information System for Education Technology) has, initially, a survey of data on tests of Software, and to provide final result as a process of standardized testing, entitled SIEP-Test, which was applied the software developed in the various clusters SIEP.

**Keywords:** software engineering, quality software and test 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, tornado-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 uma metodologia para o desenvolvimento de um produto de software espera-se um produto final que melhor agrade tanto aos clientes quanto ao próprio fornecedor do software.

Então, a partir dos resultados de qualidade e eficiência que a aplicação de teste causa sobre o software, o SIEP sugeriu a criação do SIEP-Test (SIEP-Test 2008) e delegou ao núcleo do CEFET-PB a responsabilidade de elaborar um processo que servisse como guia para orientar na condução dos testes de forma padronizada para ser usado entre os núcleos envolvidos nos módulos do SIEP Gerencial.

No entanto, o processo vai além do seu propósito principal de guiar os módulos do SIEP Gerencial na condução dos testes, pois ele dá alguns subsídios para o entendimento de temas relacionados às atividades de testes, captando e analisando as características dos vários tipos de testes disponíveis, avaliando suas capacidades, potencialidades e limitações, buscando mostrar as situações em que se aplicam e que técnicas, ferramentas poderão ser empregadas e suas principais padronizações.

## 2. Qualidade de Software

A busca da qualidade é uma das grandes preocupações da Engenharia de Software, e para isso, se faz necessário a verificação e validação para confirmar se as atividades que estão sendo feitas 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 o que está sendo construído 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.

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 Fig. (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 e processos executados para se buscar obtenção de melhores resultados num processo de teste. Uma das técnicas seria a classificação de semânticas e taxonomias, que esclarecem de forma mais intuitiva a aplicação de testes específicos de acordo com algumas problemáticas. Essa classificação permite a modelagem dos planos e especificação de teste, pois serve de guia para a escolha dos melhores testes aplicados a critérios e parâmetros específicos.

Considerando essa abordagem a equipe do SIEP-CEFETPB elaborou duas divisões de teste. Uma seguindo o aspecto funcional dos testes, relacionando mais semanticamente o conteúdo e outra avaliando os tipos de teste e relacionando as técnicas e *frameworks* aplicados na validação e verificação do software desenvolvido. Toda essa preocupação tem o intuito de levantar bases para a padronização e aperfeiçoamento de um processo de teste.

A primeira divisão de teste, seguindo o critério do aspecto funcional, foi baseada na semântica. A partir daí listou-se os teste quanto aos aspectos relativos a:

- *Acessibilidade*: a inclusão de recurso para auxílio a deficiências físicas.
- *Auditoria*: o exame cuidadoso, sistemático e independente com o objetivo de ave-



Figura 1. Atributos qualitativos do software.

riguar se a execução (de algo, 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 partir desta semântica dos testes sintetizaram-se técnicas de referência para inserção dos testes no processo de desenvolvimento de software, que seria a segunda divisão elaborada. E para melhor visualizar essa síntese montou-se a Tabela (1) com aplicação dos principais testes de software.

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 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 adaptada conforme o cenário e propósito dos interessados em aplicar os testes de software. Adicionalmente, a equipe responsável pelo teste de-

verá tomar outras decisões em relação à aplicação da norma em projetos específicos, decidindo, por exemplo, se é mais conveniente elaborar um único plano que englobe os testes de unidade, integração e aceitação, ou um plano para cada uma das fases de teste citadas.

Além disso, a norma serve como parâmetro para a elaboração de processos de teste, já que ela garante subsídio para o planejamento, execução e relato de teste. Sua correta utilização auxiliará a gerência a se concentrar tanto nas fases de planejamento e projeto quanto na fase de realização de testes propriamente dita, evitando a perigosa armadilha de só iniciar a pensar no teste de um produto de software após a conclusão da fase de codificação.

#### 4. SIEP-Test

Como uma forma de agrupar e adaptar todos os conceitos e normas de teste de software em um processo de teste, o SIEP-CEFETPB 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 de forma reduzida. A seguir serão detalhados as etapas que compõem o processo de teste representado pela Fig. (2).

##### 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

**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			

todo. São os artefatos que garantem maior subsídio a aplicação esta representado pela Fig. (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 prescreve 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.

#### 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 *scripts* de teste.

#### 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.

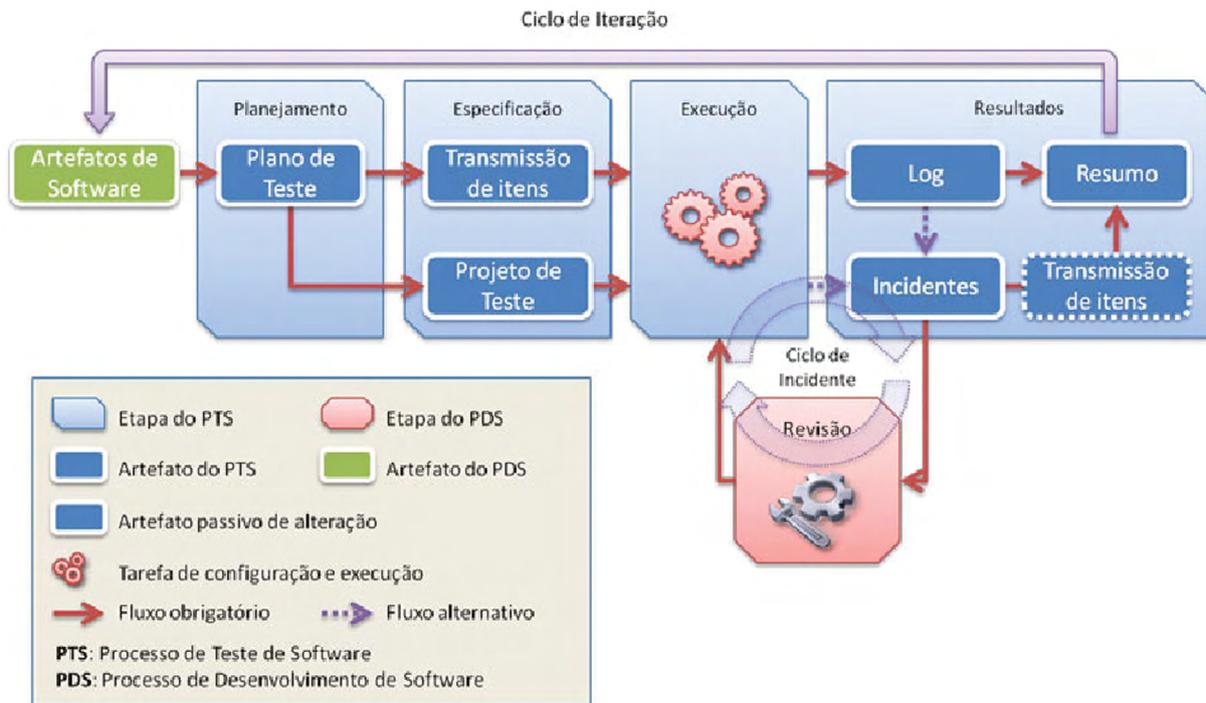
Por fim, para fins de avaliação e de homologação dos resultados obtidos, quando todos os itens tiverem sido contemplados e o funcionamento já for algo certo, é necessário produzir um Documento de Resumo de Testes, que encerra o processo. E aí deve-se ter em mente: Revisão dos Resultados e Avaliação Final e Melhoria do Processo.

Os artefatos gerados por esta etapa são:

- *Documento de Log de Teste*, como artefato da etapa de Resultados, dentro do Pro-

cesso 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



**Figura 2. Etapas do SIEP-Teste**

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 visu-

alização 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.

#### 4.5. Negociação de Ciclos do SIEP-Test



**Figura 3. Diagrama de Artefatos Obtidos na Fase de Planejamento.**

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/implantação;

#### 4.5.1. Ciclo de Incidentes

Atrelado 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.

#### 4.5.2. 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.

#### 4.5.3. 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. Conclusão

Várias ferramentas e técnicas podem auxiliar na montagem de um produto com boa qualidade. Com a execução de teste, num bem processo definido, podemos chegar a uma conclusão de que os requisitos de um produto realmente atendem corretamente a sua especificação de forma eficien-

te 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 todas as atividades de teste desenvolvidas pelo SIEP. O uso do processo aumentou a qualidade dos produtos desenvolvidos no SIEP, pois a finalização dos projetos só seria concluída com o amparo da aplicação do SIEP-Test.

O teste de software também serviu como uma ótima ferramenta para controle e evidência da situação do software, já que influenciou no aumento do controle de falhas de desenvolvimento de software.

## 6. Referência

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

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

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

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

SOMMERVILLE, I. **Engenharia de Software.** 8<sup>a</sup> 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.

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

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

**O processo do SIEP-Test,** disponível em: <http://siep.cefetpb.edu.br/guiateste/>, acessado em 10/07/2008.