

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

**FERRAMENTA WEB PARA CRIAÇÃO DE PLANO DE
TESTES BASEADA NA NORMA IEEE 829-2008**

ANA PAULA JOSLIN DE OLIVEIRA

BLUMENAU
2011

2011/1-01

ANA PAULA JOSLIN DE OLIVEIRA

FERRAMENTA WEB PARA CRIAÇÃO DE PLANO DE

TESTES BASEADA NA NORMA IEEE 829-2008

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Sistemas de Informação— Bacharelado.

Prof. Jacques Robert Heckmann, Mestre - Orientador

**BLUMENAU
2011**

2011/1-01

FERRAMENTA WEB PARA CRIAÇÃO DE PLANO DE TESTES BASEADA NA NORMA IEEE 829-2008

Por

ANA PAULA JOSLIN DE OLIVEIRA

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Jacques Robert Heckmann, Mestre – Orientador, FURB

Membro: _____
Prof. Everaldo Artur Grahl, Mestre – FURB

Membro: _____
Prof. Ricardo Alencar de Azambuja, Mestre – FURB

Blumenau, 30 de junho de 2011.

Dedico este trabalho a todos que acreditaram em sua conclusão.

AGRADECIMENTOS

Gostaria de agradecer em primeiro lugar a Deus, por ter me permitido tanta dádiva em minha vida e fazer com que eu aprendesse sempre com os meus erros e acertos. Por ter me presenteado com tantas coisas boas e não tão boas, por ter me permitido crescer em busca de um ideal sem fim.

À minha família, ao meu pai seu Rogério e a minha mãe dona Vera, por sempre estarem ao meu lado me apoiando e principalmente puxando as minhas orelhas quando preciso. À minha irmã Elizângela por sempre me incentivar a estudar, a crescer e ver que a vida tem que ser levada mais a sério. Ao meu cunhado Chris, por sempre ser positivo, divertido e empolgado com o mundo virtual.

Ao meu namorado Leonardo por ter acreditado em mim, por ter me ajudado e me levantado nos momentos mais tristes, nervosos e difíceis. Por ter sempre uma palavra amiga, uma palavra de conforto, carinho e um belo sorriso.

Aos meus amigos, por entender a minha ausência durante a confecção deste trabalho.

A todos os professores do departamento de sistemas e computação pelo ensinamento passado, disponibilidade em nos atender e às conversas descontraídas nesses anos.

Aos professores Azambuja e Everaldo que acreditaram nesse trabalho mesmo e me deram ideias, principalmente no TCC 1.

Ao professor e orientador Jacques, que acreditou muito nesse trabalho, que sentou ao meu lado para seu desenvolvimento e conclusão. Por ter passado conhecimento, conversado, questionado, argumentado e ter feito com que eu pensasse melhor e melhorasse cada vez mais este trabalho.

A todos, meu obrigada!

“A persistência é o caminho para o êxito.”
Charles Chaplin

RESUMO

Este trabalho apresenta uma ferramenta web para suporte de planos de testes, desenvolvida na tecnologia *Java Server Page* (JSP), utilizando um banco de dados MySQL. Como base para o desenvolvimento utiliza um padrão internacional de documentação, a norma IEEE-829. Desta forma, a ferramenta auxilia a criação e planejamento dos planos e casos de testes.

Palavras-chave: IEEE-829. Plano de testes. Casos de testes.

ABSTRACT

This work presents a web tool for creating tests plans, developed in *Java Server Page* (JSP) technology, using a MySQL database. As a basis for the development uses an international standard of documentation, the IEEE-829 standart. Thus, the tool helps to create and design plans and test cases.

Key-words: IEEE-829. Test Plan. Test Case.

LISTA DE FIGURAS

Figura 1 – Exemplo de teste partindo de requisitos.....	18
Figura 2 – Resumo de documentação de testes	23
Figura 3 – Menu principal do sistema.....	38
Figura 4 – Criação de caso de teste.....	39
Figura 5 – Visualização da especificação dos casos de testes da ferramenta Testlink	40
Figura 6 – Diagrama de caso de uso administrador.....	43
Figura 7 – Diagrama de caso de uso analista de testes	43
Figura 8 – Diagrama de caso de uso testador	44
Figura 9 – Modelo de entidade e relacionamento.....	45
Figura 10 – Login no Sistema	48
Figura 11 - Gerenciamento de usuários.	49
Figura 12 – Criação do usuário com status “Analista”	49
Figura 13 – Tela de boas vindas do usuário Analista	50
Figura 14 – Criação do Plano de teste.....	51
Figura 15 – Criação de MTP	52
Figura 16 – Mensagem da criação do documento	53
Figura 17– Criação de LTP	56
Figura 18 – Arelamento LTP.....	57
Figura 19 – Escolha caso de uso em LTC.....	58
Figura 20 – Inserção de Resultado de Teste.....	59
Figura 21 – Geração relatório de anomalia	60
Figura 22 – Retorno relatório de anomalia.....	60
Figura 23 – Relatório de nível de teste	61
Figura 24 – Relatório de status	62
Figura 25 – Relatório de teste mestre	63

LISTA DE QUADROS

Quadro 1 – Requisitos funcionais	42
Quadro 2 – Requisitos não funcionais	42
Quadro 3 – Criação do relatório de teste mestre	55
Quadro 4 – Quadro comparativo entre os trabalhos correlatos e a ferramenta desenvolvida ..	65
Quadro 5 – Descrição do caso de uso manter usuários e perfis	70
Quadro 6 - Descrição do caso de uso manter plano de testes.....	71
Quadro 7 - Descrição do caso de uso gerenciar caso de testes.....	72
Quadro 8 – Descrição do caso de uso emitir relatório de anomalia	72
Quadro 9 – Tabela Permissão	73
Quadro 10 – Tabela Usuário.....	73
Quadro 11 – Tabela Aprovadores.....	73
Quadro 12 – Tabela Ligacoes.....	73
Quadro 13 – Tabela Documento.....	74
Quadro 14 – Tabela Resultados de teste	75
Quadro 15 – Tabela PlanoTeste.....	76
Quadro 16 – Tabela Caso de Testes X Planos de Teste.....	76
Quadro 17 – Tabela Caso de teste	77
Quadro 18 – Tabela Caso de uso	77

LISTA DE SIGLAS

AR – *Anomaly Report*

CSS – *Cascading Style Sheet*

EA – *Enterprise Architect*

HTTP – *HyperText Transfer Protocol*

IEEE – *The Institute of Electrical and Electronics Engineers*

JSE – *Java Standard Edition*

JSP – *Java Server Page*

LITSR – *Level Interim Test Status Report*

LTC – *Level Test Case*

LTD – *Level Test Design*

LTL – *Level Test Log*

LTP – *Level Test Plan*

LTPr – *Level Test Procedure*

LTR – *Level Test Report*

MER – *Modelo Entidade/Relacionamento*

MTP – *Master Test Plan*

MTR – *Master Test Report*

RF – *Requisito Funcional*

RNF – *Requisito Não Funcional*

SQL – *Structured Query Language*

TCC – *Trabalho de Conclusão de Curso*

UML – *Unified Modeling Language*

SUMÁRIO

1. INTRODUÇÃO	13
1.1. OBJETIVOS DO TRABALHO.....	14
1.2. ESTRUTURA DO TRABALHO.....	14
2. FUNDAMENTAÇÃO TEÓRICA.....	15
2.1. TESTES DE SOFTWARE	15
2.2. PLANO DE TESTES	16
2.3. CASO DE TESTES.....	17
2.4. NORMA IEEE-829.....	18
2.4.1 Versões da Norma.....	18
2.4.1.1 Mudanças entre a versão 1998 e 2008.....	19
2.4.2 Versão IEEE Std 829-2008	20
2.4.3 Documentos da norma	21
2.4.3.1 Plano de Teste Mestre (MTP)	23
2.4.3.2 Nível de Plano de Teste (LTP).....	26
2.4.3.3 Projeto de Nível de Teste (LTD).....	29
2.4.3.4 Nível de Caso de Teste (LTC)	30
2.4.3.5 Nível de Procedimento de Teste (LTPr).....	32
2.4.4 Relatórios	33
2.4.4.1 Log de Nível de Teste (LTL)	33
2.4.4.2 Relatório de Anomalias (AR).....	34
2.4.4.3 Relatório de Status de Nível Provisório (LITSR)	35
2.4.4.4 Relatório de Nível de Teste (LTR).....	36
2.4.4.5 Relatório Mestre de Teste (MTR)	37
2.5. TRABALHOS CORRELATOS	38
2.5.1 Sander.....	38
2.5.2 Labes	39
2.5.3 TestLink	40
3. DESENVOLVIMENTO	41
3.1. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	41
3.2. ESPECIFICAÇÃO	42
3.2.1 Diagramas de Casos de Uso	43

3.2.2 Modelo de entidade e relacionamento	45
3.3. IMPLEMENTAÇÃO	46
3.3.1 Técnicas e ferramentas utilizadas	46
3.3.2 Operacionalidade da implementação	47
3.3.2.1 Acessar a ferramenta	47
3.3.2.2 Tela de Criação de Usuário	49
3.3.2.3 Tela inicial do sistema para o usuário Analista.....	50
3.3.2.4 Planos de Teste.....	51
3.3.2.5 Resultados de Teste	58
3.3.2.6 Relatórios	59
3.3.2.6.1 Relatório de anomalia	59
3.3.2.6.2 Relatório de nível de teste	61
3.3.2.6.3 Relatório de Status	61
3.3.2.6.4 Relatório de Teste Mestre.....	62
3.4. RESULTADOS E DISCUSSÃO	63
4. CONCLUSÕES	66
4.1. EXTENSÕES.....	66
REFERÊNCIAS BIBLIOGRÁFICAS	68
APÊNDICE A – Detalhamento dos casos de uso	70
APÊNDICE B – Dicionário de dados do modelo entidade-relacionamento	73

1. INTRODUÇÃO

Bartié (2002, p. 6) afirma que as empresas estão buscando tecnologia para ampliar sua forma de atuação e reduzir custos, procurando sofisticação de seus sistemas a fim de ganhar eficiência e tomar decisões. Com isso, o desenvolvimento de *software* possui um nível crescente de complexidade, ocorrendo um aumento dos riscos de mau funcionamento proporcional à complexidade desse ambiente. Porém, a maioria das empresas de *software* desconhecem os processos de engenharia de *software*, não havendo garantias de que o programa será entregue com qualidade e dentro do prazo estipulado.

A qualidade de *software* deve ser utilizada nas empresas para garantir que o sistema em desenvolvimento esteja de acordo com os requisitos, assegure o atendimento das necessidades dos usuários e tenha o menor número de falhas para que o risco do negócio seja o mínimo possível. Deste modo, os testes de *software* tornam-se fatores essenciais para a satisfação do cliente e na melhoria contínua do produto desenvolvido.

De acordo com Inthurn (2001, p. 51), teste é uma área da Engenharia de *Software* e seu objetivo é aprimorar a produtividade e evidenciar a confiabilidade do *software*, ao longo do seu desenvolvimento. Para que a qualidade seja feita de maneira organizada, os analistas de testes acompanham o projeto do *software* desde seu início, criando assim, o plano de testes. O plano de testes é uma estrutura que descreve o planejamento (escopo, riscos associados do negócio e abordagem), identifica os objetivos do teste, itens a serem testados, funcionalidades a serem testadas e as tarefas a serem realizadas para a execução do teste. É nele que os casos de testes estão inseridos.

Para Bastos et al. (2007, p.152), o caso de teste registra o planejamento dos testes dos requisitos definidos durante o ciclo do desenvolvimento do software, determinando o que será testado e identifica o maior número de cenários e variações de um requisito. É no caso de testes que se estabelecem as informações que serão utilizadas nos testes dos cenários, quais foram os resultados e a massa crítica de testes necessários para validar os requisitos.

Segundo Koscianski e Soares (2006, p. 360), a norma IEEE 829, atualmente na versão 2008, já na sua versão de 1998 trazia um modelo que define o padrão de documentação onde as informações serão registradas.

Deste modo, o presente trabalho aborda a elaboração de uma ferramenta que auxiliará na criação e gerenciamento dos planos e casos de testes, utilizando como base a norma IEEE 829-2008, facilitando a organização do trabalho do gerente de testes, analistas de testes e testadores.

1.1. OBJETIVOS DO TRABALHO

O objetivo deste trabalho é a construção de uma ferramenta *web* que auxilie a criação de plano de testes em um projeto de desenvolvimento de *software*, de acordo com padrões da norma IEEE 829-2008.

Os objetivos específicos do trabalho são:

- a) disponibilizar o cadastro de plano de testes e especificação dos casos de testes;
- b) gerar relatórios de anomalias, plano de teste mestre, nível de plano de teste a partir dos resultados feitos pelo testador;
- c) definir claramente os papéis do analista de testes e testador na ferramenta.

1.2. ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos.

O primeiro capítulo apresenta os objetivos e a estrutura do trabalho.

O segundo capítulo aborda a fundamentação teórica, constituída do plano de testes, casos de testes, norma IEEE-829 e trabalhos correlatos.

O terceiro capítulo descreve o desenvolvimento da aplicação, com detalhamento dos requisitos trabalhados, especificações, implementação, resultados e discussões.

O quarto capítulo aborda a conclusão do trabalho e suas possíveis extensões.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos os conceitos de testes de software, a norma IEEE Std 829-2008, plano de teste, casos de testes e trabalhos correlatos.

2.1. TESTES DE SOFTWARE

“O objetivo do teste é encontrar defeitos, revelando que o funcionamento do software em uma determinada situação não está de acordo com o esperado” (KONSCIANSKI; SOARES, 2006, p.337). Para Myers (1979, p.5), teste é um processo de execução de um programa a fim de encontrar erros e quando bem-sucedido, identifica erros que podem ser corrigidos durante o processo de desenvolvimento.

Bastos et al. (2007, p. 17) cita que os defeitos existentes nos *softwares*, constituem-se em riscos para o negócio e para a imagem da empresa. O objetivo do teste é minimizar estes riscos causados por defeitos provenientes do processo de desenvolvimento.

Delamaro, Maldonado e Jino (2007, p. 4) afirmam que as atividades de testes são divididas em fases, de acordo com objetivos distintos, estabelecendo os testes de unidade, integração, sistemas e os testes de regressão. Estes não são realizados durante o processo normal de desenvolvimento, mas sim na manutenção do software. As etapas definidas para a execução das atividades de testes são:

- a) planejamento;
- b) projetos de casos de teste;
- c) execução;
- d) análise.

Para Molinari (2008, p. 58), os principais tipos de testes são:

- a) testes funcionais: mostra se a aplicação funciona ou não naquilo que foi proposto em termos de funcionalidade;
- b) testes de performance: demonstra se o desempenho do aplicativo atende as metas desejadas numa carga de desempenho;
- c) testes de usabilidade: testa a facilidade do uso que as pessoas podem empregar na ferramenta, ou objeto para a realização de uma tarefa específica;

- d) teste de unidade: teste a nível de classe ou componente, utilizando uma parte do código, unidade física ou lógica;
- e) teste de cobertura de código: avaliam a qualidade e cobertura do código;
- f) teste de integração: testes de um ou mais componentes combinados visando seu correto funcionamento;
- g) teste de UAT¹: valida a necessidade do usuário, podendo dar o aceite ou não ao sistema;
- h) teste de segurança: testa a segurança da aplicação, de diversas formas;
- i) teste de regressão: ao inserir uma nova característica na aplicação deve-se testar toda a aplicação para garantir que outras partes não foram alteradas;
- j) testes de aplicações *web*: realizados em ambientes e aplicações *web*.

2.2. PLANO DE TESTES

O plano de testes visa uma melhor organização, planejamento e especificação dos testes a serem realizados em um determinado projeto, para as pessoas nele envolvidas. “Uma atividade de testes bem organizada pressupõe planejamento. O plano de testes facilita a comunicação entre os envolvidos no desenvolvimento do *software* ao propor um padrão de referência a ser seguido” (KONSCIANSKI; SOARES, 2006 p. 338).

O plano de teste é o documento mais abrangente do projeto de teste e, como tal, aquele que mostra o projeto em toda a sua dimensão. Caso o *software* venha a passar por alguma manutenção no futuro e precise ser testado outra vez, o plano de teste será o documento que orientará o trabalho dos testadores na preparação de novos testes. Além disso, se o teste precisa ser executado muitas vezes, o plano garante que os elementos críticos serão testados (BARTIÉ, 2002).

De acordo com Bastos et al. (2007), o plano de testes é importante porque descreve como o teste deverá ser executado, traçando uma linha mestra a ser seguida e evitando que alguns problemas sejam tratados apenas no momento em que venham ocorrer.

¹ *User Acceptance Testing*, termo em inglês que significa testes de aceitação pelo usuário.

Para que um teste seja planejado de maneira organizada, é preciso realizar uma estratégia de teste, identificar as pessoas envolvidas, analisar a especificação funcional e definir a contextualização de relatórios. De acordo com Rios e Moreira Filho (2006), o plano de testes está alinhado com a estratégia de testes e a partir dele os casos de testes começam a ser definidos.

2.3. CASO DE TESTES

Segundo Bastos et al. (2007), o caso de teste define-se como a especificação mais detalhada do teste, com minúcias de campos de telas, formulários, entre outros. Estabelece também quais informações serão empregadas durante os testes dos cenários e quais serão os resultados esperados.

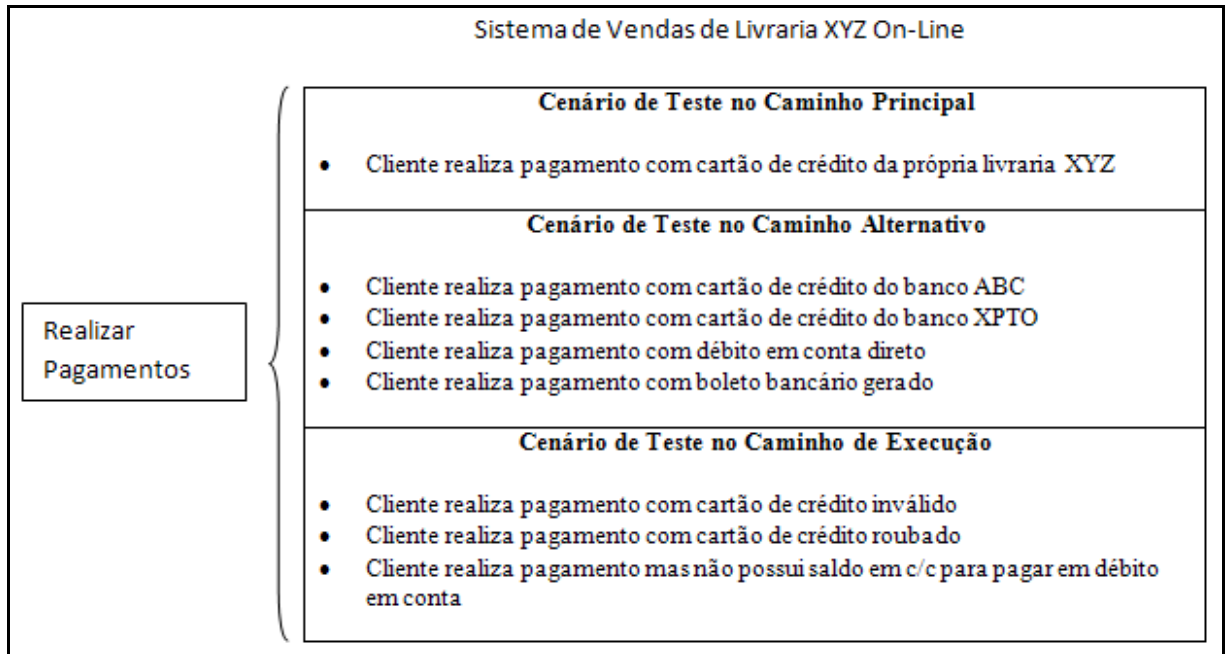
De acordo com Bartié (2002), caso de testes é um documento que registra todo o planejamento dos testes de determinados requisitos que foram estabelecidos durante o desenvolvimento do *software*. Quanto aos casos de testes, os seguintes itens devem ser abordados:

- a) identificação das condições de teste;
- b) identificação dos casos de teste (o que testar);
- c) definição de cada caso de teste identificado;
- d) detalhamento da massa de entrada;
- e) detalhamento da massa resultante;
- f) critérios especiais para geração da massa de testes;
- g) responsáveis pelo levantamento;
- h) agenda de levantamentos (como testar);
- i) cronograma de atividades.

Para que um caso de teste possa ser usado e atenda as expectativas da validação da qualidade, deve ter as seguintes características:

- a) efetivo: testar o que se planejou testar;
- b) econômico: sem passos desnecessários;
- c) reutilizável: que possa ser repetido;
- d) auto-explicativo: que possa ser testado por qualquer testador.

Isto é, segundo Molinari (2008, p.129) os casos de testes são situações de teste que podem abranger um ou mais requisitos. Devem ser objetivos e podem ser aproveitados em mais de um plano de testes. A Figura 1 exemplifica um teste partindo de requisitos.



Fonte: adaptação de Molinari (2008, p. 135).

Figura 1 – Exemplo de teste partindo de requisitos

Garantidas as características citadas, é possível criar um bom caso de teste.

2.4. NORMA IEEE-829

A norma IEEE-829 é um padrão que define os documentos básicos a serem utilizados nos testes de software. Esta norma foi elaborada pelo IEEE e possui três versões. A primeira versão foi lançada em 1983, a segunda em 1998 e a terceira em 2008.

2.4.1 Versões da Norma

De acordo com a IEEE (2010a) a primeira versão da norma, de 1983, define a descrição de um conjunto de documentos básicos de testes associados com os aspectos dinâmicos dos testes de software (execução de procedimentos e código). São definidos o

propósito, definição e conteúdo de cada documento básico. Mesmo que os documentos descritos nesta versão da norma sejam focados em testes dinâmicos, vários destes documentos podem ser aplicados às atividades de outros testes. A norma não exige metodologias de teste específico, abordagens, técnicas, equipamentos ou ferramentas e não especifica a documentação da sua utilização. Também não impõe metodologias específicas para controle de documentação, gerenciamento de configuração ou garantia de qualidade.

A segunda versão da norma, de 1998, define um conjunto de documentos de testes de software básico, especificando a forma e o conteúdo de documentos de teste individual. Esta versão não especifica o conjunto necessário de documentos de teste, de acordo com a IEEE (2010b).

Para a IEEE (2010c), a terceira versão da norma, de 2008, o foco é alterado para os processos de teste, que determinam se o desenvolvimento de produtos de uma determinada tarefa dada está conforme o requisito daquela atividade e se o sistema e/ou software satisfaz o uso pretendido e as necessidades do usuário. As tarefas de processos de testes são especificadas para diferentes níveis de integridade. Estas tarefas de processos determinam a amplitude adequada e a profundidade da documentação de teste. Os elementos de documentação para cada tipo de documentação de teste podem ser selecionados. O escopo do teste inclui os sistemas baseados em software, hardware e suas interfaces. O termo “software” também inclui *firmware*, microcódigo e documentação. Os processos de teste podem incluir inspeção, análise, demonstração, verificação e validação de software e produtos de sistema baseados em software.

2.4.1.1 Mudanças entre a versão 1998 e 2008

Para a norma IEEE 829-1998, o plano de teste determina o escopo, abordagem, recursos e cronograma das atividades de teste. Identifica os itens a serem testados, os recursos a serem testados, as tarefas de teste a ser realizado, o pessoal responsável por cada tarefa e os riscos associados ao plano.

Esta versão descreve as definições básicas para a documentação de testes que estão associados aos aspectos dinâmicos de teste de software (execuções de funções e códigos), definindo o efeito, esboço e conteúdo de cada documento básico.

Enquanto os documentos descrevem o foco da norma na dinâmica de testes, alguns deles podem ser aplicados para outras atividades de teste, como por exemplo: o plano de teste e o relatório de incidente de teste podem ser usados para desenvolver ou verificar códigos.

A versão IEEE 829-2008 traz alterações significativas em relação à IEEE 829-1998. De acordo com a IEEE (2008), a primeira mudança que ocorre na versão de 2008 é o foco do documento, passando a tratar os processos de acordo com a norma IEEE/EIA 12.207,0 TM-1996a, mantendo informações sobre a documentação de testes. Foi adicionado o conceito de nível de integridade, cujo objetivo é auxiliar as organizações na determinação de um conjunto mínimo recomendado de tarefas e seleção simultânea de documentação de teste necessária para apoiar as tarefas. Identifica as tarefas mínimas recomendadas para a amostra de nível de integridade. Foram adicionadas atividades para a escolha apropriada de documentos e conteúdos. Foi criado o plano de teste mestre (MTP) para documentar a gestão efetiva do esforço total de teste. Foi adicionado o relatório de status de nível de integridade de teste a ser divulgado durante a atividade de execução de teste. Foi adicionado o relatório de teste mestre para quando houver vários relatórios de níveis de testes que precisam de consolidação. O relatório mestre de teste mestre também pode resumir os resultados das tarefas identificadas no plano de teste.

2.4.2 Versão IEEE Std 829-2008

A norma de 2008 traz um novo objetivo, enfatizando os processos de testes e, principalmente, introduzindo o conceito de nível de integridade no auxílio da identificação do conjunto mínimo de tarefas de testes recomendadas.

Os conceitos chaves da nova versão, segundo a IEEE 829- 2008 são:

- a) nível de integridade;
- b) recomendação mínima de tarefas de testes para cada nível de integridade;
- c) intensidade e rigor aplicado às tarefas de testes;
- d) critérios detalhados para realização de testes;
- e) sistema de ponto de vista, e;
- f) seleção de documentos de teste.

A integridade é definida em quatro níveis, que descrevem a importância do software e do sistema baseado em software para o usuário.

A recomendação mínima de tarefas de testes para cada nível de integridade define as

tarefas de testes, recomendando o mínimo exigido para cada um dos quatro níveis de integridade. Isso inclui uma tabela de tarefa de testes opcionais para a adaptação do esforço de teste, para atender as necessidades do projeto e as características específicas da aplicação.

A versão também introduz a noção de intensidade e rigor aplicado na realização dos testes, que variam de acordo com o nível de integridade. Os níveis mais elevados de integridade exigem uma aplicação de maior intensidade e rigor nas tarefas de testes, de acordo com a IEEE 829-2008. A intensidade inclui uma abrangência maior de testes em todas as condições de funcionamento normal e anormal do sistema. Já o rigor inclui técnicas mais formais e procedimentos de registro.

Os critérios detalhados para a realização dos testes definem a capacidade específica para cada tarefa do teste, incluindo a recomendação mínima de critérios para a correção, coerência, integridade, precisão, legibilidade e testabilidade. As descrições da tarefa de teste incluem uma lista das entradas e saídas.

O sistema de ponto de vista inclui a recomendação mínima de tarefa de teste para responder às questões do sistema.

A seleção de documentos de teste inclui os tipos de documentação de testes e os tópicos de conteúdos dentro de cada tipo de documentação que precisam ser selecionadas com base nas tarefas de testes associadas ao nível de integridade identificada.

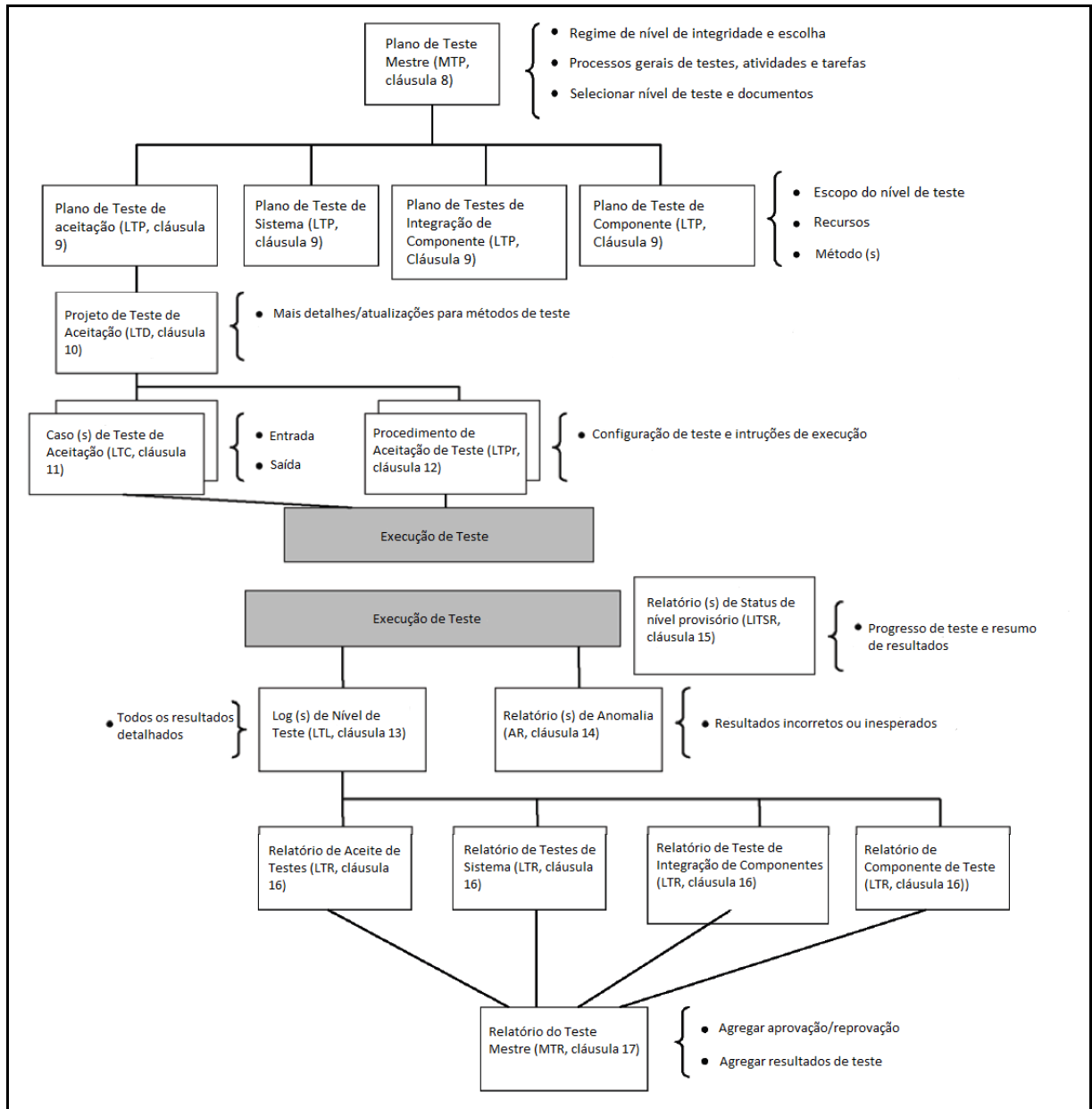
2.4.3 Documentos da norma

A norma IEEE 829-2008 traz os seguintes documentos:

- a) plano de teste mestre (MPT);
- b) nível do plano de teste (LTP):
 - plano de teste de aceitação ;
 - plano de teste de sistema;
 - plano de teste de integração de componente;
 - plano de teste de componente;
- c) projeto de teste de aceitação (LTD);
- d) caso de teste de aceitação (LTC);
- e) procedimento de teste de aceitação (LTPr);
- f) relatório de status de nível provisório (LITSR);
- g) nível do *log* de teste (LTL);

- h) relatório de anomalias (AR);
- i) relatórios de níveis de teste (LTR):
 - relatório de teste de aceite;
 - relatório de teste de sistema;
 - relatório de testes de integração de componentes;
 - relatório de testes de componente;
- j) relatório de plano de teste mestre (MTR).

A Figura 2 ilustra o resumo da documentação de testes.



Fonte: IEEE 829 (2008).

Figura 2 – Resumo de documentação de testes

2.4.3.1 Plano de Teste Mestre (MTP)

De acordo com a norma IEEE 829-2008, o plano de teste mestre é o documento para planejar e gerenciar os testes em vários níveis de teste. O plano de teste mestre descreve cada parte de um projeto de teste. Um plano de testes mestre completo deve conter: introdução, detalhes do plano de teste mestre e gerais. É nesse ponto que o esforço de teste é descrito,

incluindo a organização de teste, cronograma e o esquema de integridade. É necessário um resumo dos recursos necessários e responsabilidades. Ferramentas e técnicas também podem ser incluídas nesta seção. A introdução completa do plano de teste mestre deve seguir a seguinte estrutura:

- a) identificador do documento: identifica de forma única uma versão do documento, incluindo informações como a data de emissão, entidade emissora, autor (a), assinaturas de aprovação e status/versão. Identifica informações que também podem incluir os revisores e gestores pertinentes. Esta informação é normalmente colocada em uma página no início do documento, como a capa ou nas páginas imediatamente após;
- b) escopo: descreve o objetivo, metas e escopo do sistema e/ou esforço de teste de software. Inclui uma descrição de qualquer adaptação deste padrão que foi implementada. Identifica o projeto para o plano que está descrito, os processos específicos e os produtos abrangidos pelo esforço de testes. Descreve as inclusões, exclusões e hipóteses/limitações;
- c) referências: lista todos os documentos de aplicações referentes, sendo externas e internas. As referências externas são impostas por políticas relevantes, leis, normas e regulamentações governamentais. As referências internas são documentos como planos, descrições de tarefas que complementam o plano, tais como autorização de projeto, plano de gerenciamento de projeto e plano de garantia de qualidade;
- d) resumo de sistema e características chaves: nesta seção é descrito o propósito do sistema ou as referências em que as informações são encontradas. Descreve também as principais características do sistema ou do software abaixo do teste;
- e) resumo de teste: descreve a organização de teste, calendário, esquema de nível de integridade responsabilidades, ferramentas, técnicas e métodos necessários para o desempenho de testes. Fazem parte:
 - organização: descreve o relacionamento do processo de teste e outros processos que são desenvolvidos, gerenciamento de projetos, garantia de qualidade e gerenciamento de configuração. Inclui aprovação de produtos de testes e processos;
 - horário de teste mestre: descreve as atividades de teste dentro do ciclo de vida do projeto. Descreve a política de iteração das tarefas para re-execução das tarefas de testes e dependências;

- f) esquema de nível de integridade: descreve a identificação do esquema do nível de integridade para softwares baseados em sistemas ou software. Mapeia o esquema selecionado para o esquema de nível de integridade usado na norma;
- g) resumo de pesquisa: resume as pesquisas de testes, incluindo materiais, facilitadores, ferramentas e requisitos de processos especiais;
- h) responsabilidades: providencia um resumo dos tópicos organizacionais e responsabilidades para tarefas de teste. Identifica componentes organizacionais, seus principais e secundários;
- i) ferramentas, técnicas, métodos e métricas: descreve os documentos, hardware, software, ferramentas de teste, técnicas, métodos e teste de ambiente para usar no processo de testes.

O detalhe dos processos de teste do plano de teste mestre descreve os processos de teste, requisitos dos documentos de teste e requisitos dos relatórios de teste para todo esforço de teste. O detalhamento completo do plano de teste mestre deve seguir a seguinte estrutura:

- a) processos de testes incluindo a definição de níveis de teste: identifica as atividades e tarefas de testes a serem realizados para cada processo de teste e seus documentos de atividades e tarefas de testes. Identifica o número e a sequência dos níveis de testes;
- b) requisitos de documentação de teste: definem o propósito, formato e conteúdo de todos os outros documentos de teste que serão utilizados;
- c) requisitos de administração de testes: descrevem a solução da anomalia e informações de processos, política de interação de tarefas, procedimento de controle de normas e práticas. Estas informações são necessárias para administrar os testes durante a execução;
- d) resolução de anomalia e relatório: descreve o método do relatório e resoluções das anomalias, incluindo os padrões reportados na anomalia, a distribuição da lista do relatório de anomalia, autoridade e equipe para a resolução das anomalias;
- e) política de iteração de teste: descreve os critérios utilizados para determinar a extensão da qual a tarefa de teste é repetida, quando a entrada é alterada ou o procedimento da tarefa é mudado. Estes critérios podem incluir avaliações de mudanças, nível de integridade e efeitos no orçamento, horários e qualidade;
- f) política de desvio: descreve os processos e critérios utilizados do plano de teste mestre e documentação de nível de teste depois que eles são desenvolvidos;

- g) procedimento de controle: identifica os procedimentos de controles aplicados para as atividades de teste. Esses procedimentos descrevem como o software e seus resultados são configurados, protegidos e armazenados;
- h) normas, práticas e convenções: identifica as normas, práticas e convenções de que dirigem o desempenho incluindo as tarefas de testes, mas não limitadas, normas organizacionais internas, práticas e políticas;
- i) requisitos do relatório de testes: especifica o propósito, conteúdo, formato, destinatário e tempo de todos os relatórios de testes. O relatório de teste consiste em *logs* de teste, relatório de anomalias, relatório de status de nível provisório, relatório de nível de teste e relatório de teste mestre.

De acordo com a IEEE 829-2008, a terceira parte do plano de teste mestre é chamada de gerais. Segundo a norma, essa seção inclui o glossário de termos e acrônimos. Também pode ser descrita a frequência e os processos dos quais o plano de teste mestre é mudado. Essa terceira parte do plano de teste mestre completo deve seguir a seguinte estrutura:

- a) glossário: estabelece uma lista de termos em ordem alfabética que pode definir o requerimento dos usuários para o plano de teste mestre com suas definições correspondentes, incluindo acrônimos;
- b) procedimento de mudança de documentos e histórico: especifica o significado de identificação, aprovação e registro de mudanças do plano de teste mestre.

2.4.3.2 Nível de Plano de Teste (LTP)

Segundo a norma IEEE 829-2008, deve-se especificar o escopo, abordagem, recursos e atividades de teste para cada nível de plano de teste. Devem-se identificar os itens e recursos a serem testados, as tarefas de teste a serem realizadas, responsáveis por cada tarefa e seus riscos associados.

Para cada nível de plano de teste pode-se ter diferentes conteúdos, como por exemplo:

- a) cada unidade de software e banco de dados;
- b) unidades integradas e componentes;
- c) testes para cada requisito de software;
- d) teste de qualificação de software para todas as necessidades;
- e) integração de sistemas, podendo ter vários níveis de testes de integração;
- f) teste de qualificação do sistema para seus requisitos.

Outros possíveis níveis de testes incluem instalação, manutenção, regressão e não funcionais como segurança, usabilidade, estresse, recuperação e desempenho. Para qualquer nível pode-se ter mais um sub-nível.

Para um nível de plano de teste ser completo, é dividido em quatro partes: introdução, detalhes do nível do plano de teste, gerenciamento de teste e gerais. De acordo com a IEEE (2008), a introdução completa do nível de plano de teste deve seguir a seguinte estrutura:

- a) identificador do documento: tal como no MTP;
- b) escopo: resume os itens de recursos a serem testados por este determinado nível de teste. Pode-se incluir a necessidade de cada item e seu histórico, podendo ser uma referência a uma determinada parte do plano de teste mestre (MPT), um complemento ou mudanças do MPT;
- c) referências: tal como no MTP;
- d) nível na abrangência de sequência: exibe neste nível a sequência na hierarquia geral de testes.;
- e) classes de testes e condições gerais de testes: resume o caráter único neste nível específico de teste. Este é o detalhe adicional dentro do escopo definido no nível de plano de teste. Alguns exemplos de classes são possíveis dentro de um ou mais níveis de teste.

Na segunda parte do nível de plano de teste, a norma define o seu detalhamento. Esta seção descreve os itens específicos a ser testado no nível designado, e fornece um teste de rastreabilidade matriz, ligando os itens testados com os exigidos, descrevendo a abordagem juntamente com a aprovação/reprovação de critérios e critérios de aprovação/reprovação e resultados de teste. Deve-se seguir a seguinte estrutura:

- a) itens de teste e seus identificadores: identifica os itens de teste que serão analisados, como por exemplo os atributos específicos, instruções de instalação, utilização e procedimentos necessários para transferência de outros ambientes para o ambiente de testes;
- b) matriz de teste e rastreabilidade: fornece uma lista contendo os requisitos exercidos por esse nível de teste e exibe os casos de teste correspondentes;
- c) características a serem testadas: identifica todas as características e combinações de recursos que serão testados;
- d) recursos que não serão testados: identifica todos os recursos conhecidos e as combinações significativas dos recursos que não serão testados e as razões para exclusão;

- e) abordagem: descreve a abordagem geral para o teste nível, podendo ser detalhada para permitir a identificação das principais tarefas de teste e estimativa de tempo necessário para realizar cada uma. Inclui métodos de testes, como caixa preta, caixa branca, análise e inspeção. Esses métodos são uma combinação dos recursos a serem testados e dos recursos a não serem testados em uma tabela chamada matriz de teste;
- f) número de aprovação/reprovação de critérios: especifica os critérios a serem utilizados na determinação de cada item de teste que passou ou falhou nos testes;
- g) critérios de suspensão e retomada: especifica os critérios utilizados para suspender totalmente ou parcialmente uma determinada atividade de testes sobre os itens de teste associados ao plano. Especifica as atividades de teste a serem repetidas quando o teste é retomado;
- h) testes entregáveis: identifica todas as informações que será emitida pela atividade de teste, podendo incluir os documentos listados anteriormente. Descreve as entregas das informações para as pessoas e entidades organizacionais, podendo ser uma referência ao plano de gerenciamento de configuração.

A terceira parte do nível de plano de teste, de acordo com a IEEE 829-2008, contempla o gerenciamento de testes. Esta seção descreve as atividades de teste e tarefas para o nível especificado e suas progressões. São identificadas informações como infra-estrutura, responsabilidades e autoridade, cronogramas e riscos, quando não estiverem descritos em um documento de nível mais alto, como o plano de teste mestre. Para ser completo, deve seguir a seguinte estrutura:

- a) atividades e tarefas: identifica o conjunto de tarefas necessárias para preparar e executar os testes, as dependências de tarefas e outras restrições significativas;
- b) ambiente e infra-estrutura: especifica as propriedades desejadas e necessárias do ambiente de teste e outros dados relevantes. Pode-se incluir características físicas das instalações como hardware, software ferramentas de teste de apoio, banco de dados e pessoal. Inclui ambiente para instalação antes do teste, execução durante o teste e atividades pós-teste. Especifica o nível de segurança que devem ser previstas e questões relacionadas;
- c) responsabilidades e autoridades: identifica as pessoas ou grupos responsáveis pela gestão, concepção, preparação e execução, verificando resultados deste nível de teste para resolver as anomalias encontradas;
- d) interfaces entre as partes envolvidas: descreve os meios e conteúdos de

- comunicação entre as pessoas e/ou grupos envolvidos no nível de plano de teste;
- e) recursos e suas distribuições: descreve os recursos adicionais necessários que não estão documentados no plano, incluindo recursos internos e externos;
 - f) formação: especifica as necessidades de formação por nível de habilidade, identificando opções de treinamento;
 - g) horários, estimativas e custos: inclui os marcos de teste identificados no cronograma do projeto e seus itens de transmissão de eventos. Estima o tempo necessário para a realização de cada tarefa de teste, especifica a programação de testes para cada tarefa e para cada recurso de testes e especifica o seu período de utilização;
 - h) riscos e contingência: identifica situações de risco que podem afetar negativamente a conclusão das atividades planejadas. Especifica o impacto de cada risco com um plano de contingência para evitar e/ou diminuir o risco.

A quarta parte do nível de plano de teste descreve os procedimentos da garantia de qualidade e métricas. De acordo com a IEEE 829-2008, a parte geral do nível do plano de teste deve seguir a estrutura:

- a) procedimentos de garantia da qualidade: identifica os meios pelos quais a qualidade dos processos de análise e dos produtos serão assegurados. Inclui o acompanhamento e resolução dos processos;
- b) métricas: identifica as medidas específicas que serão coletadas, analisadas e relatadas;
- c) cobertura de teste: especifica os requisitos para a cobertura do teste;

A última parte do plano de teste nível completo deve seguir a estrutura:

- a) glossário: tal como no MTP;
- b) procedimento de mudança de documentos e histórico: tal como no MTP.

2.4.3.3 Projeto de Nível de Teste (LTD)

O projeto de nível de teste especifica qualquer refinamento da abordagem de teste e identifica os recursos a serem testados por este projeto e seus testes associados. Segundo a IEEE 829-2008, o projeto de nível de teste é dividido em três partes e a primeira deve seguir a seguinte estrutura:

- a) identificador do documento: tal como no MTP;

- b) escopo: resume os itens de recursos a serem testados por este determinado nível de teste. Pode-se incluir a necessidade de cada item e seu histórico, podendo ser uma referência a uma determinada parte do plano de teste mestre (MPT), um complemento ou mudanças do MPT;
- c) referências: tal como no MTP;

A segunda parte do projeto de nível de teste descreve os recursos a serem testados, identifica os conjuntos de casos de teste ou cenários com a aprovação/reprovação dos critérios e deve seguir a seguinte estrutura:

- a) características a serem testadas: tal como no MTP;
- b) refinamento de abordagem: especifica refinamentos aos métodos descritos no nível de plano de testes, incluindo técnicas específicas de teste, resume os atributos comuns aos casos de teste associados, necessidades de ambiente;
- c) identificação de teste: lista o identificador e uma breve descrição de cada caso de teste associado ao projeto;
- d) características de aprovação/reprovação de critérios: especifica os critérios a serem utilizados para indicar se a combinação de recurso ou o recurso foi aprovado ou não;
- e) resultados de teste: identifica as informações que foram emitidas pela atividade de teste.

A terceira parte do nível do projeto de teste descreve os documentos gerais. De acordo com a norma, a parte geral do nível de plano de teste deve seguir a mesma estrutura do MTP.

2.4.3.4 Nível de Caso de Teste (LTC)

De acordo com a IEEE 829-2008, o nível de caso de teste define as informações necessárias às entradas e saídas a partir do sistema/software que será testado. O caso de teste nível pode ser referenciado em diversos modelos de nível de teste utilizados em vários grupos. Um caso de teste completo é dividido em três partes. A introdução identifica a entidade emissora e detalhes da emissão, inclui as aprovações necessárias e status do documento. Para a primeira parte ser completa, deve ser seguir a seguinte estrutura:

- a) identificador do documento: tal como no MTP;
- b) escopo: tal como no LTD;
- c) referências: tal como no LTD;

- d) contexto: fornece o contexto necessário que não esteja descrito em outras seções do documento;
- e) notação para descrição: define os sistemas de numeração.

A segunda parte do nível de caso de teste, conforme a IEEE 829-2008, descreve cada caso de teste, contendo o seu identificador, objetivo, resultados, necessidades de ambiente e os procedimentos especiais. Para o detalhamento ser completo, deve seguir a seguinte estrutura:

- a) identificador de processo de teste: descreve o identificador único para cada caso de teste;
- b) objetivo: identifica e descreve o foco/objetivo para o caso de teste. Pode ser incluída a prioridade ou risco;
- c) entradas: especifica cada entrada de dados para a execução de cada caso de teste, especificando todas as relações necessárias entre as entradas;
- d) resultados: especifica as saídas e o comportamento esperado, fornecendo o valor exato para cada saída;
- e) necessidades de ambiente: descreve o ambiente de teste necessário para a configuração e execução de teste, como os resultados de gravação;
 - 1. hardware: especifica as características e configurações do hardware necessário para a execução do caso de teste;
 - 2. software: especifica a configuração dos softwares envolvidos na execução do caso de teste, podendo ser sistemas operacionais, simuladores e ferramentas de teste;
 - 3. outros: especifica outras exigências não incluídas, se houver;
- f) requisito processual especial: descreve as restrições do procedimento de teste que for executar no caso de teste. Pode-se referenciar a utilização de ferramentas de teste automatizado;
- g) dependências: lista os identificadores dos casos de teste a serem executados antes deste caso de teste.

A terceira parte do nível do caso de teste descreve os documentos gerais. De acordo com a norma, a parte geral do nível de plano de teste deve seguir a mesma estrutura do MTP.

2.4.3.5 Nível de Procedimento de Teste (LTPr)

O objetivo do nível de procedimento de teste, segundo a IEEE 829-2008, é a especificação dos passos a serem executados no conjunto de casos de teste, para avaliar as características de um conjunto de teste. O nível de procedimento de teste é dividido em três partes. A primeira parte é a introdução, onde é identificada a organização e os detalhes de emissão. Para que a introdução seja completa, deve seguir a seguinte estrutura:

- a) identificador do documento: tal como no MTP;
- b) escopo: descreve o escopo do procedimento em termos particulares e seu apoio à execução dos níveis de casos de testes ou níveis de plano de teste;
- c) referências: tal como no MTP;
- d) relação com outros procedimentos: descreve as exigências deste procedimento, podendo estender a outros procedimentos.

A segunda parte inclui as entradas e saídas e descrição ordenada dos passos de teste necessários para a execução de cada caso de teste. Segundo a IEEE 829-2008, para o detalhamento ser completo, deve seguir a seguinte estrutura:

- a) entradas, saídas e necessidades especiais: identifica todas as necessidades para a execução dos testes, incluindo casos de teste, bases de dados, ferramentas automatizadas, sistemas de terceiros, procedimentos de pré-requisitos, exigências de ambiente e habilidade especial;
- b) descrição ordenada dos passos a serem dados por cada participante: inclui as atividades (se aplicável) para cada processo de teste, incluindo o grau das etapas do procedimento de teste: *log*, instalação, início, continuação, medição, desligar, reiniciar, parar, *wrap-up* e contingências.

A terceira parte do nível do procedimento de teste descreve os documentos gerais. De acordo com a norma, a parte geral do nível de plano de teste deve seguir a estrutura do MTP.

2.4.4 Relatórios

Além da estrutura do plano de testes, a norma apresenta relatórios a partir dos resultados alcançados.

2.4.4.1 Log de Nível de Teste (LTL)

O objetivo do *log* de nível de teste, de acordo com a IEEE 829-2008, é fornecer um registro cronológico de informações relevantes sobre a execução dos testes. É dividida em três partes. Na introdução é identificada a organização e os detalhes de emissão e para ser completa, deve seguir a seguinte estrutura:

- a) identificador do documento: tal como no MTP;
- b) escopo: descreve o escopo do registro;
- c) referências: tal como no MTP;

A segunda parte do nível de *log* de teste inclui as informações de cada entrada do registro de teste, registrando as atividades/eventos para cada detalhe relevante, incluindo início e fim das atividades, data e hora de ocorrência e identidade do auto. Para o detalhamento ser completo, deve seguir a seguinte estrutura:

- a) descrição: identificador do nível de procedimento de teste a ser executado, registrando os participantes na execução de testes, indicando o papel de cada indivíduo;
- b) resultados dos procedimentos: cria um registro dos resultados para cada execução, informando o sucesso ou fracasso de cada caso de teste;
- c) informações de ambientes: registra as alterações no ambiente de testes que não estão de acordo com o plano;
- d) eventos anômalos: registra o ocorrido antes e depois de um evento inesperado. Caso a informação seja registrada em um relatório de anomalia, não é armazenada neste item;

- e) identificadores do relatório de anomalia: anota a identificação de cada relatório de anomalia, sempre que um é aberto.

A terceira parte do log de nível de teste descreve os documentos gerais, tal como no MTP.

2.4.4.2 Relatório de Anomalias (AR)

O relatório de anomalias tem como propósito documentar qualquer evento que ocorreu durante o processo de teste que requer investigação. Esse evento pode ser um problema, um incidente de teste, defeito, incômodo, emissão, anomalia ou informação de erro. O relatório de anomalia é dividido em três partes. De acordo com a IEEE 829-2008, a estrutura completa da introdução deve ter:

- a) introdução: identifica a organização emissora e os detalhes emitidos. Inclui requerimento de aprovação e status do documento;
- b) identificação do documento: identifica de forma única uma versão do documento, incluindo informações como a data de emissão, entidade emissora, autor (a), assinaturas de aprovação e status/versão. Identifica informações que também podem incluir os revisores e gestores pertinentes;
- c) escopo: descreve um contexto de informação não coberto em outra parte do relatório de anomalia, que é necessário fazer inteligível;
- d) referências: lista todos os documentos de aplicações referentes, sendo externas e internas.

A segunda parte do relatório de anomalias é formada pelos detalhes, que identificam os itens do relatório que inclui seu status e a tomada de ações corretivas.

- a) resumo: resumo da anomalia;
- b) data da descoberta da anomalia: registro da data que a anomalia foi identificada pela primeira vez;
- c) contexto: identifica o software ou o item do sistema, ou configuração do item, e/ou processo do ciclo de vida no qual a anomalia foi observada. Referencia o procedimento de teste, caso de teste e log de teste que pode ser fornecido. Identifica o nível de teste;
- d) descrição da anomalia: estabelece uma descrição da anomalia. Indica se a anomalia é reproduzida e estabelece informações suficientes para fazer a reprodução dela. A

descrição pode incluir os seguintes itens: entradas, resultados esperados, resultados atuais, resultados não esperados, passos de procedimentos, ambiente, tentativa de repetição, testadores e observadores;

- e) impacto: indica o impacto que esta anomalia impactará na questão técnica e de negócios. Pode incluir uma estimativa de tempo, esforço, risco e dificuldade do defeito;
- f) avaliação de urgência: provém uma evolução da necessidade de reparo imediato;
- g) descrição da ação corretiva: resume as atividades durante a tomada da ação corretiva para resolver a anomalia reportada;
- h) status da anomalia: identifica o atual status da anomalia;
- i) conclusões e recomendações: especificações e recomendações para mudança de desenvolvimento e/ou processos de testes e documentação que pode ajudar a prevenir esse tipo de anomalia futuramente;

A terceira parte do relatório é composta por dados gerais do documento:

- a) procedimento de mudança de documentos e histórico: especifica o significado de identificação, aprovação e registro de mudanças do plano de teste mestre.

2.4.4.3 Relatório de Status de Nível Provisório (LITSR)

O propósito do relatório de status de nível provisório, de acordo com a IEEE 829-2008, é resumir os resultados das atividades de teste designadas e opcionais para estabelecer avaliações recomendadas baseadas em seus resultados. O relatório é dividido em três partes, tendo como primeira parte a introdução:

- a) identificação do documento: tal como no AR;
- b) escopo: especifica o conteúdo e organização do documento. Inclui referências de qualquer informação capturada nas ferramentas automatizadas e não contém nesse documento;
- c) referências: tal como no AR.

A segunda parte do relatório é composta pelos detalhes, descrevendo o resumo do status de teste, mudanças do plano de teste e métricas do status de teste. Contém a seguinte estrutura:

- a) resumo do status de teste: resume os resultados do status de teste comparando com o plano de teste passado. Resume atual versus o status desejado do relatório de

anomalia para este nível de teste;

- b) mudanças do plano: resume os testes que foram planejados para serem executados , que ainda não foram executados e por que. Estabelece mudanças de testes permanentes;
- c) métricas do status de teste: estabelece uma métrica de status provisório para um nível de plano de teste.

A terceira parte do relatório é composta por dados gerais do documento.

2.4.4.4 Relatório de Nível de Teste (LTR)

O relatório de nível de teste tem como finalidade resumir os resultados das atividades de teste designadas e estabelecer avaliações e recomendações baseadas em seus resultados. Conforme a IEEE 829-2008, há um relatório de nível de teste para cada nível de teste definido por organização ou projeto. O relatório é dividido em três partes, tendo como primeira parte a introdução, tal como no MTP.

A segunda parte do relatório é composta por todos os detalhes dos resultados de teste, racionalidade para todas as decisões e no final, conclusões e recomendações:

- a) resumo dos resultados de teste: resume a avaliação dos itens de teste. Identifica os itens testados, indicando seu nível de versão/revisão. Indica o ambiente no qual as atividades de teste ocorreram e seus impactos;
- b) detalhamento dos resultados de teste: resume os resultados testados. Identifica todas as anomalias resolvidas e resume suas soluções. Resume as principais atividades de testes e eventos;
- c) racionalidade para decisões: especifica a emissão que foi considerada para qualquer decisão e razão para selecionar suas conclusões;
- d) conclusões e recomendações: especifica a avaliação de abrangência para cada item de teste, incluindo limitações. Essa avaliação é baseada em resultados de teste e seus itens de aprovação/critério de falha.

A terceira parte do relatório é composta por dados gerais do documento, tais como no MTP.

2.4.4.5 Relatório Mestre de Teste (MTR)

De acordo com a IEEE 829-2008, o relatório de plano de teste mestre tem como objetivo resumir os resultados dos níveis de atividades de teste e fornecer avaliações com base nesses resultados. Sempre que o plano de teste mestre for gerado ou implementado, é necessário que os resultados descritos no relatório correspondam ao plano de teste mestre. O relatório de teste mestre é dividido em três partes e para a primeira ser completa, deve seguir estrutura tal como no LITSR.

A segunda parte do relatório descreve o resumo de todos os resultados de testes agregados, racionalidade para decisões e conclusões e recomendações finais:

- a) resumo de todos os resultados de testes: resume as avaliações de níveis de teste, sendo: resumo das atividades de testes, resumos dos resultados das tarefas de teste, resumo das anomalias e soluções, avaliação de publicação/adição de qualidade e resumo da métrica final coletada;
- b) racionalidade para decisão: especifica as razões de conclusão para avaliações, falhas e avaliações condicionais para o software ou sistema;
- c) conclusões e recomendações: especifica uma avaliação do software. Descreve lições aprendidas e mudanças de resultados na prática que foram descobertos durante as conduções dos testes.

A terceira parte do relatório é composta por dados gerais do documento, tal como no MTP.

2.5. TRABALHOS CORRELATOS

Dentre os trabalhos correlatos ao tema IEEE-829 e ferramenta de suporte ao gerenciamento de testes, são apresentados a seguir os que tiveram maior relevância, os quais são: sistema de gerenciamento do teste de software, TestPlan e Testlink

2.5.1 Sander

Sander (2002) apresenta a construção de uma ferramenta que possibilita o suporte ao gerenciamento do teste de software baseados na norma ISO/IEC 12207, para o processo de gerência e atividades de teste. A ferramenta é voltada ao gerenciamento manual de testes, sendo independente da abordagem do desenvolvimento e automação de testes, podendo ser configurada conforme o objetivo da aplicação.

A Figura 3 mostra uma tela de demonstração da ferramenta de Sander (2002).



Figura 3 – Menu principal do sistema

A ferramenta implementa o registro de casos, planos e resultados de testes. Há também relatórios que são baseados nos planos e resultados de teste.

Sendo uma ferramenta desenvolvida na plataforma Dataflex e, por isso, ter uma interface DOS, na qual não se utiliza o mouse, sua usabilidade fica um tanto comprometida na navegação de vários menus consecutivos, que seria facilitado com o uso do mouse.

Além disso, a plataforma DataFlex possui um banco de dados proprietário e não suporta novos sistemas operacionais de 64 bits.

Apesar deste trabalho correlato embasar-se na norma ISO/IEC 12207, a norma IEEE-829 (versão de 2008) é embasada justamente no enfoque em processo dado pela ISO/IEC 12207 (IEEE, 2008, p. 14). De acordo com a ISO/IEC 12207 (1998), “o objetivo da ISO/IEC 12207 é estabelecer uma estrutura comum para os processos de ciclo de vida de software, com uma terminologia bem definida, que pode ser referenciada pela indústria de software”.

2.5.2 Labes

Labes (2010) apresenta a ferramenta TestPlan, que apóia o planejamento e controle das atividades do processo de testes da empresa HBSIS Informática, sugerindo a extinção do uso de planilhas Excel para a geração dos artefatos, seguindo o modelo *Capability Maturity Model Integration* (CMMI). As áreas de processos utilizadas nas criações das planilhas são a verificação e a validação.

A ferramenta foi desenvolvida a partir do *framework* .NET, na linguagem ASP, e realiza a importação dos diagramas de casos de uso a partir do *Enterprise Architect* (EA). A Figura 4 apresenta a tela de criação de casos de teste.

The screenshot shows the 'Teste-Plan - Pesquisa - Caso de Teste' web application. The interface features a top navigation bar with the HBSIS logo and a user profile 'Usuário: maiana Logout'. Below this is a secondary navigation bar with tabs for 'Testes', 'Sistemas', and 'Pessoas'. The main content area is titled 'Caso de Teste' and contains a form with the following fields:

- * Identificador: [Text input field]
- * Responsável atual: [Dropdown menu with 'Selezione' selected]
- Criado em: 06/06/2010 [Text input field]
- * Situação: [Dropdown menu with 'Em Elaboração' selected]
- * Sistema: [Dropdown menu with 'Selezione' selected]
- * Módulo: [Dropdown menu with 'Selezione' selected]
- * Desenho de teste: [Dropdown menu with 'Selezione' selected]
- * Caso de uso (importado): [Dropdown menu with 'Selezione' selected]
- Caso de uso: [Text input field]

At the bottom right of the form are 'Salvar' and 'Voltar' buttons. The footer of the application shows 'Concluído' on the left and 'Intranet local | Modo Protegido: Desativado' on the right.

Figura 4 – Criação de caso de teste

A ferramenta implementa casos de uso, desenho e caso de testes e emite relatórios

baseados nesses três documentos de testes.

Segundo Labes (2010), é uma ferramenta de alta legibilidade e de interface amigável. Facilita o trabalho do testador e/ou usuário do sistema. Por se tratar de uma ferramenta para o ambiente *web*, pode ser acessada de qualquer ponto onde se tenha acesso à internet, limitando-se apenas ao navegador Internet Explorer. Utiliza SQL Server como banco de dados, principalmente porque este é integrado com o .NET Framework.

2.5.3 TestLink

A ferramenta TestLink oferece um ambiente que facilita a criação e manutenção de casos de testes como também pode organizá-los em planos de teste. Esses planos de teste permitem que os usuários executem casos de testes, gravando seus resultados dinamicamente, gerando relatórios, traçando requisitos de *software*, priorizando e atribuindo tarefas.

Esta ferramenta tem sua interface baseada em PHP com banco de dados MYSQL, *Postgres* e MS-SQL. O ambiente também coopera com ferramentas conhecidas de verificação de erros como Bugzilla² e Mantis³ (TESTLINK, 2006). A ferramenta é de código aberto e é mantida por uma comunidade aberta de testadores. Segundo os desenvolvedores do ambiente, o Testlink torna o processo de testes fácil e organizado. A Figura 5 exibe a visualização da especificação dos casos de testes.

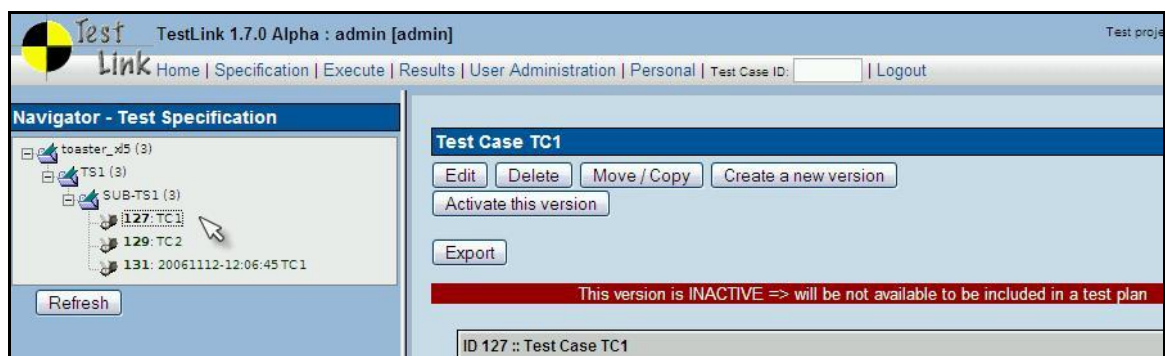


Figura 5 – Visualização da especificação dos casos de testes da ferramenta Testlink

Analisando-se a ferramenta, percebe-se que o Testlink segue a estrutura aplicada na norma IEEE Std 829-1998.

² Bugzilla é uma ferramenta *opensource* que é utilizada como rastreamento de erros (*bugs*).

³ Mantis é uma ferramenta de gerenciamento de defeitos de outros softwares.

3. DESENVOLVIMENTO

Este capítulo aborda as metodologias utilizadas para o desenvolvimento da ferramenta. A primeira parte descreve os principais requisitos funcionais e não-funcionais atendidos pela ferramenta. A segunda parte é composta pela especificação dos diagramas de casos de uso e diagrama de entidade e relacionamento (MER). A terceira parte é composta pela implementação, onde são explanadas as técnicas e ferramentas utilizadas, operacionalidade e principais telas. A quarta parte aborda os resultados e discussões do trabalho.

3.1. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Esta seção apresenta os principais requisitos funcionais e não funcionais do trabalho. O Quadro 1 apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com os casos de uso associados.

O Quadro 2 lista os requisitos não funcionais previstos para o sistema.

Requisitos Funcionais	Caso de Uso
RF01: O sistema deverá permitir manter usuários.	UC01
RF02: O sistema deverá permitir manter planos de testes.	UC02
RF03: O sistema deverá permitir manter de casos de testes dentro de um plano de testes cadastrado.	UC03
RF04: O sistema deverá permitir a inclusão de resultados dos testes.	UC04
RF05: O sistema deverá emitir relatórios de anomalias, a partir dos resultados dos casos de testes.	UC05
RF06: O sistema deverá emitir relatórios de <i>log</i> , a partir dos resultados dos casos de testes.	UC06
RF07: O sistema deverá emitir relatórios de nível de teste, de acordo com a norma IEEE 829, a partir dos resultados dos casos de testes.	UC07
RF08: O sistema deverá emitir relatório de teste mestre, de acordo com a norma IEEE 829, a partir dos resultados dos casos de testes.	UC08
RF09: O sistema deverá permitir o gerenciamento de perfis pelo administrador.	UC01

Quadro 1 – Requisitos funcionais

Requisitos Não Funcionais
RNF01: O sistema deverá ser desenvolvido na linguagem <i>Java Enterprise Edition</i> (JEE), no ambiente Eclipse WTP.
RNF02: O sistema utilizará o banco de dados MYSQL.
RNF03: O sistema deverá utilizar como servidor <i>web</i> o Apache Tomcat.
RNF05: O sistema deverá seguir a norma IEEE-829.
RNF06: O sistema deverá suportar o navegador Internet Explorer 8 e Chrome versão 1.

Quadro 2 – Requisitos não funcionais

3.2. ESPECIFICAÇÃO

Esta seção descreve os modelos e diagramas do trabalho, cuja ferramenta utilizada para desenvolvimento foi o EA.

3.2.1 Diagramas de Casos de Uso

Na Figura 6 têm-se os casos de uso sobre a atividade do administrador do sistema.

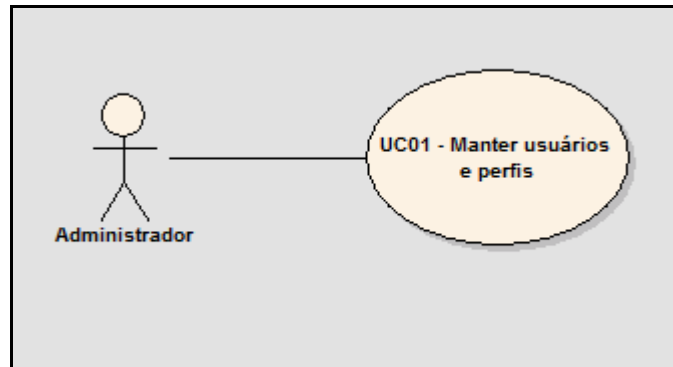


Figura 6 – Diagrama de caso de uso administrador

Na Figura 7 têm-se os casos de uso sobre as atividades do analista de testes.

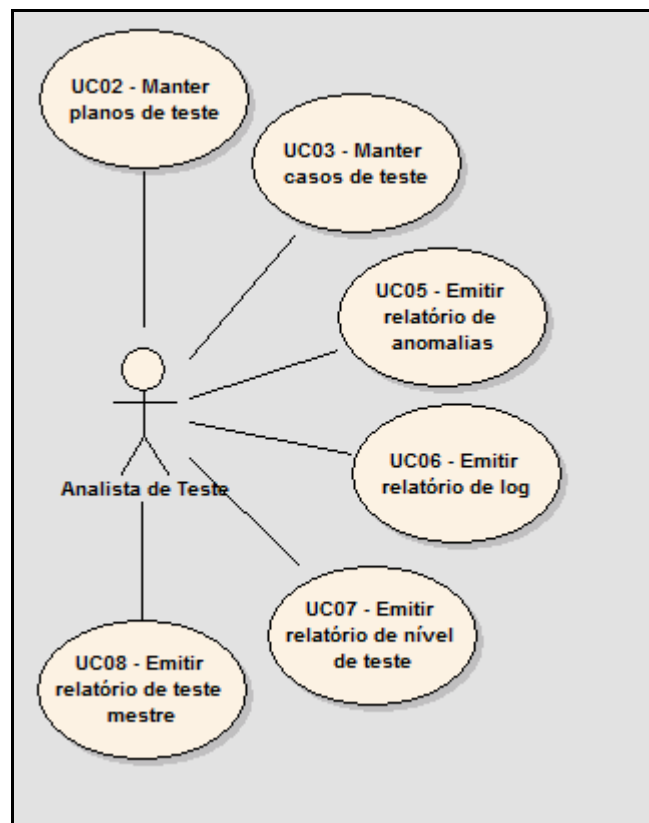


Figura 7 – Diagrama de caso de uso analista de testes

Na Figura 8 têm-se os casos de uso sobre as atividades a serem realizadas pelo testador.

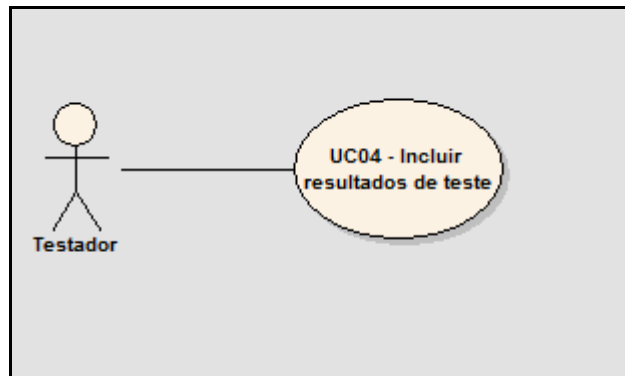


Figura 8 – Diagrama de caso de uso testador

A seguir é apresentado um breve comentário sobre os casos de uso:

- a) UC01 – Manter usuários: permite o cadastro e gerenciamento dos usuários e perfis que terão acesso à ferramenta;
- b) UC02 – Manter plano de testes: permite o cadastro e gerenciamento dos planos de testes, filhas do MTP;
- c) UC03 – Manter casos de testes: permite o cadastro e gerenciamento dos casos de testes em um plano de testes cadastrado;
- d) UC04 – Incluir resultados de testes: permite a inclusão dos resultados de testes de um determinado caso de testes, gerando a base para a emissão do relatório de anomalia;
- e) UC05 – Emitir do relatório anomalias: permite a emissão do relatório contendo um resumo das anomalias registradas pelo testador nos resultados de testes e seus impactos;
- f) UC06 – Emitir relatório de *Log*: permite a emissão do relatório a partir dos resultados de teste cadastrados;
- g) UC07 – Emitir relatório de nível de teste: permite a emissão do relatório de nível de teste, de acordo com o plano cadastrado;
- h) UC08 – Emitir do relatório de teste mestre: permite a emissão do relatório contendo todos os documentos filhos.

O detalhamento dos principais casos de uso está no Apêndice A.

3.2.2 Modelo de entidade e relacionamento

A Figura 9 apresenta as tabelas utilizadas no banco de dados do sistema.

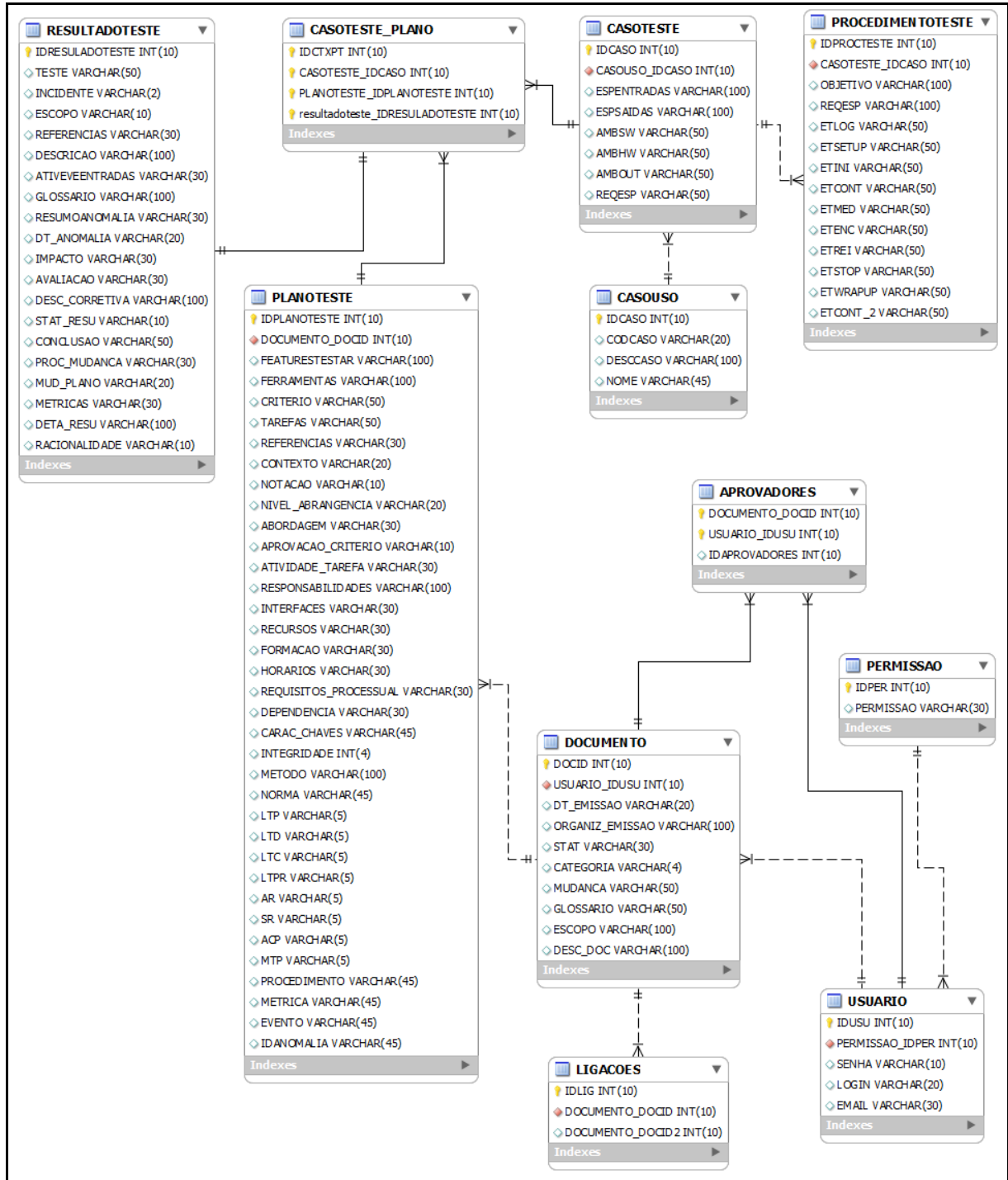


Figura 9 – Modelo de entidade e relacionamento

Estas tabelas têm como objetivos:

- a) Permissao: armazena as permissões dos usuários;

- b) `Usuario`: armazena os dados do usuário;
- c) `Aprovadores`: armazena os aprovadores de cada documento;
- d) `Documento`: armazena os principais documentos da norma;
- e) `Ligacoes`: realiza a ligação entre dois documentos;
- f) `ResultadoTeste`: armazena os resultados de teste dos casos de testes executados;
- g) `PlanoTeste`: armazena as informações do documento LTP;
- h) `CasoTeste_Plano`: armazena o relacionamento entre o LTP (tabela `PlanoTeste`) e o LTC (tabela `CasoTeste`);
- i) `CasoTeste`: armazena as informações do documento LTC;
- j) `CasoUso`: armazena as informações dos casos de uso;
- k) `ProcedimentoTeste`: armazena as informações do LTPr.

O dicionário de dados está descrito no Apêndice B.

3.3. IMPLEMENTAÇÃO

Esta seção contém o detalhamento sobre o desenvolvimento da ferramenta. A primeira parte identifica as técnicas e ferramentas utilizadas. A segunda apresenta um estudo de caso do ponto de vista do usuário e a terceira descreve os resultados obtidos.

3.3.1 Técnicas e ferramentas utilizadas

A ferramenta foi desenvolvida na linguagem JAVA, utilizando a plataforma de desenvolvimento Eclipse Helios. O servidor *web* utilizado foi o Apache Tomcat na versão 6.4. Para armazenar as informações foi utilizado o banco de dados MYSQL.

Para criação das páginas do sistema foi utilizado o JSP, recurso fornecido pela linguagem JAVA para desenvolvimento *web*, o qual comporta tanto linguagem JAVA como HTML, sendo implementados scripts JavaScript.

O MySQL é um sistema de gerenciamento de banco de dados relacional (*Relational*

Database Management System – RDBMS) multiusuário e multiencadeado, que utiliza o SQL (*Structured Query Language*) como a linguagem de consulta padrão. O MySQL utilizado está sob uma licença *Open Source*.

Para a criação do banco de dados foi utilizado a ferramenta de criação DBDesigner, onde a partir do modelo entidade-relacionamento foram exportados os comandos SQL para criação das tabelas e demais objetos. Esses comandos foram aplicados no banco de dados e alterados conforme a necessidade do sistema. Foram criados índices adicionais para aumentar o desempenho da consulta no banco de dados.

A interface foi idealizada de maneira mais amigável ao usuário, utilizando a ferramenta *Dreamweaver* para sua criação, da empresa ADOBE. Esta ferramenta também foi utilizada na criação dos CSS utilizados pelo sistema.

3.3.2 Operacionalidade da implementação

Nessa seção é apresentada a sequência de telas e operações para cada tipo de usuário, para conseguir utilizar o sistema corretamente. Também serão apresentados alguns trechos de códigos fonte.

3.3.2.1 Acessar a ferramenta

A Figura 10 apresenta a tela de Login, onde o usuário deve informar seu usuário e senha para acessar a ferramenta. Esta tela é comum a todos os usuários.

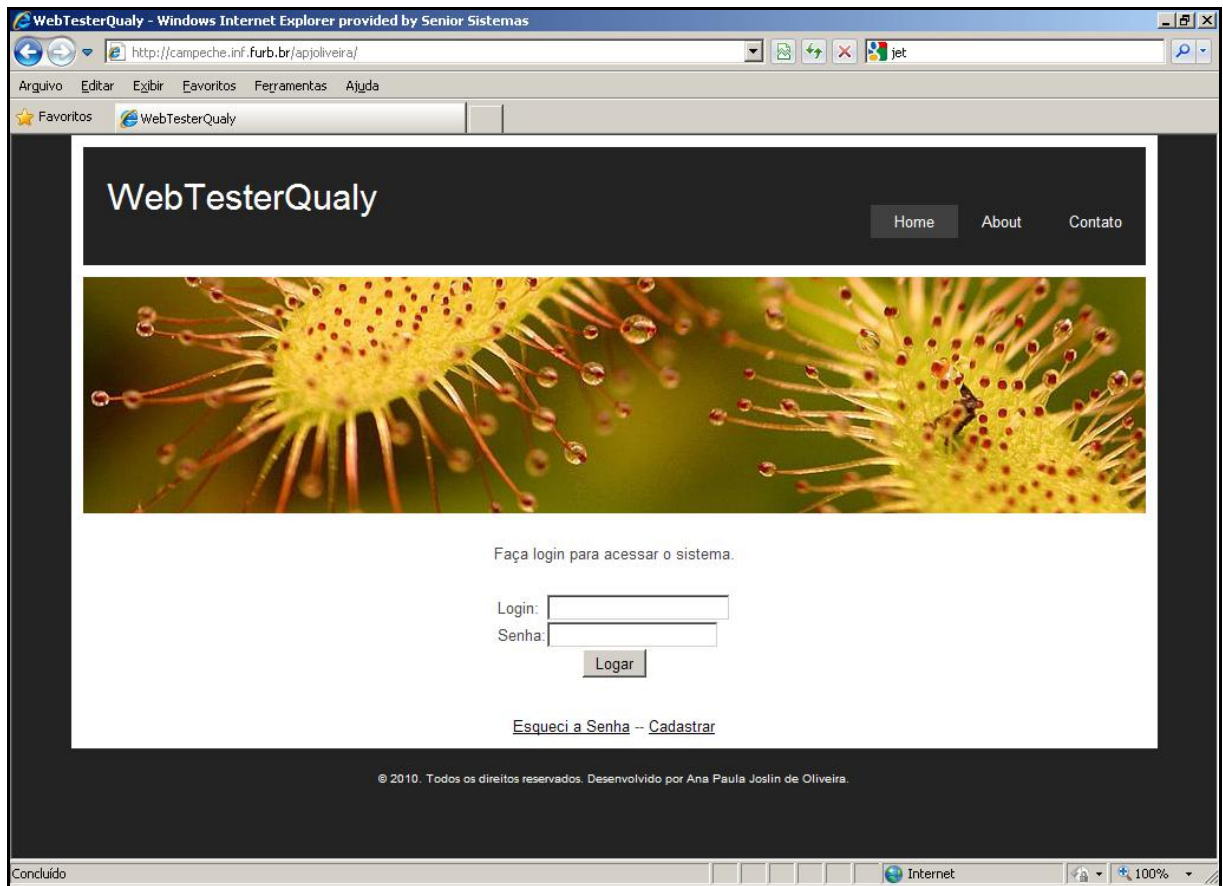


Figura 10 – Login no Sistema

Informado o usuário e senha, o usuário clica sobre o botão “Logar”. É realizada a verificação da existência do usuário. Caso não esteja cadastrado, será apresentada a mensagem: “Usuário não cadastrado!”. Se a senha não estiver correta, apresenta-se a mensagem: “Senha Inválida!”. Caso o usuário e senha estejam corretos, a tela inicial é exibida com as opções de menu, de acordo com o tipo de acesso do usuário, sendo administrador, analista ou testador.

A administração da ferramenta é realizada por um usuário com *status* do tipo “Gerente”. O usuário “admin” tem como papel o gerenciamento de usuários, conforme mostra a Figura 11.

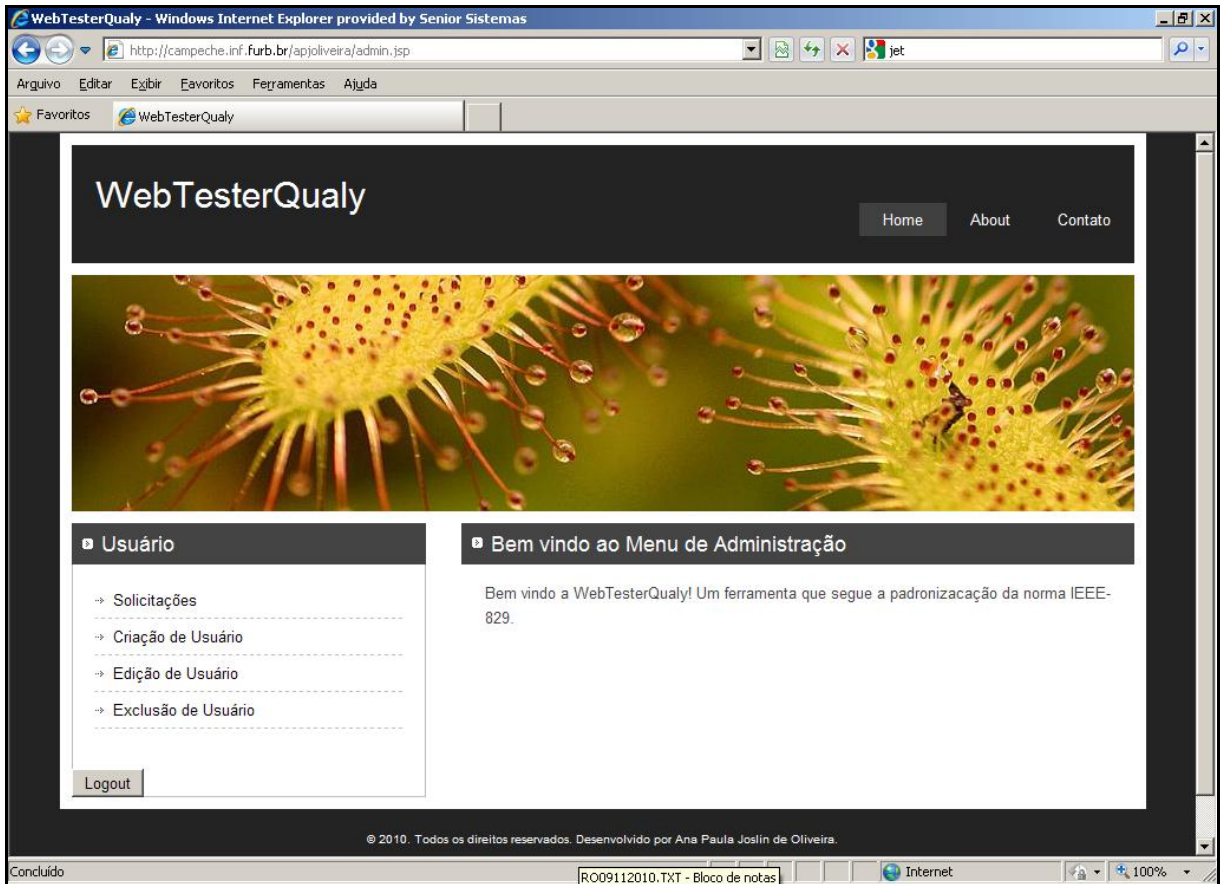


Figura 11 - Gerenciamento de usuários.

3.3.2.2 Tela de Criação de Usuário

A Figura 12 apresenta a criação de um usuário com o status de “Gerente”. Como exemplo será demonstrada a criação somente desse usuário. Porém, foram criados os usuários com as permissões “Analista” e “Tester” para dar continuidade na exemplificação.

Figura 12 – Criação do usuário com status “Analista”

Na criação do usuário, a pessoa conectada deve seguir alguns parâmetros para que a operação ocorra com sucesso, tais como: criar um usuário com um nome diferente de outro já existente, criar uma senha sem mantê-la em branco, tomar cuidado com o formato do email que está inserindo. O email deve seguir o padrão nome@(arroba)nome.(ponto)nome.

3.3.2.3 Tela inicial do sistema para o usuário Analista

A Figura 13 apresenta os menus que o usuário tipo “Analista” pode acessar. Esta tela permite a administração dos seguintes recursos da ferramenta: criação, alteração e exclusão de documentos, planos e casos de teste, casos de uso, como também a emissão de relatórios.

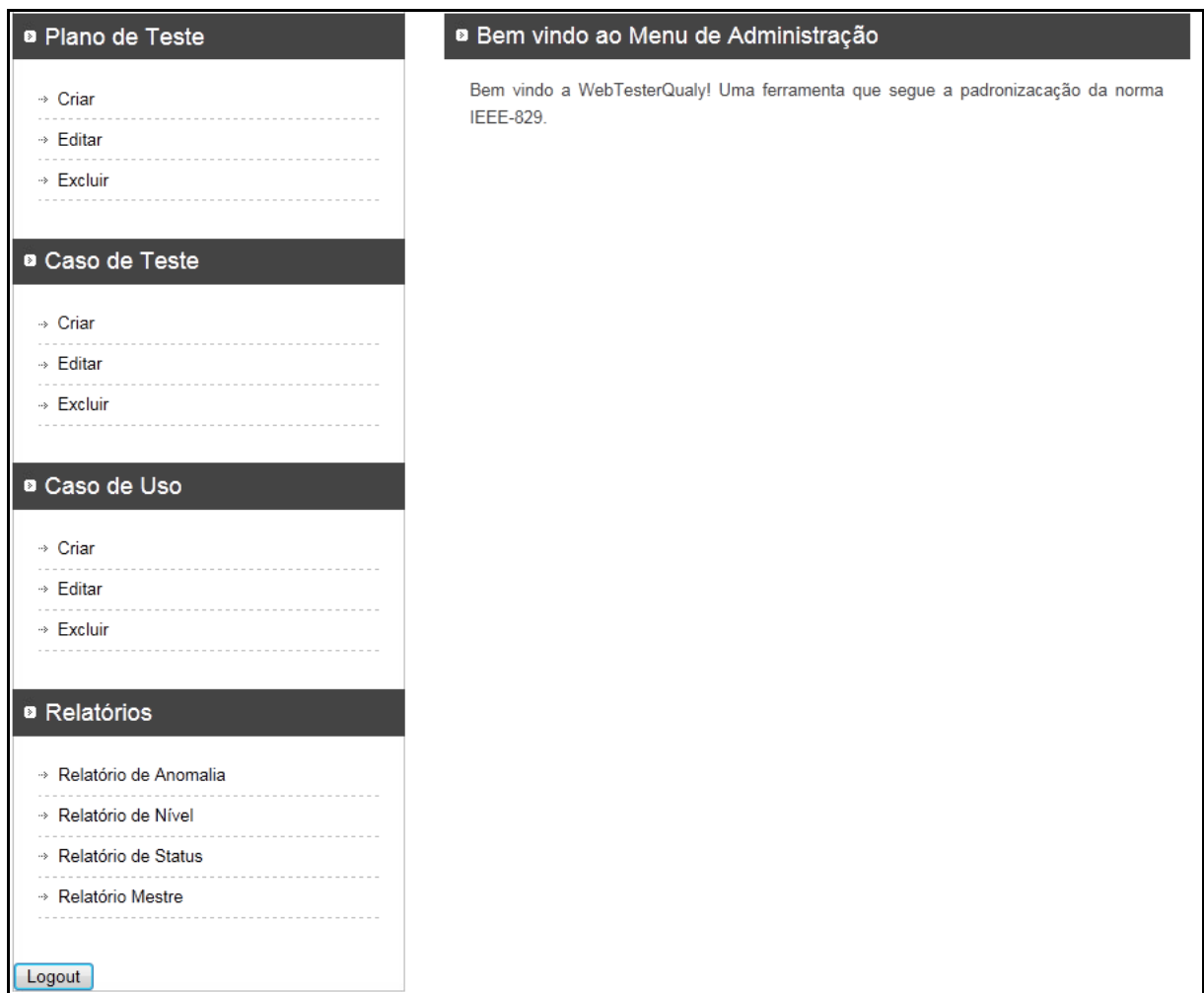


Figura 13 – Tela de boas vindas do usuário Analista

3.3.2.4 Planos de Teste

A página de criação de planos de teste é exibida na Figura 14. Pode-se criar o plano de teste mestre, nível de plano de teste, nível de procedimento de teste e nível de procedimento de aceitação de teste. O MTP deve ser o primeiro a ser inserido, pois é partir deste que os demais documentos poderão ser criados.



Figura 14 – Criação do Plano de teste

Na Figura 15 é possível verificar a tela de criação de um documento do tipo MTP. É necessário selecionar o documento e clicar no botão “Criar”. No MTP são inseridos campos gerais como escopo e glossário que servirá como base para outros documentos. Existem também campos onde é informado a data e horário de emissão do documento.

■
Criar MTP

1.Introdução

1.1 Código MTP031

1.2 Escopo

Projeto da calculadora do windows - modo simples.

Organizar Emissão

Equipe de testes Catar

Status
 Novo Andamento Concluído

1.3 Referências

"Calculadora_windows_bas.xls"

1.4 Descrição

Projeto de verificacao dos cálculos da calculadora do windows - modo básico.

1.5 Características Chaves

Somar, diminuir, ultiplicar, dividir, potência e raiz quadrada.

1.6 Nível de Integridade

Nível 3 - Crítico

1.7 Aprovador

gerente

1.8 Ferramentas

Calculadora do windows

1.9 Técnicas/Métodos

metodos.xls

1.10 Data Emissão

30/06/2011

1.11 Horário

30/06/2011 10:27PM

2.Detalhamentos dos Processos

2.1 Atividades/Tarefas

1239, 12980 3 2130.

2.2 Número/Seqüencia Nível de Teste

1

2.3 Requisitos/Documentos

UC01, UC010.]

Figura 15 – Criação de MTP

Após clicar no botão “Criar MTP” é apresentada a mensagem da criação do documento, com seu respectivo código, conforme a Figura 16.

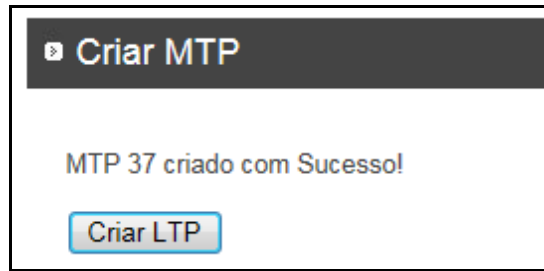


Figura 16 – Mensagem da criação do documento

O Quadro 3 exibe o código-fonte da criação do plano de teste mestre.

```

<%@include file="conecta.jsp" %>
<%@page import="java.util.*"%>
<%@include file="gcab.jsp"%>
    <h1 class="title">Criar MTP</h1><br><br>
    <form action="mtpform.jsp" method="post">
        <table>
            <tr>
                <td><b>1. Introdu&ccedil; &atilde;o</b></td>
            </tr>
            <tr>
                <td></td>
                <%
                    int codigo = 0;
                    try
                    {
                        PreparedStatement select = con.prepareStatement("select
count(*) from DOCUMENTO");
                        ResultSet selectrs = select.executeQuery();
                        while (selectrs.next())
                        {
                            codigo=selectrs.getInt(1);
                        }
                        select.close();
                        selectrs.close();
                    }
                    catch (Exception e) {out.println("Erro gerado pelo sistema
:<b>"+ e.toString()+"</b>");}
                    codigo = codigo +1;
                %>
                <td>1.1 C&oacute;digo<%= " MTP0"+codigo%></td> </td>
            </tr>
            <tr>
                <td></td>
                <td>1.2 Escopo<br>
                <textarea name="escopo" id="escopo" cols="50"
rows="3"></textarea></td>
            </tr>
            <tr>
                <td></td>
                <td>Organizar Emiss&atilde;o<br>
                <textarea name="organiz_emissao" id="organiz_emissao" cols="50"
rows="3"></textarea></td>
            </tr>
            <tr>
                <td></td>
                <td>Status<br>
                <input type="radio" name="stat" value="novo" checked> Novo
                <input type="radio" name="stat" value="andamento" > Andamento
                <input type="radio" name="stat" value="concluido">
Conclu&iacute;do </td>
            </tr>
            <tr>
                <td></td>
                <td>1.3 Refer&ecirc;ncias<br>

```

```

        <textarea name="referencia" id="referencia" cols="50"
rows="3"></textarea></td>
    </tr>
    <tr>
        <td></td>
        <td>1.4 Descri&ccedil;&atilde;o<br>
        <textarea name="desc_doc" id="desc_doc" cols="50"
rows="3"></textarea></td>
    </tr>
    <tr>
        <td></td>
        <td>1.5 Caracter&iacute;sticas Chaves<br>
        <textarea name="carac_chave" id="carac_chave" cols="50"
rows="3"></textarea></td>
    </tr>
    <tr>
        <td></td>
        <td>1.6 N&iacute;vel de Integridade<br>
        <select name="integridade" id="integridade" >
            <option value="3">N&iacute;vel 3 -
Cr&iacute;tico</option>
            <option value="3">N&iacute;vel 4 -
Catastr&oacute;fico</option>
        </select>
    </tr>
    <tr>
        <td></td>
        <td>1.7 Aprovador<br>
        <select name="aprovador" id="aprovador" >
            <%
                ArrayList <String> usuario = new
ArrayList<String>();
                try
                {
                    PreparedStatement select =
con.prepareStatement("select login from USUARIO where login <>'admin' order by
idusu");
                    ResultSet selectrs = select.executeQuery();
                    while (selectrs.next())
                    {
                        usuario.add(selectrs.getString(1));
                    }
                    select.close();
                    selectrs.close();
                }
                catch (Exception e){
                    throw new Exception(e.toString());}
                for (int j = 0; j < usuario.size(); j++)
                {
                    %><option
value="<%=usuario.get(j)%>"><%=usuario.get(j)%> </option><%
                }
                %>
            </select>
        </td>
    </tr>
    <tr>
        <td></td>
        <td>1.8 Ferramentas<br>
        <textarea name="ferramentas" id="ferramentas" cols="50"
rows="3"></textarea></td>
    </tr>
    <tr>
        <td></td>
        <td>1.9 T&eacute;cnicas/M&eacute;todos<br>
        <textarea name="metodo" id="metodo" cols="50"
rows="3"></textarea></td>
    </tr>

```

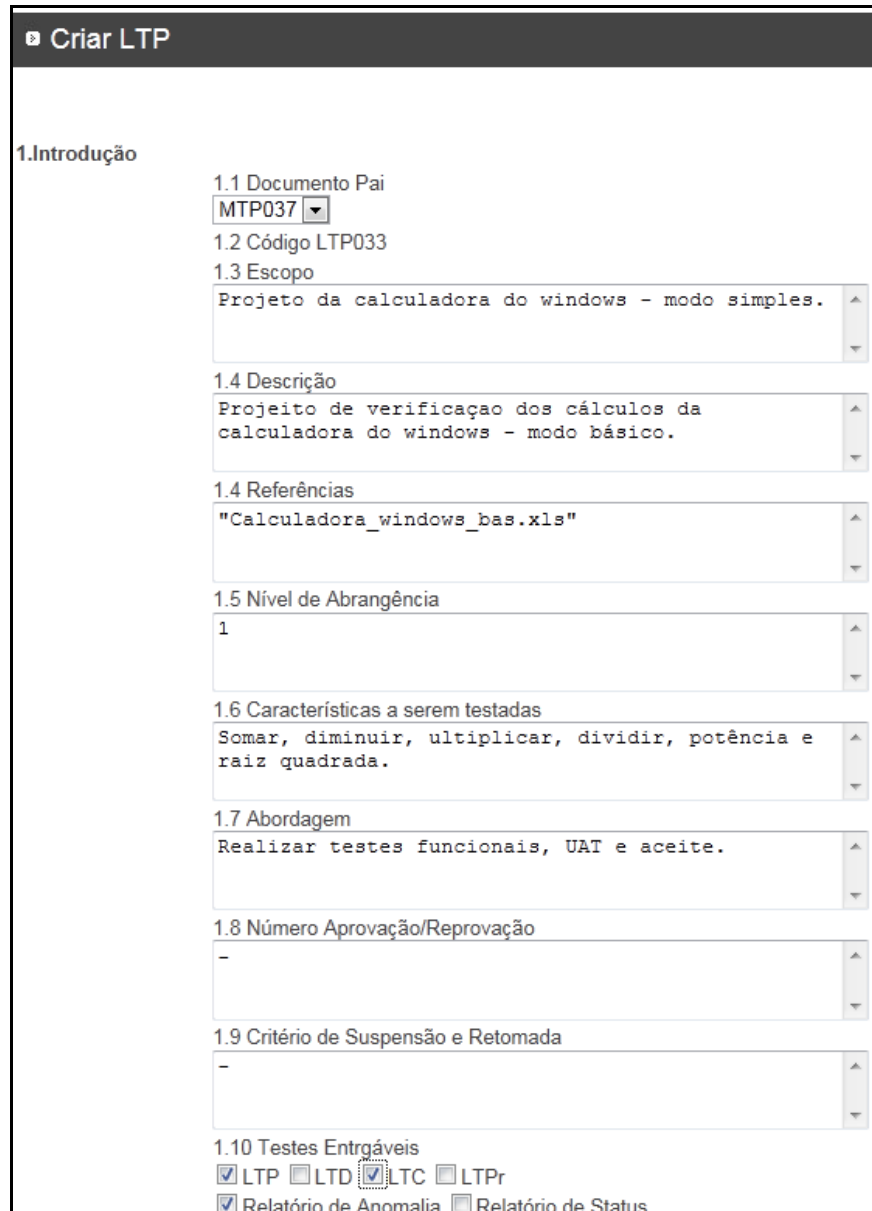
```

        <tr>
            <td></td>
            <td>1.10 Data Emiss&atilde;o<br>
            <%@include file="ano.html"%></td> </td>
        </tr>
        <tr>
            <td></td>
            <td>1.11 Hor&aacute;rio<br>
            <%@include file="clock.html"%></td> </td>
        </tr>
        <tr>
            <td><b>2.Detalhamentos dos <br>Processos</b></td>
        </tr>
        <tr>
            <td></td>
            <td>2.1 Atividades/Tarefas<br>
            <textarea name="atividade" id="atividade" cols="50"
rows="3"></textarea></td>
        </tr>
        <tr>
            <td></td>
            <td>2.2 N&uacute;mero/Seq&uuml;encia N&iacute;vel de Teste<br>
            <textarea name="sequencia" id="sequencia" cols="50"
rows="3"></textarea></td>
        </tr>
        <tr>
            <td></td>
            <td>2.3 Requisitos/Documentos<br>
            <textarea name="requisitos" id="requisitos" cols="50"
rows="3"></textarea></td>
        </tr>
        <tr>
            <td></td>
            <td>2.4 Normas<br>
            <textarea name="norma" id="norma" cols="50"
rows="3"></textarea></td>
        </tr>
        <tr>
            <td><b>3.Geral</b></td>
        </tr>
        <tr>
            <td></td>
            <td>3.1 Gloss&aacute;rio<br>
            <textarea name="glossario" id="glossario" cols="50"
rows="3"></textarea></td>
        </tr>
        <tr>
            <td></td>
            <td>3.2 Procedimentos de Mudan&ccedil;a<br>
            <textarea name="mudanca" id="mudanca" cols="50"
rows="3"></textarea></td>
        </tr>
        <tr><br>
            <td><input type="submit" name="button" value="Criar MTP"/></td>
        </tr>
    </table>
</form>
<%@include file="grid.jsp"%></td> </td>

```

Quadro 3 – Criação do relatório de teste mestre

A criação de outros documentos como o LTP depende da criação do MTP, pois cada LTP precisa ter um documento mestre (MTP) atrelado a ele quando a integridade for 3 ou 4. Na Figura 17, pode ser verificado o passo, na criação do LTP, onde o documento é relacionado a um MTP.



Criar LTP

1.Introdução

1.1 Documento Pai
MTP037

1.2 Código LTP033

1.3 Escopo
Projeto da calculadora do windows - modo simples.

1.4 Descrição
Projeto de verificação dos cálculos da calculadora do windows - modo básico.

1.4 Referências
"Calculadora_windows_bas.xls"

1.5 Nível de Abrangência
1

1.6 Características a serem testadas
Somar, diminuir, ultiplicar, dividir, potência e raiz quadrada.

1.7 Abordagem
Realizar testes funcionais, UAT e aceite.

1.8 Número Aprovação/Reprovação
-

1.9 Critério de Suspensão e Retomada
-

1.10 Testes Entrgáveis
 LTP LTD LTC LTPr
 Relatório de Anomalia Relatório de Status

Figura 17– Criação de LTP

Outro ponto relevante na criação de um LTP é atrelamento do LTP com um documento de nível menor, o LTC, esta situação pode ser observada na Figura 18.

2. Detalhes de Teste	
2.1 Atividades/Tarefas	1239, 12980 e 2130.
2.2 Ambiente e InfraEstrutura	Calculadora do windows
2.3 Responsável	analista
2.4 Interface entre as partes envolvidas	não há
2.5 Recursos Adicionais	não há
2.6 Formação	-
2.7 Riscos e Contingências	-

Figura 18 – Arelamento LTP

Após criar o relacionamento entre o LTP e LTC o documento pode ser criado. Como consequência desse relacionamento ao verificarmos a tela de criação do LTC não tem um documento pai a ser referenciado, pois a referência já foi realizada. Para o LTC escolhemos um caso de uso já criado anteriormente, que fará menção ao LTC que está sendo criado no momento. A Figura 19 demonstra esse processo detalhadamente.

❏
Criar LTC

1.Introdução

1.1 Caso de uso relacionados

1.2 Código LTC032

1.3 Escopo

1.4 Referências

1.5 Descrição

1.5 Responsável

1.6 Contexto

1.7 Notação

2.Detalhamento

2.1 Identificador Caso de Teste TC01

2.2 Objetivo

2.3 Entradas

2.4 Resultados

Figura 19 – Escolha caso de uso em LTC

3.3.2.5 Resultados de Teste

Para incluir resultados de teste referentes a um caso de teste, o usuário deve ter a permissão do tipo *tester*. Neste passo o usuário somente pode escolher para inserir resultados de testes nos casos em que ele foi configurado para ser o testador. Além de informar o teste e inserir uma breve descrição sobre os resultados o testador deve informar se o resultado a ser inserido é um incidente, pois essa informação será futuramente utilizada pelo relatório de anomalias. Na Figura 20, pode ser visto a inserção de resultados de teste.

The screenshot shows a web application interface with two main panels. The left panel, titled "Teste", contains a link "→ Inserir Resultado de Teste" and a "Logout" button. The right panel, titled "Insere Resultado de Teste", contains several input fields:

- Selecionar LTC:** A dropdown menu with the value "LTP024".
- Referências:** An empty text area.
- Descrição:** A text area containing the text "Problemas encontrados na multiplicação de 3."
- Atividades de Entrada:** A text area containing the text "3*5".
- Teste:** A text area containing the text "Informado o número 3, sinal de multiplicação e 5."

Figura 20 – Inserção de Resultado de Teste

3.3.2.6 Relatórios

Para a geração dos relatórios citados pela norma, o usuário deve ter permissão de gerente. Em seguida serão demonstrados as telas de geração de relatórios e seus respectivos retornos.

3.3.2.6.1 Relatório de anomalia

Na Figura 21, pode ser vista a tela de escolha do LTC de onde provirá todas as anomalias registradas nos resultados de teste que estão referenciados a esse documento.

Figura 21 – Geração relatório de anomalia

Na Figura 22, pode ser vista o retorno desse relatório, quando é selecionado o documento “LTCXX”.

Figura 22 – Retorno relatório de anomalia

3.3.2.6.2 Relatório de nível de teste

Na Figura 23, pode ser verificado o retorno do relatório de nível onde foi escolhido um LTC.

Gerar Relatório de Nível

Introdução
Identificação do Documento: LTC06
Data de Emissão: 15:53:41-20/07/11
Autor: analista
Aprovador: analista

Detalhes RT03
Identificação do Resultado: RT03
Resumo do Resultado: -
Referências: -
Detalhamento de Teste: Problemas encontrados na multiplicação de 3.
Racionalidade: -
Conclusões e Recomendações: -

Gerais RT03
Glossário: null
Procedimento de Mudança: -

[Voltar](#)

Figura 23 – Relatório de nível de teste

3.3.2.6.3 Relatório de Status

Na Figura 20 é apresentado o relatório status de teste. Deve-se selecionar o nível de plano a ser verificado e clicar no botão “Gerar Relatório”.

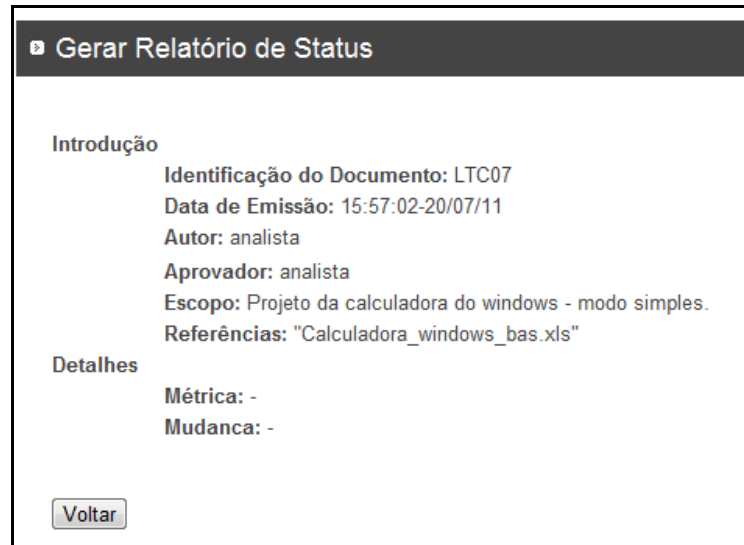


Figura 24 – Relatório de status

3.3.2.6.4 Relatório de Teste Mestre

Para gerar o relatório de teste mestre é necessário clicar em “Relatório de Status”, selecionar o Plano de testes mestre e clicar no botão “Gerar Relatório”. A Figura 25 apresenta o relatório completo.


Gerar Relatório Mestre

Introdução

Identificação do Documento: MTP05
 Data de Emissão: 20/07/2011
 Autor: gerente
 Aprovador: gerente

Gerais

Glossário: nada a acrescentar
 Procedimento de Mudança: nenhum

 **Introdução LTP07**

Identificação do Documento: LTP07
 Data de Emissão: 15:57:02-20/07/11
 Autor: analista
 Aprovador: analista

Gerais LTP07

Glossário: -
 Procedimento de Mudança: -

Figura 25 – Relatório de teste mestre

3.4. RESULTADOS E DISCUSSÃO

O principal objetivo desse trabalho foi criar uma ferramenta *web* que viesse auxiliar o planejamento dos testes de uma maneira mais organizada, de forma homogênea e amigável. Por ser uma ferramenta baseada na norma IEEE 829-2008, ela mantém os padrões elencados pela norma fazendo com que os documentos de testes sejam formatados de uma maneira de fácil legibilidade. Essa ferramenta também pode ser utilizada na substituição de planilhas, usadas para a criação dos planos e casos de teste.

Os papéis envolvidos no processo foram especificados de acordo com a norma IEEE 829-2008, a que define analistas que fazem o planejamento dos testes e os testadores que são responsáveis por executar os testes e registrar os resultados.

Em relação aos trabalhos correlatos, conclui-se que a ferramenta TestLink é o projeto que mais se aproxima do trabalho apresentado. No entanto, este difere em relação ao presente

trabalho na forma e administrações dos objetos citados pela norma IEEE 829-2008, criando uma forma específica de trabalho em relação ao plano de testes. Essa ferramenta também pode ser utilizada em empresas de desenvolvimento de software. Além disso, embora ambos possuam desenvolvimento voltado para *web*, o presente trabalho utiliza JSP como linguagem enquanto a ferramenta TestLink foi desenvolvido em PHP. Nas duas ferramentas – tanto a proposta neste trabalho quanto o TestLink – é utilizado o banco de dados MySQL para armazenamento de dados e é a que mais se aproxima do trabalho desenvolvido. No Quadro 4 é apresentado um comparativo entre os trabalhos correlatos estudados e a ferramenta desenvolvida.

Quesito	Ferramentas			
	TestLink	Sander (2002)	TestPlan	WebQualy
Plataforma	Web	Dataflex	Web	Web
Linguagem	PHP	Dataflex	C# ASP.NET	JSP
Banco de dados	MySQL	Dataflex (Banco proprietário)	SQLServer 2005	MySQL
Norma/Processo base	IEEE 829 1998	ISO/IEC 12207	Processo HBSIS + CMMI	IEEE 829 2008
Ferramentas integradas	Bugzilla e Mantis	Não atende	Enterprise Architect	Não atende
Possui registro de plano de teste	Sim	Sim	Sim	Sim
Possui registro de caso de teste	Sim	Sim	Sim	Sim
Possui registro de caso de uso	Sim	Não atende	Atende parcialmente	Atende parcialmente
Possui relatório de resultado de teste	Não atende	Sim	Não atende	Atende parcialmente
Possui relatório de erros / anomalias	Sim	Não atende	Sim	Sim

Quadro 4 – Quadro comparativo entre os trabalhos correlatos e a ferramenta desenvolvida

As ferramentas se diferenciam pela maneira em que os dados são apresentados e também nas normas/processos bases utilizados. Apenas a ferramenta de Sander não é para ambiente web e possui banco proprietário. A ferramenta Testlink possui integração com o Bugzilla e o Mantis. Já o TestPlan possui integração com o Enterprise Architect, enquanto que as ferramentas de Sander e WebQualy não possuem integração com outras ferramentas. Todas as ferramentas estudadas e o trabalho realizado possuem registros de planos e casos de teste. As ferramentas TestLink e TestPlan não possuem relatório de resultado de teste e apenas a ferramenta de Sander não possui relatório de erros/anomalias encontradas.

4. CONCLUSÕES

O objetivo do sistema é o gerenciamento do suporte ao planejamento de testes e resultados. Utilizando a plataforma *web*, a ferramenta pode ser acessada de qualquer ponto onde há uma conexão com a internet. Ela pode ser utilizada para realizar testes de integrações entre módulos de um sistema. Também pode realizar testes com análises de requisitos, criando um plano de teste mestre e novos níveis de planos.

Para a definição da ferramenta levou-se em consideração alguns estudos sobre o panorama de testes de *softwares* em algumas empresas, tendo os casos de testes administrados por planilhas. Constatou-se então a necessidade de criar uma ferramenta simples para auxiliar as pessoas a realizarem criações de testes mais facilmente.

Para que a ferramenta fosse completa e que também fosse possível registrar informações relevantes dos testes, adotou-se a norma IEEE 829-2008.

Com a maturidade da idéia foi desenvolvida uma ferramenta para facilitar as atividades do planejamento de teste. A ferramenta pode ser utilizada para ganhar agilidade no planejamento de testes, organização das informações e, principalmente, padronização dos relatórios dos objetos de teste.

Por fim, o trabalho atingiu os objetivos propostos agregando conhecimento na área de teste de software e no desenvolvimento de soluções *web*. Vale ressaltar que a ferramenta, por utilizar a padronização da norma, pode ser utilizada para realizar testes em várias áreas.

4.1. EXTENSÕES

Esse trabalho apresenta algumas funções para a criação de plano de testes via *web*. Desta forma, existem outras funcionalidades que podem ser adicionadas ou aperfeiçoadas. Ente elas destacam-se:

- a) geração de relatórios em PDF;
- b) geração de indicadores para exibir aos participantes do projeto qual o é tipo de teste mais utilizado ou qual teste levou mais tempo;
- c) aperfeiçoamento da interface de plano e caso de testes deixando-a mais dinâmica;

- d) análise de desempenho de teste realizados por testador, em horas e quantidade de testes;
- e) aperfeiçoamento do nível de plano de teste para quatro documentos distintos: plano de teste de aceitação, plano de teste de sistema, plano de teste de integração de componente e plano de teste de componente;
- f) aperfeiçoar o relatório de nível de plano de teste para quatro relatórios distintos: aceite de testes, teste de sistemas, integração de componentes e componentes de teste.

REFERÊNCIAS BIBLIOGRÁFICAS

BARTIÉ, Alexandre. **Garantia da qualidade de software: adquirindo maturidade organizacional**. Rio de Janeiro: Campus, 2002.

BASTOS, Aderson et al. **Base de conhecimento em teste de software**. 2. ed. São Paulo: Martins, 2007.

DELAMARO, Márcio E.; MALDONADO, José C.; JINO, Mario. **Introdução ao teste de software**. Rio de Janeiro: Campus, 2007

INTHURN, Cândida. **Qualidade & teste de software**. Florianópolis: Visual Books, 2001.

INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS. **IEEE Std 829**: Standard for software test documentation. New York: IEEE Computer Society, 2008.

_____. **IEEE Std 829**: standard for software test documentation. New York: IEEE Computer Society, 1998.

IEEE. **Standard for Software Test Documentation**. [S.I.], 2010a. Disponível em: <<http://standards.ieee.org/findstds/standard/829-1983.html>>. Acesso em: 17 jun. 2011.

_____. Standard for Software Test Documentation. [S.I.], 2010b. Disponível em: <<http://standards.ieee.org/findstds/standard/829-1998.html>>. Acesso em: 17 jun. 2011.

_____. Standard for Software Test Documentation. [S.I.], 2010c. Disponível em: <<http://standards.ieee.org/findstds/standard/829-2008.html>>. Acesso em: 17 jun. 2011.

ISO/IEC 12207, **Tecnologia da Informação – Processos de ciclo de vida de software**, ABNT – Associação Brasileira de Normas Técnicas, Rio de Janeiro: ABNT, 1998.

KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de software**: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. São Paulo: Novatec, 2006.

LABES, Camila. **Test-Plan: Ferramenta de apoio ao planejamento e controle das atividades do processo de testes**. 2010. 107f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MOLINARI, Leonardo. **Testes Funcionais de Software**. Florianópolis: Visual Books, 2008.

MYERS, Glenford J. **The art of software testing**. New York: John Wiley & Sons, 1979

RIOS, Emerson; MOREIRA FILHO, Trayahú. **Testes de software**. 2 ed. Rio de Janeiro: Alta Books, 2006.

SANDER, Izabela. **Sistema de gerenciamento do teste de software baseado na norma ISO/IEC 12207**. 2002. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

TESTLINK. **What is Testlink?**. [S.l.], 2006. Disponível em: <<http://www.testlink.org>>. Acesso em: 14 set. 2009.

APÊNDICE A – Detalhamento dos casos de uso

É apresentado o detalhamento dos principais casos de uso previstos nos diagramas apresentados na seção 3.2.1.

No Quadro 5 apresenta-se o caso de uso “UC01 – Manter usuários e perfis”.

Nome do Caso de Uso	Manter usuários e perfis
Descrição	O usuário do tipo “Admin” acessa o menu “Usuário” para realizar o gerenciamento de u usuário.
Ator	Usuário admin
Pré-condição	Deve haver um usuário tipo “Admin” cadastrado no sistema. O usuário deverá estar ativo.
Fluxo principal	<ol style="list-style-type: none"> 1. O usuário admin seleciona a opção “Criação de usuários”. 2. A tela para a criação do usuário é exibida. 3. O usuário admin informa o login, senha, e-mail e a permissão do novo usuário. 4. O usuário admin grava o novo usuário.
Cenário – Visualização	<ol style="list-style-type: none"> 5. O usuário criado é exibido na tela.
Cenário – Edição	<ol style="list-style-type: none"> 6. No passo 1 o usuário admin clica na opção “Edição de Usuários”. 7. O sistema exibe os campos do passo 3 para edição. 8. O usuário admin altera os dados necessários. 9. O usuário admin grava as alterações realizadas.
Cenário – Exclusão	<ol style="list-style-type: none"> 10. No passo 1 o usuário admin clica na opção “Exclusão de Usuário” 11. O usuário admin seleciona o usuário a ser excluído. 12. O sistema realiza a exclusão do usuário.
Pós-condição	Criar o usuário.

Quadro 5 – Descrição do caso de uso manter usuários e perfis

No Quadro 6 apresenta-se o caso de uso “UC02 – Manter Plano de Teste”.

Nome do Caso de Uso	Manter plano de testes
Descrição	O analista de testes acessa o menu “Plano de testes” para realizar o gerenciamento de um plano de testes.
Ator	Analista de Testes
Pré-condição	O analista deve estar ativo no sistema.
Fluxo principal	<ol style="list-style-type: none"> 1. O analista seleciona o MTP. 2. O analista insere um novo plano de testes. 3. O sistema exibe a tela de cadastro de plano de testes. 4. O analista informa os dados do plano de testes. 5. O analista seleciona os responsáveis pela execução dos testes. 6. O analista confirma o cadastro. 7. O sistema grava o novo plano de testes.
Cenário - Visualização	<ol style="list-style-type: none"> 8. O sistema exibe o código e nome do plano de testes cadastrado.
Cenário – Edição	<ol style="list-style-type: none"> 9. No passo 2 o analista seleciona um plano de testes para edição. 10. O sistema exibe os dados para edição. 11. O analista altera os dados necessários. 12. O analista grava o plano de testes.
Cenário - Exclusão	<ol style="list-style-type: none"> 13. No passo 2 o sistema exibe os planos de testes. 14. O analista seleciona o plano de teste a ser excluído. 15. O sistema verifica se existe caso de testes concluído. 16. Se existir, o sistema exibe mensagem impedindo a exclusão. 17. Caso não exista, o sistema realiza a exclusão.
Pós-condição	Criado o plano de teste.

Quadro 6 – Descrição do caso de uso manter plano de testes

No Quadro 7 apresenta-se o caso de uso “UC03 – Manter Casos de Teste”.

Nome do Caso de Uso	Manter caso de testes
Descrição	O analista de testes acessa o menu “Casos de teste” para realizar a manutenção de um caso de testes.
Ator	Analista de Testes
Pré-condição	O analista deve estar ativo no sistema. Deve ter pelo menos um plano de testes cadastrado no sistema.
Fluxo principal	<ol style="list-style-type: none"> 1. O analista seleciona o plano de testes. 2. O analista insere um novo caso de teste. 3. O sistema exibe a tela de cadastro de caso de teste. 4. O sistema exibe os dados para inserção; 5. O analista informa os dados do passo anterior. 6. O analista grava o novo caso de testes. 7. O sistema grava o caso de testes.
Pós-condição	Criar o caso de teste.

Quadro 7 - Descrição do caso de uso gerenciar caso de testes

No Quadro 8 apresenta-se o caso de uso “UC05 – Emitir relatório de anomalia”.

Nome do Caso de Uso	Emitir relatório de anomalia
Descrição	O analista de testes acessa o menu “Relatórios” para emitir o relatório de anomalias.
Ator	Analista de Testes
Pré-condição	O analista deve estar ativo no sistema. Deve ter pelo menos um plano de testes, um caso de testes e um resultado cadastrado no sistema e associados.
Fluxo principal	<ol style="list-style-type: none"> 1. O analista seleciona o caso de teste que deseja visualizar os resultados. 2. O analista clica no botão “Gerar Relatório”. 3. O sistema exibe o relatório com os dados para visualização.
Pós-condição	Emitir o relatório de anomalia.

Quadro 8 – Descrição do caso de uso emitir relatório de anomalia

APÊNDICE B – Dicionário de dados do modelo entidade-relacionamento

Os quadros a seguir apresentam as tabelas do sistema, conforme definido no MER, na seção 3.2.2.

Tabela Permissao		
Coluna	Tipo	Descrição
IDPER	INTEGER (<i>Primary Key</i>)	Número seqüencial da identificação da permissão
PERMISSÃO	VARCHAR	Tipo de permissão do usuário

Quadro 9 – Tabela Permissão

Tabela Usuario		
Coluna	Tipo	Descrição
IDUSU	INTEGER (<i>Primary Key</i>)	Número seqüencial da identificação do usuário
PERMISSAO_IDPER	INTEGER (<i>Foreign Key</i>)	Número da identificação da permissão do usuário
SENHA	VARCHAR	Senha do usuário
LOGIN	VARCHAR	Login de acesso do usuário
EMAIL	VARCHAR	E-mail do usuário

Quadro 10 – Tabela Usuário

Tabela Aprovadores		
Coluna	Tipo	Descrição
DOCUMENTO_DOCID	INTEGER (<i>Foreign Key</i>)	Número seqüencial da identificação do documento.
USUARIO_IDUSU	INTEGER (<i>Foreign Key</i>)	Número seqüencial d identificação do usuário, da tabela Usuario
IDAPROVADORES	INTEGER (<i>Primary Key</i>)	Número seqüencial da identificação dos aprovadores.

Quadro 11 – Tabela Aprovadores

Tabela Ligacoes		
Coluna	Tipo	Descrição
IDLIG	INTEGER (<i>Primary Key</i>)	Número seqüencial da ligação
DOCUMENTO_DOCID	INTEGER (<i>Foreign Key</i>)	Número seqüencial do documento de ligação 1
DOCUMENTO_DOCID2	INTEGER (<i>Foreign Key</i>)	Número seqüencial do documento de ligação 2

Quadro 12 – Tabela Ligacoes

Tabela Documento		
Coluna	Tipo	Descrição
DOCID	INTEGER (<i>Primary Key</i>)	Número seqüencial do documento
USUARIO_IDUSU	INTEGER (<i>Foreign Key</i>)	Número da identificação do usuário
DT_EMISSAO	VARCHAR	Data de emissão do documento
ORGANIZ_EMISSAO	VARCHAR	Organização emissora do documento
STAT	VARCHAR	Status do documento
CATEGORIA	VARCHAR	Categoria do documento
MUDANCA	VARCHAR	Procedimentos de mudanças do documento
GLOSSARIO	VARCHAR	Glossário do documento
ESCOPO	VARCHAR	Escopo do documento
DESC_DOC	VARCHAR	Descrição do documento

Quadro 13 – Tabela Documento

Tabela ResultadoTeste		
Coluna	Tipo	Descrição
IDRESULTADOTESTE	INTEGER (<i>Primary Key</i>)	Número seqüencial da identificação do resultado de teste
DOCUMENTO_DOCID	INTEGER (<i>Foreign Key</i>)	Número do documento relacionado ao resultado de teste
TESTE	VARCHAR	Descrição do teste realizado ao encontrar a anomalia
INCIDENTE	VARCHAR	Indica se o resultado teve incidente
ESCOPO	VARCHAR	Escopo do resultado de teste
REFERENCIAS	VARCHAR	Referências do resultado de teste
DESCRICA0	VARCHAR	Descrição da anomalia encontrada
ATIVEVEENTRADAS	VARCHAR	Atividades de entradas cuja anomalia foi encontrada
RESUMOANOMALIA	VARCHAR	Resumo da anomalia
DT_ANOMALIA	VARCHAR	Data da anomalia encontrada
IMPACTO	VARCHAR	Impacto da anomalia
AVALIACAO	VARCHAR	Avaliação da anomalia
DESC_CORRETIVA	VARCHAR	Descrição corretiva da anomalia
STAT_RESU	VARCHAR	Status do resultado da anomalia
CONCLUSAO	VARCHAR	Conclusões e recomendações da anomalia

PROC_MUDANCA	VARCHAR	Procedimentos de mudanças sobre a anomalia encontrada
MUD_PLANO	VARCHAR	Procedimento de mudança de plano em relação à anomalia
DATA_RESU	VARCHAR	Data do resultado
RACIONALIDADE	VARCHAR	Racionalidades do resultado de teste

Quadro 14 – Tabela Resultados de teste

Tabela PlanoTeste		
Coluna	Tipo	Descrição
IDPLANOTESTE	INTEGER (<i>Primary Key</i>)	Número sequencial do plano
DOCUMENTO_DOCID	INTEGER (<i>Foreign Key</i>)	Número do sequencial do documento relacionado
FEATURESTESTAR	VARCHAR	Características a serem testadas
FERRAMENTAS	VARCHAR	Ferramentas a serem utilizadas nos testes
CRITERIO	VARCHAR	Critério de suspensão e retomada
TAREFAS	VARCHAR	Tarefas a serem testadas
REFERENCIAS	VARCHAR	Documentos referentes
CONTEXTO	VARCHAR	Contexto do plano de teste
NOTACAO	VARCHAR	Notação do plano de teste
NIVEL_ABRANGENCIA	VARCHAR	Nível de abrangência do plano de teste
ABORDAGEM	VARCHAR	Abordagem geral do nível de teste, identificando principais tarefas e estimativas
APROVACAO_CRITERIO	VARCHAR	Critério de aprovação do plano de teste
ATIVIDADE_TAREFA	VARCHAR	Atividades e/ou tarefas relacionadas ao plano
RESPONSABILIDADES	VARCHAR	Pessoal responsável pelo plano
INTERFACES	VARCHAR	Interfaces entre as partes envolvidas
RECURSOS	VARCHAR	Recursos que serão testados
FORMACAO	VARCHAR	Cursos e formações para o plano de teste
HORARIOS	VARCHAR	Horários e estimativas de custos
REQUISITOS_PROCESSUAL	VARCHAR	Requisito processual especial do procedimento de teste

DEPENDENCIA	VARCHAR	Dependências do plano de teste
CARAC_CHAVES	VARCHAR	Características chaves
INTEGRIDADE	INTEGER	Nível de integridade do plano de teste
METODO	VARCHAR	Métodos a serem aplicados no plano
NORMA	VARCHAR	Normas aplicadas ao plano
LTP	VARCHAR	Indica se o plano deve gera um LTP
LTD	VARCHAR	Indica se o plano deve entregar um LTD
LTC	VARCHAR	Indica se o plano deve entregar um LTC
LTPR	VARCHAR	Indica se o plano deve entregar um LTPR
AR	VARCHAR	Indica se o plano deve entregar um AR
SR	VARCHAR	Indica se o plano deve entregar um SR
ACP	VARCHAR	Indica se o plano deve entregar um ACP
MTP	VARCAR	Indica se o plano deve entregar um MTR
PROCEDIMENTO	VARCHAR	Política de procedimento da qualidade dos processos
METRICA	VARCHAR	Métricas a serem coletadas e analisadas
EVENTO	VARCHAR	Eventos do plano
IDANOMALIA	VARCHAR	Número seqüencial da anomalia

Quadro 15 – Tabela PlanoTeste

Tabela CasoTestePlano		
Coluna	Tipo	Descrição
IDCTXPT	INTEGER (<i>Primary Key</i>)	Número seqüencial do caso plano
CASOTESTE_IDCASO	INTEGER (<i>Foreign Key</i>)	Número seqüencial do caso de teste relacionado
PLANOTESTE_IDPLANOTESTE	INTEGER (<i>Foreign Key</i>)	Número seqüencial do plano de teste relacionado

Quadro 16 – Tabela Caso de Testes X Planos de Teste

Tabela CasoTeste		
Coluna	Tipo	Descrição
IDCASO	INTEGER (<i>Primary Key</i>)	Número seqüencial do caso de teste
CASOUSO_IDCASO	INTEGER (<i>Foreign Key</i>)	Número seqüencial do caso de uso relacionado
ESPENTRADAS	VARCHAR	Entradas esperadas
ESPSAIDAS	VARCHAR	Saídas esperadas
AMBSW	VARCHAR	Necessidades de ambiente – software
AMBHW	VARCHAR	Necessidades de ambiente – hardware
AMBOUT	VARCHAR	Necessidades de ambiente – Outros
REQESP	VAHCHAR	Requisitos especiais

Quadro 17 – Tabela Caso de teste

Tabela CasoUso		
Coluna	Tipo	Descrição
IDCASO	INTEGER	Número seqüencial do caso de uso
CODCASO	VARCHAR	Código do caso de uso
NOMCASO	VARCHAR	Nome do caso de uso
DESCCASO	VARCHAR	Descrição do caso de uso

Quadro 18 – Tabela Caso de uso