

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

DESENVOLVIMENTO DE FERRAMENTA
PARA GERENCIAMENTO DE ARTEFATOS EM PROJETO
DE MELHORIA NO PROCESSO DE TESTES

BRUNA HERKENHOF

BLUMENAU
2011

2011/2-08

BRUNA HERKENHOF

**DESENVOLVIMENTO DE FERRAMENTA
PARA GERENCIAMENTO DE ARTEFATOS EM PROJETO
DE MELHORIA NO PROCESSO DE TESTES**

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 – Orientador

**BLUMENAU
2011**

2011/2-08

**DESENVOLVIMENTO DE FERRAMENTA PARA
GERENCIAMENTO DE ARTEFATOS EM PROJETO DE
MELHORIA NO PROCESSO DE TESTES**

Por

BRUNA HERKENHOF

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. Marcel Hugo, Mestre – FURB

Blumenau, 06 de dezembro de 2011.

Dedico este trabalho a toda minha família e a todos os amigos, especialmente aqueles que me ajudaram diretamente na realização deste.

AGRADECIMENTOS

A Deus, que permite a existência e força para viver, por estar sempre ao meu lado, iluminando meu caminho.

Aos professores, pela paciência e dedicação e principalmente por compartilharem conosco seus ricos conhecimentos, em especial ao meu orientador, professor Jacques Robert Heckmann pela disponibilidade e auxílio.

À minha mãe Doralice Maria Herkenhoff, pelo carinho e afeto, por confiar em mim e na minha capacidade.

Ao meu namorado Eduardo pela paciência, atenção e compreensão, agradeço pelo colo, aprendizado, carinho e por ser um bom ouvinte.

Às verdadeiras amizades desenvolvidas ao longo da faculdade por todos bons momentos vividos juntos, aos amigos Rangel, Renan e amiga em especial Bruna pelas boas conversas, risadas e palavras de conforto.

Há muitos a quem agradecer, pela colaboração e convívio nesta trajetória. A todos que contribuíram com seu apoio, críticas, estímulo, carinho, amizade e amor.

Muito obrigada!

É impossível haver progresso sem haver mudança, e quem não consegue mudar a si mesmo não muda coisa alguma.

George Bernard Shaw

RESUMO

Este trabalho apresenta uma ferramenta *web* para gerenciar os principais artefatos presentes em projeto específico aplicado à empresa Senior, o Projeto de Melhoria no Processo de Testes (PMPT). A ferramenta foi desenvolvida através do *Genexus* (ferramenta de desenvolvimento) gerando código em Java, utilizando o banco de dados SQL Server 2008 e o servidor *web* Java Apache Tomcat. A ferramenta desenvolvida facilita o gerenciamento dos principais artefatos do PMPT. Como resultado destaca-se o aumento do controle sobre a rastreabilidade dos requisitos, possibilitando evitar o esquecimento da manutenção dos artefatos relacionados, garantindo assim uma alteração precisa, segura e organizada dos artefatos envolvidos aos requisitos.

Palavras-chave: Processo de Melhoria. Processos de Testes. Ferramentas CASE.

ABSTRACT

This work introduces a web tool to manage the main artifacts present in a specific project applied to Senior Systems: Test Process Improvement Project (PMPT). The tool was developed by Genexus (a development tool), generating Java code, using the SQL Server 2008 database and the web Java Apache Tomcat Server. The developed tool eases the main PMPT artifacts management. As a result, it emphasizes the control increase in the requirements traceability, allowing to avoid the associated artifacts maintenance forgetfulness, thus ensuring a needful, reliable and organized change of the involved artifacts to the requirements.

Keywords: Improvement Process. Tests process. CASE tools.

LISTA DE FIGURAS

Figura 1 – Processo de testes	6
Figura 2 – Visões dos testes por perspectivas diferentes.....	9
Figura 3 – Diferentes níveis de abstração ao longo do processo de desenvolvimento.....	10
Figura 4 - Exemplo de Mapa de Suítes de Apuração	14
Figura 5 – Exemplo de planilha de Suíte de Testes.....	15
Figura 6 – Exemplo de uma planilha de Caso de Testes	16
Figura 7– Exemplo de planilha de Massa de Dados (<i>template</i>)	16
Figura 8– Exemplo de planilha de Requisitos	17
Figura 9– Imagem da tela principal da ferramenta.....	18
Figura 10 – Diagrama de Casos de Uso	23
Figura 11 – Diagrama de Atividades para Implantação	24
Figura 12 – Diagrama de Atividades	25
Figura 13 – Modelos de Dados Relacional.....	26
Figura 14 – Vínculo de artefatos na importação de planilha de requisitos.....	28
Figura 15 – Exportação de casos de testes para planilha <i>Excel</i>	29
Figura 16 – Tela de <i>Login</i>	30
Figura 17 – Menu Principal Analista de Qualidade	31
Figura 18 – Tela de Cadastro de Módulo	31
Figura 19 – Tela de Importar Planilhas – Suíte de Teste.....	32
Figura 20 – Tela de Importar Planilhas - Requisito.....	33
Figura 21 – Tela de Visualização de Vínculos de Artefatos	33
Figura 22 – Tela de Cadastro de Rotina	34
Figura 23 – Tela de Cadastro de Grupo.....	34
Figura 24 – Tela de Cadastro de Sub Grupo	35
Figura 25 – Tela de Cadastro de Item	35
Figura 26– Tela de Cadastro de Requisitos	36
Figura 27 – Tela de Cadastro de Suíte de Teste	37
Figura 28 – Tela de Cadastro de Caso de Teste	37
Figura 29 – Tela de Cadastro de Situação de Teste.....	38
Figura 30 – Tela de Cadastro de Vínculo de Artefatos	38
Figura 31 – Relatório de Requisitos por Data	39

Figura 32 – Relatório de Vínculo Requisito X Caso de Teste	39
--	----

LISTA DE QUADROS

Quadro 1 – Requisitos Funcionais.....	21
Quadro 2 – Requisitos Não Funcionais	22
Quadro 3 - Descrição do caso de uso <i>Logon</i> no sistema	45
Quadro 4– Descrição do caso de uso Manter Usuários.....	46
Quadro 5– Descrição do caso de uso Importar Requisito	46
Quadro 6 – Descrição do caso de uso Importar Suíte de Testes.....	46
Quadro 7 – Descrição do caso de uso Manter Requisitos.	47
Quadro 8 – Descrição do caso de uso Manter Grupos.	48
Quadro 9 – Descrição do caso de uso manter Sub Grupos.....	49
Quadro 10 – Descrição do caso de uso Manter Item.....	49
Quadro 11 - Descrição de caso de uso Manter Módulos.....	50
Quadro 12 – Descrição do caso de uso Manter Rotinas	51
Quadro 13 – Descrição do caso de uso Manter Suítes de Testes	51
Quadro 14 – Descrição do caso de uso Manter Casos de Testes.....	52
Quadro 15 – Descrição do caso de uso Manter Situações de Testes.....	53
Quadro 16 – Descrição do caso de uso Manter Vínculos entre artefatos.....	54
Quadro 17 – Descrição do caso de uso Exportar Casos de Testes.....	54
Quadro 18 – Descrição do caso de uso Emitir relatório de Requisitos Cadastrados.....	54
Quadro 19 – Descrição do caso de uso Emitir relatório de Requisitos por Data.	54
Quadro 20 – Descrição do caso de uso Emitir relatório de Vínculos entre Requisitos e Casos de Testes.	55
Quadro 22 – Descrição do caso de uso Emitir relatório de Requisitos por Rotina.	55
Quadro 23 – Descrição do caso de uso Emitir relatório de Massa de Dados pertencentes a Casos de Testes.....	55
Quadro 24 – Tabela Caso Teste.....	56
Quadro 25 – Tabela Grupo	56
Quadro 26 – Tabela GrupoModulo	56
Quadro 27 – Tabela Item	56
Quadro 28 – Tabela Módulo.....	56
Quadro 29 – Tabela Requisito	57
Quadro 30 – Tabela Requisito Caso Teste	57

Quadro 31 – Tabela Rotina.....	57
Quadro 32 – Tabela Situação Teste.....	57
Quadro 33 - Tabela Situação Teste Passos.....	58
Quadro 34 – Tabela Sub grupo.....	58
Quadro 35 – Tabela Suíte.....	58
Quadro 36 – Tabela Usuário.....	58

LISTA DE TABELAS

Tabela 1 - Relação entre os estágios de teste e os paradigmas procedimental e OO	12
--	----

LISTA DE SIGLAS

BSI – Curso de Sistemas de Informação – Bacharelado

DSC – Departamento de Sistemas e Computação

EA – *Enterprise Architect*

MER – Modelo Entidade Relacionamento

MPS.BR – Melhoria de Processos do Software Brasileiro

MPT.BR – Melhoria de Processo de Teste Brasileiro

OO – Orientado a Objetos

PCP – Planejamento Controle da Produção

PMPT – Projeto de Melhoria no Processo de Testes

SGBD – Sistema Gerenciador de Banco de Dados

TI – Tecnologia da Informação

TIGN – Tecnologia da Informação na Gestão de Negócios

UML – *Unified Modeling Language*

V&V - Validação e Verificação

SUMÁRIO

1 INTRODUÇÃO.....	2
1.1 OBJETIVOS DO TRABALHO	2
1.2 ESTRUTURA DO TRABALHO	3
2 FUNDAMENTAÇÃO TEÓRICA	4
2.1 QUALIDADE DE SOFTWARE.....	4
2.1.1 Processo de testes	5
2.1.2 Processo de testes automatizados	6
2.1.3 Artefatos de Testes	7
2.2 O PAPEL DO TESTE NA QUALIDADE.....	8
2.3 SISTEMA ATUAL	13
2.7 TRABALHOS CORRELATOS	18
3 DESENVOLVIMENTO	20
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	20
3.2 ESPECIFICAÇÃO	22
3.2.1 Diagrama de casos de uso.....	22
3.2.2 Diagrama de atividades	24
3.2.3 Modelo de dados relacional.....	25
3.3 IMPLEMENTAÇÃO	26
3.3.1 Técnicas e ferramentas utilizadas	26
3.3.1.1 SQL Server 2008	26
3.3.1.2 Ferramenta Genexus	27
3.3.1.3 Apache Tomcat 5.5.....	29
3.3.2 Operacionalidade da implementação.....	30
3.3.2.1 Logar no sistema.....	30
3.4 RESULTADOS E DISCUSSÃO	40
4 CONCLUSÕES.....	41
4.1 EXTENSÕES.....	42
REFERÊNCIAS BIBLIOGRÁFICAS	43
APÊNDICE A – Descrição dos casos de uso	45
APÊNDICE B – Dicionário de Dados	56

1 INTRODUÇÃO

A Senior S.A. completou 23 anos de existência em 2011. Contando com aproximadamente 700 colaboradores, 9500 clientes, que utilizam principalmente o software de gestão Rubi® (Administração de Pessoal) e o Ronda® Ponto Eletrônico, hoje incorporado no sistema de Gestão de Pessoas da Senior. A empresa visa sempre a qualidade de seus processos e principalmente de seus produtos, procurando assim garantir a fidelidade e a qualidade para seus clientes.

Pensando nisto a empresa adotou o Projeto de Melhoria no Processo de Testes (PMPT). Apesar de parecer ter relação com o modelo de Melhoria de Processo de Teste Brasileiro (MPT.BR¹), por utilizar os mesmos termos nominais e abordar o mesmo assunto, é importante salientar que o PMPT da empresa Senior não possui nenhuma aderência ao MPT.BR.

O PMPT foi concebido e adotado após uma análise da empresa Verothi Consultoria, a qual avaliou o processo de qualidade da Senior e sugeriu este projeto para melhorar o processo de testes, melhorando assim consequentemente todo o ciclo de qualidade dos produtos Senior.

Este trabalho apresenta uma ferramenta para auxiliar no gerenciamento dos principais artefatos utilizados no PMPT, pois atualmente eles são mantidos em planilhas *Excel* dificultando o processo de manutenção e organização.

1.1 OBJETIVOS DO TRABALHO

O objetivo do trabalho foi desenvolver uma ferramenta para gerenciar os artefatos utilizados no PMPT, sendo eles os requisitos, suítes de testes, casos de testes e massas de dados (*templates*).

¹ Como o próprio nome diz, o MPT.BR é um modelo para Melhoria de Processo de Teste Brasileiro, que está sendo desenvolvido com o princípio básico de ser compatível com o modelo de Melhoria de Processos do Software Brasileiro (MPS.BR). O MPT.BR é voltado para a melhoria das áreas de teste de software de empresas de qualquer porte. É leve e passível de ser adotado por áreas de teste de software para apurar o seu nível de maturidade, sem, com isso, onerar os seus processos anteriormente implementados. (MPT.BR, 2010).

Os objetivos específicos do trabalho são:

- a) manter os requisitos cadastrados na ferramenta, efetuando a importação daqueles já existentes nas planilhas para o ambiente da ferramenta;
- b) controlar a rastreabilidade dos artefatos para as possíveis manutenções, permitindo consultar quais casos de teste e massas de dados (*templates*) devem ser alterados para a correta manutenção de um determinado requisito.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos.

O primeiro capítulo apresenta uma introdução ao trabalho, seus objetivos e sua estrutura.

O segundo capítulo contempla os conceitos dos principais fundamentos que servem de base para o trabalho, a descrição do processo atualmente praticado na empresa, e por fim, os trabalhos correlatos.

O terceiro capítulo, por sua vez, trata do desenvolvimento da ferramenta de gerenciamento de artefatos, bem como suas especificações, modelo de entidade e relacionamento (MER) e diagramas de casos de uso e de atividades, juntamente com a implementação e a ferramenta desenvolvida.

O quarto e último capítulo, apresenta as conclusões do trabalho, suas limitações e possíveis extensões para o mesmo.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica necessária para compreensão deste trabalho. São abordados assuntos relacionados à importância da qualidade de software, as tecnologias que serão abordadas e os trabalhos correlatos ao tema em questão. Por fim apresenta-se o procedimento atual do Projeto de Melhoria no Processo de Testes utilizado pela organização.

2.1 QUALIDADE DE SOFTWARE

A qualidade de software não é um destino, é uma jornada.

“Qualquer decisão tomada durante o processo de desenvolvimento do software pode comprometer sua qualidade final. Na verdade, o produto final do processo de desenvolvimento é exatamente o somatório de todas as decisões e realizações geradas durante todo o ciclo de desenvolvimento. Se desejarmos produzir software com alta qualidade, é necessário investir em qualidade em todos os pontos do processo[...]”. (BARTIÉ. 2002).

Com base nos conceitos de BARTIÉ (2002), quando se está diante de um desafio de garantir a qualidade de um software, estruturam-se processos que possuam mecanismos de inibição e impedimento de falhas, possibilitando desta forma que os diversos artefatos gerados durante o ciclo de desenvolvimento tenham procedimentos que avaliam sua qualidade, possibilitando a identificação prematura de defeitos nesses artefatos.

Se a organização deseja produzir softwares com qualidade, uma das principais providências a serem tomadas é estabelecer um processo mínimo de desenvolvimento. Não é possível garantir a qualidade de algo que não existe.

Já quando a abordagem é qualidade do produto, o autor define que o principal objetivo é garantir a qualidade do produto tecnológico gerado durante o ciclo de desenvolvimento. Todas as atividades que tenham por objetivo “estressar” telas e funcionalidades de um sistema informatizado podem ser categorizadas na dimensão da garantia da qualidade do produto tecnológico.

Segundo BARTIÉ (2002), o grau de deficiência dessas atividades nas organizações é assustadoramente baixo, o que incentiva as empresas a substituírem essas atividades pela correção de problemas. Os principais pontos para a constatação desses fatos são: falta de planejamento das atividades de testes, ausência de testes que validem funcionalidades antigas e ausência de um processo de automação dos testes e conferências.

Eis algumas definições sobre qualidade:

"Qualidade é uma característica intrínseca e multifacetada de um produto" (BASILI, 1996).

"Qualidade é a correção dos problemas e de suas causas ao longo de toda série de fatores relacionados com marketing, projetos, engenharia, produção e manutenção, que exercem influência sobre a satisfação do usuário" (FEIGENBAUM, 1994).

"Qualidade é a conformidade do produto às suas especificações" (CROSBY, 1984). As necessidades devem ser especificadas, e a qualidade é possível quando essas especificações são obedecidas sem ocorrência de defeito.

"Qualidade é tudo aquilo que melhora o produto no ponto de vista do cliente" (DEMING, 1990). Deming associa qualidade à impressão do cliente, portanto não é estática. A dificuldade em definir qualidade está na renovação das necessidades futuras do usuário em características mensuráveis, de forma que o produto possa ser projetado e modificado para dar satisfação por um preço que o usuário possa pagar.

"Qualidade é desenvolver, projetar, produzir e comercializar um produto de qualidade que é mais econômico, mais útil e sempre satisfatório para o consumidor" (ISHIKAWA, 1993).

"Qualidade não é uma fase do ciclo de desenvolvimento de software... ... é parte de todas as fases[...]" (BARTIÉ, 2002). Baseando-se nesta afirmação pode-se perceber que deve-se garantir a qualidade de todas as etapas de um processo, o que reforça um dos objetivos deste trabalho.

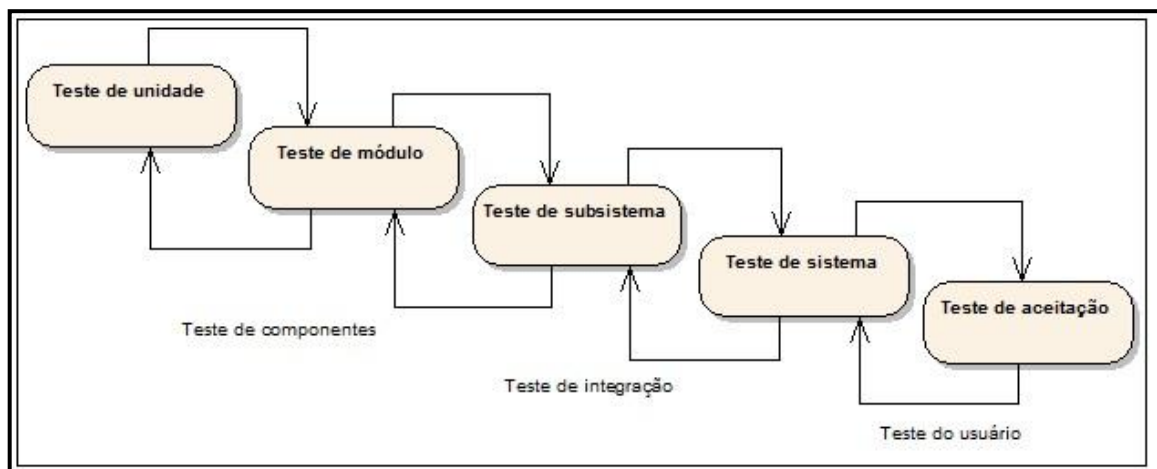
2.1.1 Processo de testes

Ao desenvolver um sistema, o teste geralmente envolve vários estágios (PFLEEGER, 2004). A literatura a respeito dos estágios do teste é ampla. Para este trabalho é adotada a nomenclatura definida por Sommerville (2003) que trata a realização dos testes incrementalmente, ou seja, em conjunto com a implementação do sistema.

Exceto para pequenos programas, os sistemas não devem ser testados como uma unidade isolada. Os grandes sistemas são constituídos a partir de subsistemas, que são construídos a partir de módulos, que são compostos por procedimentos e funções. O processo de teste deve, por conseguinte, evoluir em estágios (SOMMERVILLE, 2003):

- a) teste de unidade: são testados os componentes individuais;
- b) teste de módulo: são testados conjuntos mais amplos de procedimentos e funções, como por exemplo, uma coleção de componentes dependentes relacionados;
- c) teste de subsistema: são testados conjuntos de módulos que foram integrados em subsistemas;
- d) teste de sistema: são testadas as interações, os subsistemas são integrados para constituírem o sistema. Neste processo é verificado se os requisitos funcionais e não funcionais se cumprem e as propriedades emergentes do sistema;
- e) teste de aceitação: é testado o sistema com dados fornecidos pelo cliente, estágio final do processo, antes que o sistema seja aceito para uso operacional.

A Figura 1 apresenta um processo de testes composto de 5 estágios.



Fonte: Sommerville (2003).

Figura 1 – Processo de testes

2.1.2 Processo de testes automatizados

Recentemente com o avanço das técnicas de programação, principalmente pelo uso de ferramentas de desenvolvimento, houve um grande aumento da produtividade dos

desenvolvedores, o que resultou em uma quantidade cada vez maior de código a ser testado em um tempo cada vez menor, acabando por gerar atrasos na entrega de softwares devido ao grande tempo despendido em testes.

Em visão disto surgiram os testes automatizados que, de acordo com o conceito de BARTIÉ (2002), tratam-se da utilização de ferramentas de testes que possibilitam simular usuários ou atividades humanas de forma a não requerer procedimentos manuais no processo de execução de testes. Isto requer profissionais especializados e tempo no desenvolvimento da automação dos testes.

A automação de testes é altamente desejada por diversos fatores, inclusive em termos de custos finais. Como o processo de testes automatizados requer um investimento inicial, a automação passa a ser encarada como mais um trabalho a ser realizado. À medida que os testes são reexecutados, o ganho de tempo, controle, confiabilidade e as diversas possibilidades existentes com essa tecnologia ficam claras como vantagens inerentes a esse processo (BARTIÉ, 2002).

2.1.3 Artefatos de Testes

Um artefato é o resultado de uma atividade e pode ser utilizado posteriormente como matéria-prima para aquela ou para outra atividade a fim de gerar novos artefatos (SILVA, HUZITA, 2008).

Segundo McGregor e Sykes (2001), um processo de testes tem o foco voltado para a garantia de que as entradas do software produzam os resultados desejados de acordo com as especificações de requisitos. Desse processo pode resultar uma quantidade enorme de artefatos de testes utilizados ou gerados nas fases de planejamento, monitoramento, controle e execução de testes. Estes artefatos podem ser cenários de testes, casos de testes, *stubs*, drivers de testes e resultados dos testes.

Para o projeto PMPT, aplicado na empresa Senior, são utilizados os artefatos de requisitos de testes, suítes de testes, casos de testes e massa de dados (*templates*).

2.2 O PAPEL DO TESTE NA QUALIDADE

Validação e verificação (V&V) são processos de verificação e análise que asseguram que o software cumpra com suas especificações e atenda às necessidades dos clientes (SOMMERVILLE, 2003, p. 358). Embora sejam facilmente confundidas, V&V não são a mesma coisa. Pode-se resumir a diferença através de duas perguntas chaves:

- a) validação: estamos construindo o software certo;
- b) verificação: estamos construindo certo o software.

Validação é um conjunto de atividades que garante que o software construído corresponde aos requisitos do cliente, enquanto verificação é o conjunto de atividades que garante que o software implementa corretamente determinada função (PRESSMANN, 2002, p. 470).

Dentro da V&V pode-se utilizar duas técnicas (SOMMERVILLE, 2003, p. 358):

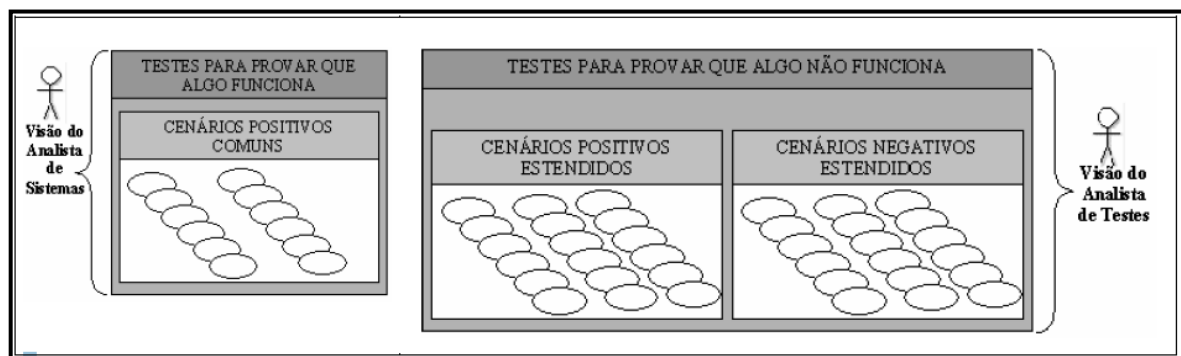
- a) inspeções de software: analisam e verificam as representações do sistema, como documento de requisitos, os diagramas do projeto e o código-fonte do programa. As inspeções podem ser aplicadas a todas as fases do processo de desenvolvimento. A inspeção é uma técnica estática, pois não requer que o software seja executado;
- b) teste de software: envolve a execução do software com dados de teste e a análise das saídas e do comportamento operacional, a fim de verificar se ele está sendo executado conforme esperado. O teste é uma técnica dinâmica, pois trabalha com uma representação executável do sistema.

O teste de software é um processo sistemático e planejado que tem por finalidade a identificação de erros (BARTIÉ, 2002, p. 22). O teste é um elemento importantíssimo da garantia da qualidade de software, pois evita que o cliente se frustre com um software cheio de problemas. As técnicas e metodologias de teste de software têm um papel importantíssimo no processo de teste, isto é, elas fornecem diretrizes sistemáticas para projetar testes que exercitam a lógica interna dos componentes de software e exercitam os domínios de entrada e saída do sistema para descobrir erros de funcionalidade, comportamento e desempenho (PRESSMAN, 2002, p. 429).

Myers (1979, p. 5), ao afirmar que a função do teste de software é mostrar a presença de erros (e não a ausência), contribuiu para que o teste de software tomasse um rumo diferente. Na prática, isto significa que mostrar que algo não funciona é mais difícil (exige

mais esforço) do que mostrar que funciona.

Desenvolvedores de software são, por natureza, pessoas construtivas. O teste de software, ao contrário, exige pessoas com natureza destrutiva. A Figura 2 ilustra visões de testes por perspectivas diferentes. A visão do analista de sistemas é, geralmente, provar que o software funciona, desta forma seus casos de testes serão limitados a cenários que provem que o sistema está funcionando. Já o analista de teste tem uma percepção mais ampla, isto é, além de testar as funcionalidades, seus casos de teste tentam mostrar falhas no software.

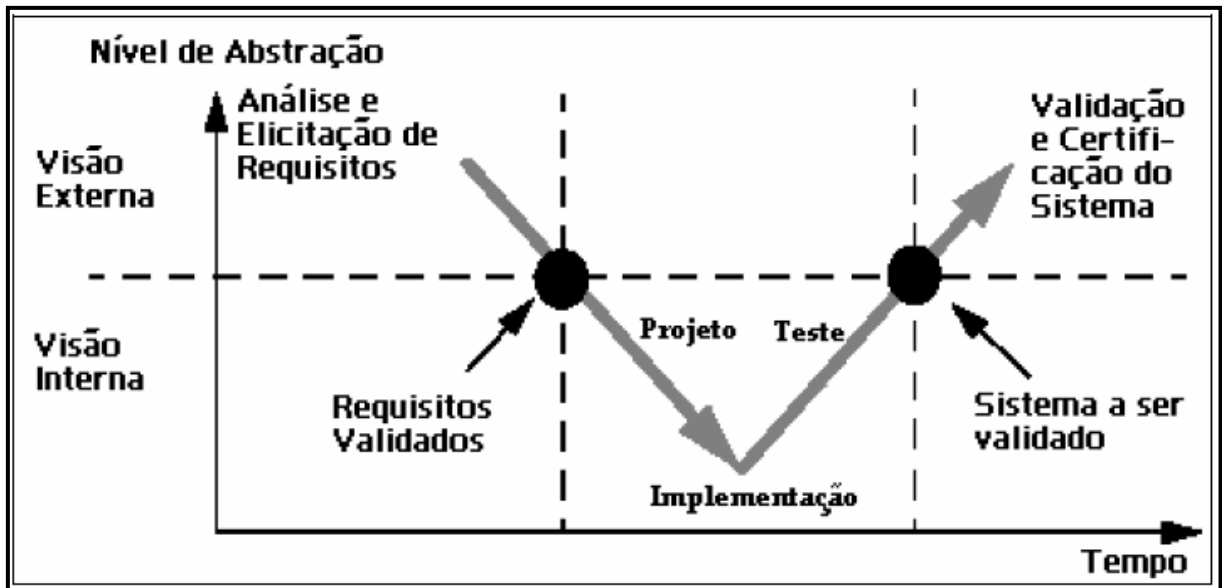


Fonte: Bartié (2002, p. 21).

Figura 2 – Visões dos testes por perspectivas diferentes

As atividades que envolvem a fase de teste são planejamento, projeto de casos de teste, execução e avaliação dos resultados. Elas devem ser conduzidas ao longo de todo o processo de desenvolvimento (ROCHA; MALDONADO; WEBER, 2001, p. 74). Pode-se salientar que a falta de atividade de planejamento é uma das causas dos problemas no desenvolvimento de software.

É importante destacar que o processo de desenvolvimento de software passa por níveis de abstração diferentes ao longo do tempo. Desta forma testes em níveis diferentes de abstração devem ser planejados, projetados, executados e avaliados. A Figura 3 mostra os diferentes níveis de abstração ao longo do processo de desenvolvimento. Nesta figura é apresentada uma visão interna e externa do processo de desenvolvimento de software ao longo do tempo. O processo começa com a análise e elicitação de requisitos de um ponto de vista mais abstrato (externo) e vai se detalhando até a implementação (visão interna). Assim que há uma implementação pronta, começam os testes, partindo de uma visão mais interna e detalhada (teste de unidade, integração e sistema) até uma visão mais externa e abstrata (teste de validação e certificação do sistema).



Fonte: Regnell, Runeson e Wohlin (2003, p.5).

Figura 3 – Diferentes níveis de abstração ao longo do processo de desenvolvimento

Antes dos testes serem efetivamente projetados, alguns princípios deveriam ser levados em conta (DAVIS apud PRESSMAN, 2002, p. 431-432):

- a) todos os testes devem ser relacionados aos requisitos do cliente: os defeitos mais indesejáveis de um software são aqueles que deixam de satisfazer os requisitos dos usuários;
- b) os testes devem ser planejados muito antes do início do teste: o planejamento dos testes pode começar tão logo o modelo de requisitos esteja definido. A definição dos casos de testes pode começar assim que o modelo do projeto tenha sido consolidado. Sendo assim, todos os testes podem ser planejados e projetados antes do código ter sido gerado;
- c) o princípio de Pareto se aplica ao teste de software: aplicando o princípio de Pareto, implica que 80% de todos os erros descobertos durante os testes serão, provavelmente, relacionados a 20% de todos os componentes do programa. O grande problema é isolar estes componentes suspeitos e testá-los rigorosamente;
- d) o teste deve começar pelas unidades do sistema até chegar ao sistema como um todo: os primeiros testes planejados e executados geralmente concentram-se nos componentes individuais. À medida que o teste progride, os testes vão sendo aplicados na integração entre os componentes e, por fim, no sistema inteiro;
- e) testar completamente não é possível: a quantidade de permutações de caminhos é excepcionalmente grande. Desta forma, é impossível executar todas as

combinações de caminhos durante o teste. É possível, no entanto, cobrir adequadamente a lógica do programa e garantir que todas as condições no nível de componentes tenham sido exercitadas;

- f) para ser mais efetivo, o teste deveria ser conduzido por terceiros: como dito anteriormente, a natureza do testador deve ser destrutiva. Por outro lado, pessoas envolvidas no processo de desenvolvimento (construção) têm natureza construtiva. Isto é, quanto menos a pessoa estiver envolvida no processo de desenvolvimento, maior é a probabilidade dela encontrar erros.

O projeto de casos de teste pode ser feito de duas formas principais (PRESSMAN, 2002, p. 435):

- a) teste de caixa preta: também conhecido como teste funcional, é o teste desenvolvido na interface do software. Apesar de ser projetado para descobrir erros, é usado para demonstrar que as funções do software estão implementadas, ou seja, que a entrada e adequadamente aceita e a saída é corretamente produzida, e que a integridade das informações externas é mantida. O teste de caixa preta examina algum aspecto fundamental do sistema preocupando-se pouco com a estrutura lógica interna do software;
- b) teste de caixa branca: também conhecido como teste estrutural, trata-se de um exame rigoroso da estrutura interna do software. Caminhos lógicos, internos ao software são testados, definindo casos de testes que exercitam conjuntos específicos de condições e/ou ciclos.

Além dos casos de testes poderem ser desenvolvidos de duas formas, estes podem ser aplicados em diferentes escopos ou estágios. O escopo do teste é a coleção de componentes de software a ser verificada. Como os testes devem ser executados sobre uma implementação, o escopo é tipicamente definido para corresponder ao componente que será testado. Tradicionalmente, o escopo de teste é dividido em (BINDER, 2000, p. 45-46): teste de unidade, teste de integração e teste de sistema.

No teste de unidade (PRESSMAN, 2002, p.476), testa-se a menor unidade do projeto de software, isto é, o menor componente do software. O teste de unidade é orientado para caixa branca e pode ser conduzido em paralelo para diversas unidades. No contexto de software convencional (desenvolvido com linguagens estruturadas), o teste de unidade deve ser conduzido pelos detalhes algorítmicos e o fluxo dos dados da unidade em teste. Já no contexto da orientação a objetos (OO), o teste de unidade é conduzido pelo teste da classe, onde a classe como um todo deve ser testada. Isto significa que não se pode mais testar uma

única operação isoladamente (um método, por exemplo) e sim como parte de uma classe. O teste de classe deve ser conduzido pelas operações encapsuladas nas classes e pelo estado de comportamento da classe.

O objetivo do teste de integração é descobrir erros associados às interfaces entre as unidades do software (ROCHA; MALDONADO; WEBER, 2001, p. 75). No contexto da OO, o teste de integração pode ser feito de duas maneiras (PRESSMAN, 2002, p. 623). Na primeira, um conjunto de classes é integrado para responder a uma entrada ou evento do sistema. Na segunda, o teste começa com as classes independentes integradas com outras classes dependentes e, então, são testadas. Isto se repete até que todo o sistema seja integrado e testado.

O escopo do teste de sistema é testar todo o sistema, levando em consideração componentes de software e hardware, ou seja, o software em seu provável ambiente de produção. Neste estágio são testadas características que estão presentes apenas no sistema como um todo. Entre os testes que são aplicados neste estágio estão testes funcionais (para validar requisitos do usuário), teste de performance (tempo de resposta) e teste de carga ou estresse (desempenho do sistema quando sobrecarregado) (BINDER, 2000, p. 45).

A Tabela 1 mostra a relação entre os estágios de teste e os paradigmas procedimental e OO.

Tabela 1 - Relação entre os estágios de teste e os paradigmas procedimental e OO

Estágio de Teste	Paradigma	
	Teste Procedimental	Teste Orientado a Objetos
Unidade	Sub-rotina ou função	Métodos das classes
Integração	Duas ou mais unidades	Classe <i>Cluster</i> Componente Subclasse
Sistema	Toda a aplicação	Toda a aplicação

Fonte: Adaptado de Rocha, Maldonado e Weber (2001, p. 76).

Além destes três estágios de teste, alguns autores como Pressman (2002, p. 487) apresentam um quarto estágio chamado de teste de validação ou teste de aceitação. A validação do software é feita por intermédio de uma série de testes de caixa preta que demonstram conformidade com os requisitos. Um plano de testes descreve os testes que

devem ser conduzidos e um procedimento de teste define os casos de testes específicos que serão usados para demonstrar a conformidade com os requisitos. Tanto o plano quanto os casos de teste são projetados para garantir que todos os requisitos funcionais sejam satisfeitos e alcançados, todas as características comportamentais sejam conseguidas, entre outros (PRESSMAN, 2002, p. 487).

Depois que cada caso de teste de validação tenha sido executado, uma de duas condições podem acontecer: ou o software está de acordo com os requisitos e é aceito ou um desvio da especificação é descoberto e é gerada uma lista de deficiências.

2.3 SISTEMA ATUAL

A Senior, pensando na garantia de qualidade de produtos e na garantia da fidelidade de seus clientes, está investindo a área de testes. Desta forma implantou um projeto chamado de PMPT, no qual existem 14 colaboradores do produto Vetorh envolvidos, divididos em:

- a) 10 colaboradores da equipe da qualidade, dentre os quais 8 são testadores e 2 analistas de qualidade, e;
- b) 4 analistas de sistemas.

Cada analista de sistemas é responsável por um ou mais módulos, sendo também responsável por efetuar o levantamento de determinada rotina definida nestes módulos. Em outras palavras, é o analista de sistemas quem efetua a análise da rotina que será testada e ele mesmo é quem alimenta primeiramente a planilha com os requisitos, juntamente com outras informações que se façam necessárias.

Após os requisitos estarem definidos, o analista de qualidade efetua o levantamento de quais partes das rotinas necessitam de casos de testes. Este levantamento é baseado nas tarefas de erros criadas pelos clientes da Senior. Assim que o levantamento termina, ele cria um Mapa de Suítes, descrevendo o objetivo de cada suíte de testes.

A Figura 4 apresenta um exemplo de uma planilha de Mapa de Suítes da rotina de apuração.

Mapa de Suítes - Apuração de Ponto					
Projeto/Operação:		5.5.2.6 - 22/11/2010 - 18/02/2011			
Versão do Módulo:		5.5.2.6			
Data do planejamento:		18/11/2010			
Responsável:		Bruna Herkenhof			
Módulos		Ronda			
Executar	Sequência	Id do TS	Nome da Suíte	Descrição da suíte	Abrangência
S	019	TS-HR-001-21	Teste Funcional de Apuração de Situação e Afastamentos	Tem o objetivo de evidenciar a conformidade da apuração de situações aos requisitos	Efetuar apuração de situações para colaboradores com os diferentes tipos de horários utilizando assinalamentos disponíveis em definições e cadastro
S	020	TS-HR-001-22	Teste Funcional de Apuração de Horário Tipo 1 – Rígido	Tem o objetivo de evidenciar a conformidade da apuração de horários tipo 1 (Rígido) aos requisitos	Efetuar apuração de colaboradores com horários tipo 1, combinando marcações, horários, programações e exceções, limites e limitrofes de marcações
S	021	TS-HR-001-23	Teste Funcional de Apuração de Horário Tipo 2 – Móvel	Tem o objetivo de evidenciar a conformidade da apuração de horário tipo 2 (Móvel) aos requisitos	Efetuar apuração de colaboradores com horários tipo 2, combinando marcações, horários, programações e exceções, limites e limitrofes de marcações

Figura 4 - Exemplo de Mapa de Suítes de Apuração

Cada suíte de teste é identificada por um código identificador, por exemplo TS-HR-001-21, composto pelas abreviações: TS, de Test Suíte; HR, a abreviação interna do módulo, neste caso Ronda Ponto; a numeração 001, que é o Código da Rotina de Apuração; e, finalmente, o número 21, correspondente ao código da Suíte de Teste.

Após definidas as suítes de testes, cada testador assume a rotina que possui conhecimento e cria os casos de testes para que os testes possam ser escritos. Os casos de testes são definidos de acordo com o objetivo da suíte de testes. A Figura 5 apresenta o exemplo de uma planilha de suíte de testes.

	A	B	C	D	E
1	Test Suite TS-HR-001-21				
2					
3					
4					
5	Nome da suite:	Teste Funcional de Apuração de Situação e Afastamentos			
6	Objetivo da suite:	Tem o objetivo de evidenciar a conformidade da apuração de situações aos requisitos			
7	Pré-condições:				
8					
9					
10	Executa	Sequência	Id do TC	Nome do caso de teste	Descrição
11	s	001	TC-HR-001-21-001	Utilização da definição de situação por empresa e sindicato.	Tem por objetivo a utilização das definições de situações de acordo com a parametrização da empresa e sindicato.
12	s	002	TC-HR-001-21-002	Afastamentos antes\durante\após horário normal de trabalho com tipo de horário "1-Rígido"	Tem por objetivo consistir as situações geradas antes\durante\após horário normal de trabalho com a definição do tipo de horário "1-Rígido"
13	s	003	TC-HR-001-21-003	Afastamentos antes\durante\após horário normal de trabalho com tipo de horário "2-Móvel"	Tem por objetivo consistir as situações geradas antes\durante\após horário normal de trabalho com a definição do tipo de horário "2-Móvel"
14	s	004	TC-HR-001-21-004	Afastamentos antes\durante\após horário normal de trabalho com tipo de horário "3-Flexível"	Tem por objetivo consistir as situações geradas antes\durante\após horário normal de trabalho com a definição do tipo de horário "3-Flexível"
15	s	005	TC-HR-001-21-005	Afastamentos antes\durante\após horário normal de trabalho com tipo de horário "4-Móvel Entrada e Saída"	Tem por objetivo consistir as situações geradas antes\durante\após horário normal de trabalho com a definição do tipo de horário "4-Móvel Entrada e Saída"

Figura 5 – Exemplo de planilha de Suíte de Testes

Como pode-se perceber uma suíte de testes nada mais é que um agrupamento de casos de testes. Assim como as suítes de testes, os casos de testes também são identificados através de um código identificador, TC-HR-001-21-001, por exemplo, no qual as abreviações TC significa *Test Case*, HR é uma abreviação interna do módulo, neste caso Ronda Ponto, a numeração 001 representa o código da rotina (Apuração), 21 representa o código da suíte de teste e, por fim, 001 representa o código do caso de testes.

As planilhas de casos de testes são o local onde os testes são literalmente escritos, linha por linha, nos quais cada situação deve ser testada. Elas também são alimentadas pelos testadores. A Figura 6 apresenta um exemplo de uma planilha de um caso de testes.

Test Case TC-HR-001-21-001									
Nome do case:		Utilização da definição de situação por Empresa e Sindicato.							
Objetivo do TC:		Tem por objetivo a utilização das definições de situações de acordo com a parametrização da Empresa e Sindicato.							
Suite de Teste:		TS-HR-001-21							
Executar	Sequencia	Situação de teste (O Que)	Condições de teste (Como)	Aplicação	Ambiente	Massa de dados	Pasta	Código Massa de Dados	Ação
S	001	Pré Requisitos	Teste utilizará arquivo de marcação	Ronda	C/S	Coletores_Modelos	Cadastro[A];Plantas[A]	t1	S
S	002		Entrar na tela da proprietária e desabilitar o Rubi.	Ronda	C/S	Tela_Configura_Proprietario		1	O
S	003		Seleciona a empresa 01	Ronda	C/S	Diversos_SelecionarEmpresa		01	O
S	005		Seleciona o cálculo 156	Ronda	C/S	Diversos_SelecionarCalculo_Individual		002	O
S	006	ST - 001 Valida as situações (Atrasos, Saída Antecipada e Saída Intermediária) em uma apuração sem a definição de situações no	Mudar a definição de situações na empresa, da empresa 1 para 2.	Ronda	C/S	Empresas_Empresas	Cadastro[A]	2	S
S	007		Retirar a definição de situação do sindicato 1.	Ronda	C/S	Tabelas_Sindicatos	Cadastro[A]	4	S

Figura 6 – Exemplo de uma planilha de Caso de Testes

As massas de dados (*templates*), relacionadas aos casos de testes, são nomeadas através do caminho do menu cuja tela está inserida no sistema. Por exemplo, existe uma massa de dados (*template*) chamada *Calculos_Apuracao_Calcular*. O nome nada mais é do que o caminho para a tela em questão. Nesta planilha serão inseridos os dados que devem ser incluídos nas telas pelos testadores.

A Figura 7 demonstra um exemplo da planilha de dados de uma massa de dados (*template*).

	A	B	C	D	E	F	G	H
1	Executar	Código Massa de	TestCase	Periodo_inicio	Periodo_Fim	Tipo_Processamento	Realocar_Marcacoes	Desfazer_Acertos_Situacoes
2	Tabela/Campos:							
3	TEMPLATE DE TELA							
4	Menu:	Cálculos/Apuração/Calcular						
5	Tela:	Cálculo de Apuração						
6	MASSA DE DADOS							
7	Executar	Código Massa de Dados	TestCase	Periodo_inicio	Periodo_Fim	Tipo_Processamento	Realocar_Marcacoes	Desfazer_Acertos_Situacoes
8	S	01	TC-HR-001-01-001; TC-HR-001-07-001;	26/06/2009	25/07/2009	0		
9	S	02	TC-HR-001-01-001	26/06/2009	26/06/2009	1	N	N
10	S	03	TC-HR-001-05-001	01/07/2009	01/07/2009	1	N	N
11	S	04	TC-HR-001-05-001	02/07/2009	15/07/2009	1	N	N
12	S	05	TC-HR-001-05-001; TC-HR-001-07-001	26/06/2009	25/07/2009	1	N	N
13	S	06	TC-HR-001-08-001	26/07/2009	25/08/2009	0		
14	S	07	TC-HR-001-08-001	05/08/2009	05/08/2009	1	N	N
15	S	08	TC-HR-001-08-002	26/05/2008	25/06/2008	0		
16	S	09	TC-HR-001-08-002	06/06/2008	06/06/2008	1	N	N
17	S	10	TC-HR-001-08-003	26/07/2009	25/08/2009	0		
18	S	11	TC-HR-001-07-001; TC-HR-001-07-002	26/07/2009	25/08/2009	0		
19	S	12	TC-HR-001-05-001	26/06/2009	25/07/2009	0		
20	S	13	TC-HR-001-07-001	03/08/2009	07/08/2009	1	N	N
21	S	14	TC-HR-001-08-004	26/07/2009	25/08/2009	0		
22	S	15	TC-HR-001-08-005	26/07/2009	25/08/2009	0		
23	S	16	TC-HR-001-09-001	26/07/2009	25/08/2009	0		
24	S	17	TC-HR-001-09-002	26/07/2009	25/08/2009	0		
25	S	18	TC-HR-001-09-002; TC-HR-001-10-002; TC-HR-001-10-	26/07/2009	25/08/2009	1	N	N
26	S	19	TC-HR-001-09-003	26/07/2009	25/08/2009	0		
27	S	20	TC-HR-001-09-003	26/07/2009	25/08/2009	0		
28	S	21	TC-HR-001-10-001	26/07/2009	25/08/2009	0		
29	S	22	TC-HR-001-07-001	29/06/2009	15/07/2009	1	N	N
30	S	23	TC-HR-001-10-010	26/04/2009	25/05/2009	0		
31	S	24	TC-HR-001-12-001	26/06/2010	25/07/2010	0		
	S	25	TC-HR-001-12-001; TC-HR-001-12-002;	28/06/2010	29/06/2010	0		

Figura 7– Exemplo de planilha de Massa de Dados (*template*)

A planilha de massa de dados (*templates*) contém todos os campos preenchíveis da tela, simulando o que ocorre no sistema.

As planilhas de dados voltam a ser revistas, quando, por algum motivo, é alterado algo no comportamento do sistema que influencie nos requisitos já documentados, ou mesmo para a inclusão de novos requisitos. O analista de sistemas abre uma nova tarefa para o analista de qualidade que deve então atualizar a planilha de requisitos. Este avisa verbalmente o testador responsável sobre esta tarefa. O testador deve verificar, dentre todas as planilhas, quais delas deverão ser atualizadas em virtude das alterações dos requisitos

A rastreabilidade é feita manualmente, verificando todas as planilhas que abrangem o requisito alterado. É necessário atentar aos apontamentos das planilhas para que nenhuma planilha que necessite de alteração fique esquecida sobre a atualização dos requisitos. Caso alguma planilha seja esquecida, os testes estarão inconsistentes, de forma que eles não estejam precisos e de acordo com o comportamento do sistema.

A Figura 8 apresenta o exemplo de uma planilha com alguns requisitos, na qual é possível perceber a complexidade dos apontamentos. As colunas Grupo, Sub Grupo e Item representam o caminho da tela no sistema.

Módulo: HR		Rotina:	Apuração de Ponto		Planilha:	Requisitos_Apuração_Ponto
Sequencia	Teste Suite	Teste Case	Grupo	Sub-Grupo	Item	Requisito
35	TS-HR-001-10; TS-HR-001-15	TC-HR-001-10-001; TC-HR-001-10-002; TC-HR-001-10-003; TC-HR-001-10-004; TC-HR-001-10-005; TC-HR-001-10-006; TC-HR-001-10-007; TC-HR-001-10-009; TC-HR-001-10-010;	Horário	Cadastro Horário	Turno	3 - 3º Turno (Das 22:00 às 5:00 horas).
36			Horário	Cadastro Horário	Turno	4 - 4º Turno (Pode ser um horário realizado apenas nos finais de semana)
37	TS-HR-001-15	TC-HR-001-15-006; TC-HR-001-15-007; TC-HR-001-15-008; TC-HR-001-15-009; TC-HR-001-15-010; TC-HR-001-15-011;	Horário	Cadastro Horário	Turno	8 - Misto (Das 18:00 às 1:00 horas).
38	TS-HR-001-15	TC-HR-001-15-006; TC-HR-001-15-007; TC-HR-001-15-008; TC-HR-001-15-009; TC-HR-001-15-010;	Horário	Cadastro Horário	Turno	9 - Geral (Das 8:00 às 18:00 horas).
39	TS-HR-001-10; TS-HR-001-15	TC-HR-001-10-001; TC-HR-001-15-006; TC-HR-001-15-007; TC-HR-001-15-008; TC-HR-001-15-009; TC-HR-001-15-010;	Horário	Cadastro Horário	Tipo de Jornada	E - Pela Entrada As marcações serão alocadas no dia em que ocorre o início da jornada.

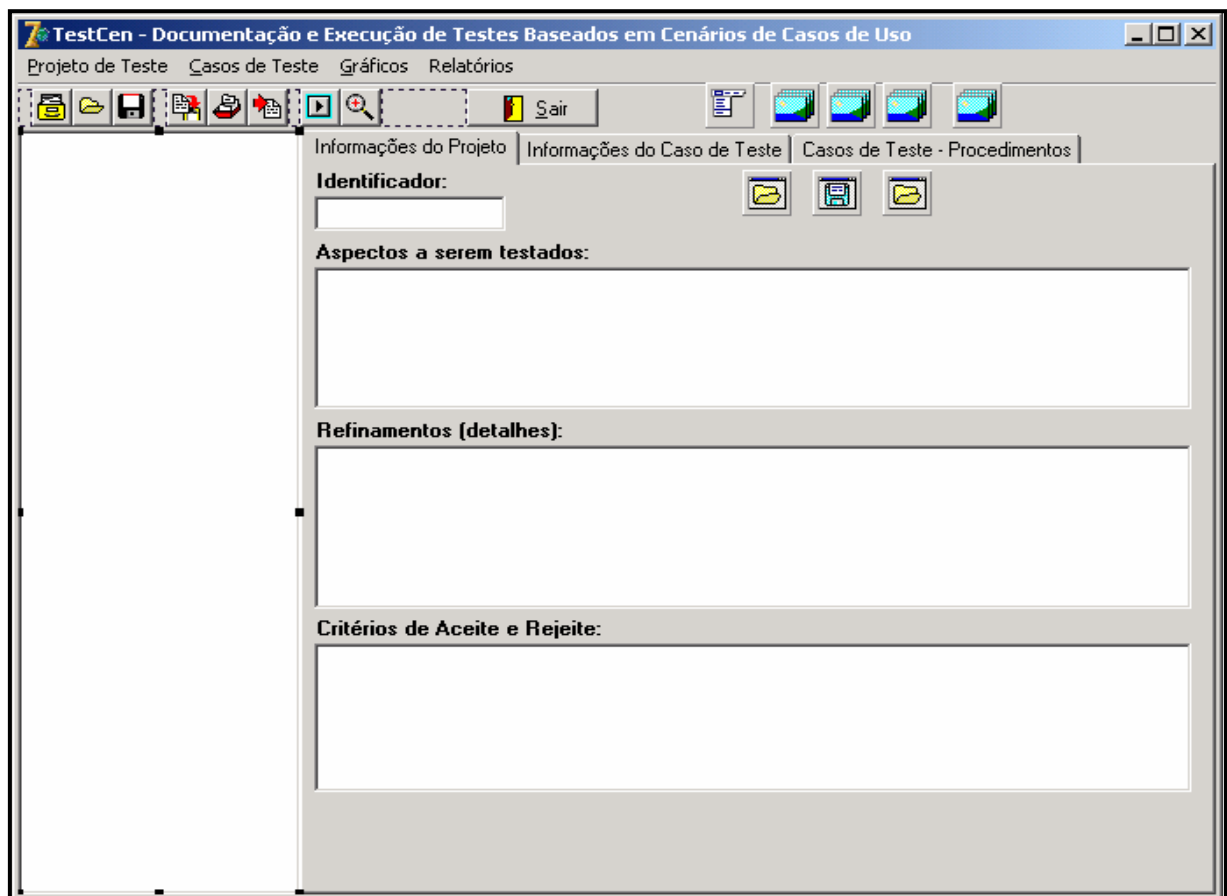
Figura 8– Exemplo de planilha de Requisitos

Por fim, quando finalizado o preenchimento de um caso de teste, ele será executado através de uma ferramenta já usada na empresa, o *TestComplete*, que executará os casos de testes escritos pela equipe de qualidade, de forma automatizada.

2.7 TRABALHOS CORRELATOS

Dentre os trabalhos pesquisados relacionados ao tema deste trabalho, são apresentados os que tiveram maior relevância para o mesmo.

O trabalho de conclusão de curso de Juliano Bianchini, desenvolvido na Universidade Regional de Blumenau (FURB), propõe uma ferramenta para suporte ao planejamento do teste funcional de software a partir da utilização de diagramas de casos de uso da UML, garantindo que o produto de software produzido esteja em conformidade com os requisitos, reduzindo os riscos do mesmo e evitando que ele apresente efeitos indesejáveis (falhas). É adaptada a ferramenta CASE ArgoUML onde foram feitas customizações, pois a ferramenta de suporte ao planejamento, controle, execução e documentação de testes desenvolvida utiliza os diagramas estendidos gerados na ferramenta CASE ArgoUML. Para o desenvolvimento deste trabalho a linguagem utilizada foi Object Pascal e componentes visuais do Delphi (BIANCHINI, 2004).



Fonte: Bianchini (2004).

Figura 9 – Imagem da tela principal da ferramenta

O trabalho de conclusão de curso de Ana Paula Zimmermann desenvolvido na Universidade do Vale do Itajaí (UNIVALI) foi aprofundar-se no processo de testes de software para a construção de uma ferramenta CASE (*Computer Aided Software Engineering*) para automação de testes de caixa preta através da técnica de particionamento de equivalência. Neste trabalho o objetivo foi de que se garantisse que os requisitos do sistema fossem plenamente atendidos pelo software. A ferramenta visa minimizar o tempo dedicado a esta atividade, bem como fornecer subsídios para incrementar a qualidade desta etapa do processo de desenvolvimento. O estudo de caso que foi escolhido para a validação da ferramenta é o módulo de caixa de software ItlSys. Trata-se de um programa crítico para as empresas na área do comércio (ZIMMERMANN, 2006).

O trabalho de conclusão de curso de Fernando Schiavo Cardoso é um estudo acadêmico sobre o método de software baseado em riscos, que é aplicado no planejamento dos testes associados aos requisitos do produto de software. Este método consiste em identificar, priorizar e criar casos de teste que exploram os riscos. O objetivo deste trabalho foi abordar as principais vantagens e limitações do método. As vantagens do teste de software baseado em riscos são agilidade no processo de teste, proteger as vítimas de possíveis erros, priorizar os esforços de teste e fazer melhor uso dos recursos de teste. A limitação apresentada se refere ao processo de análise de risco que não é uma tarefa trivial, onde dependerá muito do conhecimento do profissional. Correlaciona-se com o presente trabalho, pois os profissionais da qualidade da Senior criam testes para pontos críticos do sistema, que são riscos para o mercado (CARDOSO, 2008).

3 DESENVOLVIMENTO

Neste capítulo estão descritas as particularidades técnicas do sistema proposto tais como a descrição e a apresentação dos requisitos funcionais e não funcionais e principais diagramas.

Conforme definido nos objetivos específicos propostos neste trabalho, foi desenvolvido um gerenciador de artefatos, a fim de auxiliar e facilitar o trabalho da equipe da qualidade no projeto MPT.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Conforme Sommerville (2003) o processo de teste deve evoluir em estágios. Entre os estágios existe o teste de sistema, que se dedica a verificar se os requisitos funcionais e não funcionais são atendidos. No Quadro 1 são apresentados os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Requisitos Funcionais	Caso de Uso
RF01: A ferramenta deverá permitir ao analista de qualidade e ao testador logar no sistema.	UC01
RF02: A ferramenta deverá permitir ao analista de qualidade manter usuários.	UC02
RF03: A ferramenta deverá permitir ao analista de qualidade importar requisitos das planilhas para mantê-los apenas na ferramenta.	UC03
RF04: A ferramenta deverá permitir ao analista de qualidade importar suítes de testes das planilhas para mantê-las apenas na ferramenta.	UC04
RF05: A ferramenta deverá permitir ao analista de qualidade manter requisitos das rotinas a serem testadas.	UC05
RF06: A ferramenta deverá permitir ao analista de qualidade manter grupos.	UC06
RF07: A ferramenta deverá permitir ao analista de qualidade manter	UC07

subgrupos.	
RF08: A ferramenta deverá permitir ao analista de qualidade manter itens.	UC08
RF09: A ferramenta deverá permitir ao analista de qualidade manter módulos.	UC09
RF10: A ferramenta deverá permitir ao analista de qualidade manter rotinas dos módulos.	UC10
RF11: A ferramenta deverá permitir ao analista de qualidade manter Suítes de Testes.	UC11
RF12: A ferramenta deverá permitir ao analista de qualidade e ao testador manter casos de testes.	UC12
RF13: A ferramenta deverá permitir ao analista de qualidade e ao testador manter situações de testes.	UC13
RF14: A ferramenta deverá permitir ao analista de qualidade e ao testador manter vínculo entre os artefatos.	UC14
RF15: A ferramenta deverá permitir ao analista de qualidade e ao testador exportar caso de testes para planilha.	UC15
RF16: A ferramenta deverá permitir ao analista de qualidade e ao testador emitir relatório dos requisitos cadastrados.	UC16
RF17: A ferramenta deverá permitir ao analista de qualidade e ao testador emitir relatório de requisitos por data.	UC17
RF18: A ferramenta deverá permitir ao analista de qualidade e ao testador emitir relatório de vínculos entre casos de testes e requisitos.	UC18
RF19: A ferramenta deverá permitir ao analista de qualidade e ao testador emitir relatório de vínculos entre casos de testes, massa de dados e requisitos.	UC19
RF20: A ferramenta deverá permitir ao analista de qualidade e ao testador emitir relatório de requisitos por rotina.	UC20
RF21: A ferramenta deverá permitir ao analista de qualidade e ao testador emitir relatório de massa de dados pertencentes a casos de testes.	UC21

Quadro 1 – Requisitos Funcionais

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

Requisitos Não Funcionais
RNF01: A ferramenta deverá ter perfil de acesso para os usuários da ferramenta.
RNF02: A ferramenta deverá importar e exportar dados para planilhas Excel.
RNF03: A ferramenta deverá ser desenvolvida na ferramenta CASE GeneXus.
RNF04: A ferramenta deverá utilizar o banco de dados SQL Server 2008.
RNF05: A ferramenta deverá possuir uma interface <i>web</i> .

Quadro 2 – Requisitos Não Funcionais

3.2 ESPECIFICAÇÃO

O objetivo geral desta seção é apresentar a especificação do problema, através de diagramas, os quais representam este trabalho. Os diagramas de casos de uso e o diagrama de atividades foram desenvolvidos utilizando a ferramenta Enterprise Architect (EA). O modelo de dados relacional (MER) foi desenvolvido com a ferramenta de desenvolvimento Genexus.

3.2.1 Diagrama de casos de uso

Segundo Bezerra (2007, p. 55), “Um caso de uso representa uma determinada funcionalidade de um sistema conforme percebida externamente.”. A Figura 10 mostra o cenário com as funcionalidades do sistema que o Analista de Qualidade e Testador podem realizar. O usuário analista de qualidade tem acesso a todas as funcionalidades do sistema e o usuário testador possui apenas acesso a algumas telas conforme demonstrado no diagrama de casos de uso.

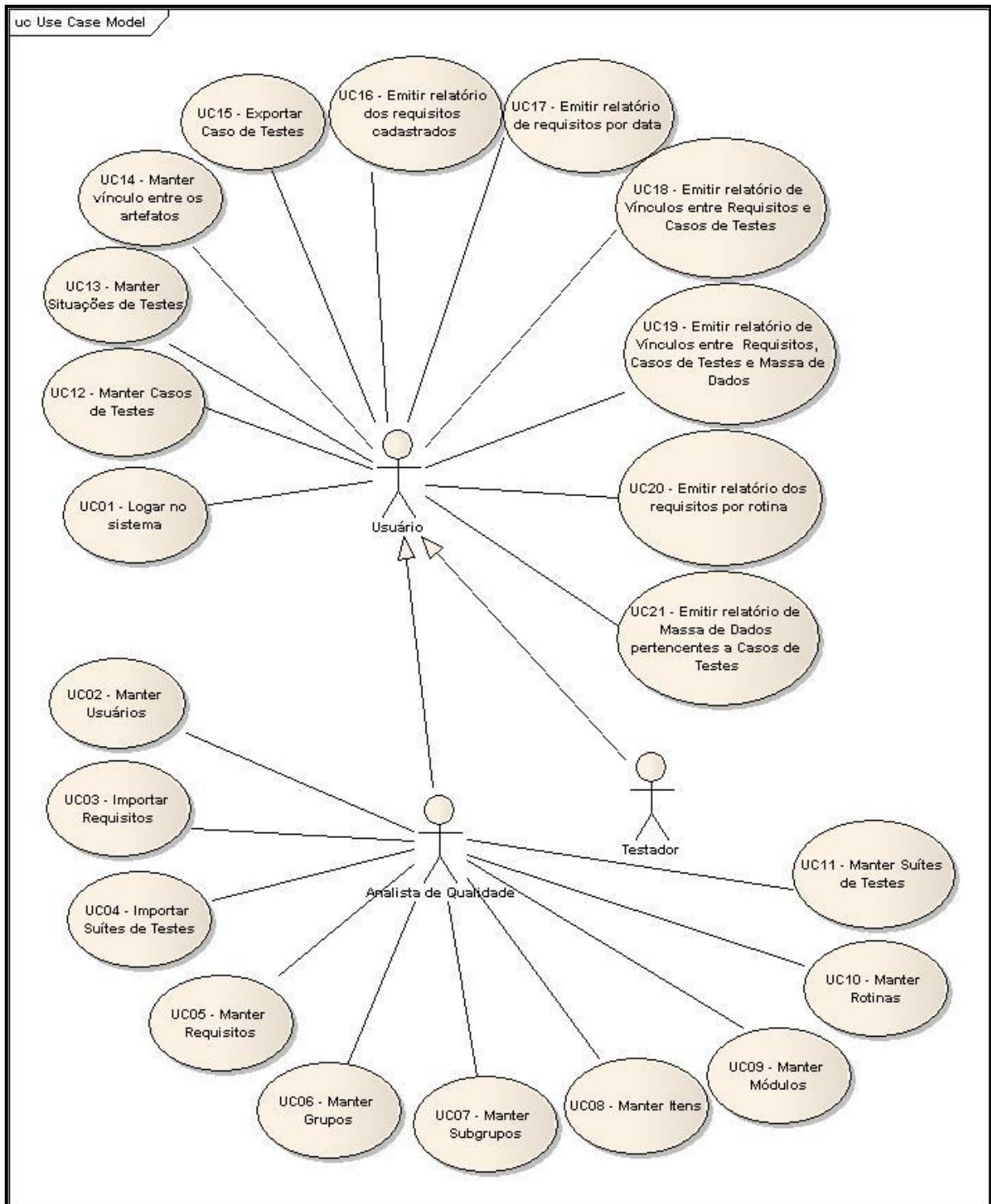


Figura 10 – Diagrama de Casos de Uso

3.2.2 Diagrama de atividades

A Figura 11 mostra o diagrama de atividades para a implantação da ferramenta, ou seja, quando se fazem necessárias importações de planilhas.

Conforme Martins (2002, p. 99), “A finalidade do Diagrama de Atividades é documentar o fluxo de execução de algum procedimento, processo ou rotina”.

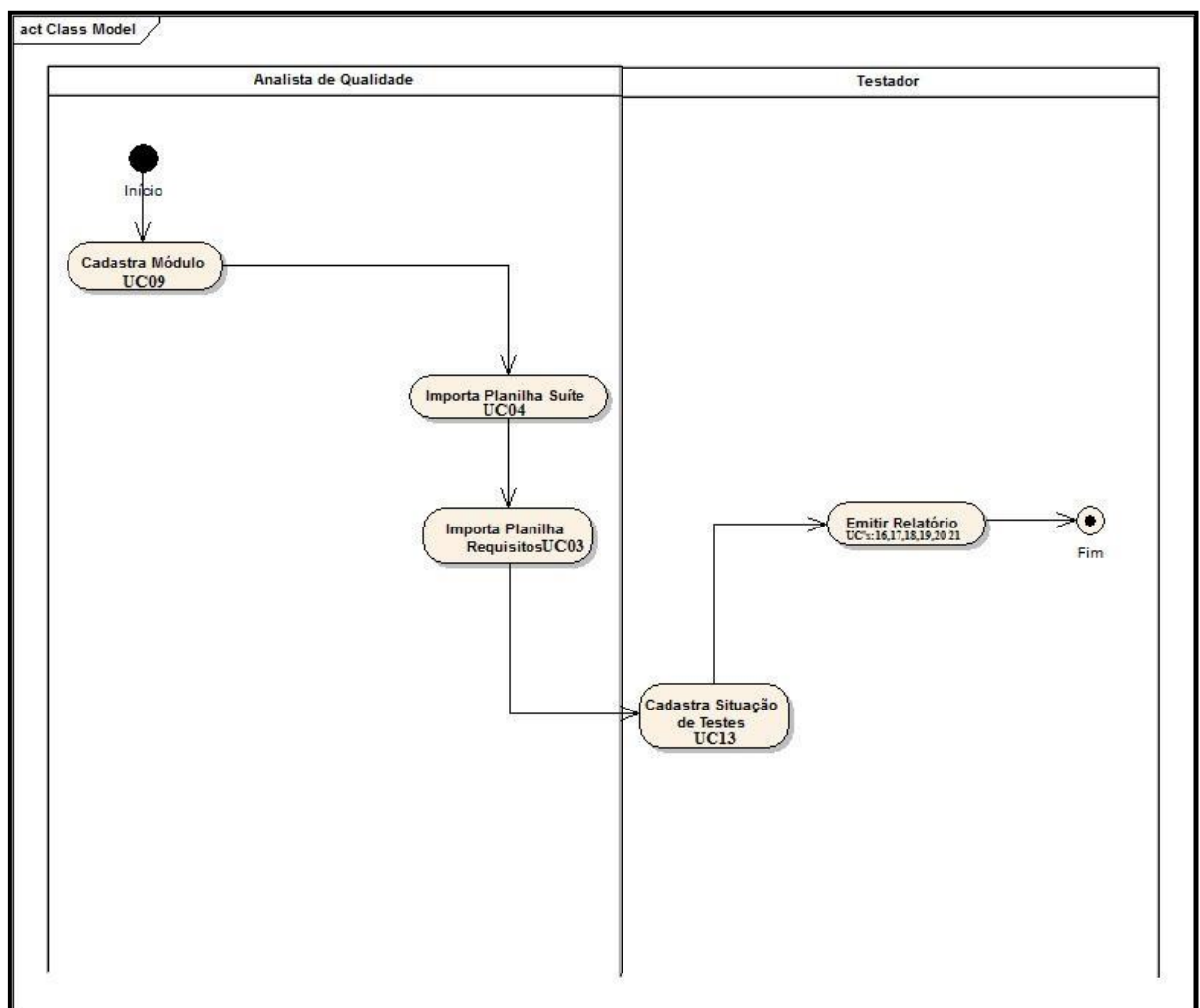


Figura 11 – Diagrama de Atividades para Implantação

A Figura 12 apresenta o diagrama de atividades que inicia no cadastro do módulo pelo analista de qualidade até a visualização das informações através dos relatórios. Neste diagrama podem ser verificadas todas as atividades a serem executadas na ferramenta desenvolvida depois de efetuada a implantação.

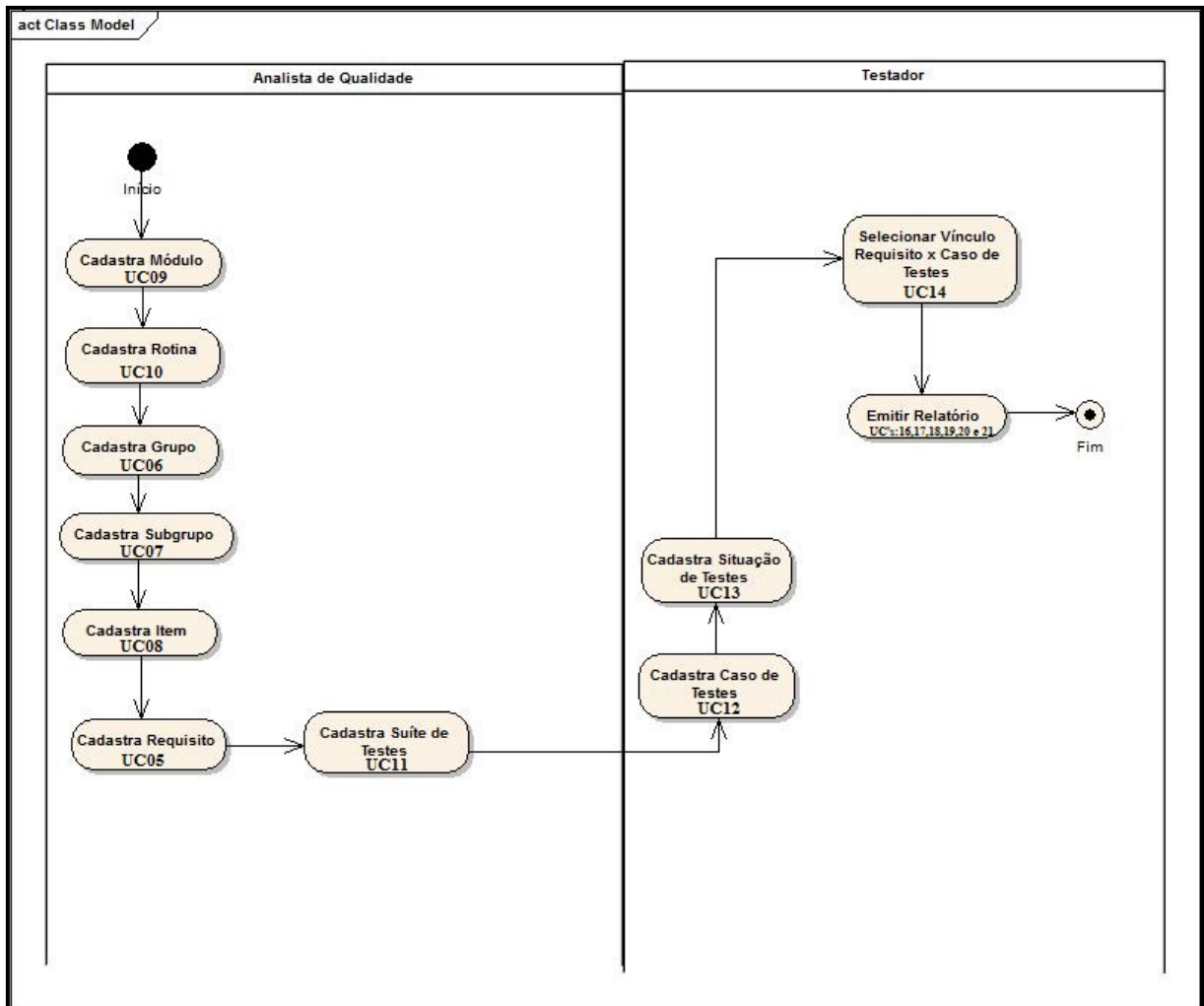


Figura 12 – Diagrama de Atividades

3.2.3 Modelo de dados relacional

A Figura 13 mostra o Modelo de dados Relacional (MER), com base nas tabelas da ferramenta e seus relacionamentos. O dicionário de dados desenvolvido para especificar o sistema, é apresentado no Apêndice B.

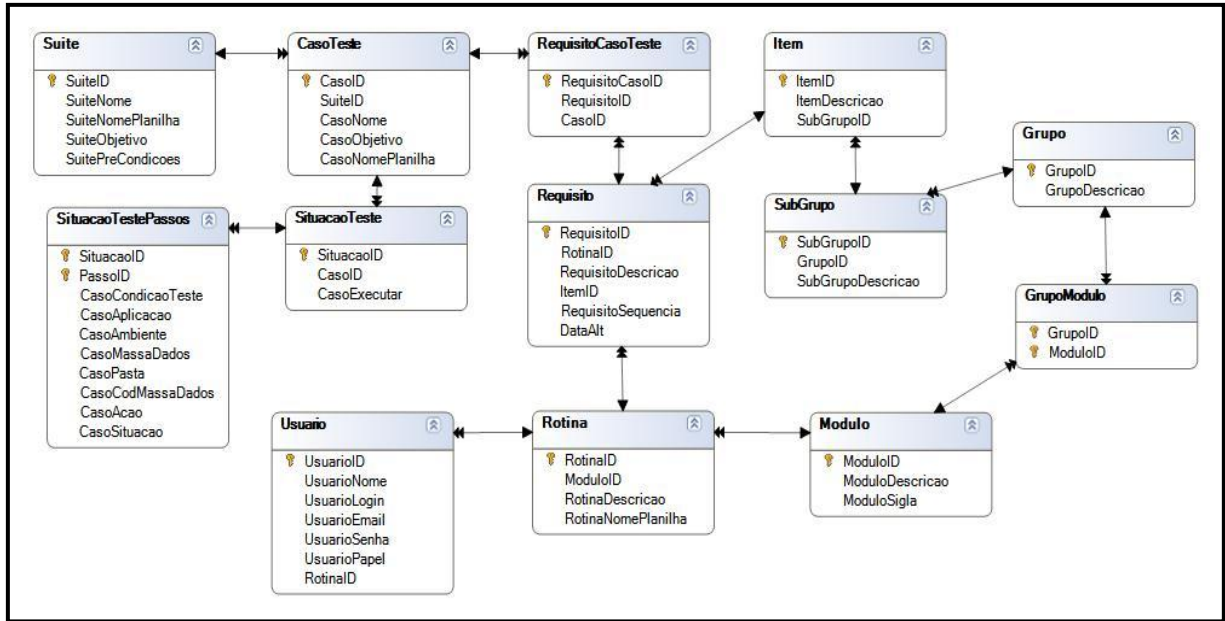


Figura 13 – Modelos de Dados Relacional

3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento deste trabalho foi utilizada a ferramenta Genexus Evolution I e para armazenamento dos dados foi utilizado o SGBD SQL Server 2008.

3.3.1.1 SQL Server 2008

A ferramenta de banco de dados, utilizada na aplicação é SQL Server 2008, desenvolvida pela Microsoft. Conforme Microsoft (2010?), algumas das vantagens oferecidas pelo banco de dados MS SQL Server 2008 são:

- elevada disponibilidade: a tecnologia de *failover clustering* e de *mirroring* da base

- de dados do SQL Server 2008 permitirá às empresas oferecerem aplicações altamente confiáveis e disponíveis aos seus colaboradores, clientes e parceiros;
- b) ferramentas de gestão: o SQL Server 2008 apresenta um pacote integrado de ferramentas de gestão e de interfaces de programação de aplicações de gestão (API), proporcionando assim facilidade de utilização e de gestão, além de suporte à operação de implementações de larga escala do SQL Server;
 - c) melhorias na segurança: o SQL Server 2008 foi projectado de forma a oferecer o nível mais elevado de segurança a dados empresariais através de funcionalidades como encriptação da base de dados, configurações de origem mais seguras, aplicação da política de palavras-passe, controlo apertado das permissões e um modelo de segurança optimizado;
 - d) escalabilidade: entre as melhorias de escalabilidade do SQL Server 2008, contam-se partição de tabelas, melhorias na replicação e suporte de 64 bits.

3.3.1.2 Ferramenta Genexus

GeneXus é uma ferramenta de desenvolvimento de software criada pela empresa uruguaia ARTech. É um poderoso software baseado em conhecimento puro que permite focar em seu negócio para desenvolver de forma incremental aplicações de missão crítica. A ferramenta GeneXus conta com um ambiente de desenvolvimento amigável, orientando à intenções e necessidades do desenvolvedor, que fazem seu uso mais intuitivo e facilitam sua aprendizagem.

A ferramenta inclui um módulo de normalização, que cria e mantém uma estrutura de banco de dados (BD) ótima, baseada no modelo de dados não normalizado definido pelo usuário, uma linguagem declarativa (baseada em regras) e uma linguagem procedural simples e poderosa.

A ferramenta GeneXus possui diversos geradores de código, que permitem gerar o software do projeto em várias linguagens, entre elas a Linguagem Java (GENEXUS, 2010?).

Na Figura 14 é apresentado o código fonte Genexus onde são efetuados os vínculos dos artefatos, suítes de testes e casos de testes com os requisitos na importação da planilha de requisitos.

```

214 sub 'VerificaRequisito'
215     if &RequisitoID.IsEmpty()
216         Msg('Erro ao importar requisito!')
217         return
218     else
219         // monta collection separador por ;
220         &separa_caso = &var_caso.SplitRegEx(';')
221         // Para cada requisito incluído, cadastra as relações
222         for &ind_j = 1 to &separa_caso.Count step 1
223             &suitenomeplanilha = 'TS' + substr( trim(&separa_caso.Item(&ind_j)), 3, 10 )
224             &SuiteID = 0
225             for each
226                 where &suitenomeplanilha = SuiteNomePlanilha
227                     &SuiteID = SuiteID
228             endfor
229             // Verifica se o caso já está cadastrado
230             &CasoId = 0
231             for each
232                 where SuiteID = &SuiteID
233                 where CasoNomePlanilha = trim(&separa_caso.Item(&ind_j))
234                     &CasoId = CaseID
235             endfor
236             if &CasoId > 0
237                 // Cria a relação requisito x caso
238                 new
239                     RequisitoID = &RequisitoID
240                     CasoID = &CasoId
241                 endnew
242             endif
243         endfor
244     endif
245 endsub

```

Figura 14 – Vínculo de artefatos na importação de planilha de requisitos

A seguir na Figura 15 é demonstrado o código fonte da exportação de casos de testes para planilhas *Excel*, a qual a ferramenta *TestComplete* utilizará para efetuar os testes.

```

1 // Início Bloco Exportação de Planilha
2 Msg('Iniciando Exportação da Planilha de Caso de Teste.')
3 // Define o template padrão para exportação
4 &planilhaExportar.Template = 'C:\Artefatos\ModeloCasoTeste.xls'
5 // Abre a planilha onde serão importados os dados.
6 // Se a planilha não existir vai usar o template.
7 &planilhaExportar.Open("C:\Artefatos\" + &CasoNomePlanilhaExportar + ".xls")
8 // Preenche cabeçalho
9 for each
10     where CasoNomePlanilha = &CasoNomePlanilhaExportar
11     &planilhaExportar.Cells(2,3).Text = Trim(CasoNomePlanilha)
12     &planilhaExportar.Cells(3,3).Text = Trim(CasoNome)
13     &planilhaExportar.Cells(4,3).Text = Trim(CasoObjetivo)
14     &casoId = CasoID
15     for each
16         where SuiteID = SuiteID
17         &planilhaExportar.Cells(5,3).Text = Trim(SuiteNomePlanilha)
18         exit
19     endfor
20     exit
21 endfor
22 // Preenche os passos
23 &ind_lin = 7
24 for each
25     where CasoID = &CasoId
26     &planilhaExportar.Cells(&ind_lin,1).Text = Trim(CasoExecutar)
27     &planilhaExportar.Cells(&ind_lin,2).Number = PassoID
28     &planilhaExportar.Cells(&ind_lin,3).Text = Trim(CasoSituacao)
29     &planilhaExportar.Cells(&ind_lin,4).Text = Trim(CasoCondicaoTeste)
30     &planilhaExportar.Cells(&ind_lin,5).Text = Trim(CasoAplicacao)
31     &planilhaExportar.Cells(&ind_lin,6).Text = Trim(CasoAmbiente)
32     &planilhaExportar.Cells(&ind_lin,7).Text = Trim(CasoMassaDados)
33     &planilhaExportar.Cells(&ind_lin,8).Text = Trim(CasoPasta)
34     &planilhaExportar.Cells(&ind_lin,9).Text = Trim(CasoCodMassaDados)
35     &planilhaExportar.Cells(&ind_lin,10).Text = Trim(CasoAcao)
36     &ind_lin = &ind_lin + 1
37 endfor
38 if (&planilhaExportar.errCode > 0)
39     &planilhaExportar.Close()
40     Msg('Problemas ao Exportar Planilha de Caso de Teste: Erro: ' +
41         Trim(&planilhaExportar.ErrDescription))
42     return
43 else
44     &planilhaExportar.Save()
45     &planilhaExportar.Close()
46     Msg('Planilha de Caso de Teste Exportada com Sucesso!')
47 endif
48 // Final Bloco Exportação de Planilha
49

```

Figura 15 – Exportação de casos de testes para planilha *Excel*

3.3.1.3 Apache Tomcat 5.5

Para a execução da ferramenta proposta optou-se pelo servidor web com base nas vantagens que oferece.

“O Tomcat é um servidor de aplicações Java para *web*. É software livre e de código aberto, surgido dentro do conceituado projeto Apache Jakarta e que teve apoio e endosso oficial da Sun Microsystems como Implementação de Referência (RI) para as tecnologias Java Servlet e JavaServer Pages (JSP). Atualmente, o Tomcat tem seu próprio projeto de desenvolvimento independente, dentro da Apache Software Foundation. O Tomcat é robusto e eficiente o suficiente para ser utilizado mesmo em um ambiente de produção”. (APACHE, 2011?).

3.3.2 Operacionalidade da implementação

Nesta seção será apresentada a sequência de telas e operações para correta utilização da ferramenta de Gerenciador de Artefatos, de acordo com os diagramas da Figura 12 e 13.

3.3.2.1 Logar no sistema

Ao abrir o sistema é apresentada a tela de *login* conforme Figura 16. O *login* terá função de diferenciar analistas de qualidade de testadores. Analistas de qualidade têm permissão de acesso para toda ferramenta, conforme Figura 17. Um analista de qualidade utilizará o usuário e senha *admin*, podendo cadastrar novos analistas e testadores. Usuários cujo papel é testador têm acesso apenas para as telas de *login*, casos de testes, situações de testes, vincular artefatos e emissão de todos os relatórios.

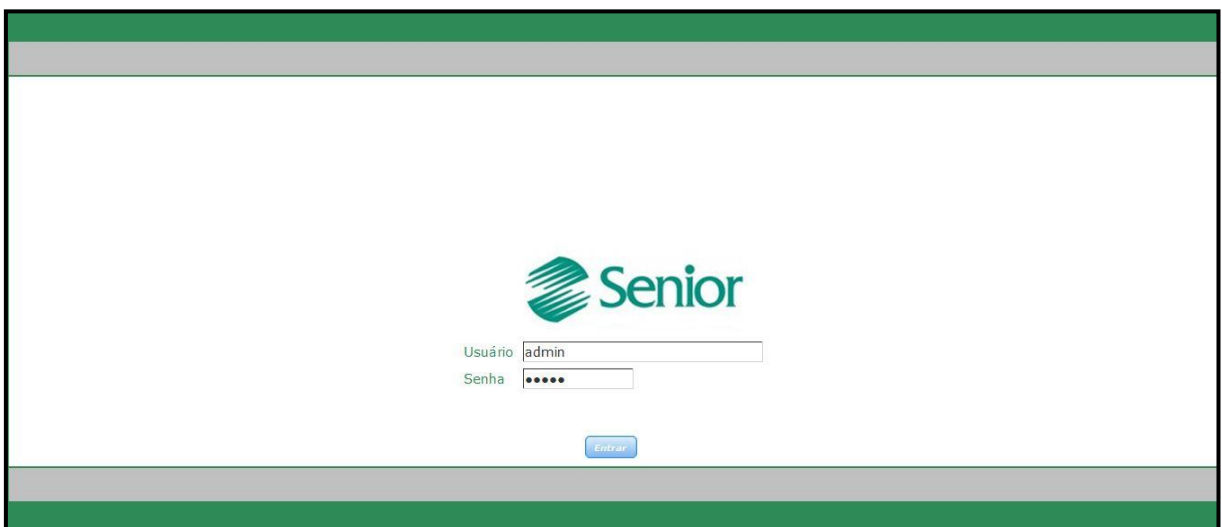


Figura 16 – Tela de *Login*



Figura 17 – Menu Principal Analista de Qualidade

O cadastro de módulos, que se refere a um agrupamento de rotinas do sistema, é necessário para o andamento do processo da ferramenta, tanto para a importação quanto para o cadastro manual. Esta tela está disponível no menu `Cadastros`.

Módulo	
Código	1
Descrição	<input type="text" value="Ronda"/>
Sigla	<input type="text" value="HR"/>
<input type="button" value="Confirmar"/> <input type="button" value="Fechar"/>	

Figura 18 – Tela de Cadastro de Módulo

A importação de planilhas de suítes de testes é feita pela tela `Importar Planilha`, presente no menu `Ações`. Basta selecionar a planilha desejada através do botão `Procurar`, conforme demonstrado na Figura 19. Com a importação desta planilha o cadastro da suíte de testes e os dados para a tabela de casos de testes pertencentes a esta suíte serão

armazenados na ferramenta.



Gerenciador de Artefatos

Cadastros ▾ Artefatos ▾ Ações ▾ Relatórios ▾ Diversos ▾ Logout

Suíte importada com sucesso!
Casos importados com sucesso!

Selecione o Tipo da Planilha para Importação:

Suíte de Teste
 Requisito

Nome do arquivo:

C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\MPT_KnowledgeBa

Figura 19 – Tela de Importar Planilhas – Suíte de Teste

Após a importação de planilhas de suítes de testes o usuário analista poderá importar a planilha de requisitos, conforme Figura 20. Para que os vínculos entre Requisito, Suítes de Testes e Casos de Testes sejam armazenados corretamente na ferramenta, a importação de todas as suítes presentes na planilha de requisitos devem ser realizadas antes dos requisitos para que a ferramenta crie os vínculos dos requisitos com as suítes de testes e casos de testes, que por sua vez foram importados para a ferramenta através da importação de suítes de testes.

Planilha importada com sucesso!

Selecione o Tipo da Planilha para Importação:

Suíte de Teste

Requisito

Nome do arquivo:

Figura 20 – Tela de Importar Planilhas - Requisito

Ao importar uma planilha de requisitos, além de a ferramenta fazer o vínculo entre os três artefatos (Requisitos, Suítes e Casos) conforme Figura 21, ela efetua o cadastro dos dados da célula Rotina e das colunas Grupo, Sub Grupo e Item demonstradas nas Figuras 22, 23, 24 e 25 respectivamente.

Informe o Requisito

Vínculo	Requisito	Requisito Descrição	Caso Nome Planilha	Suíte Nome Planilha
301	1781	Este assinalamento indica a hora inicial e final da Jornada Noturna. Estas horas tem tratamento diferenciado e sendo assim servem como base para o cálculo, seja para qualquer tipo de horário aplicado.	TC-HR-001-10-002	TS-HR-001-10
302	1781	Este assinalamento indica a hora inicial e final da Jornada Noturna. Estas horas tem tratamento diferenciado e sendo assim servem como base para o cálculo, seja para qualquer tipo de horário aplicado.	TC-HR-001-10-003	TS-HR-001-10
303	1781	Este assinalamento indica a hora inicial e final da Jornada Noturna. Estas horas tem tratamento diferenciado e sendo assim servem como base para o cálculo, seja para qualquer tipo de horário aplicado.	TC-HR-001-10-004	TS-HR-001-10

Figura 21 – Tela de Visualização de Vínculos de Artefatos

Gerenciador de Artefatos

Cadastros ▾ Artefatos ▾ Ações ▾ Relatórios ▾ Diversos ▾ Logout

⏪ ⏩ ⏴ ⏵ 🔍 📄 🔄 ✖

Rotina

Código

Descrição

Módulo

Nome Planilha

Figura 22 – Tela de Cadastro de Rotina

Gerenciador de Artefatos

Cadastros ▾ Artefatos ▾ Ações ▾ Relatórios ▾ Diversos ▾ Logout

⏪ ⏩ ⏴ ⏵ 🔍 📄 🔄 ✖

Grupo

Código

Descrição

Módulo

	Modulo ID	Modulo Descricao	Modulo Sigla
x	1	Ronda	HR
x	2	Rubi	FP

Figura 23 – Tela de Cadastro de Grupo

The screenshot shows the 'Gerenciador de Artefatos' interface. At the top, there is a logo and the title 'Gerenciador de Artefatos'. Below this is a navigation menu with items: 'Cadastros', 'Artefatos', 'Ações', 'Relatórios', 'Diversos', and 'Logout'. The main content area features a toolbar with various icons and a form titled 'Sub Grupo'. The form contains the following fields: 'Código' with the value '22', 'Descrição' with the value 'Cadastro Escala', and 'Grupo' with a dropdown menu showing 'Escalas'. At the bottom of the form are three buttons: 'Confirmar', 'Fechar', and 'Eliminar'.

Figura 24 – Tela de Cadastro de Sub Grupo

The screenshot shows the 'Gerenciador de Artefatos' interface. At the top, there is a logo and the title 'Gerenciador de Artefatos'. Below this is a navigation menu with items: 'Cadastros', 'Artefatos', 'Ações', 'Relatórios', 'Diversos', and 'Logout'. The main content area features a toolbar with various icons and a form titled 'Item'. The form contains the following fields: 'Código' with the value '15', 'Descrição' with the value 'Tipo de Escala', and 'SubGrupo' with a dropdown menu showing 'Cadastro Escala'. At the bottom of the form are three buttons: 'Confirmar', 'Fechar', and 'Eliminar'.

Figura 25 – Tela de Cadastro de Item

Caso não sejam importadas alguma das planilhas mencionadas anteriormente, o usuário analista deverá cadastrar manualmente a rotina, conforme demonstrado na Figura 22.

Após o cadastro da rotina, deve-se cadastrar grupo, sub grupo e item, conforme demonstrado nas Figuras 23, 24 e 25.

Com os cadastros acima efetuados é possível cadastrar os requisitos através da tela

Requisitos, presente no menu Artefatos. Ao inserir ou alterar o cadastro de requisito o campo data é atualizado com a data atual para controle de manutenção dos artefatos, conforme Figura 26.



The screenshot displays the 'Gerenciador de Artefatos' web application interface. At the top, there is a green header with the application logo and title. Below the header is a navigation menu with options: 'Cadastros', 'Artefatos', 'Ações', 'Relatórios', 'Diversos', and 'Logout'. The main content area is titled 'Requisito' and contains a form with the following fields:

- Código:** 1759
- Sequência:** 3
- Rotina:** Apuração de Ponto
- Descrição:** B - Busca Automática. Escala que permite ter até 6 horários definidos no dia. O critério de qual horário será utilizado no dia segue o parâmetro Tipo da Busca. Vide item Tipo da Busca Automática.
- Item:** Turno de escala
- SubGrupo:** 22
- Grupo:** 43
- Data:** 22/11/11

At the bottom of the form, there are three buttons: 'Confirmar', 'Fechar', and 'Eliminar'.

Figura 26– Tela de Cadastro de Requisitos

Quando não houver a importação de planilhas deve-se cadastrar a Suíte de Testes.

Gerenciador de Artefatos

Cadastros ▾ Artefatos ▾ Ações ▾ Relatórios ▾ Diversos ▾ Logout

Suíte de Teste

Código:

Suíte Nome Planilha:

Nome:

Objetivo:

Pré-condição:

Figura 27 – Tela de Cadastro de Suíte de Teste

O cadastro de casos de testes também deve ser efetuado manualmente quando não há importação de planilha, conforme Figura 28. Através desta tela também é possível exportar os dados cadastrados nesta juntamente com os dados cadastrados na tela Situação de Testes para uma planilha *Excel*.

Gerenciador de Artefatos

Cadastros ▾ Artefatos ▾ Ações ▾ Relatórios ▾ Diversos ▾ Logout

Caso de Teste

Código:

Nome Planilha:

Suíte de Teste:

Nome:

Objetivo:

Figura 28 – Tela de Cadastro de Caso de Teste

A seguir é apresentada a Figura 29 referente à tela de Situação de Testes,

fundamental para identificação das massas de dados (*templates*) que também devem receber manutenção quando um requisito for alterado. Nesta tela são descritos o passo a passo dos testes, que por sua vez são baseados nos requisitos.

Gerenciador de Artefatos

Cadastros ▾ Artefatos ▾ Ações ▾ Relatórios ▾ Diversos ▾ Logout

Situação de Teste

Código 3
 Caso de Teste TC-HR-001-21-001

Passos

ID	Caso Executar	Situação	Condição de Teste	Aplicação	Ambiente	Massa de Dados	Pasta	Massa de Dados	Ação
x 1	S	Pré-Requisitos	Teste utilizará arquivo	Ronda	C/S	Coletores_Modelos	Cadastro[A]; Plantas	1;1	S
x 2	B		Entrar na tela da prop	Ronda	C/S	Tela_Configura_Propri		1	O
x 3	S	ST - 001 Valida as situ	Mudar a definição de s	Ronda	C/S	Empresas_Empresas	Cadastro[A]	2	S
x 4	S		Retirar a definição de	Ronda	C/S	Tabelas_Sindicatos	Cadastro[A]	4	S

[Novo registro]

Confirmar Fechar

Figura 29 – Tela de Cadastro de Situação de Teste

Se a importação de planilha não foi efetuada, qualquer usuário poderá efetuar os vínculos entre os artefatos na tela de Vincular Artefatos. Deve-se informar quais casos de testes estão testando que requisitos, conforme Figura 30.

Gerenciador de Artefatos

Cadastros ▾ Artefatos ▾ Ações ▾ Relatórios ▾ Diversos ▾ Logout

Vínculos Requisito X Caso de Teste

Código Vínculo 59
 Requisito 1706
 Requisito Descrição P - Permanente. Escala que funciona de forma contínua. Informa-se uma data de início que não será modificada.
 Caso ID TC-HR-001-21-007
 Suite ID TS-HR-001-21

Confirmar Fechar

Figura 30 – Tela de Cadastro de Vínculo de Artefatos

Na Figura 31, é demonstrado o relatório de Requisitos por Data selecionados a partir de uma data inicial e final informados na tela de entrada do relatório. Ele auxilia na

verificação de requisitos incluídos ou alterados recentemente para a manutenção dos artefatos.

Requisitos por Data		Período: 17/11/11 - 30/11/11
Requisito	Data	
1757	28/11/11	
P - Permanente. Escala que funciona de forma contínua. Infoma-se uma data de inicio que não será modificada.		
1758	22/11/11	
D - Definida. Escala que é definida a cada mês.		

Figura 31 – Relatório de Requisitos por Data

Na Figura 32 é demonstrado o relatório que lista os vínculos entre requisitos e casos de testes, facilitando a verificação de manutenção nos artefatos, ou seja, quais casos de testes devem ser alterados quando efetuadas alterações nos requisitos.

Vínculo Requisito x Caso de Teste	
Requisito	Caso de Teste
1706	P - Permanente. Escala que funciona de forma contínua. Infoma-se uma data de inicio que não será modificada.
7	TC-HR-001-21-007
6	TC-HR-001-21-006
1709	C - Efetua rateio da Mão-de-Obra por Centro de Custo
25	TC-HR-001-15-007
26	TC-HR-001-15-008
27	TC-HR-001-15-009
28	TC-HR-001-15-010

Figura 32 – Relatório de Vínculo Requisito X Caso de Teste

Além dos demonstrados nas figuras 31 e 32, a ferramenta disponibiliza outros relatórios que facilitam a manutenção dos artefatos:

- relatório de Requisitos, listam todos os requisitos cadastrados na ferramenta;
- relatório de Requisitos por Rotina, através de uma rotina selecionada são

- listados os requisitos pertencentes a ela;
- c) relatório de Vínculos de Requisitos com Casos de testes e Massas de dados, através de um requisito são listados os casos de testes pertencentes a ele juntamente com as massas de dados de cada caso de teste;
 - d) relatório de Massa de Dados pertencentes a Casos de testes. Com estas informações o usuário tem uma visão mais ampla do tempo que terá de reservar para esta atividade quando necessitar de manutenção em algum requisito.

3.4 RESULTADOS E DISCUSSÃO

O principal objetivo deste trabalho foi desenvolver uma ferramenta para o gerenciamento de artefatos *web* para que os profissionais na equipe de qualidade da empresa Senior possam obter mais segurança e organização na rastreabilidade entre os artefatos do projeto MPT. Correlacionando com a prática atual desta empresa os objetivos foram alcançados e todos os requisitos propostos foram desenvolvidos.

No desenvolvimento do trabalho, verificou-se que a linguagem C# escolhida para utilizar na ferramenta de desenvolvimento tem algumas restrições em relação à importação de planilhas Excel, por este motivo foi alterada para a linguagem Java. Com isto a versão do banco de dados utilizado também foi alterada de SQL Server 2005 para SQL Server 2008.

Os trabalhos correlatos citados, assim como este, possuem como objetivo desenvolver ferramentas direcionadas à qualidade de softwares com o intuito de aprimorar e facilitar processos de testes, garantindo que os requisitos do sistema sejam plenamente atendidos pelo software, reduzindo os riscos e evitando efeitos indesejáveis. Desta forma, minimizando o esforço da equipe de qualidade em determinados processos, podendo assim usufruir do tempo minimizado para a realização de outras atividades. Os trabalhos auxiliaram para obter uma visão mais ampla sobre qualidade e processo de testes, o tamanho da sua importância nas fases do ciclo de desenvolvimento de um software.

Os testes da ferramenta foram aplicados por dois profissionais da equipe da qualidade que trabalham diariamente com o PMPT na empresa Senior. Utilizaram as planilhas já existentes para efetuar uma simulação real da utilização da ferramenta no PMPT. Foi observado que não obtiveram maiores dificuldades para a utilização da ferramenta, porém sugeriram algumas melhorias que serão comentadas nas extensões.

4 CONCLUSÕES

Como conclusão pode-se analisar que atualmente a área de teste de software é explorada cada vez mais, elencando vários aspectos: os sistemas estão tornando-se mais complexos; entre as empresas da área de software a concorrência é cada vez maior; e por fim os clientes tornam-se mais exigentes.

Analisando os objetivos específicos deste trabalho de conclusão de curso, conclui-se que todos foram cumpridos, desta forma, atingindo o objetivo geral. A ferramenta desenvolvida auxilia na etapa de gerenciamento dos artefatos utilizados no projeto MPT, aplicado a empresa Senior. É possível utilizar todas suítes de testes e requisitos já criados até o momento, aproveitando todo o esforço já exercido para o projeto. Através da utilização da ferramenta os artefatos serão mantidos com organização, obtendo a rastreabilidade com maior precisão, possibilitando efetuar a consulta e a manutenção com clareza dos vínculos entre os artefatos. Desta forma, na ocasião de manutenções em requisitos, será possível que os profissionais da qualidade avaliem o impacto das alterações nos artefatos, verificando as devidas alterações para o sucesso do processo de testes no projeto MPT.

A apresentação da ferramenta para os gerentes da Senior ainda não foi realizada. Assim sendo, não há até o momento do término deste trabalho respostas concretas sobre a implantação desta ferramenta na empresa Senior. Quando for apresentada serão levantados os pontos referentes ao ambiente de desenvolvimento utilizado, mais precisamente em relação à aquisição da licença da ferramenta Genexus.

Contudo, se a resposta de implantação for positiva, pode-se afirmar que a ferramenta de gerenciamento de artefatos contribuirá para a qualidade dos softwares da Senior.

4.1 EXTENSÕES

Para que tudo seja mantido na ferramenta, sugere-se mais implementações, tal que não se necessite mais manusear planilhas Excel:

- a) importação de planilhas de situações de casos de testes, para possíveis manutenções;
- b) importação de planilhas de massa de dados (*templates*), para possíveis manutenções em casos e situações de testes;
- c) integração da ferramenta com o *TestComplete*, para que a ferramenta *TestComplete* acesse diretamente a base de dados do Gerenciador de Artefatos e execute o teste, desta forma serão eliminadas todas planilhas *Excel*.

REFERÊNCIAS BIBLIOGRÁFICAS

APACHE. **The Apache Tomcat 5.5 Servlet**. Los Angeles, [2011?]. Disponível em: <<http://tomcat.apache.org/download-55.cgi>>. Acesso em: 17 nov. 2011.

BARTIÉ, Alexandre. **Garantia da Qualidade de Software**. Rio de Janeiro: Campus, 2002.

BASILI, Victor. **A validation of object-oriented design metrics as quality indicators**, *IEEE Transactions on Software Engineering*, vol. 22, nº10. Maryland, 1996.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. 2ed. Rio de Janeiro: Elsevier, 2007.

BIANCHINI, Juliano. **Ferramenta de apoio ao planejamento de teste funcional de software a partir de diagrama de caso de uso**. 2004. 75 f. Trabalho de Conclusão de Curso (Bacharel em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BINDER, Robert V. **Testing object-oriented systems: models, patterns and tools**. Addison-Wesley, 2000.

CARDOSO, Fernando S. **Teste de software baseado em risco**. 2008. 43 f. Trabalho de Conclusão de Curso (Curso de Ciência da Computação) – Faculdade de Jaguariúna, Jaguariúna.

CROSBY, Philip B. **Qualidade e investimento: a arte de garantir a qualidade**. Rio de Janeiro: J. Olympio, 1984. 327 p.

DEMING, W. Edwards (William Edwards). **Qualidade: a revolução da administração**. Rio de Janeiro : Marques-Saraiva, 1990. xxv, 367 p, il.

FEIGENBAUM, A. V. (Armand Vallin). **Controle da qualidade total**. São Paulo : Makron Books, 1994. 4v, il.

GENEXUS. **GeneXus for developers**. Montevideú, [2010?]. Disponível em: <<http://www.genexus.com/portal/hgxpp001.aspx?2,61,1022,O,P,0,MNU;E;232;3;MNU;>>>. Acesso em: 29 mar. 2011.

ISHIKAWA, Kaoru. **Controle de qualidade total : a maneira japonesa**. Rio de Janeiro : Campus, c1993. 221p.

MARTINS, José C. C. **Gestão de projetos de desenvolvimento de software (PMI - UML)**. Rio de Janeiro: Brasport, 2002.

MCGREGOR, J. D., SYKES, D. A. **A Practical Guide to Testing Object-Oriented Software**. New Jersey: Addison-Wesley, , 2001.

MICROSOFT. **Síntese de Funcionalidades do SQL Server 2008**. Portugal, [2010?]. Disponível em: <<http://www.microsoft.com/portugal/sql/prodinfo/features/features-at-a-glance.msp>>. Acesso em: 10 out. 2011.

MPT.BR. **MPT – Melhoria de Processo de Teste**. Recife, 2010. Disponível em: <<http://www.mpt.org.br/mptbr/>>. Acesso em: 02 abr. 2011.

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

PFLEEGER, Shari L. **Engenharia de software: teoria e prática**. Tradução Dino Franklin. Revisão Ana Regina Cavalcanti da Rocha. 2. ed. São Paulo: Prentice Hall, 2004.

PRESSMAN, Roger S. **Engenharia de software**. Tradução Rosângela Dellosso Penteado. Revisão Fernão Stella R. Germano. 6. ed. São Paulo: McGraw-Hill, 2002.

REGNELL, Björn; RUNESON, Per; WOHLIN, Claes. **Towards integration of use case modeling and usage-based testing**. [S.l.], 2003. Disponível em: <<http://business.usf.edu>>. Acesso em: 10 out. 2011.

ROCHA, Ana R. Cavalcanti da; MALDONADO, José C; WEBER, Kival C. **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.

SILVA, C. A. e HUZITA, E. H. M. (2008) DiSEN-SCV. **Uma Estratégia para Replicação de Repositórios e Alocação de Artefatos**. In: III Workshop de Desenvolvimento Distribuído de Software – WDDS, 2008, Campinas – SP.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução Maurício de Andrade. Revisão Prof. Dr Kechi Hirama. 6. ed. São Paulo: Addison Wesley, 2003.

ZIMMERMANN, Ana Paula. **Ferramenta para automação de testes de caixa preta**. 2006. 101 f. Trabalho de Conclusão de Curso (Curso de Ciência da Computação) – Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, Itajaí.

APÊNDICE A – Descrição dos casos de uso

Nos quadros de 03 a 23 estão as descrições dos casos de uso do sistema.

<p>Caso de Uso – Logar no sistema Ator: Usuário Objetivo: Acessar aplicação via navegador Internet e informa dados para login e senha armazenados no cadastro de usuário ou usuário <i>admin</i>. Pré-condições: Sistema deve estar hospedado no servidor <i>web</i>; Usuário deve estar cadastrado no banco de dados ou deve ser <i>admin</i> Pós-condições: Usuário entra conectado ao sistema</p> <p>Cenário Principal: 1. Usuário preenche seu <i>login</i> e sua senha; 2. Sistema valida os dados de <i>login</i> e senha do usuário; 3. Sistema direciona o Usuário para a página de menu principal da ferramenta.</p> <p>Cenário Alternativo: No passo 2, o sistema não reconhece os dados do usuário no sistema</p> <p>2.1 Sistema apresenta mensagem de erro “Usuário e/ou senha inválida” 2.2 Sistema retorna para página de <i>login</i></p>
--

Quadro 3 - Descrição do caso de uso *Logon* no sistema

<p>Caso de Uso – Manter Usuários Ator: Analista de Qualidade Objetivo: Acessar tela de Usuários para manter os dados dos usuários. Pré-condições: Analista de Qualidade deve fazer o <i>login</i> no sistema. Pós-condições: Analista de Qualidade editou, excluiu ou cadastrou um usuário.</p> <p>Cenário Principal: 1. Sistema mostra a tela de usuários; 2. Analista de qualidade opta por editar cadastro de usuário 3. Analista de qualidade digita o código do usuário 4. Sistema apresenta os dados do usuário selecionado 5. Analista de qualidade edita os campos necessários 6. Analista pressiona o botão salvar 7. Sistema salva as alterações efetuadas 8. Sistema apresenta cadastro atualizado na tela</p> <p>Cenário Alternativo: No passo 2, o analista de qualidade opta por incluir cadastro</p> <p>2.1 Analista de qualidade digita um novo código de usuário 2.2 Analista de qualidade preenche os dados da tela de usuário 2.3 Analista pressiona o botão salvar 2.4 Sistema salva as alterações efetuadas 2.5 Sistema apresenta cadastro incluso na tela</p> <p>Cenário Alternativo: No passo 2, o analista de qualidade opta por excluir cadastro</p> <p>2.1 Analista de qualidade digita o código do usuário 2.2 Sistema apresenta os dados do usuário selecionado</p>
--

- 2.3 Analista de qualidade pressiona o botão excluir
 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
 2.5 Analista de qualidade pressiona sim.
 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 4– Descrição do caso de uso Manter Usuários.

Caso de Uso - Importar requisitos

Ator: Analista de Qualidade

Objetivo: Acessar tela Importar Planilha, seleciona o campo de Requisitos e seleciona a planilha a ser importada.

Pré-condições: Sistema deverá ter o módulo cadastrado; Sistema deverá ter as planilhas de suíte de testes importadas.

Pós-condições: Sistema apresenta a mensagem "Planilha importada com sucesso!".

Cenário Principal:

1. Usuário seleciona o campo de Requisito;
2. Usuário seleciona a planilha de requisito a ser importada pelo botão Procurar;
3. Usuário pressiona botão Importar para importar a planilha

Cenário Alternativo:

No passo 2, a planilha selecionada já foi importada

- 2.1 Sistema apresenta mensagem “Planilha já importada!”

Quadro 5– Descrição do caso de uso Importar Requisito

Caso de Uso - Importar suítes de testes

Ator: Analista de Qualidade

Objetivo: Acessar tela Importar Planilha, seleciona o campo de Suíte de testes e seleciona a planilha a ser importada.

Pré-condições: Sistema deverá ter o módulo cadastrado;

Pós-condições: Sistema apresenta a mensagem "Planilha importada com sucesso!".

Cenário Principal:

1. Usuário seleciona o campo de Suíte de testes;
2. Usuário seleciona a planilha de Suíte de testes a ser importada pelo botão Procurar;
3. Usuário pressiona botão Importar para importar a planilha

Cenário Alternativo:

No passo 2, a planilha selecionada já foi importada

- 2.1 Sistema apresenta mensagem “Planilha já importada!”

Quadro 6 – Descrição do caso de uso Importar Suíte de Testes

Caso de Uso – Manter Requisitos

Ator: Analista de Qualidade

Objetivo: Acessar tela de Requisitos para manter os dados dos requisitos.

Pré-condições: Analista de Qualidade deve fazer o login no sistema; Sistema deverá ter a rotina cadastrada; Sistema deverá ter o item cadastrado

Pós-condições: Analista de Qualidade editou, excluiu ou cadastrou um requisito

Cenário Principal:

1. Sistema mostra a tela de requisitos;
2. Analista de qualidade opta por editar cadastro de requisito
3. Analista de qualidade digita o código do requisito
4. Sistema apresenta os dados do requisito selecionado
5. Analista de qualidade edita os campos necessários

6. Analista pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por incluir cadastro

- 2.1 Analista de qualidade digita um novo código de requisito
- 2.2 Analista de qualidade preenche os dados da tela de requisito
- 2.3 Analista pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por excluir cadastro

- 2.1 Analista de qualidade digita o código do requisito
- 2.2 Sistema apresenta os dados do requisito selecionado
- 2.3 Analista de qualidade pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Analista de qualidade pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 7 – Descrição do caso de uso Manter Requisitos.

Caso de Uso – Manter Grupos**Ator:** Analista de Qualidade**Objetivo:** Acessar tela de Grupos para manter os dados dos grupos**Pré-condições:** Analista de Qualidade deve fazer o login no sistema; Sistema deverá ter o módulo cadastrado; Sistema deverá ter o item cadastrado**Pós-condições:** Analista de Qualidade editou, excluiu ou cadastrou um grupo**Cenário Principal:**

1. Sistema mostra a tela de grupos;
2. Analista de qualidade opta por editar cadastro de grupo
3. Analista de qualidade digita o código do grupo
4. Sistema apresenta os dados do grupo selecionado
5. Analista de qualidade edita os campos necessários
6. Analista pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por incluir cadastro

- 2.1 Analista de qualidade digita um novo código de grupo
- 2.2 Analista de qualidade preenche os dados da tela de grupo
- 2.3 Analista pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por excluir cadastro

- 2.1 Analista de qualidade digita o código do grupo
- 2.2 Sistema apresenta os dados do grupo selecionado
- 2.3 Analista de qualidade pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Analista de qualidade pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 8 – Descrição do caso de uso Manter Grupos.

Caso de Uso – Manter Sub Grupos**Ator:** Analista de Qualidade**Objetivo:** Acessar tela de Sub Grupos para manter os dados dos sub grupos**Pré-condições:** Analista de Qualidade deve fazer o login no sistema; Sistema deverá ter grupo cadastrado;**Pós-condições:** Analista de Qualidade editou, excluiu ou cadastrou um sub grupo**Cenário Principal:**

1. Sistema mostra a tela de sub grupos;
2. Analista de qualidade opta por editar cadastro de sub grupo
3. Analista de qualidade digita o código do sub grupo
4. Sistema apresenta os dados do sub grupo selecionado
5. Analista de qualidade edita os campos necessários
6. Analista pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por incluir cadastro

- 2.1 Analista de qualidade digita um novo código de sub grupo
- 2.2 Analista de qualidade preenche os dados da tela de sub grupo

- 2.3 Analista pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por excluir cadastro

- 2.1 Analista de qualidade digita o código do sub grupo
- 2.2 Sistema apresenta os dados do sub grupo selecionado
- 2.3 Analista de qualidade pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Analista de qualidade pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 9 – Descrição do caso de uso manter Sub Grupos.

Caso de Uso – Manter Itens

Ator: Analista de Qualidade

Objetivo: Acessar tela de Itens para manter os dados dos itens

Pré-condições: Analista de Qualidade deve fazer o login no sistema; Sistema deverá ter o sub grupo cadastrado; Sistema deverá ter o item cadastrado

Pós-condições: Analista de Qualidade editou, excluiu ou cadastrou um item

Cenário Principal:

- 1. Sistema mostra a tela de item;
- 2. Analista de qualidade opta por editar cadastro de item
- 3. Analista de qualidade digita o código do item
- 4. Sistema apresenta os dados do item selecionado
- 5. Analista de qualidade edita os campos necessários
- 6. Analista pressiona o botão salvar
- 7. Sistema salva as alterações efetuadas
- 8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por incluir cadastro

- 2.1 Analista de qualidade digita um novo código de item
- 2.2 Analista de qualidade preenche os dados da tela de item
- 2.3 Analista pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por excluir cadastro

- 2.1 Analista de qualidade digita o código do item
- 2.2 Sistema apresenta os dados do item selecionado
- 2.3 Analista de qualidade pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Analista de qualidade pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 10 – Descrição do caso de uso Manter Item

Caso de Uso – Manter Módulos

Ator: Analista de Qualidade

Objetivo: Acessar tela de Módulos para manter os dados dos módulos

Pré-condições: Analista de Qualidade deve fazer o login no sistema;
Pós-condições: Analista de Qualidade editou, excluiu ou cadastrou um módulo

Cenário Principal:

1. Sistema mostra a tela de módulo
2. Analista de qualidade opta por editar cadastro de módulo
3. Analista de qualidade digita o código do módulo
4. Sistema apresenta os dados do módulo selecionado
5. Analista de qualidade edita os campos necessários
6. Analista pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por incluir cadastro

- 2.1 Analista de qualidade digita um novo código de módulo
- 2.2 Analista de qualidade preenche os dados da tela de módulo
- 2.3 Analista pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por excluir cadastro

- 2.1 Analista de qualidade digita o código do módulo
- 2.2 Sistema apresenta os dados do módulo selecionado
- 2.3 Analista de qualidade pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Analista de qualidade pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 11 - Descrição de caso de uso Manter Módulos

Caso de Uso – Manter Rotinas

Ator: Analista de Qualidade

Objetivo: Acessar tela de Rotinas para manter os dados das rotinas

Pré-condições: Analista de Qualidade deve fazer o login no sistema;

Pós-condições: Analista de Qualidade editou, excluiu ou cadastrou uma rotina

Cenário Principal:

1. Sistema mostra a tela de rotina
2. Analista de qualidade opta por editar cadastro de rotina
3. Analista de qualidade digita o código da rotina
4. Sistema apresenta os dados da rotina selecionada
5. Analista de qualidade edita os campos necessários
6. Analista pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por incluir cadastro

- 2.1 Analista de qualidade digita um novo código de rotina
- 2.2 Analista de qualidade preenche os dados da tela de rotina
- 2.3 Analista pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por excluir cadastro

- 2.1 Analista de qualidade digita o código da rotina
- 2.2 Sistema apresenta os dados da rotina selecionada
- 2.3 Analista de qualidade pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Analista de qualidade pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 12 – Descrição do caso de uso Manter Rotinas

Caso de Uso – Manter Suítes de Testes

Ator: Analista de Qualidade

Objetivo: Acessar tela de Suítes para manter os dados das suítes

Pré-condições: Analista de Qualidade deve fazer o login no sistema;

Pós-condições: Analista de Qualidade editou, excluiu ou cadastrou uma suíte

Cenário Principal:

1. Sistema mostra a tela de suíte
2. Analista de qualidade opta por editar cadastro de suíte
3. Analista de qualidade digita o código da suíte
4. Sistema apresenta os dados da suíte selecionada
5. Analista de qualidade edita os campos necessários
6. Analista pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por incluir cadastro

- 2.1 Analista de qualidade digita um novo código de suíte
- 2.2 Analista de qualidade preenche os dados da tela de suítes
- 2.3 Analista pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o analista de qualidade opta por excluir cadastro

- 2.1 Analista de qualidade digita o código da suíte
- 2.2 Sistema apresenta os dados da suíte selecionada
- 2.3 Analista de qualidade pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Analista de qualidade pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 13 – Descrição do caso de uso Manter Suítes de Testes

Caso de Uso – Manter Casos de Testes

Ator: Usuário

Objetivo: Acessar tela de Casos para manter os dados das casos

Pré-condições: Usuário deve fazer o login no sistema;

Pós-condições: Usuário editou, excluiu ou cadastrou uma casos

Cenário Principal:

1. Sistema mostra a tela de casos
2. Usuário opta por editar cadastro de caso
3. Usuário digita o código da caso
4. Sistema apresenta os dados da caso selecionada

5. Usuário edita os campos necessários
6. Usuário pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o Usuário opta por incluir cadastro

- 2.1 Usuário digita um novo código de caso
- 2.2 Usuário preenche os dados da tela de casos
- 2.3 Usuário pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o Usuário opta por excluir cadastro

- 2.1 Usuário digita o código da caso
- 2.2 Sistema apresenta os dados da caso selecionada
- 2.3 Usuário pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Usuário pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 14 – Descrição do caso de uso Manter Casos de Testes

Caso de Uso – Manter Situações de Testes**Ator:** Analista de Qualidade**Objetivo:** Acessar tela de Situações para manter os dados das situações**Pré-condições:** Usuário deve fazer o login no sistema;**Pós-condições:** Usuário editou, excluiu ou cadastrou uma situação**Cenário Principal:**

1. Sistema mostra a tela de situação
2. Usuário opta por editar cadastro de situação
3. Usuário digita o código da situação
4. Sistema apresenta os dados da situação selecionada
5. Usuário edita os campos necessários
6. Usuário pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o Usuário opta por incluir cadastro

- 2.1 Usuário digita um novo código de situação
- 2.2 Usuário preenche os dados da tela de situação
- 2.3 Usuário pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o Usuário opta por excluir cadastro

- 2.1 Usuário digita o código da situação
- 2.2 Sistema apresenta os dados da situação selecionada
- 2.3 Usuário pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Usuário pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 15 – Descrição do caso de uso Manter Situações de Testes

Caso de Uso – Manter Vínculos entre artefatos**Ator:** Usuário**Objetivo:** Acessar tela de Vincular Artefatos para manter os dados dos vínculos**Pré-condições:** Usuário deve fazer o login no sistema; Sistema deverá ter requisitos cadastrados; Sistema deverá ter suítes de testes cadastradas; Sistema deverá ter casos de testes cadastrados**Pós-condições:** Usuário editou, excluiu ou cadastrou um vínculo**Cenário Principal:**

1. Sistema mostra a tela de vínculos
2. Usuário opta por editar cadastro de vínculo
3. Usuário digita o código do vínculo
4. Sistema apresenta os dados do vínculo selecionado
5. Usuário edita os campos necessários
6. Usuário pressiona o botão salvar
7. Sistema salva as alterações efetuadas
8. Sistema apresenta cadastro atualizado na tela

Cenário Alternativo:

No passo 2, o Usuário opta por incluir cadastro

- 2.1 Usuário digita um novo código de vínculo
- 2.2 Usuário preenche os dados da tela de vínculo

- 2.3 Usuário pressiona o botão salvar
- 2.4 Sistema salva as alterações efetuadas
- 2.5 Sistema apresenta cadastro incluso na tela

Cenário Alternativo:

No passo 2, o Usuário opta por excluir cadastro

- 2.1 Usuário digita o código da vínculo
- 2.2 Sistema apresenta os dados do vínculo selecionado
- 2.3 Usuário pressiona o botão excluir
- 2.4 Sistema apresenta mensagem “Deseja realmente excluir registro?”
- 2.5 Usuário pressiona sim.
- 2.6 Sistema apresenta o código seguinte cadastrado

Quadro 16 – Descrição do caso de uso Manter Vínculos entre artefatos.

Caso de Uso: Exportar Casos de Testes

Ator: Usuário

Objetivo: Acessar tela de Casos de testes e exporta o caso cadastrado

Pré-condições: Sistema deverá ter caso de testes cadastrado; Sistema deverá ter situações de testes cadastradas;

Pós-condições: Exportar disponível na tela de Casos de Testes; Sistema apresenta a mensagem "Planilha exportada com sucesso!"

Cenário Principal:

1. O sistema mostra a tela de Casos de Testes;
2. Usuário seleciona o caso de teste a ser exportado;
3. Usuário exporta para planilha o caso de testes selecionado, pressionando o botão

Quadro 17 – Descrição do caso de uso Exportar Casos de Testes.

Caso de Uso: Emitir relatório de Requisitos Cadastrados

Ator: Usuário

Objetivo: Acessar a tela de requisitos no menu Relatórios, Requisitos.

Pré-condições: Sistema deverá ter requisito cadastrado.

Pós-condições: Usuário visualizou relatório dos requisitos cadastrados na ferramenta

Cenário Principal:

1. Usuário seleciona relatório de Requisitos no menu Relatórios, Requisitos;
2. Usuário seleciona o requisito à ser listado.
3. Sistema lista relatório dos requisitos.

Quadro 18 – Descrição do caso de uso Emitir relatório de Requisitos Cadastrados

Caso de Uso: Emitir relatório de Requisitos por Data

Ator: Usuário

Objetivo: Acessar a tela de requisitos por data no menu Relatórios, Requisitos por Data.

Pré-condições: Sistema deverá ter requisito cadastrado.

Pós-condições: Usuário visualizou relatório dos requisitos pelo período informado

Cenário Principal:

1. Usuário seleciona relatório de Requisitos por Data no menu Relatórios, Requisitos
2. Usuário seleciona o data inicial e final na tela de entrada do relatório
3. Sistema lista relatório dos requisitos por data.

Quadro 19 – Descrição do caso de uso Emitir relatório de Requisitos por Data.

Caso de Uso: Emitir relatório de Vínculos entre Requisitos e Casos de testes**Ator:** Usuário**Objetivo:** Acessar a tela de Requisitos x Casos de Testes no menu Relatórios, Requisitos**Pré-condições:** Sistema deverá ter requisito cadastrado; Sistema deverá ter casos de testes cadastrados**Pós-condições:** Usuário visualizou relatório dos requisitos com casos pertencentes**Cenário Principal:**

1. Usuário seleciona relatório de Requisitos Casos de Testes no menu Relatórios, Requisitos
2. Usuário seleciona o requisito à ser listado.
3. Sistema lista relatório dos requisitos com os casos pertencentes.

Quadro 20 – Descrição do caso de uso Emitir relatório de Vínculos entre Requisitos e Casos de Testes.

Caso de Uso: Emitir relatório de Vínculos entre Requisitos, Casos de testes e Massa de Dados**Ator:** Usuário**Objetivo:** Acessar a tela de Requisitos x Casos de Testes x Massa de Dados no menu Relatórios, Requisitos**Pré-condições:** Sistema deverá ter requisito cadastrado; Sistema deverá ter casos de testes cadastrados; Sistema deverá ter situações de testes cadastradas**Pós-condições:** Usuário visualizou relatório dos requisitos com casos pertencentes**Cenário Principal:**

1. Usuário seleciona relatório de Requisitos Casos de Testes no menu Relatórios, Requisitos;
2. Usuário seleciona o requisito à ser listado.
3. Sistema lista relatório dos requisitos com os casos e massa de dados pertencentes.

Quadro 21 – Descrição do caso de uso Emitir relatório de Vínculos entre Requisitos, Casos de Testes e Massa de Dados.

Caso de Uso: Emitir relatório de Requisitos por Rotina**Ator:** Usuário**Objetivo:** Acessar a tela de Requisitos por rotina no menu Relatórios, Requisitos**Pré-condições:** Sistema deverá ter requisito cadastrado.**Pós-condições:** Usuário visualizou relatório dos requisitos com as suas rotinas**Cenário Principal:**

1. Usuário seleciona relatório de Requisitos por Rotina no menu Relatórios, Requisitos
2. Usuário seleciona o rotina à ser listado.
3. Sistema lista relatório das rotinas com os casos pertencentes

Quadro 22 – Descrição do caso de uso Emitir relatório de Requisitos por Rotina.

Caso de Uso: Emitir relatório de Massa de dados pertencentes a Casos de testes**Ator:** Usuário**Objetivo:** Acessar a tela de Casos de testes x Massa de Dados no menu Relatórios, Casos de testes**Pré-condições:** Sistema deverá ter caso cadastrado; Sistema deverá ter situações de testes cadastradas**Pós-condições:** Usuário visualizou relatório dos casos de testes com massa de dados pertencentes**Cenário Principal:**

1. Usuário seleciona relatório de Casos de Testes x Massa de dados no menu Relatórios, Casos de testes x Massa de dados
2. Usuário seleciona o caso de teste à ser listado.
3. Sistema lista relatório as massa de dados com os casos pertencentes.

Quadro 23 – Descrição do caso de uso Emitir relatório de Massa de Dados

APÊNDICE B – Dicionário de Dados

Nos quadros de 24 a 36 estão o dicionário de dados das tabelas do sistema.

CASOTESTE – Armazena dados de casos de testes cadastrados e/ou importados.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
CasoID	Código do caso	Numeric	4	Sim
CasoNomePlanilha	Nome Planilha do caso	Character	80	Não
SuiteID	Suíte de testes	Numeric	4	Não
CasoNome	Descrição do caso	Character	400	Não
CasoObjetivo	Objetivo do caso	Character	400	Não

Quadro 24 – Tabela Caso Teste

GRUPO – Armazena dados do grupo, que são os menus das telas do sistema a ser testado.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
GrupoID	Código do grupo	Numeric	4	Sim
GrupoDescricao	Descrição do grupo	Character	50	Não

Quadro 25 – Tabela Grupo

GRUPOMODULO – Armazena os grupos do módulo.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
GrupoID	Código do grupo	Numeric	4	Sim
ModuloID	Módulo	Numeric	200	Sim

Quadro 26 – Tabela GrupoModulo

ITEM – Armazena dados do item, que são as telas do sistema a ser testado.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
ItemID	Código do item	Numeric	4	Sim
ItemDescricao	Descrição do item	Character	100	Não
SubGrupoID	Sub Grupo	Numeric	4	Não

Quadro 27 – Tabela Item

MODULO – Armazena dados do módulo.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
ModuloID	Código do módulo	Numeric	4	Sim
ModuloDescricao	Descrição do módulo	Character	20	Não
ModuloSigla	Sigla do módulo	Character	2	Não

Quadro 28 – Tabela Módulo

REQUISITO – Armazena dados de requisitos cadastrados e/ou importados.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
RequisitoID	Código do requisito	Numeric	4	Sim
RequisitoSequencia	Sequência	Numeric	4	Não
RotinaID	Rotina	Numeric	4	Não
RequisitoDescricao	Descrição do requisito	Character	1500	Não
ItemID	Item	Numeric	4	Não
SubGrupoID	Sub Grupo	Numeric	4	Não
GrupoID	Grupo	Numeric	4	Não
DataAlt	Data Alteração	Date		Não

Quadro 29 – Tabela Requisito

REQUISITOCASOTESTE – Armazena os vínculos entre requisito e caso de testes.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
RequisitoCasoID	Código do vínculo	Numeric	4	Sim
RequisitoID	Requisito	Numeric	4	Não
SuiteID	Suíte de testes	Numeric	4	Não
CasoID	Caso de testes	Numeric	4	Não

Quadro 30 – Tabela Requisito Caso Teste

ROTINA – Armazena dados da rotina, que se refere a uma parte do módulo que será testado.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
RotinaID	Código da rotina	Numeric	4	Sim
ModuloID	Módulo	Numeric	4	Não
RotinaDescricao	Descrição da rotina	Character	20	Não
RotinaNomePlanilha	Nome Planilha por rotina	Character	200	Não

Quadro 31 – Tabela Rotina

SITUACAOTESTE – Armazena dados da situação de testes.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
SituacaoID	Código da situação	Numeric	4	Sim
CasoID	Caso de testes	Numeric	4	Não
CasoExecutar	Executar	Character	1	Não

Quadro 32 – Tabela Situação Teste

SITUACAOTESTEPASSOS – Armazena dados dos passos da situação de testes.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
SituacaoID	Situação	Numeric	4	Sim
PassoID	Passo	Numeric	4	Sim
CasoCondicaoTeste	Condição de teste	Character	400	Não
CasoAplicacao	Aplicação	Character	20	Não
CasoAmbiente	Ambiente	Character	3	Não
CasoMassaDados	Massa de dados	Character	30	Não
CasoPasta	Pasta	Character	30	Não
CasoCodMassaDados	Código Massa de dados	Character	20	Não
CasoAcao	Ação	Character	1	Não
CasoSituacao	Situação de testes	Character	300	Não

Quadro 33 - Tabela Situação Teste Passos

SUBGRUPO – Armazena dados do sub grupo, que se refere a um sub menu.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
SubGrupoID	Código do sub grupo	Numeric	4	Sim
SubGrupoDescricao	Descrição do sub grupo	Character	50	Não
GrupoID	Código Grupo	Numeric	4	Não

Quadro 34 – Tabela Sub grupo

SUÍTE – Armazena dados de suítes cadastradas e/ou importadas.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
SuiteID	Código da suíte	Numeric	4	Sim
SuiteNomePlanilha	Nome Planilha de suíte	Character	200	Não
SuiteNome	Nome da suíte	Character	200	Não
SuiteObjetivo	Objetivo da suíte	Character	400	Não
SuitePreCondicoes	Pré condições	Character	400	Não

Quadro 35 – Tabela Suíte

USUÁRIO – Armazena dados de usuário.				
Campo	Descrição	Tipo	Tamanho	Chave Primária
UsuarioID	Código do usuário	Numeric	4	Sim
UsuarioNome	Nome do usuário	Character	40	Não
UsuarioLogin	Login	Character	30	Não
UsuarioEmail	Email	Character	40	Não
UsuarioSenha	Senha	Character	12	Não
UsuarioPapel	Papel do usuário	Numeric	1	Não
RotinaID	Rotina	Numeric	4	Não

Quadro 36 – Tabela Usuário