

**Instituto de Pesquisas Tecnológicas do Estado de São Paulo**

**Bruno Panicucci Pinheiro de Castro**

**Método para Implantação Gradativa do Processo de Formalização  
de Casos de Teste**

**São Paulo  
2014**

Bruno Panicucci Pinheiro de Castro

Método para Implantação Gradativa do Processo de Formalização de Casos de Teste

Dissertação de Mestrado apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Computação.

Data da aprovação \_\_\_\_/\_\_\_\_/\_\_\_\_

---

Prof. Dr. Plinio Thomaz Aquino (Orientador)  
Mestrado em Engenharia de Computação

Membros da Banca Examinadora:

Prof. Dr. Plinio Thomaz Aquino Junior (Orientador)  
Mestrado em Engenharia de Computação

Prof. Dr. Rodrigo Filev Maia (Membro)  
FEI – Centro Universitário da FEI

Prof. Dra. Gabriela Barbaran (Membro)  
FEI – Centro Universitário da FEI

Bruno Panicucci Pinheiro de Castro

Método para Implantação Gradativa do Processo de Formalização de Casos de  
Teste

Dissertação de Mestrado apresentado ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Computação.

Área de Concentração: Engenharia de Software

Orientador: Prof. Dr. Plinio T. Aquino Junior

São Paulo  
Abril / 2014

## **AGRADECIMENTOS**

Aos meus pais e irmã que incentivaram minha iniciativa e esforço ao longo de todos os dias investidos neste trabalho. À minha namorada e amigos que se colocaram à disposição para toda ajuda e apoio que precisei ao longo desta fase de dedicação.

Agradeço também ao meu orientador, Professor Doutor Plinio por toda atenção dedicada, esforço e apoio investido que tornaram possível a conclusão desta pesquisa.

## RESUMO

O conceito de automação é normalmente considerado em ambientes, onde há esforço manual repetitivo em tarefas como construção e execução de testes. As condições para a aplicação de técnicas que empreguem estes conceitos requerem modelos e padronização nos dados para sua efetiva implantação e execução. Como consequência, pequenas empresas enfrentam dificuldades para se adaptar e absorver estas mudanças de comportamento, além dos investimentos envolvidos no decorrer do processo de implantação. Este trabalho possui como propósito um método capaz de automatizar o processo de construção de casos de testes, utilizando como formato de implantação, uma abordagem flexível, onde empresas adotam gradativamente as características e regras requisitadas pelo método. A técnica de representação de requisitos de negócio por cenários é responsável por gerar os dados iniciais consumidos pelo método. Para tanto, o método propõe um modelo para a tradução de cenários de requisitos em componentes de testes e gera como resultados, artefatos que contém os fluxos de execução dos testes e as condições a serem validadas. Estes artefatos são submetidos aos especialistas de testes e do negócio, no intuito de localizar os testes que possam incluir um número maior de validações. O método é executado de forma cíclica, onde os resultados são utilizados para novas execuções, visando casos de teste cada vez mais completos. Entre as contribuições deste trabalho destaca-se uma alternativa para a implantação de conceitos de automação nos processos de empresas, levando em conta as restrições de recursos e propondo uma técnica que possa ser implantada de forma gradativa. Além disso, o método tem como resultados, documentos de teste que sigam um padrão de escrita comum a todos os envolvidos do projeto, além de uma representação ilustrativa do escopo do projeto sob a ótica de validações e testes requeridos.

**Palavras-Chave:** Automação de testes; testes baseados em modelos; requisitos baseados em cenários; geração de casos de teste

## **ABSTRACT**

### **Method for Gradual Implementation of Test Cases Formalization Process**

The automation concept is often used in environments where exists repetitive task efforts as testing documentation and execution. The conditions for the technique application which uses this concept demand models and data standardization in order to achieve a successful implantation and execution. As a consequence small companies face challenges to adapt itself and absorb the behaviour change. Besides that there is the investement of resources through the process of implantation. The purpose of this research is develop a method to automate the test cases documentation. The method implies a model responsible for the translation of requirements scenarios to test componentes and results in artifacts that contains the process of test execution and its conditions to be validated. These artifacts are delivered to testing specialists that can include a bigger number of testing validations. The method is executed on cyclic process where the last results are used as a source for the new executions, in order to achieve better results to the test case. Among the contributions of this research highlights an alternative for the implantation of automation concepts in companies processes, taking into account the resources constraints and proposing a technique that may be implanted gradually. As for the method results, there is a testing documentatios that follows a writing pattern and it can be used for everyone involved in the project. Also, it includes a illustrative representation of the project scope from the viewpoint of required tests.

**Key-Words:** Automated testing; model based testing; scenario based requirements; test cases generation

## Lista de Ilustrações

Diagrama 1	Método de Automação de Casos de Teste	32
Figura 1	Exemplos de Visão para o Cenário	17
Figura 2	Comparação de Esforço em Horas	25
Figura 3	Detalhes de um Estado do Cenário	34
Figura 4	Personalizando um Cenário	34
Figura 5	Episódio do Cenário representado por Componentes	35
Figura 6	Conjunto de Componentes por Cenário de Requisito	36
Figura 7	Modelo de Geração de Artefatos de Teste	39
Figura 8	Adaptação da Árvore de Cenários	40
Figura 9	Árvore de Cenários da Simulação 1	46
Figura 10	Árvore de Cenários da Simulação 2	54
Figura 11	Resumo de Tempo Total de Execução em Minutos	62

## Lista de Tabelas

Tabela 1	Formulário de Personalização de Testes	37
Tabela 2	Personalização de Variáveis	38
Tabela 3	Passos de Execução da Simulação	41
Tabela 4	Componentes selecionados para os Cenários	43
Tabela 5	Questionário do Cenário	43
Tabela 6	Cenário de Teste e Variáveis	44
Tabela 7	Cenários Aplicados em cada passo de execução	48
Tabela 8	Respostas para Questionários da Simulação 2	49
Tabela 9	Cenários de Testes e Variáveis de Personalização	50
Tabela 10	Questionários Atualizados na Segunda Iteração	51
Tabela 11	Cenários Aplicados em cada passo de execução	55
Tabela 12	Questionários Atualizados na Segunda Iteração	56
Tabela 13	Casos de Teste Adicionados na Segunda Iteração	56
Tabela 14	Resumo de Tempo de Execução da Primeira Simulação	61
Tabela 15	Resumo de Tempo de Execução da Segunda Simulação	62
Tabela 16	Comparação de Execuções dos Testes nas duas Situações	64



## Sumário

<b>1 INTRODUÇÃO</b>	<b>10</b>
1.1 Motivação	10
1.2 Objetivos	12
1.3 Contribuições da Pesquisa	13
1.4 Etapas de Trabalho	14
1.5 Organização do Trabalho	16
<b>2 ESTADO DA ARTE</b>	<b>17</b>
2.1 Abordagens para Elicitação de Requisitos	17
2.2 Processos e Formalizações	20
2.3 Testes e Técnicas de Automação	21
2.4 Trabalhos Relacionados	26
2.4.1 Aplicação de Modelos e Casos de Teste	26
2.4.2 Vantagens na Aplicação de Cenários de Requisitos	30
<b>3 MÉTODO PARA IMPLANTAÇÃO GRADATIVA DO PROCESSO DE FORMALIZAÇÃO DE CASOS DE TESTE</b>	<b>35</b>
3.1 Extração de Dados dos Cenários	36
3.2 Modelo de Conversão de Componentes de Teste	39
3.3 Automação de Casos de Teste - Personalização dos Cenários	40
3.4 Modelo de Geração dos Documentos	41
<b>4 SIMULAÇÃO DO MÉTODO</b>	<b>44</b>
4.1 Passos de Execução do Método	44
4.2 Validações Sistêmicas – Entrada/Saída de Estoque	44
4.3 Validações de Interface e Operação – Projeção e Gestão de Custos	47
4.4 Validações de Regras de Negócio – Integração de Sistemas	54
<b>5 ANÁLISE DOS RESULTADOS</b>	<b>61</b>
5.1 Simulações Práticas	61

5.2 Simulações em Números .....	62
5.3 Análise dos Resultados .....	65
<b>6 CONCLUSÕES .....</b>	<b>68</b>
<b>7 EXTENSÕES DE PESQUISA E TRABALHOS FUTUROS .....</b>	<b>71</b>
<b>REFERÊNCIAS .....</b>	<b>73</b>
<b>APÊNDICES .....</b>	<b>75</b>
Apêndice A .....	76
Apêndice B .....	85
Apêndice C .....	95

# 1 INTRODUÇÃO

## 1.1 Motivação

Os conceitos de automação e os benefícios gerados a partir da aplicação de suas técnicas e ferramentas são assuntos abordados nas empresas como soluções para problemas que envolvem excesso de trabalhos repetitivos ou tarefas complexas para execuções manuais. As vantagens oferecidas envolvem a redução de esforço e custo, além de melhorar o desempenho e os resultados obtidos. Graham e Fewster (2012) apontam que além dos benefícios citados acima, processos de automação oferecem como promessa a simplificação de operações no ambiente de trabalho. No cenário das empresas de tecnologia estão disponíveis padrões de sistemas, ferramentas de testes e geração de documentações formais. Suas aplicações buscam adotar padrões de organização, escrita e formato com o objetivo de acelerar a construção de documentos e evitar problemas de comunicação e entendimento. Segundo Posner (1987), escopos mal definidos e os pontos citados anteriormente fazem parte da lista de principais problemas que gestores de projetos precisam lidar.

Na área de qualidade e testes de software, um caso de teste é um conjunto de condições, entradas e passos de execução que possuem como objetivo a validação de uma dada situação de teste. A padronização de documentações como estas auxiliam a velocidade com que casos de testes são produzidos e a quantidade de informações que são disponibilizadas para a construção dos documentos, além de sua organização para posterior execução dos testes. Estudos na implantação de novos sistemas, manutenções em sistemas existentes e aplicação de novas ferramentas no ambiente de trabalho foram feitos para comprovar os benefícios de aplicações práticas dos conceitos de automação, entretanto nem todas as implantações obtém sucesso, conforme apresentado por Graham e Fewster (2012). Estes estudos obtiveram resultados negativos devido à má definição de expectativas, por fatores de alocações ineficazes de recursos como dinheiro, tempo e profissionais aptos, além de impaciência para averiguar resultados significantes e falta de entendimento nos conceitos de automação por parte da empresa.

A taxa de sucesso na implantação destas diversas técnicas de automação tende a ser mais elevada quando a empresa atinge certo nível de maturidade, quando processos e padrões estão estabelecidos dentro da rotina de trabalho e o

sentimento da empresa esteja voltado para o aumento do desempenho de seus processos, ao invés da criação ou substituição por novos. Também se deve levar em consideração que empresas de maior nível de maturidade têm interesse em investir nestas técnicas visando ganhos de desempenho e qualidade a médio ou longo prazo. Entretanto, a situação não é a mesma para empresas menores, segundo Nebut, Fleurey, Traon e Jezequel (2006). Projetos que possuem restrições orçamentárias enfrentam dificuldades com a utilização de recursos para elaboração de documentos formais, sejam de planejamento, requisitos ou testes. Embora sejam apresentados significativos benefícios para as empresas, estas possuem certo receio acerca da adoção de técnicas disponíveis no mercado, já que sua implantação requer mudanças em seu comportamento atual e futuro, além da forma como seus processos são executados. Nebut, Fleurey, Traon e Jezequel (2006) também adiciona aos fatores que geram resistência por parte das empresas, a necessidade do investimento financeiro para cada uma das medidas citadas acima, tanto para implantação como a posterior execução e manutenção.

Graham e Fewster (2012) citam alguns aspectos que são relevantes no momento da adoção de um método de automação e sua subsequente implantação. Entre os fatores ressalta-se o apoio da gestão do projeto, métricas definidas para os resultados do processo de automação, profissionais com diferentes níveis de habilidades, investimentos em ferramentas e treinamentos. Em função dos ambientes em que estas características apresentadas por Graham e Fewster (2012) surgem em maior quantidade, justifica-se o fato que métodos de automação apresentam resultados mais efetivos em empresas maiores e com estrutura organizacional mais sólida.

No entanto, métodos que se adaptem ao ambiente alvo da implantação estão aptos a se esquivar de obstáculos comuns dos cenários de pequenas empresas, conforme defendido por Nebut, Fleurey, Traon e Jezequel (2006). Portanto, o fator resultante pelo tamanho da empresa não se mostra um ponto de restrição para a implantação de um método de automação, mas sim o grau de flexibilidade que um dado método deve possuir para sua implantação.

O ponto de partida para técnicas e métodos efetivos de automação de testes é a construção de modelos que sejam reutilizáveis. Estes devem ser objetivos e garantir

que todos os envolvidos na execução entendam seu conceito. Nebut, Fleurey, Traon e Jezequel (2006) mostra a construção de um modelo e posterior aplicação em um processo automatizado de geração de testes, fazendo uso de informações obtidas em casos de usos. A integração entre os dois processos (levantamento de requisitos e construção de testes) ocorre na aplicação do método que contém o modelo proposto. A formalização do processo de levantamento de requisitos com a aplicação do modelo proposto por Nebut, Fleurey, Traon e Jezequel (2006) tem como resultados artefatos para o processo de testes. Entre os projetos em que houve implantações de métodos formais e automatização de processos de testes pode-se identificar algumas características comuns. Nebut, Fleurey, Traon e Jezequel (2006) aponta que propostas de investimento inicial de baixo custo são mais bem aceitas dentro das empresas. Além disso, métodos em que a implantação é feita de forma gradativa e que possua premissas adaptáveis ao cenário atual da empresa e seus processos, também caracterizam pontos fortes no critério de seleção de métodos para automação.

## 1.2 Objetivo

O objetivo deste trabalho é a elaboração e aplicação de um método para implantação do processo de formalização de casos de testes, utilizando conceitos de requisitos baseados em cenários, em empresas que nem sempre possuem ou fazem uso de processos formais de documentação. A estratégia proposta terá como base um modelo responsável pela conversão de cenários de requisitos em documentos padronizados de casos de testes e com uma abordagem de implantação que permita às empresas aderirem gradativamente ao método. O trabalho visa apresentar a simulação de coleta de requisitos de negócio e a geração de artefatos de testes em sistemas de diferentes níveis de complexidade, além de analisar a evolução do método em cada simulação, conforme o esforço investido pela empresa.

### 1.3 Contribuições da Pesquisa

O trabalho oferece uma alternativa para a implantação de processos automatizados de teste por meio de um método que possa ser adotado de forma parcial ou gradativamente. Adaptando-se ao comportamento atual da empresa e reduzindo impactos nos processos, além da resistência inicial por parte dos membros do projeto. Mesmo que a aplicação parcial do método não transpareça efetividade nos primeiros momentos devido ao crescimento contínuo da base de testes, o método traz benefícios imediatos quanto a padronização e organização dos testes.

O trabalho também apresenta os benefícios na utilização de levantamento de requisitos de cenários para a prática de documentação de casos de teste. Uma vez que, as técnicas de construção de cenário geram artefatos com conceitos similares ao domínio do negócio e podem ser convertidos em roteiros de execução de testes.

## 1.4 Etapas do Trabalho

O trabalho foi segmentado em 7 (sete) atividades descritas a seguir:

Atividade 1: Análise das informações providas pelo levantamento de requisitos por cenários

Esta atividade visa avaliar os processos e métodos utilizados no mercado para elicitación de requisitos através de cenários. A análise destes processos e métodos permite o mapeamento das informações e formatos resultantes, além dos dados necessários para viabilizar o uso destes processos e métodos.

Atividade 2: Levantamento de dados para viabilizar a automação dos documentos de teste

Envolve o levantamento inicial de dados para o projeto, como cenários em que empresas investiram esforço na implantação de automação de processos, documentos ou testes. Esta análise visa identificar o estado atual das técnicas de automação e os requisitos essenciais para suas aplicações. Além de validar, se os dados coletados na elicitación de requisitos por cenário atendem as características exigidas.

Atividade 3: Especificação do modelo para conversão de cenários em componentes de teste

Após a definição dos dados que serão utilizados no método, selecionados entre aqueles disponíveis no detalhamento de cenário de requisitos, esta atividade apresenta o modelo de conversão destes cenários em componentes de teste (formulários, listas, calendários...) genéricos. O objetivo do modelo é garantir que segmentos do sistema que possuam similaridades, sejam testados seguindo um roteiro de teste padrão.

Atividade 4: Especificação do modelo de automação dos casos de teste

Os componentes de teste resultantes da aplicação do modelo de conversão de cenários de requisitos são submetidos às particularidades do sistema (regras de negócio) nesta atividade. Estes dados são convertidos em casos de testes específicos do sistema para validação dos analistas de teste envolvidos no projeto.

Os casos de teste podem ser personalizados por meio de inclusão de situações para validação, tipo de variáveis, entre outros que serão abordados posteriormente. Cada uma destas validações enriquece a base de testes do modelo para a próxima iteração do método.

#### Atividade 5: Simulação de aplicação do método em cenários de baixa complexidade

Esta atividade foi proposta de modo a validar a aplicação do método em sistemas e a quantidade de interações necessárias para se gerar uma base de dados relevantes para atender os critérios de cobertura de teste para a situação de negócio proposta. Os cenários de aplicação são simplistas como calculadoras aritméticas ou módulos de controle de entrada e saída de produtos.

#### Atividade 6: Aplicações do método em ambientes de desenvolvimento de sistemas

A segunda fase de simulações ocorre em ambientes práticos com um maior número de variáveis a ser consideradas, como número de profissionais, critérios de completude próximos aos objetivos do projeto, restrições de tempo e esforço, estes são fatores que nem sempre são levados em consideração em simulações teóricas. O planejamento para as simulações do trabalho prevê a aplicação do método em dois projetos de desenvolvimento de sistemas, comparando a efetividade do método com a aplicação manual. O objetivo é coletar dados acerca de validações, interfaces com o usuário, regras de negócio complexas e fluxos de trabalho em sistemas.

#### Atividade 7: Análise dos resultados obtidos e da aplicação de técnicas convencionais

Avaliar implantação do método, incluindo a evolução da base de dados até um estado com valor significativo para as atividades de teste e regras de cobertura. Outro fator é avaliar a aceitação do método de levantamento de requisitos por cenários por parte dos especialistas do negócio. Estes dados serão utilizados para a construção dos mesmos casos de testes, fazendo uso de aplicações de métodos convencionais e manuais para efeito comparativo.



## 1.5 Organização do Trabalho

Na seção 1 foram apresentados os objetivos deste trabalho e a motivação para a realização do mesmo, além disso, esta seção também apresentou um resumo de todas as etapas planejadas e executadas para que estes objetivos pudessem ser atingidos.

A seção 2, Referências Bibliográficas, apresenta os conceitos de testes automatizados e de levantamentos de requisitos utilizando abordagens por cenários. Além disso, visa apresentar situações reais, nas quais estas técnicas foram aplicadas e as principais dificuldades em suas implantações.

Na sequência, a seção 3, Modelagem de Elicitação de Requisitos, apresenta a especificação do modelo proposto no trabalho para coleta dos requisitos de negócio, levantamento dos cenários de requisitos e conversão dos cenários em componentes de teste. Além disso, conta com informações acerca da Modelagem de Automação dos Casos de Teste, utiliza os resultados obtidos pelo modelo especificado na seção anterior para o processo de automação dos casos de testes. Esta seção denota a forma que os dados são inseridos no segundo modelo e processados, através da coleta de massas de testes e consultas a casos de teste históricos, encerrando o processo com os artefatos de teste.

A seção 4, Simulação do Método, visa apresentar todo o processo de simulação e aplicação da técnica em situações reais de desenvolvimento de sistemas, detalhando os dados coletados e os ciclos de evolução do método dentro do projeto.

A Avaliação dos Resultados é apresentada na seção 5 e contém toda a análise dos resultados obtidos ao longo da aplicação do método, priorizando a verificação da qualidade dos padrões gerados pela documentação de testes, comparativos entre tempos de implantação e execução, além dos benefícios e dificuldades que surgiram ao longo das simulações e aplicações.

O trabalho se encerra na sessão de número 6, Conclusão, apresentando as lições aprendidas ao longo do projeto, trabalhos futuros e comparações com as expectativas iniciais do projeto.

## 2 ESTADO DA ARTE

### 2.1 Abordagens para Elicitação de Requisitos

A etapa de elicitação de requisitos dentro do ciclo de desenvolvimento de sistemas possui como objetivo o entendimento do comportamento de processos, fluxos ou atividades. A correta representação do requisito em questão viabiliza o detalhamento de suas características, a visualização de seus relacionamentos com o processo como um todo e facilita a multiplicação do conhecimento envolvido, conforme conceito definido por Hsia, Samuel, Gao, Kung e Toyoshima (1994).

Os dados necessários para a construção dos requisitos de sistema geralmente estão em posse dos futuros usuários do sistema ou dos membros que dominam o modelo de negócio. No entanto, utilizando um dos argumentos apresentados no artigo de Carroll (1999), ele cita que nem todos estes indivíduos são especialistas nas áreas de desenvolvimento de sistema e como consequência, os profissionais responsáveis pela coleta destas informações precisam buscar abordagens mais objetivas e que façam parte da área de domínio dos usuários em questão. Por abordagens mais objetivas, entende-se o uso de representações ilustrativas e com interação direta por parte dos envolvidos, evitando o uso exclusivo de documentações, especificações técnicas ou subjetividade por parte do especialista.

Entre as técnicas que aproximam usuários e requisitos de sistema é válido ressaltar a utilização de representações como sequências de eventos, diagramas de fluxo de dados e UML (*Unified Modeling Language*), a construção de protótipos que apresentam uma imagem mais concreta do comportamento que o sistema visa atingir, além de textos redigidos que podem estar no formato de histórias, descritivos formais e informais. Estes por sua vez, evitam o uso de detalhamento técnico e focam em pontos de vista próximos ao domínio dos especialistas do negócio.

Projetos que tendem a ignorar a importância da elicitação de requisitos enfrentam dificuldades no entendimento de como o sistema deve se comportar, na tradução das informações que devem ser passadas aos desenvolvedores e analistas e no processo de comunicação com usuários e clientes. Exemplos destas situações podem ser encontrados nos estudos de caso apresentados por Graham e Fewster (2012). Os responsáveis pela coleta de informações ficam restritos à visão que está

sendo transmitida pelo especialista do domínio e possuem poucos recursos disponíveis para questionamentos acerca do negócio.

Uma das técnicas utilizadas para o processo de levantamento de requisitos é a representação por cenários. Hsia, Samuel, Gao, Kung e Toyoshima (1994) cita como formalização e objetivo da técnica, a apresentação do maior número possível de caminhos que um ator possui para executar um processo. Um cenário de requisito é a descrição de um evento ou situação agrupado com suas devidas características e comportamentos. Está contemplado nesta formalização a definição de todas as formas que um sistema possui para executar uma dada tarefa. Desta forma é possível utilizar os resultados para a construção de protótipos, premissas de testes e descrever comportamentos do ponto de vista do usuário.

A representação ilustrativa desta abordagem de requisitos é chamada de árvore de cenários, ela é apresentada com uma estrutura em nós, onde cada nó representa um estado, evento ou momento do sistema, e um estímulo interno ou externo indica qual será o caminho a ser seguido. Cada contribuição oferecida por usuários ou especialistas do modelo de negócio gera diferentes visões de usuário, ou seja, pontos de vista para o mesmo conjunto de cenários. Cada uma das visões tem por resultado uma nova árvore de cenários e estas são submetidas a uma análise conjunta para o melhor entendimento dos requisitos envolvidos. O conjunto de caminhos selecionados em uma interação dentro da árvore de cenários é chamado de episódio ou *scenario schema* e estes caminhos específicos são utilizados para a validação dos cenários levantados. Ao detalhar mais a utilização da técnica, deve-se partir do princípio de que quanto maior o número de usuários e especialistas do negócio, maior será a coleta de diferentes pontos de vista traduzidos em cenários. A coleta ocorre por meio de discussões, em que os responsáveis solicitam aos usuários ou especialistas acerca de suas tarefas, rotinas de trabalho e procedimentos realizados constantemente partindo de um objetivo que o analista de requisitos deseja alcançar. Essa especificação detalhada auxilia o processo de formalização, conforme citado por Malik, Lilius e Laibinis (2009). No decorrer do detalhamento é possível notar que o usuário descreve uma sequência de eventos para realizar o objetivo estipulado inicialmente e também os pontos em que usuários ou fatores externos aplicam diferentes estímulos ao processo e direcionam um dado conjunto de decisões que formam o caminho sendo descrito.

Conforme a formalização da técnica de elicitación citada anteriormente de Hsia, Samuel, Gao, Kung e Toyoshima (1994), os cenários são traduzidos a partir de estados em que o sistema ou processo aguardam algum estímulo externo para definir o comportamento que devem assumir. Estes estímulos são chamados de eventos e também são descritos na árvore de cenários. A árvore pode ser convertida em máquinas de estado conceituais para auxiliar a verificação de redundâncias, inconsistências e comportamentos superficiais.

A aplicação desta técnica tem como um de seus objetivos um processo contínuo de interação e apoio dos usuários envolvidos com o levantamento dos requisitos. Contudo, por se tratar de uma técnica com uma abordagem conceitual e de aplicação genérica, não há um método formalizado para sua utilização. Propostas de projetos e estudos como o de El-Far e Whittaker (2001), em que a elaboração de testes e sua execução seguem uma abordagem formal de cenários, ou no trabalho de Malik, Lilius e Laibinis (2009) utilizando o método Event-B apoiado por cenários fornecidos por usuários. Estes estudos propõem métodos formais para a elicitación de requisitos por cenários, apresentando regras para sua geração, verificação e validação. Demonstra-se que é possível fazer uso da técnica, mesmo que não haja um método universal formalizado. Conforme citado por El-Far e Whittaker (2001), algumas críticas direcionadas a técnica de cenários possuem como alvo, o baixo nível de detalhamento apresentado pelos resultados. Outro fator que auxilia tais críticas é a representação em árvore ou a aplicação dos episódios, não há formato único para documentar os cenários, formato de escrita ou características específica que devem existir, logo autores fazem comparações com as representações propostas pelos diagramas UML, como máquinas de estado e diagramas de sequência.

O conceito de cenários não é utilizado somente no processo de levantamento de requisitos. Aplicações podem ser encontradas nas áreas de projeto e testes, segundo exemplos apresentados por Carroll (1999), como a abordagem por cenários que servem como guia para os desenvolvedores na elaboração de projetos. Além disso, o uso da técnica se adapta bem com outros métodos da área de requisitos, incorporando visões e interações iniciais com os usuários, auxiliando o processo de validação, no qual os responsáveis possam ser participativos e integrados com a metodologia. O artefato representado pela árvore de cenários provê informações

suficientes para a construção de casos de teste, uma vez que representa o maior número possível de caminhos que o teste possa percorrer em busca de falhas.

Os dados presentes dentro de cada cenário podem ser traduzidos em componentes sistêmicos como listas, formulários, textos e entradas dos usuários. Desta forma, é possível expandir a área de atuação dos testes sem desvirtuar a visão do cenário provida pelo usuário. A repetição dos comportamentos levantados por cenários também pode ser utilizada em técnicas de automação, abstraindo padrões que devem estar presentes no sistema. Um exemplo destas visões é representado na Figura 1.

Figura 1 – Exemplos de Visão para o Cenário

O diagrama mostra um formulário de "Cadastro Pessoal" com os seguintes campos: "Nome:" (campo de texto), "RG:" (campo de texto), "Idade:" (campo de texto) e um botão "Salvar". À direita do formulário, há duas descrições de testes:

- Testes que validam o cenário de "Cadastro Pessoal"** (associado ao campo Nome)
- OU**
- Testes que validam os campos do formulário "Cadastro Pessoal"** (associado aos campos RG e Idade)

Fonte: Elaborado pelo autor

Na Figura 1 pode-se notar que o cadastro pessoal precisa de algumas validações, como por exemplo, o valor numérico na idade ou um validador de dado no campo RG. No entanto, este cadastro pessoal também está contido em um contexto maior que exige outros tipos de validações, como por exemplo o cadastro de uma pessoa para um envio de e-mail, ou para o preenchimento de um formulário que será impresso.

## 2.2 Processos e Formalizações

Uma das muitas definições para o termo processo indica a existência de uma série de ações ou passos para se atingir um dado objetivo conforme Reis e Nunes (2002). Entre as características que compõe estes processos estão artefatos, pessoas, recursos, estruturas organizacionais e restrições. Estas ações podem ser representadas de uma forma abstrata, partindo de um pensamento ou lógica de um indivíduo, ou de um conjunto de ações comuns praticadas por todos os indivíduos de um grupo. Na próxima sessão o trabalho apresenta estas representações abstratas

como modelos mentais que tem como característica um formato mais detalhado, além disso também serão discutidas as razões pelas quais ele é aplicado.

A formalização de um processo tem como objetivo documentar as sequências de ações, objetivos, artefatos que são utilizados, além de padronizar a execução, a comunicação e o entendimento dos membros envolvidos no processo, como citado por Koubarakis e Plexousakis (1993). No contexto deste trabalho almeja-se definir um método que permita a construção de casos de teste, partindo de cenários de requisitos que podem ser convertidos em componentes de teste e finalmente em artefatos de teste.

Um processo formalizado facilita o entendimento e a multiplicação do conhecimento dentro de um grupo ou empresa. O ato de formalizar algo não envolve somente os processos, mas também as representações por diagramas, além de padrões de escrita para alguns tipos de documentos, como especificações, casos de uso e roteiros de teste. Conforme Elgass, Krcmar e Oberweib (1995), processos tendem a ser elaborados de forma informal nos primeiros estágios de sua concepção ou quando estão sendo estruturados. Ao formalizar um dado processo viabiliza-se a simulação, maior detalhamento e avaliação dos mesmos.

Em eventos onde são aplicados processos informais, as características como ferramentas, representações e a própria execução são mais simples de utilizar, intuitivas e oferecem formas criativas para os usuários se expressarem, conforme abordado por Mukherjee, Dhoolia, Sinha, Rembert e Nanda (2010).

### 2.3 Testes e Técnicas de Automação

Ao abordar assuntos referentes às práticas de levantamentos de requisitos na seção anterior, nota-se a necessidade de validar os resultados obtidos e garantir a integridade das informações sendo utilizadas. Os conceitos de teste são aplicados para situações como estas com o objetivo de aumentar a confiabilidade entre os envolvidos e garantir que o processo ocorra como esperado.

Testes podem ser desenvolvidos de modo a validar diferentes etapas do processo de desenvolvimento de sistemas, seguindo a abordagem de Malik, Lilius e Laibinis (2009), podem ser utilizados para validar e verificar requisitos, documentações, funcionalidades, interfaces, códigos e projetos. Cada projeto possui características

individuais que requer esforço em análise para o planejamento dos testes e sua execução ao longo do desenvolvimento. Os testes devem ser selecionados de modo a contemplar o maior número de componentes sujeitos a falhas.

Como citado anteriormente no texto de Hsia, Samuel, Gao, Kung e Toyoshima (1994), assim como as práticas de elicitación de requisitos por cenários são utilizadas em alguns projetos de maneira informal, em diversos casos, os processos de teste passam pela mesma situação de informalidade. Embora haja documentações de testes, alguns projetos optam pela utilização de seus recursos diretamente na prática de execução dos testes, descartando total ou parcialmente o esforço envolvido na sua documentação. Como consequência deste tipo de prática, módulos similares acabam sendo testados de maneira diferente, testes são repetidos de maneira ambígua, e há pouco conhecimento por parte dos analistas acerca de áreas do sistema impactadas por mudanças, conforme os exemplos citados por Graham e Fewster (2012).

A formalização de um processo relaciona-se diretamente ao padrão escolhido para documentação e execução dos testes. Em ambas as situações há diversas ferramentas disponíveis no mercado para auxílio. Na fase de execução dos testes podem ser utilizadas técnicas de automação que reduzem esforços dos testadores para execuções repetitivas e para grandes massas de dados. Também há ferramentas que gravam e reproduzem execuções manuais auxiliando casos, em que ocorrem mudanças e correções no sistema, mostrando-se necessária uma nova execução dos casos de teste, como por exemplo, *ATRT (Automated Test and Retest)* e *QARun (Quality Assurance Run)* que pertencem às empresas *Innovative Defense Technologies* e *Compuware Technologies*, respectivamente. Nas fases de documentação de testes, o mercado disponibiliza ferramentas que padronizam a escrita e formato dos textos, replicam informações automaticamente que exigiriam esforço para preenchimento manual, além de criar códigos e relacionamentos entre casos de testes, relatos de falhas e outros artefatos do projeto.

Os profissionais envolvidos na área de testes seguem um modelo para a execução de suas tarefas. Para Hsia, Samuel, Gao, Kung e Toyoshima (1994) e El-Far e Whittaker (2001), este modelo pode existir mentalmente apenas, entretanto a lógica envolvida permite a execução dos testes. Para exemplificar este modelo mental,

pode-se apresentar a mesma interface a diferentes especialistas em testes e questionar por onde começariam a testar e quais sequências lógicas iriam seguir para buscar o maior número de falhas. Cada um dos especialistas, partindo do princípio que não possuem nenhum guia ou roteiros em mãos, vão utilizar uma linha de raciocínio que os satisfaça logicamente. Antes de qualquer análise, pode-se considerar que todos estejam certos, mas um não é capaz de reproduzir a linha de raciocínio do outro sem uma formalização ou justificativa das escolhas.

As aplicações de *Model-based Testing* seguem o conceito de formalizar um modelo e tornar a execução dos testes padronizada segundo o modelo definido, conforme Malik, Lilius e Laibinis (2009). Uma vez definido um modelo é possível reutilizá-lo e compartilhá-lo com o restante da equipe do projeto. Esta aplicação pode ser utilizada também para o processo de documentação de testes, desde que todos os profissionais da equipe em questão definam uma forma padronizada e unificada para a escrita e as informações dos documentos.

Entre os modelos apresentados por El-Far e Whittaker (2001), um modelo clássico utilizado para testes de software é o uso de máquinas de estado finitas, em que são definidos diversos estados do sistema e os comportamentos que realizam a transição entre estes cenários. Ao descrever o comportamento de todo o sistema, os testadores possuem um roteiro a ser seguido e informações acerca de requisitos dos testes, etapas de execução (conjunto de comportamentos) e os resultados esperados. Este modelo é utilizado para gerar testes com um formato padronizado de escrita e que contenham dados relevantes para execução, em comparação a uma construção manual dos testes. No entanto, El-Far e Whittaker (2001) adiciona que sistemas mais complexos exigem máquinas de estados com um número maior de informações e o esforço para sua construção nem sempre é considerado no projeto. Embora este esforço adicional seja um empecilho para sua utilização, os processos de automação podem ser aplicados em sistemas e projetos de diferentes níveis de complexidade.

A chave para a definição de um bom modelo para construção e execução de testes é conhecer o sistema em que o modelo será aplicado e quais tipos de testes o modelo explorará, segundo a abordagem apresentada nos estudos de Graham e Fewster (2012). Nestes estudos, os autores apresentaram exemplos da implantação



de métodos de testes mal sucedidos e as medidas necessárias para uma alteração nos cenários. Assim como a geração de cenários na etapa de elicitação de requisitos, devem-se impor limites para os testes resultantes da aplicação do modelo, pois quanto maior for o escopo, maiores serão os esforços para sua manutenção e execução. Tanto a organização quanto os profissionais envolvidos devem estar cientes dos objetivos que pretendem atingir, dos esforços periódicos na manutenção do modelo e da experiência necessária por parte dos responsáveis pelos testes para a sua correta execução.

Ao tratar acerca de testes baseados em modelos e padrões para reutilização de testes históricos, entram em cena as técnicas de automação citadas anteriormente. A utilização destas técnicas exige investimento por parte do projeto, tanto em recursos financeiros quanto profissionais. A implantação destas técnicas pode exigir novas adaptações a cada início de um novo projeto ou pode gerar resultados concretos somente na fase final do projeto ou em sua pós-implantação. El-Far e Whittaker (2001) cita as tarefas fundamentais para a escolha do modelo de testes mais apropriado de um projeto. Entre os pontos apresentados, um deles ressalta a necessidade de clarificar os objetivos de técnicas de automação para evitar falsas expectativas por parte dos interessados na elaboração do projeto. O uso destas técnicas representa benefícios no consumo de recursos, uma vez que há redução de trabalho repetitivo, além de auxiliar a comunicação entre os membros da equipe através de uma documentação padronizada e concisa. No entanto, a manutenção dos documentos e das ferramentas de automação tornam-se atividades periódicas que devem ser levadas em consideração durante o projeto.

Outro tópico relevante no planejamento e execução do processo de testes são as regras estabelecidas para a cobertura dos testes. Conforme Najumudheen, Mall e Samantha (2011), cada projeto deve estabelecer os objetivos que pretende atingir com os testes que serão executados. Deste modo, os padrões estipulados pela regra de cobertura auxiliam a avaliar os resultados dos testes e se os cenários atendem as expectativas do projeto.

Entre os assuntos abordados no parágrafo anterior destaca-se a preocupação com o planejamento de testes dentro do ciclo de desenvolvimento de sistemas. Os testes são elaborados em cima do objeto alvo que, por sua vez, estará sujeito à validação.

Uma vez que as falhas estejam catalogadas e a correção efetuada, ou até mesmo se surgir uma eventual mudança no código do sistema, as documentações de testes precisam indicar quais pontos foram afetados pela mudança, pois este cenário requer uma nova execução dos testes.

Em uma situação onde uma nova execução de testes precisa ser realizada a priorização de testes auxilia na escolha de quais situações precisam ser testadas novamente, pois não há necessidade do investimento de recursos em uma nova execução de todos os casos de teste do projeto. No entanto, em posse da informação citada anteriormente, é possível realizar apenas parte dos testes e ainda garantir a integridade dos resultados. Utilizando a abordagem por cenários no levantamento de requisitos e fazendo uso da árvore de cenários é possível identificar a interação dos estados e eventos representando o plano de testes de uma maneira ilustrativa. Conforme resultados apresentados por Sivashanmugam, Lin e Palanisamy (2011), os cenários e suas variações são testados dentro de seu contexto, mas a execução dos testes é feita de forma isolada. Em sua pesquisa o teste da funcionalidade ou do módulo do sistema é feito ao término de todos os testes parciais focados em cenários. Ao organizar os cenários de modo a contemplar toda a árvore de cenário, é possível verificar todas as variações da funcionalidade sob teste.

Neste capítulo, o trabalho apresentou as principais características atreladas a testes e técnicas de automação, além dos impactos positivos e negativos causados por sua adoção. Os principais fatores abordados neste trabalho ressaltam as dificuldades na implantação de técnicas automatizadas, principalmente por questões de custos, falta de entendimento das necessidades e elaboração superficial do escopo. Embora, existam vantagens comprovadas na implantação de técnicas de automação, muitos exemplos falhos estão presentes nas empresas que acabam por amedrontar empresas que não possuem especialistas para melhor informá-las. No próximo capítulo, o trabalho apresentará alguns exemplos de implantações de técnicas para formalização de testes.

## 2.4 Trabalhos Relacionados

Esta seção visa apresentar abordagens similares àquelas que foram empregadas no método proposto por esta pesquisa. Para tanto, a seção é dividida em duas subseções que englobam trabalhos relacionados a cenários de requisitos e testes.

### 2.4.1 Aplicação de Modelos e Casos de Teste

A evolução de sistemas e a construção de novas melhorias no decorrer de seu ciclo de vida são características comuns nas empresas. No entanto, garantir que a integridade do sistema não tenha sido impactada por tais mudanças é uma preocupação que os profissionais da área de qualidade devem levar em consideração. O processo de automação de testes auxilia na rapidez com que essa revisão é efetuada e possui como objetivo reduzir o trabalho manual por parte dos responsáveis por testes.

Neste contexto, Graham e Fewster (2012) que apresenta um exemplo em que mudanças no sistema ocorrem periodicamente e a execução da rotina de testes é uma atividade que consome tempo considerável do projeto. A proposta do estudo de caso foi implantar práticas de testes baseados em modelos dentro do projeto e verificar juntamente com a automação dos testes quais seriam os benefícios obtidos pela empresa no decorrer do tempo. Para tanto, construiu-se um fluxo de eventos que simulasse a operação do sistema e as tarefas que são executadas, desta forma deu-se origem a um conceito similar ao levantamento de cenários e a posterior construção da árvore de cenários. Na sequência, criaram níveis de abstração para estas tarefas para simplificar o processo de manutenção e identificação dos testes em caso de mudanças. Estes níveis de abstração podem ser traduzidos sob diferentes pontos de vistas, motivando a aplicação de diferentes testes.

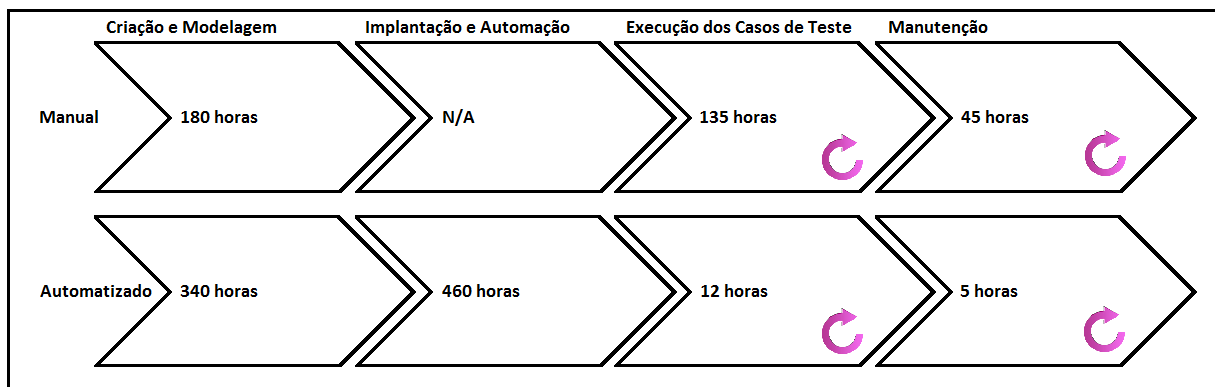
O trabalho partiu do princípio que já existia uma base com cenários testáveis, e estes testes foram aplicados ao fluxo de eventos ainda na fase de documentação. Ao aplicar os dados à ferramenta, esta gera casos de testes a partir da combinação. Este artefato é composto dos cenários a serem validados, os resultados esperados pelo usuário e passos para a execução do teste em si. Neste estágio nota-se a preocupação com a padronização do formato dos testes, de modo a criar artefatos

que sejam similares entre si e facilitem a comunicação dentro da equipe, conforme Graham e Fewster (2012).

Os resultados relatados por Graham e Fewster (2012) foram catalogados e utilizados como instrumento de comparação em relação aos resultados que o projeto obteve antes do uso do modelo proposto. Os casos de testes eram documentados manualmente pelos testadores, assim como sua manutenção e execução dos testes a cada nova mudança. Entre os benefícios obtidos, o projeto conseguiu identificar um maior controle do escopo do projeto, graças aos fluxos de tarefas. A taxa de descoberta de falhas no sistema aumentou em cerca 30% e as falhas foram descobertas em etapas iniciais do projeto com o envolvimento dos responsáveis por testes. Em relação ao tempo e esforço investidos na implantação desta estratégia declara cerca de 460 horas do projeto, entre as atividades estavam implantação da ferramenta, levantamento de dados para o modelo e treinamento da equipe. Além disso, para o tempo de construção dos testes investiram-se 340 horas, onde 160 horas representam os cenários e 180 horas representam os casos de teste. Na execução manual, cada teste leva cerca de 45 minutos para ser completado, tendo como resultado um total de 135 horas. Em contrapartida a execução automatizada levou cerca de 12 horas para todo o processo de testes, conforme Graham e Fewster (2012). Estes dados reforçam um argumento apresentado anteriormente nesta dissertação, onde o processo de automação exige maior esforço para a concepção do artefato, porém o tempo de execução é reduzido de forma relevante.

Outra informação relevante foi o tempo consumido para uma eventual manutenção nos documentos de teste, em que o processo normal consome 45 horas e o processo com a implantação do modelo foi concluído em 5 horas. Na comparação final, o novo processo automatizado consumiu, aproximadamente, 817 horas do projeto e sua realização de forma manual utilizou 360 horas. Enquanto, a implantação e automação ocorrem na primeira utilização da técnica, em quatro interações do processo automatizado, a técnica passa a consumir um número menor de horas que o processo manual. O resumo do esforço investido pode ser avaliado na figura 2.

Figura 2 – Comparação de Esforço em Horas, adaptado de Graham e Fewster (2012)



Fonte: Elaborado pelo autor

A análise do estudo de caso reforçou a participação de especialistas em testes na construção do modelo e na coleta das informações. A infraestrutura do sistema deve estar apta a receber testes automatizados e isso deve ser considerado em estágios iniciais do desenvolvimento do sistema. Este primeiro estudo ressalta o planejamento dos testes e indica que o uso de automação não é adequado para todos os projetos ou processos, uma vez que o fator infraestrutura seja relevante no momento da definição do uso de um método de automação. Além disso, outro ponto que deve ser considerado é o investimento inicial para a implantação destas ferramentas, levando em consideração a economia posterior de tempo.

Outra avaliação crucial para o processo de testes é a seleção do método que será empregado em um dado projeto. O trabalho de Sivashanmugam, Lin e Palanisamy (2011) visa apresentar alternativas para execuções de teste. Em sua pesquisa discute-se a abordagem por cenários empregada no processo de testes em uma ferramenta Microsoft, avaliando algumas métricas de qualidade previamente estipuladas. Além disso, busca-se identificar os benefícios e implicações da adoção deste tipo de técnica.

Para tanto, o trabalho propôs essencialmente uma mudança no comportamento da equipe de testes e os envolvidos no processo. O motivo desta mudança justifica-se através do conceito de cenários, onde usuários específicos identificam seus fluxos de trabalho ou sequências de tarefas para construir um cenário de requisito. No entanto, este cenário não envolve uma única funcionalidade e não pode ser tratada de forma isolada, conforme retratado por Sivashanmugam, Lin e Palanisamy (2011).

Os cenários de teste são construídos visando às tarefas do usuário dentro de um determinado contexto, desconsiderando as funcionalidades envolvidas do sistema. A representação real de características destes cenários auxilia na construção dos casos de teste e neste momento são consideradas as funcionalidades. De acordo com Sivashanmugam, Lin e Palanisamy (2011), a soma destes dois fatores têm como resultado o planejamento para execução dos testes baseados em um cenário.

O trabalho de Sivashanmugam, Lin e Palanisamy (2011) aponta o cuidado que deve haver no balanceamento das tarefas e na cultura da equipe para um bom andamento do projeto em execução. Os autores ressaltam estes dois fatores, já que o andamento natural de um projeto exige uma preocupação com os entregáveis. No entanto, para que o método funcione corretamente é necessária uma atenção contínua aos cenários de testes e a forma como eles se desenvolvem. Em relação à cultura da equipe, a falta de conhecimento e experiência neste tipo de abordagem pode gerar resistência. Esta mesma resistência também é tratada quando se aborda a implantação de ferramentas de automação sem conhecer claramente seus resultados ou a mudança no formato de coleta de requisitos, mesmo considerando os benefícios para outras fases do desenvolvimento de sistemas.

O conjunto de dados que identifica a cobertura de testes é outro assunto relevante, pois o método definido neste trabalho busca a construção automatizada de casos de testes e estes precisam respeitar as métricas de cobertura do projeto em questão. Para efeito de análise e comparação, o trabalho de Najumudheen, Mall e Samanta (2011) aborda uma técnica para análise de cobertura de testes partindo de um modelo elaborado na pesquisa chamado *Call-based Object-Oriented System Dependence Graph* (COSDG). A técnica pode ser aplicada em programas com desenvolvimento orientado a objetos. A intenção da pesquisa é avaliar a eficiência dos resultados obtidos pela ferramenta e gerados posteriormente pela análise dos dados.

Conforme Najumudheen, Mall e Samanta (2011), o procedimento para execução do trabalho parte da construção do modelo COSDG a partir de um diagrama de classes que represente o sistema e as relações de dependência entre estas classes. Para cada conjunto de classes representadas no diagrama, gera-se um modelo composto por vértices. Cada um destes vértices representa uma classe, que por sua vez

desmembra-se em submodelos. Cada vértice destes submodelos são os métodos da classe a qual pertence. O segundo passo envolve a ligação do modelo ao próprio código que através de classes e métodos identificará os testes viáveis. O último passo da execução trata-se da análise de cobertura, onde cada nova execução dos testes inicializa um conjunto diferente de testes e entradas para os mesmos. Desta forma, o modelo vai sendo varrido a partir de chamados de cada método de suas classes. Por fim, categorizam-se os vértices do modelo conforme sua execução, adotando, por exemplo, o estado “visitado” ou “não visitado”.

Entre os resultados obtidos, a pesquisa foi capaz de avaliar fluxos de trabalhos, fluxos de dados, relação de dependência entre objetos, entre outros. Najumudheen, Mall e Samanta (2011) tinham como fatores motivacionais em seu trabalho o auxílio na utilização de métricas para cobertura de testes, além disso, também havia a pretensão de facilitar o mapeamento de funcionalidades de sistema que estão representadas ou fragmentadas nos testes. Para efeitos comparativos, a execução do processo para se obter um valor quantitativo fazendo uso do COSDG consumiu menos esforço por parte dos especialistas.

Esta questão de cobertura tem um valor significativo para a construção de casos de testes automatizados propostos neste trabalho, pois deve-se definir através do primeiro modelo quais componentes testáveis serão contemplados em cada projeto. Uma vez que possam ocorrer particularidades em cada projeto, deve-se considerar possibilidades de classificação ou personalização dos componentes.

#### 2.4.2 Vantagens na Aplicação de Cenários de Requisitos

Ao selecionar o formato de elicitação de requisitos para um projeto, é comum buscar o método cuja experiência da equipe seja mais abrangente. No entanto, a combinação de mais de um método para o levantamento de requisitos pode aumentar a qualidade dos resultados alcançados. Conforme apresentado por Maiden e Robertson (2005), sua intenção era averiguar os benefícios da combinação de técnicas de casos de uso e a utilização de cenários de requisito.

Para validar tal hipótese, o trabalho de Maiden e Robertson (2005) propôs três perguntas que deveriam ser respondidas ao final da pesquisa. A primeira questão

envolve a dúvida sobre a origem dos conceitos empregados em casos de uso e por consequência os dados formalizados. A segunda questão é relacionada às formas de representação utilizadas no levantamento de requisitos com os *stakeholders* ou usuários, sendo que o trabalho previa a identificação da forma de representação mais eficiente. A última pergunta proposta questiona o custo benefício da utilização de cenários de requisitos em conjunto com casos de uso em projetos reais, restritos por tempo e recursos.

A execução do trabalho ocorreu através da utilização de uma ferramenta chamada *Requirements Engineering with Scenarios for User-Centred Engineering* (RESCUE) responsável por formalizar requisitos através de uma abordagem por requisitos. O estudo foi realizado em um projeto, cuja área de domínio envolvia controle de tráfego aéreo. Em uma de suas proposições, Maiden e Robertson (2005) apresentaram como problema, a falta de detalhamento para o uso de protótipos e histórias, recomendados pelo *Rational's Unified Process* (RUP) e utilizaram esses dados como base da ferramenta.

A ferramenta apresentada operava através de níveis de abstração dos requisitos, inicialmente coletava informações acerca da restrição do escopo do domínio em que estava sendo aplicada. O segundo passo continha um caso de uso inicial que era submetido a uma série de *workshops*, contando com o auxílio de *stakeholders* e especialistas da área de domínio. Estes *workshops* não possuem um formato definido e Maiden e Robertson (2005) propõe que estes sejam realizados através de métodos de alta interatividade com os usuários, contendo representações ilustrativas, representações do processo e passagem das telas, além de envolver os responsáveis no processo, mesmo que estes não sejam especialistas em sistemas. No último passo submetem-se os casos de uso criados até o momento a um método de tradução em cenários. Desta forma é possível visualizar as variações de caminhos dentro dos cenários com menos esforço por parte dos *stakeholders*.

O resultado, conforme Maiden e Robertson (2005) gerou detalhes acerca o comportamento de cada caso de uso, estas informações são novamente aplicadas ao RESCUE. O quarto e último passo cria uma história completa para o detalhamento do caso de uso, fazendo uso do artefato da árvore de cenários. Os resultados são validados e são mapeados modelos com atividades humanas do



processo para identificar se há alguma falha no escopo, características ou no comportamento do cenário em questão.

O projeto da aplicação se resumia a um novo sistema de controle aéreo que seria utilizado em torres de controle, teve duração de um ano e os *stakeholders* envolvidos receberam treinamento para a utilização da ferramenta RESCUE, conforme Maiden e Robertson (2005). Para a documentação das atividades humanas foram construídos quinze cenários, cujo detalhamento estava disperso entre os casos de uso existentes. O número final apresentado pela pesquisa atingiu uma grandeza que superava 900 proposições entre os requisitos detalhados.

Entre os resultados obtidos Maiden e Robertson (2005) voltam a serem apresentadas as sugestões que beneficiam o uso de métodos complementares de elicitação de requisitos como cenários e casos de uso, ou até *user stories* para aplicação de métodos ágeis. Da mesma forma, o método de automação de casos de teste segue este conceito, pois os requisitos estão intimamente ligados aos testes. Os autores destacam a importância do envolvimento de pessoas nos projeto que não são especialistas, seja em sistemas ou domínio, pois estas pessoas podem trazer dúvidas relevantes para o processo de elicitação de requisitos.

Em relação às perguntas propostas, ao realizar uma analogia com o método de automação de casos de testes, é possível extrair informações essenciais que serão utilizadas no método sendo proposto. Nas primeiras atividades apresentadas na seção do método de trabalho (seção 1.4), a pesquisa visa identificar quais informações são relevantes na construção de um cenário de requisito. A origem desta informação equivale aos resultados obtidos pela primeira pergunta, ou seja, os dados que compõe os conceitos de um caso de uso. Entre as vantagens e desvantagens apresentadas nas representações da segunda questão, pode-se identificar que o uso de histórias completas, em uma linguagem mais próxima ao domínio dos *stakeholders* e usuários, torna-se mais vantajoso e com melhores resultados. No entanto, como apresentado anteriormente nos conceitos (seção 1.4) deste trabalho, esta aplicação exige mais esforço por parte do projeto e como consequência, mais tempo. Conforme apresentado por Maiden e Robertson (2005), a última pergunta apresenta em porcentagens, os resultados obtidos na pesquisa. Apesar do investimento de um processo mais longo, a aplicação de cenários e

histórias complementando a formalização de casos de uso foi responsável pelo levantamento de 40% dos requisitos documentados, mostrando que para o estudo proposto, sua aplicação foi diferencial para a qualidade dos resultados.

No método de automação proposto por este trabalho, o objetivo da documentação de requisitos está intimamente relacionado com a elaboração dos casos de teste. Mesmo que não haja necessidade da aplicação de um método formal de casos de uso, a utilização do mesmo complementa o formato selecionado para elicitación de requisitos por este método, conforme apresentado por Maiden e Robertson (2005). Desta forma, o número de informações que serão convertidas em componentes de teste tende a aumentar.

Outro fator relevante em relação a cenários de requisitos envolve o fato de que a metodologia é utilizada muitas vezes através de um processo informal. Deste modo, empresas e especialistas criam certa resistência por não identificarem resultados concretos para o formalismo desta metodologia. No entanto, entre os estudos de cenários é possível identificar aplicações que provam o contrário, como o trabalho de Ruokonen, Pajunen e Systa (2009) que apresenta abordagem para modelagem de processos baseada em cenários.

O objetivo do trabalho de Ruokonen, Pajunen e Systa (2009) é propor uma modelagem de processos com notações objetivas e apresentações em diferentes níveis de abstração. Para tanto, mostrou-se necessário à identificação dos requisitos de negócio e sua representação em cenários, conforme processo tradicional da metodologia, sem exigir um total detalhamento do comportamento do processo. Estes cenários são submetidos a uma representação similar ao diagrama de sequência presente na UML e para cada representação de processo, seleciona-se um episódio entre os cenários, finalizando com a conversão para uma representação em máquina de estados, gerando exemplos, conforme Ruokonen, Pajunen e Systa (2009). Estes artefatos podem ser convertidos em diagramas de atividades, que por sua vez representam o esqueleto do processo para melhor refinamento do mesmo.

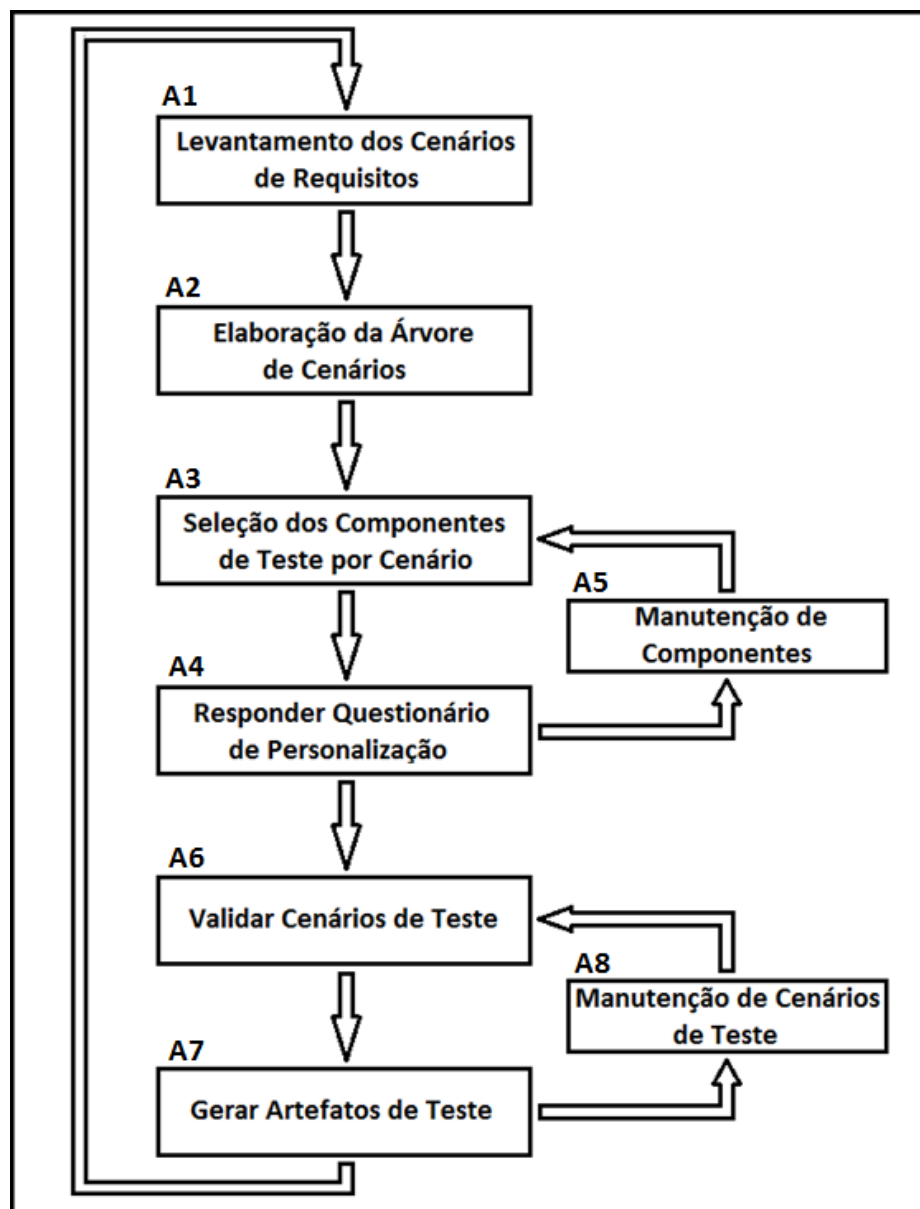
O trabalho de Ruokonen, Pajunen e Systa (2009) foi construído essencialmente em projetos teóricos, porém a conversão para representações formais pertencentes a UML apoiam a ideia de que seja possível implantar procedimentos formais com o uso de cenários. No método de automação de casos de testes sendo proposto não

há uma formalização em relação ao padrão de coleta de cenários de requisitos. No entanto, as dependências documentadas nas árvores de cenário auxiliam o processo de execução dos casos de testes, auxiliando em priorização e seleção dos casos de testes para representação de um dado episódio de cenário.

### 3 MÉTODO PARA IMPLANTAÇÃO GRADATIVA DO PROCESSO DE FORMALIZAÇÃO DE CASOS DE TESTE

A partir desta seção são apresentadas as definições para a elaboração do método proposto no objetivo. Nesta seção introdutória será apresentado no Diagrama 1, o método em uma visão macro. O item 3.1 contempla os itens A1 e A2 do diagrama, representado a parte de coleta de dados. O item 3.2 representa A3, A4 e A5, onde o método converte cenários de requisitos em componentes de testes. Finalmente, o A6, A7 e A8 representam o item 3.3 tratando sobre personalização de variáveis e casos de teste.

Diagrama 1 – Método de Automação de Casos de Teste



Fonte: Elaborado pelo autor

### 3.1. Extração de Dados dos Cenários

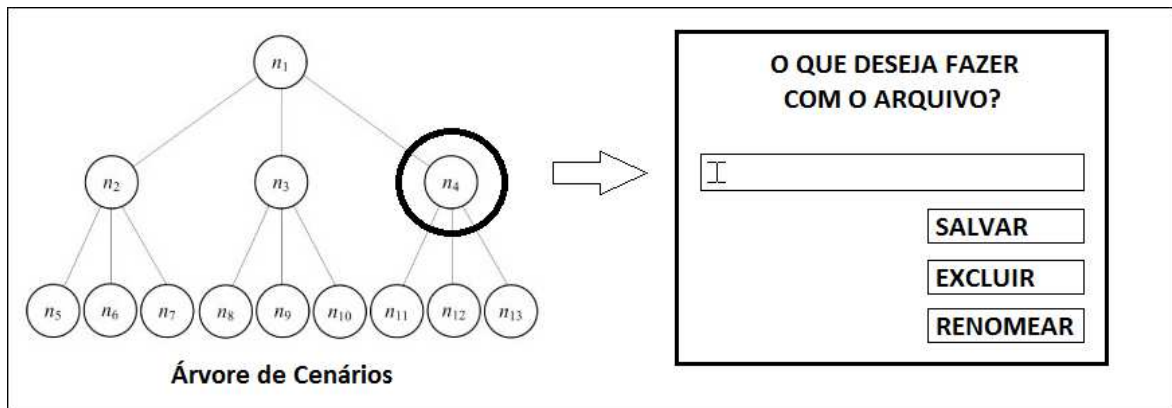
Um cenário de requisito, conforme a perspectiva do usuário envolve toda a descrição de uma tarefa com um objetivo definido. Para a realização desta tarefa deve-se seguir uma determinada sequência de passos que podem conter estímulos ou decisões por parte do usuário e do sistema.

No entanto, a construção de casos de teste exige informação com um maior nível de detalhamento. Na primeira fase, arquitetos ou analistas de teste precisam selecionar quais componentes (formulários, listas, calendários, botões, páginas, entre outros) serão utilizados para sistematizar um dado cenário de requisitos. Embora, os responsáveis pelo negócio (*stakeholders* ou analistas de negócio) tenham dificuldade em visualizar o sistema neste nível de abstração, os analistas de teste podem auxiliá-los com exemplos, protótipos, entre outras técnicas interativas para coleta de dados.

Antes de detalhar o método, é válido explicitar o significado de cada termo utilizado no trabalho para detalhamento das partes unitárias do método de automação de Casos de Teste. A árvore de cenários é a representação de todos os cenários e como ocorrem suas ligações. Cada nó da árvore é um cenário detalhado por um analista de negócio que pode ser traduzido em componentes de teste. Além disso, ao escolher um caminho da árvore, ou seja, um episódio, os cenários e condições para que o caminho seja percorrido é denominado de fluxo. O termo fluxo é utilizado porque ele determina o fluxo de testes no artefato final do método.

Na situação hipotética demonstrada na Figura 3, um analista de negócio ou requisitos pode demonstrar como pretende executar a tarefa e apresentar a interação deste cenário de requisitos com seus predecessores e sucessores. Na Figura 3, o nó N4 pode representar um momento de decisão em que o usuário optou por excluir o arquivo. Logo, temos a ação do usuário somada aos detalhes da interface, estes dois fatores vão construir as situações que precisam ser testadas e validadas. Os componentes utilizados na demonstração do cenário possuem um conjunto de características comuns que podem ser testadas e validadas posteriormente. Estes componentes podem ser agrupados em “categorias”, cada qual com suas particularidades de teste.

Figura 3 – Detalhes de um Estado do Cenário



Fonte: Elaborado pelo autor

Os conjuntos apresentados devem ser dinâmicos e podem ser personalizados conforme a necessidade dos analistas de teste, criando níveis de classificação, novas categorias e novos cenários para a execução dos testes. Estas personalizações podem ocorrer constantemente, criando novas validações que passam a integrar todos os casos de teste, ou validações particulares de um projeto ou sistema (exceções à regra). Toda a personalização e construção dos componentes referem-se à Atividade 3 das Etapas do Trabalho. A Figura 4 apresenta um exemplo de restrição do cenário, considerada como personalização. Inicialmente, havia duas restrições que precisariam ser validadas, no entanto durante a fase de testes percebeu-se que há a necessidade de incluir uma nova situação a ser testada.

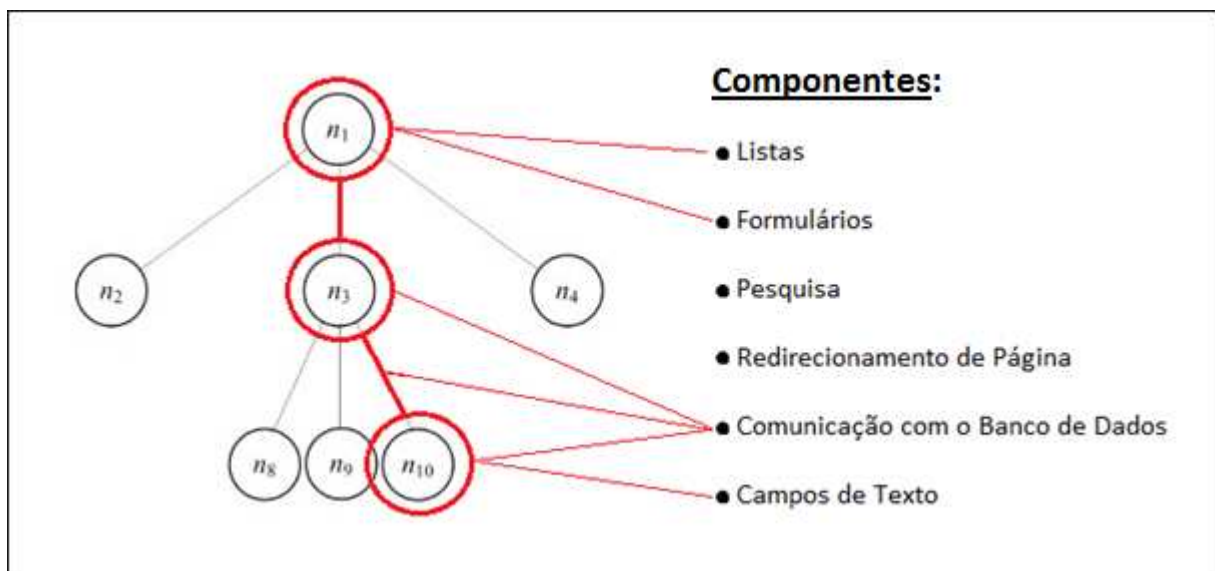
Figura 4 – Personalizando um Cenário

A interface mostra um formulário de validação. À esquerda, há dois campos de entrada rotulados 'DATA INÍCIO:' e 'DATA FIM:', e um botão 'VALIDAR' abaixo deles. À direita, o texto 'Restrições:' precede duas listas de itens: '- Preencher "Data Início" maior que "Data Fim"' e '- Preencher Data Inválida'. Abaixo do formulário principal, uma seta aponta para cima para uma caixa de texto que contém o texto: 'Nova Restrição: Não permitir a entrada de datas anteriores a 01/01/1900'.

Fonte: Elaborado pelo autor

Uma vez realizado o entendimento do processo de negócio e definido o escopo do sistema, os analistas de negócio desmembram a sequência de ações em cenários. Dentro dos cenários, podem-se identificar as condições de entrada e saída de um determinado estado do processo, além das variações no comportamento do cenário para seguir as condições expostas por uma estória ou episódio (representando um fluxo padrão do processo ou sistema). A importância deste fluxo só será identificada ao término da aplicação do método, uma vez que os casos de testes apresentam-se documentados e o fluxo do sistema serve como guia para a ordem de execução destes testes. Esta tradução se enquadra para testes que visam validar regras de negócio, conforme os exemplos apresentados por Graham e Fewster (2012) que citam estas informações como premissas para a construção de casos de teste. Conforme a Figura 4, a aplicação também é válida para testes de interface e de componentes do sistema. Nesta ilustração, a tarefa a ser validada traçou uma sequência de eventos que segue de N1 para N3 e depois para N10. Embora as regras de negócio estejam sendo validadas conforme esse fluxo tem sequencia, pode-se vincular os componentes ou categorias descritas na figura para cada cenário ou transição de cenário dentro da árvore.

Figura 5 – Episódio do Cenário representado por Componentes



Fonte: Elaborado pelo autor

Um sistema que tem como objetivo a entrada de artigos em um estoque pode ser validado ao verificar se o usuário consegue atingir seu resultado esperado. Entretanto, os analistas de teste podem validar se o formulário é adequado para a

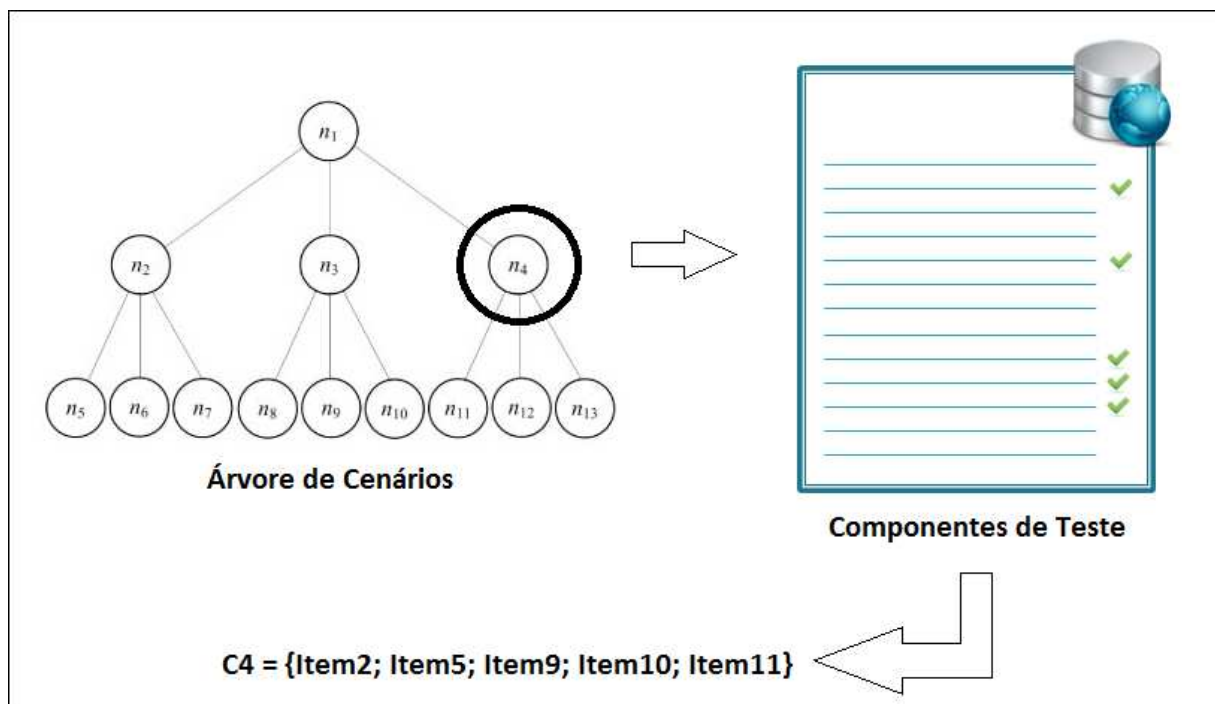
entrada dos dados, se todas as entradas possuem validações e restrições corretas, se os dados são salvos corretamente na base de dados, ou até mesmo verificar a apresentação visual da interface. Portanto, é necessário que os cenários possam ser interpretados em diversas perspectivas.

### 3.2. Modelo de Conversão de Componentes de Teste

As informações contidas na árvore de cenários viabilizam a construção do roteiro de testes, enquanto que os componentes selecionados para cada cenário de requisito (Figura 5), representam os dados de cada teste (pré-condições, passos de execução, validações, pós-condições, resultados esperados). Estes dados formam a base citada na Atividade 4 das Etapas do Trabalho.

O método consulta a base de dados que contém os cenários de testes categorizados nos diversos tipos de componentes. Os analistas de negócio podem selecionar os cenários que não se enquadram no requisito que estão especificando, de modo a evitar um excesso de testes que não serão executados, ou que não se aplicam as características do negócio.

Figura 6 – Conjunto de Componentes por Cenário de Requisito



Fonte: Elaborado pelo autor



Os conjuntos de cenários são armazenados na base de dados para posterior geração dos casos de teste. Cada variação no roteiro de testes indica uma sequência de ações selecionada na árvore de cenário e por consequência um novo conjunto de testes que pode ser personalizado. A próxima sessão, o trabalho apresenta a conversão dos componentes de teste em documentos formais.

### 3.3. Automação de Casos de Teste - Personalização dos Cenários

O trabalho apresentou no primeiro modelo representado no item 3.2, a conversão de cenários de requisitos em componentes de teste. O segundo modelo visa a inclusão de características exclusivas de cada teste e construção do documento formal. Ambas as representações seguem o conteúdo proposto no método de trabalho do item 1.4.

Entre os cenários de testes presentes na base de dados e as categorias definidas para cada um deles, estão presentes as informações que tornam cada teste particular, conforme a regra de negócio a que é submetido. Partindo do momento em que os cenários foram traduzidos em componentes de testes, o método solicita ao usuário (*stakeholders* ou analistas de negócio) a entrada de alguns dados, como massa de testes ou valores válidos para a situação que se pretende validar.

A situação descrita na Tabela 1 descreve uma personalização comum de testes para o componente “Formulários”. O formato e padrão que o teste será escrito já estão definidos, o analista de teste deve auxiliar no preenchimento das particularidades do teste. O método solicita a entrada de dados que podem variar conforme a o cenário em teste.

Tabela 1 – Formulário de Personalização de Testes

Questionário (Formulários)	Respostas		
Existem Campos Obrigatórios?	Sim	Não	Quais?
Existem Campos Numéricos?	Sim	Não	Quais?
Existem Campos de Data?	Sim	Não	Quais?
Existem Validações entre 2 Campos de Data Distintos?	Sim	Não	Quais?
Existem Restrições quanto a Quantidade de Caracteres dos Campos?	Sim	Não	Quais?
Existem Campos Alfanuméricos?	Sim	Não	Quais?
Existem Campos que Possuem Dependência entre si?	Sim	Não	Quais?
Existem Campos do Tipo Seleção Única?	Sim	Não	Quais?
Existem Campos do Tipo Seleção Múltipla?	Sim	Não	Quais?

Cada um dos testes possui um conjunto de variáveis que podem ser personalizadas em termos de quantidade e tipos, conforme apresentado na Tabela 2. A cada nova interação com o método de automação, os usuários recebem sugestões de preenchimento, baseadas em seus últimos casos de testes gerados. Além disso, os analistas de teste podem verificar quais cenários de teste não se aplicam a esta situação particular, sem prejudicar sua base padrão de testes.

Tabela 2 – Personalização de Variáveis

Componente	Categoria	Cenário de Testes
Formulário	Máximo Caracteres	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida

Variável 1:	Campo - <u>Valor de Transferência</u>
Variável 2:	Nº de Caracteres - <u>20</u>

Componente	Categoria	Cenário de Testes
Formulário	Máximo Caracteres	Checar Validação do Campo "Valor de Transferência" ao Preencher um Valor "x" que supere a 20 Caracteres

### 3.4. Modelo de Geração dos Documentos

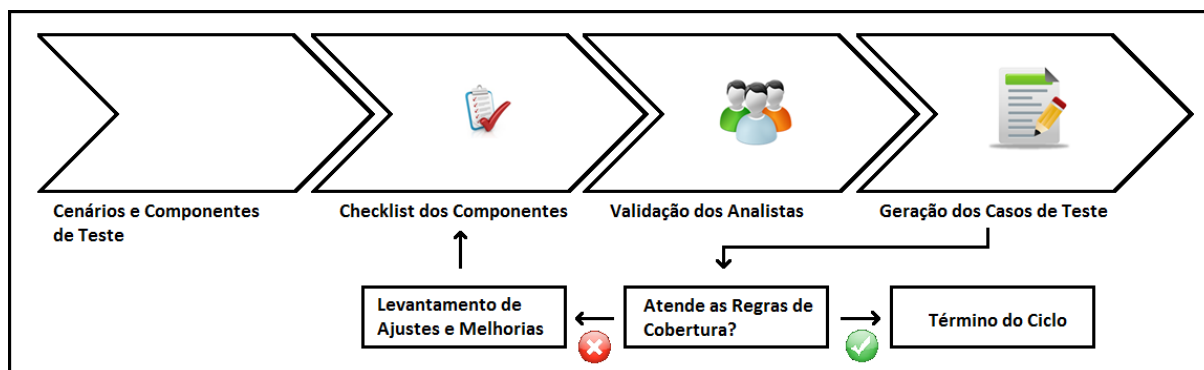
A partir da personalização dos testes efetuada anteriormente, o método recebe a árvore de cenários para composição do roteiro e as particularidades de cada cenário. O padrão de escrita de cada teste é coletado e convertido para o texto do caso de testes, enquanto que os dados preenchidos e utilizados no teste são armazenados como histórico.

Os casos de teste podem ser gerados em diversos formatos, conforme configuração de saída da base de dados. Para o estudo proposto neste trabalho, os documentos devem ser gerados em *Excel*, de modo a comparar com casos de teste que tenham sido escritos manualmente por especialistas de teste.

O documento é submetido a validação dos analistas de teste e de negócios para identificar o nível de cobertura dos testes. Na situação em que é necessária a execução de alguns ajustes nos testes ou melhorias nos dados, os usuários podem retornar a fase de dos questionários, onde ocorre a entrada das variáveis, ou podem retornar aos testes correspondentes a cada componente da árvore de cenários de

requisitos. A cada iteração, a base de dados armazena as informações e torna-se mais robusta para a próxima criação de documentos, seja na personalização de categorias dos cenários, nos roteiros de execução ou em dados para massa de testes. Conforme representado na Figura 7, o modelo pode ser aplicado o número de vezes que for necessário até que se atinja o nível de cobertura de testes esperado pelos analistas de teste e de negócios.

Figura 7 – Modelo de Geração de Artefatos de Teste

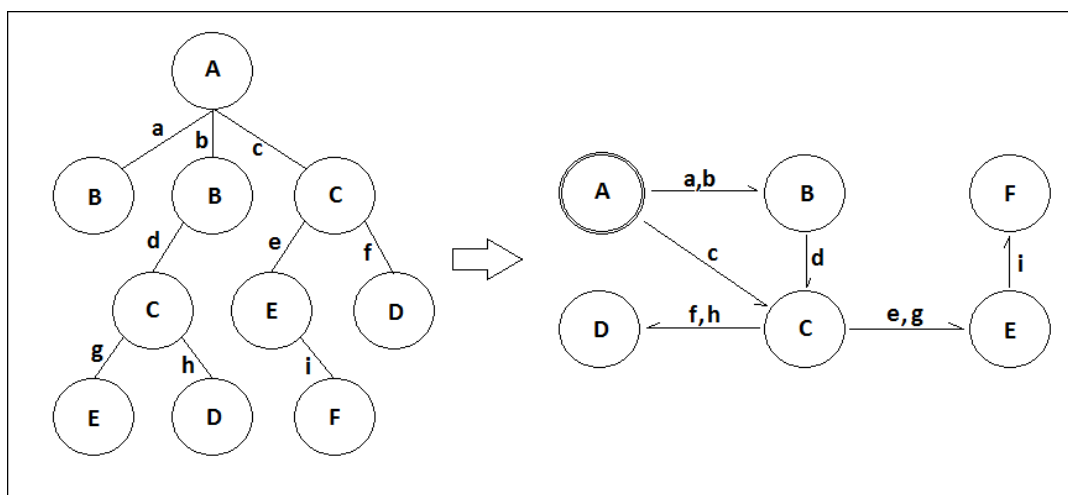


Fonte: Elaborado pelo autor

Uma vez atingido o padrão de aceitação para os testes do sistema, o método coleta a árvore de cenários de requisitos, organiza os casos de teste em cada episódio ou estória e gera o roteiro de testes. Este é apresentado como um diagrama de máquina de estado, seguindo o padrão da UML, conforme apresentado na Figura 6. O objetivo desta representação é disponibilizar uma visão completa do sistema durante a definição do escopo de testes, além de auxiliar na priorização dos testes.

A adaptação da árvore de cenários no diagrama de máquina de estados é bastante objetiva, a partir do estado atual, mapeiam-se quais são os caminhos possíveis que um dado estado pode ser suscetível. De acordo a representação na Figura 8, os estados que aparecem mais de uma vez na árvore de cenários são consolidados de modo a representar o caminho percorrido no diagrama. Para a escolha de um determinado grupo de casos de testes, o responsável pelos testes apenas seleciona os estados inicial e final que deseja validar e o caminho que deseja percorrer. Todos os casos de teste que são classificados nestes estados ou caminhos percorridos fazem parte do escopo. Na ilustração, os cenários estão representados por letras maiúsculas, enquanto que as condições a serem satisfeitas para mudança de cenário estão representadas por letras minúsculas.

Figura 8 – Adaptação da Árvore de Cenários



Fonte: Elaborado pelo autor

Para este trabalho, os padrões de aceitação são relacionados diretamente a algumas regras de cobertura parametrizadas no início do projeto, a fim de criar características e resultados que possam ser comparados. A principal variação das regras de cobertura em cada projeto envolvem regras de negócio com suas particularidades. No entanto, cada simulação proposta por este trabalho apresentará quais são as regras de cobertura que o projeto está sujeito.

Ao longo da explanação do método foi apresentada uma série de responsáveis pela execução de cada uma das etapas. No entanto, o método possui como filosofia a objetividade, permitindo que haja variação nos papéis que precisam exercer uma dada função. Mesmo assim, vale ressaltar que na construção das bases de componentes e testes é mandatória a presença de especialistas na área de testes para abordar quais são os pontos a serem testados dentro de cada componente.

Por outro lado, a concepção dos componentes, do escopo do teste e dos critérios de completude terá maior efetividade se não for construída pelos próprios analistas de teste, cabendo esta responsabilidade a um analista de negócio ou arquiteto de testes ou do projeto. Quanto à execução, a resposta dos questionários e definição de variáveis pode ser feita tanto por analistas de teste ou analistas de negócio. Finalmente, a validação dos artefatos de teste deve ser feita pelo mesmo responsável pela elaboração do escopo de teste inicial. Quanto à execução dos testes, basta definir um colaborador que esteja apto a seguir o passo a passo documento nos casos de teste.

## 4 SIMULAÇÃO DO MÉTODO

Nesta seção, os modelos de conversão de componentes e geração de documentos de teste são colocados em prova na avaliação de sistemas de diferentes níveis de complexidade. O trabalho propõe a apresentação do estado inicial de componentes de testes, os questionários preenchidos por especialistas, a tradução em casos de testes e finalmente, as melhorias submetidas às bases utilizadas.

### 4.1. Passos de Execução do Método

A Tabela 4 apresenta a sequencia de passos previstos para a execução do método. Os passos descritos abaixo se relacionam com o ciclo apresentado no Diagrama 1 (Figura X), a relação dos passos no diagrama é o número que encontra-se entre parênteses.

Tabela 3 – Passos de Execução da Simulação

#	Passo
1	Selecionar (ou Montar) Base de Cenários de Teste
2	Definir Escopo do Sistema a ser Testado (A1)
3	Elaborar os Cenários e suas Interações (A2)
4	Definir Componentes para cada um dos Cenários (A3)
5	Responder Questionário (A4 e A5)
6	Validar Documentação de Testes (A6 e A8)
7	Validar Diagrama de Interação dos Cenários (Máquina de Estado) (A7 e A8)

Assumiu-se uma base inicial de cenários de testes construída por um especialista em testes, os componentes apresentados foram construídos conforme demanda da simulação em execução.

### 4.2. Validações Sistêmicas – Entrada/Saída de Estoque

Seguindo o mesmo procedimento citado acima, este exemplo visa atender operações básicas de sistema, conforme especificado na Atividade 5 da seção 1.4, Etapas do Trabalho, adicionando fatores de validação da interface. Mesmo que estas validações sofram alterações conforme o domínio de negócio e tecnologia em

que são implementados, elas possuem características comuns que podem ser padronizadas na fase de testes.

Para tanto, conforme descrito no passo 1, foi selecionado a base inicial de componentes descrita no primeiro item do apêndice A. No intuito de capturar diferentes tipos de validação foram estipulados dois cenários de requisitos para a construção de casos de teste dentro desta simulação.

#### Cenário 1 – Criar um Registro

1. Acessar página de *login*;
2. Preencher usuário e senha válidos;
3. Acessar página de registros do estoque;
4. Clicar na opção “Novo Registro”;
5. Preencher os dados do registro sendo cadastrado;
6. Salvar o registro.

#### Cenário 2 – Editar um Registro

1. Acessar página de Login;
2. Preencher usuário e senha válidos;
3. Acessar página de registros do estoque;
4. Selecionar o item cadastrado no cenário 1;
5. Editar os dados do registro;
6. Salvar o registro.

A segunda etapa do método envolve a seleção de componentes de testes para as ações e eventos implícitos dentro dos cenários de requisito, seguindo o passo 2 da execução. Entretanto, nota-se que há a necessidade de uma atualização na base de componentes, pois o modelo está tratando de interfaces diferentes com comportamentos particulares que exigem casos de teste específicos, detalhando o passo 3. Para tanto, a base passa a contemplar dois componentes “Formulários” e “Listas”, reduzindo o esforço de especialistas na remoção de cenários de teste que não se apliquem a interface.

Além disso, adicionam-se alguns cenários de teste ao componente “Listas”, tendo como resultado a base descrita no segundo item do apêndice A. Para os cenários em que não havia necessidade de seleção de cenários e testes, o item é classificado como “Não se Aplica (N/A)”. Os resultados obtidos em nível de componentes de

teste, representados pelo passo 4, são registrados segundo os dados apresentados na Tabela 4.

Tabela 4 – Componentes selecionados para os Cenários

Execução	Cenário 1	Cenário 2
1	N/A	N/A
2	Formulários	Formulários
3	N/A	Listas
4	N/A	N/A
5	Formulários	Formulários
6	N/A	N/A

Em relação aos questionários para dados específicos dos testes, descritos no passo 5, utilizou-se o mesmo conjunto aplicado na primeira simulação. Além disso, houve a necessidade de se construir um novo questionário para atender o novo componente “Listas”. A relação de perguntas está descrita no terceiro item do apêndice A e as respostas dos questionários, representando o passo 5, são apresentadas no quarto item do apêndice A, seguindo o padrão descrito na Tabela 5.

Respostas Questionário – Cenário 1 (Formulário 2):

Tabela 5 – Questionário do Cenário

#	Respostas	
1	Sim	“Código Item” / “Descrição Item” / “Quantidade”
2	Sim	“Quantidade”
3	Não	N/A
4	Não	N/A
5	Sim	“Código” / “Descrição”
6	Sim	“Código” / “Descrição”
7	Não	N/A
8	Não	N/A
9	Não	N/A

A soma dos cenários e respostas dos questionários é submetida ao segundo modelo, onde ocorre a criação dos casos de testes, seguindo o passo 6. O artefato final obtido está apresentado no quinto item do apêndice A, segundo a representação da Tabela 6.

Tabela 6 – Cenário de Teste e Variáveis

<b>Cenário de Testes</b>	<b>Variáveis</b>
<b>Validar Opção de Acesso ao Formulário 1</b>	N/A
<b>Validar Comportamento de Acesso ao Formulário 1</b>	N/A
<b>Salvar Registro sem Preencher Valor no Campo Login</b>	N/A
<b>Salvar Registro Preenchendo Valor " " no Campo Login</b>	N/A
<b>Salvar Registro sem Preencher Valor no Campo Senha</b>	N/A
<b>Salvar Registro Preenchendo Valor " " no Campo Senha</b>	N/A

Durante a execução houve a necessidade da criação de “categorias” dentro de cada componente, o objetivo desta ação era facilitar a remoção ou inserção de um grupo específico de características dentro do componente, resultando em subcomponentes. O estado de cenários de componentes atingido pelo artefato final ocorreu com duas interações do método, tanto para cenários de testes quanto para cenários de requisitos, conforme descrito no passo 7.

#### 4.3. Validações de Interface e Operação – Projeção e Gestão de Custos

Para o primeiro caso de simulação prática foi selecionado um projeto que exige esforço considerável do operador ou usuário do sistema para entrada e manipulação de dados. Com uma interface que possua a interação direta para o lançamento de custos, lançamentos de projeções de valores e emissão de relatórios para controle, o trabalho visa extrair exemplos de interação do método em cenários onde há manipulação constante por parte do usuário. Esta simulação remete a especificação da Atividade 6 da seção 1.4, Etapas do Trabalho.

Neste projeto foram elaborados 36 cenários de requisitos que serão submetidos ao método proposto pelo trabalho. Em paralelo, estes cenários foram detalhados e submetidos a um especialista de testes. Este deveria gerar um caso de teste manualmente para comparação dos resultados ao término da simulação. A simulação detalhada pode ser encontrada no Apêndice B.

Os cenários foram contextualizados inicialmente em um formato macro, seguindo o padrão abaixo.

Cenários da Simulação – Versão Macro



1. Usuário acessa a interface
2. Seleciona o período que deseja avaliar os custos
3. Indica se deseja criar uma nova projeção ou utilizar uma salva
  - 3.1. Cria um nome para a projeção que irá começar
  - 3.2. Pesquisa na base entre as projeções existentes qual será aquela que irá abrir
4. Sistema carrega o real/projetado de custos
5. Sistema apresenta os custos consolidados por departamento
6. Custos reais são apresentados em cinza e custos projetados em azul
7. Usuário define a informação que irá lançar
  - 7.1. Incluir novo registro de custo real
  - 7.2. Incluir novo registro de custo projetado
  - 7.3. Visualiza detalhes de um consolidado de período e departamento
    - 7.3.1. Atualiza valor de um registro de custo real
    - 7.3.2. Atualiza valor de um registro de custo projetado
    - 7.3.3. Converte registro de custo projetado em custo real
8. Sistema apresenta somatória de custos do período por departamento
9. Sistema apresenta somatória de custos do departamento por período
10. Usuário emite relatório da projeção
  - 10.1. Usuário seleciona filtros para o relatório a ser emitido
  - 10.2. Usuário emite relatório sem filtros

O detalhamento destes cenários seguindo o mesmo formato proposta na simulação teórica gerou o seguinte resultado.

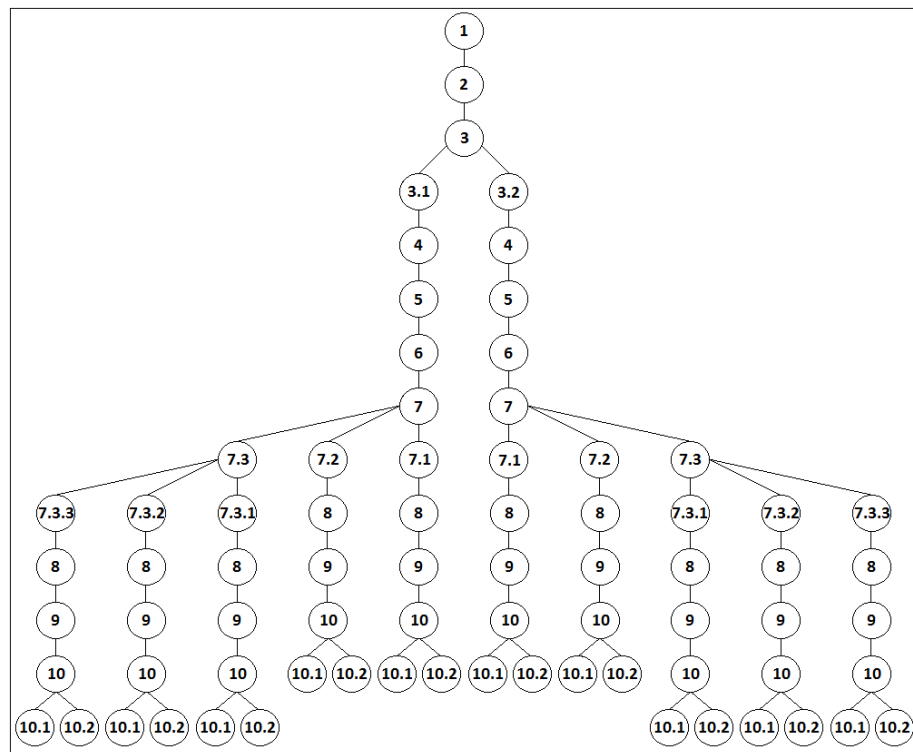
#### Cenários da Simulação – Versão Detalhada

1. Usuário acessa a interface
2. Preenche campo “Período Inicial”
3. Preenche campo “Período Final”
4. Seleciona “Nova Projeção” no campo “Tipo de Projeção”
  - 4.1. Preenche o campo “Descrição”
  - 4.2. Clica na opção “Continuar”
5. Seleciona “Projeção Existente” no campo “Tipo de Projeção”
  - 5.1. Filtra uma das projeções na tabela
  - 5.2. Seleciona uma das projeções na tabela
  - 5.3. Clica na opção “Continuar”
6. Consulta dados lançados na projeção
7. Identifica tipo de dados por cor da interface
8. Clica na opção “Novo Lançamento”
9. Seleciona a opção “Custo Real”
  - 9.1. Preenche a descrição do lançamento
  - 9.2. Preenche o valor do custo
  - 9.3. Clica na opção “Salvar”
10. Seleciona a opção “Custo Projetado”

- 10.1. Preenche a descrição do lançamento
- 10.2. Preenche o valor do custo
- 10.3. Preenche a data da projeção
- 10.4. Clica na opção “Salvar”
- 11. Seleciona um registro do tipo real da projeção
  - 11.1. Atualiza campo Descrição do Registro
  - 11.2. Atualiza campo Valor do Registro
- 12. Seleciona um registro do tipo projetado
  - 12.1. Atualiza campo Descrição do Registro
  - 12.2. Atualiza campo Valor do registro
  - 12.3. Atualiza campo Data do registro
- 13. Seleciona um registro do tipo projetado
  - 13.1. Clicar na opção “Converter para Valor Real”
- 14. Verifica colunas de somatórias na projeção
- 15. Verifica linhas de somatórias na projeção
- 16. Clica na opção “Emitir Relatório”
- 17. Marca ou Desmarca Filtros
- 18. Valida relatório em formato PDF

Uma vez detalhados os cenários de requisitos, a árvore de cenários pôde ser consolidada, seguindo a representação abaixo.

Figura 9 – Árvore de Cenários da Simulação 1



Fonte: Elaborado pelo autor

Ao selecionar um dos episódios na árvore de cenários, gera-se o cenário que será alvo do teste, onde as validações serão efetuadas, conforme o escopo desejado. Na descrição abaixo detalham-se os passos do cenário aplicado “Tornar a projeção de custo “Elétrica” de Setembro, um gasto real”.

1. Usuário acessa a interface
2. Preenche campo “Período Inicial”
3. Preenche campo “Período Final”
4. Seleciona “Projeção Existente” no campo “Tipo de Projeção”
5. Filtra uma das projeções na tabela
6. Seleciona uma das projeções na tabela
7. Clica na opção “Continuar”
8. Consulta dados lançados na projeção
9. Identifica tipo de dados por cor da interface
10. Seleciona um registro do tipo projetado
11. Atualiza campo Valor do registro
12. Atualiza campo Data do registro
13. Clicar na opção “Converter para Valor Real”

Partindo da base padrão de cenários, nota-se a existência de formulários para indicar a projeção que o usuário deseja acessar. Uma vez que a projeção é identificada e o usuário prossegue nos testes, identifica-se um novo componente proveniente da ação “clicar”. No exemplo apresentado trata-se de um botão, mas este artefato da tela pode gerar tanto um evento na interface atual, como auxiliar na navegação para outra interface.

Neste caso, o usuário deseja navegar para outra interface e a interface poderia ter feito esta ação de várias maneiras. Portanto, opta-se por utilizar o componente “Navegação” que pode ser utilizado através de botões, teclas do teclado ou “links”.

Seguindo com a execução do cenário, surge a necessidade de validar uma interação da interface em relação a cores dos registros da lista. O propósito deste realce é alertar ou comunicar algo ao usuário. Neste caso, existem duas possibilidades, o realce ou característica da tela poderia se tornar uma categoria dentro do componente “Listas”, o que implica em replicar tal categoria em todos os componentes que possam necessitar de alertas. A outra opção é construir um componente de alertas ou comunicação para agrupar todos os tipos de resposta que o sistema tenha intenção de passar aos usuários. O resultado está representado na Tabela 7.

Tabela 7 – Cenários Aplicados em cada passo de execução

<b>Execução</b>	<b>Cenário Aplicado – Simulação 1</b>
<b>1</b>	N/A
<b>2</b>	Formulário
<b>3</b>	Formulário
<b>4</b>	Formulário
<b>5</b>	Lista
<b>6</b>	N/A
<b>7</b>	Navegação
<b>8</b>	Lista
<b>9</b>	Alerta
<b>10</b>	Lista
<b>11</b>	Formulário
<b>12</b>	Formulário
<b>13</b>	Navegação

No próximo passo inicia-se o processo de resposta dos questionários. A primeira necessidade é a construção dos questionários para os componentes descritos na Tabela 7 que ainda não possuem questionários padrões, neste caso, os componentes “Navegação” e “Alertas”.

Conforme o método proposto neste trabalho, os dois questionários e os cenários de teste para estes dois componentes serão construídos, de modo a atender as necessidades deste cenário específico. A partir da evolução da base de dados e aplicação em outros projetos, estes componentes serão atualizados gradativamente. Os questionários e cenários gerados podem ser visualizados no Apêndice B.

Na tabela 8 são apresentadas as respostas para os dois questionários gerados nesta simulação. Uma vez finalizado o processo, 37% dos casos de testes gerados foram considerados irrelevantes, ou seja, receberam respostas “não se aplica”, estes envolviam formulários de pesquisa que não tinham a funcionalidade de “Salvar” registros.

Além disso, os especialistas que geraram os cenários de teste aplicaram uma abordagem diferente da simulação conceitual. Neste exemplo, os cenários foram quebrados no nível dos campos e ações, ou seja, na simulação conceitual existe um cenário para preenchimento dos campos do formulário, enquanto que nesta simulação existe um cenário para o preenchimento de cada campo.

Tabela 8 – Respostas para Questionários da Simulação 2

#	Respostas	
1	Sim	“Tela de Projeção” e “botão Continuar”
2	Sim	“Período Inicial” / “Período Final” / “Tipo de Projeção”
3	Sim	“Período Inicial” / “Período Final”
4	Não	N/A

#	Respostas	
1	Não	1
2	Não	2
3	Sim	3

Com o resultado parcial em 47 casos de teste, nota-se que estão pendentes validações essenciais para estes cenários. Estas pendências se enquadram na categoria de “Regras de Negócio”. Até esse ponto do trabalho, as validações foram elaboradas considerando validações de interface, navegação e formato dos campos.

Um exemplo dos casos de teste obtidos pode ser analisado na Tabela 9, que representam as validações do momento em que o custo projetado está sendo convertido para custo real.

Tabela 9 - Cenários de Testes e Variáveis de Personalização

<b>Cenário de Testes</b>	<b>Variáveis</b>
<b>Validar Acesso à Detalhes do Registro</b>	N/A
<b>Salvar Registro sem Preencher Valor no Campo Obrigatório "X"</b>	X = Valor do Registro
<b>Salvar Registro Preenchendo Valor " " no Campo Obrigatório "X"</b>	X = Valor do Registro
<b>Salvar Registro sem Preencher Valor no Campo Obrigatório "Y"</b>	Y = Data do Registro
<b>Salvar Registro Preenchendo Valor " " no Campo Obrigatório "Y"</b>	Y = Data do Registro
<b>Checar Validação do Campo Numérico "X" ao Preencher Valor "abc"</b>	X = Valor do Registro
<b>Checar Validação do Campo Numérico "X" ao Preencher Valor "10,00"</b>	X = Valor do Registro
<b>Checar Validação do Campo Numérico "X" ao Preencher Valor "10.00"</b>	X = Valor do Registro
<b>Checar Validação do Campo Data "Y" ao Preencher Valor "01/01/2001"</b>	Y = Data do Registro
<b>Checar Validação do Campo Data "Y" ao Preencher Valor "01-01-2001"</b>	Y = Data do Registro
<b>Checar Validação do Campo Data "Y" ao Preencher Valor "01/01/01"</b>	Y = Data do Registro
<b>Checar Validação do Campo Data "Y" ao Preencher Valor "31/02/01"</b>	Y = Data do Registro
<b>Validar se o sistema permite a mudança de página com o campo "X" preenchido</b>	X = Valor do Registro
<b>Validar se o sistema permite a mudança de página com o campo "Y" preenchido</b>	Y = Data do Registro
<b>Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"</b>	X = Tela de Projeção Y = Converter para Valor Real

Embora haja a transição de página após a conversão do valor e a validação das informações que devem ser preenchidas na tela, o caso de teste não ressalta uma validação posterior de que o evento de conversão foi bem sucedido. Por este motivo, é necessária a configuração da própria regra de negócio seguindo a filosofia que esta deve ser o mais genérica possível para facilitar a reutilização. Não existe uma

regra para a efetividade do método em relação a quão genérico os cenários, questionários e testes precisam ser. No entanto, vale ressaltar que uma base muito genérica tem como consequência maior esforço na execução do método e nas várias iterações para adição de novos cenários. Por outro lado, uma base muito específica gera casos de testes muito extensos ou exige a necessidade de investimento de esforço na seleção de cenários que não se aplicam para cada componente.

Para solucionar esta questão, um dos caminhos viáveis é tratar toda a ação dentro do sistema como entrada e transição de dados, ou seja, para ocorrer um evento, uma modificação precisa ser efetuada sobre um dado armazenado no sistema. Nesta solução não se leva em consideração eventos e funcionalidades aplicadas diretamente a objetos na tela sem alteração dos dados efetivamente. No entanto, adotando o conceito previamente discutido, caso surja um cenário, onde esta validação é necessária, a base deve ser atualizada. A tabela C apresenta o questionário para cenários em que há atualizações no sistema.

Tabela 10 – Questionários Atualizados na Segunda Iteração

<b>Questionário (Atualização)</b>	<b>Respostas</b>		
<b>Existem valores do registro que precisam ser alterados?</b>	Sim	Não	Quais?
<b>Os novos valores da alteração são conhecidos?</b>	Sim	Não	Quais?
<b>Existem valores parametrizados automaticamente no sistema?</b>	Sim	Não	Quais?
<b>Existem restrições de valores para as alterações serem feitas?</b>	Sim	Não	Quais?
<b>Existem condições do sistema que impeçam a alteração?</b>	Sim	Não	Quais?

#### 4.4. Validações de Regras de Negócio – Integração de Sistemas

Na segunda simulação, o trabalho apresenta um exemplo onde há menos interação com o usuário e que possui um elevado número de regras de negócio que devem ser executadas e validadas pelo próprio sistema. O projeto em questão tem como escopo a integração de duas bases de dados com artigos de varejo. O sistema legado e o atual foram idealizados com conceitos diferentes, logo as bases de dados diferem na modelagem e relacionamentos dos objetos.

Seguindo o mesmo molde da primeira simulação, este projeto foi traduzido em 33 cenários de requisitos e foram entregues a outro especialista de testes para a

construção manual do caso de testes. A simulação completa pode ser encontrada no Apêndice C.

Novamente, os cenários foram contextualizados em um formato macro, conforme padrão abaixo.

#### Cenários da Simulação – Versão Macro

1. Usuário acessa a interface
2. Usuário inicia processo de integração
3. Sistema busca todos os registros na base do sistema legado
4. Sistema insere registros em sua base sob um padrão sem tratamento
5. Usuário valida a importação dos registros
  - 5.1. Usuário identifica registros com problemas e os corrige
  - 5.2. Usuário avança para a próxima etapa ignorando registros com erro
  - 5.3. Usuário avança para a próxima etapa sem problemas pendentes
6. Sistema preenche dados para informações cujo relacionamento entre bases seja de 1 para 1
7. Sistema calcula para cada registro do sistema legado, o número de registros a serem gerados na nova base
8. Usuário valida a importação dos registros
  - 8.1. Usuário identifica registros com problemas e os remove
  - 8.2. Usuário avança para a próxima etapa ignorando registros inconsistentes
  - 8.3. Usuário avança para a próxima etapa sem inconsistências
9. Sistema preenche dados para informações cujo relacionamento entre bases seja de 1 para N
10. Usuário valida a importação dos registros
  - 10.1. Usuário identifica registros com problemas e os corrige
  - 10.2. Usuário avança para a próxima etapa ignorando registros com erro
  - 10.3. Usuário avança para a próxima etapa sem problemas pendentes
11. Sistema grava os registros na base e encerra a integração
12. Sistema notifica o usuário quanto a conclusão do processo

O detalhamento destes cenários teve o seguinte resultado.

#### Cenários da Simulação – Versão Detalhada

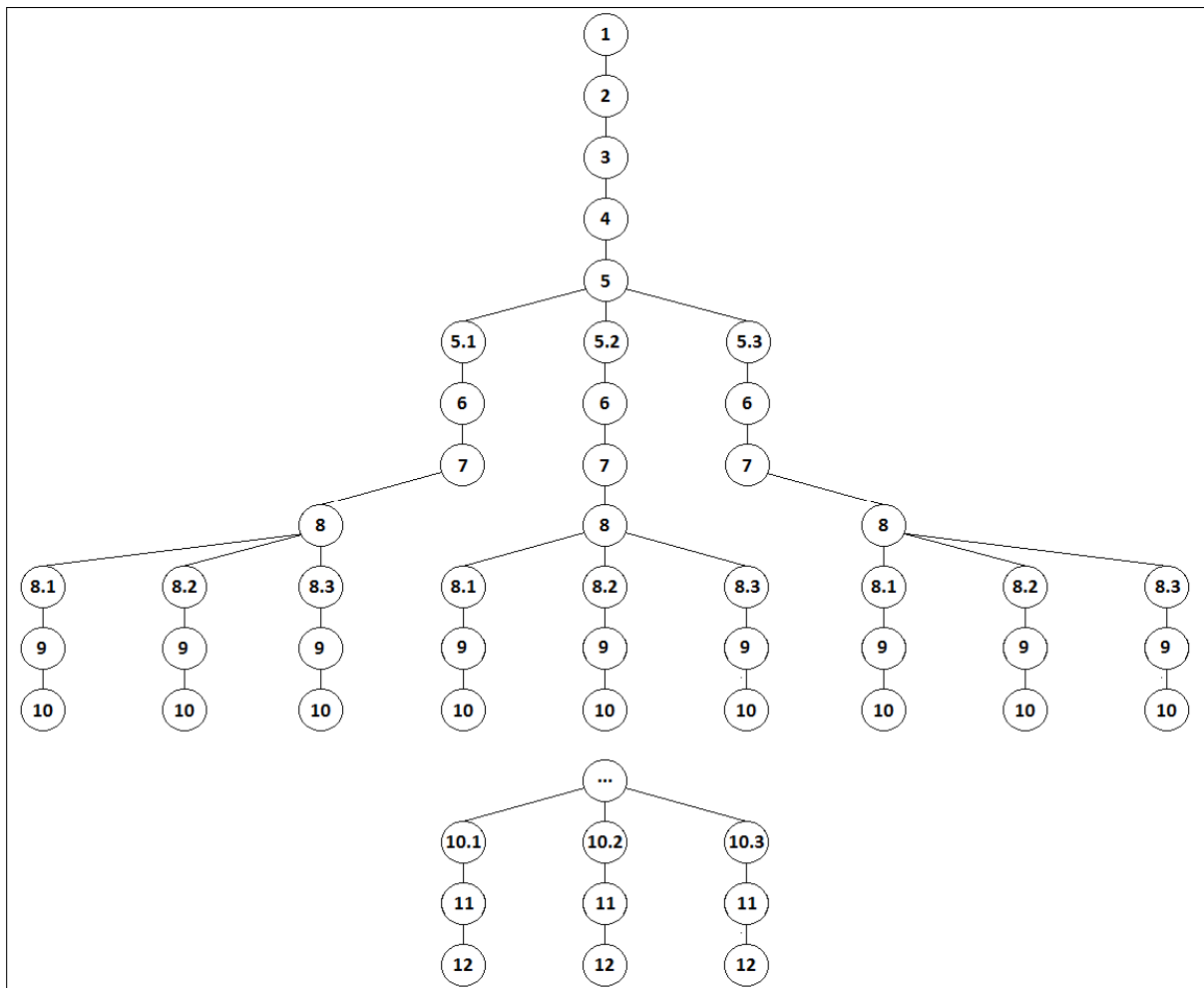
1. Acessa a interface
2. Clicar no botão “Iniciar Integração”
3. Validar registros em importação listados na tabela da interface
4. Comparar registros listados para importação com dados da base de dados do sistema legado
5. Validar registros salvos na base de dados do sistema atual
6. Marca registros em vermelho que deseja efetuar a correção



7. Clicar na opção “Exportar Excel” com os registros a serem ajustados
8. Clicar na opção “Importar Planilha” com os ajustes efetuados nos registros
9. Validar alterações nos registros conforme mudanças efetuadas na planilha
10. Clicar na opção “Avançar” ignorando problemas pendentes nos registros
11. Validar o total de registros que avançam para a próxima etapa
12. Clicar na opção “Avançar” após corrigir problemas nos registros
13. Validar o total de registros que avançam para a próxima etapa
14. Validar a conversão dos registros com o valor “Primavera” e “Verão” no campo “Estação” para o valor “Summer”
15. Validar a conversão dos registros com o valor “Outono” e “Inverno” no campo “Estação” para o valor “Winter”
16. Validar a conversão dos campos data que estejam no formato “DD-MM-AAAA” para o formato “MM-DD-AAAA”
17. Validar a combinação dos valores dos campos “Descrição 1” e “Descrição 2” no campo “Descrição do Item”
18. Validar a conversão dos valores “Verdadeiro” e “Falso” para os valores “1” e “0”, respectivamente em todos os campos
19. Verificar a quantidade de replicações por registro, segundo a quantidade de lojas em que este é comercializado
20. Verificar a quantidade de replicações por registro, segundo a quantidade de variações de código de barras que este possui
21. Marca registros em vermelho que deseja remover como replicação da original
22. Clicar na opção “Remover Registros”
23. Clicar na opção “Avançar” após remover registros em que haja ambiguidade
24. Validar o total de registros que avançam para a próxima etapa
25. Validar o preenchimento do campo “Loja” através dos relacionamentos do registro original
26. Validar o preenchimento do campo “Código de Barras” através dos relacionamentos do registro original
27. Validar a conversão do valor do campo “Loja” para seu respectivo código, segundo tabela de referência da base de dados
28. Validar o preenchimento do valor “00000” em combinação com o valor presente no campo “Código de Barras”
29. Clicar na opção “Salvar Registros” gerados
30. Aguardar o processo de armazenamento na base de dados
31. Aguardar o alerta de término do processo de integração
32. Verificar a viabilidade de se identificar os registros da nova base de dados partindo do registro da base de dados legada
33. Verificar a viabilidade de se identificar o registro da base de dados legada partindo do registro da nova base de dados

Finalmente, a árvore de cenários resultante pode ser validada abaixo.

Figura 10 - Árvore de Cenários da Simulação 2



Fonte: Elaborado pelo autor

Cenário Aplicado – Importar as frutas da estação de outono para o novo sistema removendo os registros com código de barra do tipo oito dígitos

1. Acessa a interface
2. Clicar no botão “Iniciar Integração”
3. Validar registros em importação listados na tabela da interface
4. Comparar registros listados para importação com dados da base de dados do sistema legado
5. Validar registros salvos na base de dados do sistema atual
6. Clicar na opção “Avançar” após corrigir problemas nos registros
7. Validar o total de registros que avançam para a próxima etapa
8. Validar a conversão dos registros com o valor “Outono” e “Inverno” no campo “Estação” para o valor “Winter”
9. Validar a conversão dos campos data que estejam no formato “DD-MM-AAAA” para o formato “MM-DD-AAAA”

10. Validar a combinação dos valores dos campos “Descrição 1” e “Descrição 2” no campo “Descrição do Item”
11. Validar a conversão dos valores “Verdadeiro” e “Falso” para os valores “1” e “0”, respectivamente em todos os campos
12. Verificar a quantidade de replicações por registro, segundo a quantidade de variações de código de barras que este possui
13. Marca registros em vermelho que deseja remover como replicação da original
14. Clicar na opção “Remover Registros”
15. Clicar na opção “Avançar” após remover registros em que haja ambiguidade
16. Validar o total de registros que avançam para a próxima etapa

Nesta simulação parte-se do princípio que a simulação anterior foi finalizada em sua totalidade e os responsáveis por testes possuem a sua disposição os cenários e casos de teste resultantes da simulação de gestão de custos. Portanto, cenários de alertas e regras de negócio serão aplicados a partir da primeira iteração do método.

Tabela 11 – Cenários Aplicados em cada passo de execução

<b>Execução</b>	<b>Cenário Aplicado – Simulação 2</b>
<b>1</b>	N/A
<b>2</b>	Navegação
<b>3</b>	Listas
<b>4</b>	Inserção
<b>5</b>	Inserção
<b>6</b>	Navegação
<b>7</b>	N/A
<b>8</b>	Atualização
<b>9</b>	Atualização
<b>10</b>	Atualização
<b>11</b>	Atualização
<b>12</b>	Listas / Replicação
<b>13</b>	Alertas
<b>14</b>	Navegação / Listas
<b>15</b>	Navegação
<b>16</b>	Listas

No cenário número 4, a descrição exige a validação dos dados que estão presentes na listagem do cenário número 3 em relação aos dados presentes no banco de dados do sistema legado. Seguindo o mesmo conceito da simulação anterior, utiliza-se o componente “Inserção”. No entanto, o cenário número 5 exige a mesma validação para a nova base de dados. Em ambos os casos aplica-se o componente Inserção, a diferença entre os dois testes será a forma como são executados no passo a passo.

Para o cenário número 4, a execução deve deixar claro que o testador deve checar os dados presentes na interface e compará-los com o banco de dados legado. Enquanto que no cenário número 5, a execução deve focar na presença das informações no banco de dados do novo sistema.

Nota-se a necessidade de duas alterações imediatas nos componentes. No caso das listas, todas validações tem como alvo, questões de paginação, pesquisa e filtros. No entanto, não há validações para situações onde o usuário interage com a lista ou com os registros da mesma.

Outra mudança concentra-se nas Regras de Negócio, pois o cenário número 12 solicita a validação de registros sendo replicados. No entanto, replicar registros pode agir tanto como atualização como na inserção, o que pode gerar dúvidas sobre o melhor componente a ser escolhido. Para tanto foram feitas duas alterações, a primeira envolve o questionário para o componente “Questionário” adicionando condições para alterações de dados, conforme descrito na Tabela 12. A outra alteração tem como escopo os novos cenários de teste vinculados ao componente “Replicação”, por necessidade do cenário número 12, representada na Tabela 12.

Tabela 12 – Questionários Atualizados na Segunda Iteração

<b>Questionário (Atualização)</b>	<b>Respostas</b>		
<b>Existem valores do registro que precisam ser alterados?</b>	Sim	Não	Quais?
<b>Os novos valores da alteração são conhecidos?</b>	Sim	Não	Quais?
<b>Existem valores parametrizados automaticamente no sistema?</b>	Sim	Não	Quais?
<b>Existem condições a serem atendidas para a parametrização?</b>	Sim	Não	Quais?
<b>Existem restrições de valores para as alterações serem feitas?</b>	Sim	Não	Quais?
<b>Existem condições do sistema que impeçam a alteração?</b>	Sim	Não	Quais?

Tabela 13 – Casos de Teste Adicionados na Segunda Iteração

<b>Componente</b>	<b>Categoria</b>	<b>Cenário de Testes</b>
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na interface
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na base de dados
<b>Regras de Negócio</b>	Atualização	Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento, se condição "Z" for verdadeira.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite a inclusão do valor "Y" para o campo "X" ao salvar.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite que o registro seja atualizado, caso "X" seja verdadeiro.
<b>Regras de Negócio</b>	Replicação	Validar se ambos os registros possuem todos os dados idênticos exceto o campo "X"
<b>Regras de Negócio</b>	Replicação	Validar a existência de "N" registros para cada "X" no registro original

Embora esta simulação tenha apresentado um número menor de casos de testes, dado o cenário estipulado no escopo, nota-se que seu maior foco envolve os cenários do componente de Regras de Negócio e elícita um número menor de validações que precisam ser feitas na interface.

## 5 ANÁLISE DOS RESULTADOS

Nesta seção, o trabalho apresenta os resultados obtidos com as simulações propostas e avalia as expectativas apresentadas nos capítulos 2 e 3, conforme especificado na Atividade 7 da seção 1.4, Etapas do Trabalho. Além disso, os resultados quantitativos e qualitativos do trabalho também serão descritos nesta seção.

### 5.1. Simulações Práticas

O trabalho apresentou duas situações práticas onde são aplicados modelos para a construção de casos de teste. Como dito anteriormente no texto, o objetivo é buscar uma linha de pensamento lógico capaz de validar o alvo do teste. As duas simulações seguiram o mesmo procedimento de execução, a base de componentes e testes já existente seria aplicada ao cenário do escopo e as melhorias seriam aplicadas até que se atinja o nível de cobertura esperado.

Na primeira simulação, a base de componentes e testes utilizada foi o artefato resultante das simulações conceituais de gestão de estoque. Esta possuía principalmente validações de interface e interação com o usuário, mas pouco se atentava a eventos de reação por parte do sistema ou até as próprias regras de negócio existentes em qualquer sistema.

Ao término da primeira simulação prática, apresentada na passagem do item 4.3 para o 4.4, a nova base de componentes aumentou em 20% os cenários de teste, apenas com a execução de uma estória da árvore de componentes gerada pelo escopo. Além disso, os textos relacionados para os testes foram atualizados e em alguns casos, como o componente “Formulário” que não foi capaz de conter as validações para “Navegação”, houve quebra dos cenários para que cada um pudesse contemplar um número diferente de variáveis. Outro exemplo é o componente “Alertas” que podia se enquadrar tanto no “Formulário” quanto “Listas”.

A primeira massa de componentes possuía 40 cenários de teste, ao fim da primeira interação dois novos componentes surgiram adicionando 8 cenários de teste. No entanto, foi discutido que as regras de negócio foram deixadas de lado, portanto um novo componente de teste foi criado para suprir essa necessidade com dois

subgrupos. Este componente gerou um novo aumento proporcional de 16,6% na quantidade de cenários de teste.

Nota-se uma tendência em que a cada interação do método, a quantidade de novos cenários diminui. Isso ocorre, pois os mesmos cenários se aplicam à outras variações da árvore de cenários ou porque o nível de completude esperado por especialistas de testes foi atingido. Em busca de diferentes variações na base de cenários de teste, a mesma base utilizada ao término da segunda interação na primeira simulação foi aplicada como base da segunda simulação.

Por outro lado, a segunda simulação teve um crescimento diferente da primeira, o número de componentes não cresceu, mas cenários de teste novos surgiram para estes componentes. Isso ocorreu, pois esta simulação teve como escopo, uma combinação entre as interfaces das duas simulações anteriores, a conceitual e a primeira prática, respectivamente, e uma forte dosagem de regras de negócio. A menos que a base seja sujeita a um escopo totalmente diferente, esta tendência irá persistir. Um exemplo de mudança radical no escopo seria a validação de um sistema sob a ótica de usabilidade, ignorando todas as validações de regras de negócio.

A primeira interação teve um aumento de 14,3% de cenários de teste dentro dos componentes já existentes. A diferença notável é o surgimento de mais um subgrupo do componente de “Regras de Negócio”. As regras descritas no cenário de requisito são específicas para a execução de seu domínio. Por esse motivo foi necessário um maior investimento de tempo na construção dos cenários de teste, já que a construção de cenários genéricos deve ser assumida como premissa do método aplicado.

## 5.2. Simulações em Números

O trabalho propõe a aplicação do método partindo de uma base de componentes e testes inexistente. Deste modo é necessário um número de iterações do método, buscando cenários e testes comumente utilizados por especialistas, para atingir uma base em que seja possível visualizar um caso de teste. No entanto, esta tarefa é

bastante intuitiva para os especialistas em testes levando em conta os modelos mentais que estão habituados a utilizar.

Para a simulação teórica, houve a necessidade de 3 (três) aplicações do primeiro modelo, a fim de aumentar o desempenho dos componentes de teste. Enquanto que para o segundo modelo, mais 4 (quatro) interações se mostraram necessárias, aumentando o número de restrições e cenários vinculados a componentes.

Na primeira simulação prática foram aplicadas 3 iterações do método partindo da base de componentes do modelo teórico. A primeira não teve correções, a segunda abordou os novos componentes de validação da interface, enquanto que a terceira teve como foco regras de negócio. Também na segunda iteração, os questionários foram atualizados para constar as alterações nos componentes de teste e no total foi gerado um caso de testes com 51 cenários para validação.

Em relação ao tempo de execução, a primeira iteração com a base de dados importada da simulação teórica foi executada em 47 minutos, onde a seleção dos componentes levou 16 minutos, o preenchimento das respostas do questionário levou 22 minutos e a definição dos valores para as variáveis de teste levou 9 minutos. Na segunda iteração, o tempo total foi reduzido para 20 minutos, onde 4 minutos foram usados na seleção dos componentes, outros 9 minutos para as respostas do questionário e mais 7 minutos para as variáveis de teste. Na última iteração, o tempo foi quase idêntico a segunda iteração, totalizando quase 90 minutos para toda a execução.

Além disso, é necessário contabilizar o tempo investido nas correções a cada iteração, nesta simulação cada alteração no componente de teste tem como consequência alteração em seu respectivo questionário e inclusão de variáveis. Entre as iterações foram adicionados novos 16 cenários de teste e seus componentes. Na média cada atualização levou 8 minutos, ou seja, pouco mais de 2 horas foram utilizadas por analistas de teste na atualização das bases de dados. Na segunda correção, aproximadamente 7 cenários foram atualizados, investindo-se mais 50 minutos. Ao concluir a primeira simulação prática foram investidas aproximadamente 4 horas e vinte minutos, considerando execução e atualizações. O resumo encontra-se na tabela 14, representada a seguir.



Tabela 14 – Resumo de Tempo de Execução da Primeira Simulação

# Interação	Seleção de Componentes	Responder Questionários	Definir variáveis	Atualizações	Tempo de Execução
1	16 minutos	22 minutos	9 minutos	126 minutos	173 minutos
2	4 minutos	9 minutos	7 minutos	50 minutos	70 minutos
3	4 minutos	9 minutos	7 minutos	N/A	20 minutos

Na segunda simulação prática foram aplicadas 2 iterações do método partindo da base de componentes gerada na simulação anterior. A primeira não teve correções e a segunda abordou os novos subgrupos de componentes e cenários de teste. Também na primeira iteração, os questionários foram atualizados para constar as alterações nos componentes de teste e no total foi gerado um caso de testes com 40 cenários para validação.

Nos resultados para o tempo de execução, a primeira iteração com a base de dados importada da simulação teórica foi executada em 42 minutos, onde a seleção dos componentes levou 13 minutos, o preenchimento das respostas do questionário levou 17 minutos e a definição dos valores para as variáveis de teste levou 12 minutos. Na segunda iteração, o tempo total foi reduzido para 24 minutos, onde 5 minutos foram usados na seleção dos componentes, outros 9 minutos para as respostas do questionário e mais 10 minutos para as variáveis de teste.

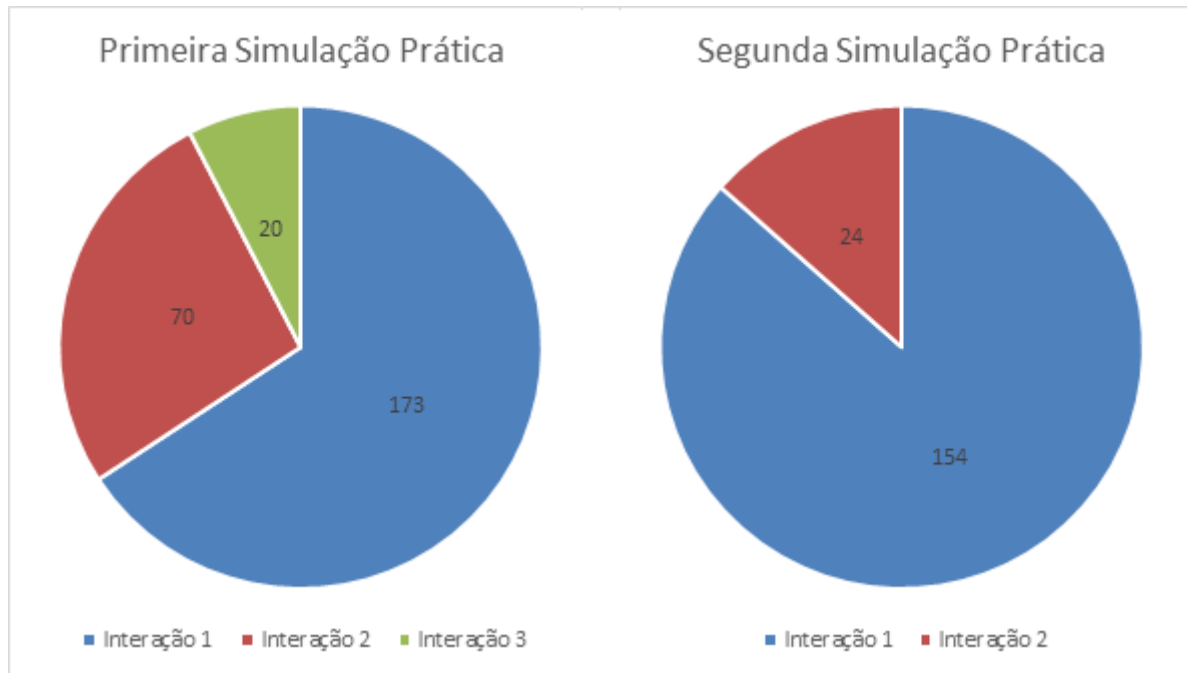
Diferente da primeira simulação prática, as correções e atualizações de componentes e questionários foram mais exaustivas na segunda simulação, pois envolveram principalmente regras de negócio. Entre as iterações foram adicionados novos 8 cenários de teste e seus componentes, na média cada atualização levou 14 minutos, ou seja, pouco menos de 2 horas foram utilizadas por analistas de teste na atualização das bases de dados.

Ao concluir a segunda simulação prática foram investidas aproximadamente 3 horas, considerando execução e atualizações. O resumo encontra-se na tabela 15, representada abaixo. Um comparativo entre as duas simulações também é feito na Figura 11, em relação ao tempo total de execução a cada interação do método.

Tabela 15 – Resumo de Tempo de Execução da Segunda Simulação

# Interação	Seleção de Componentes	Responder Questionários	Definir variáveis	Atualizações	Tempo de Execução
1	13 minutos	17 minutos	12 minutos	112 minutos	154 minutos
2	5 minutos	9 minutos	10 minutos	N/A	24 minutos

Figura 11 – Resumo de Tempo Total de Execução em Minutos



Fonte: Elaborado pelo autor

### 5.3. Análise dos Resultados

Ao comparar os resultados obtidos nas duas simulações, a expectativa em relação a tempo era uma redução considerável no período da execução da segunda simulação. No entanto, o tempo ficou bastante similar apesar do menor número de cenários no caso de testes. Isso se deve devido a diferença na quantidade de validações de interface e regras de negócio. O número de validações de interface na primeira simulação era muito maior em relação a segunda simulação.

Portanto, assume-se que há uma variação no tempo de geração do artefato de testes, em relação a proporção de cenários referentes a regras de negócio. Além disso, houve variação considerável no tempo das atualizações de componentes e questionários que envolviam regras de negócio. Cenários de teste para interfaces ou validações unitárias são simplistas e é possível identifica-los com um número

pequeno de interações. No entanto, cenários de negócio exigem um maior esforço de elaboração, pois estes tendem a se tornar muito específicos com o sistema alvo do teste. Quanto mais específico for um componente ou o caso de teste, menor é a probabilidade de chance de reaproveitá-lo totalmente em outros sistemas.

Como parte do escopo do trabalho, os dois casos de teste resultantes também foram construídos de forma manual por outros analistas de teste. O objetivo era comparar o tempo total de construção do artefato de teste com o método proposto pelo trabalho e o processo manual.

Tomando como referência a cobertura dos testes, quando os dois casos de teste atendessem as premissas estipuladas pelos analistas de teste, seriam medidos o tempo total de construção e a qualidade dos artefatos gerados.

O caso de teste gerado na primeira simulação seguindo o procedimento manual levou aproximadamente 40 minutos, ou seja, 85% do tempo total utilizado pela execução do método do trabalho. No entanto, cada reconstrução do caso de teste manualmente exigiu pelo menos 30 minutos, pois os ajustes impactam a execução do caso como um todo, este tempo representa 23% do tempo levado pela primeira atualização do método. As diferenças entre o tempo total de execução no formato manual e com a aplicação do método são apresentadas na Tabela 16.

Na segunda simulação, o procedimento manual levou aproximadamente 35 minutos, ou seja, 83% do tempo total utilizado pela execução do método do trabalho. No entanto, cada reconstrução do caso de teste manualmente exigiu pelo menos 25 minutos, cerca de 31% do tempo da primeira atualização, pois os ajustes impactam a execução do caso como um todo.

Abordando esforços de execução, na primeira simulação concluiu-se que a partir da quinta iteração, o método proposto no trabalho passa a ter um custo em esforço menor que o procedimento manual. Enquanto isso, na segunda simulação a mesma quantidade de interações também é necessária para tornar o método mais benéfico em relação a custo de esforço. O critério de parada foi o momento em que o esforço em tempo passa a ser menor na utilização do método e o momento em que o critério de cobertura foi atingido.

Tabela 16 – Comparação de Execuções dos Testes nas duas Situações

# Interação	Execução Manual	Atualização Manual	Execução do Método	Atualização do Método
1	40 minutos	30 minutos	47 minutos	126 minutos
2	40 minutos	30 minutos	20 minutos	50 minutos
3	40 minutos	30 minutos	20 minutos	19 minutos
4	40 minutos	30 minutos	20 minutos	8 minutos
5	40 minutos	30 minutos	20 minutos	-

Comparando todos os casos de teste gerados manualmente é considerável a discrepância no formato e padrão de escrita de cada analista de teste. No caso da execução do método proposto neste trabalho, todos os casos de teste foram apresentados com o mesmo padrão de escrita, viabilizando um formato mais unificado do artefato de testes.

## 6 CONCLUSÕES

Este trabalho visou a implantação de um método capaz de automatizar o processo de construção de casos de teste. Para tanto existe a necessidade de se quebrar barreiras de investimentos e custos que são comuns em projetos. O trabalho apresenta uma alternativa a esta problemática através de uma implantação gradativa que faça com o que o processo automatizado seja amadurecido aos poucos, reduzindo a resistência.

Para este trabalho foram selecionados três simulações de áreas de negócio bem distintas. Entretanto, no momento da seleção do escopo do teste, o critério descrito abaixo foi utilizado para aproximar as simulações práticas. Embora as simulações pertençam a áreas de negócio diferentes, os esforços para construção, manutenção e execução são similares. Ao mesmo tempo, devido à discrepância das áreas dos sistemas, foi possível coletar dados acerca do crescimento da base de cenários e testes quando aplicada a diferentes objetivos de testes e critérios de cobertura.

Diferente dos resultados obtidos na simulação teórica, as duas simulações práticas agregaram conteúdo em relação a diferenças de esforços entre testes de interface e regras de negócio. Inicialmente, o padrão de listagem dos componentes era classificado com somente um nível, porém a mudança para classificação em vários níveis auxiliou no processo de reutilização dos cenários de teste.

Essa mudança na quantidade de níveis de classificação ocorreu devido a simplicidade da simulação teórica e até o início da execução da primeira simulação prática, ainda não havia surgido necessidade de investir tempo considerável na seleção dos testes que se aplicavam ao cenário.

As simulações trouxeram resultados que mostram um número mínimo de iterações para começar a apresentar benefícios em esforço de construção dos artefatos de teste, durante as simulações foram identificadas cinco iterações como mínimo. No entanto, as simulações provaram o que havia sido afirmado no início do projeto, conforme apresentado na seção 5 de análise dos resultados. O investimento inicial de esforço na construção de uma base de testes que atenda os critérios de completude dos analistas é essencial.

A utilização de cenários de requisitos, principalmente na representação em árvore agregou benefícios ao método, pois foi capaz de representar a execução dos casos de testes em um formato bastante detalhado. Na utilização de outros formatos de documentação de escopo e requisitos, muitas vezes a especificação dos detalhes e o fluxo de informação pode não ficar claro e como consequência pode-se gerar dúvida na escolha dos componentes de testes. No entanto, este trabalho não realizou nenhuma simulação com outro formato de representação de requisitos, o que impossibilita a comparação entre os resultados que seriam obtidos, deixando esta como uma proposta de trabalho futuro a ser detalhada nos próximos parágrafos.

Por outro lado, o trabalho também mostrou que é possível construir e aprimorar as bases de cenários, componentes e testes aos poucos, no cenário onde empresas não tem como investir esforço na implantação do método. A maneira encontrada para viabilizar a construção foi focar somente nos cenários que precisam ser validados, sem ter que se preocupar com todas as validações que precisariam ser consideradas para cobrir totalmente um componente. Um exemplo disto foi a árvore de cenários apresentados em que apenas alguns episódios foram selecionados para a simulação. Finalmente, nos aspectos qualitativos, o método conseguiu gerar artefatos de testes padronizados e auxiliam os analistas de teste a evitar situações em que esqueçam cenários de teste que precisem ser validados.

Além disso, vale ressaltar a importância na comunicação entre os profissionais de teste, pois passam a compartilhar uma base de conhecimentos de teste que está em constante evolução, acompanhando o crescimento das empresas.

Este trabalho combinou as boas práticas de várias técnicas empregadas atualmente no mercado. Entre elas estão o levantamento de requisitos baseado em cenários, testes baseado em modelos e finalmente a automação dos testes. Ao iniciar um novo projeto de sistemas, uma das tarefas vitais é a comunicação entre especialistas em sistemas e usuários que dominam o negócio. A técnica de levantamento de requisitos que é empregada mostra-se essencial para sanar dúvidas e facilitar esta comunicação, tornando mais claros objetivos do sistema e testes a serem efetuados. Como consequência é mais simples ultrapassar a barreira

da falta de profissionais especialistas, um dos empecilhos citados no início deste trabalho.

Essa facilidade impacta diretamente as outras duas técnicas utilizadas, pois com maior entendimento, a chance de erros no modelo é menor e conforme Graham e Fewster (2012), quanto maior o entendimento dos objetivos do teste e da automação, maior é a taxa de sucesso da utilização desta técnica. Este aspecto foi afirmado neste trabalho ao decompor o sistema ou processo em partes menores e a organizarmos na árvore de cenários para facilitar o entendimento.

No caso da utilização de testes baseados em modelo, a decisão surtiu o efeito esperado em relação a padronização dos testes. Mesmo que haja padronização na execução dos testes e que todas as regras de cobertura tenham sido atendidas, cada analista de testes tem a sua forma de escrever e documentar seus testes. Uma vez que é imposto a aplicação de um modelo comum a todos os envolvidos do projeto, os analistas conseguem se desvincular de seus modelos mentais e aplicar seu conhecimento em modelos concretos e compartilhados por outros analistas.

Finalmente, a automação de testes é de extrema importância para reduzir esforço repetitivo e falhas operacionais que estão envolvidas nas rotinas diárias dos analistas de teste. Estas vantagens já foram apresentadas nas primeiras seções deste trabalho, no entanto, a implantação gradativa foi responsável por surtir maior efeito neste trabalho. Todas as simulações partiram dos resultados da simulação anterior, sendo que a primeira recebeu um investimento mínimo para a criação inicial da base de testes.

A implantação completa deste processo depende diretamente do levantamento de todos os componentes e cenários de teste atrelados a eles. Sem contar, a especificação de todas as variáveis padronizadas que devem ser utilizadas para a validação dos cenários de teste. Este trabalho precisa ser efetuado através da análise do histórico de projetos e combinação dos conhecimentos dos analistas de teste. No caso desta dissertação de mestrado, apenas os cenários necessários para aquele momento tiveram seu esforço contabilizado, posteriormente essa base crescerá conforme as necessidades surgirem no projeto.

## 7 EXTENSÕES DE PESQUISA E TRABALHOS FUTUROS

Embora os caminhos percorridos por este trabalho tenham atingido o resultado esperado, uma série de fatores foram identificados ao longo da pesquisa que poderiam gerar resultados completamente diferentes. Alguns destes fatores foram identificados no início do projeto durante a definição do escopo, enquanto outros foram se revelando ao longo da execução da pesquisa.

Inicialmente, vale citar a definição do método para elicitación dos requisitos de sistema, que para este trabalho o método selecionado foi a abordagem por cenários. Esta decisão deu-se ao fato que a estrutura de árvore de cenários e episódios representa de forma fiel, os passos de execução de um caso de teste. Além disso, segue a filosofia empregada neste trabalho de objetividade e proximidade com o usuário, uma vez que a principal preocupação está voltada para a implantação gradativa. No entanto, não foi provado que a utilização de outros métodos de levantamento de requisitos seja inviável, nem os benefícios que a utilização ou combinação de outros métodos poderia trazer para a automação dos casos de teste. Uma das propostas é a combinação ou substituição da técnica empregada para a coleta dos requisitos de sistema.

Um fator levantado durante a execução da pesquisa foi a necessidade da criação de subdivisões para a classificação dos componentes de teste. Inicialmente, haveria apenas um nível de classificação como “Listas”, “Formulários”, “Pesquisa” e assim por diante. No entanto, esse nível ficou muito genérico e gerou um esforço desnecessário na seleção dos testes que se aplicavam ao cenário. Embora tenham sido selecionados dois níveis de classificação para este trabalho, não houve avaliação quanto ao impacto nos questionários e nos casos de teste resultantes em caso de um número maior de níveis de classificação. Portanto, outra ideia de extensão é a comparação entre a construção das bases de teste utilizando diferentes quantidades de níveis de classificação dos componentes de teste.

Neste trabalho, três exemplos de sistema foram selecionados, dentro de cada um deles foi retirado um episódio com cenários distintos e com um esforço de execução equivalente. No entanto, cada um dos grupos de cenários possuíam validações razoavelmente diferentes, de modo a validar o crescimento das bases de cenários e testes. O trabalho mostrou a redução do crescimento quando a base é aplicada em



validações similares, portanto seria válido mensurar os casos de teste gerados ao aplicar o método em um projeto de sistema completo. Além disso, é válido acompanhar o crescimento da base ao longo de todo o projeto.

Como trabalhos futuros, este trabalho sugere um maior investimento de esforço nas simulações proposta, validando um maior número de caminhos da árvore de cenários e identificando variações e dificuldades em diferentes níveis de abstração para os diferentes critérios de cobertura de testes.

## REFERÊNCIAS

AUTOMATED TEST AND RETEST (ATRT). Produced by Innovative Defense Technologies. Disponível em: <<http://idtus.com/products/atrt-test-manager/>>. Acessado em 25/01/2014.

CARROLL, J.; ROSSON, M.; CHIN, G.; KOENEMANN, J. Requirements Development in Scenario-Based Design. **IEEE Transactions on Software Engineering**, v.24, n.12, Dec. 1998.

CARROLL, J. Five reasons for scenario-based design. In: Thirty-Second Annual Hawaii International Conference on System Sciences. **Proceedings... IEEE Computer Society**, n.3, v.3, p.3051, 1999.

CASTILLOS, K. C.; BOTELLA, J. Scenario Based Test Generation using Test Designer. In: INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, 4., 2011, Berling, Germany. **Proceedings... Curran Associates Inc.**, 2011. v.1, p.79-88.

EL-FAR, K.; WHITTAKER, J. Model-based Software Testing. **Proceedings... Encyclopedia of Software Engineering**, 2001.

ELGASS, P.; KRCCMAR, H.; & OBERWEIB, A. Business Process Modeling. From Informal to Formal Process Representations. **Proceedings... 3rd European Conference on Information Systems**, 1995.

GRAHAM, D.; FEWSTER, M. **Experiences of Test Automation: Case Studies of Software Test Automation**. Addison-Wesley Professional, 2012. 672p.

HSIA, P.; SAMUEL, J.; GAO, J.; KUNG, D.; TOYOSHIMA, Y.; CHEN, C. Formal Approach to Scenario Analysis. **IEEE Software Magazine**, v.11, n.2, p.33-41, Mar. 1994.

KOUBARAKIS, M.; PLEXOUSAKIS, D. A formal model for business process modeling and design. **Proceedings... Advanced Information Systems Engineering**, p.142-156, 2000.

MANSOUR, N.; TAKKOUSH, H.; NEHME, A. UML-based Regression Testing for OO Software. **Journal of Software Maintenance and Evolution: Research and Practice**, Jul. 2010.

MAIDEN, N.; ROBERTSON, S. Developing Use Cases and Scenarios in the Requirements Process. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 27., 2005, London, United Kingdom. **Proceedings... ACM**, 2005. v.1, p.561-570.

MALIK, Q.; LILIUS, J.; LAIBINIS, L. Model-based Testing Using Scenarios and Event-B Refinements. In: Methods, Models and Tools for Fault Tolerance, p.177-195, 2009.

MUKHERJEE, D.; DHOOLIA, P.; SINHA, S.; REMBERT, A. J.; NANDA, M. G. (2010). From informal process diagrams to formal process models. **Proceedings... Business Process Management**, p.145-161, 2010.

NAJUMUDHEEN, E.; MALL, R.; SAMANTA, D. Test coverage analysis based on an object-oriented program model. **JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE**, 2011, n.23, p.465-493.

NEBUT, C.; FLEUREY, F.; TRAON, Y.; JEZEQUEL, J. Automatic Test Generation: A Use Case Driven Approach. **IEEE Transactions on Software Engineering**, v.32, n.3, Mar. 2006.

POSNER, B. Z. What it takes to be a good project manager. In: *Project Management Journal*, 1987, v.1, p.51-55.

QUALITY ASSURANCE RUN (QARun). Produced by Compuware Technologies. Disponível em: < <http://supportline.microfocus.com/Documentation/books/QARun/Doc/qd491/pdf/cwqugx491.pdf>>. Acessado em 25/01/2014.

REIS, R. Q.; REIS, C. A. L.; NUNES, D. J. Automação no Gerenciamento do Processo de Engenharia de Software. In: *ESCOLA DE INFORMÁTICA NORTE*. Belém. Out. 2002

RUOKONEN, A.; PAJUNEN, L.; SYSTA, T. Scenario-Driven Approach for Business Process Modeling. In: *INTERNATIONAL CONFERENCE ON WEB SERVICES*, 7., 2009, Los Angeles, United States. **Proceedings...** Curran Associates Inc., 2009. v.1, p.123-130.

SHARMA, A.; KUSHWAHA, D. Applying Requirement Based Complexity for the Estimation of Software Development and Testing Effort. In: *ACM SIGSOFT Software Engineering Notes*, New York, United States. **Proceedings...** ACM, v.37, n.1, p.1-11, Jan. 2012.

SHIROLE, M.; KOMMURI, M.; KUMAR, R. Transition Sequence Exploration of UML Activity Diagram using Evolutionary Algorithm. **Proceedings...** 5th India Software Engineering Conference, 2012, p.97-100.

SIVASHANMUGAM, K.; LIN, D.; PALANISAMY, S. Scenario Driven Testing. In: *INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY: NEW GENERATIONS*, 8., 2011, Las Vegas, United States. **Proceedings...** Curran Associates Inc., 2011. v.1, p.299-303.

SRIKANTH, H.; BANERJEE, S. Improving test efficiency through system test prioritization. **The Journal of Systems and Softwares**, Jan. 2011.

WEIDENHAUPT, K.; POHL, K.; JARKE, M.; HAUMER, P. Scenarios in System Development: Current Practice. **IEEE Software Magazine**, Apr. 1998.

## APÊNDICES

## Apêndice A: Simulação Cenário Entrada/Saída de Estoque

O Apêndice A tem como objetivo o detalhamento de todos os passos e iterações da primeira simulação teórica apresentada neste trabalho. Tendo como foco o detalhamento do conteúdo da base de componentes, dos questionários a serem respondidos e finalmente a base de cenários de teste. Ao fim de cada iteração, são apresentados os casos de teste resultantes.

### 1. Base de Componentes herdada da Simulação 1

Componente	Categoria	Cenário de Testes
Formulário	Acesso	Validar Opção de Acesso ao Formulário
Formulário	Acesso	Validar Comportamento de Acesso ao Formulário
Formulário	Obrigatório	Salvar Registro sem Preencher Valor no Campo Obrigatório
Formulário	Obrigatório	Salvar Registro Preenchendo Valor " " no Campo Obrigatório
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "abc"
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10,00"
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10.00"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01/01/2001"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01-01-2001"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01/01/01"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "31/02/01"
Formulário	Data	Checar Validação do Campo Data1 com um Valor Menor que o Campo Data2
Formulário	Data	Checar Validação do Campo Data1 com um Valor Igual ao Campo Data2
Formulário	Data	Checar Validação do Campo Data1 com um Valor Maior que o Campo Data2
Formulário	Máx. Caracteres	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Inteiro "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Decimal "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Caracteres

		Especiais
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Cadeia de Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos
<b>Formulário</b>	Dependências	Validar Preenchimento de Listagem com Dependência
<b>Formulário</b>	Dependências	Validar Comportamento de Lista Dependente ao Alterar o Valor do Campo "Pai"
<b>Formulário</b>	Dependências	Validar Dados Presentes em uma Lista Dependente sem Preenchimento do Campo "Pai"
<b>Formulário</b>	Opção Única	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Única	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Única Seleção de Opção do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Múltipla	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Salvar Registro	Salvar Registro com todos os Campos Obrigatórios Preenchidos
<b>Formulário</b>	Salvar Registro	Salvar Registro Preenchendo Valor Idêntico ao de Outro Registro no Campo Exclusivo
<b>Formulário</b>	Salvar Registro	Validar Integridade dos Valores Preenchidos no Registro

## 2. Atualização da Base de Componentes

<b>Componente</b>	<b>Categoria</b>	<b>Cenário de Testes</b>
<b>Formulário</b>	Acesso	Validar Opção de Acesso ao Formulário
<b>Formulário</b>	Acesso	Validar Comportamento de Acesso ao Formulário
<b>Formulário</b>	Obrigatório	Salvar Registro sem Preencher Valor no Campo Obrigatório
<b>Formulário</b>	Obrigatório	Salvar Registro Preenchendo Valor " " no Campo Obrigatório
<b>Formulário</b>	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "abc"
<b>Formulário</b>	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10,00"
<b>Formulário</b>	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10.00"
<b>Formulário</b>	Data	Checar Validação do Campo Data ao Preencher Valor "01/01/2001"
<b>Formulário</b>	Data	Checar Validação do Campo Data ao Preencher Valor "01-01-2001"
<b>Formulário</b>	Data	Checar Validação do Campo Data ao Preencher Valor

		"01/01/01"
<b>Formulário</b>	Data	Checar Validação do Campo Data ao Preencher Valor "31/02/01"
<b>Formulário</b>	Data	Checar Validação do Campo Data1 com um Valor Menor que o Campo Data2
<b>Formulário</b>	Data	Checar Validação do Campo Data1 com um Valor Igual ao Campo Data2
<b>Formulário</b>	Data	Checar Validação do Campo Data1 com um Valor Maior que o Campo Data2
<b>Formulário</b>	Máx. Caracteres	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida
<b>Formulário</b>	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Inteiro "x" que supere a Quantidade Máxima de Caracteres permitida
<b>Formulário</b>	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Decimal "x" que supere a Quantidade Máxima de Caracteres permitida
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Caracteres Especiais
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Cadeia de Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos
<b>Formulário</b>	Dependências	Validar Preenchimento de Listagem com Dependência
<b>Formulário</b>	Dependências	Validar Comportamento de Lista Dependente ao Alterar o Valor do Campo "Pai"
<b>Formulário</b>	Dependências	Validar Dados Presentes em uma Lista Dependente sem Preenchimento do Campo "Pai"
<b>Formulário</b>	Opção Única	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Única	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Única Seleção de Opção do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Múltipla	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Salvar Registro	Salvar Registro com todos os Campos Obrigatórios Preenchidos
<b>Formulário</b>	Salvar Registro	Salvar Registro Preenchendo Valor Idêntico ao de Outro Registro no Campo Exclusivo
<b>Formulário</b>	Salvar Registro	Validar Integridade dos Valores Preenchidos no Registro
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "x..." do Campo

<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "...x..." do Campo
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "...x" do Campo
<b>Listas</b>	Filtros	Validar Comportamento da Seleção de um Valor para o Campo
<b>Listas</b>	Paginação	Validar Comportamento de Navegação entre Páginas de Registros
<b>Listas</b>	Registros	Validar Acesso à Detalhes do Registro
<b>Listas</b>	Registros	Validar Exclusão de Registros sem Dependências na Base de Dados
<b>Listas</b>	Registros	Validar Exclusão de Registros com Dependências na Base de Dados

### 3. Questionários para Caracterização de Componentes

<b>Questionário (Formulários)</b>	<b>Respostas</b>		
<b>Existem Campos Obrigatórios?</b>	Sim	Não	Quais?
<b>Existem Campos Numéricos?</b>	Sim	Não	Quais?
<b>Existem Campos de Data?</b>	Sim	Não	Quais?
<b>Existem Validações entre 2 Campos de Data Distintos?</b>	Sim	Não	Quais?
<b>Existem Restrições quanto a Quantid. de Caracteres dos Campos?</b>	Sim	Não	Quais?
<b>Existem Campos Alfanuméricos?</b>	Sim	Não	Quais?
<b>Existem Campos que Possuem Dependência entre si?</b>	Sim	Não	Quais?
<b>Existem Campos do Tipo Seleção Única?</b>	Sim	Não	Quais?
<b>Existem Campos do Tipo Seleção Múltipla?</b>	Sim	Não	Quais?

<b>Questionário (Listas)</b>	<b>Respostas</b>		
<b>A Lista possui funções de Pesquisa?</b>	Sim	Não	Quais?
<b>A Lista possui funções de Filtros?</b>	Sim	Não	Quais?
<b>A Lista possui funções de Paginação?</b>	Sim	Não	N/A
<b>A Lista permite acesso aos registros individuais?</b>	Sim	Não	N/A
<b>A Lista permite a exclusão de registros?</b>	Sim	Não	N/A

### 4. Respostas do Questionário – Simulação 2

<b>#</b>	<b>Cenário 1 (Formulário 1) - Respostas</b>	
1	Sim	"Login" / "Senha"
2	Não	N/A
3	Não	N/A
4	Não	N/A
5	Sim	"Login" / "Senha"
6	Sim	"Login" / "Senha"
7	Não	N/A



8	Não	N/A
9	Não	N/A

#	Cenário 1 (Formulário 2) – Respostas	
1	Sim	“Código Item” / “Descrição Item” / “Quantidade”
2	Sim	“Quantidade”
3	Não	N/A
4	Não	N/A
5	Sim	“Código” / “Descrição”
6	Sim	“Código” / “Descrição”
7	Não	N/A
8	Não	N/A
9	Não	N/A

#	Cenário 2 (Formulário 1) – Respostas	
1	Sim	“Login” / “Senha”
2	Não	N/A
3	Não	N/A
4	Não	N/A
5	Sim	“Login” / “Senha”
6	Sim	“Login” / “Senha”
7	Não	N/A
8	Não	N/A
9	Não	N/A

#	Cenário 2 (Lista 1) - Respostas	
1	Sim	“Descrição”
2	Não	N/A
3	Não	N/A
4	Sim	N/A
5	Não	N/A

#	Cenário 2 (Formulário 2) - Respostas	
1	Sim	“Código Item” / “Descrição Item” / “Quantidade”
2	Sim	“Quantidade”
3	Não	N/A
4	Não	N/A
5	Sim	“Código” / “Descrição”
6	Sim	“Código” / “Descrição”
7	Não	N/A

8	Não	N/A
9	Não	N/A

## 5. Artefato Simulação 2 – Caso de Testes

Cenário de Testes (Cenário 1)	Variáveis
Validar Opção de Acesso ao Formulário 1	N/A
Validar Comportamento de Acesso ao Formulário 1	N/A
Salvar Registro sem Preencher Valor no Campo Login	N/A
Salvar Registro Preenchendo Valor " " no Campo Login	N/A
Salvar Registro sem Preencher Valor no Campo Senha	N/A
Salvar Registro Preenchendo Valor " " no Campo Senha	N/A
Checar Validação do Campo Login ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida	X = 16
Checar Validação do Campo Senha ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida	X = 8
Checar Validação do Campo Login ao Preencher um Valor "x" que possua Caracteres Especiais	X = \$bruno
Checar Validação do Campo Login ao Preencher um Valor "x" que possua Espaços	X = br uno
Checar Validação do Campo Login ao Preencher um Valor "x" que possua Cadeia de Espaços	X = br uno
Checar Validação do Campo Senha ao Preencher um Valor "x" que possua Caracteres Especiais	X = \$bruno
Checar Validação do Campo Senha ao Preencher um Valor "x" que possua Espaços	X = br uno
Checar Validação do Campo Senha ao Preencher um Valor "x" que possua Cadeia de Espaços	X = br uno
Validar Opção de Acesso ao Formulário 2	N/A
Validar Comportamento de Acesso ao Formulário 2	N/A
Salvar Registro sem Preencher Valor no Campo Código Item	N/A
Salvar Registro Preenchendo Valor " " no Campo Código Item	N/A
Salvar Registro sem Preencher Valor no Campo Descrição	N/A
Salvar Registro Preenchendo Valor " " no Campo Descrição	N/A
Salvar Registro sem Preencher Valor no Campo Quantidade	N/A
Salvar Registro Preenchendo Valor " " no Campo Quantidade	N/A
Checar Validação do Campo Quantidade ao Preencher Valor "abc"	N/A
Checar Validação do Campo Quantidade ao Preencher Valor "10,00"	N/A
Checar Validação do Campo Quantidade ao Preencher Valor "10.00"	N/A
Checar Validação do Campo Código Item ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida	X = 10
Checar Validação do Campo Descrição ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida	X = 100
Checar Validação do Campo Código Item ao Preencher um Valor "x" que possua Caracteres Especiais	X = #354#
Checar Validação do Campo Código Item ao Preencher um Valor "x" que possua Espaços	X = 154 971
Checar Validação do Campo Código Item ao Preencher um Valor "x"	X = 154

que possua Cadeia de Espaços	971
Checar Validação do Campo Código Item ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos	X = c134G1a
Checar Validação do Campo Descrição ao Preencher um Valor "x" que possua Caracteres Especiais	X = 1-item
Checar Validação do Campo Descrição ao Preencher um Valor "x" que possua Espaços	X = ite m1
Checar Validação do Campo Descrição ao Preencher um Valor "x" que possua Cadeia de Espaços	X = ite m1
Checar Validação do Campo Descrição ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos	X = iTEm1
Salvar Registro com todos os Campos Obrigatórios Preenchidos	N/A
Salvar Registro Preenchendo Valor Idêntico ao de Outro Registro no Campo Exclusivo	N/A
Validar Integridade dos Valores Preenchidos no Registro	N/A

Cenário de Testes (Cenário 2)	Variáveis
Validar Opção de Acesso ao Formulário 1	N/A
Validar Comportamento de Acesso ao Formulário 1	N/A
Salvar Registro sem Preencher Valor no Campo Login	N/A
Salvar Registro Preenchendo Valor " " no Campo Login	N/A
Salvar Registro sem Preencher Valor no Campo Senha	N/A
Salvar Registro Preenchendo Valor " " no Campo Senha	N/A
Checar Validação do Campo Login ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida	X = 16
Checar Validação do Campo Senha ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida	X = 8
Checar Validação do Campo Login ao Preencher um Valor "x" que possua Caracteres Especiais	X = \$bruno
Checar Validação do Campo Login ao Preencher um Valor "x" que possua Espaços	X = br uno
Checar Validação do Campo Login ao Preencher um Valor "x" que possua Cadeia de Espaços	X = br uno
Checar Validação do Campo Senha ao Preencher um Valor "x" que possua Caracteres Especiais	X = \$bruno
Checar Validação do Campo Senha ao Preencher um Valor "x" que possua Espaços	X = br uno
Checar Validação do Campo Senha ao Preencher um Valor "x" que possua Cadeia de Espaços	X = br uno
Validar Pesquisa de Registros utilizando o Valor "x..." do Campo Descrição	X = mat
Validar Pesquisa de Registros utilizando o Valor "...x..." do Campo Descrição	X = ter
Validar Pesquisa de Registros utilizando o Valor "...x" do Campo Descrição	X = rial
Validar Acesso à Detalhes do Registro	N/A
Validar Exclusão de Registros sem Dependências na Base de Dados	N/A
Validar Exclusão de Registros com Dependências na Base de Dados	N/A
Validar Opção de Acesso ao Formulário 2	N/A

<b>Validar Comportamento de Acesso ao Formulário 2</b>	N/A
<b>Salvar Registro sem Preencher Valor no Campo Código Item</b>	N/A
<b>Salvar Registro Preenchendo Valor " " no Campo Código Item</b>	N/A
<b>Salvar Registro sem Preencher Valor no Campo Descrição</b>	N/A
<b>Salvar Registro Preenchendo Valor " " no Campo Descrição</b>	N/A
<b>Salvar Registro sem Preencher Valor no Campo Quantidade</b>	N/A
<b>Salvar Registro Preenchendo Valor " " no Campo Quantidade</b>	N/A
<b>Checar Validação do Campo Quantidade ao Preencher Valor "abc"</b>	N/A
<b>Checar Validação do Campo Quantidade ao Preencher Valor "10,00"</b>	N/A
<b>Checar Validação do Campo Quantidade ao Preencher Valor "10.00"</b>	N/A
<b>Checar Validação do Campo Código Item ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida</b>	X = 10
<b>Checar Validação do Campo Descrição ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida</b>	X = 100
<b>Checar Validação do Campo Código Item ao Preencher um Valor "x" que possua Caracteres Especiais</b>	X = #354#
<b>Checar Validação do Campo Código Item ao Preencher um Valor "x" que possua Espaços</b>	X = 154 971
<b>Checar Validação do Campo Código Item ao Preencher um Valor "x" que possua Cadeia de Espaços</b>	X = 154 971
<b>Checar Validação do Campo Código Item ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos</b>	X = c134G1a
<b>Checar Validação do Campo Descrição ao Preencher um Valor "x" que possua Caracteres Especiais</b>	X = 1-item
<b>Checar Validação do Campo Descrição ao Preencher um Valor "x" que possua Espaços</b>	X = ite m1
<b>Checar Validação do Campo Descrição ao Preencher um Valor "x" que possua Cadeia de Espaços</b>	X = ite m1
<b>Checar Validação do Campo Descrição ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos</b>	X = iTEm1
<b>Salvar Registro com todos os Campos Obrigatórios Preenchidos</b>	N/A
<b>Salvar Registro Preenchendo Valor Idêntico ao de Outro Registro no Campo Exclusivo</b>	N/A
<b>Validar Integridade dos Valores Preenchidos no Registro</b>	N/A

## Apêndice B: Simulação de um Módulo de Gestão de Custos

O Apêndice B tem como objetivo o detalhamento de todos os passos e iterações da primeira simulação prática apresentada neste trabalho. Tendo como foco o detalhamento do conteúdo da base de componentes, dos questionários a serem respondidos e finalmente a base de cenários de teste. Ao fim de cada iteração, são apresentados os casos de teste resultantes.

### 1. Base de Componentes herdada da Simulação 1

Componente	Categoria	Cenário de Testes
Formulário	Acesso	Validar Opção de Acesso ao Formulário
Formulário	Acesso	Validar Comportamento de Acesso ao Formulário
Formulário	Obrigatório	Salvar Registro sem Preencher Valor no Campo Obrigatório
Formulário	Obrigatório	Salvar Registro Preenchendo Valor " " no Campo Obrigatório
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "abc"
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10,00"
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10.00"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01/01/2001"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01-01-2001"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01/01/01"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "31/02/01"
Formulário	Data	Checar Validação do Campo Data1 com um Valor Menor que o Campo Data2
Formulário	Data	Checar Validação do Campo Data1 com um Valor Igual ao Campo Data2
Formulário	Data	Checar Validação do Campo Data1 com um Valor Maior que o Campo Data2
Formulário	Máx. Caracteres	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Inteiro "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Decimal "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Caracteres

		Especiais
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Cadeia de Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos
<b>Formulário</b>	Dependências	Validar Preenchimento de Listagem com Dependência
<b>Formulário</b>	Dependências	Validar Comportamento de Lista Dependente ao Alterar o Valor do Campo "Pai"
<b>Formulário</b>	Dependências	Validar Dados Presentes em uma Lista Dependente sem Preenchimento do Campo "Pai"
<b>Formulário</b>	Opção Única	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Única	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Única Seleção de Opção do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Múltipla	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Salvar Registro	Salvar Registro com todos os Campos Obrigatórios Preenchidos
<b>Formulário</b>	Salvar Registro	Salvar Registro Preenchendo Valor Idêntico ao de Outro Registro no Campo Exclusivo
<b>Formulário</b>	Salvar Registro	Validar Integridade dos Valores Preenchidos no Registro
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "x..." do Campo
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "...x..." do Campo
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "...x" do Campo
<b>Listas</b>	Filtros	Validar Comportamento da Seleção de um Valor para o Campo
<b>Listas</b>	Paginação	Validar Comportamento de Navegação entre Páginas de Registros
<b>Listas</b>	Registros	Validar Acesso à Detalhes do Registro
<b>Listas</b>	Registros	Validar Exclusão de Registros sem Dependências na Base de Dados
<b>Listas</b>	Registros	Validar Exclusão de Registros com Dependências na Base de Dados

## 2. Atualização Número 1 da Base de Componentes

<b>Componente</b>	<b>Categoria</b>	<b>Cenário de Testes</b>
<b>Navegação</b>	Pré-Condições	Validar se o sistema permite a mudança de página com o campo "X" não preenchido

<b>Navegação</b>	Pré-Condições	Validar se o sistema permite a mudança de página com o campo "X" preenchido
<b>Navegação</b>	Pré-Condições	Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"
<b>Navegação</b>	Interface	Validar se a página carregada trouxe corretamente os dados preenchidos na página anterior
<b>Navegação</b>	Interface	Verificar se é possível clicar no link ou botão, caso a condição "X" não estiver atendida
<b>Alertas</b>	Realce de Falha	Validar se os registros com falhas são apresentados na cor "X"
<b>Alertas</b>	Realce de Falha	Validar se os registros sem falhas são apresentados na cor "X"
<b>Alertas</b>	Realce de Tipo	Validar se os registros do tipo "X" são apresentados na cor "Y"

### 3. Atualização Número 2 da Base de Componentes

<b>Componente</b>	<b>Categoria</b>	<b>Cenário de Testes</b>
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na interface
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na base de dados
<b>Regras de Negócio</b>	Atualização	Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite a inclusão do valor "Y" para o campo "X" ao salvar.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite que o registro seja atualizado, caso "X" seja verdadeiro.
<b>Regras de Negócio</b>	Inserção	Validar se o campo "X" recebeu corretamente o valor "Y" na interface
<b>Regras de Negócio</b>	Inserção	Validar se o campo "X" recebeu corretamente o valor "Y" na base de dados
<b>Regras de Negócio</b>	Inserção	Validar se o sistema preencheu o valor "Y" para o campo "X" após inclusão
<b>Regras de Negócio</b>	Inserção	Validar se o sistema permite a inclusão do valor "Y" para o campo "X" ao incluir.
<b>Regras de Negócio</b>	Inserção	Validar se o sistema permite que o registro seja incluído, caso "X" seja verdadeiro.

### 4. Questionários para Caracterização de Componentes

<b>Questionário (Formulários)</b>	<b>Respostas</b>		
<b>Existem Campos Obrigatórios?</b>	Sim	Não	Quais?
<b>Existem Campos Numéricos?</b>	Sim	Não	Quais?
<b>Existem Campos de Data?</b>	Sim	Não	Quais?
<b>Existem Validações entre 2 Campos de Data Distintos?</b>	Sim	Não	Quais?
<b>Existem Restrições quanto a Quantid. de Caracteres dos Campos?</b>	Sim	Não	Quais?
<b>Existem Campos Alfanuméricos?</b>	Sim	Não	Quais?
<b>Existem Campos que Possuem Dependência entre si?</b>	Sim	Não	Quais?
<b>Existem Campos do Tipo Seleção Única?</b>	Sim	Não	Quais?

<b>Existem Campos do Tipo Seleção Múltipla?</b>	Sim	Não	Quais?
---	-----	-----	--------

<b>Questionário (Listas)</b>	<b>Respostas</b>		
<b>A Lista possui funções de Pesquisa?</b>	Sim	Não	Quais?
<b>A Lista possui funções de Filtros?</b>	Sim	Não	Quais?
<b>A Lista possui funções de Paginação?</b>	Sim	Não	N/A
<b>A Lista permite acesso aos registros individuais?</b>	Sim	Não	N/A
<b>A Lista permite a exclusão de registros?</b>	Sim	Não	N/A

#### 5. Atualização dos Questionários de Caracterização dos Componentes

<b>Questionário (Navegação)</b>	<b>Respostas</b>		
<b>A passagem de página possui destino definido? Como ativa?</b>	Sim	Não	Quais?
<b>A mudança de página exige o preenchimento de algum campo?</b>	Sim	Não	Quais?
<b>A nova página carregada busca algum dado da página anterior?</b>	Sim	Não	Quais?
<b>O artefato de navegação possui condições para ser habilitado?</b>	Sim	Não	Quais?

<b>Questionário (Alertas)</b>	<b>Respostas</b>		
<b>Existem campos que ativam alertas na falta de preenchimento?</b>	Sim	Não	Quais?
<b>Existem mensagens apresentadas junto aos alertas?</b>	Sim	Não	Quais?
<b>Existem alertas para representação por categorias do registro?</b>	Sim	Não	Quais?

<b>Questionário (Inserção)</b>	<b>Respostas</b>		
<b>Existem valores que serão inseridos neste novo registro?</b>	Sim	Não	Quais?
<b>Os novos valores da inclusão são conhecidos?</b>	Sim	Não	Quais?
<b>Existem valores parametrizados automaticamente no sistema?</b>	Sim	Não	Quais?
<b>Existem restrições de valores para as inclusões serem feitas?</b>	Sim	Não	Quais?
<b>Existem condições do sistema que impeçam a inclusão?</b>	Sim	Não	Quais?

<b>Questionário (Atualização)</b>	<b>Respostas</b>		
<b>Existem valores do registro que precisam ser alterados?</b>	Sim	Não	Sim
<b>Os novos valores da alteração são conhecidos?</b>	Sim	Não	Sim
<b>Existem valores parametrizados automaticamente no sistema?</b>	Sim	Não	Sim
<b>Existem restrições de valores para as alterações serem feitas?</b>	Sim	Não	Sim
<b>Existem condições do sistema que impeçam a alteração?</b>	Sim	Não	Sim

#### 5. Respostas do Questionário – Simulação 2



#	Respostas (Formulário 1)	
1	Sim	“Período Inicial”
2	Não	N/A
3	Sim	“Período Inicial”
4	Sim	“Período Inicial” e “Período Final”
5	Não	N/A
6	Não	N/A
7	Não	N/A
8	Não	N/A
9	Não	N/A

#	Respostas (Formulário 2)	
1	Sim	“Período Final”
2	Não	N/A
3	Sim	“Período Final”
4	Sim	“Período Final” e “Período Inicial”
5	Não	N/A
6	Não	N/A
7	Não	N/A
8	Não	N/A
9	Não	N/A

#	Respostas (Formulário 3)	
1	Sim	“Tipo de Projeção”
2	Não	N/A
3	Não	N/A
4	Não	N/A
5	Não	N/A
6	Não	N/A
7	Não	N/A
8	Sim	“Tipo de Projeção”
9	Não	N/A

#	Respostas (Lista 1)	
1	Não	N/A
2	Não	N/A
3	Sim	N/A
4	Não	N/A
5	Não	N/A

#	Respostas (Navegação 1)	
1	Sim	“Tela de Projeção” e “botão Continuar”

2	Sim	“Período Inicial” / “Período Final” / “Tipo de Projeção”
3	Sim	“Período Inicial” / “Período Final”
4	Não	N/A

#	Respostas (Lista 2)	
1	Sim	N/A
2	Sim	N/A
3	Sim	N/A
4	Não	N/A
5	Não	N/A

#	Respostas (Alerta 1)	
1	Não	N/A
2	Não	N/A
3	Sim	“Custo Real” e “Preto” / “Custo Projetado” e “Azul”

#	Respostas (Lista 3)	
1	Sim	N/A
2	Sim	N/A
3	Sim	N/A
4	Sim	N/A
5	Não	N/A

#	Respostas (Formulário 4)	
1	Sim	“Valor do Registro”
2	Sim	“Valor do Registro”
3	Não	N/A
4	Não	N/A
5	Não	N/A
6	Não	N/A
7	Não	N/A
8	Não	N/A
9	Não	N/A

#	Respostas (Formulário 5)	
1	Sim	“Data do Registro”
2	Não	N/A
3	Sim	“Data do Registro”
4	Não	N/A
5	Não	N/A
6	Não	N/A
7	Não	N/A
8	Não	N/A

9	Não	N/A
---	-----	-----

#	Respostas (Navegação 2)	
1	Sim	“Tela de Projeção” e “botão Converter para Valor Real”
2	Não	N/A
3	Não	N/A
4	Sim	“Valor do Registro” e “Data do Registro” devem estar preenchidos

#	Respostas (Atualização 1)	
1	Sim	“Data do Registro” / “Valor do Registro”
2	Sim	“01/01/2014” / “100,00”
3	Sim	“Tipo do Registro” e “Custo Real”
4	Não	N/A
5	Sim	“Custo Projetado seja convertido em Real” e “Tipo do Registro igual a Custo Real”

## 6. Artefato Simulação 2 – Caso de Testes

Escopo de Teste: Consultar um registro projetado e torna-lo um custo real

Cenário Proposto: O usuário deve acessar uma projeção existente, consultar entre os registros projetados, aquele que deseja converter e tornar tal dado um custo real.

Cenário de Testes (Cenário Proposto)	Variáveis
Validar Opção de Acesso ao Formulário	N/A
Validar Comportamento de Acesso ao Formulário	N/A
Checar Validação do Campo “X” ao Preencher Valor "01/01/2001"	X= Período Inicial
Checar Validação do Campo “X” ao Preencher Valor "01-01-2001"	X= Período Inicial
Checar Validação do Campo “X” ao Preencher Valor "01/01/01"	X= Período Inicial
Checar Validação do Campo “X” ao Preencher Valor "31/02/01"	X= Período Inicial
Checar Validação do Campo “Y” ao Preencher Valor "01/01/2001"	Y = Período Final
Checar Validação do Campo “Y” ao Preencher Valor "01-01-2001"	Y = Período Final
Checar Validação do Campo “Y” ao Preencher Valor "01/01/01"	Y = Período Final
Checar Validação do Campo “Y” ao Preencher Valor "31/02/01"	Y = Período Final
Checar Validação do Campo “X” com um Valor Menor que o Campo “Y”	X= Período Inicial Y = Período Final
Checar Validação do Campo “X” com um Valor Igual ao Campo “Y”	X= Período Inicial Y = Período Final
Checar Validação do Campo “X” com um Valor Maior que o Campo “Y”	X= Período Inicial Y = Período Final
Checar Possibilidade de Múltipla Seleção de Opções do Campo “Z”	Z = Tipo de Projeção

Validar Comportamento de Seleção da Opção "N" do Campo "Z"	Z = Tipo de Projeção
Validar Comportamento de Navegação entre Páginas de Registros	N/A
Validar se o sistema permite a mudança de página com o campo "X" não preenchido	X= Período Inicial
Validar se o sistema permite a mudança de página com o campo "X" preenchido	X= Período Inicial
Validar se o sistema permite a mudança de página com o campo "Y" não preenchido	Y = Período Final
Validar se o sistema permite a mudança de página com o campo "Y" preenchido	Y = Período Final
Validar se o sistema permite a mudança de página com o campo "Z" não preenchido	Z = Tipo de Projeção
Validar se o sistema permite a mudança de página com o campo "Z" preenchido	Z = Tipo de Projeção
Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"	X = Tela de Projeção Y = Continuar
Validar se a página carregada trouxe corretamente os dados preenchidos na página anterior	N/A
Validar Pesquisa de Registros utilizando o Valor "x..." do Campo	N/A
Validar Pesquisa de Registros utilizando o Valor "...x..." do Campo	N/A
Validar Pesquisa de Registros utilizando o Valor "...x" do Campo	N/A
Validar Comportamento da Seleção de um Valor para o Campo	N/A
Validar Comportamento de Navegação entre Páginas de Registros	N/A
Validar se os registros do tipo "X" são apresentados na cor "Y"	X = Custo Real Y = Preto
Validar se os registros do tipo "X" são apresentados na cor "Y"	X = Custo Projetado Y = Azul
Validar Acesso à Detalhes do Registro	N/A
Salvar Registro sem Preencher Valor no Campo Obrigatório "X"	X = Valor do Registro
Salvar Registro Preenchendo Valor " " no Campo Obrigatório "X"	X = Valor do Registro
Salvar Registro sem Preencher Valor no Campo Obrigatório "Y"	Y = Data do Registro
Salvar Registro Preenchendo Valor " " no Campo Obrigatório "Y"	Y = Data do Registro
Checar Validação do Campo Numérico "X" ao Preencher Valor "abc"	X = Valor do Registro
Checar Validação do Campo Numérico "X" ao Preencher Valor "10,00"	X = Valor do Registro
Checar Validação do Campo Numérico "X" ao Preencher Valor "10.00"	X = Valor do Registro
Checar Validação do Campo Data "Y" ao Preencher Valor "01/01/2001"	Y = Data do Registro
Checar Validação do Campo Data "Y" ao Preencher Valor "01-01-2001"	Y = Data do Registro
Checar Validação do Campo Data "Y" ao Preencher Valor	Y = Data do

"01/01/01"	Registro
Checar Validação do Campo Data "Y" ao Preencher Valor "31/02/01"	Y = Data do Registro
Validar se o sistema permite a mudança de página com o campo "X" preenchido	X = Valor do Registro
Validar se o sistema permite a mudança de página com o campo "Y" preenchido	Y = Data do Registro
Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"	X = Tela de Projeção Y = Converter para Valor Real
Verificar se é possível clicar no link ou botão, caso a condição "X" não estiver atendida	X = Valor e Data devem estar preenchidos
Validar se o campo "X" recebeu corretamente o valor "Y" na interface	X = Data do Registro Y = 01/09/2013
Validar se o campo "X" recebeu corretamente o valor "Y" na base de dados	X = Valor do Registro Y = 100,00
Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento.	X = Tipo do Registro Y = Custo Real
Validar se o sistema permite que "Y", caso "X" seja verdadeiro.	X = Tipo do Registro igual a Custo Real Y = Custo Projetado seja convertido em Real

## Apêndice C: Simulação de Integração de Sistemas para Varejo

O Apêndice C tem como objetivo o detalhamento de todos os passos e iterações da segunda simulação prática apresentada neste trabalho. Tendo como foco o detalhamento do conteúdo da base de componentes, dos questionários a serem respondidos e finalmente a base de cenários de teste. Ao fim de cada iteração, são apresentados os casos de teste resultantes.

### 1. Base de Componentes herdada da Simulação 2

Componente	Categoria	Cenário de Testes
Formulário	Acesso	Validar Opção de Acesso ao Formulário
Formulário	Acesso	Validar Comportamento de Acesso ao Formulário
Formulário	Obrigatório	Salvar Registro sem Preencher Valor no Campo Obrigatório
Formulário	Obrigatório	Salvar Registro Preenchendo Valor " " no Campo Obrigatório
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "abc"
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10,00"
Formulário	Numérico	Checar Validação do Campo Numérico ao Preencher Valor "10.00"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01/01/2001"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01-01-2001"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "01/01/01"
Formulário	Data	Checar Validação do Campo Data ao Preencher Valor "31/02/01"
Formulário	Data	Checar Validação do Campo Data1 com um Valor Menor que o Campo Data2
Formulário	Data	Checar Validação do Campo Data1 com um Valor Igual ao Campo Data2
Formulário	Data	Checar Validação do Campo Data1 com um Valor Maior que o Campo Data2
Formulário	Máx. Caracteres	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Inteiro "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Máx. Caracteres	Checar Validação do Campo Numérico ao Preencher um Valor Decimal "x" que supere a Quantidade Máxima de Caracteres permitida
Formulário	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Caracteres

		Especiais
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor "x" que possua Cadeia de Espaços
<b>Formulário</b>	Alfanuméricos	Checar Validação do Campo Alfanumérico ao Preencher um Valor Idêntico ao de Outro Registro com Caracteres Maiúsculos
<b>Formulário</b>	Dependências	Validar Preenchimento de Listagem com Dependência
<b>Formulário</b>	Dependências	Validar Comportamento de Lista Dependente ao Alterar o Valor do Campo "Pai"
<b>Formulário</b>	Dependências	Validar Dados Presentes em uma Lista Dependente sem Preenchimento do Campo "Pai"
<b>Formulário</b>	Opção Única	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Única	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Única Seleção de Opção do Campo
<b>Formulário</b>	Opção Múltipla	Checar Possibilidade de Múltipla Seleção de Opções do Campo
<b>Formulário</b>	Opção Múltipla	Validar Comportamento de Seleção da Opção "N" do Campo
<b>Formulário</b>	Salvar Registro	Salvar Registro com todos os Campos Obrigatórios Preenchidos
<b>Formulário</b>	Salvar Registro	Salvar Registro Preenchendo Valor Idêntico ao de Outro Registro no Campo Exclusivo
<b>Formulário</b>	Salvar Registro	Validar Integridade dos Valores Preenchidos no Registro
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "x..." do Campo
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "...x..." do Campo
<b>Listas</b>	Pesquisa	Validar Pesquisa de Registros utilizando o Valor "...x" do Campo
<b>Listas</b>	Filtros	Validar Comportamento da Seleção de um Valor para o Campo
<b>Listas</b>	Paginação	Validar Comportamento de Navegação entre Páginas de Registros
<b>Listas</b>	Registros	Validar Acesso à Detalhes do Registro
<b>Listas</b>	Registros	Validar Exclusão de Registros sem Dependências na Base de Dados
<b>Listas</b>	Registros	Validar Exclusão de Registros com Dependências na Base de Dados
<b>Navegação</b>	Pré-Condições	Validar se o sistema permite a mudança de página com o campo "X" não preenchido
<b>Navegação</b>	Pré-Condições	Validar se o sistema permite a mudança de página com o campo "X" preenchido
<b>Navegação</b>	Pré-Condições	Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"
<b>Navegação</b>	Interface	Validar se a página carregada trouxe corretamente os

		dados preenchidos na página anterior
<b>Navegação</b>	Interface	Verificar se é possível clicar no link ou botão, caso a condição "X" não estiver atendida
<b>Alertas</b>	Realce de Falha	Validar se os registros com falhas são apresentados na cor "X"
<b>Alertas</b>	Realce de Falha	Validar se os registros sem falhas são apresentados na cor "X"
<b>Alertas</b>	Realce de Tipo	Validar se os registros do tipo "X" são apresentados na cor "Y"
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na interface
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na base de dados
<b>Regras de Negócio</b>	Atualização	Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite a inclusão do valor "Y" para o campo "X" ao salvar.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite que o registro seja atualizado, caso "X" seja verdadeiro.
<b>Regras de Negócio</b>	Inserção	Validar se o campo "X" recebeu corretamente o valor "Y" na interface
<b>Regras de Negócio</b>	Inserção	Validar se o campo "X" recebeu corretamente o valor "Y" na base de dados
<b>Regras de Negócio</b>	Inserção	Validar se o sistema preencheu o valor "Y" para o campo "X" após inclusão
<b>Regras de Negócio</b>	Inserção	Validar se o sistema permite a inclusão do valor "Y" para o campo "X" ao incluir.
<b>Regras de Negócio</b>	Inserção	Validar se o sistema permite que o registro seja incluído, caso "X" seja verdadeiro.

## 2. Atualização Número 1 da Base de Componentes

<b>Componente</b>	<b>Categoria</b>	<b>Cenário de Testes</b>
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na interface
<b>Regras de Negócio</b>	Atualização	Validar se o campo "X" recebeu corretamente o valor "Y" na base de dados
<b>Regras de Negócio</b>	Atualização	Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento, se condição "Z" for verdadeira.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite a inclusão do valor "Y" para o campo "X" ao salvar.
<b>Regras de Negócio</b>	Atualização	Validar se o sistema permite que o registro seja atualizado, caso "X" seja verdadeiro.
<b>Regras de Negócio</b>	Replicação	Validar se ambos os registros possuem todos os dados idênticos exceto o campo "X"
<b>Regras de Negócio</b>	Replicação	Validar a existência de "N" registros para cada "X" no registro original

## 4. Questionários para Caracterização de Componentes



Questionário (Formulários)	Respostas		
Existem Campos Obrigatórios?	Sim	Não	Quais?
Existem Campos Numéricos?	Sim	Não	Quais?
Existem Campos de Data?	Sim	Não	Quais?
Existem Validações entre 2 Campos de Data Distintos?	Sim	Não	Quais?
Existem Restrições quanto a Quantid. de Caracteres dos Campos?	Sim	Não	Quais?
Existem Campos Alfanuméricos?	Sim	Não	Quais?
Existem Campos que Possuem Dependência entre si?	Sim	Não	Quais?
Existem Campos do Tipo Seleção Única?	Sim	Não	Quais?
Existem Campos do Tipo Seleção Múltipla?	Sim	Não	Quais?

Questionário (Listas)	Respostas		
A Lista possui funções de Pesquisa?	Sim	Não	Quais?
A Lista possui funções de Filtros?	Sim	Não	Quais?
A Lista possui funções de Paginação?	Sim	Não	N/A
A Lista permite acesso aos registros individuais?	Sim	Não	N/A
A Lista permite a exclusão de registros?	Sim	Não	N/A

Questionário (Navegação)	Respostas		
A passagem de página possui destino definido? Como ativa?	Sim	Não	Quais?
A mudança de página exige o preenchimento de algum campo?	Sim	Não	Quais?
A nova página carregada busca algum dado da página anterior?	Sim	Não	Quais?
O artefato de navegação possui condições para ser habilitado?	Sim	Não	Quais?

Questionário (Alertas)	Respostas		
Existem campos que ativam alertas na falta de preenchimento?	Sim	Não	Quais?
Existem mensagens apresentadas junto aos alertas?	Sim	Não	Quais?
Existem alertas para representação por categorias do registro?	Sim	Não	Quais?

Questionário (Inserção)	Respostas		
Existem valores que serão inseridos neste novo registro?	Sim	Não	Quais?
Os novos valores da inclusão são conhecidos?	Sim	Não	Quais?
Existem valores parametrizados automaticamente no sistema?	Sim	Não	Quais?
Existem restrições de valores para as inclusões serem feitas?	Sim	Não	Quais?
Existem condições do sistema que impeçam a inclusão?	Sim	Não	Quais?

Questionário (Atualização)	Respostas		
Existem valores do registro que precisam ser alterados?	Sim	Não	Sim
Os novos valores da alteração são conhecidos?	Sim	Não	Sim

Existem valores parametrizados automaticamente no sistema?	Sim	Não	Sim
Existem restrições de valores para as alterações serem feitas?	Sim	Não	Sim
Existem condições do sistema que impeçam a alteração?	Sim	Não	Sim

#### 5. Atualização dos Questionários de Caracterização dos Componentes

Questionário (Atualização)	Respostas		
Existem valores do registro que precisam ser alterados?	Sim	Não	Quais?
Os novos valores da alteração são conhecidos?	Sim	Não	Quais?
Existem valores parametrizados automaticamente no sistema?	Sim	Não	Quais?
Existem condições a serem atendidas para a parametrização?	Sim	Não	Quais?
Existem restrições de valores para as alterações serem feitas?	Sim	Não	Quais?
Existem condições do sistema que impeçam a alteração?	Sim	Não	Quais?

Questionário (Replicação)	Respostas		
Existem condições relacionadas ao número de replicações?	Sim	Não	Quais?
Existem números fixos para a quantidade de replicações?	Sim	Não	Quais?
Existem campos que não terão os dados replicados no registro?	Sim	Não	Quais?
Os valores dos campos não replicados são conhecidos?	Sim	Não	Quais?

#### 6. Respostas do Questionário – Simulação 3

#	Respostas (Navegação 1)	
1	Sim	“Importação de Registros” e “botão Iniciar Integração”
2	Não	N/A
3	Não	N/A
4	Não	N/A

#	Respostas (Lista 1)	
1	Não	N/A
2	Sim	“Inconsistências”
3	Não	N/A
4	Não	N/A
5	Sim	N/A

#	Respostas (Inserção 1)	
1	Não	N/A
2	Não	N/A
3	Sim	“Estação” / “Data da Inclusão” / “Descrição 1” / “Descrição 2” / “Perecível?”
4	Não	N/A
5	Não	N/A

#	Respostas (Inserção 2)	
1	Não	N/A
2	Não	N/A
3	Sim	“Estação” / “Data da Inclusão” / “Descrição 1” / “Descrição 2” / “Perecível?”
4	Não	N/A
5	Não	N/A

#	Respostas (Navegação 2)	
1	Sim	“Conversão de Valores” e “Avançar”
2	Não	N/A
3	Sim	Registros Importados
4	Sim	“ao menos 1 registro foi importado”

#	Respostas (Atualização 1)	
1	Não	N/A
2	Sim	“Winter”
3	Sim	“Estação”
4	Sim	“Estação é igual a Outono” / “Estação é igual a Inverno”
5	Não	N/A
6	Não	N/A

#	Respostas (Atualização 2)	
1	Não	N/A
2	Sim	“05/15/2014”
3	Sim	“Data de Inclusão”
4	Sim	“Data de Inclusão igual a 15/05/2014”
5	Não	N/A
6	Não	N/A

#	Respostas (Atualização 3)	
1	Não	N/A
2	Sim	“Descrição 1 – Descrição 2”
3	Sim	“Descrição 1”
4	Não	N/A
5	Não	N/A
6	Não	N/A

#	Respostas (Atualização 4)	
1	Não	N/A
2	Sim	“1” / “0”
3	Sim	“Perecível?”
4	Não	“Perecível é igual a Verdadeiro” / “Perecível é igual a Falso”

5	Não	N/A
6	Não	N/A

#	Respostas (Lista 2)	
1	Não	N/A
2	Sim	"Inconsistências"
3	Não	N/A
4	Não	N/A
5	Sim	N/A

#	Respostas (Replicação 1)	
1	Sim	"Valor de Código de Barras Exclusivo"
2	Não	N/A
3	Sim	"Número do Código de Barras"
4	Sim	N/A

#	Respostas (Alertas 1)	
1	Não	N/A
2	Não	N/A
3	Sim	"Registro Válido" e "Branco" / "Inconsistência" e "Vermelho"

#	Respostas (Navegação 3)	
1	Não	N/A
2	Não	N/A
3	Não	N/A
4	Sim	"ao menos 1 registro em vermelho estar marcado"

#	Respostas (Lista 3)	
1	Não	N/A
2	Não	N/A
3	Não	N/A
4	Não	N/A
5	Sim	N/A

#	Respostas (Navegação 4)	
1	Sim	"Encerramento da Integração" e "Avançar"
2	Não	N/A
3	Sim	Registros Replicados
4	Sim	"ao menos 1 registro foi replicado"

#	Respostas (Lista 4)	
1	Não	N/A
2	Sim	"Inconsistências"

3	Não	N/A
4	Não	N/A
5	Não	N/A

### 7. Artefato Simulação 3 – Caso de Testes

Escopo de Teste: Selecionar alguns registros e convertê-los para o novo formato

Cenário Proposto: Importar as frutas da estação de outono para o novo sistema removendo os registros com código de barra do tipo oito dígitos.

Cenário de Testes (Cenário Proposto)	Variáveis
Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"	X = Importação de Registros Y = Iniciar Integração
Validar Comportamento da Seleção de "X" para o Filtro	X = Inconsistências
Validar Exclusão de Registros sem Dependências na Base de Dados	N/A
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão (na interface)	X = Estação
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão (na interface)	X = Data da Inclusão
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão (na interface)	X = Descrição 1
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão (na interface)	X = Descrição 2
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão (na interface)	X = Perecível?
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão	X = Estação
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão	X = Data da Inclusão
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão	X = Descrição 1
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão	X = Descrição 2
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão	X = Perecível?
Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"	X = Conversão de Valores Y = Avançar
Validar se a página carregada trouxe corretamente os dados preenchidos na página anterior	N/A
Verificar se é possível clicar no link ou botão, caso a condição "X" não estiver atendida	X = ao menos 1 registro foi importado
Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento, se condição "Z" for verdadeira.	X = Estação Y = Winter Z = Estação é igual a Inverno
Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento, se condição "Z" for verdadeira.	X = Estação Y = Winter

	Z = Estação é igual a Outono
Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento, se condição "Z" for verdadeira.	X = Data de Inclusão Y = 05/15/2014 Z = Data de Inclusão igual a 15/05/2014
Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento.	X = Descrição 1 Y = Descrição 1 + Descrição 2
Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento, se condição "Z" for verdadeira.	X = Perecível? Y = 1 Z = Perecível é igual a Verdadeiro
Validar se o sistema preencheu o valor "Y" para o campo "X" após o evento, se condição "Z" for verdadeira.	X = Perecível? Y = 0 Z = Perecível é igual a Falso
Validar Comportamento da Seleção de "X" para o Filtro	X = Inconsistências
Validar Exclusão de Registros sem Dependências na Base de Dados	N/A
Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"	X = Importação de Registros Y = Iniciar Integração
Validar Comportamento da Seleção de "X" para o Filtro	X = Inconsistências
Validar Exclusão de Registros sem Dependências na Base de Dados	N/A
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão (na interface)	X = Estação
Validar se o sistema preencheu o valor correto para o campo "X" após inclusão (na interface)	X = Data da Inclusão
Validar se ambos os registros possuem todos os dados idênticos exceto o campo "X"	X = Número do Código de Barras
Validar a existência de "N" registros para cada "X" no registro original	X = Valor de Código de Barras Exclusivo
Validar se os registros do tipo "X" são apresentados na cor "Y"	X = Registro Válido Y = Branco
Validar se os registros do tipo "X" são apresentados na cor "Y"	X = Inconsistência Y = Vermelho
Verificar se é possível clicar no botão "Remover Registros", caso a condição "X" não for atendida	X = ao menos 1 registro em vermelho estar marcado
Validar Exclusão de Registros sem Dependências na Base de Dados	N/A
Validar a página "X" para onde o usuário é direcionado ao clicar na opção "Y"	X = Encerramento da Integração Y = Avançar
Validar se a página carregada trouxe corretamente os "registros replicados" na página anterior	N/A
Verificar se é possível clicar no link ou botão, caso a condição "X" não estiver atendida	X = ao menos 1 registro foi replicado
Validar Comportamento da Seleção de "X" para o Filtro	X = Inconsistências