UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

PROPOSTA METODOLÓGICA DE AUTOMATIZAÇÃO DE TESTES DE SISTEMA UTILIZANDO SELENIUM WEBDRIVER

Monografia de Conclusão do Curso de

Autor(a)

Jônatas Aimbiré de Oliveira Santos

Orientador(a):

Doutor Vitório Bruno Mazzola

Florianópolis, 11 de Setembro de 2012

Jônatas Aimbiré de Oliveira Santos

Proposta Metodológica de Automatização de Testes de Sistema Utilizando Selenium WebDriver

Trabalho de Conclusão de curso
apresentado como parte das atividades para
obtenção do título de Bacharel do curso de
Sistemas de Informação da Universidade
Federal de Santa Catarina.

Professor Orientador: Doutor Vitório Bruno Mazzola

Florianópolis, 2012

Página intencionalmente em branco.

Esse trabalho é dedicado a toda minha família, em especial aos meus pais, que me deram a educação mais importante que eu poderia receber, e sempre me apoiaram nas decisões que tomei.

Aos professores João Dovicchi e Vitório Mazzola que me guiaram pacientemente durante esta caminhada.

E não menos importante, a todos meus colegas, companheiros, demais professores e os que estiveram presentes comigo nesses oito anos e meio de universidade, meu muito obrigado.

Resumo

O propósito deste trabalho é demonstrar todo o processo de transformação de uma rotina de testes manual em uma aplicação web que são lentos e mais propensos a erros para uma versão automatizada do mesmo processo, esta mais rápida e que permite ser executada sem necessitar de um recurso humano no processo.

Esta modelagem será feita utilizando uma ferramenta de automação em navegadores, o Selenium WebDriver, aliada ao *framework* TestNG para construção dos casos. Os resultados de tal aplicação se mostraram bastante positivos e alinhados ao esperado, já que a também automação deste processo permite que seja possível integrar esta etapa da qualidade de software em um processo de integração contínua, utilizando o apoio de Jenkins CI e Apache Ant..

Esta proposta se demonstrou um grande desafio, porém com resultados tão positivos, a mesma pode se considerar um sucesso na otimização dos testes de interface de usuário para softwares web.

PALAVRAS-CHAVE: Automação; Testes de Sistema; Selenium WebDriver; Web; Jenkins CI; Apache Ant; TestNG;

Abstract

The purpose of this work is to demonstrate all the process on transforming a manual test routine for web applications that are usually slower and more likely to have errors for a automatized version of the same process, this one quicker and allowing to be run without the need of a human resource on the process.

The model followed will be using a browser automation tool,Selenium WebDriver allied to the framework TestNG, for structuring the cases. The results of this implementation showed to be positive and aligned with the expected, since this automation also allows to integrate this step on software quality assurance on a continuous integration process, using for support Jenkins CI and Apache Ant.

This proposal demonstrate to be a great challenge, although with such positive results, the same can be considered a success in the optimization of user interface tests for web softwares.

KEYWORDS: Automation; System Tests; Selenium WebDriver; Web; Jenkins CI; Apache Ant; TestNG;

Página intencionalmente em branco.

LISTA DE FIGURAS

LISTA DE SIGLAS

CRUD – Create, read, update, delete.

FLV – Frutas, legumes e verduras.

XML - Extensible Markup Language

JRE – Java Runtime Environment

JDK – Java Development Kit

JAR – Java ARchive

AJAX - Asynchronous JavaScript and XML

DOM – Document Object Model

HTML - HyperText Markup Language

PATENTES

SUMÁRIO

1 Introdução

Teste de sistema automatizados é um processo importante na etapa de verificação de qualidade dos softwares, sendo considerado o cenário principal de testes a ser coberto [20]. Diversas vezes são feitos por um processo manual onde casos de teste são escritos e necessitam que se faça uma conferência dos resultados esperados, seguindo um roteiro e descrevendo os passos que devem ser utilizados.

> "Primeiramente, o termo automação de testes precisa ser esclarecido, pois existem diversos tipos de tecnologia na automação de testes. Categorias de automação de testes incluem: 1) gerência de teste, 2) teste unitário, 3) teste de geração de dados, 4) teste de performance, e 5) teste funcional/de sistema/de regressão (i.e. ferramentas de execução de testes). O mais popular destes são tecnologias de automação em "execução de testes" (também conhecida como Capturar/Reproduzir ou Gravar/Reproduzir) com tecnologia ek teste de performance em segundo lugar." (WISSINK e AMARO, 2006).

O processo atual de testes de sistema na empresa PariPassu [5] é semelhante a diversas companhias e segue a metodologia de conferência utilizando um roteiro de casos escrito, sendo estes verificados por um processo manual. O problema deste tipo de abordagem se deve ao fato de que essa forma de realizar os casos de teste é algo abre a possibilidade para erros humanos. No cenário atual, a atualização constante de produtos web é algo comum, o caso do produto Rastreador 3G [6] que servirá de modelo para a aplicação deste trabalho, segue esta metodologia, com versões novas sendo publicadas ao menos uma vez por semana.

A utilização do processo manual ocupa um colaborador que poderia estar realizando tarefas e testes mais humanos. Isto afeta diretamente o desempenho no que se diz a tempo de duração, já que a conferência manual dos casos de teste, entre outros fatores como a própria velocidade de execução, necessita da disponibilidade do colaborador, isto afeta o tempo total de aplicação do roteiro, que pode variar conforme seja o conhecimento do funcionário no produto em questão.

Em casos de correção urgente, agilidade serve como palavra-chave, já que em *softwares* web, geralmente os produtos não podem ficar fora de disponibilidade e a velocidade da correção é crucial para assimilar uma melhor confiabilidade ao produto. Nesse cenário se deve evitar ao máximo esta situação, com o *software* em questão fora do ar, ou com defeitos sem correção.

A proposta deste trabalho, leva em consideração que automatizar esse processo deve trazer maior garantia e velocidade, causando em uma diminuição do tempo de espera nos casos citados acima. No mercado já está disponível uma ferramenta *open-source,* os desenvolvedores principais estão ativos por diversos canais de comunicação com seus usuários. Além de contar com o grande apoio da comunidade, esta ferramenta criada com o objetivo de simular o processo de um ser humano utilizando um navegador web é conhecida por Selenium WebDriver [1].

É possível utilizar desta ferramenta para incluir uma rotina de testes automatizados no caminho de uma integração contínua, permitindo uma menor interação humana, bem como maior tranquilidade aos colaboradores e sócios da empresa no que se diz respeito a funcionalidade do produto que é carro chefe. Para apoio a ferramenta utilizada será o Jenkins CI [4] um projeto que saiu baseado e continua muito semelhante a versão corporativa Hudson CI [26], hoje da empresa americana Oracle [27].

Esta opção não só livra a necessidade de um colaborador verificar manualmente o processo de todo o sistema, evitando erros humanos ou ações de má-fé na interação do processo de roteiro de testes, bem como alia a rapidez dos testes que permitirem um *redeploy* do produto mais rapidamente. Isto diminui o tempo de espera da correção ir ao ar, no servidor final.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver e implantar um sistema para automatizar os testes de sistema e funções críticas de uma aplicação *web*, utilizando para ilustração o produto Rastreador 3G [6] da empresa PariPassu [5], otimizando o processo da equipe de qualidade durante a etapa de aprovação das novas versões dos aplicativos web da companhia.

1.1.2 Objetivos Específicos

a) Automatizar uma aplicação web, substituindo o roteiro manual de testes de sistema, permitindo maior velocidade e confiabilidade na verificação.

b) Capturar métricas de tempo no final de cada execução, auxiliando a gerência na

avaliação do desempenho do processo de automatização.

c) Possibilitar que os desenvolvedores também possam realizar testes de sistema em seus ambientes de desenvolvimento, melhorando a qualidade das *releases* antes de passar pelo setor de qualidade.

d) Aplicar o projeto de automatização em um progresso de integração contínua.

1.2 Justificativa

A área de testes em qualidade de software necessita constantemente de recursos humanos, pois existem diversas metodologias para execução de testes. Uma das áreas é a de testes de sistema, estes cuja finalidade é verificar a funcionalidade básica do próprio, provando que o mesmo esteja em condições de uso e sem maiores problemas.

A abordagem atual nessa situação é a aplicação de roteiros descritos em diversas maneiras de documentação, sem um paradigma definido para o mesmo. Essa aplicação da metodologia se torna uma pratica maçante e não pouco frequente, erros humanos são encontrados na abordagem dos mesmos, diversas vezes causando na aprovação de um *software* que não se encontra nas ideais condições de uso.

No ambiente de software web, o impacto do lançamento de uma versão com defeitos básicos afeta diretamente a credibilidade do fornecedor do serviço de modo que qualquer tipo de manutenção corretiva em diversos casos causa na interrupção do serviço. Se o mesmo for hospedado em um servidor que provê outras aplicações, todas serão afetadas por um erro de aplicação de roteiro.

Outro ponto que merece ser levado em consideração é de que um roteiro manual acaba causando num alto custo de tempo para aplicação. Todo *software web* de qualidade precisa ser verificado em diversos navegadores, que podem ser fechados em um escopo. No cenário atual, muitas empresas optam por aceitarem de 3 (três) a mais navegadores.

Aplicando um roteiro manual, implica de que o testador precisa executar o mesmo cenário, na quantidade de navegadores que a provedora da aplicação *web* garante a funcionalidade do sistema, a única maneira nesse cenário de acelerar o processo seria a de passar a responsabilidade para mais colaboradores, que nem sempre é um cenário possível.

A proposta metodológica deste trabalho então seria, de utilizar a opção de automatizar esse processo, que tem como seu principal objetivo acelerar todo o processo, que desta

maneira com um serviço de cliente-servidor bem configurado permitiria a execução simultânea paralelamente em diversos navegadores, bem como a velocidade de execução superior a de uma pessoa lendo e executando os casos de teste.

Isso evitará também o mal uso de recursos humanos, que poderão ser colocados em funções mais humanas, levando em consideração após o tempo utilizado para codificar a automação dos testes de sistema, permitindo os mesmos a executarem outros tipos de metodologias de teste nas quais precisa-se mais ativamente de uma interação humana como i.e. Testes de Usabilidade.

E com a automação seria enfim, permitido integrar continuamente esses testes, um cenário aonde cada alteração no repositório do sistema os testes poderão ser disparados automaticamente por regras de iniciação ou até mesmo manualmente, dando aos desenvolvedores uma resposta mais enérgica as recentes alterações do sistema.

1.3 Estrutura do trabalho

Este trabalho está divido em cinco capítulos: 1. Introdução; 2. Referencial Teórico; 3. Proposta de uma abordagem de testes; 4. Aplicando a abordagem; 5. Ambiente e Experimentos; 6. Conclusões e Trabalhos Futuros;

O primeiro capítulo serve apenas para descrever as motivações do trabalho e apresentar a justificativa da proposta apresentada nos seguintes. No segundo capítulo, descreve-se um referencial teórico, explicando pontos importantes do cenário de testes e integração contínua.

A proposta de abordagem bem como sua aplicação, são dividas nos capítulos três e quatro. O primeiro descreve a situação mais comum e o cenário atual que foi utilizado como base para a aplicação da proposta, no segundo explica-se com detalhes como foi realizada o desenvolvimento da ferramenta e sua integração com ambientes de *Continuous Integration.*

Por fim, no capítulo cinco se avalia a aplicação e implantação da proposta, discorrendo apenas no último capítulo sobre as vantagens, desvantagens e limitações da proposta bem como a sugestão de trabalhos futuros para o mesmo.

2 Referencial Teórico

Neste capítulo, o objetivo é esclarecer conceitos de testes de software, explicando as diversas técnicas bem como os tipos de testes que estas disponibilizam e suas abordagens e níveis de abstração. Servirá também para embasar o conceito de integração contínua, ativamente utilizado na proposta de automação do trabalho.

2.1 Testes de software

Uma das áreas mais importantes da informática, comandada por um equipe de qualidade, separada a de desenvolvimento em diversos cenários, tem como principal meta capturar, registrar e documentar defeitos e falhas no processo de desenvolvimento e manutenção de software.

Um paradigma muito importante de ser verificado é de que defeitos e falhas nem sempre são causados por falta de qualidade da parte do código de desenvolvimento de uma ferramenta, condições externas como falta de capacidade de hardware para execução da aplicação ou até mesmo problemas como falhas de segurança, usabilidade e escalabilidade não necessariamente estão ligados ao código gerado.

Problemas em *softwares* são comuns e fazem parte de um ciclo de vida que pode ser descrito desde o momento em que o desenvolvedor comete um erro, no qual é apresentado na aplicação como um defeito. Os defeitos por si só não são considerados necessariamente falhas, mas devem ser levados com total seriedade já que prejudicam a imagem da ferramenta.

Atualmente o cenário de testes de software procura evitar e utilizar suas equipes de qualidade para buscar defeitos em *softwares* em tempo de execução, tendo como objetivo uma prevenção de que os problemas sejam capturados o quanto antes, reduzindo drasticamente o tempo de desenvolvimento, já que considera-se tempo gasto em correção e manutenção de aplicações como desenvolvimento por si só, como também reduzir consideravelmente o custo da aplicação.

Diversas técnicas e tipos de testes podem ser aplicados, todos eles variam conforme o processo estrutural de desenvolvimento de software, que pode aplicar algo como CMMI ou MPS.BR, ou até mesmo aplicações ágeis como SCRUMM.

2.2 Técnicas de testes de software

Toda e qualquer técnica de testes de *software* tem por fim um objetivo claro definido, buscar melhorar qualidade do mesmo tentando encontrar falhas no mesmo. No entanto diferentes técnicas podem ser utilizadas para abordar tal cenário, a utilizada na proposta se encontra como uma técnica chamada de regressão.

Existe uma diferenciação também quanto a dinâmica dos testes, podendo ser classificados mais grosseiramente apenas como testes estáticos ou dinâmicos. No primeiro caso, revisões de código, guias de uso e inspeções são sim, consideradas uma forma da área de qualidade de executar testes. Usualmente esse tipo de teste não é aplicado em qualidade de *software*, no entanto a importância do mesmo anda lado a lado com a de executarem os testes dinâmicos.

Os testes dinâmicos no entanto são todos aqueles feitos no *software* em tempo de execução. Não é necessário que o mesmo já esteja concluído, dando espaço para tipos de teste em partes específicas do sistema, essa é uma abordagem muito bem utilizada, porém nem sempre da maneira correta.

Na hora de classificar todas as técnicas de software, podemos dividir em dois grupos distintos, primeiramente falando sobre os testes de caixa, que abrem a possibilidade para três cenários específicos:

- Testes de caixa-branca: São técnicas utilizadas para executar testes na estrutura interna de um software. Nessa técnica podem ser aplicados os tipos de teste de unidade, integração e sistema. Porém a metodologia mais utilizada é a de testes de unidade. O objetivo principal desta técnica é de encontrar defeitos no fluxo de dados buscando códigos de baixa qualidade, buscando verificar cada linha no código e verificando qual deveria ser a saída da mesma.
- Testes de caixa-preta: Considerado de certa forma o oposto dos testes de caixa-branca, nesse cenário não se avalia a estrutura interna, ou seja o código da aplicação, tendo inclusive total desconhecimento sobre a linguagem do mesmo ou como foi implementado. Existem diversos maneiras de se executar essas técnica de testes, sendo as mais disseminadas os testes exploratórios, testes de tabela de decisão e casos de uso. Cenários como os de testes de integração e sistema, utilizados na técnica de caixa-branca, também podem ser utilizados

aqui.

Toda a verificação feita nesse nível, descreve ao testador apenas a o que deve ser esperado do sistema em tempo de execução, os casos de teste devem descrever apenas pontos de aceitação no esquema de entrada e saída [Figura 1]

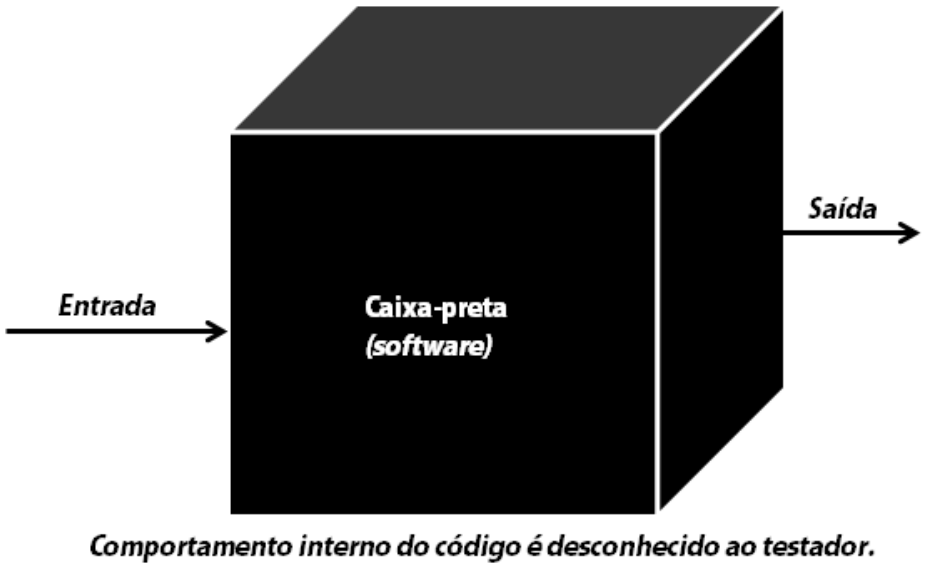

Figura 1: Fluxo dos testes de caixa-preta

- Testes de caixa-cinza: Essa técnica envolve o pensamento de que o testador tem conhecimento do código interno do *software*, utilizando dessa informação para desenvolver os casos de teste. No entanto a execução é feita em nível de caixa-preta, durante essa fase o testador não necessariamente precisa ter conhecimento do código interno, no entanto a manipulação dos dados do mesmo é desencorajada, já que quebra a ideia dessa técnica, removendo a caixa-preta.

  Essa técnica é bastante interessante já que permite ao testador executar escolhas mais bem informadas já que o mesmo conhece a lógica interna. A ideia da técnica é escrever cenários de teste que sejam mais inteligente mas com informações ainda assim limitadas.

No entanto, existe mais uma técnica de teste de software que precisa ser esclarecida, que é a de testes visuais. Esse cenário inclusive é muito bem aceito em ambientes aonde métodos ágeis são utilizados, já que nessa situação abre maior possibilidade de conversação

entre desenvolvedores e testadores.

O objetivo principal desse tipo de técnica é demonstrar a alguém o problema de um *software* apresentando a informação ao invés de descrever a mesma, nessa metodologia, é necessário, no entanto que todo o processo seja memorizado via gravação. Dentre as vantagens dessa abordagem, pode se apontar a melhoria da qualidade de comunicação entre as equipes de desenvolvimento e qualidade e com toda falha de teste gravada, permite ao desenvolvedor focar mais tempo na causa do problema do que exatamente como o mesmo ocorre.

2.3 Tipos de testes de software

Quando se classifica tipos de teste, diversas possibilidades podem ser definidas, no entanto existe um consenso geral com quatro muito utilizados. Vale ressaltar que os tipos de teste diferem de objetivos de teste, como regressão, usabilidade e outras possibilidades. Tipos de teste são basicamente formas que se pode testar um software, se diferenciam das técnicas pois serem mais voltados a forma de se realizar o teste e não quanto ao ambiente do mesmo.

Podemos discorrer em quatro tipos bem definidos como:

- Testes de Unidade: Também chamados de Testes Unitários, esse método bastante utilizado na base da pirâmide citada por COHN [47], visam isolar os testes em uma porção específica de código. Não seria errado assumir que essa forma de teste é a menor parte e primária possibilidade de verificação.

  Numa visão de desenvolvimento orientado a objetos, essa verificação constantemente se mantém a cada classe ou até mesmo interface. Podendo ser mitigada para testes em simples métodos. Geralmente essa etapa é desenvolvida pela própria equipe que constrói o projeto, evitando o contato com os testadores, no entanto equipes de qualidade podem ser treinadas para executarem os testes de caixa-branca nessas situações.

- Testes de Integração: Essa categoria é focada aonde se testam diferentes módulos da aplicação e sua integração entre outros funcionando como um grupo, difere dos testes de sistema que geralmente são feitos em um *software* já completo. Geralmente são executados após os testes de unidade só que em um escopo maior. Sua conclusão libera a etapa de testes de sistema.

- Testes de Sistema: O foco deste trabalho, essa verificação se baseia em avaliar baseado em requisitos uma completa verificação das funcionalidades e integração do sistema. Constantemente andam alinhados a técnica de caixa-preta e são executados após os testes de integração, caso os próprios tenham sido executados com sucesso.

  Constantemente confundidos com testes de integração, o escopo deste é bastante limitado, também por não ter conhecimento da lógica de desenvolvimento, mas por ter que buscar defeitos no sistema como um todo.

- Testes de Aceitação: Uma área não específica apenas aos *softwares*, são conduzidos no entanto em um ambiente de caixa-preta, geralmente descritos por roteiros que verificam condições corretadas da aplicação, roteiros com diversos casos são descritos para garantir que especificações e partes do contrato da ferramenta estão sendo cumpridos.

  Essa etapa não verifica as funcionalidades do sistema, que já devem estar previamente verificadas. Grande parte do seu objetivo é apenas garantir que o produto está de acordo com as definições propostas anteriormente. Tipicamente verificações visuais ou especificações de uso como tamanho de arquivos permitidos para envio ou recebimento.

2.4 Testes de software em aplicações web

Da mesma forma que uma aplicação desenvolvida voltada dedicadamente para dispositivos, os softwares web também precisam de testes. Apesar de que testes em aplicações baseadas em web dividem os mesmos objetivos de testes em aplicações “tradicionais”. Na maioria dos casos, teorias e métodos de teste tradicionais não podem ser utilizados da forma como são, por conta das peculiaridades e complexidades das aplicações Web. [46].

A geração de defeitos em construções de *software* pode ser minimizada, mas todo sistema está suscetível a erros durante a construção de suas funcionalidades. Boas práticas e processos bem definidos auxiliam mas não impedem problemas eventuais, causados por sobrecarga ou a falta de uma visão externa dos casos.

Para ANDRADE e VIEIRA [39], o problema que um defeito pode propagar futuramente em uma aplicação implica em prejuízos ao usuário e um custo alto de correção

caso seja encontrado tardiamente. Frequentemente seguem um padrão de aceitação de testes utilizando mão de obra e processos manuais, que facilmente podem levar a erros humanos, além de causar um desperdício de tempo já que esta forma de atuar é bem lenta [22].

Andrade e Vieira citam que "De acordo com um estudo comissionado pelo *National Institute of Standards and Technology* em 2002, os custos de um defeito num software doméstico chegou a atingir os 59.6 bilhões de dólares nos EUA. " [39].

Aplicações Web podem ser consideradas sistemas distribuídos, com uma arquitetura se assemelhando a um cliente-servidor ou de múltiplas camadas [46]. Também é necessário levar em consideração a apresentação condizente apenas com o ambiente de aplicações online, no qual os usuários que acessam podem estar utilizando diversos navegadores e sistemas operacionais. Isto, mesmo sendo possível limitar essas possibilidades por forma de contratos ou avisos de que as aplicações não irão funcionar com sua total capacidade caso os requisitos não sejam cumpridos.

É inviável no nível corporativo nos moldes de gestão atual, ocupar um empregado para realizar testes de sistema de uma forma manual. Existem outras vertentes e técnicas como testes de usabilidade, que necessitam verdadeiramente de um estudo que observe o comportamento de uma pessoa utilizando o ambiente, causando um impacto de custo muito alto por conta de um tempo que está sendo desperdiçado. Em um processo de lançamento de software, onde surgem novas versões rapidamente e com clientes cada vez mais exigentes, um modelo de testes que siga o mesmo padrão se faz necessário.

Testes web mais comuns são os de sistema, nos quais são verificadas suas funcionalidades básicas. O seu objetivo principal dos testes de sistema é verificar defeitos que afetem toda a aplicação [46]. Os testes de acessibilidade, garantem que todos os usuários consigam visualizar a página com sucesso e também testes de carga para verificar a quantidade de tráfego que o sistema aceita permitindo prever a quantidade de usuários. Outras áreas de teste também são verificadas, mas estas três se tornam as principais por que afetam diretamente a qualidade no ponto de vista do usuário.

2.4.1 Automação de testes web

Uma alternativa bastante comum utilizada na automação de testes de software web é outra ferramenta do projeto Selenium, chamada de Selenium IDE [40]. Trata-se de um

complemento ligado diretamente ao navegador Mozilla Firefox[1], aonde se é possível realizar a automação seguindo o molde Capturar/Reproduzir. No entanto, as desvantagens dessa alternativa são diversas.

O primeiro problema se deve a falta de flexibilidade, já que os testes apenas serão executados no navegador Mozilla Firefox, que não é o mais utilizado mundialmente. [42, 43, 44][2]. Vale ressaltar também que este complemento, não simula as ações do usuário se comunicando com as funções nativas do navegador, apenas injetando JavaScript, para poder realizar suas ações. Outro problema é não ser possível integrar esses testes com uma ferramenta de integração contínua.

Outras ferramentas corporativas existem no mercado, sendo a mais conhecida o TestComplete [41]. Porém o fato de não ser dedicada exclusivamente ao ambiente web, o alto custo de licença de uso, aliada à necessidade de realizar um treinamento do software e a falta de flexibilidade, torna essa opção desencorajadora. Isto porque existe outra opção considerada, não somente por ser *open-source*, mas pela portabilidade e possibilidade de integrar o código em uma ferramenta de integração contínua. Isto não vai de encontro com os objetivos específicos deste projeto.

2.5 Integração contínua

Essa prática da engenharia de software foi introduzida com a ideia de *extreme programming*, sua tendência é de que durante diversas vezes ao dia os ambientes de desenvolvimento dos desenvolvedores sejam sincronizados permitindo verificações constantes da estabilidade da ferramenta.

Criado para ser combinado com automação de testes de unidade em ambientes que propagam a metodologia de *test-driven development*, a ideia maturou e hoje é utilizado para controle de qualidade em diversas possibilidades, no caso desta proposta em testes de sistema.

Ao código ser sincronizado no repositório central, o ambiente que controla a integração contínua, geralmente uma ferramenta específica para o mesmo, dispara a construção de um ciclo que pode ser melhor visualizado na [Figura 2].

1 http://www.mozilla.org/en-US/firefox/new/

2 Informação foi obtida no período de out. 2012.

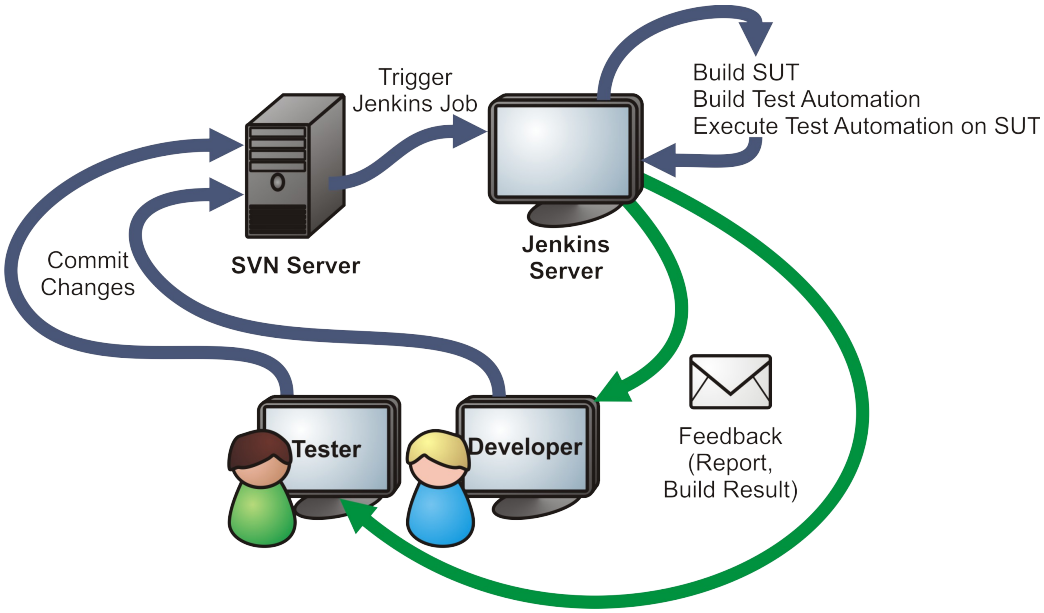

Figura 2: Fluxo da integração contínua.

Esse ambiente verifica tanto as alterações no código da aplicação bem como no próprio código do roteiro de testes automatizados, independentemente do tipo que o mesmo seja, reportando imediatamente aos interessados o resultado da construção de tal tarefa, descrita na imagem como J*ob*. Sendo o *Jenkins* a ferramenta que controla a integração contínua.

As regras que definem as condições que disparam cada construção no entanto não precisam ser tão rígidas, podendo se adequar ao cenário mais pertinente a ferramenta que está sendo desenvolvida, bem como a própria política do ambiente de desenvolvimento.

Essa técnica garante ao ambiente de desenvolvimento e qualidade de *software* um monitoramento mais transparente das aplicações e projetos que são desenvolvidos/mantidos por cada companhia, um processo desses precisa no entanto que testes automatizados sejam escritos para que ele faça sentido, para este trabalho a proposta é repassar os testes manuais de sistema e integrar eles nesse ambiente, atualmente não existente no caso aonde foi aplicado.

3 Proposta de uma abordagem de testes

Este capítulo tem como objetivo descrever e diferenciar os estado atual de testes manuais, levando em consideração seus problemas e limitações e apresentar a proposta do que deseja ser feito integrando com seu resultado final.

Também durante o capítulo, será apresentado as ferramentas e *frameworks* que servirão de apoio para a implementação de tal metodologia.

3.1 Cenário atual

A execução dos testes de sistema atualmente são descritos em um roteiro de texto aonde o testador precisa verificar etapas e funcionalidades do sistema caso a caso em cada navegador que a empresa atesta que seu *software* executará sem problemas no seu funcionamento.

O processo é um fluxo de único sentido que segue um processo definido da seguinte forma: Ao sair uma nova versão do sistema, candidata a publicação, o testador deve pegar o roteiro e passar caso a caso verificando as funcionalidades do sistema. Por ser um *software web,* para cada navegador o mesmo roteiro precisa ser aplicado, no cenário atual isso é realizado três vezes. Após cada execução outro processo que precisa ser repetido é a parte de reportar os resultados em uma ferramenta de gerenciamento dos casos. [Figura 3]

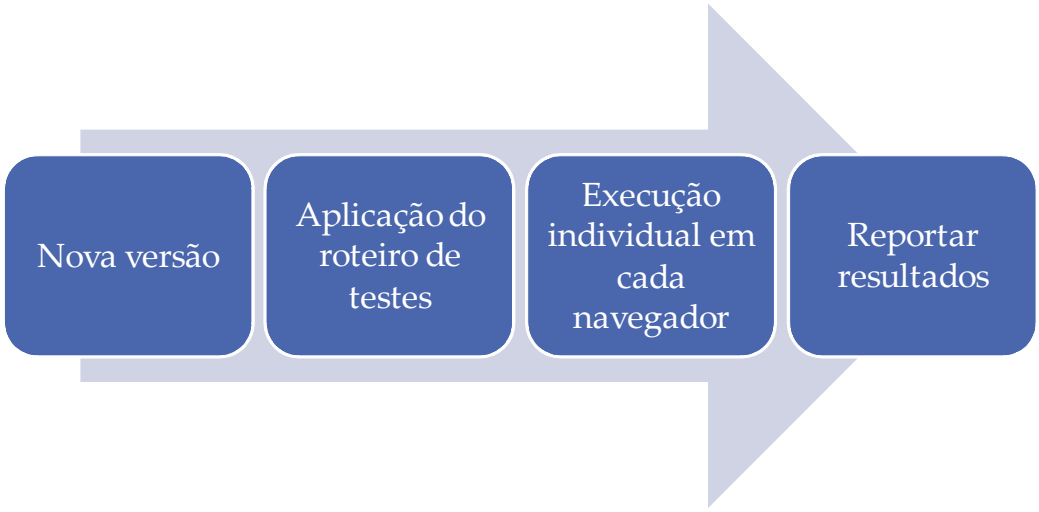

Figura 3: Fluxo de um roteiro manual

Não é preciso de muito para observar que como já comentado, o processo se torna maçante ao funcionário que precisa aplicar tal fluxo. Vale lembrar que se algum defeito for encontrado logo na primeira verificação de navegador e uma versão candidata sair durante o processo de testes, este é interrompido e reiniciado do primeiro passo até a execução. Além de lento, isso demonstra a alta dependência desta forma de processo.

3.2 Cenário proposto

Com uma rotina automatizada de testes, é possível introduzir todo esse fluxo em um ambiente de integração contínua, aonde o processo ganha uma autonomia e independência do setor de qualidade, visto que a tarefa é executada sem necessidade de uma requisição no departamento de testes [Figura 4].

Ao acontecer alguma alteração nos arquivos do repositório os testes de sistema automatizados são chamados. Desenvolvidos para funcionarem independentemente, são verificados todos os navegadores simultaneamente, mesmo sendo um sistema CRUD, variáveis bem definidas permitem que nenhum afete o outro.

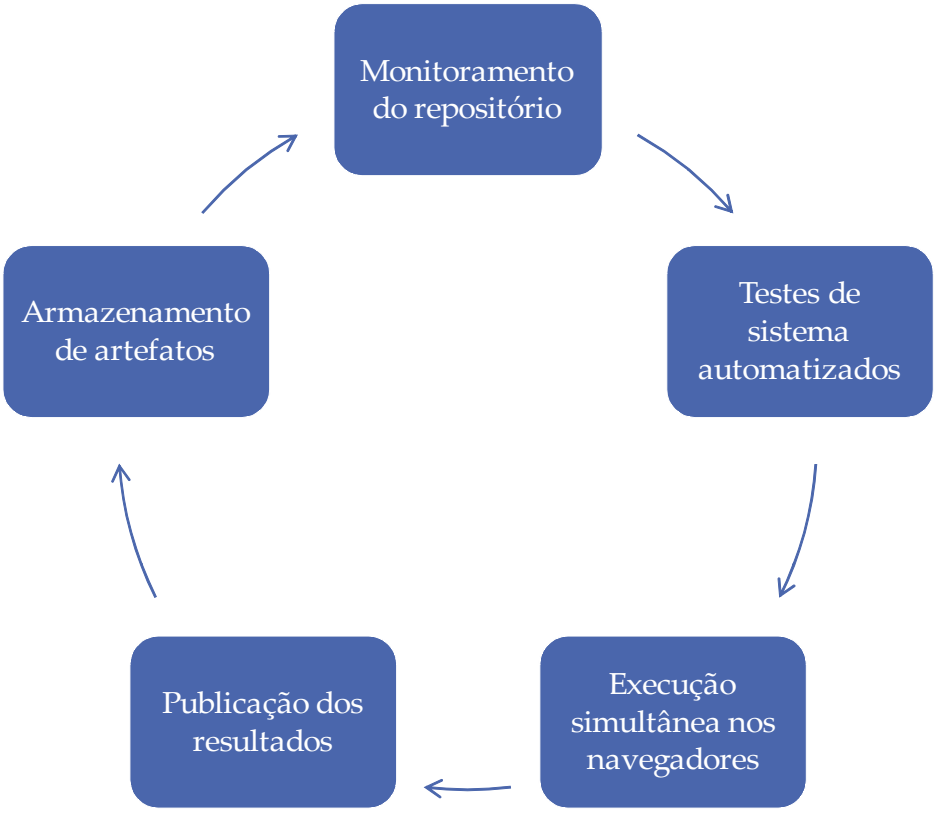

Figura 4: Fluxo da automação integrada continuamente.

A execução apesar de centralizar em um mesmo servidor é distribuída em diversos clientes, em máquinas independentes possibilitando a melhor utilização do *hardware* e não afetando dessa forma a velocidade de execução das mesmas. Ao finalizar os resultados são publicados e automaticamente cadastrados na ferramenta de *bug tracking*, conforme sejam definidos os parâmetros, não existe nenhuma necessidade por parte da equipe, de fazer uma vistoria do relatório.

Outra vantagem desse processo é a possibilidade de armazenar os artefatos com maior clareza, após cada construção, que é datada e diferenciada com um identificador, todas as etapas pertinentes do processo são armazenadas, não apenas como casos em uma ferramenta de *bug tracking*, mas como os próprios relatórios dos testes, arquivos do sistema sendo testado.

Isso dá ao setor de qualidade maior qualidade na informação que será repassada ao desenvolvimento, em diversas situações uma ferramenta está espalhada em diversos ramos de uma árvore no repositório, com a clareza de como eram os arquivos no tempo de execução do teste, o desenvolvedor pode assim, investigar os defeitos baseados na versão específica que foi construída pela tarefa da ferramente de integração contínua. Esse processo pode ser feito no cenário atual, porém se tornaria uma burocracia que poderia facilmente ter problemas de manutenção de histórico.

3.3 Ferramentas para aplicação

3.3.1 Apache Ant

O software Apache Ant [17] (sigla para *"another neat tool")*, é uma ferramenta *open-source* desenvolvida para automatizar o processo de construção de software. Sua primeira versão oficial está datada de 19 de Julho do ano 2000.

Implementada na linguagem Java, e melhor utilizado para projetos que utilizem a mesma tecnologia, seu processo de construção é descrito em um XML. Apesar de semelhante a outra ferramenta com o mesmo propósito, o Apache Ant [17] tem total portabilidade, permitindo que seja executado, assim como códigos Java, em qualquer ambiente de trabalho.

Inicialmente, o software estava integrado ao Apache Tomcat [33]. Seu objetivo era construir o próprio Tomcat e nada mais. No entanto, diversos projetos observaram que o software poderia resolver os problemas existentes com os arquivos da outra ferramenta

utilizada até então. O software se disseminou rapidamente e eventualmente passou a fazer parte de um projeto paralelo.

3.3.1.1 Dependências

Por ser implementado em Java, o Apache Ant [17] precisa de uma versão do Java adequada para ser executado. Mesmo algumas tarefas funcionando apenas com a JRE

3.3.2 TestNG

Pouco é falado sobre quando o TestNG [3] surgiu, mas a primeira data oficial considerada é a de sua versão 1.0, onde consta o ano de 2004 no dia 28 de Abril. O nome surge com a junção da palavra *test*, o qual em inglês significa propriamente teste, e o NG, que é uma sigla para *Next Generation.* Este *framework* é inspirado em JUnit e NUnit, adicionando novas funcionalidades.

No processo de decisão por qual *framework* utilizar, decidiu-se pelo TestNG [3] contra o JUnit por diversos motivos, principalmente por ser possível notar novas versões do anterior seguindo o modelo da ferramenta utilizada neste trabalho.

Dois fatores influenciaram na decisão: o primeiro foi o fato de ser diferente das outras opções pesquisadas. O *framework* permite agrupar testes em grupos e definir dependências, permitindo que os métodos sigam uma ordem pré-definida. Esta dependência se torna necessária, já que a interação precisa seguir uma ordem simulando os estágios de vida dos objetos, pois o produto utilizado para apoiar na construção deste trabalho é um sistema CRUD e requer esta ordem.

O segundo fator que não foi decisivo, mas auxiliou e é característica marcante do *framework,* foram as anotações mais robustas, que permitiram reutilizar métodos, poupando diversas linhas de código, bem como auxiliando o próprio trabalho de manutenção do projeto..

3.3.2.1 Invocando o TestNG

É possível invocar o *framework* de diversas maneiras, desde linha de comando, utilizando o Apache Ant [17] ou diretamente executando a suíte de testes pelo arquivo XML. É possível definir o sequenciamento das classes de testes a serem executadas, inclusive

pacotes completos de classes Java.

3.3.2.2 Anotações

As anotações no TestNG [3] servem para definir qual a função do método Java no fluxo de testes. Muito mais extensas que em outras ferramentas, elas permitem descrever melhor e diminuir a manutenção de código evitando repetir diversas configurações que se fazem necessárias em testes.

3.3.3 Jenkins CI

Jenkins CI [4] é uma ferramenta de integração contínua que tem como principal objetivo monitorar a execução de tarefas. Ela automatiza o processo de construção de projetos Java, bem como o processo de testes, continuamente, durante todo o ciclo de vida de um produto. A ferramenta, monitora conforme definição alterações em um repositório de arquivos[3].

A ferramenta em questão surgiu derivado do Hudson CI [26] da empresa Americana Oracle [27]. Após o mesmo ser adquirido, problemas de patente com o nome Hudson CI [26] geraram dúvidas quanto a quem mantinha o projeto [28] Com isso uma votação foi iniciada e o nome foi trocado para Jenkins. [29]. A Oracle garantiu que o Hudson continuaria a ser desenvolvido e tratou o Jenkins como um *fork* ao invés de uma renomeação [32].

Um dos argumentos utilizados por COLLINS e LUCENA Jr., 2012 [21], para a utilização de uma ferramenta de mesmo utilizando a versão corporativa e um repositório em outra tecnologia, se aplicam ao cenário já que a derivação segue fiel ao produto do qual foi baseado.

> "Hudson monitorou as mudanças no Subversion e iniciou o processo de integração contínua. Desenvolvedores podem facilmente executar os testes de aceitação garantindo a funcionalidade do sistema e Testers podem executar manualmente testes exploratórios, de segurança e usabilidade no ambiente de Testes" [21].

O autor, no entanto, optou pela versão derivada, que é no cenário atual largamente utilizada com um apoio grande da comunidade *open-source.* As atualizações se mantêm em curtos espaços de tempo no que se diz da ferramenta, mas extensíveis nos diversos *plugins* que são utilizados, estendendo a funcionalidade do software. No momento da divisão de

3 O projeto foi sincronizado em um servidor CVS, porém outras formas estão disponíveis.

projetos um comentário chamou a atenção e auxiliou na decisão de optar pelo Jenkins CI [4].

> "Dos dois competidores, minha visão é de que o Jenkins CI irá vencer a competição sobre o Hudson e minhas razões são simples: Jenkins CI vive no espírito original do qual originou Hudson" [24].

3.3.4 Selenium WebDriver

Criado em 2004 por Jason Huggins, o nome Selenium veio de uma brincadeira contra o Mercury[4], outra ferramenta de automação. Durante uma troca de e-mails dos criadores, foi dito que se pode curar envenenamento de Mercúrio (*Mercury)*, com Selênio (*Selenium).* A versão 2.0, chamada de Selenium WebDriver [1] veio de uma união com uma outra ferramenta *open-source* de automação de browsers web, baseada em Ruby o *Watir*[5]*,* se tornou Selenium WebDriver.

Selenium que é uma coleção de ferramentas para automação de navegadores em diversas plataformas. Possui também outras ferramentas além das utilizadas neste trabalho. Para este trabalho, optou-se por citar apenas as funcionalidades da versão utilizada no projeto.

O Selenium WebDriver [1] trabalha criando um servidor local alocado a uma porta aleatória e se comunica através dela para instanciar um navegador. Conforme definido via código gerado pelo usuário. Através deste servidor é possível se comunicar com o browser, enviando comandos que tem como fim simular a interação de um usuário no mesmo. A ferramenta é uma evolução de Selenium Remote Control [31], não precisando, no entanto, que seja instanciado um servidor, para que os testes sejam executados.

> "Selenium Remote Control (RC) permite que você escreva testes UI web automatizados em qualquer linguagem sobre qualquer site HTTP utilizando um popular navegador habilitado em JavaScript" [22].

Cada navegador possui uma forma diferente na qual a ferramenta interage em *background,* no entanto, a percepção visual que se tem em tempo de execução é semelhante. O que ocorre durante a automação é uma simulação de como o usuário interage com a aplicação *web,* manipulando os elementos da página, em um caso próximo da situação real.

3.3.4.1 Selenium Grid 2

Selenium Grid2 [2] é segunda versão oficial do Selenium. Ele dispensa que os *drivers*

4 A ferramenta hoje faz parte de um conjunto de ferramentas conhecida como: HP Quality Center.

5 Pronunciado *water.*

tenham que criar sozinhos os servidores, utilizando um *hub* que controla instâncias *nodes* que são os antigos servidores locais. Os que de fato conversam com os navegadores enviam informações que simulem ações de usuário.

No entanto, uma das principais vantagens é permitir que os testes sejam executados em diversas máquinas diferentes, já que os clientes *(nodes)* se registram no *hub* definido e os testes escritos conversam com o último. Isto permite que o trabalho foque em uma conexão central, liberando recursos da máquina de desenvolvimento.

Esse comportamento faz com que testes sejam paralelizados com mais facilidade, possibilitando executar os mesmos em diversos sistemas operacionais e versões dos navegadores, algo muito importante na área de desenvolvimento para web. Isto porque os *nodes* podem ser instanciados em qualquer máquina, desde que apontem corretamente para o endereço do *hub* central.

Todo o processo, desde a criação de um servidor (*hub*) até o registro de um a vários clientes (*nodes*), é feito utilizando apenas o mesmo JAR, configurando parâmetros que definem o papel de cada um, bem como informações sobre tempo máximo de sessão, portas a serem utilizadas, quantidade de navegadores permitidos, como alguns exemplos. Os comandos são enviados através da linha de comando, independentemente do sistema operacional a ser utilizado, que no caso do trabalho atual está sendo implantado no sistema operacional Windows 7.

3.3.4.2 Escolha da ferramenta

Dentre diversas ferramentas disponíveis para testes disponíveis atualmente, optou-se por utilizar o Selenium WebDriver [1] devido as seguintes circunstâncias: a plataforma ser *open-source* e existir um bom material disponível. Os próprios desenvolvedores estão disponíveis para suporte em fóruns, ferramentas de *bug-tracking* e inclusive um canal de dedicado de IRC[6]. Inclusive o Selenium WebDriver está se candidatando a padrões W3C [18], o que facilitaria a comunicação nativa com os navegadores, já que os mesmos devem se adequar a estes valores.

6 *Internet Relay Chat*

4 Aplicando a abordagem

Este capítulo tem como objetivo apresentar todo o desenvolvimento do trabalho, exemplificando com códigos e imagens o processo de implementação dos testes automatizados de sistema.

4.1 Desenvolvendo a solução

Ao desenvolver a solução, recomenda-se que sejam seguidos antes da execução a ordem proposta nos subcapítulos como descritos neste documento. Essa proposta foi observada como a mais ideal, permitindo o melhor desempenho no momento de construir as automações de um sistema proposto.

4.1.1 Criando uma instância do driver

Ao instanciar o driver que apontará para o *Hub* do Selenium Grid [2], é importante ressaltar que o processo é semelhante porém diferente nos diversos navegadores. No exemplo, os três mais populares atualmente estão previstos em um *switch-case* que verifica qual o valor no parâmetro *browserType* do tipo *Enum*. [Apêndice: Util.java].

O método chamado de `getWebDriver()` não retorna nenhum valor, já que a instância será chamada diretamente por todas as classes de teste em uma variável estática. Essa abordagem impede de que sejam executados mais de dois navegadores em uma sessão, mas não interrompe outros testes caso sejam executados por outra chamada via Apache Ant [17].

Ao instanciar, dois parâmetros são esperados: uma *String* definida na classe *Configs* que aponta para o endereço onde se encontra o h*ub* e um objeto de *DesiredCapabilities*, que define que tipo de informação este driver representa, principalmente o navegador e suas opções de iniciação.

Semelhantes às construções do Internet Explorer e Mozilla Firefox, seus *drivers* são mantidos pelos desenvolvedores do Selenium. Já no caso do navegador Google Chrome, que a implementação é feita por uma equipe de desenvolvedores do *Chromium*, algumas definições são diferentes quanto a definição do navegador em tela cheia e bloqueio de janelas *pop-up*.

4.1.2 Preparando o Selenium Grid 2

Na instanciação do driver, um objeto da classe RemoteWebDriver é criado. Este tem apontado o endereço principal do *hub* do Selenium Grid [2] e nada mais é que a nomenclatura dada pelos desenvolvedores para o servidor que cuidará de redirecionar as requisições aos clientes, chamados de *nodes*. Todos os elementos dessa equação precisam ser iniciados separadamente, utilizando um arquivo JAR.

Após mover o arquivo para uma pasta apropriada, é necessário instanciar primeiramente o *hub*, todo o processo de iniciação é feito via linha de comando . É recomendável que para acessos externos, o *firewall* da máquina que irá hospedar o servidor tenha suas permissões definidas, por padrão, a porta que o mesmo utiliza é a 4444. Podendo ser manualmente definida através do parâmetro '-port'. Por padrão basta digitar o comando:

```
'java -jar selenium-server-standalone-2.25.0.jar -role hub'[7]
```

Após criado, é possível verificar utilizando um navegador o status do servidor. Caso o mesmo esteja devidamente iniciado, ao acessar a página do console no browser, será possível ver algumas informações que descrevem a versão atual apenas uma página em branco, que aguarda nós (*nodes*) que se conectem no *hub*, para que ele possa distribuir as requisições para os mesmos.

É necessário que clientes se conectem a este servidor. Por si só, o servidor não inicia nenhum navegador; são os nós (*nodes*) que se responsabilizam por criar instâncias de navegadores para que o Selenium WebDriver [1] consiga se comunicar. O trabalho do *hub* é simplesmente direcionar estas requisições para um cliente correto.

Para instanciar os clientes que irão se conectar ao *hub,* o mesmo arquivo é utilizado e vale ressaltar que diversos parâmetros alteram a forma no qual o nó (*node)* é criado. Por padrão, ele é instanciado na porta 5555, permitindo até 5 (cinco) sessões de navegadores concorrentes, o que impede um número superior a este de navegadores abertos ao mesmo tempo. Tais informações devem ser alinhadas à potência de hardware da máquina onde serão instanciados os nós (*nodes*).

Por padrão, um cliente é iniciado com 11 (onze)[8] navegadores preparados, sendo 5 (cinco) instâncias do Firefox, da Mozilla Corporation, mais 5 (cinco) do Chrome, navegador da empresa Google, e apenas 1 (um) do Internet Explorer da Microsoft. Este último é o único

7 Caso a versão do Grid seja diferente da utilizada no trabalho, o nome do arquivo deve diferir do citado.

8 Caso o sistema operacional seja 64 bits, o dobro é criado por padrão, simulando as versões de 32 também.

dos três que não permite testes concorrentes. Por isso é desnecessário preparar mais instâncias. Por padrão, o comando que inicia um nó (*node*) é:

'`java -jar selenium-server-stand-alone-2.25-0.jar -role node -hub http://localhost:4444/grid/register`'.

Observando o console do *hub* no navegador, é possível ver ícones que servem para representar a quantidade de instâncias criadas, caso alguma delas esteja em uso. A opacidade da imagem é alterada, de forma que seja visível a diferença com as demais.

4.1.3 Preparando o arquivo de construção

Ao ser invocado, por padrão, o arquivo de construção do Apache Ant [17] busca pela chamada de um *target* com o atributo *name* definido como: 'build-all'. No mesmo foram definidos executar três tarefas em sequência, sendo elas *clean*, que apaga os diretórios onde são criados as versões compiladas do código Java, bem como os relatórios de TestNG [3]. Com isso, temos um cenário onde cada execução estará em um ambiente limpo de informações desatualizadas.

O segundo *target* é chamado *build.* O seu objetivo é compilar os arquivos Java criados. E o terceiro *testIntegracao* chama a tarefa do TestNG [3] criando um diretório onde irão ficar os relatórios de resultado final do teste e define-se as propriedades que irão decidir qual o endereço do *hub*, da página de acesso inicial dos testes e em qual navegador será executado os testes.

4.1.3.1 Definição dos parâmetros em desenvolvimento

Durante o desenvolvimento foi utilizada a ferramenta Eclipse IDE [35]. Todas as execuções foram feitas iniciando-se pela execução do arquivo *build.xml* [Apêndice: build.xml]. Esse motivo se baseou no fato que a execução final via Jenkins CI [4] iria utilizar este procedimento, sendo mais viável testar previamente este cenário. Todos os parâmetros podem ser definidos na execução avançada, na aba *Properties*

Existem cinco propriedades que precisam ser definidas: *profile.user, test.suite, webdriver.hub, portal.url, browser.name.* A primeira serve para identificar em qual pacotes do projeto estão localizados os arquivos de teste. A seguinte define qual é o arquivo XML da suite de testes que será executado, levando em consideração que o mesmo se encontra no pacote definido.

Os três parâmetros finais definem onde está localizado o *hub* do Selenium Grid 2 [2], qual o endereço do portal[9] e o navegador no qual serão aplicados os testes.

4.1.4 Criando a suíte de testes para execução

Com o TestNG [3] a criação da suíte de testes é feita em um arquivo XML. Apesar de não ser complicado, existem algumas definições que valem ressaltar para esclarecer os motivos que levaram a essa escolha de modelo. [Apêndice: default.xml]

Para melhor visualização, cada grupo de testes foi definido com *verbose* nível 2, permitindo que seja verificado no console textual em qual situação se encontra a execução dos testes. Cada grupo foi definido conforme características do sistema, como elementos de cadastro ou de funções rastreamento, e suas etapas, inserções, edições e exclusões.

É possível com o TestNG [3] definir em anotações nas classes Java, grupos de execução, permitindo que testes de cada tela específica sejam implementados no mesmo arquivo, auxiliando na manutenção e entendimento do objetivo de cada classe de teste. Estes podem ser inseridos/removidos conforme decisão própria do desenvolvedor para que sejam executados ou não.

Se a configuração *preserve-order* estiver definida como verdadeiro (o que foi considerado necessário, já que os testes são em um sistema CRUD), é fácil identificar porque os três grupos principais são colocados em ordem *insert, edit e delete*. Ao definir que certo grupo será executado, outros métodos com anotações diferentes da citada não serão executados.

4.1.4.1 Criando e encerrando a sessão do navegador.

Para utilizar independentemente qualquer uma das classes futuramente, é necessário que o objeto da classe *WebDriver* seja instanciado anteriormente a quaisquer grupo de testes. Para isso, duas classes únicas são criadas com apenas um método.

A primeira classe é a responsável pela criação do *driver*. Os objetivos da mesma são capturar os parâmetros enviados pela *build* do Apache Ant [17] e inserir os mesmos em atributos estáticos da classe *Configs*, para que possam ser acessados pela classe *Util*, que executa dois métodos; o primeiro executa um método que entre outras funções, cria o driver a ser utilizado durante todo o tempo de execução. Já o segundo, utilizando o *driver* recém

9 Portal de acesso de usuários dos produtos PariPassu [5].

instanciado, acessa o sistema Rastreador 3G [6] e finalmente pode-se iniciar os testes dentro do produto.

A anotação @BeforeTest no método, garante que este mesmo será a primeira função a ser executada no começo de um grupo de classes dentro da demarcação na suíte do TestNG <*test*></*test*>. No caso em questão, a própria anotação recebe um parâmetro alwaysRun = true, que mesmo em caso de falha de testes ou outras configurações anteriores, o método será chamado. [Apêndice: SetUpWebDriver.java]

Para encerrar os testes, a classe TearDownWebDriver apenas chama um método implementado na classe *Util,* que envia um comando ao objeto *driver* para encerrar a sessão, liberando assim o espaço no *node* registrado no Grid.

Semelhante a anotação na classe anterior, esta inclui @AfterTest, que tem como função exatamente o oposto da classe SetUpWebDriver, visando garantir que este método apenas seja executado ao final da marcação <*test*></*test*>. Também inclui o parâmetro "alwaysRun = true" para que, mesmo com falhas de testes anteriores, a sessão do navegador seja encerrada de tato. [Apêndice: TearDownWebDriver.java]

Vale ressaltar que apesar da suíte do TestNG [3] entender propriamente o momento correto das anotações *@BeforeTest* e *@AfterTest*, permitindo que as classes sejam inseridas em qualquer posição para melhor visibilidade e auxílio na manutenção, as mesmas são referenciadas no começo e final do grupo de testes.

4.1.5 Montando a primeira classe teste

4.1.5.1 Criando os métodos

O primeiro teste a ser automatizado foi o do cadastro de origens [TestOrigem.java], que é o primeiro ponto da cadeia produtiva no rastreamento. Outro fator crucial é que esta tela de cadastro possui todas as tecnologias utilizadas no produto Rastreador 3G [6], desconstruindo assim todas as dificuldades no aprendizado da ferramenta.

Antes da execução de qualquer teste seja de inserção, edição ou exclusão, os métodos precisam estar acessando a tela principal do cadastro de origens. Para isso, foi descrito um método que precisaria ser executado antes de qualquer grupo de testes, mas não necessariamente de método. Felizmente a anotação *@BeforeClass* permite exatamente isso, executando antes de qualquer outro código da classe.

Isso se mostrou crucial, pois evita a repetição de diversas linhas de código, auxiliando na manutenção futuramente. Outra anotação inserida é a @AfterClass, que realiza o inverso da citada anteriormente, executando sempre ao final de todos os métodos executados. No capítulo 3.1.4.1, anotações semelhantes são mostradas, mas é importante reparar na diferença entre as duas.

4.1.5.2 Agrupando e ordenando os testes

Apesar de ser uma ferramenta para testes unitários, o TestNG [3] permite criar dependências entre os testes. Para definir que um método é um teste e não uma configuração, a anotação acima do método *@Test* deverá estar explícita. Métodos sem nenhuma anotação não são considerados teste ou configuração e devem ser utilizados dentro dos mesmos para serem executados.

As anotações permitem que parâmetros sejam inseridos, onde se definem os grupos, dependências e tempo máximo que o teste pode levar (auxiliando em testes de performance). As dependências são descritas com `dependsOnMethods`. Se inseridas, garantem que os testes não serão executados antes que o informado seja finalizado com sucesso (é possível realizar a execução mesmo com falha anterior se o parâmetro `alwaysRun`, estiver definido como verdadeiro.

Uma informação que não está clara na documentação do *framework* é de que se em alguma das classes de testes, houver um ou mais métodos que possuem um parâmetro de dependência na anotação de teste ou configuração, estes serão executados primeiramente. Este comportamento vai de desencontro com a definição *preserve-order* declarada na suíte, desconsiderando a totalmente. Mesmo que as classes não estejam como as primeiras descritas, vão passar na frente da lista no que se diz a ordem de execução [Apêndice: default.xml]

4.1.5.3 Manipulando elementos das páginas

Na hora de manipular os testes da página, precisa-se instanciar um objeto da classe *WebElement* [37], que serve diretamente como um elo entre o código e os comandos nativos do navegador utilizado durante o tempo de execução dos testes. É neste objeto que serão executados todos os comandos necessários para simular a ação de um usuário no navegador, o qual conversará nativamente com o mesmo.

Se por algum motivo for recarregada a informação disponível na DOM, é necessário

que o elementos seja repopulado, já que por sua vez a informação anterior não condiz com o elemento originalmente identificado.

Esse elemento que é instanciado, recebe um objeto da classe *By* [37], que por sua vez é gerado com base em alguma forma de identificação do elemento disponível no documento HTML, sendo mais comuns a utilização dos atributos *id* e *name*. Isto permite outras 6 (seis) maneiras de recuperar esta informação.

Uma maneira simples de instanciar um objeto e utilizar o mesmo para simular o processo de um clique centralizado com o ponteiro mouse sobre um elemento qualquer com o atributo *id* definido com 'objetoHtml', pode ser descrita como:

```
WebElement elementoHtml = driver.findElement(By.id(“objetoHtml”));
elementoHtml.click();
```

Da mesma forma, é possível executar diretamente a ação sem necessariamente guardar a informação em uma variável no código Java. Essa maneira permite a garantia de que toda vez que o objeto precisar ser manipulado, o elo de ligação com o mesmo será refeito, evitando o problema de acessar um objeto que não exista mais devido a uma renovação do código fonte da DOM. Apesar de não ser o único modo utilizado, foi o mais amplamente proposto no trabalho, já que o produto a ser automatizado em questão, utilizada intensivamente bibliotecas JavaScript, que apresentam o mesmo desempenho.

4.1.5.4 Esperando elementos AJAX

Uma das grandes dificuldades encontradas durante o desenvolvimento da automação de testes foi devido ao alto uso de AJAX, que no Rastreador 3G [6] é feito utilizando a biblioteca jQuery [36]. Essa abordagem é positiva, já que as requisições assíncronas evitam que a página seja recarregada completamente, diminuindo a quantidade de transferência a ser feita e melhorando a usabilidade da aplicação.

O problema é que esse tipo de requisição recarrega a DOM da página, que é a definição do documento HTML em uma estrutura de árvore. É neste modelo que se permite manipular e acessar os elementos de uma página. Porém, ao recarregar as informações, o elo com o objeto *WebElement* é perdido, propagando a exceção *StaleElementReferenceException* [37].

Esta exceção indica que a referência ao elemento que se tenta acesso está velha e não se apresenta mais na página indicada. Isso ocorre, pois se no meio tempo de capturar um elemento e finalmente executar o comando para simular uma ação, a página estiver no processo de execução de uma requisição AJAX e finalizar, o elemento não faz mais parte da nova DOM, e a referência foi perdida.

Outra exceção muito comum é *NoSuchElementException* [37] que informa que o objeto a ser manipulado não foi encontrado na DOM. A diferença deste para o anterior é que não se trata de uma ligação que foi perdida durante a execução dos testes, e sim que o elemento em questão não foi encontrado. Um dos casos mais comuns para esse tipo de exceção é que a requisição AJAX não foi finalizada a tempo.

O Selenium WebDriver [1] automaticamente aguarda requisições de *PageLoad*, ou seja, não é necessário controlar manualmente as esperas quando está carregando a página. Porém, se a página já tiver sido finalizada, para evitar os problemas, é preciso aguardar os elementos e requisições manualmente, inserindo a informação claramente no código dos testes.

Existem diversas soluções para esperar os elementos. Não é difícil encontrar diversos exemplos que utilizem `Thread.sleep()`, porém essa é uma alternativa considerada a pior possível [38], já que a utilização deste método interrompe a execução desperdiçando tempo e recursos de processamento. Essa alternativa também não garante que o objeto estará disponível após o tempo de espera ser finalizado.

É possível utilizar uma espera implícita, definida logo após a instanciação do *driver*. Isso permite que o objeto se recorde de sempre aguardar um tempo definido se o objeto a ser procurado não for encontrado imediatamente. Porém, acredita-se que esta não é a melhor alternativa, já que se assemelha a alternativa apresentada anteriormente, já que ao não encontrar, o processo simplesmente para por um determinado tempo e também não garante que o objeto vai estar disponível após.

Existe, no entanto uma maneira muito mais agradável que é criar um objeto da classe *WebDriverWait* [37, 38]. Utilizando este objeto é possível aguardar até o exato momento que o elemento esteja disponível. Esta alternativa é melhor pois não desperdiça tempo e, ainda mais, garante que quando o objeto estiver disponível, será utilizado. Se o objeto não for encontrado no tempo especificado, a exceção *NoSuchElementException* [37]é lançada.

Diversas condições de espera já estão definidas na classe *ExpectedConditions* [37],

porém também é possível definir uma própria condição de espera. Durante o projeto foi necessário utilizar desta maneira em apenas um dos testes [TestOrigem.java], já que todos os outros casos já estão contemplados.

4.1.5.5 Artefatos pós-execução

Ao término da execução, o TestNG gera automaticamente relatórios de estado da execução dos testes, tanto em caso de sucesso, apresentando uma lista com métricas temporais e organização de execução, como em caso de falha, apresentando relatórios que auxiliam na compreensão dos problemas e imprime as *stacktraces*, que auxiliam a isolar o problema.

No entanto, mesmo com este apoio, um grande problema durante a conferência dos artefatos de testes é a verificação dos motivos que possam ter acarretado nos problemas. Frequentemente a informação textual pode estar apontando para um erro de uma linha específica do código que não representa a real causa do problema. Um objeto pode não ter sido acessado, pois outro anterior bloqueia a interação do usuário com ele.

Este problema pode ser contornado com uma verificação visual de um funcionário, observando a execução da automação, mas o processo é arcaico e ineficaz, já que uma pequena distração pode significar em tempo desperdiçado. No entanto, é possível utilizar de uma alternativa para apoiar a equipe de qualidade utilizando ao mesmo tempo recursos disponíveis na ferramenta TestNG e Selenium WebDriver.

A alternativa proposta no parágrafo anterior utiliza de *listeners* do TestNG, aliados ao recurso implementado no Selenium WebDriver de realizar capturas de tela de toda a área do navegador, e não dependente da resolução de tela que executa o mesmo. Isto permite que seja verificado o artefato gerado por ser estático com mais cautela, apoiando no possível cadastro de defeitos futuramente.

Para que isso seja possível, é necessário que seja incluindo uma demarcação <listeners></listeners> no arquivo da suíte do TestNG [Apêndice: default.xml], referenciando a classe Java que implementa a solução [Apêndice: ScreenshotListener.java]. Este *listener* verifica se a condição está tratada. Em caso positivo, executa o método antes de encerrar suas funções. É preciso também que na declaração do *driver* ele seja definido com a função *augment* da classe *Augmenter* [37][Apêndice: Util.java]; é com esta definição que é possível realizar a captura de telas.

4.1.6 Implantando a solução na integração contínua

O Jenkins CI [4], permite diversas formas de se realizar a instalação. Para este trabalho, optou-se por descompactar o arquivo WAR[10] do produto em um servidor Tomcat [33]. De qualquer forma, é recomendado que o Jenkins seja a única aplicação em execução no servidor. É possível também utilizar um instalador nativo, disponível para diversos sistemas operacionais[11]. Este instalador cria um servidor próprio onde pode se gerenciar e controlar as *Jobs*.

Com o projeto finalizado no nível de desenvolvimento, a última etapa é criar uma *Job* no Jenkins CI [4] que integre os testes à rotina de construção do produto Rastreador 3G [6]. Esta rotina foi criada durante o desenvolvimento da solução de automatizar os testes de sistema e compreende em executar diariamente uma restauração do banco de dados online para, em seguida, realizar o *deploy* dos produtos, alinhando em um ambiente de teste os dados mais atuais, bem como as versões mais recentes dos produtos.

O último passo dessa construção serão os testes automatizados de sistema. Caso este último retorne uma resposta de sucesso, é possível dar a garantia de que o sistema está pronto para ser utilizado, e pode ser realizado o *deploy* no servidor de produção, para que as correções ou novas funcionalidades sejam utilizadas.

A criação da *Job* não possui nenhuma dificuldade, no processo de configuração foram necessários definir apenas três etapas. A primeira definindo onde está disponível, no repositório de arquivos o projeto de automação, para que o Jenkins possa realizar o *checkout*. Para evitar gastos desnecessários com tempo e recursos, foi selecionado o campo para realizar apenas atualizações, caso contrário. A cada construção, toda a *workspace* é removida antes de um novo *checkout.*

Com isso definido, a segunda etapa é informar ao Jenkins CI [4], quais passos da construção ele deverá executar e monitorar os resultados. No caso, foi inserido esta etapa que instrui o Jenkins a executar uma invocação do arquivo de construção Ant, apontando onde está no *workspace* o arquivo em questão. Vale ressaltar que devem se utilizar os mesmos parâmetros vistos anteriormente no capítulo 3.1.3.1. Estes (os parâmetros) devem estar inseridos no campo 'Propriedades', da etapa de construção nomeada 'Invocar Ant'.

---

10 Web application ARchive

11 Em out. 2012: Windows; Ubuntu/Debian; Red Hat/Fedora/CentOS; Mac OS X; openSUSE; FreeBSD; OpenBSD; Solaris;Open/Indiana; Gentoo;

Resta apenas definir, após o sucesso da outra tarefa, que esta construção se iniciará automaticamente. Mesmo sendo possível executar a mesma manualmente, para atingir o objetivo, esta informação é crucial para que se tenha de fato, a integração contínua de desenvolvimento, encerrando assim a última das três etapas de configuração.

Após a construção ter sido finalizada, um estado final, dentre quatro possíveis, define o resultado da operação, sendo eles:

- *Abort* – Ocorre apenas no momento em que um usuário do Jenkins, cancela manualmente a tarefa durante o tempo de execução. A ferramenta então exibe juntamente com o nome do processo/usuário que iniciou, a identificação do autor da interrupção.
- *Success* – A construção do Ant foi finalizada com sucesso, com todas as suas *targets* finalizadas sem problema. Para a equipe de qualidade está é a informação de que o sistema está em ordem, e todos os testes foram executados com sucesso.
- *Unstable* – A construção do Ant foi finalizada com sucesso, porém algum teste falhou. É possível configurar a *target* do TestNG para não propagar uma falha na construção. Isso auxilia aos colaboradores a visualizar que o problema aconteceu especificamente durante os testes.
- *Fail* – Durante a construção da tarefa no Jenkins, alguma das *targets* apresentou problemas. Isso difere do estado *Unstable,* já que aponta problemas específicos. Foi observador que os casos mais comuns foram problemas de acesso ao repositório de arquivos, ou o simples problema nas targets.

4.2 Resultados esperados

Com esse trabalho se espera conseguir que os colaboradores de qualidade da PariPassu [5] possam ter mais tempo para focar em outros tipos de testes, acelerar os *deploys* do produto Rastreador 3G [6] através da utilização de uma integração contínua com a ferramenta Jenkins CI [4], e também evitar os problemas relacionados a erros humanos no momento de uma certificação via roteiro de testes manual.

Automatizando os testes de sistema, será possível também auxiliar a própria equipe de desenvolvimento a executar os testes, ajudando também na integração das áreas de

programação e qualidade. Se espera que com essa automação, os testes sejam executados em 9 (nove) versões de navegadores diferentes com apenas uma execução de *job* no Jenkins CI [4].

O sucesso de construções irá fornecer métricas de tempo, para dar maior apoio a gerência quanto ao tempo de finalização de testes de sistema da ferramenta, o que auxilia a equipe de suporte a dar uma previsão mais precisa para o cliente, assim que o código estiver corrigido e de em quanto tempo o projeto deverá estar no ar novamente.

5 Ambiente e Experimentos

Este capítulo tem como objetivo discorrer sobre como o processo foi realizado e validar os resultados obtidos ao final da implantação do projeto.

5.1 Implantação e validação

O primeiro passo durante o projeto foi um estudo do produto a ser automatizado, em conjunto com o roteiro atual de testes. Durante esta etapa foi possível verificar que o próprio se encontrava bastante desatualizado, não cobrindo todas as funcionalidades atuais. Para isso foi necessário, juntamente com o analista de negócios, avaliar e documentar quais seriam os resultados esperados e cenários de falha.

Iniciou-se então a primeira estrutura do projeto, utilizando como teste piloto, o cadastro de origens, o primeiro ciclo da cadeia produtiva no rastreamento. Esta primeira estrutura se mostrou falha e inconsistente. Registraram-se as lições aprendidas e optou-se por uma segunda versão, que foi escrita partindo de um modelo em branco.

A segunda tentativa se apoiou com os erros cometidos anteriormente e o foco se manteve levando em consideração todos os testes futuros, baseando as ações com a estrutura do *framework* da aplicação Rastreador 3G. Os testes foram escritos com TestNG e as ações futuras e testes eram realizados localmente. Esse processo se manteve inalterado até o final do desenvolvimento dos testes.

Com todo o projeto completo e sincronizado no repositório, dois dos primeiros objetivos já haviam sido conquistados. O primeiro foi a substituição do roteiro manual de testes, permitindo que os colaboradores da equipe de qualidade executem os testes de sistema do produto, sem precisar passar pelo maçante processo de verificação manual.

Um segundo objetivo também foi contemplado. Com o projeto finalizado, os desenvolvedores puderam executar os testes em suas máquinas locais, passível de um curto treinamento, permitindo verificar alterações e garantindo a integridade do seu trabalho.

Validado dois resultados, restou a coleta de métricas e a adequação para uma integração contínua. A segunda alia duas tecnologias: a utilização do Apache Ant e a reestruturação do *driver*, utilizando uma instância que fizesse uso do servidor Selenium Grid 2, permitindo os testes serem realizados remotamente, em máquinas dedicadas para testes. O

apoio do Ant permite que o projeto seja executado independente de inserir bibliotecas do Selenium WebDriver no ambiente de desenvolvimento.

Com o projeto rodando em uma integração contínua, o último objetivo também é validado, já que a ferramenta Jenkins captura diversas métricas temporais, desde o tempo total de execução da automação, até a separação por tempo de execução de cada teste, possibilitando verificar divergências em tempo de execução.

5.2 Análise dos resultados

Com o projeto de automação é possível verificar que os testes levam aproximadamente 30 (trinta) minutos para verificar todo o produto Rastreador 3G [6]. Este ganho é importante, pois além de significar um tempo menor do que utilizando testes manuais, dá um valor, já que não existiam métricas para avaliar quanto tempo os testes duravam.

Com a automação do produto em uma ferramenta de integração contínua, os colaboradores da equipe de desenvolvimento podem realizar os testes de sistema, antes mesmo de realizar uma sincronização no repositório. Este processo reduz os problemas de retrabalho, evitando que os colaboradores da equipe de qualidade sejam mal utilizados, sendo que a verificação das funcionalidades básicas pode ser feita no seu ambiente de desenvolvimento [21].

No entanto não foi possível realizar automaticamente a execução de todos os testes em apenas uma job no Jenkins CI [4], a solução adotada seria utilizar outras para cada tipo que apontem para *hub*s que tenham os nós corretos, no entanto isso causa diversos problemas como a necessidade de realizar mais do que apenas uma execução de acessos e *checkouts* no repositório aumentando a quantidade de recursos necessários em tempo de execução.

6 Conclusões e Trabalhos Futuros

Durante todo o projeto, houve um aprendizado constante com a utilização das ferramentas e linguagens utilizadas, desconhecidas até então quanto ao seu uso. Este processo, no entanto, pode ter sido um dos fatores que estendeu o tempo de desenvolvimento do projeto, bem como causou transtornos na iniciação, já que, em algumas ocasiões, foi necessário ser recomeçada quase que por inteiro de seu estado inicial por conta de erros na montagem da arquitetura organizacional.

Mesmo existindo outros produtos que permitam a automação de testes em aplicações web além do Selenium WebDriver, o mesmo excedeu as expectativas, se adequando e cumprindo todos os objetivos propostos no início dos trabalhos. Este modelo permitiu que os testes fossem executados com rapidez e segurança e, com o apoio do Selenium Grid 2, permitiu que pudessem ser verificados em diversos ambientes, possibilitando identificar problemas com rapidez, gerando artefatos automaticamente para cadastro de defeitos.

Vale ressaltar que se mostrou muito importante uma maior comunicação da equipe de desenvolvimento com o setor de qualidade, visto que todo o processo de automação pode ser facilitado e menos penoso se os produtos tiverem um código HTML bem escrito segundo padrões claros e bem definidos.

O presente trabalho demonstrou que o processo de automação em aplicações web é uma tendência que deve ser seguida em todas as companhias, juntamente com a evolução de seus produtos, a independência de garantir o funcionamento de seus recursos básicos não pode depender de recursos humanos e um longo processo que desmotiva funcionários da área.

6.1 Aspectos positivos e negativos

A automação dos testes de sistema permitiu, com todos os objetivos cumpridos, que os colaboradores da equipe de qualidade pudessem alterar sua rotina para realizar testes mais humanos, sendo possível verificar a necessidade de sua utilização. Os testes manuais poderiam ser executados por pessoas que não atuam especificamente na área de *Quality Assurance*.

Foi possível verificar também, uma melhora na atuação dos testes de manutenção, já que o *deploy* automático da ferramenta na integração contínua, evitou o transtorno de uma

comunicação desnecessária quanto a criação de arquivos WAR. Isto porque o *deploy* automático permite que o sistema esteja pronto para uso também, com um simples processo de invocar uma *job* no Jenkins CI.

Dentre outras vantagens vale destacar que o tempo para aplicação de um roteiro de testes de sistema foi incrivelmente mais rápido, devido a paralelização da execução em cada navegador separadamente bem como a própria velocidade do objeto de automação se for comparado com a leitura e execução manual.

6.2 Dificuldades e limitações

Entre os aspectos negativos vale esclarecer o fato de que tal processo pode ocasionalmente apresentar falhas que, se não forem devidamente monitoradas, podem causar problemas futuros. O processo, apesar de ser automático, precisa ser auditado por colaboradores com uma periodicidade variável a estabilidade da automação, para que se dê a garantia em um curto prazo.

A automação também apresentou diversas dificuldades no momento de automatizar alguns comportamentos da aplicação *web* que utilizava de propostas dinâmicas, como componentes que são carregados via requisições AJAX, os mesmos precisam ter um tratamento diferenciado e se forem escritos códigos fracos de automação tendem a causar problemas ocasionais.

O *framework* do Selenium WebDriver, por ser mantido por grupos de desenvolvedores diferentes para cada navegador, apresenta também por conta da própria estabilidade dos navegadores em questão, uma estabilidade, comportamento e velocidade diferente entre eles. É possível verificar isso claramente em componentes de *upload* de arquivos ou outros elementos nativos do HTML, não chegando a causar problemas, mas apenas tratamentos diferenciados na automação.

No entanto alguns componentes e funcionalidades não conseguem ser automatizados, parte por própria limitação do *framework* mas também por regras de negócio que não permitem a automação. Mesmo sendo muito robusta, alguns elementos da interface de usuário, precisam de uma verificação manual dos mesmos. Não conseguindo eliminar em sua totalidade a necessidade dos testes manuais de sistema.

6.3 Trabalhos futuros

Com a finalização deste trabalho, abre-se um leque para possíveis trabalhos futuros:

- Melhorar a estrutura lógica dos testes, utilizando mais elementos de orientação a objetos, como, herança, por exemplo, para auxiliar na manutenção preditiva do sistema de automação de testes.
- Utilizar os testes automatizados de sistema para realizar testes de desempenho, simulando a conexão de diversos usuários no sistema e permitindo prever antecipadamente a necessidade de compra de *hardware* ou ampliação na capacidade dos servidores externos.
- Automatizar outros produtos da empresa PariPassu, encerrando os testes manuais de sistema na empresa e permitindo que a equipe de qualidade de software atue com mais ênfase em outras áreas como testes de usabilidade, acessibilidade, carga e performance.
- Remover os atributos estáticos e paralelizar classes testes que não são dependentes entre si, otimizando o tempo de execução dos testes, para acelerar o processo de integração contínua.

Referências

[1] Selenium WebDriver. Disponível em: <http://seleniumhq.org/docs/>. Acesso em: set. 2012.

[2] Grid2. <http://code.google.com/p/selenium/wiki/Grid2>. Acesso em: set. 2012.

[3] TestNG. Disponível em: <http://testng.org/doc/documentation-main.html>. Acesso em: set. 2012.

[4] Jenkins CI. Disponível em: <https://wiki.jenkins-ci.org/display/JENKINS/Use+Jenkins>. Acesso em: set. 2012.

[5] PariPassu. Disponível em: <http://www.paripassuaplicativos.com.br>. Acesso em: set. 2012.

[6] Rastreador 3G. Disponível em: <http://www.paripassuaplicativos.com.br/produtos_servicos.asp?Id=1>. Acesso em: set. 2012.

[7] GLOBALGAP. Disponível em: <http://www.globalgap.org/cms/front_content.php?idcat=9>. Acesso em: set. 2012.

[8] GS1 Brasil. Disponível em: <http://www.gs1br.org/>. Acesso em: set. 2012.

[9] Produce Marketing Association. Disponível em: <http://ww.pma.com>. Acesso em: set. 2012.

[10] SEBRAE. Disponível em: <http://www.sebrae.com.br/>. Acesso em: set. 2012.

[11] Conscious Capitalism Institude. Disponível em: <http://consciouscapitalism.org/institute/>. Acesso em: set. 2012.

[12] Whole Foods Market. Disponível em: <http://www.wholefoodsmarket.com/>. Acesso em: set. 2012.

[13] Trader Joe's. Disponível em: <http://www.traderjoes.com/>. Acesso em: set. 2012.

[14] The Container Store. Disponível em: <http://www.containerstore.com/welcome.htm>. Acesso em: set. 2012.

[15] Patagonia Outdoor Clothing. Disponível em:

<http://www.patagonia.com/international> Acesso em: set. 2012.

[16] OpenQA. Disponível em: <http://openqa.org/>. Acesso em: set. 2012.

[17] Apache Ant. Disponível em: <http://ant.apache.org/manual/index.html>. Acesso em: set. 2012.

[18] WebDriver – W3C Working Draft. Disponível em: <http://www.w3.org/TR/2012/WD-webdriver-20120710/>. Acesso em: set. 2012.

[19] Automated Web Testing with Selenium. Disponível em: http://pluralsight.com/training/Courses/TableOfContents/selenium, Acesso em: set. 2012.

[20] WISSINK, Tom. AMARO, Carlos. Successful Test Automation for Software Maintenance. IEEE International Conference on Software Maintenance, 2006.

[21] COLLINS, Eliane Figueiredo. LUCENA Jr., Vicente Ferreira de. Software Test Automation Practices in Agile Development Environment: An Industry Experience Report, Zurich, Suíça, 2012.

[22] WANG, Fei. DU, Wencai. A Test Automation Framework Based on WEB. Haikou, China. 2012.

[23] WANG, Lingfeng. Issues on Software Testing For Safety-Critical Real-Time Automation Systems. University of Virginia, United States. 2004.

[24] Hudson vs Jenkins CI. Disponível em: <http://tedone.typepad.com/blog/2011/03/hudson-vs-jenkins-ci.html>. Acesso em: out. 2012.

[25] BuscaRastro. Disponível em: <http://www.paripassuaplicativos.com.br/produtos_servicos.asp?Id=2>. Acesso em: out. 2012.

[26] Hudson Continuous Integration. Disponível em: <http://hudson-ci.org/docs/index.html>. Acesso em: out. 2012.

[27] Oracle Submits Proposal to Make Hudson an Eclipse Foundation Project. Disponível em: <http://www.oracle.com/us/corporate/press/393483>. Acesso em: out. 2012.

[28] Who's driving this thing? Disponível em: <http://jenkins-ci.org/content/whos-

driving-thing> Acesso em: out. 2012.

[29] Hudson Process Proposal. Disponível em: <http://hudson-ci.org/docs/process_summary.html>. Acesso em: out. 2012.

[30] HP Closes Landmark Mercury Aquisition. Disponível em: <http://www.hp.com/hpinfo/newsroom/press/2006/061107xa.html>. Acesso em: out. 2012.

[31] Selenium 1 (Selenium RC). Disponível em: <http://seleniumhq.org/docs/05_selenium_rc.html>. Acesso em: out. 2012.

[32] The future of Hudson. Disponível em: <http://java.net/projects/hudson/lists/dev/archive/2011-02/message/0>. Acesso em: out. 2012.

[33] Apache Tomcat 7.0 Documentation. Disponível em: <http://tomcat.apache.org/tomcat-7.0-doc/>. Acesso em: out. 2012.

[34] JOIA, Luiz Antonio. OLIVEIRA, Luiz Cláudio Barbosa De. **Criação e teste de um modelo para avaliação de *websites* de comércio eletrônico**. *RAM, Rev. Adm. Mackenzie*. 2008, vol.9, n.1, pp. 11-36. ISSN 1678-6971.

[35] Eclipse Download. Disponível em: <http://www.eclipse.org/downloads/>. Acesso em: out. 2012.

[36] jQuery Wiki. Disponível em: <http://docs.jquery.com/>. Acesso em: out. 2012.

[37] WebDriver API. Disponível em: <http://selenium.googlecode.com/svn/trunk/docs/api/java/index.html>. Acesso em: out. 2012.

[38] WebDriver: Advanced Usage. Disponível em: <http://seleniumhq.org/docs/04_webdriver_advanced.html>. Acesso em: out. 2012.

[39] ANDRADE, Luidi Villela de Abreu. VIEIRA, Luiz Gustavo Schroeder. Testes em Software Livre. Universidade Federal de Santa Catarina, 2011.

[40] Selenium IDE. Disponível em: <http://seleniumhq.org/docs/02_selenium_ide.html>. Acesso em: out. 2012.

[41] TestComplete. Disponível em: <http://support.smartbear.com/viewarticle/26550/>. Acesso em: out. 2012.

[42] Browser Statistics. Disponível em: <http://www.w3schools.com/browsers/browsers_stats.asp>. Acesso em: out. 2012.

[43] Top 5 Browsers. Disponível em: <http://gs.statcounter.com/#browser-ww-monthly-200807-201206>. Acesso em: out. 2012.

[44] Top 5 Browsers in Brazil. Disponível em: <http://gs.statcounter.com/#browser-BR-monthly-200807-201206>. Acesso em: out. 2012.

[45] BAU, Jason. BURSZTEIN, Elie. GUPTA, Divij, MITCHELL, John. State of the Art: Automated Black-Box Web Application Vulnerability Testing. Stanford University. Maio de 2010.

[46] LUCCA, Giuseppe A. Di. FASOLINO, Anna Rita. Testing Web-based applications: The state of the art and future trends. Information and Software Technology. 2006

[47] COHN, Mike. Succeeding with Agile: Software Development Using Scrum. 2009, ISBN-10: **0321579364.**

APÊNDICE A – Util.java

```
	public static WebDriver driver;
	public static WebDriverWait wait;

	public static void getWebDriver() throws MalformedURLException {
		startTestMsg("WebDriver");
		BrowserType browserType = Configs.browser;

		switch (browserType) {
		case FIREFOX:
			DesiredCapabilities firefoxCapabilities =
DesiredCapabilities.firefox();

	firefoxCapabilities.setCapability(CapabilityType.TAKES_SCREENSHOT,
true);
			driver = new RemoteWebDriver(new
URL(Configs.ENDERECO_HUB), firefoxCapabilities);
			driver.manage().window().maximize();
			break;
		case IE:
			DesiredCapabilities ieCapabilities =
DesiredCapabilities.internetExplorer();

	ieCapabilities.setCapability(CapabilityType.TAKES_SCREENSHOT, true);
			driver = new RemoteWebDriver(new
URL(Configs.ENDERECO_HUB), ieCapabilities);
			driver.manage().window().maximize();
			break;
		case CHROME:
			DesiredCapabilities chromeCapabilities =
DesiredCapabilities.chrome();
			chromeCapabilities.setCapability("chrome.switches",
				Arrays.asList("--start-maximized --disable-
popup-blocking"));

	chromeCapabilities.setCapability(CapabilityType.TAKES_SCREENSHOT,
true);
			driver = new RemoteWebDriver(new
URL(Configs.ENDERECO_HUB), chromeCapabilities);
			break;
		default:
			throw new RuntimeException("Navegador não suportado.");
```

```
        }

        driver = new Augmenter().augment(driver);
        wait = new WebDriverWait(driver, Configs.INT_TIMEOUT);
    }

    public static enum BrowserType {
        FIREFOX, IE, CHROME
    }
```

APÊNDICE B – build.xml

```
<!--?xml version="1.0"?-->
<project name="testng_selenium" basedir="." default="build-all">

      <property name="src.dir" value="automacao" />
      <property name="reports.dir" value="testng_reports" />
      <property name="lib.dir" value="${src.dir}/lib" />
      <property name="profile.dir" value="${src.dir}/perfil/$
{profile.user}" />
      <property name="build.dir" value="${profile.dir}/bin" />

      <taskdef resource="testngtasks" classpath="${lib.dir}/testng-
6.5.2.jar" />

      <path id="master-classpath">
            <fileset dir="${lib.dir}">
                  <include name="*.jar" />
            </fileset>
            <pathelement path="${build.dir}" />
      </path>

      <target name="clean" description="Removes build directory and test
results.">
            <delete dir="${build.dir}" />
            <delete dir="${reports.dir}" />
      </target>

      <target name="build" depends="clean" description="Build Java files.">
            <mkdir dir="${build.dir}" />
            <javac destdir="${build.dir}" debug="true" deprecation="false"
includeantruntime="true" optimize="false" failonerror="true">
                  <src path="${src.dir}" />
                  <classpath refid="master-classpath" />
            </javac>
      </target>

      <target name="testIntegracao" description="Teste de Integracao.">
            <mkdir dir="${reports.dir}" />
            <testng classpathref="master-classpath" outputdir="$
{reports.dir}" classpath="${src.dir}" haltonfailure="false"
delegateCommandSystemProperties="true">
                  <sysproperty key="ant.grid.hub" value="$
{webdriver.hub}" />
                  <sysproperty key="ant.grid.url" value="${portal.url}" />
                  <sysproperty key="ant.grid.browser" value="$
{browser.name}" />
                  <xmlfileset dir="." includes="${profile.dir}/$
{test.suite}.xml" />
            </testng>
      </target>

      <target name="build-all" description="Build and test all, implies
clean." depends="clean,build,testIntegracao" />

</project>
```

APÊNDICE C – default.xml

```
	<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite de Integracao Rastreador - Perfil 'auto'" verbose="1"
parallel="false" preserve-order="true" configfailurepolicy="skip">
	<listeners>
		<listener class-name="util.ScreenshotListener" />
	</listeners>

	<test name="Instanciação do WebDriver - geral" preserve-order="true"
verbose="2">
		<classes>
			<class name="perfil.geral.util.SetUpWebDriver" />
		</classes>
	</test>

	<test name="Inserções de Cadastros" preserve-order="true"
verbose="2">
		<groups>
			<run>
				<include name="insert" />
			</run>
		</groups>
		<classes>
			<class
name="perfil.geral.cadastro.meuCadastro.TestMeuCadastro" />
			[...]
		</classes>
	</test>

	[...]

	<test name="Edições de Cadastros" preserve-order="true" verbose="2">
		<groups>
			<run>
				<include name="edit" />
			</run>
		</groups>
		<classes>
```

```
                  <class
name="perfil.geral.cadastro.meuCadastro.TestMeuCadastro" />
                  [...]
            </classes>
      </test>

      [...]

      <test name="Exclusão de Cadastros" preserve-order="true" verbose="2">
            <groups>
                  <run>
                        <include name="delete" />
                  </run>
            </groups>
            <classes>
                  [...]
                  <class
name="perfil.geral.cadastro.meuCadastro.TestMeuCadastro" />
            </classes>
      </test>

      <test name="Finalização do WebDriver - geral" preserve-order="true"
verbose="2">
            <classes>
                  <class name="perfil.geral.util.TearDownWebDriver" />
            </classes>
      </test>
</suite>
```

APÊNDICE D – SetUpWebDriver.java

```
public class SetUpWebDriver {

    @BeforeTest(alwaysRun = true)
    public void setUpWebDriver(ITestContext context)
          throws InterruptedException, MalformedURLException {

      Properties sysProps = System.getProperties();
      String browser = sysProps.getProperty("ant.grid.browser");
      Configs.URL_BASE = sysProps.getProperty("ant.grid.url");
      Configs.ENDERECO_HUB = sysProps.getProperty("ant.grid.hub");

      if (browser.equals("firefox")) {
          Configs.browser = Util.BrowserType.FIREFOX;
      } else if (browser.equals("ie")) {
          Configs.browser = Util.BrowserType.IE;
      } else if (browser.equals("chrome")) {
          Configs.browser = Util.BrowserType.CHROME;
      }

      Util.getWebDriver();
      Util.getRastreador(Padroes.USUARIO_PADRAO_NOME);
    }

}
```

APÊNDICE E – TearDownWebDriver.java

```
public class TearDownWebDriver {

	@AfterTest(alwaysRun = true)
	public void tearDownWebDriver() throws InterruptedException {
		Util.encerrarSessao();
	}

}
```

APÊNDICE F – ScreenshotListener.java

```
public class ScreenshotListener extends TestListenerAdapter {


	String screensFolder = "testng_reports/exceptions";

	String dataAtual = Util.dateFormat(true);


	@Override

	public void onTestFailure(ITestResult result) {

		new File(screensFolder).mkdirs();

		File srcFile = ((TakesScreenshot)
Util.driver).getScreenshotAs(OutputType.FILE);

		try {

			FileUtils.copyFile(srcFile, new File(screensFolder + "/"
+ result.getMethod().getClass() + "_"

				+ result.getMethod().getMethodName() +
".png"));

		} catch (IOException e) {

			e.printStackTrace();

			System.out.println("Não foi possível gerar a ScreenShot
com o erro");

		}

	}

}
```

APÊNDICE 8 – Código Completo

```
package perfil.geral.cadastro.destino;

import org.openqa.selenium.By;

public class ElementsDestino {

    // Elementos da Tela Principal - Botões
    public static final By NOVO_REGISTRO = By
          .id("startForm:telaElobasepanelcontentlistgiNovoRegistro");

    // Elementos da Tela Principal - Tabela
    public static final String TABLE = "startForm:telaElodataTable:";
    public static final By TABLE_ORDER = By

.id("startForm:telaElodataTable:telaElodataTablecolumn0header:sortDiv");
    public static final String TABLE_VALUE =
":telaElodataTablecolumn0otValue0";

    // Elementos do Formulário
    public static final By FORM_PESSOA_FISICA = By
          .id("startForm:telaEloradioSelect:1");
    public static final By FORM_NOME_EMPRESA = By
          .id("startForm:telaElotpElotabDadosGeraispn1nome");
    public static final By FORM_NOME_RESPONSAVEL = By
          .id("startForm:telaElotpElotabDadosGeraispn1nomeResponsavel");
    public static final By FORM_CPF = By
          .id("startForm:telaElotpElotabDadosGeraispn1cpfResponsavel");
    public static final By FORM_CODIGO = By
          .id("startForm:telaElotpElotabDadosGeraispn1codigoFornecedor");
    public static final By FORM_ENDERECO = By
          .id("startForm:telaElotpElotabDadosGeraispn1logradouro");
    public static final By FORM_UF = By
          .id("startForm:telaElotpElotabDadosGeraispn1uf");
    public static final By FORM_MUNICIPIO = By
          .id("startForm:telaElotpElotabDadosGeraispn1municipio");
    public static final By FORM_CONTATO = By
          .id("startForm:telaElotpElotabDadosGeraispn3contato");
    public static final By FORM_TELEFONE = By
```

```
            .id("startForm:telaElotpElotabDadosGeraispn3telefone");
      public static final By FORM_EMAIL = By
            .id("startForm:telaElotpElotabDadosGeraispn3email");
      public static final By FORM_OUTROS_CONTATOS = By
            .id("startForm:telaElotpElotabDadosGeraispn3outrosContatos");
      public static final By FORM_DESCRICAO = By
            .id("startForm:telaElotpElotabDadosGeraispn3descricao");
      public static final By FORM_SITE = By
            .id("startForm:telaElotpElotabDadosGeraispn3url");
      public static final By FORM_SITUACAO = By
            .id("startForm:telaElotpElotabDadosGeraispn3situacao");

      // Mensagem de validação dos campos obrigatorios
      public static String MSG_VALIDA_CAMPOS_PJ = "Erro" + "\n"
            + "O campo [Nome da empresa] é obrigatório." + "\n"
            + "O campo [Razão social] é obrigatório." + "\n"
            + "O campo [CNPJ] é obrigatório." + "\n"
            + "O campo [Endereço] é obrigatório." + "\n"
            + "O campo [Município] é obrigatório." + "\n"
            + "O campo [Latitude] é obrigatório." + "\n"
            + "O campo [Longitude] é obrigatório." + "\n"
            + "O campo [Situação] é obrigatório.";
      public static String MSG_VALIDA_CAMPOS_PF = "Erro" + "\n"
            + "O campo [Nome da empresa] é obrigatório." + "\n"
            + "O campo [Nome do responsável] é obrigatório." + "\n"
            + "O campo [CPF do responsável] é obrigatório." + "\n"
            + "O campo [Endereço] é obrigatório." + "\n"
            + "O campo [Município] é obrigatório." + "\n"
            + "O campo [Latitude] é obrigatório." + "\n"
            + "O campo [Longitude] é obrigatório." + "\n"
            + "O campo [Situação] é obrigatório.";
}

package perfil.geral.cadastro.destino;


import org.openqa.selenium.WebDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.testng.Assert;

import org.testng.annotations.*;
```

```
import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestDestino {
	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpDestino() {
		driver = Util.driver;
		Util.startTestMsg(TestDestino.class.getSimpleName());
		Util.abreGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.DESTINO);
	}

	@Test(groups = { "insert" })
	public void abreTelaInsertDestino() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
no.NOVO_REGISTRO));
		driver.findElement(ElementsDestino.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertDestino")
	public void insertUniqueDestino() {
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
no.FORM_NOME_EMPRESA));


driver.findElement(ElementsDestino.FORM_NOME_EMPRESA).sendKeys(Padroes.NOME
_DESTINO_CADASTRO);
            Util.scrollUp();
            driver.findElement(ElementsDestino.FORM_PESSOA_FISICA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
no.FORM_CPF));
            driver.findElement(ElementsDestino.FORM_NOME_RESPONSAVEL)
                        .sendKeys("Responsavel " +
Padroes.NOME_DESTINO_CADASTRO);

      driver.findElement(ElementsDestino.FORM_CPF).sendKeys(Padroes.CPF);


driver.findElement(ElementsDestino.FORM_CODIGO).sendKeys("255467987");

      driver.findElement(ElementsDestino.FORM_ENDERECO).sendKeys("Endereco
" + Padroes.NOME_DESTINO_CADASTRO);
            Util.selectOption(ElementsDestino.FORM_UF, "ES");
            Util.waitJsBlock();


Util.wait.until(ExpectedConditions.presenceOfAllElementsLocatedBy(ElementsD
estino.FORM_MUNICIPIO));
            Util.selectOption(ElementsDestino.FORM_MUNICIPIO, "Cariacica");

      driver.findElement(ElementsRastreador.LAT_GRAUS).sendKeys("41");

      driver.findElement(ElementsRastreador.LAT_MINUTOS).sendKeys("28");
```

```
      driver.findElement(ElementsRastreador.LAT_SEGUNDOS).sendKeys("13");

      driver.findElement(ElementsRastreador.LNG_GRAUS).sendKeys("12");

      driver.findElement(ElementsRastreador.LNG_MINUTOS).sendKeys("08");

      driver.findElement(ElementsRastreador.LNG_SEGUNDOS).sendKeys("21");

      driver.findElement(ElementsDestino.FORM_CONTATO).sendKeys("Contato "
+ Padroes.NOME_DESTINO_CADASTRO);

      driver.findElement(ElementsDestino.FORM_TELEFONE).sendKeys("(29)
4566-5412");


driver.findElement(ElementsDestino.FORM_EMAIL).sendKeys("destino01@email.co
m");

      driver.findElement(ElementsDestino.FORM_OUTROS_CONTATOS).sendKeys(
                        "Outros contatos" + Padroes.NOME_DESTINO_CADASTRO);


driver.findElement(ElementsDestino.FORM_DESCRICAO).sendKeys("Descricao " +
Padroes.NOME_DESTINO_CADASTRO);


driver.findElement(ElementsDestino.FORM_SITE).sendKeys("www.meudestino.com.
br");
            Util.selectOption(ElementsDestino.FORM_SITUACAO,
Padroes.ATIVO);
            driver.findElement(ElementsRastreador.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));
```

```
	driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
	}

	@Test(groups = { "edit" })
	public void setUpEditDestino() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
no.NOVO_REGISTRO));
		Util.acionaElemento(ElementsDestino.TABLE,
ElementsDestino.TABLE_VALUE, Padroes.NOME_DESTINO_CADASTRO,
				ElementsRastreador.TABLE_EDITAR);
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEditDestino")
	public void editUniqueDestino() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
no.FORM_NOME_EMPRESA));

driver.findElement(ElementsDestino.FORM_NOME_EMPRESA).sendKeys(Padroes.EDIT
ADO);
		driver.findElement(ElementsDestino.FORM_CODIGO).clear();

driver.findElement(ElementsDestino.FORM_CODIGO).sendKeys("98451621321");

driver.findElement(ElementsDestino.FORM_ENDERECO).sendKeys(Padroes.EDITADO)
;
		Util.selectOption(ElementsDestino.FORM_UF, "MS");
		Util.waitJsBlock();
```

```
            Util.selectOption(ElementsDestino.FORM_MUNICIPIO, "Amambái");
            driver.findElement(ElementsRastreador.LAT_GRAUS).clear();

      driver.findElement(ElementsRastreador.LAT_GRAUS).sendKeys("18");
            driver.findElement(ElementsRastreador.LAT_MINUTOS).clear();

      driver.findElement(ElementsRastreador.LAT_MINUTOS).sendKeys("52");
            driver.findElement(ElementsRastreador.LAT_SEGUNDOS).clear();

      driver.findElement(ElementsRastreador.LAT_SEGUNDOS).sendKeys("87");
            driver.findElement(ElementsRastreador.LNG_GRAUS).clear();

      driver.findElement(ElementsRastreador.LNG_GRAUS).sendKeys("21");
            driver.findElement(ElementsRastreador.LNG_MINUTOS).clear();

      driver.findElement(ElementsRastreador.LNG_MINUTOS).sendKeys("85");
            driver.findElement(ElementsRastreador.LNG_SEGUNDOS).clear();

      driver.findElement(ElementsRastreador.LNG_SEGUNDOS).sendKeys("61");


driver.findElement(ElementsDestino.FORM_CONTATO).sendKeys(Padroes.EDITADO);
            driver.findElement(ElementsDestino.FORM_TELEFONE).clear();

      driver.findElement(ElementsDestino.FORM_TELEFONE).sendKeys("(11)
6584-7845");
            driver.findElement(ElementsDestino.FORM_EMAIL).clear();


driver.findElement(ElementsDestino.FORM_EMAIL).sendKeys("destino01editado@e
mail.com");


driver.findElement(ElementsDestino.FORM_OUTROS_CONTATOS).sendKeys(Padroes.E
DITADO);
```

```
driver.findElement(ElementsDestino.FORM_DESCRICAO).sendKeys(Padroes.EDITADO
);
            driver.findElement(ElementsRastreador.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
no.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueDestino() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
no.NOVO_REGISTRO));
            Util.acionaElemento(ElementsDestino.TABLE,
ElementsDestino.TABLE_VALUE, Padroes.NOME_DESTINO_CADASTRO
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosDestinoPessoaJuridica() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
```

```
no.NOVO_REGISTRO));

            driver.findElement(ElementsDestino.NOVO_REGISTRO).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SALVAR));

            driver.findElement(ElementsRastreador.FORM_SALVAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));

Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsDestino.MSG_VALIDA_CAMPOS_PJ,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();

Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.scrollUp();
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosDestinoPessoaJuridica")
      public void validacaoCamposObrigatoriosDestinoPessoaFisica() {
            driver.findElement(ElementsDestino.FORM_PESSOA_FISICA).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesti
```

```
no.FORM_CPF));

            driver.findElement(ElementsRastreador.FORM_SALVAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));

Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsDestino.MSG_VALIDA_CAMPOS_PF,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
            Util.scrollUp();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.DESTINO);
            Util.endTestMsg(TestDestino.class.getSimpleName());
      }
}

package perfil.geral.cadastro.fornecedor;

import org.openqa.selenium.By;

public class ElementsFornecedor {

    // Elementos da Tela Principal - Botões
    public static final By NOVO_REGISTRO = By
```

```
.id("startForm:telaFornecedorbasepanelcontentlistgiNovoRegistro");

    // Elementos da Tela Principal - Tabela
    public static final String TABLE =
"startForm:telaFornecedordataTable:";
    public static final String TABLE_VALUE =
":telaFornecedordataTablecolumn0otValue0";
    public static final By TABLE_ORDER = By

.id("startForm:telaFornecedordataTable:telaFornecedordataTablecolumn0header
:sortDiv");

    // Elementos do Formulário
    public static final By FORM_NOME = By
          .id("startForm:telaFornecedornomeFornecedor");
    public static final By FORM_TIPO_DESPESA = By
          .id("startForm:telaFornecedorcomboTipoDespesa");
    public static final By FORM_ADICIONAR = By
          .id("startForm:telaFornecedorbotaoAdicionarbt");
    public static final By FORM_SALVAR = By
          .id("startForm:telaFornecedorbasepanelcontentformbtsalvarbt");
    public static final By FORM_LISTAR = By
          .id("startForm:telaFornecedorbasepanelcontentformbtlistarbt");
    public static final By FORM_APAGAR_DESPESA = By
          .id("startForm:telaFornecedordataTableTiposDespesa:0:gidelete");
    public static final By FORM_TIPO_DESPESA_ADICIONADO = By

.id("startForm:telaFornecedordataTableTiposDespesa:0:telaFornecedordataTabl
eTiposDespesacolumn0otValue0");
    public static final By FORM_TIPO_DESPESA_ADICIONADO_MATERIA_PRIMA = By

.id("startForm:telaFornecedordataTableTiposDespesa:1:telaFornecedordataTabl
eTiposDespesacolumn0otValue0");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Nome] é obrigatório.";
}

package perfil.geral.cadastro.fornecedor;
```

```
import org.openqa.selenium.NoSuchElementException;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestFornecedor {
	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpFornecedor() {
		Util.startTestMsg(TestFornecedor.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.FORNECEDOR);
	}

	@Test(groups = { "insert" })
	public void setUpInsertFornecedor() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.NOVO_REGISTRO));
		driver.findElement(ElementsFornecedor.NOVO_REGISTRO).click();
```

```
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"setUpInsertFornecedor")
	public void insertUniqueFornecedor() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.FORM_NOME));

driver.findElement(ElementsFornecedor.FORM_NOME).sendKeys(Padroes.NOME_FORN
ECEDOR);
		Util.selectOption(ElementsFornecedor.FORM_TIPO_DESPESA,
Padroes.DESPESA);
		driver.findElement(ElementsFornecedor.FORM_ADICIONAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.FORM_TIPO_DESPESA_ADICIONADO));
		Util.selectOption(ElementsFornecedor.FORM_TIPO_DESPESA,
Padroes.NOME_MATERIA_PRIMA);
		driver.findElement(ElementsFornecedor.FORM_ADICIONAR).click();
		Util.wait.until(ExpectedConditions
				.visibilityOfElementLocated(ElementsFornecedor.FORM_
TIPO_DESPESA_ADICIONADO_MATERIA_PRIMA));
		driver.findElement(ElementsFornecedor.FORM_SALVAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

	driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
	}
```

```
	@Test(groups = { "edit" })
	public void setUpEdit() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.NOVO_REGISTRO));
		Util.acionaElemento(ElementsFornecedor.TABLE,
ElementsFornecedor.TABLE_VALUE, Padroes.NOME_FORNECEDOR,
			ElementsRastreador.TABLE_EDITAR);
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
	public void editFornecedor() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.FORM_NOME));

driver.findElement(ElementsFornecedor.FORM_NOME).sendKeys(Padroes.EDITADO);
		driver.findElement(ElementsFornecedor.FORM_SALVAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

	driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.NOVO_REGISTRO));
	}

	@Test(groups = { "delete" })
	public void deleteUniqueFornecedor() {
```

```
            boolean existeDespesa = true;


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.NOVO_REGISTRO));
            Util.acionaElemento(ElementsFornecedor.TABLE,
ElementsFornecedor.TABLE_VALUE, Padroes.NOME_FORNECEDOR
                        + Padroes.EDITADO, ElementsRastreador.TABLE_EDITAR);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.FORM_APAGAR_DESPESA));
            while (existeDespesa) {
                  try {

      driver.findElement(ElementsFornecedor.FORM_APAGAR_DESPESA).click();
                        Util.waitJsBlock();
                  } catch (NoSuchElementException nsee) {
                        existeDespesa = false;
                  }
            }
            driver.findElement(ElementsFornecedor.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.NOVO_REGISTRO));
            Util.acionaElemento(ElementsFornecedor.TABLE,
ElementsFornecedor.TABLE_VALUE, Padroes.NOME_FORNECEDOR
                        + Padroes.EDITADO,
```

```
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosFornecedor() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.NOVO_REGISTRO));

            driver.findElement(ElementsFornecedor.NOVO_REGISTRO).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsForne
cedor.FORM_SALVAR));

            driver.findElement(ElementsFornecedor.FORM_SALVAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));


Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsFornecedor.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
```

```
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.FORNECEDOR);
            Util.endTestMsg(TestFornecedor.class.getSimpleName());
      }
}
```

```
package perfil.geral.cadastro.materiaPrima;

import org.openqa.selenium.By;

public class ElementsMateriaPrima {

    // Elementos da tela principal - Botões
    public static final By NOVO_REGISTRO = By

.id("startForm:telaTipoDespesabasepanelcontentlistgiNovoRegistro");

    // Elementos da tela principal - Tabela
    public static final By TABLE_ORDER = By

.id("startForm:telaTipoDespesadataTable:telaTipoDespesadataTablecolumn1head
er:sortDiv");
    public static final String TABLE =
"startForm:telaTipoDespesadataTable:";
    public static final String TABLE_VALUE =
":telaTipoDespesadataTablecolumn1otValue1";

    // Elementos do formulário - Botões
    public static final By FORM_SALVAR = By
          .id("startForm:telaTipoDespesabasepanelcontentformbtsalvarbt");
    public static final By FORM_LISTAR = By
          .id("startForm:telaTipoDespesabasepanelcontentformbtlistarbt");

    // Elementos do formulário - Select
    public static final By FORM_CATEGORIA = By
          .id("startForm:telaTipoDespesadespesaPai");
    public static final By FORM_UNIDADE_MEDIDA = By
          .id("startForm:telaTipoDespesaunidade");
```

```
    // Elementos do formulário - Campos
    public static final By FORM_NOME =
By.id("startForm:telaTipoDespesanome");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Nome] é obrigatório.";
}

package perfil.geral.cadastro.materiaPrima;

import org.testng.Assert;
import org.testng.annotations.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestMateriaPrima {
      WebDriver driver;

      @BeforeClass(alwaysRun = true)
      public void setUpMateriaPrima() {
            Util.startTestMsg(TestMateriaPrima.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.MATERIA_PRIMA);
      }
```

```
	@Test(groups = { "insert" })
	public void setUpInsertMateriaPrima() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.NOVO_REGISTRO));
		driver.findElement(ElementsMateriaPrima.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"setUpInsertMateriaPrima")
	public void insertUniqueMateriaPrima() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.FORM_CATEGORIA));
		Util.selectOption(ElementsMateriaPrima.FORM_CATEGORIA,
Padroes.NOME_CATEGORIA);
		Util.waitJsBlock();

driver.findElement(ElementsMateriaPrima.FORM_NOME).sendKeys(Padroes.NOME_MA
TERIA_PRIMA);
		Util.selectOption(ElementsMateriaPrima.FORM_UNIDADE_MEDIDA,
Padroes.NOME_UNIDADE_MEDIDA);
		driver.findElement(ElementsMateriaPrima.FORM_SALVAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

	driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
	}
```

```
	@Test(groups = { "edit" })
	public void setUpEdit() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.NOVO_REGISTRO));
		Util.acionaElemento(ElementsMateriaPrima.TABLE,
ElementsMateriaPrima.TABLE_VALUE, Padroes.NOME_MATERIA_PRIMA,
				ElementsRastreador.TABLE_EDITAR);
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
	public void editUniqueMateriaPrima() {

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.FORM_NOME));

driver.findElement(ElementsMateriaPrima.FORM_NOME).sendKeys(Padroes.EDITADO
);
		driver.findElement(ElementsMateriaPrima.FORM_SALVAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

	driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.NOVO_REGISTRO));
	}

	@Test(groups = { "delete" })
```

```
	public void deleteUniqueMateriaPrima() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.NOVO_REGISTRO));
		Util.acionaElemento(ElementsMateriaPrima.TABLE,
ElementsMateriaPrima.TABLE_VALUE, Padroes.NOME_MATERIA_PRIMA
				+ Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
		Util.excluirRegistro(true, true);
	}

	@Test(groups = { "verify", "campos" })
	public void validacaoCamposObrigatoriosMateriaPrima() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.NOVO_REGISTRO));

		driver.findElement(ElementsMateriaPrima.NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMater
iaPrima.FORM_SALVAR));

		driver.findElement(ElementsMateriaPrima.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
```

```
                        ElementsMateriaPrima.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.MATERIA_PRIMA);
            Util.endTestMsg(TestMateriaPrima.class.getSimpleName());
      }
}


package perfil.geral.cadastro.meuCadastro;

import org.openqa.selenium.By;

public class ElementsMeusCadastros {

      // Elementos da tela principal - Botões
      public static final By NOVO_REGISTRO =
By.id("startForm:telaElobasepanelcontentlistgiNovoRegistro");

      // Elementos da tela principal - Tabela
      public static final String TABLE = "startForm:telaElodataTable:";
      public static final String TABLE_VALUE =
":telaElodataTablecolumn0otValue0";
      public static final By TABLE_ORDER =
By.id("startForm:telaElodataTable:telaElodataTablecolumn0header:sortDiv");

      // Elementos do Forumlário - Abas
      public static final By TAB_CERTIFICADO =
By.id("startForm:telaElotpElotabOrganico_lbl");

      // Elementos do Formulário - Geral
      public static final By FORM_SALVAR =
```

```
By.id("startForm:telaElobasepanelcontentformbtsalvarbt");

      // Elementos do Formulário - Aba Dados gerais
      public static final By FORM_GERAIS_RADIO_PESSOA_FISICA =
By.id("startForm:telaEloradioSelect:1");
      public static final By FORM_GERAIS_NOME_FANTASIA =
By.id("startForm:telaElotpElotabDadosGeraispn1nome");
      public static final By FORM_GERAIS_INSCRICAO_RURAL =
By.id("startForm:telaElotpElotabDadosGeraispn1inscricaoRural");
      public static final By FORM_GERAIS_NOME_RESPONSAVEL = By
                  .id("startForm:telaElotpElotabDadosGeraispn1nomeResponsav
el");
      public static final By FORM_GERAIS_CPF =
By.id("startForm:telaElotpElotabDadosGeraispn1cpfResponsavel");
      public static final By FORM_GERAIS_ENDERECO =
By.id("startForm:telaElotpElotabDadosGeraispn1logradouro");
      public static final By FORM_GERAIS_UF =
By.id("startForm:telaElotpElotabDadosGeraispn1uf");
      public static final By FORM_GERAIS_MUNICIPIO =
By.id("startForm:telaElotpElotabDadosGeraispn1municipio");
      public static final By FORM_GERAIS_CONTATO =
By.id("startForm:telaElotpElotabDadosGeraispn3contato");
      public static final By FORM_GERAIS_TELEFONE =
By.id("startForm:telaElotpElotabDadosGeraispn3telefone");
      public static final By FORM_GERAIS_EMAIL =
By.id("startForm:telaElotpElotabDadosGeraispn3email");
      public static final By FORM_GERAIS_OUTROS_CONTATOS =
By.id("startForm:telaElotpElotabDadosGeraispn3outrosContatos");
      public static final By FORM_GERAIS_SITE =
By.id("startForm:telaElotpElotabDadosGeraispn3url");
      public static final By FORM_GERAIS_DESCRICAO =
By.id("startForm:telaElotpElotabDadosGeraispn3descricao");
      public static final By FORM_GERAIS_SITUACAO =
By.id("startForm:telaElotpElotabDadosGeraispn3situacao");

      // Elementos do Formulário - Aba Certificado
      public static final By FORM_CERTIFICADO_NOVO_REGISTRO = By
                  .id("startForm:telaElotpElotabOrganicopnNovoRegistrogiNov
oRegistro");
      public static final By FORM_CERTIFICADO_NUMERO =
By.id("formModalPanel:telaElonumeroCertificado");
```

```
      public static final By FORM_CERTIFICADO_CERTIFICADOR =
By.id("formModalPanel:telaElocertificador");
      public static final By FORM_CERTIFICADO_DT_INICIAL =
By.id("formModalPanel:telaElodataInicialInputDate");
      public static final By FORM_CERTIFICADO_DT_FINAL =
By.id("formModalPanel:telaElodataFinalInputDate");
      public static final By FORM_CERTIFICADO_TIPO =
By.id("formModalPanel:telaElotipoCertificado");
      public static final By FORM_CERTIFICADO_PRODUTO =
By.id("formModalPanel:telaElotipoProdutoCertificado");
      public static final By FORM_CERTIFICADO_QTDE =
By.id("formModalPanel:telaEloquantidadeCertificado");
      public static final By FORM_CERTIFICADO_UNIDADE_MEDIDA =
By.id("formModalPanel:telaElopnCertificadopnFormunidade");
      public static final By FORM_CERTIFICADO_ADICIONAR =
By.id("formModalPanel:telaElobtAdicionartpcertbt");
      public static final By FORM_CERTIFICADO_INCLUIR =
By.id("formModalPanel:telaElopnCertificadobtsalvarbt");
      public static final By FORM_CERTIFICADO_PRODUTO_ADICIONADO = By
                  .id("formModalPanel:telaElodataTableCertificadoTiposProdu
to:0:telaElodataTableCertificadoTiposProdutocolumn0otValue0");
      public static final By FORM_CERTIFICADO_INLCUIDO = By
                  .id("startForm:telaElotbOrganicos:0:telaElotbOrganicoscol
umn0otValue0");
      public static final By FORM_CERTIFICADO_FECHAR =
By.id("formModalPanel:telaElopnCertificadobtfecharbt");
      public static By FORM_ERRO =
By.id("formModalPanel:telaElopnCertificadopnErrorWrappertext");

      public static final By BODY =
By.id("startForm:telaElobasepanel_body");

      // Mensagem de validação dos campos obrigatorios
      public static String MSG_VALIDA_CAMPOS_PJ = "Erro" + "\n" + "O campo
[Nome fantasia] é obrigatório." + "\n"
                  + "O campo [Razão social] é obrigatório." + "\n" + "O
campo [CNPJ] é obrigatório." + "\n"
                  + "O campo [Endereço] é obrigatório." + "\n" + "O campo
[Município] é obrigatório." + "\n"
                  + "O campo [Latitude] é obrigatório." + "\n" + "O campo
[Longitude] é obrigatório." + "\n"
```

```
                + "O campo [Situação] é obrigatório.";
      public static String MSG_VALIDA_CAMPOS_PF = "Erro" + "\n" + "O campo
[Nome fantasia] é obrigatório." + "\n"
                + "O campo [Nome do responsável] é obrigatório." + "\n" +
"O campo [CPF do responsável] é obrigatório." + "\n"
                + "O campo [Endereço] é obrigatório." + "\n" + "O campo
[Município] é obrigatório." + "\n"
                + "O campo [Latitude] é obrigatório." + "\n" + "O campo
[Longitude] é obrigatório." + "\n"
                + "O campo [Situação] é obrigatório.";
      public static String MSG_VALIDA_CAMPOS_CERTIFICADO = "O campo [Número
do certificado] é obrigatório." + "\n"
                + "O campo [Certificador] é obrigatório." + "\n" + "O
campo [Tipo de certificado] é obrigatório." + "\n"
                + "O campo [Data inicial] é obrigatório." + "\n" + "O
campo [Data final] é obrigatório.";

}

package perfil.geral.cadastro.meuCadastro;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestMeuCadastro {
      WebDriver driver;
      String dataAtual = Util.dataAtual();

      @BeforeClass(alwaysRun = true)
      public void setUpMeuCadastro() {
            Util.startTestMsg(TestMeuCadastro.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.CADASTRO,
```

```
ElementsRastreador.MEUS_CADASTROS);
      }

      @Test(groups = { "insert" })
      public void setUpInsertCadastro() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.NOVO_REGISTRO));

      driver.findElement(ElementsMeusCadastros.NOVO_REGISTRO).click();
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"setUpInsertCadastro")
      public void preencheDadosGeraisMeusCadastros() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_GERAIS_NOME_FANTASIA));


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_NOME_FANTASIA).sendKey
s(Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_INSCRICAO_RURAL).sendK
eys(Padroes.NUMERO_VINTE);
            Util.scrollUp();


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_RADIO_PESSOA_FISICA).c
lick();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_GERAIS_CPF));


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_NOME_RESPONSAVEL).send
Keys(
```

```
                        "Responsavel " + Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_CPF).sendKeys(Padroes.
CPF);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_ENDERECO).sendKeys("En
dereco " + Padroes.NOME_MEU_CADASTRO);
            Util.selectOption(ElementsMeusCadastros.FORM_GERAIS_UF, "MG");
            Util.waitJsBlock();
            Util.selectOption(ElementsMeusCadastros.FORM_GERAIS_MUNICIPIO,
"Alfenas");

      driver.findElement(ElementsRastreador.LAT_GRAUS).sendKeys("15");

      driver.findElement(ElementsRastreador.LAT_MINUTOS).sendKeys("25");

      driver.findElement(ElementsRastreador.LAT_SEGUNDOS).sendKeys("35");

      driver.findElement(ElementsRastreador.LNG_GRAUS).sendKeys("50");

      driver.findElement(ElementsRastreador.LNG_MINUTOS).sendKeys("60");

      driver.findElement(ElementsRastreador.LNG_SEGUNDOS).sendKeys("70");


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_CONTATO).sendKeys("Con
tato " + Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_TELEFONE).sendKeys("(3
1) 5687-1548");


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_EMAIL).sendKeys("cadas
tro01@email.com");


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_OUTROS_CONTATOS).sendK
eys(
```

```
                        "Outros contatos " + Padroes.NOME_MEU_CADASTRO);
            driver.findElement(ElementsMeusCadastros.FORM_GERAIS_DESCRICAO)
                        .sendKeys("Descricao " + Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_SITE).sendKeys("www.me
ucadastro.com.br");
            Util.selectOption(ElementsMeusCadastros.FORM_GERAIS_SITUACAO,
"Ativo");
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"preencheDadosGeraisMeusCadastros", enabled = false)
      public void preencheCertificadoMeusCadastros() {
            Util.scrollUp();

      driver.findElement(ElementsMeusCadastros.TAB_CERTIFICADO).click();
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsMeusCadastros.FO
RM_CERTIFICADO_NOVO_REGISTRO));


driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_NOVO_REGISTRO).cl
ick();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_CERTIFICADO_NUMERO));
            Util.waitJsBlock();


driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_NUMERO).sendKeys(
Padroes.NUMERO_VINTE);


Util.selectOption(ElementsMeusCadastros.FORM_CERTIFICADO_CERTIFICADOR,
Padroes.NOME_CERTIFICADOR);
            Util.waitJsBlock();
            Util.selectOption(ElementsMeusCadastros.FORM_CERTIFICADO_TIPO,
Padroes.TIPO_CERTIFICADO);
            Util.waitJsBlock();
```

```
driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_DT_INICIAL).sendK
eys(dataAtual);


driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_DT_FINAL).sendKey
s(dataAtual);

      Util.selectOption(ElementsMeusCadastros.FORM_CERTIFICADO_PRODUTO,
Padroes.NOME_PRODUTO);


driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_QTDE).sendKeys("1
1.5");


Util.selectOption(ElementsMeusCadastros.FORM_CERTIFICADO_UNIDADE_MEDIDA,
"kg");


driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_ADICIONAR).click(
);
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsMeusCadastros.FO
RM_CERTIFICADO_PRODUTO_ADICIONADO));


driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"preencheDadosGeraisMeusCadastros")
      public void insertUniqueMeusCadastros() {
            driver.findElement(ElementsMeusCadastros.FORM_SALVAR).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.NOVO_REGISTRO));
            Util.acionaElemento(ElementsMeusCadastros.TABLE,
ElementsMeusCadastros.TABLE_VALUE, Padroes.NOME_MEU_CADASTRO,
                        ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueCadastro() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_GERAIS_NOME_FANTASIA));


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_NOME_FANTASIA).sendKey
s(Padroes.EDITADO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_NOME_RESPONSAVEL).send
Keys(Padroes.EDITADO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_CPF).sendKeys(Padroes.
CPF_EDITADO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_ENDERECO).sendKeys(Pad
roes.EDITADO);
            Util.selectOption(ElementsMeusCadastros.FORM_GERAIS_UF, "CE");
            Util.waitJsBlock();
```

```
            Util.selectOption(ElementsMeusCadastros.FORM_GERAIS_MUNICIPIO,
"Acarape");
            driver.findElement(ElementsRastreador.LAT_GRAUS).clear();

      driver.findElement(ElementsRastreador.LAT_GRAUS).sendKeys("31");
            driver.findElement(ElementsRastreador.LAT_MINUTOS).clear();

      driver.findElement(ElementsRastreador.LAT_MINUTOS).sendKeys("57");
            driver.findElement(ElementsRastreador.LAT_SEGUNDOS).clear();

      driver.findElement(ElementsRastreador.LAT_SEGUNDOS).sendKeys("87");
            driver.findElement(ElementsRastreador.LNG_GRAUS).clear();

      driver.findElement(ElementsRastreador.LNG_GRAUS).sendKeys("45");
            driver.findElement(ElementsRastreador.LNG_MINUTOS).clear();

      driver.findElement(ElementsRastreador.LNG_MINUTOS).sendKeys("87");
            driver.findElement(ElementsRastreador.LNG_SEGUNDOS).clear();

      driver.findElement(ElementsRastreador.LNG_SEGUNDOS).sendKeys("12");


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_CONTATO).sendKeys(Padr
oes.EDITADO);


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_TELEFONE).clear();


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_TELEFONE).sendKeys("(6
1) 7842-6587");

      driver.findElement(ElementsMeusCadastros.FORM_GERAIS_EMAIL).clear();


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_EMAIL).sendKeys("cadas
tro01editado@email.com");


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_OUTROS_CONTATOS).sendK
eys(Padroes.EDITADO);
```

```
driver.findElement(ElementsMeusCadastros.FORM_GERAIS_DESCRICAO).sendKeys(Pa
droes.EDITADO);
            driver.findElement(ElementsMeusCadastros.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueCadastro() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.NOVO_REGISTRO));
            Util.acionaElemento(ElementsMeusCadastros.TABLE,
ElementsMeusCadastros.TABLE_VALUE, Padroes.NOME_MEU_CADASTRO
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosMeuCadastroPessoaJuridica() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.NOVO_REGISTRO));


      driver.findElement(ElementsMeusCadastros.NOVO_REGISTRO).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_SALVAR));

            driver.findElement(ElementsMeusCadastros.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsMeusCadastros.MSG_VALIDA_CAMPOS_PJ,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.scrollUp();
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosMeuCadastroPessoaJuridica")
      public void validacaoCamposObrigatoriosMeuCadastroPessoaFisica() {


driver.findElement(ElementsMeusCadastros.FORM_GERAIS_RADIO_PESSOA_FISICA).c
lick();



Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_GERAIS_CPF));
            driver.findElement(ElementsMeusCadastros.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));
```

```
Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsMeusCadastros.MSG_VALIDA_CAMPOS_PF,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
            Util.scrollUp();
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosMeuCadastroPessoaFisica")
      public void validacaoCamposObrigatoriosCertificadoMeuCadastro() {

      driver.findElement(ElementsMeusCadastros.TAB_CERTIFICADO).click();
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsMeusCadastros.FO
RM_CERTIFICADO_NOVO_REGISTRO));



driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_NOVO_REGISTRO).cl
ick();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_CERTIFICADO_INCLUIR));



driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_INCLUIR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMeusC
adastros.FORM_ERRO));



Assert.assertEquals(driver.findElement(ElementsMeusCadastros.FORM_ERRO).get
Text(),
```

```
                        ElementsMeusCadastros.MSG_VALIDA_CAMPOS_CERTIFICADO,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);


driver.findElement(ElementsMeusCadastros.FORM_CERTIFICADO_FECHAR).click();
            Util.scrollUp();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.MEUS_CADASTROS);
            Util.endTestMsg(TestMeuCadastro.class.getSimpleName());
      }
}

package perfil.geral.cadastro.origem;

import org.openqa.selenium.By;

public class ElementsOrigens {

      // Elementos da tela principal - Botões
      public static final By NOVO_REGISTRO =
By.id("startForm:telaElobasepanelcontentlistgiNovoRegistro");

      // Elementos da tela principal - Tabela
      public static final String TABLE = "startForm:telaElodataTable:";
      public static final By TABLE_ORDER =
By.id("startForm:telaElodataTable:telaElodataTablecolumn0header:sortDiv");
      public static final String TABLE_VALUE =
":telaElodataTablecolumn0otValue0";

      // Elementos do formulário - Abas
      public static final By TAB_TIPOS_PRODUTO =
By.id("startForm:telaElotpElotabProdutos_shifted");
      public static final By TAB_TALHOES =
By.id("startForm:telaElotpElotabTalhoes_lbl");
      public static final By TAB_DEFENSIVOS =
By.id("startForm:telaElotpElotabDefensivos_lbl");
      public static final By TAB_CERTIFICADO =
```

```
By.id("startForm:telaElotpElotabOrganico_lbl");
      public static final By TAB_FOTOS =
By.id("startForm:telaElotpElotabFotos_lbl");
      public static final By TAB_DOCUMENTOS =
By.id("startForm:telaElotpElotabDocumentos_lbl");

      // Elementos do formulário - Aba 'Dados gerais'
      public static final By FORM_RADIO_PESSOA_FISICA =
By.id("startForm:telaEloradioSelect:1");
      public static final By FORM_NOME_PROPRIEDADE =
By.id("startForm:telaElotpElotabDadosGeraispn1nome");
      public static final By FORM_INSCRICAO_RURAL =
By.id("startForm:telaElotpElotabDadosGeraispn1inscricaoRural");
      public static final By FORM_NOME_RESPONSAVEL =
By.id("startForm:telaElotpElotabDadosGeraispn1nomeResponsavel");
      public static final By FORM_CPF_RESPONSAVEL =
By.id("startForm:telaElotpElotabDadosGeraispn1cpfResponsavel");
      public static final By FORM_RESPONSAVEL_TECNICO =
By.id("startForm:telaElotpElotabDadosGeraispn1nomeResInt");
      public static final By FORM_NU_REGISTRO_CONSELHO = By
                  .id("startForm:telaElotpElotabDadosGeraispn1nomeIdentific
acaoTec");
      public static final By FORM_ENDERECO =
By.id("startForm:telaElotpElotabDadosGeraispn1logradouro");
      public static final By FORM_UF =
By.id("startForm:telaElotpElotabDadosGeraispn1uf");
      public static final By FORM_MUNICIPIO =
By.id("startForm:telaElotpElotabDadosGeraispn1municipio");
      public static final By FORM_CONTATO =
By.id("startForm:telaElotpElotabDadosGeraispn3contato");
      public static final By FORM_TELEFONE =
By.id("startForm:telaElotpElotabDadosGeraispn3telefone");
      public static final By FORM_EMAIL =
By.id("startForm:telaElotpElotabDadosGeraispn3email");
      public static final By FORM_DEMAIS_CONTATOS =
By.id("startForm:telaElotpElotabDadosGeraispn3outrosContatos");
      public static final By FORM_AREA_TOTAL =
By.id("startForm:telaEloareaTotal");
      public static final By FORM_DESCRICAO =
By.id("startForm:telaElotpElotabDadosGeraispn3descricao");
      public static final By FORM_SITE =
```

```
By.id("startForm:telaElotpElotabDadosGeraispn3url");
      public static final By FORM_SITUACAO =
By.id("startForm:telaElotpElotabDadosGeraispn3situacao");

      // Elementos do formulário - Aba 'Tipos de Produto'
      public static final By FORM_TIPO_PRODUTO =
By.id("startForm:telaElotpElotabProdutoscombocomboboxField");
      public static final By FORM_TIPO_PRODUTO_SELECIONAR = By
                  .xpath("//div[@id='startForm:telaElotpElotabProdutoscombo
list']/span[1]");
      public static final By FORM_TIPO_PRODUTO_ADICIONAR =
By.id("startForm:telaElobtAdicionarbt");

      // Elementos do formulário - Aba 'Talhões'
      public static final By FORM_TALHAO_NOVO_REGISTRO = By
                  .id("startForm:telaElotpElotabTalhoespnNovoRegistrogiNovo
Registro");
      public static final By FORM_TALHAO_NOME =
By.id("formModalPanel:telaElopnTalhaopnFormnomeTalhao");
      public static final By FORM_TALHAO_SITUACAO =
By.id("formModalPanel:telaElopnTalhaopnFormsituacaoTalhao");
      public static final By FORM_TALHAO_LAT_GRAUS =
By.id("formModalPanel:telaElopnTalhaopnFormlatGrausTalhao");
      public static final By FORM_TALHAO_LAT_MINUTOS =
By.id("formModalPanel:telaElopnTalhaopnFormlatMinutosTalhao");
      public static final By FORM_TALHAO_LAT_SEGUNDOS =
By.id("formModalPanel:telaElopnTalhaopnFormlatSegundos");
      public static final By FORM_TALHAO_LNG_GRAUS =
By.id("formModalPanel:telaElopnTalhaopnFormlngGrausTalhao");
      public static final By FORM_TALHAO_LNG_MINUTOS =
By.id("formModalPanel:telaElopnTalhaopnFormlngMinutosTalhao");
      public static final By FORM_TALHAO_LNG_SEGUNDOS =
By.id("formModalPanel:telaElopnTalhaopnFormlngSegundosTalhao");
      public static final By FORM_TALHAO_AREA =
By.id("formModalPanel:telaElopnTalhaopnFormareaProdutiva");
      public static final By FORM_TALHAO_DESCRICAO =
By.id("formModalPanel:telaElopnTalhaopnFormdescricaoTalhao");
      public static final By FORM_TALHAO_INCLUIR =
By.id("formModalPanel:telaElopnTalhaobtsalvarbt");
      public static final By FORM_TALHAO_EDITAR =
By.id("startForm:telaElotbTalhoes:0:telaElotbTalhoespnEditDeletegiEdit");
```

```
      public static final By FORM_TALHAO_INSERIDO =
By.id("startForm:telaElotbTalhoes:0:telaElotbTalhoescolumn0otValue0");
      public static final By FORM_TALHAO_FECHAR =
By.id("formModalPanel:telaElopnTalhaobtfecharbt");
      public static By FORM_TALHAO_ERRO =
By.id("formModalPanel:telaElopnTalhaopnErrorWrappertext");

      // Elementos do formulário - Aba 'Defensivos'
      public static final By FORM_DEFENSIVO_NOVO_REGISTRO = By
                  .id("startForm:telaElotpElotabDefensivospnNovoRegistrogiN
ovoRegistro");
      public static final By FORM_DEFENSIVO_PRODUTO =
By.id("formModalPanel:telaElopnDefensivopnFormprodutoDefensivo");
      public static final By FORM_DEFENSIVO_NOME =
By.id("formModalPanel:telaElopnDefensivopnFormcombocomboboxField");
      public static final By FORM_DEFENSIVO_NOME_SELECIONAR = By
                  .xpath("//div[@id='formModalPanel:telaElopnDefensivopnFor
mcombolist']/span[1]");
      public static final By FORM_DEFENSIVO_ADICIONAR =
By.id("formModalPanel:telaElobtAdicionarDefSebt");
      public static final By FORM_DEFENSIVO_INCLUIR = By
                  .id("formModalPanel:telaElopnDefensivopnBotoesbtIncluirDe
fensivobt");
      public static final By FORM_DEFENSIVO_INSERIDO = By
                  .id("startForm:telaElotbDefensivos:0:telaElotbDefensivosc
olumn0otValue0");
      public static final By FORM_DEFENSIVO_FECHAR =
By.id("formModalPanel:telaElopnDefensivobtfecharbt");
      public static final By FORM_DEFENSIVO_ERRO =
By.id("formModalPanel:telaElopnDefensivopnErrosDefensivootMsgErro");

      // Elementos do formulário - Aba 'Certificado'
      public static final By FORM_CERTIFICADO_PRODUTO_ADICIONADO = By
                  .id("formModalPanel:telaElodataTableCertificadoTiposProdu
to:0:telaElodataTableCertificadoTiposProdutocolumn0otValue0");
      public static final By FORM_CERTIFICADO_NOVO_REGISTRO = By
                  .id("startForm:telaElotpElotabOrganicopnNovoRegistrogiNov
oRegistro");
      public static final By FORM_CERTIFICADO_NUMERO =
By.id("formModalPanel:telaElonumeroCertificado");
      public static final By FORM_CERTIFICADO_CERTIFICADOR =
```

```
By.id("formModalPanel:telaElocertificador");
      public static final By FORM_CERTIFICADO_TIPO =
By.id("formModalPanel:telaElotipoCertificado");
      public static final By FORM_CERTIFICADO_DT_INICIAL =
By.id("formModalPanel:telaElodataInicialInputDate");
      public static final By FORM_CERTIFICADO_DT_FINAL =
By.id("formModalPanel:telaElodataFinalInputDate");
      public static final By FORM_CERTIFICADO_PRODUTO =
By.id("formModalPanel:telaElotipoProdutoCertificado");
      public static final By FORM_CERTIFICADO_QTDE =
By.id("formModalPanel:telaEloquantidadeCertificado");
      public static final By FORM_CERTIFICADO_UNIDADE =
By.id("formModalPanel:telaElopnCertificadopnFormunidade");
      public static final By FORM_CERTIFICADO_PRODUTO_ADICIONAR =
By.id("formModalPanel:telaElobtAdicionartpcertbt");
      public static final By FORM_CERTIFICADO_INCLUIR =
By.id("formModalPanel:telaElopnCertificadobtsalvarbt");
      public static final By FORM_CERTIFICADO_FECHAR =
By.id("formModalPanel:telaElopnCertificadobtfecharbt");
      public static By FORM_CERTIFICADO_ERRO =
By.id("formModalPanel:telaElopnCertificadopnErrorWrappertext");

      // Elementos do formulário - Aba 'Fotos'
      public static final By FORM_FOTOS =
By.id("startForm:telaElofileUpload");
      public static final By FORM_FOTOS_ADICIONAR =
By.id("startForm:telaElofileUpload:file");
      public static final By FORM_FOTOS_ADICIONADO =
By.id("startForm:telaElodataTableFotos:0:textoLink");
      public static final By FORM_FOTOS_TP =
By.id("startForm:telaElodataTableFotos:0:selectTnull");

      // Elementos do formulário - Aba 'Documentos'
      public static final By FORM_DOCUMENTOS =
By.id("startForm:telaElopnTabDocumentosuploadDoc");
      public static final By FORM_DOCUMENTOS_ADICIONAR =
By.id("startForm:telaElopnTabDocumentosuploadDoc:file");
      public static final By FORM_DOCUMENTOS_ADICIONADO =
By.id("startForm:telaElodataTableDocumentos:0:textoLink");
      public static final By FORM_DOCUMENTOS_TP =
By.id("startForm:telaElodataTableDocumentos:0:selectTDocnull");
```

```
	// Mensagem de validação dos campos obrigatorios
	public static String MSG_VALIDA_CAMPOS_PJ = "Erro" + "\n" + "O campo
[Nome da propriedade] é obrigatório." + "\n"
			+ "O campo [Razão social] é obrigatório." + "\n" + "O
campo [CNPJ] é obrigatório." + "\n"
			+ "O campo [Endereço] é obrigatório." + "\n" + "O campo
[Município] é obrigatório." + "\n"
			+ "O campo [Latitude] é obrigatório." + "\n" + "O campo
[Longitude] é obrigatório." + "\n"
			+ "O campo [Situação] é obrigatório.";
	public static String MSG_VALIDA_CAMPOS_PF = "Erro" + "\n" + "O campo
[Nome da propriedade] é obrigatório." + "\n"
			+ "O campo [Nome do responsável] é obrigatório." + "\n" +
"O campo [CPF do responsável] é obrigatório." + "\n"
			+ "O campo [Endereço] é obrigatório." + "\n" + "O campo
[Município] é obrigatório." + "\n"
			+ "O campo [Latitude] é obrigatório." + "\n" + "O campo
[Longitude] é obrigatório." + "\n"
			+ "O campo [Situação] é obrigatório.";
	public static String MSG_VALIDA_CAMPOS_TALHAO = "O campo [Nome] é
obrigatório." + "\n"
			+ "O campo [Situação] é obrigatório.";
	public static String MSG_VALIDA_CAMPOS_DEFENSIVO = "O campo
[Produtos] é obrigatório." + "\n"
			+ "O campo [Defensivos Utilizados] é obrigatório.";
	public static String MSG_VALIDA_CAMPOS_CERTIFICADO = "O campo [Número
do certificado] é obrigatório." + "\n"
			+ "O campo [Certificador] é obrigatório." + "\n" + "O
campo [Tipo de certificado] é obrigatório." + "\n"
			+ "O campo [Data inicial] é obrigatório." + "\n" + "O
campo [Data final] é obrigatório.";
}

package perfil.geral.cadastro.origem;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;
```

```
import perfil.geral.util.Padroes;

import util.Configs;
import util.ElementsFiles;
import util.ElementsRastreador;
import util.MsgErro;
import util.Util;
import util.Util.BrowserType;

public class TestOrigem {
	WebDriver driver;
	String dataAtual = Util.dataAtual();

	@BeforeClass(alwaysRun = true)
	public void setUpOrigem() {
		Util.startTestMsg(TestOrigem.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.ORIGENS);
	}

	@Test(groups = { "insert" })
	public void setUpInsertOrigem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.NOVO_REGISTRO));
		driver.findElement(ElementsOrigens.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods = "setUpInsertOrigem")
	public void insertFirstOrigemGerais() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_NOME_PROPRIEDADE));


driver.findElement(ElementsOrigens.FORM_NOME_PROPRIEDADE).sendKeys(Padroes.
NOME_ORIGEM);
```

```
driver.findElement(ElementsOrigens.FORM_INSCRICAO_RURAL).sendKeys(Padroes.N
UMERO_VINTE);
            Util.scrollUp();

      driver.findElement(ElementsOrigens.FORM_RADIO_PESSOA_FISICA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_CPF_RESPONSAVEL));


driver.findElement(ElementsOrigens.FORM_NOME_RESPONSAVEL).sendKeys("Respons
avel " + Padroes.NOME_ORIGEM);


driver.findElement(ElementsOrigens.FORM_CPF_RESPONSAVEL).sendKeys(Padroes.C
PF);


driver.findElement(ElementsOrigens.FORM_RESPONSAVEL_TECNICO).sendKeys(Padro
es.NOME_ORIGEM + " Resp. Técnico");


driver.findElement(ElementsOrigens.FORM_NU_REGISTRO_CONSELHO).sendKeys(Padr
oes.NUMERO_VINTE);

      driver.findElement(ElementsOrigens.FORM_ENDERECO).sendKeys("Endereço
" + Padroes.NOME_ORIGEM);
            Util.selectOption(ElementsOrigens.FORM_UF, "DF");
            Util.waitJsBlock();
            Util.selectOption(ElementsOrigens.FORM_MUNICIPIO, "Recanto das
Emas");

      driver.findElement(ElementsRastreador.LAT_GRAUS).sendKeys("28");

      driver.findElement(ElementsRastreador.LAT_MINUTOS).sendKeys("37");

      driver.findElement(ElementsRastreador.LAT_SEGUNDOS).sendKeys("53");
```

```
      driver.findElement(ElementsRastreador.LNG_GRAUS).sendKeys("18");

      driver.findElement(ElementsRastreador.LNG_MINUTOS).sendKeys("61");

      driver.findElement(ElementsRastreador.LNG_SEGUNDOS).sendKeys("47");


driver.findElement(ElementsOrigens.FORM_CONTATO).sendKeys(Padroes.NOME_ORIG
EM + " contato");

      driver.findElement(ElementsOrigens.FORM_TELEFONE).sendKeys("(91)
8745-1259");


driver.findElement(ElementsOrigens.FORM_EMAIL).sendKeys("origem01@email.com
");


driver.findElement(ElementsOrigens.FORM_DEMAIS_CONTATOS).sendKeys(Padroes.N
OME_ORIGEM + " demais contatos");


driver.findElement(ElementsOrigens.FORM_AREA_TOTAL).sendKeys("1.450,00");


driver.findElement(ElementsOrigens.FORM_DESCRICAO).sendKeys("Descricao " +
Padroes.NOME_ORIGEM);


driver.findElement(ElementsOrigens.FORM_SITE).sendKeys("www.minhaorigem.com
.br");
            Util.selectOption(ElementsOrigens.FORM_SITUACAO, "Ativo");
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertFirstOrigemGerais")
      public void insertFirstOrigemTiposProduto() throws
InterruptedException {
            Util.scrollUp();
            driver.findElement(ElementsOrigens.TAB_TIPOS_PRODUTO).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TIPO_PRODUTO));
            Util.waitJsBlock();
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver ecDriver) {


ecDriver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO).sendKeys(Padroes.NO
ME_PRODUTO);
                        return
ecDriver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO).getAttribute("value
")
                                    .equals(Padroes.NOME_PRODUTO);
                  }
            });


driver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO_SELECIONAR).click();


driver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO_ADICIONAR).click();
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertFirstOrigemTiposProduto")
      public void insertFirstOrigemTalhoes() {
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {

      driver.findElement(ElementsOrigens.TAB_TALHOES).click();
                        return
driver.findElement(ElementsOrigens.FORM_TALHAO_NOVO_REGISTRO).isDisplayed()
;
                  }
            });


driver.findElement(ElementsOrigens.FORM_TALHAO_NOVO_REGISTRO).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TALHAO_NOME));

      driver.findElement(ElementsOrigens.FORM_TALHAO_NOME).sendKeys("Talhao
" + Padroes.NOME_ORIGEM);
            Util.selectOption(ElementsOrigens.FORM_TALHAO_SITUACAO,
"Ativo");


driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_GRAUS).sendKeys("28");


driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_MINUTOS).sendKeys("37");


driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_SEGUNDOS).sendKeys("53")
;


driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_GRAUS).sendKeys("18");


driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_MINUTOS).sendKeys("61");


driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_SEGUNDOS).sendKeys("47")
;


driver.findElement(ElementsOrigens.FORM_TALHAO_AREA).sendKeys("1.400,00");


driver.findElement(ElementsOrigens.FORM_TALHAO_DESCRICAO).sendKeys("Descric
ao talhão " + Padroes.NOME_ORIGEM);

      driver.findElement(ElementsOrigens.FORM_TALHAO_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
```

```
treador.MODAL));
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"insertFirstOrigemTalhoes")
	public void insertFirstOrigemDefensivo() {
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {

	driver.findElement(ElementsOrigens.TAB_DEFENSIVOS).click();
				return
driver.findElement(ElementsOrigens.FORM_DEFENSIVO_NOVO_REGISTRO).isDisplaye
d();
			}
		});


driver.findElement(ElementsOrigens.FORM_DEFENSIVO_NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_DEFENSIVO_PRODUTO));
		Util.selectOption(ElementsOrigens.FORM_DEFENSIVO_PRODUTO,
Padroes.NOME_PRODUTO);


driver.findElement(ElementsOrigens.FORM_DEFENSIVO_NOME).sendKeys("Nut");


driver.findElement(ElementsOrigens.FORM_DEFENSIVO_NOME_SELECIONAR).click();

	driver.findElement(ElementsOrigens.FORM_DEFENSIVO_ADICIONAR).click();

	driver.findElement(ElementsOrigens.FORM_DEFENSIVO_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
	}
```

```
	@Test(groups = { "insert" }, dependsOnMethods =
"insertFirstOrigemDefensivo")
	public void insertFirstOrigemCertificado() {
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {

	driver.findElement(ElementsOrigens.TAB_CERTIFICADO).click();
				return
driver.findElement(ElementsOrigens.FORM_CERTIFICADO_NOVO_REGISTRO).isDispla
yed();
			}
		});


driver.findElement(ElementsOrigens.FORM_CERTIFICADO_NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_CERTIFICADO_NUMERO));


driver.findElement(ElementsOrigens.FORM_CERTIFICADO_NUMERO).sendKeys(Padroe
s.NUMERO_VINTE);

	Util.selectOption(ElementsOrigens.FORM_CERTIFICADO_CERTIFICADOR,
Padroes.NOME_CERTIFICADOR);
		Util.waitJsBlock();
		Util.selectOption(ElementsOrigens.FORM_CERTIFICADO_TIPO,
Padroes.TIPO_CERTIFICADO);
		Util.waitJsBlock();


driver.findElement(ElementsOrigens.FORM_CERTIFICADO_DT_INICIAL).sendKeys(da
taAtual);


driver.findElement(ElementsOrigens.FORM_CERTIFICADO_DT_FINAL).sendKeys(data
Atual);
		Util.selectOption(ElementsOrigens.FORM_CERTIFICADO_PRODUTO,
Padroes.NOME_PRODUTO);
```

```
driver.findElement(ElementsOrigens.FORM_CERTIFICADO_QTDE).sendKeys("15");


driver.findElement(ElementsOrigens.FORM_CERTIFICADO_UNIDADE).sendKeys(Padro
es.NOME_UNIDADE_MEDIDA.toLowerCase());


driver.findElement(ElementsOrigens.FORM_CERTIFICADO_PRODUTO_ADICIONAR).clic
k();

      driver.findElement(ElementsOrigens.FORM_CERTIFICADO_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertFirstOrigemCertificado")
      public void insertFirstOrigemFoto() {
            Util.scrollUp();
            BrowserType browserType = Configs.browser;

            if (browserType == Util.BrowserType.FIREFOX) {
                  Util.wait.until(new ExpectedCondition<Boolean>() {
                        @Override
                        public Boolean apply(WebDriver arg0) {

      driver.findElement(ElementsOrigens.TAB_FOTOS).click();
                              return
driver.findElement(ElementsOrigens.FORM_FOTOS).isDisplayed();
                        }
                  });
                  Util.selectFile(ElementsOrigens.FORM_FOTOS_ADICIONAR,
ElementsFiles.LOGO_PARIPASSU_PNG);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_FOTOS_TP));
```

```
                  Util.selectOption(ElementsOrigens.FORM_FOTOS_TP,
"Logotipo");
            }
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertFirstOrigemFoto")
      public void insertFirstOrigemDocumento() {
            Util.scrollUp();
            BrowserType browserType = Configs.browser;
            if (browserType == Util.BrowserType.FIREFOX) {
                  Util.wait.until(new ExpectedCondition<Boolean>() {
                        @Override
                        public Boolean apply(WebDriver arg0) {

      driver.findElement(ElementsOrigens.TAB_DOCUMENTOS).click();
                              return
driver.findElement(ElementsOrigens.FORM_DOCUMENTOS).isDisplayed();
                        }
                  });

      Util.selectFile(ElementsOrigens.FORM_DOCUMENTOS_ADICIONAR,
ElementsFiles.DOC_PARIPASSU_PDF);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_DOCUMENTOS_TP));
                  Util.selectOption(ElementsOrigens.FORM_DOCUMENTOS_TP,
"Contrato de Abasto");
            }
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertFirstOrigemDocumento")
      public void saveInsertFirstOrigem() {
            driver.findElement(ElementsRastreador.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_SIM));
```

```
	driver.findElement(ElementsRastreador.FORM_SUCESSO_SIM).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"saveInsertFirstOrigem")
	public void insertSecondOrigemGerais() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_NOME_PROPRIEDADE));


driver.findElement(ElementsOrigens.FORM_NOME_PROPRIEDADE).sendKeys(Padroes.
NOME_ORIGEM_2);
		Util.scrollUp();

	driver.findElement(ElementsOrigens.FORM_RADIO_PESSOA_FISICA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_CPF_RESPONSAVEL));


driver.findElement(ElementsOrigens.FORM_NOME_RESPONSAVEL).sendKeys("Respons
avel " + Padroes.NOME_ORIGEM_2);


driver.findElement(ElementsOrigens.FORM_CPF_RESPONSAVEL).sendKeys(Padroes.C
PF);

	driver.findElement(ElementsOrigens.FORM_ENDERECO).sendKeys("Endereço
" + Padroes.NOME_ORIGEM_2);
		Util.selectOption(ElementsOrigens.FORM_UF, "DF");
		Util.waitJsBlock();
		Util.selectOption(ElementsOrigens.FORM_MUNICIPIO, "Recanto das
Emas");

	driver.findElement(ElementsRastreador.LAT_GRAUS).sendKeys("28");

	driver.findElement(ElementsRastreador.LAT_MINUTOS).sendKeys("37");
```

```
	driver.findElement(ElementsRastreador.LAT_SEGUNDOS).sendKeys("53");

	driver.findElement(ElementsRastreador.LNG_GRAUS).sendKeys("18");

	driver.findElement(ElementsRastreador.LNG_MINUTOS).sendKeys("61");

	driver.findElement(ElementsRastreador.LNG_SEGUNDOS).sendKeys("47");
		Util.selectOption(ElementsOrigens.FORM_SITUACAO, "Ativo");
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"insertSecondOrigemGerais")
	public void insertSecondOrigemTiposProduto() throws
InterruptedException {
		Util.scrollUp();
		driver.findElement(ElementsOrigens.TAB_TIPOS_PRODUTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TIPO_PRODUTO));
		Util.waitJsBlock();
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver ecDriver) {


ecDriver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO).sendKeys(Padroes.NO
ME_PRODUTO);
				return
ecDriver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO).getAttribute("value
")
					.equals(Padroes.NOME_PRODUTO);
			}
		});


driver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO_SELECIONAR).click();


driver.findElement(ElementsOrigens.FORM_TIPO_PRODUTO_ADICIONAR).click();
	}
```

```
	@Test(groups = { "insert" }, dependsOnMethods =
"insertSecondOrigemTiposProduto")
	public void insertSecondOrigemTalhoes() {
		Util.scrollUp();
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {

	driver.findElement(ElementsOrigens.TAB_TALHOES).click();
				return
driver.findElement(ElementsOrigens.FORM_TALHAO_NOVO_REGISTRO).isDisplayed()
;
			}
		});


driver.findElement(ElementsOrigens.FORM_TALHAO_NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TALHAO_NOME));

	driver.findElement(ElementsOrigens.FORM_TALHAO_NOME).sendKeys("Talhao
" + Padroes.NOME_ORIGEM);
		Util.selectOption(ElementsOrigens.FORM_TALHAO_SITUACAO,
"Ativo");

	driver.findElement(ElementsOrigens.FORM_TALHAO_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"insertSecondOrigemTalhoes")
	public void saveInsertSecondOrigem() {
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {
```

```
      driver.findElement(ElementsRastreador.FORM_SALVAR).click();
                        return
driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).isDisplayed();
                  }
            });

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.NOVO_REGISTRO));
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.NOVO_REGISTRO));
            Util.acionaElemento(ElementsOrigens.TABLE,
ElementsOrigens.TABLE_VALUE, Padroes.NOME_ORIGEM,
                        ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editFirstOrigemGerais() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated((ElementsOrig
ens.FORM_NOME_PROPRIEDADE)));


driver.findElement(ElementsOrigens.FORM_NOME_PROPRIEDADE).sendKeys(Padroes.
EDITADO);


driver.findElement(ElementsOrigens.FORM_NOME_RESPONSAVEL).sendKeys(Padroes.
EDITADO);

      driver.findElement(ElementsOrigens.FORM_CPF_RESPONSAVEL).clear();
```

```
driver.findElement(ElementsOrigens.FORM_CPF_RESPONSAVEL).sendKeys(Padroes.C
PF_EDITADO);


driver.findElement(ElementsOrigens.FORM_RESPONSAVEL_TECNICO).sendKeys(Padro
es.EDITADO);


driver.findElement(ElementsOrigens.FORM_ENDERECO).sendKeys(Padroes.EDITADO)
;
            Util.selectOption(ElementsOrigens.FORM_UF, "RS");
            Util.waitJsBlock();
            Util.selectOption(ElementsOrigens.FORM_MUNICIPIO, "Agudo");
            driver.findElement(ElementsRastreador.LAT_GRAUS).clear();

      driver.findElement(ElementsRastreador.LAT_GRAUS).sendKeys("31");
            driver.findElement(ElementsRastreador.LAT_MINUTOS).clear();

      driver.findElement(ElementsRastreador.LAT_MINUTOS).sendKeys("15");
            driver.findElement(ElementsRastreador.LAT_SEGUNDOS).clear();

      driver.findElement(ElementsRastreador.LAT_SEGUNDOS).sendKeys("28");
            driver.findElement(ElementsRastreador.LNG_GRAUS).clear();

      driver.findElement(ElementsRastreador.LNG_GRAUS).sendKeys("29");
            driver.findElement(ElementsRastreador.LNG_MINUTOS).clear();

      driver.findElement(ElementsRastreador.LNG_MINUTOS).sendKeys("12");
            driver.findElement(ElementsRastreador.LNG_SEGUNDOS).clear();

      driver.findElement(ElementsRastreador.LNG_SEGUNDOS).sendKeys("68");


driver.findElement(ElementsOrigens.FORM_CONTATO).sendKeys(Padroes.EDITADO);
            driver.findElement(ElementsOrigens.FORM_TELEFONE).clear();

      driver.findElement(ElementsOrigens.FORM_TELEFONE).sendKeys("(98)
1457-9856");
            driver.findElement(ElementsOrigens.FORM_EMAIL).clear();
```

```
driver.findElement(ElementsOrigens.FORM_EMAIL).sendKeys("origem01editado@em
ail.com");


driver.findElement(ElementsOrigens.FORM_DEMAIS_CONTATOS).sendKeys(Padroes.E
DITADO);
            driver.findElement(ElementsOrigens.FORM_AREA_TOTAL).clear();


driver.findElement(ElementsOrigens.FORM_AREA_TOTAL).sendKeys("28.000,00");


driver.findElement(ElementsOrigens.FORM_DESCRICAO).sendKeys(Padroes.EDITADO
);
      }

      @Test(groups = { "edit" }, dependsOnMethods =
"editFirstOrigemGerais")
      public void editFirstOrigemTalhoes() {
            Util.scrollUp();
            driver.findElement(ElementsOrigens.TAB_TALHOES).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TALHAO_EDITAR));
            driver.findElement(ElementsOrigens.FORM_TALHAO_EDITAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TALHAO_LAT_GRAUS));

      driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_GRAUS).clear();


driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_GRAUS).sendKeys("31");

      driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_MINUTOS).clear();
```

```
driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_MINUTOS).sendKeys("15");

      driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_SEGUNDOS).clear();


driver.findElement(ElementsOrigens.FORM_TALHAO_LAT_SEGUNDOS).sendKeys("28")
;

      driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_GRAUS).clear();


driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_GRAUS).sendKeys("29");

      driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_MINUTOS).clear();


driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_MINUTOS).sendKeys("12");

      driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_SEGUNDOS).clear();


driver.findElement(ElementsOrigens.FORM_TALHAO_LNG_SEGUNDOS).sendKeys("68")
;
            driver.findElement(ElementsOrigens.FORM_TALHAO_AREA).clear();


driver.findElement(ElementsOrigens.FORM_TALHAO_AREA).sendKeys("3.000,00");


driver.findElement(ElementsOrigens.FORM_TALHAO_DESCRICAO).sendKeys(Padroes.
EDITADO);

      driver.findElement(ElementsOrigens.FORM_TALHAO_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
      }

      @Test(groups = { "edit" }, dependsOnMethods =
"editFirstOrigemTalhoes")
```

```
	public void editFirstOrigemSave() {
		driver.findElement(ElementsRastreador.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

		driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.NOVO_REGISTRO));
	}

	@Test(groups = { "delete" })
	public void deleteFirstSecondOrigem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.NOVO_REGISTRO));
		Util.acionaElemento(ElementsOrigens.TABLE,
ElementsOrigens.TABLE_VALUE, Padroes.NOME_ORIGEM + Padroes.EDITADO,
				ElementsRastreador.TABLE_EXCLUIR);
		Util.excluirRegistro(true, true);
		Util.acionaElemento(ElementsOrigens.TABLE,
ElementsOrigens.TABLE_VALUE, Padroes.NOME_ORIGEM_2,
				ElementsRastreador.TABLE_EXCLUIR);
		Util.excluirRegistro(true, true);
	}

	@Test(groups = { "verify", "campos" })
	public void validacaoCamposObrigatoriosOrigensGeralPessoaJuridica() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.NOVO_REGISTRO));

		driver.findElement(ElementsOrigens.NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
```

```
eador.FORM_SALVAR));

            driver.findElement(ElementsRastreador.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsOrigens.MSG_VALIDA_CAMPOS_PJ,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.scrollUp();
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosOrigensGeralPessoaJuridica")
      public void validacaoCamposObrigatoriosOrigensGeralPessoaFisica() {

      driver.findElement(ElementsOrigens.FORM_RADIO_PESSOA_FISICA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_CPF_RESPONSAVEL));

            driver.findElement(ElementsRastreador.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));
```

```
Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsOrigens.MSG_VALIDA_CAMPOS_PF,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
            Util.scrollUp();
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosOrigensGeralPessoaFisica")
      public void validacaoCamposObrigatoriosOrigensTalhao() {
            driver.findElement(ElementsOrigens.TAB_TALHOES).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TALHAO_NOVO_REGISTRO));



driver.findElement(ElementsOrigens.FORM_TALHAO_NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TALHAO_INCLUIR));


      driver.findElement(ElementsOrigens.FORM_TALHAO_INCLUIR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_TALHAO_ERRO));



Assert.assertEquals(driver.findElement(ElementsOrigens.FORM_TALHAO_ERRO).ge
tText(),
                        ElementsOrigens.MSG_VALIDA_CAMPOS_TALHAO,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);
            driver.findElement(ElementsOrigens.FORM_TALHAO_FECHAR).click();
      }
```

```
	@Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosOrigensTalhao")
	public void validacaoCamposObrigatoriosOrigensDefensivo() {
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {

	driver.findElement(ElementsOrigens.TAB_DEFENSIVOS).click();
				return
driver.findElement(ElementsOrigens.FORM_DEFENSIVO_NOVO_REGISTRO).isDisplaye
d();
			}
		});



driver.findElement(ElementsOrigens.FORM_DEFENSIVO_NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_DEFENSIVO_INCLUIR));


	driver.findElement(ElementsOrigens.FORM_DEFENSIVO_INCLUIR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_DEFENSIVO_ERRO));



Assert.assertEquals(driver.findElement(ElementsOrigens.FORM_DEFENSIVO_ERRO)
.getText(),
				ElementsOrigens.MSG_VALIDA_CAMPOS_DEFENSIVO,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

	driver.findElement(ElementsOrigens.FORM_DEFENSIVO_FECHAR).click();
		Util.waitJsBlock();
	}
```

```
	@Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosOrigensDefensivo")
	public void validacaoCamposObrigatoriosOrigensCertificado() {
		driver.findElement(ElementsOrigens.TAB_CERTIFICADO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_CERTIFICADO_NOVO_REGISTRO));



driver.findElement(ElementsOrigens.FORM_CERTIFICADO_NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_CERTIFICADO_INCLUIR));


	driver.findElement(ElementsOrigens.FORM_CERTIFICADO_INCLUIR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsOrige
ns.FORM_CERTIFICADO_ERRO));



Assert.assertEquals(driver.findElement(ElementsOrigens.FORM_CERTIFICADO_ERR
O).getText(),
			ElementsOrigens.MSG_VALIDA_CAMPOS_CERTIFICADO,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

	driver.findElement(ElementsOrigens.FORM_CERTIFICADO_FECHAR).click();
	}

	@AfterClass(alwaysRun = true)
	public void tearDown() {
		Util.fechaGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.ORIGENS);
		Util.endTestMsg(TestOrigem.class.getSimpleName());
	}
}
```

```
package perfil.geral.cadastro.receita;

import org.openqa.selenium.By;

public class ElementsReceita {

	// Elementos da tela principal - Botões
	public static final By NOVO_REGISTRO =
By.id("startForm:telaReceitabasepanelcontentlistgiNovoRegistro");

	// Elementos da tela principal - Tabela
	public static final String TABLE = "startForm:telaReceitadataTable:";
	public static final By TABLE_ORDER =
By.id("startForm:telaReceitadataTable:telaReceitadataTablecolumn0otHeader0"
);
	public static final String TABLE_VALUE =
":telaReceitadataTablecolumn0otValue0";

	// Elementos do formulário
	public static final By FORM_NOME =
By.id("startForm:telaReceitatipoProduto");
	public static final By FORM_CALCULA_FATOR = By
			.id("startForm:telaReceitatpReceitatabDadosGeraiscalculaF
atorckProdutor");
	public static final By FORM_DESCRICAO =
By.id("startForm:telaReceitatpReceitatabDadosGeraisdescricao");
	public static final By FORM_TIPO_INGREDIENTE =
By.id("startForm:telaReceitatipoIngrediente");
	public static final By FORM_PROPORCAO =
By.id("startForm:telaReceitaproporcao");
	public static final By FORM_INGREDIENTE =
By.id("startForm:telaReceitatpReceitatabDadosGeraiscombocomboboxField");
	public static final By FORM_ADICIONAR =
By.id("startForm:telaReceitatpReceitatabDadosGeraisbtAdicionarbt");
	public static final By FORM_INGREDIENTE_PRIMEIRO = By
			.xpath("//div[@id='startForm:telaReceitatpReceitatabDados
Geraiscombolist']/span[1]");
	public static final By FORM_INGREDIENTE_CADASTRADO = By
			.id("startForm:telaReceitatbIngredientes:0:telaReceitatbI
ngredientescolumn1otValue1");
```

```
	public static final By FORM_INGREDIENTE_CADASTRADO_MATERIA_PRIMA = By
			.id("startForm:telaReceitatbIngredientes:1:telaReceitatbI
ngredientescolumn1otValue1");
	public static final By FORM_FATOR_CORRECAO =
By.id("startForm:telaReceitafatorCorrecao");
	public static final By FORM_SALVAR =
By.id("startForm:telaReceitabasepanelcontentformbtsalvarbt");

	// Mensagem de validação dos campos obrigatorios
	public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
[Nome da Receita] é obrigatório." + "\n"
			+ "O campo [Ingredientes] é obrigatório.";

}

package perfil.geral.cadastro.receita;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestReceita {
	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpReceita() {
		Util.startTestMsg(TestReceita.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.RECEITA);
	}

	@Test(groups = { "insert" })
```

```
	public void setUpInsertReceita() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.NOVO_REGISTRO));
		driver.findElement(ElementsReceita.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods = "setUpInsertReceita")
	public void insertUniqueReceita() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.FORM_NOME));


driver.findElement(ElementsReceita.FORM_NOME).sendKeys(Padroes.NOME_RECEITA
);
		driver.findElement(ElementsReceita.FORM_CALCULA_FATOR).click();
		Util.waitJsBlock();
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {

	driver.findElement(ElementsReceita.FORM_CALCULA_FATOR).click();
				return !
driver.findElement(ElementsReceita.FORM_CALCULA_FATOR).isSelected();
			}
		});


driver.findElement(ElementsReceita.FORM_DESCRICAO).sendKeys("Descrição " +
Padroes.NOME_RECEITA);
		Util.selectOption(ElementsReceita.FORM_TIPO_INGREDIENTE,
Padroes.NOME_TIPO_INGREDIENTE);
		Util.waitJsBlock();

	driver.findElement(ElementsReceita.FORM_PROPORCAO).sendKeys("25,00");


driver.findElement(ElementsReceita.FORM_INGREDIENTE).sendKeys(Padroes.NOME_
```

```
PRODUTO);


driver.findElement(ElementsReceita.FORM_INGREDIENTE_PRIMEIRO).click();
            driver.findElement(ElementsReceita.FORM_ADICIONAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.FORM_INGREDIENTE_CADASTRADO));

            Util.selectOption(ElementsReceita.FORM_TIPO_INGREDIENTE,
Padroes.NOME_CATEGORIA);

      driver.findElement(ElementsReceita.FORM_PROPORCAO).sendKeys("30,00");
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsReceita.FORM_INGREDIENTE).sendKeys(Padroes.NOME_
MATERIA_PRIMA);
                        return
driver.findElement(ElementsReceita.FORM_INGREDIENTE).getAttribute("value")
                                    .equals(Padroes.NOME_MATERIA_PRIMA);
                  }
            });


driver.findElement(ElementsReceita.FORM_INGREDIENTE_PRIMEIRO).click();
            driver.findElement(ElementsReceita.FORM_ADICIONAR).click();
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsReceita.FORM_ING
REDIENTE_CADASTRADO_MATERIA_PRIMA));

            driver.findElement(ElementsReceita.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
```

```
	}

	@Test(groups = { "edit" })
	public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.NOVO_REGISTRO));
		Util.acionaElemento(ElementsReceita.TABLE,
ElementsReceita.TABLE_VALUE, Padroes.NOME_RECEITA,
			ElementsRastreador.TABLE_EDITAR);
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
	public void editUniqueReceita() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.FORM_NOME));


driver.findElement(ElementsReceita.FORM_NOME).sendKeys(Padroes.EDITADO);
		driver.findElement(ElementsReceita.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

	driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.NOVO_REGISTRO));
	}

	@Test(groups = { "delete" })
	public void deleteUniqueReceita() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.NOVO_REGISTRO));
```

```
            Util.acionaElemento(ElementsReceita.TABLE,
ElementsReceita.TABLE_VALUE, Padroes.NOME_RECEITA + Padroes.EDITADO,
                        ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosReceita() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.NOVO_REGISTRO));

            driver.findElement(ElementsReceita.NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRecei
ta.FORM_SALVAR));

            driver.findElement(ElementsReceita.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsReceita.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CADASTRO,
ElementsRastreador.RECEITA);
            Util.endTestMsg(TestReceita.class.getSimpleName());
      }
```

```
}
package perfil.geral.configuracoes.cadastroNutricional;

import org.openqa.selenium.By;

public class ElementsCadastroNutricional {

	public static final By BODY =
By.id("startForm:telaCadastroNutricionalbasepanel_body");
	public static final String TABLE =
"startForm:telaCadastroNutricionaldataTable:";
	public static final By TABLE_ORDER = By
			.id("startForm:telaCadastroNutricionaldataTable:telaCadas
troNutricionaldataTablecolumn0header");
	public static final String TABLE_VALUE_FIRST_COL =
":telaCadastroNutricionaldataTablecolumn0otValue0";
	public static final String TABLE_VALUE_SECOND_COL =
":telaCadastroNutricionaldataTablecolumn1otValue1";
	public static final By FORM_PORCAO =
By.id("startForm:telaCadastroNutricionalitPorcao");
	public static final By FORM_POSSUI_GORDURA =
By.id("startForm:telaCadastroNutricionalgordura");
	public static final By FORM_CONSENTIMENTO =
By.id("startForm:telaCadastroNutricionalConsentimento");
	public static final By FORM_SALVAR =
By.id("startForm:telaCadastroNutricionalbasepanelcontentformbtsalvarbt");
	public static final By FORM_LISTAR =
By.id("startForm:telaCadastroNutricionalbasepanelcontentformbtlistarbt");

	public static final By FORM_QTDE_PORCAO_1 = By
			.id("startForm:telaCadastroNutricionaldataTableParametro:
0:telaCadastroNutricionalpnQtditQtd");
	public static final By FORM_QTDE_PORCAO_2 = By
			.id("startForm:telaCadastroNutricionaldataTableParametro:
1:telaCadastroNutricionalpnQtditQtd");
	public static final By FORM_QTDE_PORCAO_3 = By
			.id("startForm:telaCadastroNutricionaldataTableParametro:
2:telaCadastroNutricionalpnQtditQtd");
	public static final By FORM_QTDE_PORCAO_4 = By
			.id("startForm:telaCadastroNutricionaldataTableParametro:
3:telaCadastroNutricionalpnQtditQtd");
```

```
      public static final By FORM_QTDE_PORCAO_5 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
4:telaCadastroNutricionalpnQtditQtd");
      public static final By FORM_QTDE_PORCAO_6 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
5:telaCadastroNutricionalpnQtditQtd");
      public static final By FORM_QTDE_PORCAO_7 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
6:telaCadastroNutricionalpnQtditQtd");
      public static final By FORM_QTDE_PORCAO_8 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
7:telaCadastroNutricionalpnQtditQtd");

      public static final By FORM_VALOR_DIARIO_1 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
0:telaCadastroNutricionalpnVDitVD");
      public static final By FORM_VALOR_DIARIO_2 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
1:telaCadastroNutricionalpnVDitVD");
      public static final By FORM_VALOR_DIARIO_3 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
2:telaCadastroNutricionalpnVDitVD");
      public static final By FORM_VALOR_DIARIO_4 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
3:telaCadastroNutricionalpnVDitVD");
      public static final By FORM_VALOR_DIARIO_5 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
4:telaCadastroNutricionalpnVDitVD");
      public static final By FORM_VALOR_DIARIO_6 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
5:telaCadastroNutricionalpnVDitVD");
      public static final By FORM_VALOR_DIARIO_7 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
6:telaCadastroNutricionalpnVDitVD");
      public static final By FORM_VALOR_DIARIO_8 = By
                  .id("startForm:telaCadastroNutricionaldataTableParametro:
7:telaCadastroNutricionalpnVDitVD");

      public static final By FORM_CODIGOS_EAN =
By.id("startForm:telaCadastroNutricionallistSelectOneEan");
```

```
	public static final By FORM_CODIGOS_EAN_DIGITOS =
By.id("startForm:telaCadastroNutricionalpnFormCodigoEan");
	public static final By FORM_CODIGOS_EAN_EMBALAGEM =
By.id("startForm:telaCadastroNutricionalpnFormporcaoEan");

	public static final By FORM_CODIGOS_EAN_ADICIONAR =
By.id("startForm:telaCadastroNutricionalbtAdicionarbt");
	public static final By FORM_CODIGOS_EAN_REMOVER =
By.id("startForm:telaCadastroNutricionalbtRemoverbt");

	// Mensagem de validação dos campos obrigatorios
	public static String MSG_VALIDA_CAMPOS = "O campo [Porção] é
obrigatório. Quantidade por porção e Valor diário são obrigatórios para
todos os parâmetros. O campo [Código Ean (12 Dígitos)] é obrigatório. Você
precisa aceitar o termo de consentimento.";
}

package perfil.geral.configuracoes.cadastroNutricional;

import org.testng.Assert;
import org.testng.annotations.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestCadastroNutricional {
	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpCadastroNutricional() {

	Util.startTestMsg(TestCadastroNutricional.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CADASTRO_NUTRICIONAL);
		Util.acionaElemento(ElementsCadastroNutricional.TABLE,
```

```
ElementsCadastroNutricional.TABLE_VALUE_FIRST_COL,
                        Padroes.NOME_PRODUTO,
ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "config" })
      public void editUniqueCadastroNutricional() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsCadas
troNutricional.FORM_PORCAO));
            UtilCadastroNutricional.preencheCadastroNutricional(driver);
            UtilCadastroNutricional.preencheCodigoEAN(driver);


driver.findElement(ElementsCadastroNutricional.FORM_CONSENTIMENTO).click();

      driver.findElement(ElementsCadastroNutricional.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));


      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosCadatroNutricional() {

      driver.findElement(ElementsCadastroNutricional.FORM_SALVAR).click();

            Assert.assertEquals(

      driver.findElement(ElementsRastreador.MODAL_ERRO).getText()
                                    .equals(ElementsCadastroNutricional.MSG_
VALIDA_CAMPOS), false,
                        MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);


      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
```

```
	}

	@AfterClass(alwaysRun = true)
	public void tearDown() {
		Util.fechaGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CADASTRO_NUTRICIONAL);
		Util.endTestMsg(TestCadastroNutricional.class.getSimpleName());
	}
}

package perfil.geral.configuracoes.cadastroNutricional;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;

import perfil.geral.util.Padroes;

import util.Util;

public class UtilCadastroNutricional {

	public static void preencheCadastroNutricional(WebDriver driver) {
		if (!
(driver.findElement(ElementsCadastroNutricional.FORM_POSSUI_GORDURA).isSele
cted())) {


driver.findElement(ElementsCadastroNutricional.FORM_POSSUI_GORDURA).click()
;
		}


	driver.findElement(ElementsCadastroNutricional.FORM_PORCAO).clear();



driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_1).clear();
```

```
driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_2).clear();

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_3).clear();

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_4).clear();

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_5).clear();

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_6).clear();

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_7).clear();

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_8).clear();

driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_1).clear()
;

driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_2).clear()
;

driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_3).clear()
;

driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_4).clear()
;

driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_5).clear()
;
```

```
driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_6).clear()
;

driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_7).clear()
;

driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_8).clear()
;


driver.findElement(ElementsCadastroNutricional.FORM_PORCAO).sendKeys("Porçã
o de 100g");


driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_1).sendKeys
("176,3");

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_2).sendKeys
("6,94");

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_3).sendKeys
("2,31");

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_4).sendKeys
("3,14");

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_5).sendKeys
("0.012");

driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_6).sendKeys
```

```
("17,34");


driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_7).sendKeys
("0,00");


driver.findElement(ElementsCadastroNutricional.FORM_QTDE_PORCAO_8).sendKeys
("17,34");



driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_1).sendKey
s("8,8");


driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_2).sendKey
s("5,3");


driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_3).sendKey
s("4,3");


driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_4).sendKey
s("8,2");


driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_5).sendKey
s("0,0");


driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_6).sendKey
s("5,7");


driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_7).sendKey
s("0,0");


driver.findElement(ElementsCadastroNutricional.FORM_VALOR_DIARIO_8).sendKey
```

```
s("5,7");
      }

      public static void preencheCodigoEAN(WebDriver driver) {
            WebElement webElementListaCodigosEan =
driver.findElement(ElementsCadastroNutricional.FORM_CODIGOS_EAN);
            List<WebElement> webElementCodigosEan =
webElementListaCodigosEan.findElements(By.tagName("option"));

            if (webElementCodigosEan.size() > 0) {
                  for (WebElement codigoEan : webElementCodigosEan) {
                        codigoEan.click();


driver.findElement(ElementsCadastroNutricional.FORM_CODIGOS_EAN_REMOVER).cl
ick();

      Util.wait.until(ExpectedConditions.stalenessOf(codigoEan));
                  }
            }



driver.findElement(ElementsCadastroNutricional.FORM_CODIGOS_EAN_DIGITOS).se
ndKeys(Padroes.PRODUTO_CODIGO_EAN);


driver.findElement(ElementsCadastroNutricional.FORM_CODIGOS_EAN_EMBALAGEM).
sendKeys(Padroes.PRODUTO_EMBALAGEM_EAN);


driver.findElement(ElementsCadastroNutricional.FORM_CODIGOS_EAN_ADICIONAR).
click();
            Util.waitJsBlock();
      }

}

package perfil.geral.configuracoes.configuracaoEtiqueta;

import org.openqa.selenium.By;
```

```
public class ElementsConfiguracaoEtiqueta {

    // Elementos da Tela Principal - Tabela
    public static final String TABLE =
"startForm:telaConfiguracaoEtiquetaListdataTable:";
    public static final String TABLE_VALUE =
":telaConfiguracaoEtiquetaListdataTablecolumn1otValue1";
    public static final By TABLE_ORDER = By

.id("startForm:telaConfiguracaoEtiquetaListdataTable:telaConfiguracaoEtique
taListdataTablecolumn1otHeader1");

    // Elementos da Tela Principal
    public static final By NOVO_REGISTRO = By

.id("startForm:telaConfiguracaoEtiquetaListbasepanelcontentlistgiNovoRegist
ro");
    public static final By FORM_TIPO_IMPRESSAO = By
          .id("startForm:telaConfiguracaoEtiquetaListcomboTipoImpressao");
    public static final By FORM_TIPO_ETIQUETA = By
          .id("startForm:telaConfiguracaoEtiquetaListcomboTipoEtiqueta");
    public static final By FORM_LOGO_IMPRESSO = By
          .id("startForm:telaConfiguracaoEtiquetaListcomboOpcaoLogotipo");
    public static final By FORM_LINGUAGEM = By

.id("startForm:telaConfiguracaoEtiquetaListcomboLinguagemImpressao");
    public static final By FORM_DATA_EMB_FAB = By

.id("startForm:telaConfiguracaoEtiquetaListcomboOpcaoDataEmbalagemFabricaca
o");
    public static final By FORM_DATA_ENTREGA = By

.id("startForm:telaConfiguracaoEtiquetaListcomboOpcaoDataEntregaVisivel");
    public static final By FORM_SALVAR = By

.id("startForm:telaConfiguracaoEtiquetaListbasepanelcontentformbtsalvarbt")
;
    public static final By FORM_SUCESSO_OK = By
          .id("formModalPanel:pnMsgSucessoCadastrobtYesbt");
```

```
    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Selecione o Tipo de Impressão
Selecione o tipo de etiqueta";
    public static String MSG_VALIDA_CAMPOS_2 = "Selecione a Linguagem de
Impressão Selecione se deseja etiqueta com logotipo";
}

package perfil.geral.configuracoes.configuracaoEtiqueta;

import org.openqa.selenium.WebDriver;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestConfiguracaoEtiqueta {
      WebDriver driver;

      @BeforeClass(alwaysRun = true)
      public void setUpConfiguracaoEtiqueta() {

      Util.startTestMsg(TestConfiguracaoEtiqueta.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CONFIGURACAO_ETIQUETA);
      }

      @Test(groups = { "insert" })
      public void setUpInsertConfiguracaoEtiqueta() {


driver.findElement(ElementsConfiguracaoEtiqueta.NOVO_REGISTRO).click();
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"setUpInsertConfiguracaoEtiqueta")
      public void insertUniqueConfiguracaoEtiqueta() {
```

```
      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_TIPO_IMPRESSAO,
"Térmica");

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_TIPO_ETIQUETA,
Padroes.NOME_ETIQUETA);
            Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_LINGUAGEM,
"ppla");

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_LOGO_IMPRESSO,
Padroes.NAO);

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_DATA_EMB_FAB,
"Data de Embalagem");

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_DATA_ENTREGA,
Padroes.SIM);

      driver.findElement(ElementsConfiguracaoEtiqueta.FORM_SALVAR).click();

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
      }

      @Test(groups = { "edit" })
      public void setUpEditConfiguracaoEtiqueta() {
            Util.acionaElemento(ElementsConfiguracaoEtiqueta.TABLE,
ElementsConfiguracaoEtiqueta.TABLE_VALUE,
                        Padroes.NOME_ETIQUETA,
ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods =
"setUpEditConfiguracaoEtiqueta")
      public void editUniqueConfiguracaoEtiqueta() {
            Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_LINGUAGEM,
"ZPL");

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_LOGO_IMPRESSO,
Padroes.SIM);

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_DATA_EMB_FAB,
```

```
"Data de Fabricação");

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_DATA_ENTREGA,
Padroes.NAO);

      driver.findElement(ElementsConfiguracaoEtiqueta.FORM_SALVAR).click();

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();
      }

      @Test(groups = { "delete" })
      public void deleteUniqueConfiguracaoEtiqueta() {
            Util.acionaElemento(ElementsConfiguracaoEtiqueta.TABLE,
ElementsConfiguracaoEtiqueta.TABLE_VALUE,
                        Padroes.NOME_ETIQUETA,
ElementsRastreador.TABLE_EXCLUIR);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosConfiguracaoEtiqueta() {


driver.findElement(ElementsConfiguracaoEtiqueta.NOVO_REGISTRO).click();

      driver.findElement(ElementsConfiguracaoEtiqueta.FORM_SALVAR).click();

            Assert.assertEquals(

      driver.findElement(ElementsRastreador.MODAL_ERRO).getText()
                                    .equals(ElementsConfiguracaoEtiqueta.MSG
_VALIDA_CAMPOS), false,
                        MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);


      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();


      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_TIPO_IMPRESSAO,
"Térmica");

      Util.selectOption(ElementsConfiguracaoEtiqueta.FORM_TIPO_ETIQUETA,
```

```
Padroes.NOME_ETIQUETA);

      driver.findElement(ElementsConfiguracaoEtiqueta.FORM_SALVAR).click();

            Assert.assertEquals(

      driver.findElement(ElementsRastreador.MODAL_ERRO).getText()
                                    .equals(ElementsConfiguracaoEtiqueta.MSG
_VALIDA_CAMPOS_2), false,
                        MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);


      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CONFIGURACAO_ETIQUETA);

      Util.endTestMsg(TestConfiguracaoEtiqueta.class.getSimpleName());
      }
}

package perfil.geral.configuracoes.configuracaoGeral;

import org.openqa.selenium.By;

public class ElementsConfiguracaoGeral {

    // Div principal
    public static final By BODY = By
          .id("startForm:telaConfiguracaobasepanelcontent_body");

    // Elementos do Formulário
    public static final By FORM_PRECO_COMPRA_SIM = By
          .id("startForm:telaConfiguracaockPrecoProdutos:0");
    public static final By FORM_INSPECAO_RECEBIMENTO_SIM = By
          .id("startForm:telaConfiguracaockRealizaEntrada:0");
    public static final By FORM_REALIZA_CONTROLE_SIM = By
```

```
.id("startForm:telaConfiguracaockrealizaControleMotoristaVeiculo:0");
    public static final By FORM_CHECK_TODOS = By.id("startForm:ckTodos");
    public static final By FORM_SALVAR = By

.id("startForm:telaConfiguracaobasepanelcontentpnBotoesbtsalvarbt");
    public static final By FORM_TALHAO_SINGULAR = By

.id("startForm:telaConfiguracaobasepanelcontentpn1talhaoSingular");
    public static final By FORM_TALHAO_PLURAL = By
          .id("startForm:telaConfiguracaobasepanelcontentpn1talhaoPlural");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "O campo [Talhão (singular)] é
obrigatório. O campo [Talhão (plural)] é obrigatório.";
}

package perfil.geral.configuracoes.configuracaoGeral;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;
import
perfil.geralInterno.configuracoes.configuracaoGeralInterno.ElementsConfigur
acaoGeralInterno;

import util.ElementsRastreador;
import util.Util;

public class TestConfiguracaoGeral {
      WebDriver driver;

      @BeforeClass(alwaysRun = true)
      public void setUpConfiguracaoGeral() {
            Util.startTestMsg(TestConfiguracaoGeral.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CONFIGURACAO_GERAL);
      }
```

```
	@Test
	public void setConfiguracoesGerais() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsConfi
guracaoGeral.FORM_TALHAO_SINGULAR));


driver.findElement(ElementsConfiguracaoGeral.FORM_TALHAO_SINGULAR).clear();


driver.findElement(ElementsConfiguracaoGeral.FORM_TALHAO_SINGULAR).sendKeys
(Padroes.TALHAO);


driver.findElement(ElementsConfiguracaoGeral.FORM_TALHAO_PLURAL).clear();


driver.findElement(ElementsConfiguracaoGeral.FORM_TALHAO_PLURAL).sendKeys(P
adroes.TALHOES);


driver.findElement(ElementsConfiguracaoGeral.FORM_PRECO_COMPRA_SIM).click()
;


driver.findElement(ElementsConfiguracaoGeral.FORM_INSPECAO_RECEBIMENTO_SIM)
.click();


driver.findElement(ElementsConfiguracaoGeral.FORM_REALIZA_CONTROLE_SIM).cli
ck();


driver.findElement(ElementsConfiguracaoGeral.FORM_CHECK_TODOS).click();
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {
```

```
driver.findElement(ElementsConfiguracaoGeralInterno.FORM_SALVAR).click();
                        return
driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).isDisplayed();
                  }
            });

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDownConfiguracaoGeral() {
            Util.fechaGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CONFIGURACAO_GERAL);
            Util.endTestMsg(TestConfiguracaoGeral.class.getSimpleName());
      }

}
package perfil.geral.configuracoes.gerenciamentoBiblioteca;

import org.openqa.selenium.By;

public class ElementsGerenciamentoBiblioteca {

      public static final String TABLE =
"startForm:telaGerenciamentoBibliotecadataTableDocumentos:";
      public static final By TABLE_TITLE =
By.id("startForm:telaGerenciamentoBibliotecabasepanelotTitle");
      public static final String TABLE_VALUE =
":telaGerenciamentoBibliotecadataTableDocumentoscolumn0otValue0";
      public static final By TABLE_ORDER = By
                  .id("startForm:telaGerenciamentoBibliotecadataTableDocume
ntos:telaGerenciamentoBibliotecadataTableDocumentoscolumn0otHeader0");
      public static final By FORM_NOME =
By.id("startForm:telaGerenciamentoBibliotecapnForminputNomeTela");
      public static final By FORM_LINGUAGEM =
By.id("startForm:telaGerenciamentoBibliotecapnFormcbLinguagem");
      public static final By FORM_ADICIONAR_DOCUMENTO =
By.id("startForm:telaGerenciamentoBibliotecapnFormuploadDoc:file");
      public static final By FORM_ADICIONAR =
By.id("startForm:btnRealizaUploadArquivoBibliotecabt");
```

```
	public static final By FORM_SALVAR =
By.id("startForm:telaGerenciamentoBibliotecabasepanelcontentformbtsalvarbt"
);
	public static final By FORM_SUCESSO_OK =
By.id("formModalPanel:pnMsgSucessoCadastrobtYesbt");
	public static final By SPAN_MSG_SUCESSO = By
			.id("startForm:telaGerenciamentoBibliotecabasepanelconten
tpnFormpnSucessoUploadotMsgSucesso");
	public static final By ADICIONAR_NOVO = By
			.id("startForm:telaGerenciamentoBibliotecabasepanelconten
tlistgiNovoRegistro");
	public static final By MSG_UPLOAD_SUCESSO =
By.className("sucessoUploadArquivo");
	public static final By FORM_AGRUPADOR = By
			.id("startForm:telaGerenciamentoBibliotecapnFormcomboAgru
padorescomboboxField");
	public static final By TABELA_AGRUPADOR_PRIMEIRO_REGISTRO = By
			.id("startForm:telaGerenciamentoBibliotecadataTableAgrupa
dores:0:telaGerenciamentoBibliotecadataTableAgrupadorescolumn0otValue0");
}

package perfil.geral.configuracoes.gerenciamentoBiblioteca;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsFiles;
import util.ElementsRastreador;
import util.Util;

public class TestGerenciamentoBiblioteca {
	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpGerenciamentoBiblioteca() {
		driver = Util.driver;

	Util.startTestMsg(TestGerenciamentoBiblioteca.class.getSimpleName());
```

```
            Util.abreGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.GERENCIAMENTO_BIBLIOTECA);
      }

      @Test(groups = { "insert" })
      public void insertUniqueGerenciamentoBiblioteca() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsGeren
ciamentoBiblioteca.ADICIONAR_NOVO));


driver.findElement(ElementsGerenciamentoBiblioteca.ADICIONAR_NOVO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsGeren
ciamentoBiblioteca.FORM_NOME));


driver.findElement(ElementsGerenciamentoBiblioteca.FORM_NOME).sendKeys("_Bi
blioteca WebDriver");

      Util.selectOption(ElementsGerenciamentoBiblioteca.FORM_LINGUAGEM,
"Português");


Util.selectFile(ElementsGerenciamentoBiblioteca.FORM_ADICIONAR_DOCUMENTO,
ElementsFiles.DOC_PARIPASSU_PDF);


Util.wait.until(ExpectedConditions.textToBePresentInElement(ElementsGerenci
amentoBiblioteca.MSG_UPLOAD_SUCESSO,
                        "Sucesso"));



driver.findElement(ElementsGerenciamentoBiblioteca.FORM_AGRUPADOR).sendKeys
(Padroes.NOME_GPA);


driver.findElement(ElementsGerenciamentoBiblioteca.FORM_ADICIONAR).click();
```

```
            Util.wait.until(ExpectedConditions.textToBePresentInElement(

      ElementsGerenciamentoBiblioteca.TABELA_AGRUPADOR_PRIMEIRO_REGISTRO,
Padroes.NOME_GPA));



driver.findElement(ElementsGerenciamentoBiblioteca.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));


      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsGeren
ciamentoBiblioteca.ADICIONAR_NOVO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueGerenciamentoBiblioteca() {
            Util.acionaElemento(ElementsGerenciamentoBiblioteca.TABLE,
ElementsGerenciamentoBiblioteca.TABLE_VALUE,
                        Padroes.NOME_BIBLIOTECA,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.GERENCIAMENTO_BIBLIOTECA);

      Util.endTestMsg(TestGerenciamentoBiblioteca.class.getSimpleName());
      }

}
package perfil.geral.configuracoes.identidadeVisual;
```

```
import org.openqa.selenium.By;

public class ElementsIdentidadeVisual {

    //Elementos da Tela Principal
    public static final By FORM_TEMA =
By.id("startForm:telaLogotipoSistemaselectSkin");
    public static final By FORM_LOGOTIPO_ADICIONAR =
By.id("startForm:telaLogotipoSistemafuLogotipo:file");
    public static final By MSG_SUCESSO_UPLOAD_LOGOTIPO =
By.id("startForm:telaLogotipoSistemamsgSucesso");
    public static final By MSG_SUCESSO_UPLOAD_TEMA =
By.id("startForm:telaLogotipoSistemamsgTrocaSkin");
}

package perfil.geral.configuracoes.identidadeVisual;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.annotations.*;

import util.Configs;
import util.ElementsFiles;
import util.ElementsRastreador;
import util.Util;
import util.Util.BrowserType;

public class TestIdentidadeVisual {
      WebDriver driver;

      @BeforeClass(alwaysRun = true)
      public void setUpIdentidadeVisual() {
            Util.startTestMsg(TestIdentidadeVisual.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.IDENTIDADE_VISUAL);
      }

      @Test(groups = { "config " })
      public void setIdentidadeVisual() {
            BrowserType browserType = Configs.browser;
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsIdent
idadeVisual.FORM_TEMA));
            Util.selectOption(ElementsIdentidadeVisual.FORM_TEMA, "Roxo");


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsIdent
idadeVisual.MSG_SUCESSO_UPLOAD_TEMA));

            if (browserType == Util.BrowserType.FIREFOX) {

      Util.selectFile(ElementsIdentidadeVisual.FORM_LOGOTIPO_ADICIONAR,
ElementsFiles.LOGO_PARIPASSU_PNG);
            }
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.IDENTIDADE_VISUAL);
            Util.endTestMsg(TestIdentidadeVisual.class.getSimpleName());
      }
}

package perfil.geral.embalagens.embalagem;

import org.openqa.selenium.By;

public class ElementsEmbalagem {

    // Elementos da Tela Principal - Botões
    public static final By NOVO_REGISTRO = By
          .id("startForm:telaEmbalagembasepanelcontentlistgiNovoRegistro");

    // Elementos da Tela Principal - Tabela
    public static final String TABLE = "startForm:telaEmbalagemdataTable:";
    public static final By TABLE_ORDER = By

.id("startForm:telaEmbalagemdataTable:telaEmbalagemdataTablecolumn0header:s
```

```
ortDiv");
    public static final String TABLE_NOME =
":telaEmbalagemdataTablecolumn0otValue0";
    public static final String TABLE_CAPACIDADE =
":telaEmbalagemdataTablecolumn2otValue2";

    // Elementos da Tela Principal - Formulário
    public static final By FORM_NOME =
By.id("startForm:telaEmbalagemnome");
    public static final By FORM_DESCRICAO = By
          .id("startForm:telaEmbalagemdescricao");
    public static final By FORM_TIPO_EMBALAGEM = By
          .id("startForm:telaEmbalagempnForm1tipoEmbalagem");
    public static final By FORM_CAPACIDADE = By
          .id("startForm:telaEmbalagemcapacidade");
    public static final By FORM_UNIDADE_MEDIDA = By
          .id("startForm:telaEmbalagempnForm1unidade");
    public static final By FORM_SALVAR = By
          .id("startForm:telaEmbalagembasepanelcontentformbtsalvarbt");

    public static final By BODY = By
          .id("startForm:telaEmbalagembasepanel_body");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Nome] é obrigatório." + "\n"
          + "O campo [Capacidade] é obrigatório." + "\n"
          + "O campo [Unidade de Medida] é obrigatório." + "\n"
          + "O campo [Tipo Embalagem] é obrigatório.";
}

package perfil.geral.embalagens.embalagem;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
```

```
import util.MsgErro;
import util.Util;

public class TestEmbalagem {
	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpTipoEmbalagem() {
		Util.startTestMsg(TestEmbalagem.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.EMBALAGENS,
ElementsRastreador.EMBALAGEM);
	}

	@Test(groups = { "insert" })
	public void setUpInsertEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.NOVO_REGISTRO));
		driver.findElement(ElementsEmbalagem.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"setUpInsertEmbalagem")
	public void insertUniqueEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.FORM_NOME));


driver.findElement(ElementsEmbalagem.FORM_NOME).sendKeys(Padroes.NOME_EMBAL
AGEM);


driver.findElement(ElementsEmbalagem.FORM_DESCRICAO).sendKeys("Descricao "
+ Padroes.NOME_EMBALAGEM);
		Util.selectOption(ElementsEmbalagem.FORM_TIPO_EMBALAGEM,
Padroes.NOME_TIPO_EMBALAGEM);
```

```
      driver.findElement(ElementsEmbalagem.FORM_CAPACIDADE).sendKeys("1");
            Util.selectOption(ElementsEmbalagem.FORM_UNIDADE_MEDIDA, "kg");
            driver.findElement(ElementsEmbalagem.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.NOVO_REGISTRO));
            Util.acionaElemento(ElementsEmbalagem.TABLE,
ElementsEmbalagem.TABLE_NOME, Padroes.NOME_EMBALAGEM,
                        ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.FORM_NOME));


driver.findElement(ElementsEmbalagem.FORM_NOME).sendKeys(Padroes.EDITADO);


driver.findElement(ElementsEmbalagem.FORM_DESCRICAO).sendKeys(Padroes.EDITA
DO);
            driver.findElement(ElementsEmbalagem.FORM_CAPACIDADE).clear();

      driver.findElement(ElementsEmbalagem.FORM_CAPACIDADE).sendKeys("2");
            Util.selectOption(ElementsEmbalagem.FORM_UNIDADE_MEDIDA, "un");
            driver.findElement(ElementsEmbalagem.FORM_SALVAR).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.NOVO_REGISTRO));
            Util.acionaElemento(ElementsEmbalagem.TABLE,
ElementsEmbalagem.TABLE_NOME, Padroes.NOME_EMBALAGEM
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.NOVO_REGISTRO));

            driver.findElement(ElementsEmbalagem.NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.FORM_SALVAR));

            driver.findElement(ElementsEmbalagem.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
```

```
eador.MODAL_ERRO_TITLE));


Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsEmbalagem.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @Test(groups = { "verify", "routine" })
      public void verifyEmbalagem() {
            String valorEmbalagem = Util.getRandomValue(0, 1, 2);
            if (valorEmbalagem.equals("0.00")) {
                  valorEmbalagem = "0.01";
            }

            Util.acionaElemento(ElementsEmbalagem.TABLE,
ElementsEmbalagem.TABLE_NOME, Padroes.NOME_EMBALAGEM_QUALIDADE,
                        ElementsRastreador.TABLE_EDITAR);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
agem.FORM_CAPACIDADE));
            driver.findElement(ElementsEmbalagem.FORM_CAPACIDADE).clear();


driver.findElement(ElementsEmbalagem.FORM_CAPACIDADE).sendKeys(valorEmbalag
em);
            driver.findElement(ElementsEmbalagem.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEmbal
```

```
agem.NOVO_REGISTRO));

            String strValorCapacidade =
Util.retornaValorTabela(ElementsEmbalagem.TABLE,
ElementsEmbalagem.TABLE_NOME,
                        ElementsEmbalagem.TABLE_CAPACIDADE,
Padroes.NOME_EMBALAGEM_QUALIDADE);
            Double valorCapacidade =
Double.parseDouble(strValorCapacidade);

            Assert.assertEquals(String.format("%1$.2f",
valorCapacidade).replace(",", "."), valorEmbalagem,
                        MsgErro.FAIL_VALOR_DIFERENTE);
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.EMBALAGENS,
ElementsRastreador.EMBALAGEM);
            Util.endTestMsg(TestEmbalagem.class.getSimpleName());
      }
}

package perfil.geral.embalagens.tipoEmbalagem;

import org.openqa.selenium.By;

public class ElementsTipoEmbalagem {

      // Elementos da tela principal - Botões
      public static final By NOVO_REGISTRO =
By.id("startForm:telaTipoEmbalagembasepanelcontentlistgiNovoRegistro");

      // Elementos da tela principal - Tabela
      public static final String TABLE =
"startForm:telaTipoEmbalagemdataTable:";
      public static final By TABLE_ORDER = By
                  .id("startForm:telaTipoEmbalagemdataTable:telaTipoEmbalag
emdataTablecolumn0header:sortDiv");
      public static final String TABLE_VALUE =
":telaTipoEmbalagemdataTablecolumn0otValue0";
```

```
	// Elementos do formulário
	public static final By FORM_NOME =
By.id("startForm:telaTipoEmbalagemnome");
	public static final By FORM_DESCRICAO =
By.id("startForm:telaTipoEmbalagemdescricao");
	public static final By FORM_SALVAR =
By.id("startForm:telaTipoEmbalagembasepanelcontentformbtsalvarbt");

	public static final By BODY =
By.id("startForm:telaTipoEmbalagembasepanel_body");

	// Mensagem de validação dos campos obrigatorios
	public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
[Nome] é obrigatório.";

}

package perfil.geral.embalagens.tipoEmbalagem;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestTipoEmbalagem {

	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpEmbalagem() {
		Util.startTestMsg(TestTipoEmbalagem.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.EMBALAGENS,
ElementsRastreador.TIPO_EMBALAGEM);
```

```
	}

	@Test(groups = { "insert" })
	public void setUpInsertTipoEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
mbalagem.NOVO_REGISTRO));

	driver.findElement(ElementsTipoEmbalagem.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"setUpInsertTipoEmbalagem")
	public void insertUniqueTipoEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
mbalagem.FORM_NOME));


driver.findElement(ElementsTipoEmbalagem.FORM_NOME).sendKeys(Padroes.NOME_T
IPO_EMBALAGEM);


driver.findElement(ElementsTipoEmbalagem.FORM_DESCRICAO).sendKeys("Descrica
o " + Padroes.NOME_TIPO_EMBALAGEM);
		driver.findElement(ElementsTipoEmbalagem.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

	driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();
	}

	@Test(groups = { "edit" })
	public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
```

```
mbalagem.NOVO_REGISTRO));
            Util.acionaElemento(ElementsTipoEmbalagem.TABLE,
ElementsTipoEmbalagem.TABLE_VALUE, Padroes.NOME_TIPO_EMBALAGEM,
                        ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueTipoEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
mbalagem.FORM_NOME));


driver.findElement(ElementsTipoEmbalagem.FORM_NOME).sendKeys(Padroes.EDITAD
O);


driver.findElement(ElementsTipoEmbalagem.FORM_DESCRICAO).sendKeys(Padroes.E
DITADO);
            driver.findElement(ElementsTipoEmbalagem.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
mbalagem.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueTipoEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
mbalagem.NOVO_REGISTRO));
            Util.acionaElemento(ElementsTipoEmbalagem.TABLE,
ElementsTipoEmbalagem.TABLE_VALUE, Padroes.NOME_TIPO_EMBALAGEM
```

```
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosTipoEmbalagem() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
mbalagem.NOVO_REGISTRO));


      driver.findElement(ElementsTipoEmbalagem.NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsTipoE
mbalagem.FORM_SALVAR));

            driver.findElement(ElementsTipoEmbalagem.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsTipoEmbalagem.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.EMBALAGENS,
ElementsRastreador.TIPO_EMBALAGEM);
            Util.endTestMsg(TestTipoEmbalagem.class.getSimpleName());
      }
```

```
}

package perfil.geral.indicadores.biblioteca;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.Util;

public class TestBiblioteca {
	WebDriver driver;

	@BeforeClass(alwaysRun = true)
	public void setUpBiblioteca() {
		Util.startTestMsg(TestBiblioteca.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.INDICADORES,
ElementsRastreador.BIBLIOTECA);
	}

	@Test(groups = { "generate" })
	public void openBiblioteca() {
		String handleJanelaPrincipal = driver.getWindowHandle();
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver ecDriver) {
				Util.acionaElemento(ElementsBiblioteca.TABLE,
ElementsBiblioteca.TABLE_LINK, Padroes.NOME_BIBLIOTECA,
						ElementsBiblioteca.TABLE_LINK);
				return (2 == driver.getWindowHandles().size());
			}
		});

		for (String winHandle : driver.getWindowHandles()) {
			if (!winHandle.equals(handleJanelaPrincipal)) {
				driver.switchTo().window(winHandle);
```

```
                  }
            }
            Util.testMsg(TestBiblioteca.class.getSimpleName(),
driver.getCurrentUrl() + " / FILE");
            driver.close();
            driver.switchTo().window(handleJanelaPrincipal);
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.INDICADORES,
ElementsRastreador.BIBLIOTECA);
            Util.endTestMsg(TestBiblioteca.class.getSimpleName());
      }
}

package perfil.geral.indicadores.biblioteca;


public class ElementsBiblioteca {

    //Elementos da Tela Principal - Tabela
    public static final String TABLE =
"startForm:telaBibliotecaArquivosdataTableDocumentos:";
    public static final String TABLE_VALUE =
":telaBibliotecaArquivosdataTableDocumentoscolumn0otValue0";
    public static final String TABLE_LINK = ":textoLink";
}

package perfil.geral.indicadores.relatorios;

import org.openqa.selenium.By;

public class ElementsRelatorios {

    // Elementos da Tela Principal
    public static final By FORM_TIPO_RELATORIO = By
          .id("startForm:telaRelatorios3GNovatipoRelatorio");
    public static final By FORM_DT_INICIO = By

.id("startForm:telaRelatorios3GNovapnFormpnDataInicialcalDataInicioInputDat
```

```
e");
    public static final By FORM_DT_FIM = By

.id("startForm:telaRelatorios3GNovapnFormpnDataFinalcalDataFimInputDate");
    public static final By FORM_DESTINO_TODOS = By
          .id("startForm:telaRelatorios3GNovapnEloTodosCk");
    public static final By FORM_PRODUTO_TODOS = By
          .id("startForm:telaRelatorios3GNovapnTipoProdutoTodosCk");
    public static final By FORM_PDF = By
          .id("startForm:telaRelatorios3GNovaimgPdf");
    public static final By FORM_EXCEL = By
          .id("startForm:telaRelatorios3GNovaimgExcel");
    public static final By FORM_GRAFICO = By
          .id("startForm:telaRelatorios3GNovaimgGrafico");
    public static final By FORM_COD_RASTREIO = By
          .id("startForm:telaRelatorios3GNovapnFormcodigoRastreamento");
    public static final By FORM_ORIGEM = By
          .id("startForm:telaRelatorios3GNovaorigem");
    public static final By FORM_VOLTAR = By
          .id("startForm:telaRelatorios3GNovabtChartVoltarbt");
    public static final By FORM_DESTINO =
By.id("startForm:telaRelatorios3GNovapnElo42621ck");
}

package perfil.geral.indicadores.relatorios;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.*;

import perfil.geral.rastreamento.envioDireto.TestEnvioDireto;
import perfil.geral.util.Padroes;

import util.Configs;
import util.ElementsRastreador;
import util.Util;

public class TestRelatorios {
      WebDriver driver;
```

```
	String dataAtual = Util.dataAtual();
	WebDriverWait waitRelatorio = null;

	public void preencheRelatorio() {


driver.findElement(ElementsRelatorios.FORM_DT_INICIO).sendKeys("01012012");


driver.findElement(ElementsRelatorios.FORM_DT_FIM).sendKeys(Util.dataAtual(
));

	driver.findElement(ElementsRelatorios.FORM_DESTINO_TODOS).click();

	driver.findElement(ElementsRelatorios.FORM_PRODUTO_TODOS).click();
	}

	public void clicaPdf() {
		String originalHandle = driver.getWindowHandle();
		driver.findElement(ElementsRelatorios.FORM_PDF).click();
		for (String winHandle : driver.getWindowHandles()) {
			if (!winHandle.equals(originalHandle)) {
				driver.switchTo().window(winHandle);
			}
		}
		System.out.println(driver.getCurrentUrl());
		driver.close();
		driver.switchTo().window(originalHandle);
	}

	@BeforeClass(alwaysRun = true)
	public void setUpRelatorios() {
		Util.startTestMsg(TestRelatorios.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.INDICADORES,
ElementsRastreador.RELATORIOS);
	}

	@Test(groups = { "generate" })
	public void generateRelatorios() throws InterruptedException {
```

```
            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Relatório de Produtos Enviados");
            preencheRelatorio();
            clicaPdf();
            Util.testMsg("Relatorio", "1 de 12 - Relatório de Produtos
Enviados");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Romaneio de saída de cargas");
            preencheRelatorio();
            clicaPdf();
            Util.testMsg("Relatorio", "2 de 12 - Romaneio de saída de
cargas");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Lista de Defensivos Utilizados");

      driver.findElement(ElementsRelatorios.FORM_DESTINO_TODOS).click();

      driver.findElement(ElementsRelatorios.FORM_PRODUTO_TODOS).click();
            clicaPdf();
            Util.testMsg("Relatorio", "3 de 12 - Lista de Defensivos
Utilizados");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Romaneio Padrão");
            clicaPdf();
            Util.testMsg("Relatorio", "4 de 12 - Romaneio Padrão");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Relatório de Produtos Recebidos");
            preencheRelatorio();
            clicaPdf();
            Util.testMsg("Relatorio", "5 de 12 - Relatório de Produtos
Recebidos");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Relatório de Estoque");

      driver.findElement(ElementsRelatorios.FORM_DESTINO_TODOS).click();
```

```
      driver.findElement(ElementsRelatorios.FORM_PRODUTO_TODOS).click();
            clicaPdf();
            Util.testMsg("Relatorio", "6 de 12 - Relatório de Estoque");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Histórico de inspeção de recebimento de produtos");


driver.findElement(ElementsRelatorios.FORM_DT_INICIO).sendKeys(Util.dataAtu
al());


driver.findElement(ElementsRelatorios.FORM_DT_FIM).sendKeys(Util.dataAtual(
));
            selecionaLocalRecebimento(Padroes.NOME_ORIGEM);

      driver.findElement(ElementsRelatorios.FORM_PRODUTO_TODOS).click();

            driver.findElement(ElementsRelatorios.FORM_GRAFICO).click();
            waitRelatorio = new WebDriverWait(driver,
Configs.MEGA_TIMEOUT);


waitRelatorio.until(ExpectedConditions.visibilityOfElementLocated(ElementsR
elatorios.FORM_VOLTAR));
            driver.findElement(ElementsRelatorios.FORM_VOLTAR).click();
            Util.testMsg("Relatorio", "7 de 12 - Histórico de inspeção de
recebimento de produtos");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Comparativo de inspeção de recebimento de produtos");


driver.findElement(ElementsRelatorios.FORM_DT_INICIO).sendKeys(Util.dataAtu
al());


driver.findElement(ElementsRelatorios.FORM_DT_FIM).sendKeys(Util.dataAtual(
));
            selecionaLocalRecebimento(Padroes.NOME_ORIGEM);
```

```
      driver.findElement(ElementsRelatorios.FORM_PRODUTO_TODOS).click();

            driver.findElement(ElementsRelatorios.FORM_GRAFICO).click();
            waitRelatorio = new WebDriverWait(driver,
Configs.MEGA_TIMEOUT);


waitRelatorio.until(ExpectedConditions.visibilityOfElementLocated(ElementsR
elatorios.FORM_VOLTAR));
            driver.findElement(ElementsRelatorios.FORM_VOLTAR).click();
            Util.testMsg("Relatorio", "8 de 12 - Comparativo de inspeção de
recebimento de produtos");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Relatório de Movimentação de Produto");


driver.findElement(ElementsRelatorios.FORM_COD_RASTREIO).sendKeys(TestEnvio
Direto.codigoEnvio);
            clicaPdf();
            Util.testMsg("Relatorio", "9 de 12 - Relatório de Movimentação
de Produto");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Relatório de Descartes");
            preencheRelatorio();
            clicaPdf();
            Util.testMsg("Relatorio", "10 de 12 - Relatório de Descartes");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Relatório de Origem");
            Util.selectOption(ElementsRelatorios.FORM_ORIGEM,
Padroes.NOME_ORIGEM);
            clicaPdf();
            Util.testMsg("Relatorio", "11 de 12 - Relatório de Origem");

            Util.selectOption(ElementsRelatorios.FORM_TIPO_RELATORIO,
"Relatório de Rendimento");
            preencheRelatorio();
            clicaPdf();
            Util.testMsg("Relatorio", "12 de12 - Relatório de Rendimento");
```

```
	}

	private void selecionaLocalRecebimento(String localRecebimento) {
		By elementSpanLocalRecebimento = null;
		String elementSpanLocalRecebimentoFullAddr = "";
		String elementSpanLocalRecebimentoPre = "";
		String elementSpanLocalRecebimentoSuf = "";
		String idOtLocalRecebimento = "";
		String idCkLocalRecebimento = "";
		boolean notFound = true;
		int index = 2;

		if (Configs.browser == Util.BrowserType.IE) {
			elementSpanLocalRecebimentoPre =
"//form[2]/div[2]/table/tbody/tr/td[2]/div[1]/div/div/span/div/div/div[2]/d
iv/div/div/div[2]/div/div[4]/div/div[";
			elementSpanLocalRecebimentoSuf = "]/div/span";
		} else {
			elementSpanLocalRecebimentoPre = "html > body > form:nth-
of-type(2) > div:nth-of-type(2) > table > tbody > tr > td:nth-of-type(2) >
div:nth-of-type(1) > div > div > span > div > div > div:nth-of-type(2) >
div > div > div > div:nth-of-type(2) > div > div:nth-of-type(4) > div >
div:nth-of-type(";
			elementSpanLocalRecebimentoSuf = ") > div > span";
		}

		while (notFound) {
			elementSpanLocalRecebimentoFullAddr =
elementSpanLocalRecebimentoPre + index + elementSpanLocalRecebimentoSuf;
			if (Configs.browser == Util.BrowserType.IE) {
				elementSpanLocalRecebimento =
By.xpath(elementSpanLocalRecebimentoFullAddr);
			} else {
				elementSpanLocalRecebimento =
By.cssSelector(elementSpanLocalRecebimentoFullAddr);
			}

			if
(driver.findElement(elementSpanLocalRecebimento).getText().equals(localRece
bimento)) {
				idOtLocalRecebimento =
```

```
driver.findElement(elementSpanLocalRecebimento).getAttribute("id");
                        break;
                  }
                  index++;
            }

            idCkLocalRecebimento = idOtLocalRecebimento.substring(0,
(idOtLocalRecebimento.length() - 2)) + "ck";
            if (!
(driver.findElement(By.id(idCkLocalRecebimento)).isSelected())) {
                  driver.findElement(By.id(idCkLocalRecebimento)).click();
            }
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.INDICADORES,
ElementsRastreador.RELATORIOS);
            Util.endTestMsg(TestRelatorios.class.getSimpleName());
      }
}

package perfil.geral.questionario.gerenciamentoQuestionario;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestGerenciamentoQuestionario {
      WebDriver driver;

      @BeforeClass(alwaysRun = true)
      public void setUpGerenciamentoQuestionario() {
```

```
Util.startTestMsg(TestGerenciamentoQuestionario.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.QUESTIONARIO,
ElementsRastreador.GERENCIAMENTO_QUESTIONARIO);
      }

      @Test(groups = { "insert" })
      public void insertUniqueGerenciamentoQuestionario() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsGeren
ciamentoQuestionario.FORM_TIPO_PRODUTO));


driver.findElement(ElementsGerenciamentoQuestionario.FORM_TIPO_PRODUTO).sen
dKeys(
                        Padroes.NOME_PRODUTO_QUESTIONARIO);


driver.findElement(ElementsGerenciamentoQuestionario.FORM_PAIS).sendKeys("B
rasil");


driver.findElement(ElementsGerenciamentoQuestionario.FORM_TIPO_QUESTIONARIO
).sendKeys(Padroes.NOME_QUESTIONARIO);


driver.findElement(ElementsGerenciamentoQuestionario.FORM_ADICIONAR).click(
);
      }

      @Test(groups = { "delete" })
      public void deleteUniqueGerenciamentoQuestionario() {
            Util.acionaElemento(ElementsGerenciamentoQuestionario.TABLE,
ElementsGerenciamentoQuestionario.TABLE_VALUE,
                        Padroes.NOME_QUESTIONARIO,
ElementsGerenciamentoQuestionario.TABLE_EXCLUIR);
      }
```

```
	@Test(groups = { "verify", "campos" })
	public void validacaoCamposObrigatoriosGerenciamentoQuestionario() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsGeren
ciamentoQuestionario.FORM_ADICIONAR));
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsGerenciamentoQuestionario.FORM_ADICIONAR).click(
);
				return
driver.findElement(ElementsGerenciamentoQuestionario.MODAL_ERRO).isDisplaye
d();
			}
		});



Assert.assertEquals(driver.findElement(ElementsGerenciamentoQuestionario.MO
DAL_ERRO).getText(),
				ElementsGerenciamentoQuestionario.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);
	}

	@AfterClass(alwaysRun = true)
	public void tearDown() {
		Util.fechaGrupoMenu(ElementsRastreador.QUESTIONARIO,
ElementsRastreador.GERENCIAMENTO_QUESTIONARIO);

	Util.endTestMsg(TestGerenciamentoQuestionario.class.getSimpleName());
	}
}
```

```
package perfil.geral.questionario.gerenciamentoQuestionario;

import org.openqa.selenium.By;

public class ElementsGerenciamentoQuestionario {
```

```
	// Elementos da Tela Principal - Tabela
	public static final String TABLE =
"startForm:telaGerenciamentoQuestionariosPepsicotabelaConfiguracao:";
	public static final String TABLE_VALUE =
":telaGerenciamentoQuestionariosPepsicotabelaConfiguracaocolumn2otValue2";
	public static final By TABLE_ORDER = By
			.id("startForm:telaGerenciamentoQuestionariosPepsicotabel
aConfiguracao:telaGerenciamentoQuestionariosPepsicotabelaConfiguracaocolumn
2otHeader2");
	public static final String TABLE_EXCLUIR =
":telaGerenciamentoQuestionariosPepsicotabelaConfiguracaopnEditDeletegiDele
te";

	// Elementos Tela Principal
	public static final By FORM_TIPO_PRODUTO = By
			.id("startForm:telaGerenciamentoQuestionariosPepsicobasep
anelcontentpnFormpncombocomboboxField");
	public static final By FORM_TIPO_PRODUTO_SELECIONAR = By
			.xpath("//div[@id='startForm:telaGerenciamentoQuestionari
osPepsicobasepanelcontentpnFormpncombolist']/span[1]");
	public static final By FORM_PAIS = By
			.id("startForm:telaGerenciamentoQuestionariosPepsicobasep
anelcontentpnFormpncomboPaiscomboboxField");
	public static final By FORM_PAIS_SELECIONAR = By
			.xpath("//div[@id='startForm:telaGerenciamentoQuestionari
osPepsicobasepanelcontentpnFormpncomboPaislist']/span[1]");
	public static final By FORM_TIPO_QUESTIONARIO = By
			.id("startForm:telaGerenciamentoQuestionariosPepsicobasep
anelcontentpnFormpncomboTipoQuestionariocomboboxField");
	public static final By FORM_TIPO_QUESTIONARIO_SELECIONAR = By
			.xpath("//div[@id='startForm:telaGerenciamentoQuestionari
osPepsicobasepanelcontentpnFormpncomboTipoQuestionariolist']/span[1]");
	public static final By FORM_ADICIONAR =
By.id("startForm:telaGerenciamentoQuestionariosPepsicobtadicionarbt");
	public static final By MODAL_ERRO = By
			.id("startForm:telaGerenciamentoQuestionariosPepsicobasep
anelcontentpnFormotMsgErro");

	// Mensagem de validação dos campos obrigatorios
	public static String MSG_VALIDA_CAMPOS = "Preencha todos os campos.";
```

```
}

package perfil.geral.questionario.aplicacaoQuestionario;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestAplicacaoQuestionario {
	WebDriver driver;
	String dataAtual = Util.dataAtual();

	@BeforeClass(alwaysRun = true)
	public void setUpAplicacaoQuestionario() {

	Util.startTestMsg(TestAplicacaoQuestionario.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.QUESTIONARIO,
ElementsRastreador.APLICACAO_QUESTIONARIO);
	}

	@Test(groups = { "insert" })
	public void setUpInsertAplicacaoQuestionario() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.NOVO_REGISTRO));


driver.findElement(ElementsAplicacaoQuestionario.NOVO_REGISTRO).click();
```

```
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"setUpInsertAplicacaoQuestionario")
	public void insertUniqueAplicacaoQuestionario() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.FORM_TIPO_QUESTIONARIO));


Util.selectOption(ElementsAplicacaoQuestionario.FORM_TIPO_QUESTIONARIO,
Padroes.NOME_QUESTIONARIO);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.FORM_QUESTOES_QUESTIONARIO));
		Util.selectOption(ElementsAplicacaoQuestionario.FORM_AVALIADO,
Padroes.NOME_AVALIADO);
		Util.waitJsBlock();


driver.findElement(ElementsAplicacaoQuestionario.FORM_DATA).sendKeys(dataAt
ual);


driver.findElement(ElementsAplicacaoQuestionario.FORM_HORA_INICIAL).sendKey
s("13:00");


driver.findElement(ElementsAplicacaoQuestionario.FORM_HORA_FINAL).sendKeys(
"15:00");


driver.findElement(ElementsAplicacaoQuestionario.FORM_CHECKLIST_1).sendKeys
("_WebDriver");


driver.findElement(ElementsAplicacaoQuestionario.FORM_CHECKLIST_2).sendKeys
("0123456789");
```

```
driver.findElement(ElementsAplicacaoQuestionario.FORM_CHECKLIST_3).sendKeys
(dataAtual);

            List<WebElement> headerTopicos =
driver.findElements(By.className("rich-panelbar-header"));
            for (WebElement headerTopico : headerTopicos) {
                  if (headerTopico.getText().equals("Tópico 2")) {
                        headerTopico.click();
                        break;
                  }
            }

      Util.selectOption(ElementsAplicacaoQuestionario.FORM_CHECKLIST_4,
Padroes.SIM);



driver.findElement(ElementsAplicacaoQuestionario.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.NOVO_REGISTRO));
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.NOVO_REGISTRO));
            Util.acionaElemento(ElementsAplicacaoQuestionario.TABLE,
ElementsAplicacaoQuestionario.TABLE_VALUE,
                        Padroes.NOME_QUESTIONARIO,
ElementsRastreador.TABLE_EDITAR);
```

```
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
	public void editUniqueAplicacaoQuestionario() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.FORM_DATA));

	driver.findElement(ElementsAplicacaoQuestionario.FORM_DATA).clear();


driver.findElement(ElementsAplicacaoQuestionario.FORM_DATA).sendKeys(dataAt
ual);


driver.findElement(ElementsAplicacaoQuestionario.FORM_HORA_INICIAL).clear()
;


driver.findElement(ElementsAplicacaoQuestionario.FORM_HORA_INICIAL).sendKey
s("16:00");


driver.findElement(ElementsAplicacaoQuestionario.FORM_HORA_FINAL).clear();


driver.findElement(ElementsAplicacaoQuestionario.FORM_HORA_FINAL).sendKeys(
"18:00");


driver.findElement(ElementsAplicacaoQuestionario.FORM_CHECKLIST_1).sendKeys
(Padroes.EDITADO);


driver.findElement(ElementsAplicacaoQuestionario.FORM_CHECKLIST_2).clear();


driver.findElement(ElementsAplicacaoQuestionario.FORM_CHECKLIST_2).sendKeys
("9876543210");
```

```
driver.findElement(ElementsAplicacaoQuestionario.FORM_CHECKLIST_3).sendKeys
(dataAtual);

            List<WebElement> headerTopicos =
driver.findElements(By.className("rich-panelbar-header"));
            for (WebElement headerTopico : headerTopicos) {
                  if (headerTopico.getText().equals("Tópico 2")) {
                        headerTopico.click();
                        break;
                  }
            }


      Util.selectOption(ElementsAplicacaoQuestionario.FORM_CHECKLIST_4,
Padroes.NAO);


driver.findElement(ElementsAplicacaoQuestionario.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueAplicacaoQuestionario() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.NOVO_REGISTRO));
            Util.acionaElemento(ElementsAplicacaoQuestionario.TABLE,
ElementsAplicacaoQuestionario.TABLE_VALUE,
                        Padroes.NOME_QUESTIONARIO,
ElementsRastreador.TABLE_EXCLUIR);
```

```
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosAplicacaoQuestionario() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.NOVO_REGISTRO));



driver.findElement(ElementsAplicacaoQuestionario.NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsAplic
acaoQuestionario.FORM_SALVAR));



driver.findElement(ElementsAplicacaoQuestionario.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsAplicacaoQuestionario.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.QUESTIONARIO,
ElementsRastreador.APLICACAO_QUESTIONARIO);
```

```
      Util.endTestMsg(TestAplicacaoQuestionario.class.getSimpleName());
      }
}

package perfil.geral.questionario.aplicacaoQuestionario;

import org.openqa.selenium.By;

public class ElementsAplicacaoQuestionario {

    // Elementos da Tela Principal - Botões
    public static final By NOVO_REGISTRO = By

.id("startForm:telaAplicacaoQuestionariobasepanelcontentlistgiNovoRegistro"
);

    // Elementos da Tela Principal - Tabela
    public static final String TABLE =
"startForm:telaAplicacaoQuestionariodataTable:";
    public static final String TABLE_VALUE =
":telaAplicacaoQuestionariodataTablecolumn0otValue0";

    // Elementos do Formulário
    public static final By FORM_TIPO_QUESTIONARIO = By
          .id("startForm:nulltipoQuestionario");
    public static final By FORM_AVALIADO =
By.id("startForm:nulltipoAvaliado");
    public static final By FORM_DATA =
By.id("startForm:nulldataInputDate");
    public static final By FORM_HORA_INICIAL = By
          .id("startForm:nullhraInicial");
    public static final By FORM_HORA_FINAL =
By.id("startForm:nullhraFinal");
    public static final By FORM_CHECKLIST_1 = By
          .id("startForm:cp_questao19166");
    public static final By FORM_CHECKLIST_2 = By
          .id("startForm:cp_questao19167");
    public static final By FORM_CHECKLIST_3 = By
          .id("startForm:cp_questao19168InputDate");
    public static final By FORM_CHECKLIST_4 = By
          .id("startForm:cp_questao19169");
```

```
    public static final By FORM_SALVAR = By

.id("startForm:telaAplicacaoQuestionariobasepanelcontentformbtsalvarbt");
      public static final By FORM_QUESTOES_QUESTIONARIO =
By.id("startForm:telaAplicacaoQuestionariobasepanelcontentformpnQuestoestab
Panel");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Tipo de Questionario] é obrigatório." + "\n"
          + "O campo [Avaliado] é obrigatório." + "\n"
          + "O campo [Data] é obrigatório." + "\n"
          + "O conteúdo do campo [Hora Inicial] é inválido." + "\n"
          + "O conteúdo do campo [Hora Final] é inválido.";
}

package util;

import org.testng.Assert;
import java.io.File;
import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Random;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.NoSuchElementException;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.remote.Augmenter;
import org.openqa.selenium.remote.CapabilityType;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
```

```
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

import perfil.geral.util.Padroes;

public class Util {

	public static WebDriver driver;
	public static WebDriverWait wait;

	public static void getWebDriver() throws MalformedURLException {
		startTestMsg("WebDriver");
		BrowserType browserType = Configs.browser;

		switch (browserType) {
		case FIREFOX:
			DesiredCapabilities firefoxCapabilities =
DesiredCapabilities.firefox();

	firefoxCapabilities.setCapability(CapabilityType.TAKES_SCREENSHOT,
true);
			driver = new RemoteWebDriver(new
URL(Configs.ENDERECO_HUB), firefoxCapabilities);
			driver.manage().window().maximize();
			break;
		case IE:
			DesiredCapabilities ieCapabilities =
DesiredCapabilities.internetExplorer();

	ieCapabilities.setCapability(CapabilityType.TAKES_SCREENSHOT, true);
			driver = new RemoteWebDriver(new
URL(Configs.ENDERECO_HUB), ieCapabilities);
			driver.manage().window().maximize();
			break;
		case CHROME:
			DesiredCapabilities chromeCapabilities =
DesiredCapabilities.chrome();
			chromeCapabilities.setCapability("chrome.switches",
					Arrays.asList("--start-maximized --disable-
popup-blocking"));
```

```
      chromeCapabilities.setCapability(CapabilityType.TAKES_SCREENSHOT,
true);
                  driver = new RemoteWebDriver(new
URL(Configs.ENDERECO_HUB), chromeCapabilities);
                  break;
            default:
                  throw new RuntimeException("Navegador não suportado.");
            }

            driver = new Augmenter().augment(driver);
            wait = new WebDriverWait(driver, Configs.INT_TIMEOUT);
      }

      public static void getRastreador(String nomeUsuario) {
            driver.get(Configs.URL_BASE);
            selectOption(By.id("selectLanguage"), "Português");
            driver.findElement(By.id("usuario")).sendKeys(nomeUsuario);

      driver.findElement(By.id("senha")).sendKeys(Padroes.SENHA_PADRAO);
            driver.findElement(By.id("botao")).click();
            getPortal();


wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastreador
.SELECT_LINGUA));
            selectOption(ElementsRastreador.SELECT_LINGUA, "Português");
      }

      public static void getRastreadorSenhaInterna(String nomeUsuario) {
            driver.get(Configs.URL_BASE);
            selectOption(By.id("selectLanguage"), "Português");
            driver.findElement(By.id("usuario")).sendKeys(nomeUsuario);


driver.findElement(By.id("senha")).sendKeys(Padroes.SENHA_USUARIO_INTERNO);
            driver.findElement(By.id("botao")).click();


wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastreador
.SELECT_LINGUA));
            selectOption(ElementsRastreador.SELECT_LINGUA, "Português");
```

```
	}

	public static void getPortal() {
		List<WebElement> formProdutos =
driver.findElements(By.tagName("form"));
		formProdutos.remove(0); // Pulldown de línguas.
		for (WebElement formProduto : formProdutos) {
			if
(formProduto.findElement(ElementsRastreador.CLASS_TITULO_SISTEMA).getText()
.equals(Padroes.NOME_SISTEMA)) {
				formProduto.submit();
				break;
			}
		}
	}

	public static void encerrarSessao() {
		driver.findElement(ElementsRastreador.SAIR).click();
		tearDownWebDriver();
	}

	public static String dataAtual() {
		SimpleDateFormat formatoData = new
SimpleDateFormat("ddMMyyyy");
		return new String(formatoData.format(new Date()));
	}

	public static void tirarScreenShot(String filename) throws
IOException {
		WebDriver augmentedDriver = new Augmenter().augment(driver);
		String pathArquivo = Configs.DIRECTORY + dateFormat(true) +
filename + ".png";
		File arquivo = new File(pathArquivo);
		File scrFile = ((TakesScreenshot)
augmentedDriver).getScreenshotAs(OutputType.FILE);
		FileUtils.copyFile(scrFile, arquivo);
	}

	public static boolean procuraElemento(String table, String
tableValue, String nomeBusca, String segundoTableValue,
			String segundoNomeBusca) {
```

```
            By elementoTexto = null;
            By elementoVerificacao = null;

            for (int tr = 0;; tr++) {
                  elementoTexto = By.id(table + tr + tableValue);
                  try {
                        if
(nomeBusca.equals(driver.findElement(elementoTexto).getText())) {
                              elementoVerificacao = By.id(table + tr +
segundoTableValue);
                              if
(segundoNomeBusca.equals(driver.findElement(elementoVerificacao).getText())
) {
                                    return true;
                              } else {
                                    return false;
                              }
                        }
                  } catch (NoSuchElementException e) {
                        List<WebElement> elementosPaginacao =
driver.findElements(ElementsRastreador.PAGINACAO);
                        System.out.println("Elementos Paginação: " +
elementosPaginacao);
                        Assert.fail(MsgErro.FAIL_BUSCA + nomeBusca + "'");
                        return false;
                  }
            }
      }

      public static String retornaValorTabela(String table, String
tableValueBusca, String tableValueRetorno,
                  String nomeBusca) {
            boolean paginacaoIncompleta = true;
            int tr = 0;
            List<WebElement> elementosPaginacao = null;
            elementosPaginacao =
driver.findElements(ElementsRastreador.PAGINACAO);
            elementosPaginacao.get(0).click();

            while (paginacaoIncompleta) {
                  By elementoBusca = By.id(table + tr + tableValueBusca);
```

```
                By elementoRetorno = By.id(table + tr +
tableValueRetorno);

                try {
                      if
(nomeBusca.equals(driver.findElement(elementoBusca).getText())) {
                            return
driver.findElement(elementoRetorno).getText();
                      } else {
                            tr++;
                      }
                } catch (Exception exc) {
                      try {
                            elementosPaginacao =
driver.findElements(ElementsRastreador.PAGINACAO);
                            for (WebElement elementoPaginacao :
elementosPaginacao) {
                                  if
(elementoPaginacao.getText().equals("»")) {
                                        if
(elementoPaginacao.getAttribute("class").equals(ElementsRastreador.PAGINACA
O_INACTIVE)) {
                                              throw new
NoSuchElementException("Paginação completa sem elemento.");
                                        }
                                        elementoPaginacao.click();
                                        break;
                                  }
                            }
                      } catch (NoSuchElementException e) {
                            Assert.fail(MsgErro.FAIL_BUSCA + nomeBusca +
"'");
                            e.printStackTrace();
                            break;
                      }
                }
          }

          return "";
     }
```

```
      public static void acionaElemento(String table, String tableValue,
String nomeBusca, String tableAction) {
            boolean paginacaoIncompleta = true;
            int controleLinhas = 0;
            List<WebElement> elementosPaginacao = null;

            waitJsBlock();


wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastreador
.PAGINACAO));

            elementosPaginacao =
driver.findElements(ElementsRastreador.PAGINACAO);
            elementosPaginacao.get(0).click();

            while (paginacaoIncompleta) {
                  By elementoTexto = By.id(table + controleLinhas +
tableValue);
                  By botaoAcao = By.id(table + controleLinhas +
tableAction);

                  try {
                        if
(nomeBusca.equals(driver.findElement(elementoTexto).getText())) {
                              scrollUp();
                              driver.findElement(botaoAcao).click();
                              break;
                        } else {
                              controleLinhas++;
                        }
                  } catch (NoSuchElementException exc) {
                        try {
                              elementosPaginacao =
driver.findElements(ElementsRastreador.PAGINACAO);
                              for (WebElement elementoPaginacao :
elementosPaginacao) {
                                    if
(elementoPaginacao.getText().equals("»")) {
                                          if
(elementoPaginacao.getAttribute("class").equals(ElementsRastreador.PAGINACA
```

```
O_INACTIVE)) {
                                                throw new
NoSuchElementException("Paginação completa sem elemento.");
                                          }
                                          elementoPaginacao.click();
                                          waitJsBlock();
                                          break;
                                    }
                              }
                        } catch (NoSuchElementException e) {
                              Assert.fail(MsgErro.FAIL_BUSCA + nomeBusca +
"'");
                              e.printStackTrace();
                              break;
                        }
                  }
            }
      }

      public static Boolean selectOption(By locator, String value) {
            WebElement select = driver.findElement(locator);
            List<WebElement> options = null;
            options = select.findElements(By.tagName("option"));
            for (WebElement option : options) {
                  if (option.getText().equals(value)) {
                        option.click();
                        return true;
                  }
            }
            return false;
      }

      public static void selectOption(By locator, Integer index) {
            WebElement select = driver.findElement(locator);
            List<WebElement> options = null;
            options = select.findElements(By.tagName("option"));
            options.get(index).click();
      }

      public static void scrollUp() {
            JavascriptExecutor js = (JavascriptExecutor) driver;
```

```
            js.executeScript("window.scrollBy(0,-999999)");
      }

      public static void scrollDown() {
            JavascriptExecutor js = (JavascriptExecutor) driver;
            js.executeScript("window.scrollBy(0,999999)");
      }

      public static void selectFile(By locator, String file) {
            String path = new File("//pardc01/dados$/Paripassu/Biblioteca
Virtual/Webdriver/files/", file).getAbsolutePath();
            driver.findElement(locator).sendKeys(path);
      }

      public static enum BrowserType {
            FIREFOX, IE, CHROME
      }

      public static void tearDownWebDriver() {
            driver.quit();
            endTestMsg("WebDriver");
      }

      public static void responderQuestionarioGpa(String tipoTela, String
nomeTela, String[] respostas) {
            String idCampo = "";
            int questao = 0;

            for (String resposta : respostas) {
                  questao++;
                  idCampo = tipoTela + nomeTela +
"QuestionariopnQuestoescp_questao" + questao;
                  selectOption(By.id(idCampo), resposta);
            }
      }

      public static String dateFormat(boolean print) {
            SimpleDateFormat dateFormat = new SimpleDateFormat();
            if (print) {
                  dateFormat.applyPattern("yyyy-MM-dd_HH-mm-ss_");
            } else {
```

```
                dateFormat.applyPattern("HH:mm:ss");
            }
            Date date = new Date();
            return dateFormat.format(date);
      }

      public static void startTestMsg(String className) {
            System.out.println("{ " + dateFormat(false) + " } [ " +
className + " / START ]");
      }

      public static void endTestMsg(String className) {
            System.out.println("{ " + dateFormat(false) + " } [ " +
className + " / END ]");
      }

      public static void testMsg(String className, String info) {
            System.out.println("{ " + dateFormat(false) + " } [ " +
className + " / " + info + " ]");
      }

      public static String capturaCodigo(String mensagem) {
            List<String> mensagemSucesso = Arrays.asList(mensagem.split("
"));
            String codigoGerado = null;

            for (String palavraMensagem : mensagemSucesso) {
                  try {
                        Double.parseDouble(palavraMensagem);
                        codigoGerado = palavraMensagem;
                        break;
                  } catch (NumberFormatException nfe) {
                  }
            }
            return codigoGerado;
      }

      public static void abreGrupoMenu(By grupo, By menu) {
            scrollUp();
            driver.findElement(grupo).click();
            driver.findElement(menu).click();
```

```
            waitJsBlock();
      }

      public static void fechaGrupoMenu(By grupo, By menu) {
            scrollUp();
            while (driver.findElement(menu).isDisplayed()) {
                  driver.findElement(grupo).click();
            }
      }

      public static String getRandomValue(final int low, final int top,
final int decimal) {
            final Random rnd = new Random();

            if (low < 0 || top <= low || decimal < 0) {
                  throw new IllegalArgumentException("Valores para geração
do número random estão incorretos.");
            }

            final double dbl = ((rnd == null ? new Random() :
rnd).nextDouble() * (top - low)) + low;
            return String.format("%1$." + decimal + "f", dbl).replace(",",
".");
      }

      public static void excluirRegistro(boolean exclusaoPermitida, boolean
realizarExclusao) {


wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastreador
.MODAL));
            if (exclusaoPermitida) {
                  if (realizarExclusao) {

      driver.findElement(ElementsRastreador.MODAL_DELETE_SIM).click();
                  } else {

      driver.findElement(ElementsRastreador.MODAL_DELETE_NAO).click();
                  }
            } else {
```

```
	driver.findElement(ElementsRastreador.MODAL_DELETE_FECHAR).click();
		}


wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRastread
or.MODAL));
	}

	public static void waitJsBlock() {
		try {


Util.wait.until(ExpectedConditions.stalenessOf(driver.findElement(ElementsR
astreador.BLOCKUI)));
		} catch (NoSuchElementException nsee) {
			// do nothing
		}
	}

};

package util;

import java.io.File;
import java.io.IOException;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.testng.ITestResult;
import org.testng.TestListenerAdapter;

public class ScreenshotListener extends TestListenerAdapter {

	String screensFolder = "testng_reports/exceptions";
	String dataAtual = Util.dateFormat(true);

	@Override
	public void onTestFailure(ITestResult result) {
		new File(screensFolder).mkdirs();
		File srcFile = ((TakesScreenshot)
```

```
Util.driver).getScreenshotAs(OutputType.FILE);
            try {
                  FileUtils.copyFile(srcFile, new File(screensFolder + "/"
+ result.getMethod().getClass() + "_"
                              + result.getMethod().getMethodName() +
".png"));
            } catch (IOException e) {
                  e.printStackTrace();
                  System.out.println("Não foi possível gerar a ScreenShot
com o erro");
            }
      }

      @Override
      public void onConfigurationFailure(ITestResult result) {
            new File(screensFolder).mkdirs();
            File srcFile = ((TakesScreenshot)
Util.driver).getScreenshotAs(OutputType.FILE);
            try {
                  FileUtils.copyFile(srcFile, new File(screensFolder + "/"
+ result.getMethod().getClass() + "_"
                              + result.getMethod().getMethodName() +
".png"));
            } catch (IOException e) {
                  e.printStackTrace();
                  System.out.println("Não foi possível gerar a ScreenShot
com o erro");
            }
      }

}
```

```
package util;

public class MsgSistema {

      public static String CADASTRO_NULL = "Código null gerado com sucesso.
Cadastrar outro?";
      public static String CADASTRO_NULL_REPLICANDO = "Dados cadastrados
com sucesso. Cadastrar outro? Código: null";
      public static String CARGA_RECEBIDA = "Essa carga já foi recebida no
```

```
destino, não é possível editá-la.";
      public static String ERRO_QUANTIDADE = "[ERRO] A quantidade informada
não pode ser menor que a quantidade enviada";

}

package util;

public class MsgErro {

    public static final String FAIL_CADASTRO = "Cadastro não realizado,";
    public static final String FAIL_LOAD_INSERT = "Não carregou tela de
novo registro,";
    public static final String FAIL_LOAD_EDIT = "Não carregou tela de
edição de registro,";
    public static final String FAIL_LISTAGEM = "Não foi possível listar os
elementos,";
    public static final String FAIL_BUSCA = "Não foi possível encontrar '";
    public static final String FAIL_EDICAO = "Edição não realizada,";
    public static final String FAIL_ADICIONAR_ELEMENTO = "Elemento não foi
adicionado,";
    public static final String FAIL_ADICIONAR_INGREDIENTE = "Ingrediente
não foi adicionado,";
    public static final String FAIL_CADASTRO_EXISTENTE = "Cadastro foi
realizado,";
    public static final String FAIL_DELETE_COMPLETE = "Cadastro não foi
excluido com sucesso,";
    public static final String FAIL_REMOVER_ELEMENTO = "Elemento não foi
removido,";
    public static final String FAIL_PESSOA_FISICA = "Informações de pessoa
física não exibidos,";
    public static final String FAIL_LOAD_ABA = "Não foi possível carregar a
aba,";
    public static final String FAIL_ADICIONAR_PRODUTO = "Produto não foi
adicionado,";
    public static final String FAIL_ADICIONAR_TALHAO = "Talhão não foi
adicionado,";
    public static final String FAIL_LOAD_MODAL = "Erro carregando o
modal,";
    public static final String FAIL_ADICIONAR_DEFENSIVO = "Defensivo não
foi adicionado,";
```

```
    public static final String FAIL_LOAD_PAGE = "Página não foi
carregada,";
    public static final String FAIL_LOAD_FILTER = "Filtro não foi
carregado,";
    public static final String FAIL_ELEMENTO_NAO_VISIVEL = "Elemento não
está visível,";
    public static final String FAIL_EDICAO_ENVIO_RECEBIDO = "Foi possível
editar um envio já recebido.";
    public static final String FAIL_EDICAO_QUANTIDADE = "Foi possível
editar a quantidade para uma quantidade menor que a enviada.";
    public static final String FAIL_MSG_LOCALIZACAO = "Localização não
encontrada,";
    public static final String FAIL_VALOR_DIFERENTE = "Valores não estão
corretos,";
    public static final String FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS = "Texto
está incorreto,";

}

package util;

import org.openqa.selenium.By;

public class ElementsRastreador {

      // Elementos de acesso do sistema
      public static final By PAINEL_CONTROLE =
By.id("startForm:telaPainelControlepanelPainelControle");
      public static final By SAIR = By.id("logoutDiv");

      // Grupo Rastreamento
      public static final By RASTREAMENTO =
By.id("iconstartForm:pnGrupo321");

      // Menus do grupo Rastreamento
      public static final By RECEBIMENTO =
By.id("iconstartForm:pnItemMenu322");
      public static final By ENVIO_SIMPLES =
By.id("iconstartForm:pnItemMenu323");
      public static final By ENVIO_MISTURA_ORIGENS =
By.id("iconstartForm:pnItemMenu324");
```

```
      public static final By ENVIO_DIRETO =
By.id("iconstartForm:pnItemMenu325");
      public static final By PROCESSAMENTO =
By.id("iconstartForm:pnItemMenu327");
      public static final By DESCARTE =
By.id("iconstartForm:pnItemMenu328");
      public static final By PROCESSAMENTO_PRODUTO =
By.id("iconstartForm:pnItemMenu389");
      public static final By PROCESSAMENTO_INSUMO =
By.id("iconstartForm:pnItemMenu556");
      public static final By PROCESSAMENTO_DIRETO =
By.id("iconstartForm:pnItemMenu421");
      public static final By ENVIO_DIRETO_MISTURA_ORIGENS =
By.id("iconstartForm:pnItemMenu549");
      public static final By RECEBIMENTO_CODIGO =
By.id("iconstartForm:pnItemMenu326");
      public static final By PROCESSAMENTO_PESCADO =
By.id("iconstartForm:pnItemMenu467");

      // Grupo Cadastro
      public static final By CADASTRO = By.id("iconstartForm:pnGrupo68");
      // Menus do grupo Cadastro
      public static final By MEUS_CADASTROS =
By.id("iconstartForm:pnItemMenu82");
      public static final By ORIGENS = By.id("iconstartForm:pnItemMenu81");
      public static final By DESTINO = By.id("iconstartForm:pnItemMenu83");
      public static final By MATERIA_PRIMA =
By.id("iconstartForm:pnItemMenu320");
      public static final By RECEITA =
By.id("iconstartForm:pnItemMenu113");
      public static final By ANALISE =
By.id("iconstartForm:pnItemMenu102");
      public static final By TURNO = By.id("iconstartForm:pnItemMenu261");
      public static final By ENVASADORA =
By.id("iconstartForm:pnItemMenu534");
      public static final By LINHA_PRODUCAO =
By.id("iconstartForm:pnItemMenu262");
      public static final By FORNECEDOR =
By.id("iconstartForm:pnItemMenu539");

      // Grupo Configurações
```

```
      public static final By CONFIGURACOES =
By.id("iconstartForm:pnGrupo103");
      // Menus do grupo Configurações
      public static final By CONFIGURACAO_ETIQUETA =
By.id("iconstartForm:pnItemMenu256");
      public static final By CONFIGURACAO_GERAL =
By.id("iconstartForm:pnItemMenu104");
      public static final By TIPO_ANALISE =
By.id("iconstartForm:pnItemMenu177");
      public static final By CADASTRO_NUTRICIONAL =
By.id("iconstartForm:pnItemMenu561");
      public static final By IDENTIDADE_VISUAL =
By.id("iconstartForm:pnItemMenu161");
      public static final By GERENCIAMENTO_BIBLIOTECA =
By.id("iconstartForm:pnItemMenu559");

      // Grupo Embalagens
      public static final By EMBALAGENS =
By.id("iconstartForm:pnGrupo335");
      // Menus do grupo Embalagens
      public static final By TIPO_EMBALAGEM =
By.id("iconstartForm:pnItemMenu336");
      public static final By EMBALAGEM =
By.id("iconstartForm:pnItemMenu337");

      // Grupo Questionario
      public static final By QUESTIONARIO =
By.id("iconstartForm:pnGrupo361");
      // Menus do grupo Questionario
      public static final By APLICACAO_QUESTIONARIO =
By.id("iconstartForm:pnItemMenu362");
      public static final By GERENCIAMENTO_QUESTIONARIO =
By.id("iconstartForm:pnItemMenu492");

      // Grupo Indicadores
      public static final By INDICADORES =
By.id("iconstartForm:pnGrupo329");
      // Menus do grupo Questionario
      public static final By RELATORIOS =
By.id("iconstartForm:pnItemMenu330");
      public static final By BIBLIOTECA =
```

```
By.id("iconstartForm:pnItemMenu560");

      // Elementos gerais de formulário
      public static final By FORM_SALVAR =
By.id("startForm:telaElobasepanelcontentformbtsalvarbt");
      public static final By FORM_SUCESSO_SIM =
By.id("formModalPanel:pnMsgSucessoCadastrobtYesbt");
      public static final By FORM_SUCESSO_NAO =
By.id("formModalPanel:pnMsgSucessoCadastrobtNobt");
      public static final By FORM_SUCESSO_REPLICAR_SIM = By
                  .id("formModalPanel:pnMsgSucessoCadastroReplicandoDadosbt
YesReplicandobt");
      public static final By FORM_SUCESSO_REPLICAR_NAO =
By.id("formModalPanel:pnMsgSucessoCadastroReplicandoDadosbtNobt");
      public static final By FORM_ERRO_FECHAR =
By.id("formModalPanel:pnErrosbtClosebt");
      public static final By FORM_EDICAO_FECHAR =
By.id("formModalPanel:pnMsgConfirmacaoEdicaobtClosebt");
      public static final By FORM_SUCESSO_FECHAR = By
                  .id("formModalPanel:pnMsgSucessoCadastroConfiguracaoCodig
obtOKConfCodigobt");

      // Elementos de Modais
      public static final By MODAL_DELETE_FECHAR =
By.id("formModalPanel:pnErroIntegridadeReferencialbtClosebt");
      public static final By MODAL_DELETE_SIM =
By.id("formModalPanel:pnDeletebtYesbt");
      public static final By MODAL_DELETE_NAO =
By.id("formModalPanel:pnDeletebtNobt");
      public static final By MODAL_CONFIRMA =
By.id("formModalPanel:pnMsgSucessoCadastrootMsg");
      public static final By MODAL_CONFIRMA_REPLICANDO =
By.id("formModalPanel:pnMsgSucessoCadastroReplicandoDadosotMsg");
      public static final By MODAL_ERRO = By.id("formModalPanel");
      public static final By MODAL_ERRO_TITLE =
By.id("formModalPanel:pnErrospnTitle");
      public static final By MODAL_FECHAR =
By.id("formModalPanel:pnErrosbtClosebt");

      // Elementos de Tabelas
      public static final String TABLE_EDITAR = ":giedit";
```

```
      public static final String TABLE_EXCLUIR = ":gidelete";
      public static final String TABLE_SEARCH = ":gisearch";

      // Elementos de localização
      public static final By LAT_GRAUS =
By.id("startForm:telaElolatGraus");
      public static final By LAT_MINUTOS =
By.id("startForm:telaElolatMinutos");
      public static final By LAT_SEGUNDOS =
By.id("startForm:telaElolatSegundos");
      public static final By LNG_GRAUS =
By.id("startForm:telaElolngGraus");
      public static final By LNG_MINUTOS =
By.id("startForm:telaElolngMinutos");
      public static final By LNG_SEGUNDOS =
By.id("startForm:telaElolngSegundos");

      // Elementos de i18n
      public static final By SELECT_LINGUA =
By.id("startForm:selectLanguage2");

      // Tipos de tela
      public static final String STARTFORM = "startForm:";
      public static final String FORMMODAL = "formModalPanel:";

      // Paginação
      public static final By PAGINA_ATIVA = By.className("rich-datascr-
act");
      public static final By PAGINA_INATIVA = By.className("rich-datascr-
inact");
      public static final By PAGINACAO = By.className("rich-datascr-
button");
      public static final String PAGINACAO_INACTIVE = "rich-datascr-button-
dsbld rich-datascr-button";

      // Tags de html
      public static final By TAG_FORM = By.tagName("form");

      // Classe do titulo do portal.
      public static final By CLASS_TITULO_SISTEMA =
By.className("tituloSistema");
```

```
      public static final By MODAL = By.id("mpDefaultCDiv");
      public static final By BLOCKUI = By.className("blockUI");
}

package util;

public class ElementsFiles {

    public static final String LOGO_PARIPASSU_PNG = "logo_paripassu.png";
    public static final String DOC_PARIPASSU_PDF = "documento_padrao.pdf";

}

package util;

import org.openqa.selenium.By;

import util.Util.BrowserType;

public class Configs {

    public static String URL_BASE = null;
    public static final String TIMEOUT = "60000";
    public static final int INT_TIMEOUT = 30;
    public static final By LOADING_BAR =
By.xpath("//*[@id='progressBarDiv']/span/span");
    public static final String DIRECTORY =
"//pardc01/dados$/Paripassu/Util/webdriver/";
    public static final int HOLD = 3000;
    public static final int WAIT = 1000;
    public static final int SUP = 500;
    public static final long MEGA_TIMEOUT = 300;
    public static String ENDERECO_HUB = null;
    public static BrowserType browser = null;

}

package perfil.geralInterno.util;

import org.testng.annotations.AfterTest;
```

```
import util.Util;

public class TearDownWebDriver {

    @AfterTest(alwaysRun = true)
    public void tearDownWebDriver() throws InterruptedException {
      Util.encerrarSessao();
    }

}

package perfil.geralInterno.util;

import java.net.MalformedURLException;
import java.util.Properties;

import org.testng.ITestContext;
import org.testng.annotations.BeforeTest;

import perfil.geral.util.Padroes;

import util.Configs;
import util.Util;

public class SetUpWebDriver {

    @BeforeTest(alwaysRun = true)
    public void setUpWebDriver(ITestContext context)
          throws InterruptedException, MalformedURLException {

      Properties sysProps = System.getProperties();
      String browser = sysProps.getProperty("ant.grid.browser");
      Configs.URL_BASE = sysProps.getProperty("ant.grid.url");
      Configs.ENDERECO_HUB = sysProps.getProperty("ant.grid.hub");

      if (browser.equals("firefox")) {
          Configs.browser = Util.BrowserType.FIREFOX;
      } else if (browser.equals("ie")) {
          Configs.browser = Util.BrowserType.IE;
      } else if (browser.equals("chrome")) {
```

```
            Configs.browser = Util.BrowserType.CHROME;
      }

      Util.getWebDriver();
      Util.getRastreadorSenhaInterna(Padroes.USUARIO_PADRAO_NOME);
    }

}

package perfil.geralInterno.configuracoes.configuracaoGeralInterno;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.annotations.*;

import util.ElementsRastreador;
import util.Util;

public class TestConfiguracaoGeralInterno {
      WebDriver driver;

      @BeforeClass(alwaysRun = true)
      public void setUpConfiguracaoGeralInterno() {


Util.startTestMsg(TestConfiguracaoGeralInterno.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CONFIGURACAO_GERAL);
      }

      @Test
      public void setConfiguracoesGeraisInterno() {
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsConfiguracaoGera
lInterno.FORM_RASTREAMENTO_PROCESSAMENTO_NAO));


driver.findElement(ElementsConfiguracaoGeralInterno.FORM_RASTREAMENTO_PROCE
SSAMENTO_NAO).click();
```

```
driver.findElement(ElementsConfiguracaoGeralInterno.FORM_DEVOLUCAO_PRODUTO_
RECEBIMENTO_NAO).click();


driver.findElement(ElementsConfiguracaoGeralInterno.FORM_REPROCESSAMENTO_NA
O).click();


driver.findElement(ElementsConfiguracaoGeralInterno.FORM_PRODUTOS_ORGANICOS
_NAO).click();


driver.findElement(ElementsConfiguracaoGeralInterno.FORM_TALHAO_BASICO_SIM)
.click();


driver.findElement(ElementsConfiguracaoGeralInterno.FORM_FILTROS_OTIMIZACAO
_NAO).click();
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsConfiguracaoGeralInterno.FORM_SALVAR).click();
                        return
driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).isDisplayed();
                  }
            });

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDownConfiguracaoGeralInterno() {
            Util.fechaGrupoMenu(ElementsRastreador.CONFIGURACOES,
ElementsRastreador.CONFIGURACAO_GERAL);

      Util.endTestMsg(TestConfiguracaoGeralInterno.class.getSimpleName());
      }
```

```
}

package perfil.geralInterno.configuracoes.configuracaoGeralInterno;

import org.openqa.selenium.By;

public class ElementsConfiguracaoGeralInterno {

	// Div principal
	public static final By BODY = By.id("startForm:pnParipassu_body");

	// Elementos do Formulário
	public static final By FORM_RASTREAMENTO_PROCESSAMENTO_NAO = By
			.id("startForm:telaConfiguracaockrealizaRealizaControleCo
digo:1");
	public static final By FORM_DEVOLUCAO_PRODUTO_RECEBIMENTO_NAO = By
			.id("startForm:telaConfiguracaockrealizaDevolucao:1");
	public static final By FORM_REPROCESSAMENTO_NAO = By
			.id("startForm:telaConfiguracaockrealizaDevolucaoReproces
samento:1");
	public static final By FORM_PRODUTOS_ORGANICOS_NAO = By
			.id("startForm:telaConfiguracaocktrabalhaComOrganicos:1")
;
	public static final By FORM_TALHAO_BASICO_SIM = By
			.id("startForm:telaConfiguracaocktalhaoBasico:0");
	public static final By FORM_FILTROS_OTIMIZACAO_NAO = By
			.id("startForm:telaConfiguracaockfiltrosOtimizacao:1");
	public static final By FORM_SALVAR = By
			.id("startForm:telaConfiguracaobasepanelcontentpnBotoesbt
salvarbt");
}

package perfil.geral.util;

import org.testng.annotations.AfterTest;

import util.Util;

public class TearDownWebDriver {
```

```
      @AfterTest(alwaysRun = true)
      public void tearDownWebDriver() throws InterruptedException {
            Util.encerrarSessao();
      }

}

package perfil.geral.util;

import java.net.MalformedURLException;
import java.util.Properties;

import org.testng.ITestContext;
import org.testng.annotations.BeforeTest;

import util.Configs;
import util.Util;

public class SetUpWebDriver {

    @BeforeTest(alwaysRun = true)
    public void setUpWebDriver(ITestContext context)
          throws InterruptedException, MalformedURLException {

      Properties sysProps = System.getProperties();
      String browser = sysProps.getProperty("ant.grid.browser");
      Configs.URL_BASE = sysProps.getProperty("ant.grid.url");
      Configs.ENDERECO_HUB = sysProps.getProperty("ant.grid.hub");

      if (browser.equals("firefox")) {
          Configs.browser = Util.BrowserType.FIREFOX;
      } else if (browser.equals("ie")) {
          Configs.browser = Util.BrowserType.IE;
      } else if (browser.equals("chrome")) {
          Configs.browser = Util.BrowserType.CHROME;
      }

      Util.getWebDriver();
      Util.getRastreador(Padroes.USUARIO_PADRAO_NOME);
    }
```

```
}

package perfil.geral.util;

public class Padroes {

	public static final Object NOME_SISTEMA = "Rastreador 3G";

	// Usuários do Rastreador
	public static final String USUARIO_PADRAO_NOME = "auto";
	public static final String USUARIO_ADMINISTRADOR = "lucianes";
	public static final String SENHA_PADRAO = "123";

	// Nomes de cadastro
	public static final String NOME_FORNECEDOR = "_Fornecedor WebDriver";
	public static final String NOME_RECEITA = "_Receita WebDriver";
	public static final String NOME_DESTINO_CADASTRO = "_Destino
WebDriver";
	public static final String NOME_DESTINO_UTILIZADO = "DESTINO GPA";
	public static final String NOME_MATERIA_PRIMA = "_Materia-Prima
WebDriver";
	public static final String NOME_ORIGEM = "_Origem WebDriver";
	public static final String NOME_ORIGEM_2 = "_Origem WebDriver_2";
	public static final String NOME_TIPO_EMBALAGEM = "_Tipo Embalagem
WebDriver";
	public static final String NOME_EMBALAGEM = "_Embalagem WebDriver";
	public static final String NOME_EMBALAGEM_QUALIDADE = "_Embalagem
Qualidade";
	public static final String NOME_MEU_CADASTRO = "_Cadastro WebDriver";
	public static final String NOME_QUESTIONARIO = "_Questionario
WebDriver Rastreador";
	public static final String NOME_AVALIADO = "_Origem Qualidade";
	public static final String NOME_ETIQUETA = "50mm x 120mm x 2 (data de
validade e embalamento)";
	public static final String NUMERO_VINTE = "01234567890123456789";
	public static final String NUMERO_CINQUENTA =
"01234567890123456789012345678901234567890123456789";
	public static final String NOME_CERTIFICADOR = "Control Union";
	public static final String TIPO_CERTIFICADO = "Orgânico";
	public static final String NOME_TURNO = "_Meu Turno";
	public static final String NOME_ENVASADORA = "_Envasadora WebDriver";
```

```
      public static final String NOME_LINHA_PRODUCAO = "_Linha Producao
WebDriver";
      public static final String NOME_TIPO_ANALISE = "_Tipo Analise
WebDriver";
      public static final String NOME_LABORATORIO = "PRÓPRIO";
      public static final String OBSERVACAO = "_Observacao WebDriver";
      public static final String COMPLEMENTO = "_Complemento WebDriver";

      public static final String DESPESA = "Açúcar";
      public static final String EDITADO = " Editado";
      public static final String CPF = "388.548.043-39";
      public static final String CPF_EDITADO = "500.951.220-36";
      public static final String ALVO_ANALISE = "Fornecedor";
      public static final String PARAMETRO = "CTC";
      public static final String PARAMETRO_SUFIXO = " (cmolc/dm³)";
      public static final String QTDE = "0123456789";
      public static final String QTDE_EDITADO = "9876543210";
      public static final String NOTA_FISCAL = "78912346";
      public static final String NOTA_FISCAL_EDITADO = "01234567";
    public static final String SENHA_USUARIO_INTERNO = "auto@123";

      public static final String ATIVO = "Ativo";
      public static final String NOME_PRODUTO = "Abacate";
      public static final String PRODUTO_CODIGO_EAN = "176369423131";
      public static final String PRODUTO_EMBALAGEM_EAN = "100 g";
      public static final String NOME_TIPO_INGREDIENTE = "Produto";
      public static final String NOME_CATEGORIA = "Matéria-prima";
      public static final String NOME_UNIDADE_MEDIDA = "KG";
      public static final String CODIGO_EAN = "789012345678";
      public static final String QTD_PORCAO = "012345";
      public static final String VL_DIARIO = "012";
      public static final String CODIGO_PEPSICO = "012345678903";

      // Elementos booleanos
      public static final String SIM = "Sim";
      public static final String NAO = "Não";

      // Elementos de arquivo.
      public static final String PATH_TESTE = "c:/selenium/";
      public static final String ARQUIVO_TESTE = "documentoAutomacao.txt";
```

```
	// Nomes gerais
	public static final String TALHAO = "Talhão";
	public static final String TALHOES = "Talhões";
	public static final CharSequence PORCAO_PRODUTO = "Porção de 100g";
	public static final String ETIQUETA = "Térmica - IFCO (50 mm x 160 mm
x 2) - ppla plus - Logo: Não";
	public static final CharSequence NOME_PRODUTO_QUESTIONARIO =
"Batata";
	public static final String NOME_BIBLIOTECA = "_Biblioteca WebDriver";
	public static final String NOME_GPA = "Grupo Pão de Açúcar";

}

package perfil.geral.rastreamento.recebimento;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestRecebimento {
	WebDriver driver;
	String dataAtual = Util.dataAtual();
	public static List<String> codigosRecebimento = new
ArrayList<String>();
	public static List<BigDecimal> totalEstoqueFirstRecebimento =
Arrays.asList(new BigDecimal(0.00), new BigDecimal(
			0.00), new BigDecimal(0.00));
	public static List<BigDecimal> totalEstoqueSecondRecebimento =
```

```
Arrays.asList(new BigDecimal(0.00), new BigDecimal(
                  0.00), new BigDecimal(0.00));

      @BeforeClass(alwaysRun = true)
      public void setUpRecebimento() {
            Util.startTestMsg(TestRecebimento.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.RECEBIMENTO);
      }

      @Test(groups = { "insert", "verify", "campos" })
      public void abreTelaInsertRecebimento() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));
            driver.findElement(ElementsRecebimento.NOVO_REGISTRO).click();
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertRecebimento")
      public void insertFirstRecebimento() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.FORM_DT_COLHEITA));


driver.findElement(ElementsRecebimento.FORM_DT_COLHEITA).sendKeys(dataAtual
);
            Util.selectOption(ElementsRecebimento.FORM_ORIGEM,
Padroes.NOME_ORIGEM);
            Util.waitJsBlock();
            Util.selectOption(ElementsRecebimento.FORM_PRODUTO,
Padroes.NOME_PRODUTO);
            totalEstoqueFirstRecebimento.set(0, new BigDecimal(450.00));
            totalEstoqueFirstRecebimento.set(2, new BigDecimal(450.00));


driver.findElement(ElementsRecebimento.FORM_QTDE).sendKeys(totalEstoqueFirs
```

```
tRecebimento.get(0).toString());
            Util.selectOption(ElementsRecebimento.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsRecebimento.FORM_NOTA_FISCAL).sendKeys("78912346
");

      driver.findElement(ElementsRecebimento.FORM_PRECO).sendKeys("3.99");


driver.findElement(ElementsRecebimento.FORM_PLACA_VEICULO).sendKeys("ABC
4589");


driver.findElement(ElementsRecebimento.FORM_NOME_MOTORISTA).sendKeys("Carla
Viviane");


driver.findElement(ElementsRecebimento.FORM_DT_RECEBIMENTO).sendKeys(dataAt
ual);
            Util.selectOption(ElementsRecebimento.FORM_LOCAL_RECEBIMENTO,
Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsRecebimento.FORM_OBSERVACAO).sendKeys("Observaca
o recebimento");
            driver.findElement(ElementsRecebimento.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.FORM_SUCESSO_REPLICAR_SIM));
            Assert.assertEquals(


driver.findElement(ElementsRecebimento.MODAL_CONFIRMA).getText().equals(Msg
Sistema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);


codigosRecebimento.add(Util.capturaCodigo(driver.findElement(ElementsRecebi
```

```
mento.MODAL_CONFIRMA).getText()));
            Util.testMsg("TestRecebimento", "Código: " +
codigosRecebimento.get(0));


driver.findElement(ElementsRecebimento.FORM_SUCESSO_REPLICAR_SIM).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertFirstRecebimento")
      public void insertSecondRecebimento() {


driver.findElement(ElementsRecebimento.FORM_DT_COLHEITA).sendKeys(dataAtual
);
            Util.selectOption(ElementsRecebimento.FORM_ORIGEM,
Padroes.NOME_ORIGEM);
            Util.waitJsBlock();
            Util.selectOption(ElementsRecebimento.FORM_PRODUTO,
Padroes.NOME_PRODUTO);
            totalEstoqueSecondRecebimento.set(0, new BigDecimal(520.00));
            totalEstoqueSecondRecebimento.set(2, new BigDecimal(520.00));


driver.findElement(ElementsRecebimento.FORM_QTDE).sendKeys(totalEstoqueSeco
ndRecebimento.get(0).toString());
            Util.selectOption(ElementsRecebimento.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsRecebimento.FORM_NOTA_FISCAL).sendKeys("75634534
2");

      driver.findElement(ElementsRecebimento.FORM_PRECO).sendKeys("0.89");


driver.findElement(ElementsRecebimento.FORM_PLACA_VEICULO).sendKeys("KJH
```

```
8956");


driver.findElement(ElementsRecebimento.FORM_NOME_MOTORISTA).sendKeys("Julia
na Gast");


driver.findElement(ElementsRecebimento.FORM_DT_RECEBIMENTO).sendKeys(dataAt
ual);
            Util.selectOption(ElementsRecebimento.FORM_LOCAL_RECEBIMENTO,
Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsRecebimento.FORM_OBSERVACAO).sendKeys("Observaca
o recebimento");
            Util.selectOption(ElementsRecebimento.FORM_INSPECAO, "Inspeção
recebimento");


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.FORM_CHECKLIST_1));


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_1).sendKeys("6");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_2).sendKeys("13");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_3).sendKeys("10");
            Util.selectOption(ElementsRecebimento.FORM_CHECKLIST_4,
"Conforme");
            Util.selectOption(ElementsRecebimento.FORM_CHECKLIST_5,
"Conforme");
            driver.findElement(ElementsRecebimento.FORM_TOPICO_2).click();


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_6).sendKeys("5");
            Util.selectOption(ElementsRecebimento.FORM_CHECKLIST_7,
"Conforme");
```

```
driver.findElement(ElementsRecebimento.FORM_CHECKLIST_8).sendKeys("5");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_9).sendKeys("5");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_10).sendKeys("5");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_11).sendKeys("5");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_12).sendKeys("5");
            driver.findElement(ElementsRecebimento.FORM_TOPICO_3).click();


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_13).sendKeys("5");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_14).sendKeys("5");


driver.findElement(ElementsRecebimento.FORM_CHECKLIST_15).sendKeys("5");
            driver.findElement(ElementsRecebimento.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRecebimento.MODAL_CONFIRMA).getText().equals(Msg
Sistema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);


codigosRecebimento.add(Util.capturaCodigo(driver.findElement(ElementsRecebi
mento.MODAL_CONFIRMA).getText()));
            Util.testMsg("TestRecebimento", "Código: " +
codigosRecebimento.get(1));
```

```
driver.findElement(ElementsRecebimento.FORM_SUCESSO_REPLICAR_NAO).click();

            // Impressão das mensagens para verificação de estoque
            Util.testMsg("Recebimento: #" + codigosRecebimento.get(0),
"{ Rec: "
                        + totalEstoqueFirstRecebimento.get(0).setScale(2) +
" | Env: "
                        + totalEstoqueFirstRecebimento.get(1).setScale(2) +
" | Disp: "
                        + totalEstoqueFirstRecebimento.get(2).setScale(2) +
" } / EXPECTED");
            Util.testMsg("Recebimento: #" + codigosRecebimento.get(1),
"{ Rec: "
                        + totalEstoqueSecondRecebimento.get(0).setScale(2) +
" | Env: "
                        + totalEstoqueSecondRecebimento.get(1).setScale(2) +
" | Disp: "
                        + totalEstoqueSecondRecebimento.get(2).setScale(2) +
" } / EXPECTED");
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertSecondRecebimento")
      public void verifyInsertRecebimento() {
            List<String> valorEstoqueTabelaFirstRecebimento = new
ArrayList<String>();
            List<String> valorEstoqueTabelaSecondRecebimento = new
ArrayList<String>();



Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));


valorEstoqueTabelaFirstRecebimento.add(Util.retornaValorTabela(ElementsRece
bimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
```

```
                        codigosRecebimento.get(0)));


valorEstoqueTabelaFirstRecebimento.add(Util.retornaValorTabela(ElementsRece
bimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        codigosRecebimento.get(0)));


valorEstoqueTabelaFirstRecebimento.add(Util.retornaValorTabela(ElementsRece
bimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        codigosRecebimento.get(0)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaFirstRecebimento.get(0),
totalEstoqueFirstRecebimento.get(0).setScale(2)
                        .toString(), MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaFirstRecebimento.get(1),
totalEstoqueFirstRecebimento.get(1).setScale(2)
                        .toString(), MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaFirstRecebimento.get(2),
totalEstoqueFirstRecebimento.get(2).setScale(2)
                        .toString(), MsgErro.FAIL_CADASTRO);

            Util.testMsg(
                        "Recebimento: #" + codigosRecebimento.get(0),
                        "{ Rec: " +
valorEstoqueTabelaFirstRecebimento.get(0) + " | Env: "
                                    +
valorEstoqueTabelaFirstRecebimento.get(1) + " | Disp: " +
valorEstoqueTabelaFirstRecebimento.get(2)
                                    + " } / RESULT");

            // Executa asserts do segundo recebimento (recebimento,
enviado,
            // disponível).
```

```
valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        codigosRecebimento.get(1)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(0),
totalEstoqueSecondRecebimento.get(0).setScale(2)
                        .toString(), MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(1),
totalEstoqueSecondRecebimento.get(1).setScale(2)
                        .toString(), MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(2),
totalEstoqueSecondRecebimento.get(2).setScale(2)
                        .toString(), MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" + codigosRecebimento.get(1),
"{ Rec: " + valorEstoqueTabelaSecondRecebimento.get(0)
                        + " | Env: " +
valorEstoqueTabelaSecondRecebimento.get(1) + " | Disp: "
                        + valorEstoqueTabelaSecondRecebimento.get(2) + " } /
RESULT");
```

```
	}

	@Test(groups = { "edit" })
	public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));
		Util.acionaElemento(ElementsRecebimento.TABLE,
ElementsRecebimento.TABLE_VALUE_ORIGEM, Padroes.NOME_ORIGEM
				+ Padroes.EDITADO, ElementsRastreador.TABLE_EDITAR);
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
	public void editFirstRecebimento() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.FORM_DT_COLHEITA));

	driver.findElement(ElementsRecebimento.FORM_DT_COLHEITA).clear();


driver.findElement(ElementsRecebimento.FORM_DT_COLHEITA).sendKeys(dataAtual
);
		driver.findElement(ElementsRecebimento.FORM_QTDE).clear();

	driver.findElement(ElementsRecebimento.FORM_QTDE).sendKeys("550");

	driver.findElement(ElementsRecebimento.FORM_PLACA_VEICULO).clear();


driver.findElement(ElementsRecebimento.FORM_PLACA_VEICULO).sendKeys("ASD
7894");

	driver.findElement(ElementsRecebimento.FORM_NOME_MOTORISTA).clear();


driver.findElement(ElementsRecebimento.FORM_NOME_MOTORISTA).sendKeys("Carol
ina Gonçalves");
```

```
      driver.findElement(ElementsRecebimento.FORM_NOTA_FISCAL).clear();


driver.findElement(ElementsRecebimento.FORM_NOTA_FISCAL).sendKeys("12345678
9");
            driver.findElement(ElementsRecebimento.FORM_PRECO).clear();

      driver.findElement(ElementsRecebimento.FORM_PRECO).sendKeys("1.19");

      driver.findElement(ElementsRecebimento.FORM_DT_RECEBIMENTO).clear();


driver.findElement(ElementsRecebimento.FORM_DT_RECEBIMENTO).sendKeys(dataAt
ual);


driver.findElement(ElementsRecebimento.FORM_OBSERVACAO).sendKeys(Padroes.ED
ITADO);
            driver.findElement(ElementsRecebimento.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteFirstSecondRecebimento() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));
            Util.acionaElemento(ElementsRecebimento.TABLE,
ElementsRecebimento.TABLE_VALUE_ORIGEM, Padroes.NOME_ORIGEM
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
```

```
            Util.excluirRegistro(true, true);
            Util.acionaElemento(ElementsRecebimento.TABLE,
ElementsRecebimento.TABLE_VALUE_ORIGEM, Padroes.NOME_ORIGEM
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertRecebimento")
      public void validacaoCamposObrigatoriosRecebimento() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.FORM_SALVAR));

            driver.findElement(ElementsRecebimento.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsRecebimento.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.RECEBIMENTO);
            Util.endTestMsg(TestRecebimento.class.getSimpleName());
      }
}

package perfil.geral.rastreamento.recebimento;
```

```
import org.openqa.selenium.By;

public class ElementsRecebimento {

    // Elementos da tela principal - Botões
    public static final By NOVO_REGISTRO = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentlistgiNovoRegistro"
);
    public static final String TABLE_IMPRIMIR =
":telaRecebimentoRomaneio3GdataTablepnEtiquetabtImprimirEtiqueta";
    public static final By FILTRO_NOVO = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentfiltertoppgrightlkn
ewfilter");

    // Elementos da tela principal - Tabela
    public static final By TABLE_ORDER = By

.id("startForm:telaRecebimentoRomaneio3GdataTable:telaRecebimentoRomaneio3G
dataTablecolumn7header:sortDiv");
    public static final String TABLE =
"startForm:telaRecebimentoRomaneio3GdataTable:";
    public static final String TABLE_VALUE_CODIGO_RASTREAMENTO =
":telaRecebimentoRomaneio3GdataTablecolumn1otValue1";
    public static final String TABLE_VALUE_TOTAL_RECEBIDO =
":telaRecebimentoRomaneio3GdataTablecolumn3otValue3";
    public static final String TABLE_VALUE_TOTAL_ENVIADO =
":telaRecebimentoRomaneio3GdataTablecolumn4otValue4";
    public static final String TABLE_VALUE_TOTAL_DISPONIVEL =
":telaRecebimentoRomaneio3GdataTablecolumn5otValue5";
    public static final String TABLE_VALUE_ORIGEM =
":telaRecebimentoRomaneio3GdataTablecolumn7otValue7";
    public static final By TABLE_VALUE_CODIGO = By

.id("startForm:telaRecebimentoRomaneio3GdataTable:0:telaRecebimentoRomaneio
3GdataTablecolumn1otValue1");

    // Elementos do filtro
    public static final By FILTRO_LIMPAR = By
```

```
.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentfiltercontrolslkcle
arfilter");
    public static final By FILTRO_BUSCAR = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentfiltercontrolslkfil
ter");

    // Elementos da tela de impressão
    public static final By ETIQUETA_GERAR = By

.id("startForm:telaRecebimentoRomaneio3GpnGeracaoEtiquetaclGerar");

    // Elementos do formulário
    public static final By FORM_DT_COLHEITA = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnldataColheita
InputDate");
    public static final By FORM_ORIGEM = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn1origem");
    public static final By FORM_PRODUTO = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn1tipoProduto"
);
    public static final By FORM_QTDE = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn3quantidade")
;
    public static final By FORM_EMBALAGEM = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn3embalagem");
    public static final By FORM_NOTA_FISCAL = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn3notaFiscal")
;
    public static final By FORM_PRECO = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn3preco");
    public static final By FORM_PLACA_VEICULO = By
```

```
.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn1placaVeiculo
");
    public static final By FORM_DT_RECEBIMENTO = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn3dataEntregaI
nputDate");
    public static final By FORM_LOCAL_RECEBIMENTO = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn3destino");
    public static final By FORM_OBSERVACAO = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn3observacao")
;
    public static final By FORM_SALVAR = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformbtsalvarbt");
    public static final By FORM_NOME_MOTORISTA = By

.id("startForm:telaRecebimentoRomaneio3Gbasepanelcontentformpn1nomeMotorist
a");
    public static final By FORM_INSPECAO = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariot
ipoQuestionario");

    // Elementos do modal
    public static final By MODAL_CONFIRMA = By

.id("formModalPanel:telaRecebimentoRomaneio3GpnMsgSucessoCadastroReplicando
Dados");
    public static final By FORM_SUCESSO_REPLICAR_SIM = By

.id("formModalPanel:telaRecebimentoRomaneio3GpnMsgSucessoCadastroReplicando
DadosbtYesbt");
    public static final By FORM_SUCESSO_REPLICAR_NAO = By

.id("formModalPanel:telaRecebimentoRomaneio3GpnMsgSucessoCadastroReplicando
DadosbtNobt");

    // Elementos do questionario
    public static final By FORM_CHECKLIST_1 = By
```

```
.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5806");
    public static final By FORM_CHECKLIST_2 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5807");
    public static final By FORM_CHECKLIST_3 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5808");
    public static final By FORM_CHECKLIST_4 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5814");
    public static final By FORM_CHECKLIST_5 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5813");
    public static final By FORM_TOPICO_2 = By

.xpath("//*[@id='startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQ
uestionariopnQuestoespbi_topico3317']/div[1]");
    public static final By FORM_CHECKLIST_6 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5809");
    public static final By FORM_CHECKLIST_7 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5812");
    public static final By FORM_CHECKLIST_8 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5817");
    public static final By FORM_CHECKLIST_9 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5818");
    public static final By FORM_CHECKLIST_10 = By
```

```
.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5850");
    public static final By FORM_CHECKLIST_11 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5819");
    public static final By FORM_CHECKLIST_12 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5851");
    public static final By FORM_TOPICO_3 = By

.xpath("//*[@id='startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQ
uestionariopnQuestoespbi_topico3318']/div[1]");
    public static final By FORM_CHECKLIST_13 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5810");
    public static final By FORM_CHECKLIST_14 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5816");
    public static final By FORM_CHECKLIST_15 = By

.id("startForm:telaRecebimentoRomaneio3GbasepanelcontentformpnQuestionariop
nQuestoescp_questao5811");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Origem] é obrigatório." + "\n"
          + "O campo [Produto] é obrigatório." + "\n"
          + "O campo [Quantidade] é obrigatório." + "\n"
          + "O campo [Embalagem] é obrigatório." + "\n"
          + "O campo [Local de recebimento] é obrigatório.";
}

package perfil.geral.rastreamento.processamentoProduto;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.rastreamento.recebimento.ElementsRecebimento;
import perfil.geral.rastreamento.recebimento.TestRecebimento;
import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestProcessamentoProduto {
	WebDriver driver;
	String dataAtual = Util.dataAtual();
	public static String codigoProcessamento = null;
	public static BigDecimal quantidadeProcessada = new BigDecimal(15);

	@BeforeClass(alwaysRun = true)
	public void setUpProcessamentoProduto() {

	Util.startTestMsg(TestProcessamentoProduto.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_PRODUTO);
	}

	@Test(groups = { "insert" })
	public void setUpInsert() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.NOVO_REGISTRO));


driver.findElement(ElementsProcessamentoProduto.NOVO_REGISTRO).click();
	}
```

```
	@Test(groups = { "insert" }, dependsOnMethods = "setUpInsert")
	public void insertUniqueProcessamentoProduto() {
		Util.wait
				.until(ExpectedConditions.visibilityOfElementLocated
(ElementsProcessamentoProduto.FORM_DT_PROCESSAMENTO));


driver.findElement(ElementsProcessamentoProduto.FORM_DT_PROCESSAMENTO).send
Keys(dataAtual);


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_INICIAL).sendKeys
("13:00");


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_FINAL).sendKeys("
15:00");
		Util.selectOption(ElementsProcessamentoProduto.FORM_RECEITA,
Padroes.NOME_RECEITA);
		Util.wait.until(ExpectedConditions
				.visibilityOfElementLocated(ElementsProcessamentoPro
duto.FORM_INSUMOS_ADICIONADO));

	Util.selectOption(ElementsProcessamentoProduto.FORM_INGREDIENTE, 1);
		Util.waitJsBlock();


driver.findElement(ElementsProcessamentoProduto.FORM_QTDE).sendKeys(quantid
adeProcessada.toString());
		TestRecebimento.totalEstoqueSecondRecebimento.set(1,


TestRecebimento.totalEstoqueSecondRecebimento.get(1).add(quantidadeProcessa
da));
		TestRecebimento.totalEstoqueSecondRecebimento.set(2,
TestRecebimento.totalEstoqueSecondRecebimento.get(2)
				.subtract(quantidadeProcessada));
		Util.scrollUp();


driver.findElement(ElementsProcessamentoProduto.FORM_ADICIONAR_INGREDIENTES
```

```
).click();
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsProcessamentoPro
duto.FORM_INGREDIENTE_ADICIONADO));


      Util.selectOption(ElementsProcessamentoProduto.FORM_LOCAL_SAIDA,
Padroes.NOME_MEU_CADASTRO);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.FORM_ADICIONAR_CARGA));


driver.findElement(ElementsProcessamentoProduto.FORM_ADICIONAR_CARGA).click
();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.FORM_DESTINO));
            Util.selectOption(ElementsProcessamentoProduto.FORM_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsProcessamentoPro
duto.FORM_QUESTOES_QUESTIONARIO));


driver.findElement(ElementsProcessamentoProduto.FORM_QTDE_CARGA).sendKeys("
10");
            Util.selectOption(ElementsProcessamentoProduto.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsProcessamentoProduto.FORM_DT_ENTREGA).sendKeys(d
ataAtual);


driver.findElement(ElementsProcessamentoProduto.FORM_COMPLEMENTO).sendKeys(
"_Complemento WebDriver");
```

```
driver.findElement(ElementsProcessamentoProduto.FORM_HORA_INICIAL_CARGA).se
ndKeys("15:30");


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_FINAL_CARGA).send
Keys("16:30");


driver.findElement(ElementsProcessamentoProduto.FORM_NOTA_FISCAL).sendKeys(
Padroes.NOTA_FISCAL);


driver.findElement(ElementsProcessamentoProduto.FORM_PRECO).sendKeys("10,63
");


driver.findElement(ElementsProcessamentoProduto.FORM_PLACA_VEICULO).sendKey
s("IGI 2358");


driver.findElement(ElementsProcessamentoProduto.FORM_NOME_MOTORISTA).sendKe
ys("Monteiro Lobato");
            String[] respostas_tres = { "Fora do Padrão", "100%", "Acima de
51%", "Não se aplica", "0-10%", "0-15%",
                        "Acima de 7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamentoProduto.NOME_TELA,
                        respostas_tres);


driver.findElement(ElementsProcessamentoProduto.FORM_INCLUIR_CARGA).click()
;


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));



driver.findElement(ElementsProcessamentoProduto.FORM_ADICIONAR_CARGA).click
();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.FORM_DESTINO));
            Util.selectOption(ElementsProcessamentoProduto.FORM_DESTINO,
Padroes.NOME_DESTINO_CADASTRO);
            Util.waitJsBlock();


driver.findElement(ElementsProcessamentoProduto.FORM_QTDE_CARGA).sendKeys("
20");
            Util.selectOption(ElementsProcessamentoProduto.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsProcessamentoProduto.FORM_DT_ENTREGA).sendKeys(d
ataAtual);


driver.findElement(ElementsProcessamentoProduto.FORM_COMPLEMENTO).sendKeys(
"_Complemento WebDriver");


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_INICIAL_CARGA).se
ndKeys("17:30");


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_FINAL_CARGA).send
Keys("18:30");


driver.findElement(ElementsProcessamentoProduto.FORM_NOTA_FISCAL).sendKeys(
Padroes.NOTA_FISCAL);


driver.findElement(ElementsProcessamentoProduto.FORM_PRECO).sendKeys("25,66
");


driver.findElement(ElementsProcessamentoProduto.FORM_PLACA_VEICULO).sendKey
s("MAE 3658");
```

```
driver.findElement(ElementsProcessamentoProduto.FORM_NOME_MOTORISTA).sendKe
ys("Castro Alves");


driver.findElement(ElementsProcessamentoProduto.FORM_INCLUIR_CARGA).click()
;


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));

      driver.findElement(ElementsProcessamentoProduto.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);
            codigoProcessamento =
Util.capturaCodigo(driver.findElement(ElementsRastreador.MODAL_CONFIRMA).ge
tText());
            Util.testMsg("TestProcessamentoProduto", "Código: " +
codigoProcessamento);

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.NOVO_REGISTRO));
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertUniqueProcessamentoProduto")
      public void verifyEstoqueRecebimentoAfterProcessamentoProduto() {
            List<String> valorEstoqueTabelaSecondRecebimento = new
```

```
ArrayList<String>();

            driver.findElement(ElementsRastreador.RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));

            // Executa asserts do segundo recebimento (recebimento,
enviado,
            // disponível).


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        TestRecebimento.codigosRecebimento.get(1)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(0),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(0).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(1),
```

```
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(1).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(2),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(2).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        + valorEstoqueTabelaSecondRecebimento.get(0) + " |
Env: " + valorEstoqueTabelaSecondRecebimento.get(1)
                        + " | Disp: " +
valorEstoqueTabelaSecondRecebimento.get(2) + " } / RESULT");

            // Impressão das mensagens para verificação de estoque
            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(0).setScale(2) + " | Env:
"
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(1).setScale(2) + " |
Disp: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(2).setScale(2) + " } /
EXPECTED");
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamentoProduto.TABLE,
ElementsProcessamentoProduto.TABLE_VALUE,
                        Padroes.NOME_RECEITA + Padroes.EDITADO,
ElementsRastreador.TABLE_EDITAR);
      }
```

```
      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueProcessamentoProduto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.FORM_DT_PROCESSAMENTO));


driver.findElement(ElementsProcessamentoProduto.FORM_DT_PROCESSAMENTO).clea
r();


driver.findElement(ElementsProcessamentoProduto.FORM_DT_PROCESSAMENTO).send
Keys(dataAtual);


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_INICIAL).clear();


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_INICIAL).sendKeys
("14:00");


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_FINAL).clear();


driver.findElement(ElementsProcessamentoProduto.FORM_HORA_FINAL).sendKeys("
16:00");


driver.findElement(ElementsProcessamentoProduto.FORM_EDITAR_CARGA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.FORM_QTDE_CARGA));


driver.findElement(ElementsProcessamentoProduto.FORM_QTDE_CARGA).clear();


driver.findElement(ElementsProcessamentoProduto.FORM_QTDE_CARGA).sendKeys("
```

```
20");

driver.findElement(ElementsProcessamentoProduto.FORM_DT_ENTREGA).clear();

driver.findElement(ElementsProcessamentoProduto.FORM_DT_ENTREGA).sendKeys(d
ataAtual);

driver.findElement(ElementsProcessamentoProduto.FORM_COMPLEMENTO).clear();

driver.findElement(ElementsProcessamentoProduto.FORM_COMPLEMENTO).sendKeys(
Padroes.EDITADO);

driver.findElement(ElementsProcessamentoProduto.FORM_HORA_INICIAL_CARGA).cl
ear();

driver.findElement(ElementsProcessamentoProduto.FORM_HORA_INICIAL_CARGA).se
ndKeys("17:30");

driver.findElement(ElementsProcessamentoProduto.FORM_HORA_FINAL_CARGA).clea
r();

driver.findElement(ElementsProcessamentoProduto.FORM_HORA_FINAL_CARGA).send
Keys("19:30");

driver.findElement(ElementsProcessamentoProduto.FORM_NOTA_FISCAL).clear();

driver.findElement(ElementsProcessamentoProduto.FORM_NOTA_FISCAL).sendKeys(
Padroes.NOTA_FISCAL_EDITADO);

      driver.findElement(ElementsProcessamentoProduto.FORM_PRECO).clear();
```

```
driver.findElement(ElementsProcessamentoProduto.FORM_PRECO).sendKeys("20,25
");


driver.findElement(ElementsProcessamentoProduto.FORM_PLACA_VEICULO).clear()
;


driver.findElement(ElementsProcessamentoProduto.FORM_PLACA_VEICULO).sendKey
s("ICQ 9961");


driver.findElement(ElementsProcessamentoProduto.FORM_NOME_MOTORISTA).clear(
);


driver.findElement(ElementsProcessamentoProduto.FORM_NOME_MOTORISTA).sendKe
ys("Mario Quintana");


driver.findElement(ElementsProcessamentoProduto.FORM_ATUALIZAR_CARGA).click
();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));

      driver.findElement(ElementsProcessamentoProduto.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.NOVO_REGISTRO));
      }
```

```
	@Test(groups = { "delete" })
	public void deleteUniqueProcessamentoProduto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.NOVO_REGISTRO));
		Util.acionaElemento(ElementsProcessamentoProduto.TABLE,
ElementsProcessamentoProduto.TABLE_VALUE,
				Padroes.NOME_RECEITA + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
		Util.excluirRegistro(true, true);
	}

	@Test(groups = { "verify", "campos" })
	public void
validacaoCamposObrigatoriosProcessamentoProdutoIngrediente() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.NOVO_REGISTRO));



driver.findElement(ElementsProcessamentoProduto.NOVO_REGISTRO).click();
		Util.wait.until(ExpectedConditions
				.visibilityOfElementLocated(ElementsProcessamentoPro
duto.FORM_ADICIONAR_INGREDIENTES));



driver.findElement(ElementsProcessamentoProduto.FORM_ADICIONAR_INGREDIENTES
).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoProduto.MENSAGEM_ERRO));



Assert.assertEquals(driver.findElement(ElementsProcessamentoProduto.MENSAGE
M_ERRO).getText(),
```

```
      ElementsProcessamentoProduto.MSG_VALIDA_CAMPOS_INGREDIENTE,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosProcessamentoProdutoIngrediente")
      public void validacaoCamposObrigatoriosProcessamentoProdutoGeral() {

      driver.findElement(ElementsProcessamentoProduto.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsProcessamentoProduto.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_PRODUTO);

      Util.endTestMsg(TestProcessamentoProduto.class.getSimpleName());
      }
}

package perfil.geral.rastreamento.processamentoProduto;

import org.openqa.selenium.By;

public class ElementsProcessamentoProduto {

      // Elementos da tela principal - Botões
```

```
	public static final By NOVO_REGISTRO = By
			.id("startForm:telaProcessamentoSemDestinobasepanelconten
tlistgiNovoRegistro");

	// Elementos da tela principal - Tabelas
	public static final String TABLE =
"startForm:telaProcessamentoSemDestinodataTable:";
	public static final String TABLE_VALUE =
":telaProcessamentoSemDestinodataTablecolumn4otValue4";
	public static final By MENSAGEM_ERRO =
By.id("startForm:telaProcessamentoSemDestinopnFormotMsgErro");

	// Elementos do formulário - Campos
	public static final By FORM_DT_PROCESSAMENTO = By
			.id("startForm:telaProcessamentoSemDestinopnFormdataProce
ssamentoInputDate");
	public static final By FORM_HORA_INICIAL =
By.id("startForm:telaProcessamentoSemDestinopnFormhoraInicial");
	public static final By FORM_HORA_FINAL =
By.id("startForm:telaProcessamentoSemDestinopnFormhoraFinal");
	public static final By FORM_QTDE =
By.id("startForm:telaProcessamentoSemDestinopnFormqtdCodigoProcessamento");
	public static final By FORM_QTDE_INSUMO = By
			.id("startForm:telaProcessamentoSemDestinodataTableInsumo
s:0:itQtdeProcessamento");
	public static final By FORM_LOTE =
By.id("startForm:telaProcessamentoSemDestinodataTableInsumos:0:itLote");
	public static final By FORM_QTDE_CARGA = By
			.id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraisquantidade");
	public static final By FORM_COMPLEMENTO = By
			.id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraiscomplemento");
	public static final By FORM_HORA_INICIAL_CARGA = By
			.id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraishoraInicialCarga");
	public static final By FORM_HORA_FINAL_CARGA = By
			.id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraishoraFinalCarga");
	public static final By FORM_NOTA_FISCAL = By
			.id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
```

```
adosGeraisnotaFiscal");
      public static final By FORM_PRECO =
By.id("formModalPanel:telaProcessamentoSemDestinopnCargapnDadosGeraispreco"
);
      public static final By FORM_PLACA_VEICULO = By
                  .id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraisplacaVeiculo");
      public static final By FORM_NOME_MOTORISTA = By
                  .id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraisnomeMotorista");
      public static final String NOME_TELA =
"telaProcessamentoSemDestinopnCargapn";

      // Elementos do formulário - Selects
      public static final By FORM_LINHA_PRODUCAO =
By.id("startForm:telaProcessamentoSemDestinopnFormlinhaProducao");
      public static final By FORM_TURNO =
By.id("startForm:telaProcessamentoSemDestinopnFormselectTurno");
      public static final By FORM_RECEITA =
By.id("startForm:telaProcessamentoSemDestinopnFormtipoProduto");
      public static final By FORM_INGREDIENTE =
By.id("startForm:telaProcessamentoSemDestinoingrediente");
      public static final By FORM_FORNECEDOR = By
                  .id("startForm:telaProcessamentoSemDestinodataTableInsumo
s:0:smFornecedor");
      public static final By FORM_LOCAL_SAIDA =
By.id("startForm:telaProcessamentoSemDestinopnFormdestino");
      public static final By FORM_DESTINO =
By.id("formModalPanel:telaProcessamentoSemDestinopnCargapnDadosGeraisdestin
o");
      public static final By FORM_EMBALAGEM = By
                  .id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraisembalagem");
      public static final By FORM_DT_ENTREGA = By
                  .id("formModalPanel:telaProcessamentoSemDestinopnCargapnD
adosGeraisdataEntregaInputDate");

      // Elementos do formulário - Botões
      public static final By FORM_ADICIONAR_INGREDIENTES = By
                  .id("startForm:telaProcessamentoSemDestinobtAdicionarIngr
edientebt");
```

```
	public static final By FORM_ADICIONAR_CARGA = By
			.id("startForm:telaProcessamentoSemDestinobasepanelconten
tformpnNovoRegistrogiNovoRegistro");
	public static final By FORM_INCLUIR_CARGA = By
			.id("formModalPanel:telaProcessamentoSemDestinopnCargapnB
otoesbtIncluirCargabt");
	public static final By FORM_SALVAR =
By.id("startForm:telaProcessamentoSemDestinobasepanelcontentformbtsalvarbt"
);
	public static final By FORM_EDITAR_CARGA = By
			.id("startForm:telaProcessamentoSemDestinodataTableCargas
:0:telaProcessamentoSemDestinodataTableCargaspnEditDeletegiEdit");
	public static final By FORM_ATUALIZAR_CARGA = By
			.id("formModalPanel:telaProcessamentoSemDestinopnCargapnB
otoesbtAtualizarCargabt");
	public static final By FORM_INSUMOS_ADICIONADO = By
			.id("startForm:telaProcessamentoSemDestinodataTableInsumo
s:0:telaProcessamentoSemDestinodataTableInsumoscolumn0otValue0");
	public static final By FORM_INGREDIENTE_ADICIONADO =
By.id("startForm:telaProcessamentoSemDestinodataTableIngredientes:0:telaPro
cessamentoSemDestinodataTableIngredientescolumn0otValue0");
	public static final By FORM_QUESTOES_QUESTIONARIO =
By.id("formModalPanel:telaProcessamentoSemDestinopnCargapnQuestionariopnQue
stoespbi_topico1pg");

	// Mensagem de validação dos campos obrigatorios
	public static String MSG_VALIDA_CAMPOS_INGREDIENTE = "[ERRO] Os
campos Tipo Ingrediente, Ingrediente, Quantidade são obrigatórios";
	public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
[Receita] é obrigatório." + "\n"
			+ "O campo [Ingredientes] é obrigatório." + "\n" + "O
campo [Cargas] é obrigatório.";
}


package perfil.geral.rastreamento.processamentoPescado;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.rastreamento.recebimento.ElementsRecebimento;
import perfil.geral.rastreamento.recebimento.TestRecebimento;
import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestProcessamentoPescado {
      WebDriver driver;
      String dataAtual = Util.dataAtual();
      public static String codigoProcessamento = null;
      public static BigDecimal quantidadeProcessada = new BigDecimal(15);

      @BeforeClass(alwaysRun = true)
      public void setUpProcessamentoPescado() {

      Util.startTestMsg(TestProcessamentoPescado.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_PESCADO);
      }

      @Test(groups = { "insert" })
      public void setUpInsert() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.NOVO_REGISTRO));


driver.findElement(ElementsProcessamentoPescado.NOVO_REGISTRO).click();
      }

      @Test(groups = { "insert" }, dependsOnMethods = "setUpInsert")
```

```
      public void insertUniqueProcessamentoPescado() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.FORM_PRODUTO));
            Util.selectOption(ElementsProcessamentoPescado.FORM_PRODUTO,
Padroes.NOME_PRODUTO);
            Util.waitJsBlock();


Util.selectOption(ElementsProcessamentoPescado.FORM_CODIGO_RECEBIMENTO, 1);


driver.findElement(ElementsProcessamentoPescado.FORM_DT_PROCESSAMENTO).send
Keys(dataAtual);


driver.findElement(ElementsProcessamentoPescado.FORM_OBSERVACAO).sendKeys(P
adroes.OBSERVACAO);


Util.selectOption(ElementsProcessamentoPescado.FORM_PRODUTO_RESULTANTE,
Padroes.NOME_RECEITA);



driver.findElement(ElementsProcessamentoPescado.FORM_QTDE).sendKeys(quantid
adeProcessada.toString());
            TestRecebimento.totalEstoqueSecondRecebimento.set(1,
TestRecebimento.totalEstoqueSecondRecebimento.get(0));
            TestRecebimento.totalEstoqueSecondRecebimento.set(2, new
BigDecimal(0));

            Util.selectOption(ElementsProcessamentoPescado.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsProcessamentoPescado.FORM_DT_ENTREGA).sendKeys(d
ataAtual);
            Util.selectOption(ElementsProcessamentoPescado.FORM_DESTINO,
Padroes.NOME_DESTINO_CADASTRO);
```

```
            Util.scrollUp();


driver.findElement(ElementsProcessamentoPescado.FORM_ADICIONAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.FORM_PRIMEIRA_SAIDA));


      driver.findElement(ElementsProcessamentoPescado.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.NOVO_REGISTRO));
            codigoProcessamento =
Util.capturaCodigo(driver.findElement(ElementsProcessamentoPescado.CODIGO_R
ASTREAMENTO)
                        .getText());
            Util.testMsg("TestProcessamentoPescado", "Código: " +
codigoProcessamento);

            // Impressão das mensagens para verificação de estoque
            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(0).setScale(2) + " | Env:
"
                        +
```

```
TestRecebimento.totalEstoqueSecondRecebimento.get(1).setScale(2) + " |
Disp: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(2).setScale(2) + " } /
EXPECTED");
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertUniqueProcessamentoPescado")
      public void verifyEstoqueRecebimentoAfterProcessamentoPescado() {
            List<String> valorEstoqueTabelaSecondRecebimento = new
ArrayList<String>();

            driver.findElement(ElementsRastreador.RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));

            // Executa asserts do segundo recebimento (recebimento,
enviado,
            // disponível).


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
```

```
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        TestRecebimento.codigosRecebimento.get(1)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(0),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(0).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(1),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(1).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(2),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(2).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        + valorEstoqueTabelaSecondRecebimento.get(0) + " |
Env: " + valorEstoqueTabelaSecondRecebimento.get(1)
                        + " | Disp: " +
valorEstoqueTabelaSecondRecebimento.get(2) + " } / RESULT");
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamentoPescado.TABLE,
ElementsProcessamentoPescado.TABLE_VALUE,
                        Padroes.NOME_PRODUTO,
ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueProcessamentoPescado() {
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.FORM_DT_PROCESSAMENTO));


driver.findElement(ElementsProcessamentoPescado.FORM_DT_PROCESSAMENTO).clea
r();


driver.findElement(ElementsProcessamentoPescado.FORM_DT_PROCESSAMENTO).send
Keys(dataAtual);

      driver.findElement(ElementsProcessamentoPescado.FORM_QTDE).clear();


driver.findElement(ElementsProcessamentoPescado.FORM_QTDE).sendKeys("30");


driver.findElement(ElementsProcessamentoPescado.FORM_DT_ENTREGA).clear();


driver.findElement(ElementsProcessamentoPescado.FORM_DT_ENTREGA).sendKeys(d
ataAtual);

      driver.findElement(ElementsProcessamentoPescado.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueProcessamentoPescado() {
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamentoPescado.TABLE,
ElementsProcessamentoPescado.TABLE_VALUE,
                        Padroes.NOME_PRODUTO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void
validacaoCamposObrigatoriosProcessamentoPescadoProdutosResultantes() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.NOVO_REGISTRO));



driver.findElement(ElementsProcessamentoPescado.NOVO_REGISTRO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.FORM_ADICIONAR));



driver.findElement(ElementsProcessamentoPescado.FORM_ADICIONAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoPescado.MENSAGEM_ERRO));



Assert.assertEquals(driver.findElement(ElementsProcessamentoPescado.MENSAGE
M_ERRO).getText(),

      ElementsProcessamentoPescado.MSG_VALIDA_CAMPOS_PRODUTO,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);
```

```
	}

	@Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosProcessamentoPescadoProdutosResultantes")
	public void validacaoCamposObrigatoriosProcessamentoPescadoGeral() {

	driver.findElement(ElementsProcessamentoPescado.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
			ElementsProcessamentoPescado.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

	driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
	}

	@AfterClass(alwaysRun = true)
	public void tearDown() {
		Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_PESCADO);

	Util.endTestMsg(TestProcessamentoPescado.class.getSimpleName());
	}
}

package perfil.geral.rastreamento.processamentoPescado;

import org.openqa.selenium.By;

public class ElementsProcessamentoPescado {

	// Elementos da tela principal - Botões
	public static final By NOVO_REGISTRO = By
			.id("startForm:telaProcessamentoPescado3Gbasepanelcontent
listgiNovoRegistro");
```

```
	// Elementos da tela principal - Tabelas
	public static final String TABLE =
"startForm:telaProcessamentoPescado3GdataTable:";
	public static final String TABLE_VALUE =
":telaProcessamentoPescado3GdataTablecolumn2otValue2";
	public static final By TABLE_VALUE_CODIGO = By
			.id("startForm:telaProcessamentoPescado3GdataTable:0:tela
ProcessamentoPescado3GdataTablecolumn1otValue1");
	public static final By MENSAGEM_ERRO =
By.id("startForm:telaProcessamentoPescado3GbasepanelcontentformpnFormotErro
");

	// Elementos do formulário - Campos
	public static final By FORM_DT_PROCESSAMENTO = By
			.id("startForm:telaProcessamentoPescado3Gbasepanelcontent
formpnFormdataProcessamentoInputDate");
	public static final By FORM_OBSERVACAO = By
			.id("startForm:telaProcessamentoPescado3Gbasepanelcontent
formpnFormobservacao");
	public static final By FORM_QTDE =
By.id("startForm:telaProcessamentoPescado3GbasepanelcontentformpnFormquanti
dade");

	// Elementos do formulário - Selects
	public static final By FORM_PRODUTO = By
			.id("startForm:telaProcessamentoPescado3Gbasepanelcontent
formpnFormtipoProduto");
	public static final By FORM_CODIGO_RECEBIMENTO = By
			.id("startForm:telaProcessamentoPescado3Gbasepanelcontent
formpnFormcodigosDisponiveis");
	public static final By FORM_PRODUTO_RESULTANTE = By
			.id("startForm:telaProcessamentoPescado3Gbasepanelcontent
formpnFormprodutoResultante");
	public static final By FORM_DESTINO =
By.id("startForm:telaProcessamentoPescado3GbasepanelcontentformpnFormdestin
o");
	public static final By FORM_EMBALAGEM = By
			.id("startForm:telaProcessamentoPescado3Gbasepanelcontent
formpnFormembalagem");
	public static final By FORM_DT_ENTREGA = By
```

```
                .id("startForm:telaProcessamentoPescado3Gbasepanelcontent
formpnFormdataEntregaInputDate");

      // Elementos do formulário - Botões
      public static final By FORM_ADICIONAR =
By.id("startForm:telaProcessamentoPescado3GbtAdicionarbt");
      public static final By FORM_SALVAR =
By.id("startForm:telaProcessamentoPescado3Gbasepanelcontentformbtsalvarbt")
;

      public static final By CODIGO_RASTREAMENTO = By
                .id("startForm:telaProcessamentoPescado3GdataTable:0:tela
ProcessamentoPescado3GdataTablecolumn1otValue1");

      public static final By FORM_PRIMEIRA_SAIDA =
By.id("startForm:telaProcessamentoPescado3GdataTableProcessados:0:telaProce
ssamentoPescado3GdataTableProcessadoscolumn0otValue0");

      // Mensagem de validação dos campos obrigatorios
      public static String MSG_VALIDA_CAMPOS_PRODUTO = "Preencha todos os
campos de produtos resultantes.";
      public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
[Código de recebimento] é obrigatório." + "\n"
                + "O campo [Saídas] é obrigatório.";

}

package perfil.geral.rastreamento.processamentoInsumo;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;
```

```
public class TestProcessamentoInsumo {
	WebDriver driver;
	String dataAtual = Util.dataAtual();
	public static String codigoProcessamento = null;

	@BeforeClass(alwaysRun = true)
	public void setUpProcessamentoInsumo() {

	Util.startTestMsg(TestProcessamentoInsumo.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_INSUMO);
	}

	@Test(groups = { "insert", "verify", "campos" })
	public void abreTelaInsertProcessamentoInsumo() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.NOVO_REGISTRO));


driver.findElement(ElementsProcessamentoInsumo.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertProcessamentoInsumo")
	public void insertUniqueProcessamentoInsumo() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.FORM_DT_PROCESSAMENTO));


driver.findElement(ElementsProcessamentoInsumo.FORM_DT_PROCESSAMENTO).sendK
eys(dataAtual);


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_INICIAL).sendKeys(
"13:00");
```

```
driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_FINAL).sendKeys("1
5:00");
            Util.selectOption(ElementsProcessamentoInsumo.FORM_RECEITA,
Padroes.NOME_RECEITA);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.FORM_PRIMEIRO_INSUMO));


driver.findElement(ElementsProcessamentoInsumo.FORM_QTDE_INSUMO).sendKeys("
15");
            Util.selectOption(ElementsProcessamentoInsumo.FORM_FORNECEDOR,
Padroes.NOME_FORNECEDOR);


driver.findElement(ElementsProcessamentoInsumo.FORM_LOTE).sendKeys("1");
            Util.selectOption(ElementsProcessamentoInsumo.FORM_LOCAL_SAIDA,
Padroes.NOME_MEU_CADASTRO);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.FORM_ADICIONAR_CARGA));


driver.findElement(ElementsProcessamentoInsumo.FORM_ADICIONAR_CARGA).click(
);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.FORM_DESTINO));
            Util.selectOption(ElementsProcessamentoInsumo.FORM_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsProcessamentoIns
umo.FORM_QUESTOES_QUESTIONARIO));


driver.findElement(ElementsProcessamentoInsumo.FORM_QTDE_CARGA).sendKeys("1
```

```
0");
            Util.selectOption(ElementsProcessamentoInsumo.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsProcessamentoInsumo.FORM_DT_ENTREGA).sendKeys(da
taAtual);


driver.findElement(ElementsProcessamentoInsumo.FORM_COMPLEMENTO).sendKeys("
_Complemento WebDriver");


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_INICIAL_CARGA).sen
dKeys("20:30");


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_FINAL_CARGA).sendK
eys("21:30");


driver.findElement(ElementsProcessamentoInsumo.FORM_NOTA_FISCAL).sendKeys(P
adroes.NOTA_FISCAL);


driver.findElement(ElementsProcessamentoInsumo.FORM_PRECO).sendKeys("10,63"
);


driver.findElement(ElementsProcessamentoInsumo.FORM_PLACA_VEICULO).sendKeys
("MII 6587");


driver.findElement(ElementsProcessamentoInsumo.FORM_NOME_MOTORISTA).sendKey
s("Castro Alves");
            String[] respostas_um = { "De Acordo", "Não se aplica", "Abaixo
de 51%", "0-20%", "Acima de 31%", "16-30%",
                        "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamentoInsumo.NOME_TELA, respostas_um);
```

```
driver.findElement(ElementsProcessamentoInsumo.FORM_INCLUIR_CARGA).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.waitJsBlock();

      driver.findElement(ElementsProcessamentoInsumo.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);
            codigoProcessamento =
Util.capturaCodigo(driver.findElement(ElementsRastreador.MODAL_CONFIRMA).ge
tText());
            Util.testMsg("TestProcessamentoInsumo", "Código: " +
codigoProcessamento);

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.NOVO_REGISTRO));
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamentoInsumo.TABLE,
ElementsProcessamentoInsumo.TABLE_VALUE,
                        Padroes.NOME_RECEITA + Padroes.EDITADO,
```

```
ElementsRastreador.TABLE_EDITAR);
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
	public void editUniqueProcessamentoInsumo() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.FORM_DT_PROCESSAMENTO));


driver.findElement(ElementsProcessamentoInsumo.FORM_DT_PROCESSAMENTO).clear
();


driver.findElement(ElementsProcessamentoInsumo.FORM_DT_PROCESSAMENTO).sendK
eys(dataAtual);


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_INICIAL).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_INICIAL).sendKeys(
"16:00");


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_FINAL).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_FINAL).sendKeys("1
7:00");


driver.findElement(ElementsProcessamentoInsumo.FORM_QTDE_INSUMO).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_QTDE_INSUMO).sendKeys("
20");

	driver.findElement(ElementsProcessamentoInsumo.FORM_LOTE).clear();
```

```
driver.findElement(ElementsProcessamentoInsumo.FORM_LOTE).sendKeys("2");
            Util.scrollDown();


driver.findElement(ElementsProcessamentoInsumo.FORM_EDITAR_CARGA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.FORM_QTDE_CARGA));


driver.findElement(ElementsProcessamentoInsumo.FORM_QTDE_CARGA).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_QTDE_CARGA).sendKeys("2
5");


driver.findElement(ElementsProcessamentoInsumo.FORM_DT_ENTREGA).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_DT_ENTREGA).sendKeys(da
taAtual);


driver.findElement(ElementsProcessamentoInsumo.FORM_COMPLEMENTO).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_COMPLEMENTO).sendKeys(P
adroes.EDITADO);


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_INICIAL_CARGA).cle
ar();


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_INICIAL_CARGA).sen
dKeys("22:30");
```

```
driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_FINAL_CARGA).clear
();


driver.findElement(ElementsProcessamentoInsumo.FORM_HORA_FINAL_CARGA).sendK
eys("23:30");


driver.findElement(ElementsProcessamentoInsumo.FORM_NOTA_FISCAL).sendKeys(P
adroes.NOTA_FISCAL_EDITADO);

      driver.findElement(ElementsProcessamentoInsumo.FORM_PRECO).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_PRECO).sendKeys("22,32"
);


driver.findElement(ElementsProcessamentoInsumo.FORM_PLACA_VEICULO).clear();


driver.findElement(ElementsProcessamentoInsumo.FORM_PLACA_VEICULO).sendKeys
("MAE 1011");


driver.findElement(ElementsProcessamentoInsumo.FORM_NOME_MOTORISTA).clear()
;


driver.findElement(ElementsProcessamentoInsumo.FORM_NOME_MOTORISTA).sendKey
s("Guimarães Rosa ");
            String[] respostas_dois = { "Fora do Padrão", "Não se aplica",
"Não se aplica", "0-20%", "0-10%", "0-15%", "4-7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamentoInsumo.NOME_TELA, respostas_dois);


driver.findElement(ElementsProcessamentoInsumo.FORM_ATUALIZAR_CARGA).click(
);
```

```
Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));

      driver.findElement(ElementsProcessamentoInsumo.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueProcessamentoInsumo() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamentoInsumo.TABLE,
ElementsProcessamentoInsumo.TABLE_VALUE,
                        Padroes.NOME_RECEITA + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertProcessamentoInsumo")
      public void validacaoCamposObrigatoriosProcessamentoInsumo() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoInsumo.FORM_SALVAR));
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {

      driver.findElement(ElementsProcessamentoInsumo.FORM_SALVAR).click();
```

```
                        return
driver.findElement(ElementsRastreador.MODAL_ERRO_TITLE).isDisplayed();
                  }
            });


Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsProcessamentoInsumo.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_INSUMO);
            Util.endTestMsg(TestProcessamentoInsumo.class.getSimpleName());
      }
}

package perfil.geral.rastreamento.processamentoInsumo;

import org.openqa.selenium.By;

public class ElementsProcessamentoInsumo {

      // Elementos da tela principal - Tabelas
      public static final String TABLE =
"startForm:telaProcessamentoInsumosdataTable:";
      public static final String TABLE_VALUE =
":telaProcessamentoInsumosdataTablecolumn4otValue4";

      // Elementos da tela principal - Botões
      public static final By NOVO_REGISTRO =
By.id("startForm:telaProcessamentoInsumosbasepanelcontentlistgiNovoRegistro
");

      // Elementos do formulário - Campos
      public static final By FORM_DT_PROCESSAMENTO = By
```

```
                .id("startForm:telaProcessamentoInsumospnFormdataProcessa
mentoInputDate");
      public static final By FORM_HORA_INICIAL =
By.id("startForm:telaProcessamentoInsumospnFormhoraInicial");
      public static final By FORM_HORA_FINAL =
By.id("startForm:telaProcessamentoInsumospnFormhoraFinal");
      public static final By FORM_QTDE_INSUMO = By
                .id("startForm:telaProcessamentoInsumosdataTableInsumos:0
:itQtdeProcessamento");
      public static final By FORM_LOTE =
By.id("startForm:telaProcessamentoInsumosdataTableInsumos:0:itLote");
      public static final By FORM_QTDE_CARGA = By
                .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraisquantidade");
      public static final By FORM_COMPLEMENTO = By
                .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraiscomplemento");
      public static final By FORM_HORA_INICIAL_CARGA = By
                .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraishoraInicialCarga");
      public static final By FORM_HORA_FINAL_CARGA = By
                .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraishoraFinalCarga");
      public static final By FORM_NOTA_FISCAL = By
                .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraisnotaFiscal");
      public static final By FORM_PRECO =
By.id("formModalPanel:telaProcessamentoInsumospnCargapnDadosGeraispreco");
      public static final By FORM_PLACA_VEICULO = By
                .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraisplacaVeiculo");
      public static final By FORM_NOME_MOTORISTA = By
                .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraisnomeMotorista");
      public static final String NOME_TELA =
"telaProcessamentoInsumospnCargapn";
      public static final By FORM_PRIMEIRO_INSUMO =
By.id("startForm:telaProcessamentoInsumosdataTableInsumos:0:telaProcessamen
toInsumosdataTableInsumoscolumn0otValue0");
      public static final By FORM_QUESTOES_QUESTIONARIO =
By.id("formModalPanel:telaProcessamentoInsumospnCargapnQuestionariopnQuesto
```

```
espbi_topico1pg");

      // Elementos do formulário - Selects
      // public static final By FORM_LINHA_PRODUCAO = By
      // .id("startForm:telaProcessamentoInsumospnFormlinhaProducao");
      // public static final By FORM_TURNO = By
      // .id("startForm:telaProcessamentoInsumospnFormselectTurno");
      public static final By FORM_RECEITA =
By.id("startForm:telaProcessamentoInsumospnFormtipoProduto");
      public static final By FORM_FORNECEDOR =
By.id("startForm:telaProcessamentoInsumosdataTableInsumos:0:smFornecedor");
      public static final By FORM_LOCAL_SAIDA =
By.id("startForm:telaProcessamentoInsumospnFormdestino");
      public static final By FORM_DESTINO =
By.id("formModalPanel:telaProcessamentoInsumospnCargapnDadosGeraisdestino")
;
      public static final By FORM_EMBALAGEM = By
                  .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraisembalagem");
      public static final By FORM_DT_ENTREGA = By
                  .id("formModalPanel:telaProcessamentoInsumospnCargapnDado
sGeraisdataEntregaInputDate");

      // Elementos do formulário - Botões
      public static final By FORM_ADICIONAR_CARGA = By
                  .id("startForm:telaProcessamentoInsumosbasepanelcontentfo
rmpnNovoRegistrogiNovoRegistro");
      public static final By FORM_INCLUIR_CARGA = By
                  .id("formModalPanel:telaProcessamentoInsumospnCargapnBoto
esbtIncluirCargabt");
      public static final By FORM_SALVAR =
By.id("startForm:telaProcessamentoInsumosbasepanelcontentformbtsalvarbt");
      public static final By FORM_EDITAR_CARGA = By
                  .id("startForm:telaProcessamentoInsumosdataTableCargas:0:
telaProcessamentoInsumosdataTableCargaspnEditDeletegiEdit");
      public static final By FORM_ATUALIZAR_CARGA = By
                  .id("formModalPanel:telaProcessamentoInsumospnCargapnBoto
esbtAtualizarCargabt");

      // Mensagem de validação dos campos obrigatorios
      public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
```

```
[Receita] é obrigatório." + "\n"
                  + "O campo [Cargas] é obrigatório." + "\n" + "Preencha ao
menos um insumo.";
}

package perfil.geral.rastreamento.processamentoDireto;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestProcessamentoDireto {
      WebDriver driver;
      String dataAtual = Util.dataAtual();
      public static String codigoProcessamento = null;

      @BeforeClass(alwaysRun = true)
      public void setUpProcessamentoDireto() {

      Util.startTestMsg(TestProcessamentoDireto.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_DIRETO);
      }

      @Test(groups = { "insert", "verify", "campos" })
      public void abreTelaInsertProcessamentoDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.NOVO_REGISTRO));
```

```
driver.findElement(ElementsProcessamentoDireto.NOVO_REGISTRO).click();
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertProcessamentoDireto")
      public void insertUniqueProcessamentoDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.FORM_DT_PROCESSAMENTO));


driver.findElement(ElementsProcessamentoDireto.FORM_DT_PROCESSAMENTO).sendK
eys(dataAtual);
            Util.selectOption(ElementsProcessamentoDireto.FORM_RECEITA,
Padroes.NOME_RECEITA);
            Util.waitJsBlock();


driver.findElement(ElementsProcessamentoDireto.FORM_OBSERVACAO).sendKeys("O
bservação Processamento");


driver.findElement(ElementsProcessamentoDireto.FORM_ADICIONAR_INGREDIENTES)
.click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.FORM_DT_COLHEITA));


driver.findElement(ElementsProcessamentoDireto.FORM_DT_COLHEITA).sendKeys(d
ataAtual);
            Util.selectOption(ElementsProcessamentoDireto.FORM_PRODUTO,
Padroes.NOME_PRODUTO);
            Util.selectOption(ElementsProcessamentoDireto.FORM_ORIGEM,
Padroes.NOME_ORIGEM);
            Util.waitJsBlock();
```

```
driver.findElement(ElementsProcessamentoDireto.FORM_QUANTIDADE_INGREDIENTE)
.sendKeys("100");


Util.selectOption(ElementsProcessamentoDireto.FORM_EMBALAGEM_INGREDIENTE,
Padroes.NOME_EMBALAGEM);

      Util.selectOption(ElementsProcessamentoDireto.FORM_LOCAL_RECEBIMENTO,
Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsProcessamentoDireto.FORM_DT_RECEBIMENTO).sendKey
s(dataAtual);


driver.findElement(ElementsProcessamentoDireto.FORM_INCLUIR_INGREDIENTES).c
lick();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));



driver.findElement(ElementsProcessamentoDireto.FORM_ADICIONAR_CARGA).click(
);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.FORM_DESTINO));
            Util.selectOption(ElementsProcessamentoDireto.FORM_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsProcessamentoDir
eto.FORM_QUESTOES_QUESTIONARIO));


driver.findElement(ElementsProcessamentoDireto.FORM_QUANTIDADE_CARGA).sendK
eys("100");

      Util.selectOption(ElementsProcessamentoDireto.FORM_EMBALAGEM_CARGA,
```

```
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsProcessamentoDireto.FORM_DT_ENTREGA).sendKeys(da
taAtual);


driver.findElement(ElementsProcessamentoDireto.FORM_COMPLEMENTO).sendKeys("
_Complemento WebDriver");


driver.findElement(ElementsProcessamentoDireto.FORM_PLACA_VEICULO).sendKeys
("ERJ 7850");


driver.findElement(ElementsProcessamentoDireto.FORM_NOME_MOTORISTA).sendKey
s("Paz Moreno");


driver.findElement(ElementsProcessamentoDireto.FORM_NOTA_FISCAL).sendKeys(P
adroes.NOTA_FISCAL);


driver.findElement(ElementsProcessamentoDireto.FORM_PRECO).sendKeys("5.99")
;
            String[] respostas_um = { "De Acordo", "Não se aplica", "Abaixo
de 51%", "0-20%", "Acima de 31%", "16-30%",
                        "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamentoDireto.NOME_TELA, respostas_um);


driver.findElement(ElementsProcessamentoDireto.FORM_INCLUIR_CARGA).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));

      driver.findElement(ElementsProcessamentoDireto.FORM_SALVAR).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);
            codigoProcessamento =
Util.capturaCodigo(driver.findElement(ElementsRastreador.MODAL_CONFIRMA).ge
tText());
            Util.testMsg("TestProcessamentoDireto", "Código: " +
codigoProcessamento);


      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.NOVO_REGISTRO));
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamentoDireto.TABLE,
ElementsProcessamentoDireto.TABLE_VALUE,
                        Padroes.NOME_RECEITA + Padroes.EDITADO,
ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueProcessamentoDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.FORM_DT_PROCESSAMENTO));
```

```
driver.findElement(ElementsProcessamentoDireto.FORM_DT_PROCESSAMENTO).clear
();


driver.findElement(ElementsProcessamentoDireto.FORM_DT_PROCESSAMENTO).sendK
eys(dataAtual);


driver.findElement(ElementsProcessamentoDireto.FORM_OBSERVACAO).clear();


driver.findElement(ElementsProcessamentoDireto.FORM_OBSERVACAO).sendKeys(Pa
droes.EDITADO);
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsProcessamentoDireto.FORM_EDITAR_INGREDIENTE).cli
ck();
                        return
driver.findElement(ElementsProcessamentoDireto.FORM_DT_COLHEITA).isDisplaye
d();
                  }
            });


driver.findElement(ElementsProcessamentoDireto.FORM_DT_COLHEITA).clear();


driver.findElement(ElementsProcessamentoDireto.FORM_DT_COLHEITA).sendKeys(d
ataAtual);


driver.findElement(ElementsProcessamentoDireto.FORM_QUANTIDADE_INGREDIENTE)
.clear();


driver.findElement(ElementsProcessamentoDireto.FORM_QUANTIDADE_INGREDIENTE)
.sendKeys("50");
```

```
driver.findElement(ElementsProcessamentoDireto.FORM_DT_RECEBIMENTO).clear()
;


driver.findElement(ElementsProcessamentoDireto.FORM_DT_RECEBIMENTO).sendKey
s(dataAtual);


driver.findElement(ElementsProcessamentoDireto.FORM_ATUALIZAR_INGREDIENTES)
.click();
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsProcessamentoDireto.FORM_EDITAR_CARGA).click();
                        return
driver.findElement(ElementsProcessamentoDireto.FORM_QUANTIDADE_CARGA).isDis
played();
                  }
            });


driver.findElement(ElementsProcessamentoDireto.FORM_QUANTIDADE_CARGA).clear
();


driver.findElement(ElementsProcessamentoDireto.FORM_QUANTIDADE_CARGA).sendK
eys("50");


driver.findElement(ElementsProcessamentoDireto.FORM_DT_ENTREGA).clear();


driver.findElement(ElementsProcessamentoDireto.FORM_DT_ENTREGA).sendKeys(da
taAtual);


driver.findElement(ElementsProcessamentoDireto.FORM_COMPLEMENTO).clear();
```

```
driver.findElement(ElementsProcessamentoDireto.FORM_COMPLEMENTO).sendKeys(P
adroes.EDITADO);


driver.findElement(ElementsProcessamentoDireto.FORM_PLACA_VEICULO).clear();


driver.findElement(ElementsProcessamentoDireto.FORM_PLACA_VEICULO).sendKeys
("JRE 0587");


driver.findElement(ElementsProcessamentoDireto.FORM_NOME_MOTORISTA).clear()
;


driver.findElement(ElementsProcessamentoDireto.FORM_NOME_MOTORISTA).sendKey
s("Ana Paula Leme");

      driver.findElement(ElementsProcessamentoDireto.FORM_PRECO).clear();


driver.findElement(ElementsProcessamentoDireto.FORM_PRECO).sendKeys("7.52")
;
            String[] respostas_tres = { "Fora do Padrão", "100%", "Acima de
51%", "Não se aplica", "0-10%", "0-15%",
                        "Acima de 7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamentoDireto.NOME_TELA, respostas_tres);


driver.findElement(ElementsProcessamentoDireto.FORM_ATUALIZAR_CARGA).click(
);


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));

      driver.findElement(ElementsProcessamentoDireto.FORM_SALVAR).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueProcessamentoDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamentoDireto.TABLE,
ElementsProcessamentoDireto.TABLE_VALUE,
                        Padroes.NOME_RECEITA + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertProcessamentoDireto")
      public void validacaoCamposObrigatoriosProcessamentoDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamentoDireto.FORM_SALVAR));
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {

      driver.findElement(ElementsProcessamentoDireto.FORM_SALVAR).click();
                        return
driver.findElement(ElementsRastreador.MODAL_ERRO_TITLE).isDisplayed();
                  }
            });
```

```
Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsProcessamentoDireto.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO_DIRETO);
            Util.endTestMsg(TestProcessamentoDireto.class.getSimpleName());
      }
}

package perfil.geral.rastreamento.processamentoDireto;

import org.openqa.selenium.By;

public class ElementsProcessamentoDireto {

    // Elementos da tela principal - Botões
    public static final By NOVO_REGISTRO = By

.id("startForm:telaProcessamentoDiretobasepanelcontentlistgiNovoRegistro");
    public static final By FORM_SALVAR = By

.id("startForm:telaProcessamentoDiretobasepanelcontentformbtsalvarbt");

    // Elementos da tela principal - Tabela
    public static final String TABLE =
"startForm:telaProcessamentoDiretodataTable:";
    public static final String TABLE_VALUE =
":telaProcessamentoDiretodataTablecolumn2otValue2";

    // Elementos do formulário - Campos
    public static final By FORM_OBSERVACAO = By
          .id("startForm:telaProcessamentoDiretopnFormobservacao");
    public static final By FORM_QUANTIDADE_INGREDIENTE = By
```

```
.id("formModalPanel:telaProcessamentoDiretopnRecebimentopnquantidade");
    public static final By FORM_QUANTIDADE_CARGA = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraisquantidade")
;
    public static final By FORM_COMPLEMENTO = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraiscomplemento"
);
    public static final By FORM_PLACA_VEICULO = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraisplacaVeiculo
");
    public static final By FORM_NOME_MOTORISTA = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraisnomeMotorist
a");
    public static final By FORM_NOTA_FISCAL = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraisnotaFiscal")
;
    public static final By FORM_PRECO = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraispreco");
    public static final String NOME_TELA =
"telaProcessamentoDiretopnCargapn";

    // Elementos do formulário - Selects
    public static final By FORM_DT_PROCESSAMENTO = By

.id("startForm:telaProcessamentoDiretopnFormdataProcessamentoInputDate");
    public static final By FORM_RECEITA = By
          .id("startForm:telaProcessamentoDiretopnFormtipoProduto");
    public static final By FORM_DT_COLHEITA = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopndataColheitaInput
Date");
    public static final By FORM_PRODUTO = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopntipoProdutoIngred
```

```
iente");
    public static final By FORM_ORIGEM = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopnorigem");
    public static final By FORM_EMBALAGEM_INGREDIENTE = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopnembalagem");
    public static final By FORM_LOCAL_RECEBIMENTO = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopn3destino");
    public static final By FORM_DT_RECEBIMENTO = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopn3dataEntregaInput
Date");
    public static final By FORM_DESTINO = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraisdestino");
    public static final By FORM_EMBALAGEM_CARGA = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraisembalagem");
    public static final By FORM_DT_ENTREGA = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnDadosGeraisdataEntregaI
nputDate");

    // Elementos do formulário - Botões
    public static final By FORM_ADICIONAR_INGREDIENTES = By

.id("startForm:telaProcessamentoDiretopanelIngredientespnNovoIngredientegiN
ovoRegistro");
    public static final By FORM_INCLUIR_INGREDIENTES = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopn3pnBotoesbtInclui
rIngredientebt");
    public static final By FORM_ADICIONAR_CARGA = By

.id("startForm:telaProcessamentoDiretobasepanelcontentformpnNovoRegistrogiN
ovoRegistro");
    public static final By FORM_INCLUIR_CARGA = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnBotoesbtIncluirCargabt"
```

```
);
    public static final By FORM_EDITAR_INGREDIENTE = By

.id("startForm:telaProcessamentoDiretodataTableIngredientes:0:telaProcessam
entoDiretodataTableIngredientespnDeleteUpdategiUpdate");
    public static final By FORM_ATUALIZAR_INGREDIENTES = By

.id("formModalPanel:telaProcessamentoDiretopnRecebimentopn3pnBotoesbtAtuali
zarIngredientebt");
    public static final By FORM_EDITAR_CARGA = By

.id("startForm:telaProcessamentoDiretodataTableCargas:0:telaProcessamentoDi
retodataTableCargaspnEditDeletegiEdit");
    public static final By FORM_ATUALIZAR_CARGA = By

.id("formModalPanel:telaProcessamentoDiretopnCargapnBotoesbtAtualizarCargab
t");
      public static final By FORM_QUESTOES_QUESTIONARIO =
By.id("formModalPanel:telaProcessamentoDiretopnCargapnQuestionariopnQuestoe
spbi_topico1pg");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Receita] é obrigatório." + "\n"
          + "O campo [Ingredientes] é obrigatório." + "\n"
          + "O campo [Cargas] é obrigatório.";
}

package perfil.geral.rastreamento.processamento;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.rastreamento.recebimento.ElementsRecebimento;
import perfil.geral.rastreamento.recebimento.TestRecebimento;
```

```
import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestProcessamento {
	WebDriver driver;
	String dataAtual = Util.dataAtual();
	public String codigoProcessamento = null;
	public BigDecimal quantidadeProcessada = new BigDecimal(15);

	@BeforeClass(alwaysRun = true)
	public void setUpProcessamento() {
		Util.startTestMsg(TestProcessamento.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO);
	}

	@Test(groups = { "insert" })
	public void setUpInsert() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.NOVO_REGISTRO));

	driver.findElement(ElementsProcessamento.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods = "setUpInsert")
	public void insertUniqueProcessamento() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_RECEITA));
		Util.selectOption(ElementsProcessamento.FORM_RECEITA,
Padroes.NOME_RECEITA);
		Util.waitJsBlock();
```

```
driver.findElement(ElementsProcessamento.FORM_OBSERVACAO).sendKeys("Observa
ção Processamento");


driver.findElement(ElementsProcessamento.FORM_DT).sendKeys(dataAtual);
            Util.selectOption(ElementsProcessamento.FORM_TIPO_INGREDIENTE,
Padroes.NOME_TIPO_INGREDIENTE);
            Util.waitJsBlock();
            Util.selectOption(ElementsProcessamento.FORM_INGREDIENTE, 1);
            Util.waitJsBlock();


driver.findElement(ElementsProcessamento.FORM_QTDE).sendKeys(quantidadeProc
essada.toString());
            TestRecebimento.totalEstoqueSecondRecebimento.set(1,


TestRecebimento.totalEstoqueSecondRecebimento.get(1).add(quantidadeProcessa
da));
            TestRecebimento.totalEstoqueSecondRecebimento.set(2,
TestRecebimento.totalEstoqueSecondRecebimento.get(2)
                        .subtract(quantidadeProcessada));
            Util.scrollUp();

      driver.findElement(ElementsProcessamento.FORM_ADICIONAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_INGREDIENTE_ADICIONADO));
            Util.selectOption(ElementsProcessamento.FORM_LOCAL_SAIDA,
Padroes.NOME_MEU_CADASTRO);
            Util.waitJsBlock();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_NOVO_CARGA));

      driver.findElement(ElementsProcessamento.FORM_NOVO_CARGA).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_CARGA_DESTINO));
            Util.selectOption(ElementsProcessamento.FORM_CARGA_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_QUESTOES_QUESTIONARIO));


driver.findElement(ElementsProcessamento.FORM_CARGA_QTDE).sendKeys("250");
            Util.selectOption(ElementsProcessamento.FORM_CARGA_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsProcessamento.FORM_CARGA_DT_ENTREGA).sendKeys(da
taAtual);


driver.findElement(ElementsProcessamento.FORM_CARGA_PLACA_VEICULO).sendKeys
("ERJ 7850");


driver.findElement(ElementsProcessamento.FORM_CARGA_NOME_MOTORISTA).sendKey
s("Paz Moreno");


driver.findElement(ElementsProcessamento.FORM_CARGA_COMPLEMENTO).sendKeys("
super");


driver.findElement(ElementsProcessamento.FORM_CARGA_PRECO).sendKeys("2.39")
;
            String[] respostas_um = { "De Acordo", "Não se aplica", "Abaixo
de 51%", "0-20%", "Acima de 31%", "16-30%",
                        "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamento.NOME_TELA, respostas_um);

      driver.findElement(ElementsProcessamento.FORM_CARGA_INCLUIR).click();
```

```
Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));

      driver.findElement(ElementsProcessamento.FORM_NOVO_CARGA).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_CARGA_DESTINO));
            Util.selectOption(ElementsProcessamento.FORM_CARGA_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_QUESTOES_QUESTIONARIO));

driver.findElement(ElementsProcessamento.FORM_CARGA_QTDE).sendKeys("320");
            Util.selectOption(ElementsProcessamento.FORM_CARGA_EMBALAGEM,
Padroes.NOME_EMBALAGEM);

driver.findElement(ElementsProcessamento.FORM_CARGA_DT_ENTREGA).sendKeys(da
taAtual);

driver.findElement(ElementsProcessamento.FORM_CARGA_PLACA_VEICULO).sendKeys
("ANT 8095");

driver.findElement(ElementsProcessamento.FORM_CARGA_NOME_MOTORISTA).sendKey
s("Thaís Dávila");

driver.findElement(ElementsProcessamento.FORM_CARGA_COMPLEMENTO).sendKeys("
super");

driver.findElement(ElementsProcessamento.FORM_CARGA_PRECO).sendKeys("1.19")
;
```

```
            String[] respostas_dois = { "Fora do Padrão", "Não se aplica",
"Não se aplica", "0-20%", "0-10%", "0-15%", "4-7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamento.NOME_TELA, respostas_dois);

      driver.findElement(ElementsProcessamento.FORM_CARGA_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.waitJsBlock();

            driver.findElement(ElementsProcessamento.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);
            codigoProcessamento =
Util.capturaCodigo(driver.findElement(ElementsRastreador.MODAL_CONFIRMA).ge
tText());
            Util.testMsg("TestProcessamento", "Código: " +
codigoProcessamento);

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.NOVO_REGISTRO));

            // Impressão das mensagens para verificação de estoque
            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(0).setScale(2) + " | Env:
"
```

```
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(1).setScale(2) + " |
Disp: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(2).setScale(2) + " } /
EXPECTED");
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertUniqueProcessamento")
      public void verifyEstoqueRecebimentoAfterProcessamento() {
            List<String> valorEstoqueTabelaSecondRecebimento = new
ArrayList<String>();

            driver.findElement(ElementsRastreador.RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));

            // Executa asserts do segundo recebimento (recebimento,
enviado,
            // disponível).


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
```

```
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        TestRecebimento.codigosRecebimento.get(1)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(0),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(0).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(1),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(1).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(2),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(2).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        + valorEstoqueTabelaSecondRecebimento.get(0) + " |
Env: " + valorEstoqueTabelaSecondRecebimento.get(1)
                        + " | Disp: " +
valorEstoqueTabelaSecondRecebimento.get(2) + " } / RESULT");
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamento.TABLE,
ElementsProcessamento.TABLE_VALUE, Padroes.NOME_RECEITA
                        + Padroes.EDITADO, ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueProcessamento() {
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_OBSERVACAO));


driver.findElement(ElementsProcessamento.FORM_OBSERVACAO).sendKeys(Padroes.
EDITADO);


driver.findElement(ElementsProcessamento.FORM_EDITAR_PRIMEIRA_CARGA).click(
);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_CARGA_QTDE));

      driver.findElement(ElementsProcessamento.FORM_CARGA_QTDE).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_QTDE).sendKeys("320");


driver.findElement(ElementsProcessamento.FORM_CARGA_DT_ENTREGA).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_DT_ENTREGA).sendKeys(da
taAtual);


driver.findElement(ElementsProcessamento.FORM_CARGA_PLACA_VEICULO).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_PLACA_VEICULO).sendKeys
("JRE 0587");


driver.findElement(ElementsProcessamento.FORM_CARGA_NOME_MOTORISTA).clear()
;
```

```
driver.findElement(ElementsProcessamento.FORM_CARGA_NOME_MOTORISTA).sendKey
s("Ana Paula Leme");


driver.findElement(ElementsProcessamento.FORM_CARGA_COMPLEMENTO).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_COMPLEMENTO).sendKeys("
pp");

      driver.findElement(ElementsProcessamento.FORM_CARGA_PRECO).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_PRECO).sendKeys("1.59")
;
            String[] respostas_tres = { "Fora do Padrão", "100%", "Acima de
51%", "Não se aplica", "0-10%", "0-15%",
                        "Acima de 7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamento.NOME_TELA, respostas_tres);


driver.findElement(ElementsProcessamento.FORM_CARGA_ATUALIZAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));


driver.findElement(ElementsProcessamento.FORM_EDITAR_SEGUNDA_CARGA).click()
;


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_CARGA_QTDE));

      driver.findElement(ElementsProcessamento.FORM_CARGA_QTDE).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_QTDE).sendKeys("120");
```

```
driver.findElement(ElementsProcessamento.FORM_CARGA_DT_ENTREGA).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_DT_ENTREGA).sendKeys(da
taAtual);


driver.findElement(ElementsProcessamento.FORM_CARGA_PLACA_VEICULO).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_PLACA_VEICULO).sendKeys
("TNA 5908");


driver.findElement(ElementsProcessamento.FORM_CARGA_NOME_MOTORISTA).clear()
;


driver.findElement(ElementsProcessamento.FORM_CARGA_NOME_MOTORISTA).sendKey
s("Jehane Jahn");


driver.findElement(ElementsProcessamento.FORM_CARGA_COMPLEMENTO).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_COMPLEMENTO).sendKeys("
pp");

      driver.findElement(ElementsProcessamento.FORM_CARGA_PRECO).clear();


driver.findElement(ElementsProcessamento.FORM_CARGA_PRECO).sendKeys("2.09")
;
            String[] respostas_quatro = { "De Acordo", "100%", "Acima de
51%", "0-20%", "Acima de 31%", "Acima de 31%",
                        "Acima de 7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsProcessamento.NOME_TELA, respostas_quatro);
```

```
driver.findElement(ElementsProcessamento.FORM_CARGA_ATUALIZAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            driver.findElement(ElementsProcessamento.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueProcessamento() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.NOVO_REGISTRO));
            Util.acionaElemento(ElementsProcessamento.TABLE,
ElementsProcessamento.TABLE_VALUE, Padroes.NOME_RECEITA
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" })
      public void validacaoCamposObrigatoriosProcessamentoIngrediente() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.NOVO_REGISTRO));


      driver.findElement(ElementsProcessamento.NOVO_REGISTRO).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.FORM_ADICIONAR));


      driver.findElement(ElementsProcessamento.FORM_ADICIONAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsProce
ssamento.MODAL_ERRO_INGREDIENTES));



Assert.assertEquals(driver.findElement(ElementsProcessamento.MODAL_ERRO_ING
REDIENTES).getText(),

      ElementsProcessamento.MSG_VALIDA_CAMPOS_INGREDIENTES,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"validacaoCamposObrigatoriosProcessamentoIngrediente")
      public void validacaoCamposObrigatoriosProcessamentoGeral() {
            driver.findElement(ElementsProcessamento.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsProcessamento.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
```

```
	}

	@AfterClass(alwaysRun = true)
	public void tearDown() {
		Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.PROCESSAMENTO);
		Util.endTestMsg(TestProcessamento.class.getSimpleName());
	}
}

package perfil.geral.rastreamento.processamento;

import org.openqa.selenium.By;

public class ElementsProcessamento {

	// Elementos da tela principal - Botões
	public static final By NOVO_REGISTRO =
By.id("startForm:telaProcessamento3GbasepanelcontentlistgiNovoRegistro");
	public static final By FILTRO_NOVO = By
			.id("startForm:telaProcessamento3Gbasepanelcontentfiltertoppgrightlknewfilter");
	public static final String TABLE_IMPRIMIR =
":telaProcessamento3GdataTablepnEtiquetabtImprimirEtiqueta";

	// Elementos da tela principal - Tabela
	public static final String TABLE =
"startForm:telaProcessamento3GdataTable:";
	public static final By TABLE_ORDER = By
			.id("startForm:telaProcessamento3GdataTable:telaProcessamento3GdataTablecolumn3header:sortDiv");
	public static final String TABLE_VALUE =
":telaProcessamento3GdataTablecolumn2otValue2";

	// Elementos do filtro
	public static final By FILTRO_LIMPAR = By
			.id("startForm:telaProcessamento3Gbasepanelcontentfiltercontrolslkclearfilter");
	public static final By FILTRO_BUSCAR =
By.id("startForm:telaProcessamento3Gbasepanelcontentfiltercontrolslkfilter"
);
```

```
	// Elementos da tela de impressão
	public static final By ETIQUETA_GERAR =
By.id("startForm:telaProcessamento3GpnGeracaoEtiquetaclGerar");

	// Elementos do formulário - Geral
	public static final By FORM_RECEITA =
By.id("startForm:telaProcessamento3GpnFormtipoProduto");
	public static final By FORM_OBSERVACAO =
By.id("startForm:telaProcessamento3GpnFormobservacao");
	public static final By FORM_DT =
By.id("startForm:telaProcessamento3GpnFormdataProcessamentoInputDate");
	public static final By FORM_TIPO_INGREDIENTE =
By.id("startForm:telaProcessamento3GpnFormtipoIngrediente");
	public static final By FORM_INGREDIENTE =
By.id("startForm:telaProcessamento3Gingrediente");
	public static final By FORM_ADICIONAR =
By.id("startForm:telaProcessamento3GbtAdicionarIngredientebt");
	public static final By FORM_QTDE =
By.id("startForm:telaProcessamento3GpnFormqtdCodigoProcessamento");
	public static final By FORM_LOCAL_SAIDA =
By.id("startForm:telaProcessamento3GpnFormdestino");
	public static final By FORM_NOVO_CARGA = By
			.id("startForm:telaProcessamento3GbasepanelcontentformpnN
ovoRegistrogiNovoRegistro");
	public static final By FORM_EDITAR_PRIMEIRA_CARGA = By
			.id("startForm:telaProcessamento3GdataTableCargas:0:telaP
rocessamento3GdataTableCargaspnEditDeletegiEdit");
	public static final By FORM_EDITAR_SEGUNDA_CARGA = By
			.id("startForm:telaProcessamento3GdataTableCargas:1:telaP
rocessamento3GdataTableCargaspnEditDeletegiEdit");
	public static final By FORM_SALVAR =
By.id("startForm:telaProcessamento3GbasepanelcontentformbtsalvarbtEdit");
	public static final By FORM_INGREDIENTE_ADICIONADO = By
			.id("startForm:telaProcessamento3GdataTableIngredientes:0
:telaProcessamento3GdataTableIngredientescolumn0otValue0");

	// Elementos do formulário - Carga
	public static final By FORM_CARGA_DESTINO =
By.id("formModalPanel:telaProcessamento3GpnCargapnDadosGeraisdestino");
	public static final By FORM_CARGA_QTDE =
```

```
By.id("formModalPanel:telaProcessamento3GpnCargapnDadosGeraisquantidade");
      public static final By FORM_CARGA_EMBALAGEM = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnDadosGera
isembalagem");
      public static final By FORM_CARGA_DT_ENTREGA = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnDadosGera
isdataEntregaInputDate");
      public static final By FORM_CARGA_PLACA_VEICULO = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnDadosGera
isplacaVeiculo");
      public static final By FORM_CARGA_NOME_MOTORISTA = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnDadosGera
isnomeMotorista");
      public static final By FORM_CARGA_COMPLEMENTO = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnDadosGera
iscomplemento");
      public static final By FORM_CARGA_PRECO =
By.id("formModalPanel:telaProcessamento3GpnCargapnDadosGeraispreco");
      public static final By FORM_CARGA_INCLUIR = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnBotoesbtI
ncluirCargabt");
      public static final By FORM_CARGA_ATUALIZAR = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnBotoesbtA
tualizarCargabt");
      public static final By MODAL_ERRO_INGREDIENTES =
By.id("startForm:telaProcessamento3GpnFormotMsgErro");
      public static final String NOME_TELA =
"telaProcessamento3GpnCargapn";
      public static final By FORM_QUESTOES_QUESTIONARIO = By
                  .id("formModalPanel:telaProcessamento3GpnCargapnQuestiona
riopnQuestoespbi_topico1pg");

      // Mensagem de validação dos campos obrigatorios
      public static String MSG_VALIDA_CAMPOS_INGREDIENTES = "[ERRO] Os
campos Tipo Ingrediente, Ingrediente, Quantidade são obrigatórios";
      public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
[Receita] é obrigatório." + "\n"
                  + "O campo [Ingredientes] é obrigatório." + "\n" + "O
campo [Cargas] é obrigatório.";

}
```

```
package perfil.geral.rastreamento.envioSimples;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.rastreamento.recebimento.ElementsRecebimento;
import perfil.geral.rastreamento.recebimento.TestRecebimento;
import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestEnvioSimples {
	WebDriver driver;
	String dataAtual = Util.dataAtual();
	public static List<String> codigosEnvio = new ArrayList<String>();
	public static List<BigDecimal> enviosSimples = Arrays.asList(new
BigDecimal(45), new BigDecimal(10));

	@BeforeClass(alwaysRun = true)
	public void setUpEnvioSimples() {
		Util.startTestMsg(TestEnvioSimples.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_SIMPLES);
	}

	@Test(groups = { "insert", "verify", "campos" })
	public void abreTelaInsertEnvioSimples() {
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.NOVO_REGISTRO));
            driver.findElement(ElementsEnvioSimples.NOVO_REGISTRO).click();
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertEnvioSimples")
      public void insertFirstEnvio() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.FORM_DESTINO));
            Util.selectOption(ElementsEnvioSimples.FORM_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);
            Util.waitJsBlock();
            Util.selectOption(ElementsEnvioSimples.FORM_PRODUTO,
Padroes.NOME_PRODUTO);


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.BLOCKUI));
            Util.selectOption(ElementsEnvioSimples.FORM_CODIGO_RECEBIMENTO,
1);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.FORM_QUESTOES_QUESTIONARIO));
            // Executa as operações de alteração de valores de estoque de
            // recebimento.


driver.findElement(ElementsEnvioSimples.FORM_QTDE).sendKeys(enviosSimples.g
et(0).toString());
            TestRecebimento.totalEstoqueSecondRecebimento.set(1,


TestRecebimento.totalEstoqueSecondRecebimento.get(1).add(enviosSimples.get(
0))); // Índice


                                    // enviado
```

```
            TestRecebimento.totalEstoqueSecondRecebimento.set(2,
TestRecebimento.totalEstoqueSecondRecebimento.get(2)
                        .subtract(enviosSimples.get(0))); // Índice
disponível

            Util.selectOption(ElementsEnvioSimples.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsEnvioSimples.FORM_COMPLEMENTO).sendKeys("pp");


driver.findElement(ElementsEnvioSimples.FORM_DT_ENTREGA).sendKeys(dataAtual
);


driver.findElement(ElementsEnvioSimples.FORM_NOTA_FISCAL).sendKeys("147258"
);


driver.findElement(ElementsEnvioSimples.FORM_PLACA_VEICULO).sendKeys("LIM
4789");


driver.findElement(ElementsEnvioSimples.FORM_NOME_MOTORISTA).sendKeys("Vivi
ane de Paula");

      driver.findElement(ElementsEnvioSimples.FORM_PRECO).sendKeys("3.18");


driver.findElement(ElementsEnvioSimples.FORM_OBSERVACAO).sendKeys("Observac
ao envio simples");
            String[] respostas_um = { "De Acordo", "100%", "Acima de 51%",
"0-20%", "0-10%", "0-15%", "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.STARTFORM,
ElementsEnvioSimples.NOME_TELA, respostas_um);
            driver.findElement(ElementsEnvioSimples.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.MODAL_CONFIRMA_REPLICANDO));
```

```
            Assert.assertEquals(

driver.findElement(ElementsEnvioSimples.MODAL_CONFIRMA_REPLICANDO).getText(
)
                                        .equals(MsgSistema.CADASTRO_NULL_REPLICA
NDO), false, MsgErro.FAIL_CADASTRO);

codigosEnvio.add(Util.capturaCodigo(driver.findElement(ElementsEnvioSimples
.MODAL_CONFIRMA_REPLICANDO).getText()));
            Util.testMsg("TestEnvioSimples", "Código: " +
codigosEnvio.get(0));

driver.findElement(ElementsEnvioSimples.FORM_SUCESSO_REPLICAR_SIM).click();

Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
      }

      @Test(groups = { "insert" }, dependsOnMethods = "insertFirstEnvio")
      public void insertSecondEnvio() {
            // Executa as operações de alteração de valores de estoque de
            // recebimento.

driver.findElement(ElementsEnvioSimples.FORM_QTDE).sendKeys(enviosSimples.g
et(1).toString());
            TestRecebimento.totalEstoqueSecondRecebimento.set(1,

TestRecebimento.totalEstoqueSecondRecebimento.get(1).add(enviosSimples.get(
1))); // Índice

                                        // enviado
            TestRecebimento.totalEstoqueSecondRecebimento.set(2,
TestRecebimento.totalEstoqueSecondRecebimento.get(2)
                        .subtract(enviosSimples.get(1))); // Índice
```

```
disponível
            Util.selectOption(ElementsEnvioSimples.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);
            Util.selectOption(ElementsEnvioSimples.FORM_CODIGO_RECEBIMENTO,
1);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.FORM_QUESTOES_QUESTIONARIO));


driver.findElement(ElementsEnvioSimples.FORM_OBSERVACAO).sendKeys("Observac
ao envio simples");
            String[] respostas_dois = { "De Acordo", "100%", "Acima de
51%", "0-20%", "0-10%", "0-15%", "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.STARTFORM,
ElementsEnvioSimples.NOME_TELA, respostas_dois);
            driver.findElement(ElementsEnvioSimples.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.MODAL_CONFIRMA_REPLICANDO));
            Assert.assertEquals(


driver.findElement(ElementsEnvioSimples.MODAL_CONFIRMA_REPLICANDO).getText(
)
                                    .equals(MsgSistema.CADASTRO_NULL_REPLICA
NDO), false, MsgErro.FAIL_CADASTRO);


codigosEnvio.add(Util.capturaCodigo(driver.findElement(ElementsEnvioSimples
.MODAL_CONFIRMA_REPLICANDO).getText()));
            Util.testMsg("TestEnvioSimples", "Código: " +
codigosEnvio.get(1));


driver.findElement(ElementsEnvioSimples.FORM_SUCESSO_REPLICAR_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
```

```
Simples.NOVO_REGISTRO));

            // Impressão das mensagens para verificação de estoque
            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(0).setScale(2) + " | Env:
"
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(1).setScale(2) + " |
Disp: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(2).setScale(2) + " } /
EXPECTED");
      }

      @Test(groups = { "insert" }, dependsOnMethods = "insertSecondEnvio")
      public void verifyEstoqueRecebimentoAfterEnvioSimples() {
            List<String> valorEstoqueTabelaSecondRecebimento = new
ArrayList<String>();

            driver.findElement(ElementsRastreador.RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));

            // Executa asserts do segundo recebimento (recebimento,
enviado,
            // disponível).


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
```

```
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        TestRecebimento.codigosRecebimento.get(1)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(0),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(0).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(1),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(1).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(2),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(2).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        + valorEstoqueTabelaSecondRecebimento.get(0) + " |
Env: " + valorEstoqueTabelaSecondRecebimento.get(1)
                        + " | Disp: " +
valorEstoqueTabelaSecondRecebimento.get(2) + " } / RESULT");
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.NOVO_REGISTRO));
```

```
            Util.acionaElemento(ElementsEnvioSimples.TABLE,
ElementsEnvioSimples.TABLE_VALUE, Padroes.NOME_EMBALAGEM
                        + Padroes.EDITADO, ElementsRastreador.TABLE_EDITAR);

      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editFirstEnvioSimples() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.FORM_OBSERVACAO));


driver.findElement(ElementsEnvioSimples.FORM_OBSERVACAO).sendKeys(Padroes.E
DITADO);
            driver.findElement(ElementsEnvioSimples.FORM_QTDE).clear();

      driver.findElement(ElementsEnvioSimples.FORM_QTDE).sendKeys("60");

      driver.findElement(ElementsEnvioSimples.FORM_COMPLEMENTO).clear();


driver.findElement(ElementsEnvioSimples.FORM_COMPLEMENTO).sendKeys("super")
;

      driver.findElement(ElementsEnvioSimples.FORM_DT_ENTREGA).clear();


driver.findElement(ElementsEnvioSimples.FORM_DT_ENTREGA).sendKeys(dataAtual
);

      driver.findElement(ElementsEnvioSimples.FORM_PLACA_VEICULO).clear();


driver.findElement(ElementsEnvioSimples.FORM_PLACA_VEICULO).sendKeys("LIM
3344");

      driver.findElement(ElementsEnvioSimples.FORM_NOME_MOTORISTA).clear();
```

```
driver.findElement(ElementsEnvioSimples.FORM_NOME_MOTORISTA).sendKeys("Mich
elle Poligamia");

      driver.findElement(ElementsEnvioSimples.FORM_NOTA_FISCAL).clear();


driver.findElement(ElementsEnvioSimples.FORM_NOTA_FISCAL).sendKeys("2210251
8");
            driver.findElement(ElementsEnvioSimples.FORM_PRECO).clear();

      driver.findElement(ElementsEnvioSimples.FORM_PRECO).sendKeys("6.89");
            String[] repostas_quatro = { "Fora do Padrão", "Não se aplica",
"Não se aplica", "Não se aplica", "Acima de 31%",
                        "Acima de 31%", "Acima de 7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.STARTFORM,
ElementsEnvioSimples.NOME_TELA, repostas_quatro);
            driver.findElement(ElementsEnvioSimples.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteFirstSecondEnvioSimples() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.NOVO_REGISTRO));
            Util.acionaElemento(ElementsEnvioSimples.TABLE,
ElementsEnvioSimples.TABLE_VALUE, Padroes.NOME_EMBALAGEM
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
            Util.acionaElemento(ElementsEnvioSimples.TABLE,
```

```
ElementsEnvioSimples.TABLE_VALUE, Padroes.NOME_EMBALAGEM
                        + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertEnvioSimples")
      public void validacaoCamposObrigatoriosEnvioSimples() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Simples.FORM_SALVAR));

            driver.findElement(ElementsEnvioSimples.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsEnvioSimples.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_SIMPLES);
            Util.endTestMsg(TestEnvioSimples.class.getSimpleName());
      }
}

package perfil.geral.rastreamento.envioSimples;

import org.openqa.selenium.By;
```

```
public class ElementsEnvioSimples {

    // Elementos da Tela Principal - Botões
    public static final By NOVO_REGISTRO = By

.id("startForm:telaEnvioUmaEmbalagem3GbasepanelcontentlistgiNovoRegistro");
    public static final By FILTRO_NOVO = By

.id("startForm:telaEnvioUmaEmbalagem3Gbasepanelcontentfiltertoppgrightlknew
filter");
    public static final String TABLE_IMPRIMIR =
":telaEnvioUmaEmbalagem3GdataTablepnEtiquetabtImprimirEtiqueta";

    // Elementos da Tela Principal - Tabela
    public static final String TABLE =
"startForm:telaEnvioUmaEmbalagem3GdataTable:";
    public static final String TABLE_VALUE =
":telaEnvioUmaEmbalagem3GdataTablecolumn4otValue4";
    public static final By TABLE_ORDER = By

.id("startForm:telaEnvioUmaEmbalagem3GdataTable:telaEnvioUmaEmbalagem3Gdata
Tablecolumn4header:sortDiv");

    // Elementos do Filtro
    public static final By FILTRO_LIMPAR = By

.id("startForm:telaEnvioUmaEmbalagem3Gbasepanelcontentfiltercontrolslkclear
filter");
    public static final By FILTRO_BUSCAR = By

.id("startForm:telaEnvioUmaEmbalagem3Gbasepanelcontentfiltercontrolslkfilte
r");

    // Elementos da tela de impressão
    public static final By ETIQUETA_GERAR = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnGeracaoEtiquetaclGerar");

    // Elementos de Formulário
    public static final By FORM_DESTINO = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormdestino");
```

```
    public static final By FORM_PRODUTO = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormtipoProduto");
    public static final By FORM_CODIGO_RECEBIMENTO = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormcodigosDisponiveis");
    public static final By FORM_QTDE = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormquantidade");
    public static final By FORM_EMBALAGEM = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormembalagem");
    public static final By FORM_COMPLEMENTO = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormcomplemento");
    public static final By FORM_DT_ENTREGA = By

.id("startForm:telaEnvioUmaEmbalagem3GpnFormdataEntregaInputDate");
    public static final By FORM_NOTA_FISCAL = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormnotaFiscal");
    public static final By FORM_PLACA_VEICULO = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormplacaVeiculo");
    public static final By FORM_NOME_MOTORISTA = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormnomeMotorista");
    public static final By FORM_PRECO = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormpreco");
    public static final By FORM_OBSERVACAO = By
          .id("startForm:telaEnvioUmaEmbalagem3GpnFormobservacao");
    public static final By FORM_INSPECAO = By

.id("startForm:telaEnvioUmaEmbalagem3GbasepanelcontentformpnQuestionariotip
oQuestionario");
    public static final By FORM_SALVAR = By

.id("startForm:telaEnvioUmaEmbalagem3Gbasepanelcontentformbtsalvarbt");
    public static final String NOME_TELA =
"telaEnvioUmaEmbalagem3GbasepanelcontentformpnQ";

    // Elementos do modal
    public static final By MODAL_CONFIRMA_REPLICANDO = By

.id("formModalPanel:telaEnvioUmaEmbalagem3GpnMsgSucessoCadastroReplicandoDa
dosotMsg");
    public static final By FORM_SUCESSO_REPLICAR_SIM = By

.id("formModalPanel:telaEnvioUmaEmbalagem3GpnMsgSucessoCadastroReplicandoDa
```

```
dosbtYesReplicandobt");
    public static final By FORM_SUCESSO_REPLICAR_NAO = By

.id("formModalPanel:telaEnvioUmaEmbalagem3GpnMsgSucessoCadastroReplicandoDa
dosbtNobt");

    public static final By FORM_QUESTOES_QUESTIONARIO =
By.id("startForm:telaEnvioUmaEmbalagem3GbasepanelcontentformpnQuestionariop
nQuestoespbi_topico1pg");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Destino] é obrigatório." + "\n"
          + "O campo [Produto] é obrigatório." + "\n"
          + "O campo [Código de recebimento] é obrigatório." + "\n"
          + "O campo [Quantidade] é obrigatório." + "\n"
          + "O campo [Embalagem] é obrigatório.";
}

package perfil.geral.rastreamento.envioMisturaOrigens;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.rastreamento.recebimento.ElementsRecebimento;
import perfil.geral.rastreamento.recebimento.TestRecebimento;
import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestEnvioMisturaOrigens {
```

```
	WebDriver driver;
	String dataAtual = Util.dataAtual();
	public static String codigoEnvio = null;
	public static List<BigDecimal> enviosMisturaOrigens =
Arrays.asList(new BigDecimal(60), new BigDecimal(30));

	@BeforeClass(alwaysRun = true)
	public void setUpEnvioMisturaOrigens() {

	Util.startTestMsg(TestEnvioMisturaOrigens.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_MISTURA_ORIGENS);
	}

	@Test(groups = { "insert", "verify", "campos" })
	public void abreTelaInsertEnvioMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.NOVO_REGISTRO));

	driver.findElement(ElementsMisturaOrigens.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertEnvioMisturaOrigens")
	public void insertUniqueEnvioMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_PRODUTO));
		Util.selectOption(ElementsMisturaOrigens.FORM_PRODUTO,
Padroes.NOME_PRODUTO);
		Util.waitJsBlock();


driver.findElement(ElementsMisturaOrigens.FORM_NOVO_RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
```

```
raOrigens.FORM_RECEBIMENTO_CODIGO));

      Util.selectOption(ElementsMisturaOrigens.FORM_RECEBIMENTO_CODIGO, 1);
            Util.waitJsBlock();


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_QTDE).sendKeys(
                        enviosMisturaOrigens.get(0).setScale(2).toString());
            TestRecebimento.totalEstoqueSecondRecebimento.set(1,


TestRecebimento.totalEstoqueSecondRecebimento.get(1).add(enviosMisturaOrige
ns.get(0)));
            TestRecebimento.totalEstoqueSecondRecebimento.set(2,
TestRecebimento.totalEstoqueSecondRecebimento.get(2)
                        .subtract(enviosMisturaOrigens.get(0)));



driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_INCLUIR).click()
;


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));



driver.findElement(ElementsMisturaOrigens.FORM_NOVO_RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_RECEBIMENTO_CODIGO));

      Util.selectOption(ElementsMisturaOrigens.FORM_RECEBIMENTO_CODIGO, 2);
            Util.waitJsBlock();


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_QTDE).sendKeys(
                        enviosMisturaOrigens.get(1).setScale(2).toString());
            TestRecebimento.totalEstoqueFirstRecebimento.set(1,
```

```
TestRecebimento.totalEstoqueFirstRecebimento.get(1).add(enviosMisturaOrigen
s.get(1)));
            TestRecebimento.totalEstoqueFirstRecebimento.set(2,

TestRecebimento.totalEstoqueFirstRecebimento.get(2).subtract(enviosMisturaO
rigens.get(1)));



driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_INCLUIR).click()
;


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));

            Util.selectOption(ElementsMisturaOrigens.FORM_LOCAL_SAIDA,
Padroes.NOME_MEU_CADASTRO);
            Util.waitJsBlock();

      driver.findElement(ElementsMisturaOrigens.FORM_NOVA_CARGA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_CARGA_DESTINO));
            Util.selectOption(ElementsMisturaOrigens.FORM_CARGA_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_QUESTIONARIO_QUESTOES));


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_QTDE).sendKeys("25");
            Util.selectOption(ElementsMisturaOrigens.FORM_CARGA_EMBALAGEM,
Padroes.NOME_EMBALAGEM);
```

```
driver.findElement(ElementsMisturaOrigens.FORM_CARGA_DT_ENTREGA).sendKeys(d
ataAtual);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PLACA_VEICULO).sendKey
s("ABC 4567");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_NOME_MOTORISTA).sendKe
ys("Cristine Saur");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_COMPLEMENTO).sendKeys(
"pp");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PRECO).sendKeys("1.99"
);
            String[] respostas_um = { "Fora do Padrão", "Não se aplica",
"Abaixo de 51%", "21-50%", "11-30%", "16-30%",
                        "4-7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsMisturaOrigens.NOME_TELA, respostas_um);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));


      driver.findElement(ElementsMisturaOrigens.FORM_NOVA_CARGA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_CARGA_DESTINO));
            Util.selectOption(ElementsMisturaOrigens.FORM_CARGA_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_QUESTIONARIO_QUESTOES));


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_QTDE).sendKeys("65");
            Util.selectOption(ElementsMisturaOrigens.FORM_CARGA_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_DT_ENTREGA).sendKeys(d
ataAtual);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PLACA_VEICULO).sendKey
s("EFG 8080");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_NOME_MOTORISTA).sendKe
ys("Ariela Medeiros");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_COMPLEMENTO).sendKeys(
"super");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PRECO).sendKeys("4.52"
);
            String[] respostas_dois = { "Fora do Padrão", "Não se aplica",
"Abaixo de 51%", "21-50%", "11-30%", "16-30%",
                        "4-7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsMisturaOrigens.NOME_TELA, respostas_dois);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
```

```
driver.findElement(ElementsMisturaOrigens.FORM_OBSERVACAO).sendKeys("Observ
acao envio mistura de origens");
            driver.findElement(ElementsMisturaOrigens.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);
            codigoEnvio =
Util.capturaCodigo(driver.findElement(ElementsRastreador.MODAL_CONFIRMA).ge
tText());
            Util.testMsg("TestEnvioMisturaOrigens", "Código: " +
codigoEnvio);

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.NOVO_REGISTRO));
            TestRecebimento.totalEstoqueFirstRecebimento.get(0);

            // Impressão das mensagens para verificação de estoque
            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(0), "{ Rec: "
                        +
TestRecebimento.totalEstoqueFirstRecebimento.get(0).setScale(2) + " | Env:
"
                        +
TestRecebimento.totalEstoqueFirstRecebimento.get(1).setScale(2) + " | Disp:
"
                        +
TestRecebimento.totalEstoqueFirstRecebimento.get(2).setScale(2) + " } /
EXPECTED");
            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        +
```

```
TestRecebimento.totalEstoqueSecondRecebimento.get(0).setScale(2) + " | Env:
"
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(1).setScale(2) + " |
Disp: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(2).setScale(2) + " } /
EXPECTED");
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertUniqueEnvioMisturaOrigens")
      public void verifyEstoqueRecebimentoAfterEnvioMisturaOrigens() {
            List<String> valorEstoqueTabelaFirstRecebimento = new
ArrayList<String>();
            List<String> valorEstoqueTabelaSecondRecebimento = new
ArrayList<String>();

            driver.findElement(ElementsRastreador.RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
imento.NOVO_REGISTRO));



valorEstoqueTabelaFirstRecebimento.add(Util.retornaValorTabela(ElementsRece
bimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        TestRecebimento.codigosRecebimento.get(0)));


valorEstoqueTabelaFirstRecebimento.add(Util.retornaValorTabela(ElementsRece
bimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        TestRecebimento.codigosRecebimento.get(0)));


valorEstoqueTabelaFirstRecebimento.add(Util.retornaValorTabela(ElementsRece
```

```
bimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        TestRecebimento.codigosRecebimento.get(0)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaFirstRecebimento.get(0),
TestRecebimento.totalEstoqueFirstRecebimento
                        .get(0).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaFirstRecebimento.get(1),
TestRecebimento.totalEstoqueFirstRecebimento
                        .get(1).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaFirstRecebimento.get(2),
TestRecebimento.totalEstoqueFirstRecebimento
                        .get(2).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(0), "{ Rec: "
                        + valorEstoqueTabelaFirstRecebimento.get(0) + " |
Env: " + valorEstoqueTabelaFirstRecebimento.get(1)
                        + " | Disp: " +
valorEstoqueTabelaFirstRecebimento.get(2) + " } / RESULT");

            // Executa asserts do segundo recebimento (recebimento,
enviado,
            // disponível).


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
```

```
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        TestRecebimento.codigosRecebimento.get(1)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(0),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(0).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(1),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(1).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(2),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(2).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        + valorEstoqueTabelaSecondRecebimento.get(0) + " |
Env: " + valorEstoqueTabelaSecondRecebimento.get(1)
                        + " | Disp: " +
valorEstoqueTabelaSecondRecebimento.get(2) + " } / RESULT");
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
```

```
raOrigens.NOVO_REGISTRO));
            Util.acionaElemento(ElementsMisturaOrigens.TABLE,
ElementsMisturaOrigens.TABLE_VALUE,
                        Padroes.NOME_DESTINO_UTILIZADO,
ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueEnvioMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_OBSERVACAO));


driver.findElement(ElementsMisturaOrigens.FORM_OBSERVACAO).sendKeys(Padroes
.EDITADO);


driver.findElement(ElementsMisturaOrigens.FORM_EDITAR_PRIMEIRO_RECEBIMENTO)
.click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_RECEBIMENTO_QTDE));


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_QTDE).clear();


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_QTDE).sendKeys("
40");


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_ATUALIZAR).click
();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.waitJsBlock();
```

```
driver.findElement(ElementsMisturaOrigens.FORM_EDITAR_SEGUNDO_RECEBIMENTO).
click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_RECEBIMENTO_QTDE));


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_QTDE).clear();


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_QTDE).sendKeys("
10");


driver.findElement(ElementsMisturaOrigens.FORM_RECEBIMENTO_ATUALIZAR).click
();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.waitJsBlock();


driver.findElement(ElementsMisturaOrigens.FORM_EDITAR_PRIMEIRA_CARGA).click
();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_CARGA_QTDE));

      driver.findElement(ElementsMisturaOrigens.FORM_CARGA_QTDE).clear();


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_QTDE).sendKeys("20");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_DT_ENTREGA).sendKeys(d
ataAtual);
```

```
driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PLACA_VEICULO).clear()
;


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PLACA_VEICULO).sendKey
s("ABC 1234");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_NOME_MOTORISTA).clear(
);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_NOME_MOTORISTA).sendKe
ys("Giselle Vinotti");


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_COMPLEMENTO).clear();


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_COMPLEMENTO).sendKeys(
"super");

      driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PRECO).clear();


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PRECO).sendKeys("3.05"
);
            String[] respostas_tres = { "De Acordo", "100%", "Acima de
51%", "0-20%", "0-10%", "0-15%", "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsMisturaOrigens.NOME_TELA, respostas_tres);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_ATUALIZAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.waitJsBlock();
```

```
driver.findElement(ElementsMisturaOrigens.FORM_EDITAR_SEGUNDA_CARGA).click(
);

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_CARGA_QTDE));

      driver.findElement(ElementsMisturaOrigens.FORM_CARGA_QTDE).clear();

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_QTDE).sendKeys("30");

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PLACA_VEICULO).clear()
;

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PLACA_VEICULO).sendKey
s("EFG 7085");

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_DT_ENTREGA).clear();

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_DT_ENTREGA).sendKeys(d
ataAtual);

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_NOME_MOTORISTA).clear(
);

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_NOME_MOTORISTA).sendKe
ys("Anna Lara Cim");

driver.findElement(ElementsMisturaOrigens.FORM_CARGA_COMPLEMENTO).clear();
```

```
driver.findElement(ElementsMisturaOrigens.FORM_CARGA_COMPLEMENTO).sendKeys(
"super");

      driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PRECO).clear();


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_PRECO).sendKeys("5.05"
);
            String[] respostas_quatro = { "De Acordo", "100%", "Acima de
51%", "0-20%", "Acima de 31%", "Acima de 31%",
                        "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsMisturaOrigens.NOME_TELA, respostas_quatro);


driver.findElement(ElementsMisturaOrigens.FORM_CARGA_ATUALIZAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.waitJsBlock();
            driver.findElement(ElementsMisturaOrigens.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsMis
turaOrigens.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueEnvioMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.NOVO_REGISTRO));
            Util.acionaElemento(ElementsMisturaOrigens.TABLE,
```

```
ElementsMisturaOrigens.TABLE_VALUE,
                        Padroes.NOME_DESTINO_UTILIZADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertEnvioMisturaOrigens")
      public void validacaoCamposObrigatoriosMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsMistu
raOrigens.FORM_SALVAR));
            driver.findElement(ElementsMisturaOrigens.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));


Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsMisturaOrigens.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_MISTURA_ORIGENS);
            Util.endTestMsg(TestEnvioMisturaOrigens.class.getSimpleName());
      }
}

package perfil.geral.rastreamento.envioMisturaOrigens;

import org.openqa.selenium.By;

public class ElementsMisturaOrigens {
```

```
	// Elementos da tela principal - Botões
	public static final By NOVO_REGISTRO = By
			.id("startForm:telaEnvioMisturaOrigens3Gbasepanelcontentl
istgiNovoRegistro");
	public static final By FILTRO_NOVO = By
			.id("startForm:telaEnvioMisturaOrigens3Gbasepanelcontentf
iltertoppgrightlknewfilter");
	public static final String TABLE_IMPRIMIR =
":telaEnvioMisturaOrigens3GdataTablepnEtiquetabtImprimirEtiqueta";

	// Elementos da tela principal - Tabela
	public static final String TABLE =
"startForm:telaEnvioMisturaOrigens3GdataTable:";
	public static final By TABLE_ORDER = By
			.id("startForm:telaEnvioMisturaOrigens3GdataTable:telaEnv
ioMisturaOrigens3GdataTablecolumn3header:sortDiv");
	public static final String TABLE_VALUE =
":telaEnvioMisturaOrigens3GdataTablecolumn3otValue3";

	// Elementos do filtro
	public static final By FILTRO_LIMPAR = By
			.id("startForm:telaEnvioMisturaOrigens3Gbasepanelcontentf
iltercontrolslkclearfilter");
	public static final By FILTRO_BUSCAR = By
			.id("startForm:telaEnvioMisturaOrigens3Gbasepanelcontentf
iltercontrolslkfilter");

	// Elementos da tela de impressão
	public static final By ETIQUETA_GERAR =
By.id("startForm:telaEnvioMisturaOrigens3GpnGeracaoEtiquetaclGerar");

	// Elementos do formulário - Geral
	public static final By FORM_PRODUTO =
By.id("startForm:telaEnvioMisturaOrigens3GpnFormtipoProduto");
	public static final By FORM_OBSERVACAO =
By.id("startForm:telaEnvioMisturaOrigens3GpnFormobservacao");
	public static final By FORM_NOVO_RECEBIMENTO = By
			.id("startForm:telaEnvioMisturaOrigens3Gbasepanelcontentf
ormpnNovaOrigemgiNovoRegistro");
	public static final By FORM_LOCAL_SAIDA =
```

```
By.id("startForm:telaEnvioMisturaOrigens3GpnCargaslocalSaida");
      public static final By FORM_NOVA_CARGA = By
                  .id("startForm:telaEnvioMisturaOrigens3Gbasepanelcontentf
ormpnNovaCargagiNovoRegistro");
      public static final By FORM_EDITAR_PRIMEIRO_RECEBIMENTO = By
                  .id("startForm:telaEnvioMisturaOrigens3GdataTableOrigens:
0:telaEnvioMisturaOrigens3GdataTableOrigenspnEditDeletegiEdit");
      public static final By FORM_EDITAR_SEGUNDO_RECEBIMENTO = By
                  .id("startForm:telaEnvioMisturaOrigens3GdataTableOrigens:
1:telaEnvioMisturaOrigens3GdataTableOrigenspnEditDeletegiEdit");
      public static final By FORM_EDITAR_PRIMEIRA_CARGA = By
                  .id("startForm:telaEnvioMisturaOrigens3GdataTableCargas:0
:telaEnvioMisturaOrigens3GdataTableCargaspnEditDeletegiEdit");
      public static final By FORM_EDITAR_SEGUNDA_CARGA = By
                  .id("startForm:telaEnvioMisturaOrigens3GdataTableCargas:1
:telaEnvioMisturaOrigens3GdataTableCargaspnEditDeletegiEdit");
      public static final By FORM_SALVAR =
By.id("startForm:telaEnvioMisturaOrigens3Gbasepanelcontentformbtsalvarbt");

      // Elementos do formulário - Recebimento
      public static final By FORM_RECEBIMENTO_CODIGO = By
                  .id("formModalPanel:telaEnvioMisturaOrigens3GpnOrigempnDa
dosGeraiscodigosDisponiveis");
      public static final By FORM_RECEBIMENTO_QTDE = By
                  .id("formModalPanel:telaEnvioMisturaOrigens3GpnOrigempnDa
dosGeraisqtdOrigem");
      public static final By FORM_RECEBIMENTO_INCLUIR = By
                  .id("formModalPanel:telaEnvioMisturaOrigens3GpnOrigempnBo
toesbtIncluirOrigembt");
      public static final By FORM_RECEBIMENTO_ATUALIZAR = By
                  .id("formModalPanel:telaEnvioMisturaOrigens3GpnOrigempnBo
toesbtAtualizarOrigembt");

      // Elementos do formulário - Carga
      public static final By FORM_CARGA_DESTINO = By
                  .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDad
osGeraisdestino");
      public static final By FORM_CARGA_QTDE = By
                  .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDad
osGeraisquantidade");
      public static final By FORM_CARGA_EMBALAGEM = By
```

```
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDad
osGeraisembalagem");
      public static final By FORM_CARGA_DT_ENTREGA = By
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDad
osGeraisdataEntregaInputDate");
      public static final By FORM_CARGA_PLACA_VEICULO = By
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDad
osGeraisplacaVeiculo");
      public static final By FORM_CARGA_NOME_MOTORISTA = By
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDad
osGeraisnomeMotorista");
      public static final By FORM_CARGA_COMPLEMENTO = By
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDad
osGeraiscomplemento");
      public static final By FORM_CARGA_PRECO =
By.id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnDadosGeraispreco");
      public static final By FORM_CARGA_INCLUIR = By
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnBot
oesbtIncluirCargabt");
      public static final By FORM_CARGA_ATUALIZAR = By
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnBot
oesbtAtualizarCargabt");
      public static final String NOME_TELA =
"telaEnvioMisturaOrigens3GpnCargapn";
      public static final By FORM_QUESTIONARIO_QUESTOES = By
                .id("formModalPanel:telaEnvioMisturaOrigens3GpnCargapnQue
stionariopnQuestoespbi_topico1pg");

      // Mensagem de validação dos campos obrigatorios
      public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
[Produto] é obrigatório." + "\n"
                + "O campo [Recebimentos] é obrigatório" + "\n" + "O
campo [Cargas] é obrigatório";

}


package perfil.geral.rastreamento.envioDiretoMisturaOrigens;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
```

```
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestEnvioDiretoMisturaOrigens {
      WebDriver driver;
      String dataAtual = Util.dataAtual();
      public static String codigoEnvio = null;

      @BeforeClass(alwaysRun = true)
      public void setUpEnvioDiretoMisturaOrigens() {


Util.startTestMsg(TestEnvioDiretoMisturaOrigens.class.getSimpleName());
            driver = Util.driver;
            Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_DIRETO_MISTURA_ORIGENS);
      }

      @Test(groups = { "insert", "verify", "campos" })
      public void abreTelaInsertEnvioDiretoMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
DiretoMisturaOrigens.NOVO_REGISTRO));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.NOVO_REGISTRO).click()
;
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertEnvioDiretoMisturaOrigens")
      public void insertUniqueEnvioDiretoMisturaOrigens() {
```

```
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsEnvioDiretoMistu
raOrigens.FORM_TIPO_PRODUTO));


Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_TIPO_PRODUTO,
Padroes.NOME_PRODUTO);
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsEnvioDiretoMistu
raOrigens.FORM_NOVO_RECEBIMENTO));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_OBSERVACAO).sendK
eys(Padroes.OBSERVACAO);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_NOVO_RECEBIMENTO)
.click();
            Util.wait
                        .until(ExpectedConditions.visibilityOfElementLocated
(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).send
Keys(dataAtual);

      Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_ORIGEM,
Padroes.NOME_ORIGEM);
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsEnvioDiretoMistu
raOrigens.FORM_TALHOES_REQUIRED));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_QTDE)
.sendKeys("60");

      Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_LOCAL_RECEBIMENTO,
```

```
Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_RECEBIMENTO).s
endKeys(dataAtual);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_INCLU
IR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));



driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_NOVO_RECEBIMENTO)
.click();
            Util.wait
                        .until(ExpectedConditions.visibilityOfElementLocated
(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).send
Keys(dataAtual);

      Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_ORIGEM,
Padroes.NOME_ORIGEM_2);
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsEnvioDiretoMistu
raOrigens.FORM_TALHOES_REQUIRED));
            Util.waitJsBlock();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_QTDE)
.sendKeys("90");

      Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_EMBALAGEM,
Padroes.NOME_EMBALAGEM);
```

```
Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_LOCAL_RECEBIMENTO,
Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_RECEBIMENTO).s
endKeys(dataAtual);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_INCLU
IR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));



driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_NOVA_CARGA).click
();
            Util.wait.until(ExpectedConditions
                        .visibilityOfElementLocated(ElementsEnvioDiretoMistu
raOrigens.FORM_CARGA_DESTINO));


Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_DESTINO,
Padroes.NOME_DESTINO_CADASTRO);
            Util.waitJsBlock();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_QTDE).sendK
eys("150");


Util.selectOption(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_DT_ENTREGA)
.sendKeys(dataAtual);
```

```
driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_COMPLEMENTO
).sendKeys(Padroes.COMPLEMENTO);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_NOTA_FISCAL
).sendKeys(Padroes.NOTA_FISCAL);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_INCLUIR).cl
ick();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);
            codigoEnvio =
Util.capturaCodigo(driver.findElement(ElementsRastreador.MODAL_CONFIRMA).ge
tText());
            Util.testMsg("TestEnvioDiretoMisturaOrigens", "Código: " +
codigoEnvio);

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
DiretoMisturaOrigens.NOVO_REGISTRO));
      }

      @Test(groups = { "edit" })
```

```
	public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
DiretoMisturaOrigens.NOVO_REGISTRO));
		Util.acionaElemento(ElementsEnvioDiretoMisturaOrigens.TABLE,
ElementsEnvioDiretoMisturaOrigens.TABLE_VALUE,
				Padroes.NOME_DESTINO_CADASTRO,
ElementsRastreador.TABLE_EDITAR);
	}

	@Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
	public void editUniqueEnvioDiretoMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
DiretoMisturaOrigens.FORM_OBSERVACAO));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_OBSERVACAO).sendK
eys(Padroes.EDITADO);
		Util.wait.until(new ExpectedCondition<Boolean>() {
			@Override
			public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_EDITAR_SEGUNDO_RE
CEBIMENTO).click();
				return
driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).isDi
splayed();
			}
		});


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).clea
r();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).send
Keys(dataAtual);
```

```
driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_QTDE)
.clear();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_QTDE)
.sendKeys("120");


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_RECEBIMENTO).c
lear();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_RECEBIMENTO).s
endKeys(dataAtual);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_ATUAL
IZAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_EDITAR_PRIMEIRO_R
ECEBIMENTO).click();
                        return
driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).isDi
splayed();
                  }
            });


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).clea
r();
```

```
driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_COLHEITA).send
Keys(dataAtual);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_QTDE)
.clear();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_QTDE)
.sendKeys("50");


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_RECEBIMENTO).c
lear();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_DT_RECEBIMENTO).s
endKeys(dataAtual);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_RECEBIMENTO_ATUAL
IZAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.wait.until(new ExpectedCondition<Boolean>() {
                  @Override
                  public Boolean apply(WebDriver arg0) {


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_EDITAR_CARGA).cli
ck();
                        return
driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_QTDE).isDis
played();
                  }
            });
```

```
driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_QTDE).clear
();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_QTDE).sendK
eys("170");


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_DT_ENTREGA)
.clear();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_DT_ENTREGA)
.sendKeys(dataAtual);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_COMPLEMENTO
).sendKeys(Padroes.EDITADO);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_NOTA_FISCAL
).clear();


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_NOTA_FISCAL
)
                        .sendKeys(Padroes.NOTA_FISCAL_EDITADO);


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_CARGA_ATUALIZAR).
click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
```

```
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
DiretoMisturaOrigens.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueEnvioDiretoMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
DiretoMisturaOrigens.NOVO_REGISTRO));
            Util.acionaElemento(ElementsEnvioDiretoMisturaOrigens.TABLE,
ElementsEnvioDiretoMisturaOrigens.TABLE_VALUE,
                        Padroes.NOME_DESTINO_CADASTRO + Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertEnvioDiretoMisturaOrigens")
      public void validacaoCamposObrigatoriosEnvioDiretoMisturaOrigens() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
DiretoMisturaOrigens.FORM_SALVAR));


driver.findElement(ElementsEnvioDiretoMisturaOrigens.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));


Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsEnvioDiretoMisturaOrigens.MSG_VALIDA_CAMPOS,
```

```
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_DIRETO_MISTURA_ORIGENS);

      Util.endTestMsg(TestEnvioDiretoMisturaOrigens.class.getSimpleName());
      }
}

package perfil.geral.rastreamento.envioDiretoMisturaOrigens;

import org.openqa.selenium.By;

public class ElementsEnvioDiretoMisturaOrigens {

    // Elementos da tela principal - Botões
    public static final By NOVO_REGISTRO = By

.id("startForm:telaEnvioDiretoMisturaOrigensbasepanelcontentlistgiNovoRegis
tro");

    // Elementos da tela principal - Tabela
    public static final String TABLE =
"startForm:telaEnvioDiretoMisturaOrigensdataTable:";
    public static final By TABLE_ORDER = By

.id("startForm:telaEnvioDiretoMisturaOrigensdataTable:telaEnvioDiretoMistur
aOrigensdataTablecolumn3header:sortDiv");
    public static final String TABLE_VALUE =
":telaEnvioDiretoMisturaOrigensdataTablecolumn3otValue3";

    // Elementos do formulário - Geral
    public static final By FORM_TIPO_PRODUTO = By
          .id("startForm:telaEnvioDiretoMisturaOrigenspnFormtipoProduto");
    public static final By FORM_OBSERVACAO = By
          .id("startForm:telaEnvioDiretoMisturaOrigenspnFormobservacao");
```

```
    public static final By FORM_NOVO_RECEBIMENTO = By

.id("startForm:telaEnvioDiretoMisturaOrigenspanelIngredientespnNovoIngredie
ntegiNovoRegistro");
    public static final By FORM_NOVA_CARGA = By

.id("startForm:telaEnvioDiretoMisturaOrigenspnFormpnNovoRegistrogiNovoRegis
tro");
    public static final By FORM_EDITAR_PRIMEIRO_RECEBIMENTO = By

.id("startForm:telaEnvioDiretoMisturaOrigensdataTableIngredientes:0:telaEnv
ioDiretoMisturaOrigensdataTableIngredientespnDeleteUpdategiUpdate");
    public static final By FORM_EDITAR_SEGUNDO_RECEBIMENTO = By

.id("startForm:telaEnvioDiretoMisturaOrigensdataTableIngredientes:1:telaEnv
ioDiretoMisturaOrigensdataTableIngredientespnDeleteUpdategiUpdate");
    public static final By FORM_EDITAR_CARGA = By

.id("startForm:telaEnvioDiretoMisturaOrigensdataTableCargas:0:telaEnvioDire
toMisturaOrigensdataTableCargaspnEditDeletegiEdit");
    public static final By FORM_SALVAR = By

.id("startForm:telaEnvioDiretoMisturaOrigensbasepanelcontentformbtsalvarbt"
);

    // Elementos do formulário - Recebimento
    public static final By FORM_DT_COLHEITA = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopndataColheit
aInputDate");
    public static final By FORM_ORIGEM = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopnorigem");
    public static final By FORM_RECEBIMENTO_QTDE = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopnquantidade"
);
    public static final By FORM_EMBALAGEM = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopnembalagem")
;
```

```
    public static final By FORM_LOCAL_RECEBIMENTO = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopndestino");
    public static final By FORM_DT_RECEBIMENTO = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopndataEntrega
InputDate");
    public static final By FORM_RECEBIMENTO_INCLUIR = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopnpnBotoesbtI
ncluirIngredientebt");
    public static final By FORM_RECEBIMENTO_ATUALIZAR = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopnpnBotoesbtA
tualizarIngredientebt");

    // Elementos do formulário - Carga
    public static final By FORM_CARGA_DESTINO = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnDadosGeraisdestin
o");
    public static final By FORM_CARGA_QTDE = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnDadosGeraisquanti
dade");
    public static final By FORM_CARGA_EMBALAGEM = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnDadosGeraisembala
gem");
    public static final By FORM_CARGA_DT_ENTREGA = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnDadosGeraisdataEn
tregaInputDate");
    public static final By FORM_CARGA_COMPLEMENTO = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnDadosGeraiscomple
mento");
    public static final By FORM_CARGA_NOTA_FISCAL = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnDadosGeraisnotaFi
scal");
```

```
    public static final By FORM_CARGA_INCLUIR = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnBotoesbtIncluirCa
rgabt");
    public static final By FORM_CARGA_ATUALIZAR = By

.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnCargapnBotoesbtAtualizar
Cargabt");
      public static final By FORM_TALHOES_REQUIRED =
By.id("formModalPanel:telaEnvioDiretoMisturaOrigenspnRecebimentopngiRequire
d");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Tipo de produto] é obrigatório." + "\n"
          + "O campo [Cargas de Recebimento] é obrigatório." + "\n"
          + "O campo [Cargas] é obrigatório.";

}


package perfil.geral.rastreamento.envioDireto;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.MsgSistema;
import util.Util;

public class TestEnvioDireto {
      WebDriver driver;
      String dataAtual = Util.dataAtual();
      public static String codigoEnvio = null;

      @BeforeClass(alwaysRun = true)
```

```
	public void setUpEnvioDireto() {
		Util.startTestMsg(TestEnvioDireto.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_DIRETO);
	}

	@Test(groups = { "insert", "verify", "campos" })
	public void abreTelaInsertEnvioDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.NOVO_REGISTRO));
		driver.findElement(ElementsEnvioDireto.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertEnvioDireto")
	public void insertUniqueEnvioDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_DT_COLHEITA));


driver.findElement(ElementsEnvioDireto.FORM_DT_COLHEITA).sendKeys(dataAtual
);
		Util.selectOption(ElementsEnvioDireto.FORM_ORIGEM,
Padroes.NOME_ORIGEM);
		Util.waitJsBlock();
		Util.selectOption(ElementsEnvioDireto.FORM_PRODUTO,
Padroes.NOME_PRODUTO);


driver.findElement(ElementsEnvioDireto.FORM_DT_RECEBIMENTO).sendKeys(dataAt
ual);
		Util.selectOption(ElementsEnvioDireto.FORM_LOCAL_RECEBIMENTO,
Padroes.NOME_MEU_CADASTRO);


driver.findElement(ElementsEnvioDireto.FORM_NOTA_FISCAL).sendKeys("134567")
```

```
;


driver.findElement(ElementsEnvioDireto.FORM_PLACA_VEICULO).sendKeys("ABC -
7894");


driver.findElement(ElementsEnvioDireto.FORM_NOME_MOTORISTA).sendKeys("Gisli
ane Wiatrowsk");


driver.findElement(ElementsEnvioDireto.FORM_OBSERVACAO).sendKeys("Observaca
o Envio Direto");


      driver.findElement(ElementsEnvioDireto.FORM_NOVA_CARGA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_CARGA_DESTINO));
            Util.selectOption(ElementsEnvioDireto.FORM_CARGA_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_QUESTOES_QUESTIONARIO));


driver.findElement(ElementsEnvioDireto.FORM_CARGA_QTDE).sendKeys("35");
            Util.selectOption(ElementsEnvioDireto.FORM_CARGA_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsEnvioDireto.FORM_CARGA_DT_ENTREGA).sendKeys(data
Atual);


driver.findElement(ElementsEnvioDireto.FORM_CARGA_PLACA_VEICULO).sendKeys("
ABC 1545");
```

```
driver.findElement(ElementsEnvioDireto.FORM_CARGA_NOME_MOTORISTA).sendKeys(
"Natália Casassola");


driver.findElement(ElementsEnvioDireto.FORM_CARGA_COMPLEMENTO).sendKeys("pp
");


driver.findElement(ElementsEnvioDireto.FORM_CARGA_PRECO).sendKeys("1.19");
            String[] respostas_um = { "De Acordo", "100%", "Acima de 51%",
"0-20%", "0-10%", "0-15%", "0-3%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsEnvioDireto.NOME_TELA, respostas_um);

      driver.findElement(ElementsEnvioDireto.FORM_CARGA_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));


      driver.findElement(ElementsEnvioDireto.FORM_NOVA_CARGA).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_CARGA_DESTINO));
            Util.selectOption(ElementsEnvioDireto.FORM_CARGA_DESTINO,
Padroes.NOME_DESTINO_UTILIZADO);


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_QUESTOES_QUESTIONARIO));


driver.findElement(ElementsEnvioDireto.FORM_CARGA_QTDE).sendKeys("28");
            Util.selectOption(ElementsEnvioDireto.FORM_CARGA_EMBALAGEM,
Padroes.NOME_EMBALAGEM);


driver.findElement(ElementsEnvioDireto.FORM_CARGA_DT_ENTREGA).sendKeys(data
Atual);
```

```
driver.findElement(ElementsEnvioDireto.FORM_CARGA_PLACA_VEICULO).sendKeys("
BLK 4988");


driver.findElement(ElementsEnvioDireto.FORM_CARGA_NOME_MOTORISTA).sendKeys(
"Veridiana Freitas");


driver.findElement(ElementsEnvioDireto.FORM_CARGA_COMPLEMENTO).sendKeys("su
per");


driver.findElement(ElementsEnvioDireto.FORM_CARGA_PRECO).sendKeys("2.15");
            String[] respostas_dois = { "Fora do Padrão", "Não se aplica",
"Abaixo de 51%", "21-50%", "11-30%", "16-30%",
                        "4-7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsEnvioDireto.NOME_TELA, respostas_dois);

      driver.findElement(ElementsEnvioDireto.FORM_CARGA_INCLUIR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            Util.waitJsBlock();
            driver.findElement(ElementsEnvioDireto.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_CONFIRMA));
            Assert.assertEquals(


driver.findElement(ElementsRastreador.MODAL_CONFIRMA).getText().equals(MsgS
istema.CADASTRO_NULL), false,
                        MsgErro.FAIL_CADASTRO);
            codigoEnvio =
Util.capturaCodigo(driver.findElement(ElementsRastreador.MODAL_CONFIRMA).ge
tText());
```

```
            Util.testMsg("TestEnvioDireto", "Código: " + codigoEnvio);

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.NOVO_REGISTRO));
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.NOVO_REGISTRO));
            Util.acionaElemento(ElementsEnvioDireto.TABLE,
ElementsEnvioDireto.TABLE_VALUE, Padroes.NOME_ORIGEM
                        + Padroes.EDITADO, ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueEnvioDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_DT_COLHEITA));

      driver.findElement(ElementsEnvioDireto.FORM_DT_COLHEITA).clear();


driver.findElement(ElementsEnvioDireto.FORM_DT_COLHEITA).sendKeys(dataAtual
);

      driver.findElement(ElementsEnvioDireto.FORM_DT_RECEBIMENTO).clear();


driver.findElement(ElementsEnvioDireto.FORM_DT_RECEBIMENTO).sendKeys(dataAt
ual);

      driver.findElement(ElementsEnvioDireto.FORM_NOTA_FISCAL).clear();
```

```
driver.findElement(ElementsEnvioDireto.FORM_NOTA_FISCAL).sendKeys("456789")
;

      driver.findElement(ElementsEnvioDireto.FORM_PLACA_VEICULO).clear();

driver.findElement(ElementsEnvioDireto.FORM_PLACA_VEICULO).sendKeys("APE
4795");

driver.findElement(ElementsEnvioDireto.FORM_OBSERVACAO).sendKeys(Padroes.ED
ITADO);

      driver.findElement(ElementsEnvioDireto.FORM_EDITAR_CARGA_UM).click();

Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_CARGA_QTDE));

      driver.findElement(ElementsEnvioDireto.FORM_CARGA_QTDE).clear();

driver.findElement(ElementsEnvioDireto.FORM_CARGA_QTDE).sendKeys("38");

driver.findElement(ElementsEnvioDireto.FORM_CARGA_DT_ENTREGA).clear();

driver.findElement(ElementsEnvioDireto.FORM_CARGA_DT_ENTREGA).sendKeys(data
Atual);

driver.findElement(ElementsEnvioDireto.FORM_CARGA_PLACA_VEICULO).clear();

driver.findElement(ElementsEnvioDireto.FORM_CARGA_PLACA_VEICULO).sendKeys("
ABC 7080");

driver.findElement(ElementsEnvioDireto.FORM_CARGA_NOME_MOTORISTA).clear();
```

```
driver.findElement(ElementsEnvioDireto.FORM_CARGA_NOME_MOTORISTA).sendKeys(
"Dani Lopes");


driver.findElement(ElementsEnvioDireto.FORM_CARGA_COMPLEMENTO).clear();


driver.findElement(ElementsEnvioDireto.FORM_CARGA_COMPLEMENTO).sendKeys("su
per");


driver.findElement(ElementsEnvioDireto.FORM_CARGA_NOTA_FISCAL).clear();


driver.findElement(ElementsEnvioDireto.FORM_CARGA_NOTA_FISCAL).sendKeys("36
9258");

      driver.findElement(ElementsEnvioDireto.FORM_CARGA_PRECO).clear();


driver.findElement(ElementsEnvioDireto.FORM_CARGA_PRECO).sendKeys("1.29");
            String[] resposta_tres = { "Fora do Padrão", "Não se aplica",
"Não se aplica", "Não se aplica", "Acima de 31%",
                        "Acima de 31%", "Acima de 7%" };
            Util.responderQuestionarioGpa(ElementsRastreador.FORMMODAL,
ElementsEnvioDireto.NOME_TELA, resposta_tres);

      driver.findElement(ElementsEnvioDireto.FORM_CARGA_ATUALIZAR).click();


Util.wait.until(ExpectedConditions.invisibilityOfElementLocated(ElementsRas
treador.MODAL));
            driver.findElement(ElementsEnvioDireto.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.NOVO_REGISTRO));
	}

	@Test(groups = { "delete" })
	public void deleteUniqueEnvioDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.NOVO_REGISTRO));
		Util.acionaElemento(ElementsEnvioDireto.TABLE,
ElementsEnvioDireto.TABLE_VALUE, Padroes.NOME_ORIGEM
				+ Padroes.EDITADO,
ElementsRastreador.TABLE_EXCLUIR);
		Util.excluirRegistro(true, true);
	}

	@Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertEnvioDireto")
	public void validacaoCamposObrigatoriosEnvioDireto() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsEnvio
Direto.FORM_SALVAR));
		driver.findElement(ElementsEnvioDireto.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));


Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
				ElementsEnvioDireto.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);

	driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
	}
```

```
	@AfterClass(alwaysRun = true)
	public void tearDown() {
		Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.ENVIO_DIRETO);
		Util.endTestMsg(TestEnvioDireto.class.getSimpleName());
	}
}

package perfil.geral.rastreamento.envioDireto;

import org.openqa.selenium.By;

public class ElementsEnvioDireto {

	// Elementos da Tela Principal - Botões
	public static final By NOVO_REGISTRO =
By.id("startForm:telaEnvioDireto3GbasepanelcontentlistgiNovoRegistro");
	public static final By TABLE_IMPRIMIR =
By.id(":telaEnvioDireto3GdataTablepnEtiquetabtImprimirEtiqueta");
	public static final By FILTRO_NOVO =
By.id("startForm:telaEnvioDireto3Gbasepanelcontentfiltertoppgrightlknewfilt
er");

	// Elementos da Tela Principal - Tabela
	public static final String TABLE =
"startForm:telaEnvioDireto3GdataTable:";
	public static final By TABLE_ORDER = By
			.id("startForm:telaEnvioDireto3GdataTable:telaEnvioDireto
3GdataTablecolumn2header:sortDiv");
	public static final String TABLE_VALUE =
":telaEnvioDireto3GdataTablecolumn2otValue2";

	// Elementos do Filtro
	public static final By FILTRO_LIMPAR = By
			.id("startForm:telaEnvioDireto3Gbasepanelcontentfiltercontrolslkclearfilter");
	public static final By FILTRO_BUSCAR =
By.id("startForm:telaEnvioDireto3Gbasepanelcontentfiltercontrolslkfilter");

	// Elementos da tela de impressão
	public static final By ETIQUETA_GERAR =
```

```
By.id("startForm:telaEnvioDireto3GpnGeracaoEtiquetaclGerar");

      // Elementos do Formulário - Geral
      public static final By FORM_ORIGEM =
By.id("startForm:telaEnvioDireto3Gbasepanelcontentformpn1origem");
      public static final By FORM_DT_COLHEITA = By
                  .id("startForm:telaEnvioDireto3Gbasepanelcontentformpn1da
taColheitaInputDate");
      public static final By FORM_DT_RECEBIMENTO = By
                  .id("startForm:telaEnvioDireto3Gbasepanelcontentformpn3da
taEntregaInputDate");
      public static final By FORM_PRODUTO =
By.id("startForm:telaEnvioDireto3Gbasepanelcontentformpn1tipoProduto");
      public static final By FORM_LOCAL_RECEBIMENTO = By
                  .id("startForm:telaEnvioDireto3Gbasepanelcontentformpn3lo
calDeRecebimento");
      public static final By FORM_NOTA_FISCAL = By
                  .id("startForm:telaEnvioDireto3Gbasepanelcontentformpn3no
taFiscalRecebimento");
      public static final By FORM_PLACA_VEICULO =
By.id("startForm:telaEnvioDireto3Gbasepanelcontentformpn3placaVeiculo");
      public static final By FORM_NOME_MOTORISTA = By
                  .id("startForm:telaEnvioDireto3Gbasepanelcontentformpn3no
meMotorista");
      public static final By FORM_OBSERVACAO =
By.id("startForm:telaEnvioDireto3Gbasepanelcontentformpn3observacao");
      public static final By FORM_NOVA_CARGA = By
                  .id("startForm:telaEnvioDireto3GbasepanelcontentformpnNov
oRegistrogiNovoRegistro");
      public static final By FORM_EDITAR_CARGA_UM = By
                  .id("startForm:telaEnvioDireto3GdataTableCargas:0:telaEnv
ioDireto3GdataTableCargaspnEditDeletegiEdit");
      public static final By FORM_SALVAR =
By.id("startForm:telaEnvioDireto3Gbasepanelcontentformbtsalvarbt");

      // Elementos do Formulário - Carga
      public static final By FORM_CARGA_DESTINO =
By.id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGeraisdestino");
      public static final By FORM_CARGA_INCLUIR =
By.id("formModalPanel:telaEnvioDireto3GpnCargapnBotoesbtIncluirCargabt");
      public static final By FORM_CARGA_EMBALAGEM =
```

```
By.id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGeraisembalagem");
      public static final By FORM_CARGA_QTDE =
By.id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGeraisquantidade");
      public static final By FORM_CARGA_DT_ENTREGA = By
                  .id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGerais
dataEntregaInputDate");
      public static final By FORM_CARGA_PLACA_VEICULO = By
                  .id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGerais
placaVeiculo");
      public static final By FORM_CARGA_NOME_MOTORISTA = By
                  .id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGerais
nomeMotorista");
      public static final By FORM_CARGA_COMPLEMENTO = By
                  .id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGerais
complemento");
      public static final By FORM_CARGA_PRECO =
By.id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGeraispreco");
      public static final By FORM_CARGA_ATUALIZAR = By
                  .id("formModalPanel:telaEnvioDireto3GpnCargapnBotoesbtAtu
alizarCargabt");
      public static final By FORM_CARGA_NOTA_FISCAL = By
                  .id("formModalPanel:telaEnvioDireto3GpnCargapnDadosGerais
notaFiscal");
      public static final String NOME_TELA = "telaEnvioDireto3GpnCargapn";
      public static final By FORM_QUESTOES_QUESTIONARIO = By
                  .id("formModalPanel:telaEnvioDireto3GpnCargapnQuestionari
opnQuestoespbi_topico1pg");

      // Mensagem de validação dos campos obrigatorios
      public static String MSG_VALIDA_CAMPOS = "Erro" + "\n" + "O campo
[Origem] é obrigatório." + "\n"
                  + "O campo [Produto] é obrigatório." + "\n" + "O campo
[Local de recebimento] é obrigatório." + "\n"
                  + "O campo [Cargas] é obrigatório.";
}

package perfil.geral.rastreamento.descarte;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.testng.Assert;
import org.testng.annotations.*;

import perfil.geral.rastreamento.recebimento.ElementsRecebimento;
import perfil.geral.rastreamento.recebimento.TestRecebimento;
import perfil.geral.util.Padroes;

import util.ElementsRastreador;
import util.MsgErro;
import util.Util;

public class TestDescarte {
	WebDriver driver;
	public static String codigoRastreamento = null;
	public static BigDecimal quantidadeDescarte = new BigDecimal(20);
	public static BigDecimal quantidadeSobrepeso = new BigDecimal(20);

	@BeforeClass(alwaysRun = true)
	public void setUpDescarte() throws Exception {
		Util.startTestMsg(TestDescarte.class.getSimpleName());
		driver = Util.driver;
		Util.abreGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.DESCARTE);
	}

	@Test(groups = { "insert", "verify", "campos" })
	public void abreTelaInsertDescarte() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.NOVO_REGISTRO));
		driver.findElement(ElementsDescarte.NOVO_REGISTRO).click();
	}

	@Test(groups = { "insert" }, dependsOnMethods =
"abreTelaInsertDescarte")
	public void insertUniqueDescarte() {
		codigoRastreamento = TestRecebimento.codigosRecebimento.get(1);
```

```
Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.FORM_CODIGO_DESCARTE));


driver.findElement(ElementsDescarte.FORM_CODIGO_DESCARTE).sendKeys(codigoRa
streamento);
            driver.findElement(ElementsDescarte.FORM_PESQUISAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.FORM_QTD_DESCARTADA));



driver.findElement(ElementsDescarte.FORM_QTD_DESCARTADA).sendKeys(quantidad
eDescarte.toString());
            TestRecebimento.totalEstoqueSecondRecebimento.set(1,


TestRecebimento.totalEstoqueSecondRecebimento.get(1).add(quantidadeDescarte
)); // Índice


                                    // enviado
            TestRecebimento.totalEstoqueSecondRecebimento.set(2,
TestRecebimento.totalEstoqueSecondRecebimento.get(2)
                        .subtract(quantidadeDescarte)); // Índice disponível


driver.findElement(ElementsDescarte.FORM_QTD_SOBREPESO).sendKeys(quantidade
Sobrepeso.toString());
            TestRecebimento.totalEstoqueSecondRecebimento.set(1,


TestRecebimento.totalEstoqueSecondRecebimento.get(1).add(quantidadeSobrepes
o)); // Índice
```

```
                                    // enviado
            TestRecebimento.totalEstoqueSecondRecebimento.set(2,
TestRecebimento.totalEstoqueSecondRecebimento.get(2)
                        .subtract(quantidadeSobrepeso)); // Índice
disponível
            driver.findElement(ElementsDescarte.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.FORM_SUCESSO_NAO));

      driver.findElement(ElementsRastreador.FORM_SUCESSO_NAO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.NOVO_REGISTRO));

            // Impressão das mensagens para verificação de estoque
            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(0).setScale(2) + " | Env:
"
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(1).setScale(2) + " |
Disp: "
                        +
TestRecebimento.totalEstoqueSecondRecebimento.get(2).setScale(2) + " } /
EXPECTED");
      }

      @Test(groups = { "insert" }, dependsOnMethods =
"insertUniqueDescarte")
      public void verifyEstoqueRecebimentoAfterDescarte() {
            List<String> valorEstoqueTabelaSecondRecebimento = new
ArrayList<String>();

            driver.findElement(ElementsRastreador.RECEBIMENTO).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsReceb
```

```
imento.NOVO_REGISTRO));

            // Executa asserts do segundo recebimento (recebimento,
enviado,
            // disponível).


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_RECEBIDO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_ENVIADO,
                        TestRecebimento.codigosRecebimento.get(1)));


valorEstoqueTabelaSecondRecebimento.add(Util.retornaValorTabela(ElementsRec
ebimento.TABLE,
                        ElementsRecebimento.TABLE_VALUE_CODIGO_RASTREAMENTO,
ElementsRecebimento.TABLE_VALUE_TOTAL_DISPONIVEL,
                        TestRecebimento.codigosRecebimento.get(1)));

            // Executa asserts do primeiro recebimento (recebimento,
enviado,
            // disponível).
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(0),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(0).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(1),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(1).setScale(2).toString(),
MsgErro.FAIL_CADASTRO);
            Assert.assertEquals(valorEstoqueTabelaSecondRecebimento.get(2),
TestRecebimento.totalEstoqueSecondRecebimento
                        .get(2).setScale(2).toString(),
```

```
MsgErro.FAIL_CADASTRO);

            Util.testMsg("Recebimento: #" +
TestRecebimento.codigosRecebimento.get(1), "{ Rec: "
                        + valorEstoqueTabelaSecondRecebimento.get(0) + " |
Env: " + valorEstoqueTabelaSecondRecebimento.get(1)
                        + " | Disp: " +
valorEstoqueTabelaSecondRecebimento.get(2) + " } / RESULT");
      }

      @Test(groups = { "edit" })
      public void setUpEdit() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.NOVO_REGISTRO));
            Util.acionaElemento(ElementsDescarte.TABLE,
ElementsDescarte.TABLE_VALUE, Padroes.NOME_PRODUTO,
                        ElementsRastreador.TABLE_EDITAR);
      }

      @Test(groups = { "edit" }, dependsOnMethods = "setUpEdit")
      public void editUniqueDescarte() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.FORM_QTD_DESCARTADA));

      driver.findElement(ElementsDescarte.FORM_QTD_DESCARTADA).clear();


driver.findElement(ElementsDescarte.FORM_QTD_DESCARTADA).sendKeys("15");

      driver.findElement(ElementsDescarte.FORM_QTD_SOBREPESO).clear();


driver.findElement(ElementsDescarte.FORM_QTD_SOBREPESO).sendKeys("15");
            driver.findElement(ElementsDescarte.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
```

```
eador.FORM_EDICAO_FECHAR));

      driver.findElement(ElementsRastreador.FORM_EDICAO_FECHAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.NOVO_REGISTRO));
      }

      @Test(groups = { "delete" })
      public void deleteUniqueDescarte() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.NOVO_REGISTRO));
            Util.acionaElemento(ElementsDescarte.TABLE,
ElementsDescarte.TABLE_VALUE, Padroes.NOME_PRODUTO,
                        ElementsRastreador.TABLE_EXCLUIR);
            Util.excluirRegistro(true, true);
      }

      @Test(groups = { "verify", "campos" }, dependsOnMethods =
"abreTelaInsertDescarte")
      public void validacaoCamposObrigatoriosDescarte() {


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsDesca
rte.FORM_SALVAR));

            driver.findElement(ElementsDescarte.FORM_SALVAR).click();


Util.wait.until(ExpectedConditions.visibilityOfElementLocated(ElementsRastr
eador.MODAL_ERRO_TITLE));



Assert.assertEquals(driver.findElement(ElementsRastreador.MODAL_ERRO).getTe
xt(),
                        ElementsDescarte.MSG_VALIDA_CAMPOS,
MsgErro.FAIL_VALIDACAO_CAMPOS_OBRIGATORIOS);
```

```
      driver.findElement(ElementsRastreador.FORM_ERRO_FECHAR).click();
      }

      @AfterClass(alwaysRun = true)
      public void tearDown() {
            Util.fechaGrupoMenu(ElementsRastreador.RASTREAMENTO,
ElementsRastreador.DESCARTE);
            Util.endTestMsg("TestDescarte");
      }
}

package perfil.geral.rastreamento.descarte;

import org.openqa.selenium.By;

public class ElementsDescarte {

    // Elementos da Tela Principal - Tabela
    public static final String TABLE =
"startForm:telaDescarteRastreamento3GdataTable:";
    public static final By TABLE_ORDER = By

.id("startForm:telaDescarteRastreamento3GdataTable:telaDescarteRastreamento
3GdataTablecolumn2otHeader2");
    public static final String TABLE_VALUE =
":telaDescarteRastreamento3GdataTablecolumn2";

    // Elementos da Tela Principal - Botões
    public static final By NOVO_REGISTRO = By

.id("startForm:telaDescarteRastreamento3GbasepanelcontentlistgiNovoRegistro
");

    // Elementos do formulário - Campos
    public static final By FORM_CODIGO_DESCARTE = By
          .id("startForm:telaDescarteRastreamento3GpnFormidRastreamento");
    public static final By FORM_QTD_DESCARTADA = By

.id("startForm:telaDescarteRastreamento3GdataTableCargas:0:telaDescarteRast
reamento3GdataTableCargaspnEditDeletequant");
```

```
    public static final By FORM_QTD_SOBREPESO = By

.id("startForm:telaDescarteRastreamento3GdataTableCargas:0:telaDescarteRast
reamento3GdataTableCargaspnEditDeletequantSobrepeso");
    public static final By FORM_QTD = By

.id("startForm:telaDescarteRastreamento3GdataTableCargas:0:telaDescarteRast
reamento3GdataTableCargascolumn1otValue1");

    // Elementos do formulário - Botões
    public static final By FORM_PESQUISAR = By
          .id("startForm:telaDescarteRastreamento3GbtPesquisarbt");
    public static final By FORM_SALVAR = By

.id("startForm:telaDescarteRastreamento3Gbasepanelcontentformbtsalvarbt");
    public static final By FORM_LISTAR = By

.id("startForm:telaDescarteRastreamento3Gbasepanelcontentformbtlistarbt");

    // Mensagem de validação dos campos obrigatorios
    public static String MSG_VALIDA_CAMPOS = "Erro" + "\n"
          + "O campo [Código Descartado] é obrigatório." + "\n"
          + "O campo [Cargas] é obrigatório.";
}
```