

3 Processo

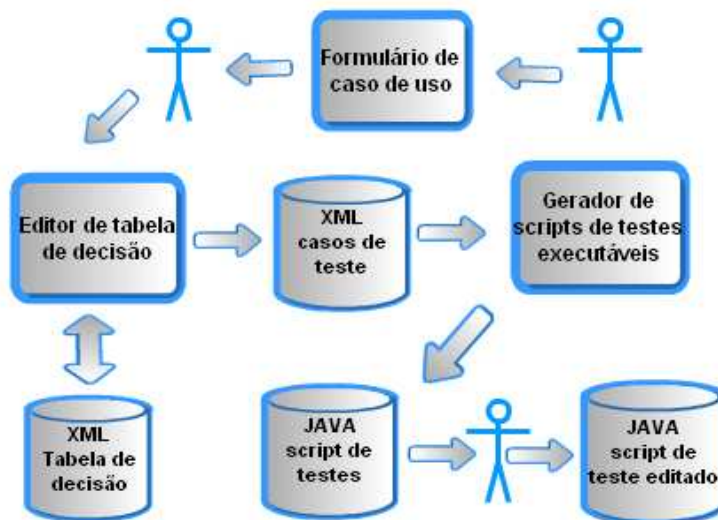


Figura 9 – Visão geral do processo.

Neste capítulo será apresentado o processo de testes presente neste trabalho. Como pode ser visualizado na figura acima, este processo inicia na escrita de um caso de uso segundo um padrão que, através de interpretação manual, facilite a um técnico criar uma tabela de decisão, na qual as suas condições e ações são identificadas a partir da descrição do caso de uso. Este caso de uso pode ser derivado de uma história de usuário. A partir da tabela de decisão para o caso de uso é gerado automaticamente um arquivo contendo os casos de teste semânticos extraídos da tabela. Este arquivo contendo os testes é utilizado como entrada para outra ferramenta, cuja função é gerar, também de forma automatizada, os scripts de teste automatizados contendo a implementação dos casos de teste. O script de teste pode então ser editado pelo desenvolvedor de testes para eventuais mudanças ou correções. A massa de teste gerada estará em conformidade com a funcionalidade descrita no caso de uso.

3.1. Redação da descrição dos casos de uso

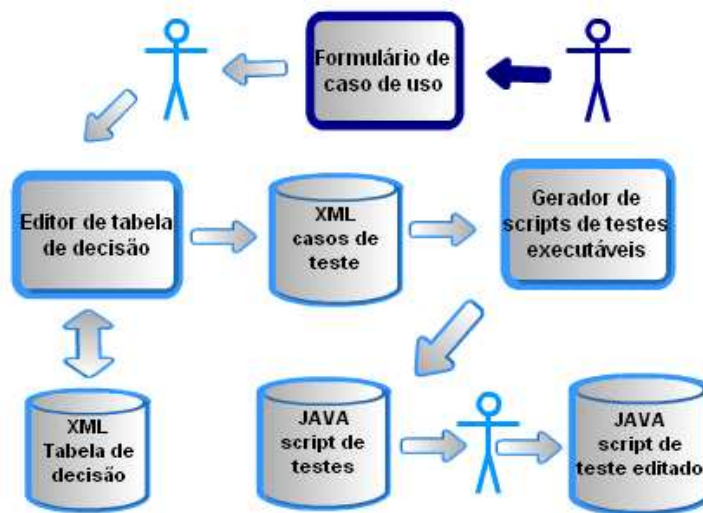


Figura 10 – Realce do processo de redação do caso de uso.

Dentre as várias informações que um caso de uso pode apresentar, está a descrição dos fluxos de eventos principais e secundários descrevendo a interação dos atores com o sistema. É comum o uso de linguagem natural na descrição destes fluxos, pois tal redação facilita o entendimento da funcionalidade que o caso de uso trata por pessoas que não possuem treinamento em computação, como pode ser o caso do cliente. Este formato também permite que o cliente acompanhe a formulação do caso de uso juntamente com o analista e valide a sua descrição. No entanto, a linguagem natural pode abrir margem para interpretações errôneas e ambiguidades na descrição do caso de uso.

Para evitar esse problema, a linguagem utilizada deve ser um português estruturado (Díaz et al, 2004)(Staa, 2010a), a fim de evitar a ocorrência de ambiguidades, mas que seja de fácil entendimento para quem não tem treinamento em computação. Este padrão também deve ser detalhado o suficiente para que se possa extrair da descrição o maior número de informações possíveis para a formação da tabela de decisão.

A tabela abaixo apresenta o *layout* de formulário, proposto por (Staa, 2010) e adaptado de (Cockburn, 2000), que foi usado neste trabalho com algumas pequenas adaptações.

Caso de uso	Nome (identificação) do caso de uso.	
Resumo	Descrição resumida do objetivo principal.	
Escopo	O que é abrangido pelo caso de uso.	
Atores	Nome (identificação) do ator	Descrição do interesse ou objetivo do ator
Invariante	Condições que deverão estar satisfeitas antes e após a execução do caso de uso.	
Pré-condições	Condições que precisam estar satisfeitas antes de iniciar o caso de uso.	
Fluxo principal	Sequência de ações esperadas.	
Fluxos alternativos	Sequências de ações que tratam de exceções e desvios do fluxo principal.	
Pós-condições	Condições que devem estar satisfeitas ao terminar a execução normal do caso de uso.	
Garantia mínima	Condições que devem estar satisfeitas sempre, independente de como termine o caso de uso.	
Requisitos	Requisitos adicionais, tais como requisitos não funcionais e outras características desejáveis.	
Regras de negócio	Especificações das regras a serem satisfeitas nos diferentes passos.	
Casos de uso correlatos	Relação de casos de uso correlacionados com o presente caso de uso.	
Artefatos correlatos	Relação de documentos, artigos, mensagens, fontes de informação, etc. que contém alguma informação relevante para o caso de uso.	

Tabela 2 – Layout de formulário de caso de uso.

Dentre os campos apresentados na tabela anterior, os de maior interesse para auxiliar na geração da tabela de decisão neste trabalho são:

- Atores: para identificar quando um ator interage com a interface;
- Pré-condições: por apresentar quais condições devem ser válidas. Pode, por exemplo, indicar que um determinado caso de uso foi executado antes deste, informação esta que pode ser inserida no campo Pré-condição da tabela de decisão (veja seção 4.1.1.4);
- Fluxo principal: por apresentar quais os passos esperados na execução normal do caso de uso. Estados ou comportamentos

esperados ao final do fluxo normal de execução podem ser usados como oráculos para os testes;

- Fluxos alternativos: fornecem os passos executados em comportamentos alternativos ou em caso de algum erro, desvio ou exceção relativos ao fluxo principal. As ações tomadas para fluxos alternativos resultantes de erros de validação de formulários, por exemplo, podem ser usadas como oráculos para os testes, a fim de verificar o correto tratamento destes erros;
- Pós-condições: as condições que devem estar satisfeitas ao final do fluxo normal do caso de uso, podem servir como oráculo para os testes;
- Garantia mínima: condições que devem estar satisfeitas independentemente do fluxo de execução do caso de uso;
- Regras de negócio: esta seção pode conter, entre outras coisas, as restrições de campos de entrada de dados, como a lista de caracteres válidos, número de caracteres, formato, etc.

Cada frase da descrição dos passos do fluxo principal ou dos secundários deve começar com um artigo definido seguido por um dos atores. Esta restrição visa facilitar a identificação das ações executadas pelo ator cujo papel é representado pelo usuário, já que as ações do usuário na interface provavelmente serão transformadas em condições na tabela de decisão.

O emprego de período composto na descrição dos fluxos do caso de uso pode tornar mais difícil o seu entendimento e testabilidade, visto que um mesmo passo poderia descrever mais de uma interação do ator com o sistema. Queremos que cada passo do fluxo represente uma única interação ou ação executada pelo ator para, se necessário, mapear este passo diretamente como uma condição na tabela de decisão. Assim, como proposto em (Díaz et al, 2004), a descrição do fluxo principal e secundários deve ser através de períodos simples e com substantivos, ou expressões substantivas, que designem uma única entidade.

Dado que queremos automatizar a interação do usuário com o sistema através de uma ferramenta, o padrão de redação dos passos do fluxo principal e secundário deve seguir um formato que permita identificar qual elemento da

interface será exercitado pelo usuário. Para facilitar esta identificação o padrão de escrita adotado para ações do usuário na interface deve ser o da figura abaixo.

`<Artigo definido><Ator><Padrão de ação><Nome do elemento>`

Figura 11 - Padrão para passo de fluxo.

Neste trabalho estamos interessados em casos de uso para sistemas com interface web, assim os elementos de interface e seus respectivos valores para “Padrão de ação” são:

Tipo de Elemento	Padrão de ação
Campo de entrada de texto.	“preenche o campo” “preenche campo” “preenche o” “preenche a” “preenche” “fornece” “fornece o” “fornece a”
Elementos que disparam ação através de cliques, como botões e <i>link</i> .	“clica em” “clica no” “clica na” “clica”
Checkbox e Radio Buttons	“marca” “marca a opção” “desmarca” “desmarca a opção”
Seletores	“seleciona” “seleciona o” “seleciona a” “seleciona a opção” “seleciona as opções”

Tabela 3 – Padrões para tipos de elementos.

Tomemos como exemplo uma tela de autenticação como na Figura 12. Para autenticar-se no sistema, o usuário precisa fornecer um nome, uma senha e pressionar um botão. O fluxo principal para o caso de uso deste exemplo é exibido

na Figura 13. Assim como proposto, cada ação sobre a interface é representada por um passo no fluxo. Estes passos devem estar enumerados de acordo com sua ordem de execução. O tipo de elemento sobre o qual a ação é executada pode ser determinado pelo padrão de escrita do passo, da mesma forma, o sujeito da ação, que deve ser um entre os atores da lista de atores, é facilmente identificado.

Nome	<input type="text"/>
Senha	<input type="text"/>
<input type="button" value="Ingressar"/>	

Figura 12 - Exemplo de tela de autenticação.

Trilha principal	<ol style="list-style-type: none"> 1. O usuário preenche o campo Nome. 2. O usuário preenche o campo Senha. 3. O usuário clica no botão Ingressar. 4. O sistema exibe a mensagem 'Bem-vindo'. 5. FIM DO CASO DE USO.
------------------	---

Figura 13 – Fluxo principal do caso de uso Realizar Autenticação.

Na descrição do caso de uso deve existir uma seção para informar quais são as restrições de cada campo presente na interface. A Figura 14 exemplifica quais seriam essas restrições para os campos presentes no exemplo da tela de autenticação. As ações tomadas quando estas restrições são violadas devem fazer parte do conjunto de fluxos alternativos para o caso de uso.

Regras de negócio	<p>-Restrições de Campos:</p> <p>Nome:</p> <ol style="list-style-type: none"> 1.não pode ser vazio. 2.deve estar cadastrado. <p>Senha:</p> <ol style="list-style-type: none"> 1.não pode ser vazio. 2.deve estar cadastrada, considerando o nome fornecido.
-------------------	---

Figura 14 – Restrições dos campos do caso de uso Realizar Autenticação.

Fluxos alternativos devem indicar o passo do fluxo principal para qual são uma alternativa e também informar, se for o caso, para qual passo do caminho

principal devem retornar a execução quando cessadas as instruções da execução alternativa. Fluxos que levem ao fim do caso de uso, seja o principal ou secundário, devem indicar o término do mesmo através de um passo com a frase “Fim do caso de uso”. Na Figura 15 é exemplificado como seria, seguindo o padrão proposto, os passos alternativos para o fluxo da Figura 13. Cada fluxo alternativo é identificado por um evento enumerado e indica à qual passo do fluxo principal está relacionado.

-Fluxos Alternativos

```

EVENTO E1: O usuário preenche o campo Nome com valor vazio.
ALTERNATIVA AO PASSO: 1.
    E1.1 O sistema exibe a mensagem 'O nome não pode ser vazio'.
    E1.2. RETORNA AO PASSO 1.
FIM EVENTO E1.

EVENTO E2: O usuário preenche o campo Nome com valor não cadastrado.
ALTERNATIVA AO PASSO: 1.
    E2.1. O sistema exibe a mensagem 'Nome e/ou senha não cadastrados'.
    E2.2. RETORNA AO PASSO 1.
FIM EVENTO E2.

EVENTO E3: O usuário preenche o campo senha com valor vazio.
ALTERNATIVA AO PASSO: 2.
    E3.1. O sistema exibe a mensagem 'A senha não pode ser vazia'.
    E3.2. RETORNA AO PASSO 2.
FIM EVENTO E3.

EVENTO E4: O usuário preenche o campo senha com valor não cadastrado.
ALTERNATIVA AO PASSO: 2.
    E4.1. O sistema exibe a mensagem 'Nome e/ou senha não cadastrados'.
    E4.2. RETORNA AO PASSO 2.
FIM EVENTO E4.

```

Figura 15 – Fluxos alternativos do caso de uso Realizar Autenticação.

No caso de seletores, se o restante do predicado, após o verbo, estiver no plural significa que o elemento admite mais de uma opção selecionada, caso contrário apenas uma. Essa informação pode ser importante ao preencher a tabela de decisão, como será explicado no capítulo 4. Para a geração dos testes também é importante saber quais as opções que um seletor apresenta, visto que cada uma será selecionada em algum teste. Para informar quais são essas opções o redator do caso de uso deve incluí-las, ao final do passo descrevendo o seletor, entre parênteses como na Figura 16, onde o seletor “Tipo de Sensor” possui as opções “Tipo A”, “Tipo B” e “Tipo C”.

- Título: Cadastrar Sensor
- Atores: usuário, sistema
- Pré-Requisito: executar caso de uso Cadastrar Tipo de Local
 executar caso de uso Cadastrar Local
 executar caso de uso Listar Locais
- Fluxo Principal:
1. O usuário clica em Novo Sensor.
 2. O usuário seleciona o Tipo de Sensor(TIPO A; TIPO B; TIPO C).
 - SE
 - 2A.1. O usuário seleciona o Tipo de Sensor TIPO A.
 - 2A.2. O usuário seleciona o Modelo de Sensor(MODELO A1; MODELO A2).
 - 2A.3. O usuário preenche campo Usuário.
 - 2A.4. O usuário preenche campo Senha.
 - SE
 - 2A.5. O usuário marca a checkbox Online.
 - 2A.6. O usuário preenche o campo Endereço.
 - SENÃO
 - 2B.1. O usuário seleciona o Tipo de Sensor(TIPO B; TIPO C).
 - SE
 - 2B.1A.1. O usuário seleciona o Tipo de sensor TIPO B.
 - 2B.1A.2. O usuário seleciona o Modelo de Sensor(MODELO B1).
 - SENÃO
 - 2B.1B.1. O usuário seleciona o Tipo de sensor TIPO C.
 - 2B.1B.2. O usuário seleciona o Modelo de Sensor(MODELO C1).
 - 2B.2. O usuário preenche o campo Endereço.
 3. O usuário fornece a descrição.
 4. O usuário marca a opção ativo.
 5. O usuário clica no botão salvar.
 6. O sistema exibe a tela de locais cadastrados.
 7. FIM DO CASO DE USO.

Figura 16 – Primeira parte do caso de uso Cadastrar Sensor.

Um exemplo de caso de uso com seletores e marcador é apresentado na Figura 16. Neste exemplo o usuário deseja criar uma entidade no sistema chamada sensor. Para isso ele precisa escolher, entre três candidatos, o tipo de sensor que será criado. Após este passo, o usuário deve selecionar qual o tipo de modelo do sensor, onde estes modelos variam de acordo com o tipo de sensor escolhido.

A escolha de um tipo de sensor leva a diferentes fluxos de execução, por exemplo, o sensor do tipo “Tipo A” pode ser *online* ou *off-line*, e requer a entrada de informações adicionais, como “Usuário” e “Senha”, para a utilização do sensor físico. Já, tanto para sensores do tipo “Tipo B” quanto do “Tipo C”, a lista de modelos disponíveis é alterada. No entanto, estes fluxos mutuamente exclusivos, se executados corretamente, cumprem a finalidade do caso de uso que é, no caso em questão, a criação de um sensor. Dessa forma, é da opinião do autor que, para a identificação das condições e passos a serem automatizados, é mais fácil que estes potenciais fluxos alternativos, que criam um sensor, estejam presentes no fluxo principal. A identificação de diferentes caminhos no fluxo principal dar-se-ia então pelo uso das palavras-chave “SE” e “SENÃO” e do padrão de numeração utilizados para identificar os passos.

Caminhos mutuamente exclusivos são identificados por terem o mesmo numeral acompanhado de letras diferentes, como no caso dos passos do caminho 2A e 2B, que são desmembramentos do passo 2. O passo 2B.1 por sua vez é desmembrado nos caminhos mutuamente exclusivos 2B.1A e 2B.1B. Passos que sofreram desmembramento, como o passo 2 e 2B.1, não se tornarão condições na tabela principal, apenas os passos resultados do desmembramento, desde que também não sofram desmembramento. Com esta numeração o uso das palavras-chave “SE” e “SENÃO” no fluxo principal poderia ser descartado, mas seu emprego visa facilitar a leitura. Se necessário, pode-se aplicar outras estruturas de controle na descrição dos passos dos fluxos como: “**ENQUANTO** <condição> **FAÇA**”, “**ATÉ QUE** <condição> **FAÇA**”, “**APÓS** <condição> **FAÇA**”, “**PARA CADA** <elemento> (contido em | no) <conjunto> **FAÇA**” (Staa, 2010a) (Díaz et al., 2004).

Os fluxos alternativos constariam assim de caminhos que não satisfazem a execução da funcionalidade do caso de uso, como, por exemplo, o usuário pressionar o botão cancelar (não criando o sensor) ou algum erro de validação dos campos. A Figura 17 apresenta os fluxos alternativos para o caso de uso anterior e as restrições para cada campo de texto.

Embora o formulário proposto e, principalmente, a estrutura para descrever os fluxos, possam ser mais complicados para redigir e ler, do que muitas propostas presentes na literatura (Cockburn, 2000), o autor acha que um formato menos informal e a numeração de eventos, exceções e passos que possam ocorrer, auxilia na identificação das condições e na elicitação de quais testes serão necessários. Assim, essa formalização torna-se importante para a identificação das condições e ações que farão parte da subsequente tabela de decisão.

-Restrições de Campos:

Endereço:

- 1.não pode ser vazio.
- 2.deve estar semanticamente correto.

Usuário:

- 1.não pode ser vazio.
- 2.deve estar cadastrado.

Senha:

- 1.não pode ser vazio.
- 2.deve estar cadastrado.

-Fluxos Alternativos:

EVENTO E1: O usuário não seleciona o Tipo de Sensor.

ALTERNATIVA AO PASSO: 2.

E1.1. O sistema exibe a mensagem 'É preciso selecionar o tipo de sensor'.

E1.2. RETORNA AO PASSO 2.

FIM EVENTO E1.

EVENTO E2: O usuário não seleciona o Modelo de Sensor.

ALTERNATIVA AO PASSO: 2A.2, 2B.1A.2, 2B.1B.2.

E2.1. O sistema exibe a mensagem 'É preciso selecionar o modelo de sensor'.

E2.2. RETORNA AO PASSO 2A.2 OU 2B.1A.2 OU 2B.1B.2.

FIM EVENTO E2.

EVENTO E3: O usuário não preenche o campo Endereço.

ALTERNATIVA AO PASSO: 2B.2.

E3.1. O sistema exibe a mensagem 'É preciso fornecer o endereço'.

E3.2. O sistema exibe o campo endereço com fundo vermelho.

E3.3. RETORNA AO PASSO 2B.2.

FIM EVENTO E3.

EVENTO E4: O usuário preenche o campo Endereço semanticamente incorreto.

ALTERNATIVA AO PASSO: 2B.2.

E4.1. O sistema exibe a mensagem 'O endereço fornecido é inválido'.

E4.2. O sistema exibe o campo endereço com fundo vermelho.

E4.3. O sistema exibe o campo endereço com o valor fornecido.

E4.4. RETORNA AO PASSO 2B.2.

FIM EVENTO E4.

EVENTO E5: O usuário não preenche o campo Usuário..

ALTERNATIVA AO PASSO: 2A.3.

E5.1. O sistema exibe a mensagem 'O campo Usuário não pode ter valor vazio'.

E5.2. O sistema exibe o campo Usuário com fundo vermelho.

E5.3. O sistema exibe o campo Usuário com o valor fornecido.

E5.4. RETORNA AO PASSO 2A.3.

FIM EVENTO E5.

EVENTO E6: O usuário preenche o campo Usuário com valor não cadastrado.

ALTERNATIVA AO PASSO: 2A.3.

E6.1. O sistema exibe a mensagem 'Usuário e/ou senha não cadastrados'.

E6.2. O sistema exibe o campo Usuário com fundo vermelho.

E6.3. O sistema exibe o campo Usuário com o valor fornecido.

E6.4. RETORNA AO PASSO 2A.3.

FIM EVENTO E6.

Evento E7: O usuário não preenche o campo Senha.

ALTERNATIVA AO PASSO: 2A.4.

E7.1. O sistema exibe a mensagem 'O campo Senha não pode ter valor vazio'.

E7.2. O sistema exibe o campo Senha com fundo vermelho.

E7.3. RETORNA AO PASSO 2A.4.

FIM EVENTO E7.

Evento E8: O usuário preenche o campo Senha com valor não cadastrado.

ALTERNATIVA AO PASSO: 2A.4.

E8.1. O sistema exibe a mensagem 'Usuário e/ou senha não cadastrados'.

E8.2. O sistema exibe o campo Senha com fundo vermelho.

E8.3. RETORNA AO PASSO 2A.4.

FIM EVENTO E8.

EVENTO E9: O usuário clica em Cancelar.

ALTERNATIVA AO PASSO: 5.

E9.1. O sistema não salva o sensor.

E9.2. RETORNA AO PASSO 6.

FIM EVENTO E9.

Figura 17 – Segunda parte do caso de uso Cadastrar Sensor.

3.2. Criação da tabela de decisão

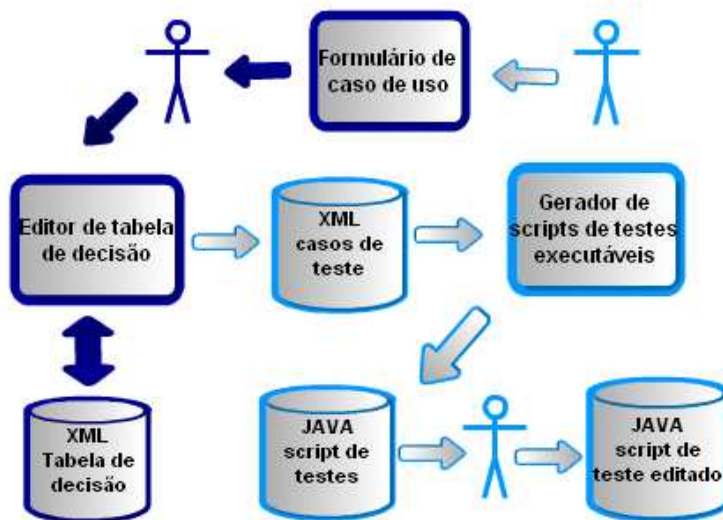


Figura 18 - Realce do processo de criação da tabela de decisão.

Após a redação do caso de uso, o próximo passo é a criação da tabela de decisão. A descrição do caso de uso servirá como entrada para a criação desta tabela.

As condições da tabela de decisão devem ser extraídas da descrição do caso de uso a partir da interpretação destes por um membro da equipe de testes. Daí a importância de um formulário de caso de uso com uma linguagem restrita e semi-estruturada a fim de evitar ambiguidades.

Cada condição pode representar um elemento na interface. Nesse caso, durante a edição da tabela de decisão, devem ser fornecidas informações adicionais para cada condição, como o tipo de campo ao qual a condição se refere (seção 4.1.1.1). Esta informação pode ser extraída a partir do padrão utilizado na descrição dos passos do caso de uso. Como queremos, ao final do processo, ter testes que sejam capazes de executar ações sobre a interface do sistema, também é necessário fornecer o identificador (seção 4.1.1.1) que será usado pela ferramenta que automatiza a interação com o navegador para que esta possa localizar o elemento referenciado. Os tipos de campos disponíveis são dependentes de quantos elementos de interface a ferramenta que automatiza o navegador suporta.

Outra informação que pode ser fornecida diz respeito aos relacionamentos entre as condições. Por exemplo, a existência de uma condição pode só ser justificada dado que outra condição seja verdadeira, ou, tomando outro exemplo, pode ocorrer de uma condição verdadeira implicar que outra condição é obrigatoriamente falsa. Se pudermos expressar essas restrições, quando existirem, então conseguiremos reduzir, em alguns casos drasticamente, a maior dificuldade no uso de tabelas de decisão: o potencialmente grande número de colunas.

Para cada combinação de valores das condições, que são as chamadas regras na tabela de decisão, também deve ser atribuída uma ou mais ações da tabela. Estas ações servirão como oráculos nos testes gerados, a fim de verificar se um dado teste foi bem sucedido.

Como parte deste trabalho, foi desenvolvida uma ferramenta para auxiliar na criação e edição da tabela de decisão. Esta ferramenta foi implementada tomando como base o trabalho realizado por (Lachtermacher, 2010).

O principal benefício do uso da tabela de decisão é a possibilidade de verificação de sua completeza, corretude e não ambiguidade. A ferramenta desenvolvida é capaz de executar essas verificações e alertar o usuário se a tabela de decisão não passar em alguma delas. Mais detalhes sobre a utilização desta ferramenta podem ser obtidos na seção 4.1. Como resultado desta etapa do processo, tem-se uma tabela de decisão completa e validada salva em formato XML.

3.3. Geração automática dos casos de teste

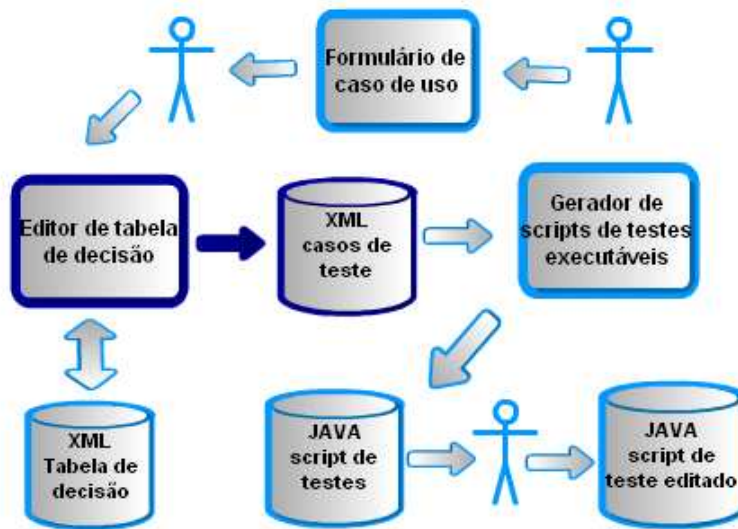


Figura 19 - Realce do processo de geração automática de casos de teste semânticos.

A geração dos casos de teste se dá automaticamente, através da ferramenta da seção anterior, a partir da tabela de decisão. Cada coluna da tabela de decisão dá origem a um caso de teste. Cada condição da tabela de decisão que implique em interação com um elemento da interface torna-se um comando a ser executado em cada caso de teste.

Como resultado desta etapa tem-se um arquivo XML contendo os casos de teste gerados. A estrutura deste arquivo é explicada mais à frente na seção 4.1.4.1.

3.4. Geração automática dos scripts de teste

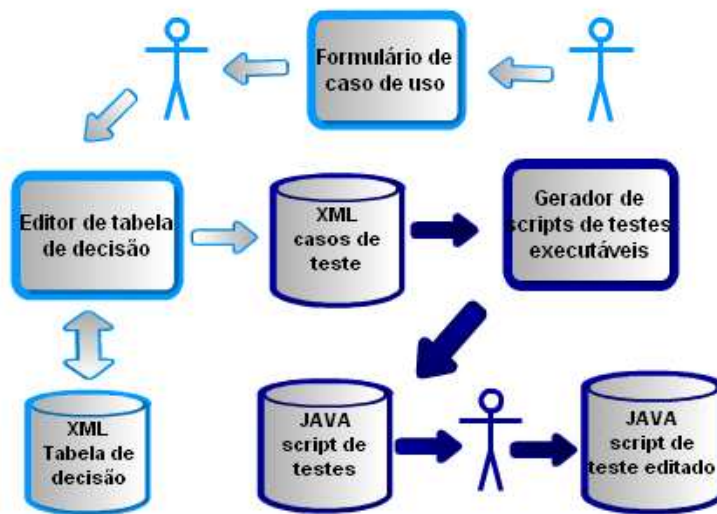


Figura 20 - Realce do processo de geração automática do script de teste.

Nesta etapa, a geração dos scripts de testes deve ser feita automaticamente a partir da conversão dos casos de teste em representação XML para scripts de testes executáveis. Para este fim, foi desenvolvida uma segunda ferramenta que recebe como entrada o XML descrevendo os casos de teste (gerado pela ferramenta de edição de tabelas de decisão), e gera como saída um script que automatizará os testes.

O script de teste gerado deve estar em conformidade com algum *framework* de automação de testes em alguma linguagem de programação. Nesta implementação o script é gerado em linguagem de programação Java e utiliza o framework JUnit (JUnit, 2009) para a automação dos testes.

O script gerado também deve ser capaz de automatizar a interação do usuário com a interface do sistema sob teste. Neste trabalho, em particular, estamos interessados em interfaces de sistemas web. Sendo assim, o módulo de Selenium (Selenium, 2009) para Java foi utilizado para realizar tal automação.

Após a geração do código de testes, este pode então ser editado para possíveis correções ou adições de passos e verificações. Maiores detalhes da implementação desta etapa neste trabalho podem ser obtidos na seção 4.2.