

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**PROTÓTIPO DE UM SISTEMA ESPECIALISTA PARA**  
**CONTROLE E ALERTA DE DESASTRES NATURAIS**

**MAYRA ZANCHETT MANCHEIN**

**BLUMENAU**  
**2014**

**2014/1-18**

**MAYRA ZANCHETT MANCHEIN**

**PROTÓTIPO DE UM SISTEMA ESPECIALISTA PARA  
CONTROLE E ALERTA DE DESASTRES NATURAIS**

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

Prof. Roberto Heinzle, Doutor - Orientador

**BLUMENAU  
2014**

**2014/1-18**

# **PROTÓTIPO DE UM SISTEMA ESPECIALISTA PARA CONTROLE E ALERTA DE DESASTRES NATURAIS**

Por

**MAYRA ZANCHETT MANCHEIN**

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

Presidente: \_\_\_\_\_  
Prof. Roberto Heinzle, Doutor – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Dalton Solano dos Reis, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Everaldo Artur Grahl, Mestre – FURB

Blumenau, 01 de julho de 2014.

Dedico este trabalho a todos os meus amigos e professores que me ajudaram diretamente na realização deste e a minha família que desde o início dos estudos sempre me apoiou.

## **AGRADECIMENTOS**

A Deus, por ter posto em meu caminho pessoas sábias e maravilhosas, que me ajudaram a alcançar meus ideais.

À minha família, que mesmo longe, nunca deixou de me apoiar.

Aos meus amigos, pelos bons momentos de trabalho e diversão.

Ao meu orientador, professor Roberto Heinzle, por ter acreditado na conclusão deste trabalho.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

Ao CEOPS por ter me cedido as informações para a construção da base de conhecimento, em especial ao Marcos Rodrigo Momo, que acompanhou o desenvolvimento deste trabalho, me dando ideias e me apoiando.

O futuro pertence àqueles que acreditam na  
beleza de seus sonhos.

Eleanor Roosevelt

## RESUMO

Este trabalho tem como objetivo desenvolver um Sistema Especialista (SE) que auxilie o usuário a ter um controle contra enchentes e escorregamentos na região da bacia hidrográfica do rio Itajaí-Açu. Em Blumenau, principalmente, a ocorrência deste tipo de desastre é comum e a população precisa ter um meio de estar sempre informada e saber como agir neste tipo de situação. O Centro de Operação do Sistema de Alerta da Bacia do Itajaí (CEOPS) é um exemplo disto, pois é um meio de informação muito útil quando algum desastre ocorre. Com o intuito de proporcionar uma alternativa a mais sobre tais questões, foi desenvolvido um sistema onde o usuário possa simular situações, funcionando como uma ferramenta para previsão de enchentes e escorregamentos. A base de conhecimento foi criada com informações disponibilizadas pelo CEOPS e trabalhos científicos existentes da área de estudo. Foi desenvolvido na linguagem Java integrado com a ferramenta *Java Embedded Object-Oriented Production Systems* (JEOPS), responsável pela construção da base de conhecimento e inferência das regras.

Palavras-chave: Sistemas especialistas. Desastres naturais. JEOPS. CEOPS.

## **ABSTRACT**

This work aims to develop an Expert system (ES) which assists the user to have a control against floods and landslides in the watershed of the Itajaí-Açu River region. In Blumenau, mainly, the occurrence of this kind of disaster is common and the people need to have means to always be informed and to know how to act in this kind of situation. The Operation Center of Alert System Itajaí Basin (CEOPS) is an example of it, because it is a mean of very useful information when some disaster occurs. In order to provide an extra alternative about these questions, it was developed a system where the user can simulate situations, working as a tool for forecasting floods and landslides. The knowledge base was created with information provided by CEOPS and existing scientific works of the study area. It was developed in the Java language integrated with the Java Embedded Object-Oriented Production Systems (JEOPS) tool, responsible for the construction of the knowledge base and the rules inference.

Key-words: Expert system. Natural disasters. JEOPS. CEOPS.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de ocupação na planície de inundação.....	16
Quadro 1 – Exemplo de regra de produção .....	22
Quadro 2 – Base de Conhecimento que contém definição de regras. ....	23
Figura 2 – Arquitetura do JEOPS .....	24
Quadro 3 – Cálculo NSE .....	29
Quadro 4 – Cálculo NRE.....	30
Quadro 5 – Regras de produção .....	30
Quadro 6 – Matriz de rastreabilidade .....	32
Figura 3 – Diagrama de caso de uso.....	33
Quadro 7 – Detalhamento do caso de uso Responder Questionário.....	33
Figura 4 – Diagrama de classes .....	35
Quadro 8 – Criação de uma base de conhecimento.....	37
Figura 5 – Compilando o arquivo .rules .....	37
Figura 6 – Setando o caminho do arquivo.....	38
Quadro 9 – Método: ActionOkquestionariol .....	39
Quadro 10 – Continuação do método ActionOkquestionariol.....	40
Quadro 11 – Métodos chamados ao final do questionário .....	40
Quadro 12 – Método: retornaTextoJTextAreaResultadoParcialEnchente...41	
Quadro 13 – Método: retornaRegrasEnchente .....	41
Quadro 14 – Método: obterResultadosAnterioresQuestl .....	41
Quadro 15 – Método: carregarTxtNoHashMapEscorregamento .....	42
Quadro 16 – Arquivo de regras .....	43
Quadro 17 – Método: Suscetibilidade.....	43
Quadro 18 – Método: valores.....	44
Figura 7 – Tela principal .....	44
Figura 8 – Menu administrador .....	45
Figura 9 – Tela de <i>login</i> .....	45
Figura 10 – Abrir arquivo.....	45
Figura 11 – Mostrando conteúdo do arquivo .....	46
Figura 12 – Criando nova base .....	46

Figura 13 – Salvar nova base.....	46
Figura 14 – Menu usuário.....	47
Figura 15 – Questionário enchente.....	47
Figura 16 – Informações do questionário enchente.....	48
Figura 17 – Resultado final questionário enchente.....	48
Figura 18 – Regras utilizadas.....	49
Figura 19 – Tabela de risco enchente.....	49
Figura 20 – Questionário escorregamento.....	50
Figura 21 – Resultado final questionário escorregamento.....	50
Figura 22 – Regras escorregamento.....	51
Figura 23 – Tabela de suscetibilidade escorregamento.....	51
Figura 24 – Informações escorregamento.....	52
Figura 25 – Resultados anteriores.....	52
Quadro 19 – Comparação entre trabalhos correlatos.....	53
Figura 26 – Questão um.....	54
Figura 27 – Questão dois.....	54
Figura 28 – Questão três.....	54
Figura 29 – Questão quatro.....	54
Figura 30 – Questão cinco.....	54
Figura 31 – Questão seis.....	55
Quadro 20 – Base de conhecimento enchente.....	61
Quadro 21 – Base de conhecimento escorregamento.....	78
Quadro 22 – Caso de uso Consultar Resultados Anteriores.....	100
Quadro 23 – Caso de uso Visualizar Regras.....	100
Quadro 24 – Caso de uso Alterar Regras na Base de Conhecimento.....	101
Quadro 25 – Caso de uso Visualizar Tabela de Risco.....	101
Quadro 26 – Caso de uso Abrir Base de conhecimento.....	102
Quadro 27 – Caso de uso Criar Nova Base de Conhecimento.....	102
Quadro 28 – Caso de uso Efetuar <i>login</i> .....	103
Quadro 29 – Caso de uso Consultar Informações.....	103

## **LISTA DE SIGLAS**

CEOPS – Centro de Operação do Sistema de Alerta da Bacia do Itajaí

EA – *Enterprise Architect*

IA – Inteligência Artificial

JEOPS – *Java Embedded Object-Oriented Production Systems*

JESS – *Java Expert System Shell*

RF – Requisito Funcional

RNF – Requisito Não-Funcional

SE – Sistema Especialista

SIG – Sistema de Informações Geográficas

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVOS DO TRABALHO .....	14
1.2 ESTRUTURA DO TRABALHO .....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
2.1 DESASTRES NATURAIS .....	15
2.1.1 Causas de desastres .....	15
2.1.2 Enchentes .....	16
2.1.3 Escorregamentos .....	17
2.1.4 A prevenção de desastres .....	17
2.1.5 CEOPS .....	18
2.2 SISTEMAS ESPECIALISTAS .....	19
2.2.1 Estrutura de um SE .....	20
2.2.1.1 Base de conhecimento .....	20
2.2.1.2 Motor de inferência.....	20
2.2.2 Aquisição do conhecimento .....	21
2.2.3 Representação do conhecimento .....	21
2.2.3.1 Regras de Produção .....	22
2.3 A FERRAMENTA JEOPS.....	23
2.3.1 Regras em JEOPS .....	23
2.3.2 Arquitetura interna .....	24
2.3.2.1 A base de conhecimentos.....	25
2.3.2.2 A base de objetos .....	25
2.3.2.3 A base interna de regras.....	26
2.3.2.4 A rede Rete .....	26
2.4 TRABALHOS CORRELATOS .....	27
2.4.1 Determinação de áreas de fragilidade utilizando um sistema especialista.....	27
2.4.2 Sistema especialista difuso para previsão de cheias em tempo atual.....	27
<b>3 DESENVOLVIMENTO DO PROTÓTIPO .....</b>	<b>29</b>
3.1 LEVANTAMENTO E AQUISIÇÃO DO CONHECIMENTO.....	29
3.2 REQUISITOS PRINCIPAIS DO PROTÓTIPO .....	30
3.3 ESPECIFICAÇÃO .....	31

3.3.1 Matriz de rastreabilidade.....	31
3.3.2 Diagrama de casos de uso .....	32
3.3.3 Diagrama de classes .....	34
3.4 IMPLEMENTAÇÃO .....	35
3.4.1 Técnicas e ferramentas utilizadas.....	36
3.4.1.1 Criação de regras JEOPS .....	36
3.4.2 Implementação da realização do questionário enchente .....	38
3.4.3 Implementação dos resultados anteriores do questionário enchente.....	41
3.4.4 Carregar arquivo de regras para visualização .....	41
3.4.5 Implementação dos cálculos para determinação do nível de risco .....	43
3.4.6 Operacionalidade da implementação .....	44
3.4.6.1 Tela principal .....	44
3.4.6.1.1 Administrador .....	45
3.4.6.1.2 Tela de <i>login</i> .....	45
3.4.6.1.3 Abrir base .....	45
3.4.6.1.4 Criando nova base .....	46
3.4.6.2 Usuário.....	47
3.4.6.2.1 Responder Questionário .....	47
3.4.6.2.2 Resultados anteriores .....	52
3.5 RESULTADOS E DISCUSSÃO .....	52
<b>4 CONCLUSÕES.....</b>	<b>56</b>
4.1 EXTENSÕES .....	56
<b>REFERÊNCIAS .....</b>	<b>58</b>
<b>APÊNDICE A – Regras de produção.....</b>	<b>61</b>
<b>APÊNDICE B – Detalhamento dos casos de uso .....</b>	<b>100</b>

## 1 INTRODUÇÃO

Os Sistemas Especialistas (SE) são programas inteligentes de computador que se utilizam de métodos inferenciais para a resolução de problemas técnicos e altamente especializados. Por utilizar-se da Inteligência Artificial (IA), um ramo da computação que estuda a capacidade das máquinas raciocinarem e aprenderem como os seres humanos, os SEs interagem com seu usuário em uma linguagem natural de perguntas e respostas, propondo e auxiliando na solução de problemas complexos (CRIPPA, 1999).

Tendo em vista que os SEs auxiliam na resolução de um determinado assunto fazendo a análise de situações, está sendo aplicado um SE que auxilie moradores em áreas de risco, próximas de rios, ribeirões ou encostas que são diretamente atingidos por desastres naturais como enchentes ou escorregamentos. No Brasil, os desastres naturais têm sido tratados de forma segmentada entre os diversos setores da sociedade. Nos últimos anos vem ocorrendo uma intensificação dos prejuízos causados por estes fenômenos devido ao mau planejamento urbano. Ações integradas entre comunidade e universidade são fundamentais para que os efeitos dos desastres naturais sejam minimizados (KOBİYAMA, 2006, p. 11).

Segundo Kobiyama (2006, p. 122), nas últimas décadas o número de registro de desastres naturais em várias partes do mundo vem aumentando consideravelmente. Isso se deve, principalmente, ao aumento da população, a ocupação desordenada e ao intenso processo de urbanização e industrialização.

Para Blumenau, a prevenção contra desastres naturais, principalmente as enchentes, é uma necessidade iminente. Assim sendo, este trabalho visa desenvolver um SE para identificar as possíveis situações que podem ocorrer, a fim de alertar sobre possíveis enchentes e escorregamentos, visando minimizar os impactos causados por estes desastres.

A base de conhecimento deste sistema foi construída através da coleta de dados disponibilizadas pelo Centro de Operação do Sistema de Alerta da Bacia do Itajaí (CEOPS), pelo sistema AlertaBlu da Prefeitura Municipal de Blumenau, nos trabalhos científicos existentes da área de estudo como o de Vieira (2007) que utilizou Sistemas de Informações Geográficas (SIG) para mapeamento das áreas suscetíveis a escorregamentos em Blumenau e de entrevistas com especialistas da área, bem como quando necessário complementadas com dados hipotéticos. A base de conhecimento é implementada em forma de regras, representadas por um par condição-ação, onde a ação só será executada se a condição for verdadeira. Estas regras, que formam a representação do conhecimento utilizado neste SE, são denominadas regras de produção (PEREIRA, 2000).

Foi utilizada a ferramenta *Java Embedded Object-Oriented Production Systems* (JEOPS) para a elaboração da base de conhecimento do SE, integrado a linguagem de programação Java onde a mesma foi utilizada para a implementação da interface gráfica.

### 1.1 OBJETIVOS DO TRABALHO

O objetivo principal deste trabalho é desenvolver um SE que auxilie o usuário a ter um controle contra enchentes e escorregamentos na região da bacia hidrográfica do rio Itajaí-Açu, alertando-o sobre possíveis desastres que venham a ocorrer.

Os objetivos específicos do trabalho são:

- a) formalizar o conhecimento através de SEs baseados em regras de produção;
- b) construir uma base de conhecimento de desastres naturais (enchentes e escorregamentos);
- c) disponibilizar um protótipo que aplique as regras definidas através da ferramenta JEOPS.

### 1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo é apresentada a introdução do trabalho onde está contido os objetivos do tema escolhido.

No segundo capítulo apresenta-se a fundamentação teórica do tema, onde são descritos os conceitos e informações sobre desastres naturais e SEs, bem como as tecnologias utilizadas no desenvolvimento do trabalho.

O terceiro capítulo contém o desenvolvimento do trabalho, onde tem-se os requisitos funcionais (RF) e não-funcionais (RNF) do protótipo, diagramas construídos, o sistema proposto com a implementação e telas do mesmo, e os resultados obtidos.

As conclusões e extensões do trabalho encontram-se no quarto capítulo.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos, técnicas e ferramentas abordados neste trabalho. Com foco em desastres naturais, como enchentes e escorregamentos, apresentando suas causas e prevenção. Também são abordados conceitos do SE e suas técnicas como a forma de representação do conhecimento. Assim como, a ferramenta utilizada na implementação e alguns trabalhos correlatos a esse.

### 2.1 DESASTRES NATURAIS

Inundações, escorregamentos, secas, furacões, entre outros, são fenômenos naturais severos, fortemente influenciados pelas características regionais, tais como, rocha, solo, topografia, vegetação e condições meteorológicas. Quando estes fenômenos intensos ocorrem em locais onde os seres humanos vivem, resultando em danos (materiais e humanos) e prejuízos (socioeconômico) são considerados como desastres naturais (KOBİYAMA, 2006, p. 19).

Pode-se definir desastres naturais como um evento natural atípico, que ocasiona danos diretos e indiretos a uma sociedade e que gera um grande número de vítimas (mortos, desalojados e desabrigados). Segundo Kobiyama (2006, p. 19), os desastres são normalmente súbitos e inesperados, de uma gravidade e magnitude capaz de produzir danos e prejuízos diversos, resultando em mortos e feridos. Portanto, exigem ações preventivas e restituidoras, que envolvem diversos setores governamentais e privados, visando uma recuperação que não pode ser alcançada por meio de procedimentos rotineiros.

Os desastres naturais constituem um tema cada vez mais presente no cotidiano das pessoas, independentemente destas residirem ou não em áreas de risco. Ainda que em um primeiro momento o termo leve a associá-lo com terremotos, tsunamis, erupções vulcânicas, ciclones e furacões, os desastres naturais contemplam, também, processos e fenômenos mais localizados tais como deslizamentos, inundações, subsidências e erosões, que podem ocorrer naturalmente ou induzidos pelo homem (TOMINAGA, 2009, p. 10).

#### 2.1.1 Causas de desastres

De modo geral, os desastres naturais são determinados a partir da relação entre o homem e a natureza. Em outras palavras, desastres naturais resultam das tentativas humanas em dominar a natureza, que, em sua maioria, acabam derrotadas. Assim, grande parte da história da humanidade foi influenciada pela ocorrência de desastres naturais, principalmente os de grande magnitude (KOBİYAMA, 2006, p. 13).



Sendo assim, estes desastres que tanto influenciam as atividades humanas vêm historicamente intensificando-se devido ao mau gerenciamento das bacias hidrográficas, especialmente pela falta de planejamento urbano. Além disso, o aquecimento global tem aumentado a frequência e a intensidade das adversidades climáticas, como precipitações extremas, vendavais, granizos entre outros, o que acarreta no aumento da incidência de desastres naturais (KOBİYAMA, 2006, p. 14).

Os desastres naturais podem ser provocados por diversos fenômenos, tais como, inundações, escorregamentos, erosão, terremotos, tornados, furacões, tempestades, estiagem, entre outros. Além da intensidade dos fenômenos naturais, o acelerado processo de urbanização verificado nas últimas décadas, em várias partes do mundo, inclusive no Brasil, levou ao crescimento das cidades, muitas vezes em áreas impróprias à ocupação, aumentando as situações de perigo e de risco a desastres naturais (TOMINAGA, 2009, p. 14).

Além disso, diversos estudos indicam que a variabilidade climática atual, com tendência para o aquecimento global, está associada a um aumento de extremos climáticos. Nesta situação, os eventos de temporais, de chuvas intensas, de tornados ou de estiagens severas, entre outros, podem tornar-se mais frequentes, aumentando a possibilidade de incidência de desastres naturais (TOMINAGA, 2009, p. 14).

### 2.1.2 Enchentes

Enchente, também conhecido como inundação, é o aumento do nível dos rios além da sua vazão normal, ocorrendo o transbordamento de suas águas sobre as áreas próximas a ele (KOBİYAMA, 2006, p. 45). As inundações têm como causa a precipitação anormal de água que, ao transbordar dos leitos dos rios, lagos, canais e áreas represadas, invade os terrenos adjacentes, provocando danos (CASTRO, 2003, p. 40). Atualmente, vêm aumentando gradativamente a frequência com que ocorrem as inundações e também os prejuízos que elas causam. Isto pode estar associado ao aumento do número de ocupações nas planícies de inundação (KOBİYAMA, 2006, p. 49). Na Figura 1 tem-se um exemplo de área sujeita a inundação.



Fonte: Kobiyama (2006, p.49).

Quando se constroem estradas, casas, prédios e outras edificações, ocorre um processo de impermeabilização do solo, isto é, acaba-se cobrindo o solo com cimento e asfalto. Impedindo desta forma que as águas das chuvas sejam absorvidas pelo solo. Neste caso, as águas escoam diretamente para os rios aumentando rapidamente seu nível. Os desmatamentos também aumentam o escoamento superficial e aceleram o processo de perda de solo, resultando no assoreamento dos cursos d'água (KOBİYAMA, 2006, p. 49).

### 2.1.3 Escorregamentos

Escorregamento (ou deslizamento) é o movimento coletivo de massa e/ou material sólido encosta abaixo, como solos, rochas e vegetação, sob a influência direta da gravidade (SELBY, 1993). Estes movimentos podem ocorrer principalmente com elevados volumes de precipitação e/ou terremotos. Tanto chuvas intensas de curta duração quanto de longa duração (chuvas contínuas) fornecem condições propícias para a diminuição da resistência do solo, atuando como um dos principais agentes deflagradores de movimentos de encostas em ambientes tropicais úmidos (GUIDICINI e IWASA, 1976).

Já, o aumento da população nas áreas urbanas pode agravar esta situação devido à ocupação de áreas inadequadas. Casseti (1991) cita que, a partir do momento em que o homem apropria-se de uma encosta, através dos desmatamentos, cortes e aterros, alterando sua estabilidade, aumenta-se a probabilidade de desencadear um escorregamento.

### 2.1.4 A prevenção de desastres

Evitar que fenômenos naturais severos ocorram foga da capacidade humana. Entretanto, através da prevenção, podem-se desenvolver medidas que minimizem os impactos causados pelos mesmos (KOBİYAMA, 2006, p. 42). Segundo Kobiyama et al. (2004), existem dois tipos de medidas preventivas básicas: as estruturais e as não-estruturais. As medidas estruturais envolvem obras de engenharia, como as realizadas para a contenção de cheias, tais como: barragens, diques, alargamento de rios, reflorestamento, etc. Contudo, tais obras são complexas e caras. As medidas não estruturais geralmente envolvem ações de planejamento e gerenciamento, como sistemas de alerta e zoneamento ambiental. Neste caso, dois aspectos devem ser considerados:

- a) a implantação da infraestrutura necessária às atividades humanas deve ser orientada por um zoneamento ambiental que considere a possibilidade de riscos ambientais, o que, na prática, é representado por mapas de áreas de risco. As restrições de uso são dependentes do risco ao qual está submetida uma área. Por

exemplo, em algumas áreas de alto risco são permitidas apenas as ocupações para fins comunitários (parques, praças etc.);

- b) no caso da existência de atividades humanas já implantadas em áreas suscetíveis a desastres (centros urbanos onde ocorrem inundações, edificações construídas em encostas íngremes, etc.), a criação de um sistema de alerta nestas áreas pode auxiliar na redução dos danos e prejuízos. Para tanto, os principais fatores causadores dos desastres devem ser monitorados continuamente e, paralelamente, os dados devem alimentar um modelo capaz de simular os fenômenos em tempo real. Assim, no momento em que o sistema identifica a aproximação de uma condição crítica, inicia-se o processo de alerta e retirada da população do local de risco.

Sistemas de previsão e alerta são efetivos quando associados a outras medidas, como a divulgação de mapas de inundação, com a exposição de regiões a serem alagadas. A educação do público potencialmente afetado, para interpretação das informações divulgadas e providências pessoais, é fundamental (PEDROLLO, 2000).

Para Kobiyama (2006), um sistema de alerta implantado em bacia hidrográfica, deve possuir os seguintes componentes: monitoramento, transmissão dos dados, modelagem e simulação, orientação para as instituições responsáveis e alerta para a população localizada nas áreas de risco. A aquisição dos dados em tempo real é cada vez mais importante para o sistema de alerta.

#### 2.1.5 CEOPS

O CEOPS é um centro de estudo de ações ligado a Universidade Regional de Blumenau (FURB) que surgiu para auxiliar, criar medidas não-estruturais, planos de ações e estratégias a fim de minimizar o impacto das cheias nas cidades da bacia do Rio Itajaí-Açu (CEOPS, 2014). Seguido ao evento de 1983, que atingiu a cota de 15 m e 34 cm, a FURB criou o Projeto Crise, tendo objetivo de desenvolver e implantar um sistema de alerta da bacia do rio Itajaí-Açu, com algumas horas de antecedência, bem como definir procedimentos operacionais às obras hidráulicas existentes. O Crise consistia em um conjugado de subprojetos que envolvia as áreas de Meteorologia, Hidrologia, Cartografia e Pesquisa Operacional, objetivando amenizar os impactos causados pelas cheias no Vale do Itajaí (CEOPS, 2014).

Em virtude dos desastres ocasionados pelas grandes cheias de 1983 e 1984, passaram a desenvolver ações frente à problemática das enchentes, visando pesquisar e propor medidas

não estruturais para sanar os impactos (CEOPS, 2014). Surgiu assim o CEOPS, que veio sendo gerenciado pelo então Projeto Crise, coordenado pela FURB. O Crise almejava medidas não estruturais, as quais se caracterizam por todos os tipos de intervenções que poderiam ser tomadas de maneira a proporcionar um convívio com as enchentes, reduzindo o impacto e suas consequências. O CEOPS seria a contrapartida, atuando com as medidas não-estruturais e os planos de ação e estratégias (CEOPS, 2014).

## 2.2 SISTEMAS ESPECIALISTAS

Os SEs são sistemas computacionais projetados e desenvolvidos para solucionarem problemas que normalmente exigem especialistas humanos com conhecimento na área de domínio da aplicação. Tal como um especialista, o sistema deve ser capaz de emitir decisões justificadas acerca de um determinado assunto a partir de uma substancial base de conhecimentos. Para tomar uma decisão, o especialista busca, em sua memória, conhecimentos prévios, formula hipóteses, verifica os fatos que encontra e compara-os com as informações já conhecidas e então emite a decisão. Neste processo o especialista realimenta a sua base de conhecimentos acerca do assunto (HEINZLE, 1995, p. 7).

Segundo Genaro (1987, p. 7), os SEs são programas intensivamente baseados em conhecimento. Estes sistemas podem resolver problemas que normalmente requerem experiência humana. Os SEs conseguem efetuar muitas das funções secundárias que os peritos executam, como perguntar questões relevantes e explicar suas razões.

Para Rabuske (1995, p. 72), SEs devem, também, ter habilidade para aprender com a experiência e explicar o que estão fazendo e porque o fazem. Esta última é uma das principais características que distinguem estes sistemas dos tradicionais sistemas de informações. Figueiredo (2011) comenta que para um SE entrar em funcionamento, o mesmo deve ser mantido com informações específicas e orientado a executar determinadas tarefas para a qual se requer o seu uso, como também devem ser construídas bases com o conhecimento de um especialista da área ao qual o problema se relaciona, programando as variáveis usadas e posteriormente simuladas com base no conhecimento do especialista, de tal forma que ao final o SE seja capaz de oferecer sugestões e conselhos aos usuários e, também, adquirir novos conhecimentos e heurísticas com essa interação (PY, 2006). Portanto, o estudo dos problemas que o SE terá de tratar torna-se essencial.

### 2.2.1 Estrutura de um SE

Baseada em uma estrutura modular, composta pelos módulos: engenharia do conhecimento, base de conhecimentos, mecanismo de inferência e memória de trabalho, a arquitetura dos SEs muitas vezes provê reaproveitamento dos módulos entre diferentes soluções (LUGER, 1998). Existem várias arquiteturas de SEs sendo usadas. Dentre elas a mais simples de compreender e a mais difundida apresenta uma composição que pode ser dividida em: engenharia do conhecimento, base de conhecimento, mecanismo ou máquina de inferência e memória de trabalho ou quadro negro (MANCHINI, 2005).

#### 2.2.1.1 Base de conhecimento

A base de conhecimento é um elemento permanente, mas específico de um SE. Contém conhecimento, sob a forma de regras de produção, quadros, redes semânticas, ou outro formalismo qualquer. Poder-se-ia dizer que a base de conhecimentos contém um somatório de fatos, de heurísticas e de crenças. Ela é criativa, capaz de certos tipos de controle sobre si mesma, podendo até suprir algumas informações ausentes. É especialmente esta última característica que a distingue das tradicionais bases de dados (SAVARIS, 2002).

Segundo estudo realizado por Nikolopoulos (1997), a base de conhecimento, como o nome já sugere, é onde são armazenados, na forma de regras, geralmente do tipo *if-then*, os conhecimentos adquiridos no módulo de aquisição de conhecimento. Na base de conhecimento encontra-se o conhecimento do especialista, dividido entre fatos e regras, modelado de acordo com o domínio de representação computacional (FLORES, 2003; LIA, 1999). Já Feigenbaum (1983), registra que a potência de um SE deriva do conhecimento que ele possui e não dos formalismos e esquemas específicos que ele emprega.

#### 2.2.1.2 Motor de inferência

É a parte do SE responsável pelas deduções sobre a base de conhecimentos, examinando o conteúdo da base e decidindo a ordem das inferências (FLORES, 2003; LIA, 1999). Máquina de inferência ou motor de inferência é um elemento permanente, que pode inclusive ser reutilizado por vários SEs. É a parte responsável pela busca das regras da base de conhecimento para serem avaliadas, direcionando o processo de inferência. O conhecimento deve estar preparado para uma boa interpretação e os objetos devem estar em uma determinada ordem, representados por uma árvore de contexto (SAVARIS, 2002).

Basicamente a máquina de inferência é dividida em tarefas que são: selecionar, buscar, avaliar e procurar. Resumindo as tarefas acima, podemos dizer que, as regras necessárias para se chegar a uma meta devem ser buscadas na base de conhecimento (SAVARIS, 2002).

O motor de inferência implementa os algoritmos que decidirão quais as regras que serão satisfeitas pelos fatos ou objetos. Posteriormente, prioriza essas regras e executa aquela que obtiver maior prioridade (GIARRATANO, 1998). Em geral, sua implementação é independente do domínio da aplicação, ou seja, dependerá principalmente da forma como o conhecimento foi representado na base de conhecimento (NIKOLOPOULOS, 1997).

### 2.2.2 Aquisição do conhecimento

É um módulo que permite ao sistema ampliar e alterar seu conhecimento. Aparece, geralmente, munido de recursos para trabalhar o conhecimento (editores, ordenadores, classificadores, etc.) que auxiliam na difícil tarefa de extraí-lo e aproveitá-lo adequadamente. Em muitos sistemas é a única forma de aprendizado (SAVARIS, 2002). Parte mais sensível e crítica no desenvolvimento de um SE (NOGUEIRA e SILVA, 1997; BITTENCOURT, 2006), a aquisição do conhecimento não se limita apenas a adicionar novos elementos à base de conhecimentos, mesclando os novos conhecimentos, com os conhecimentos já armazenados na base (BITTENCOURT, 2008).

A expressão aquisição do conhecimento corresponde a todos os mecanismos utilizados para adquirir o conhecimento utilizado na modelagem de um domínio de conhecimento. Diversas fontes podem ser utilizadas na aquisição do conhecimento, algumas delas são especialistas humanos, livros e exemplos. Em outra definição, a aquisição de conhecimento é mencionada como o processo que visa obter a informação necessária para construção de uma base de conhecimento (HENRIQUE; NEUMANN; CÁPUA, 2003).

### 2.2.3 Representação do conhecimento

A representação do conhecimento tem importância fundamental para um SE. Para armazenar o conhecimento do sistema e posteriormente recuperá-lo, raciocinando com base nele e, adquirindo novos conhecimentos, é necessária a utilização de um método de representação de conhecimento (SAVARIS, 2002).

Existem várias formas de se modelar o conhecimento de um especialista, tais como, lógica matemática, regras de produção, raciocínio baseado em casos, redes probabilísticas, entre outros (FLORES, 2003; DIEHL, 2000). As mais utilizadas, segundo Nogueira e Silva (1997), são as regras de produção (ou regras de realização), que utilizam a teoria das

probabilidades, garantindo assim uma maior legibilidade da base de conhecimentos. SEs que usam regras para representar o conhecimento são chamados SEs baseados em regras (NIKOLOPOULOS, 1997).

### 2.2.3.1 Regras de Produção

De acordo com o Laboratório de Inteligência Artificial da Universidade Federal do Ceará (1999), a arquitetura mais comum de um sistema especialista é a que envolve regras de produção que são simplesmente um conjunto de condições no estilo SE... ENTÃO..., com a possibilidade de inclusão de conectivos lógicos relacionando os atributos no escopo do conhecimento e o uso de probabilidades.

Descrevem o conhecimento na forma de regras formadas por condições e ações antecedentes ligadas por conectivos lógicos e conclusões. Ações e condições são agrupadas, ligados pelo conectivo lógico E, associando-se a conclusões da parte ENTÃO da regra. Eventualmente, podem aparecer negações das ações como condição de uma regra. Quando estas premissas são comprovadas, as conclusões são deduzidas como verdadeiras. As regras são disparadas de acordo com o conjunto de fatos, inicialmente fornecido pelo usuário e posteriormente ampliado por deduções feitas pelo sistema (ABEL, 1998). No Quadro 1 é demonstrado um exemplo de regras em uma situação de possível ocorrência de enchente.

Quadro 1 – Exemplo de regra de produção

```

regra26
  Se estado = alerta
  E acumulado1h >= 50
  Entao valor1h = 4

regra29
  Se estado = alerta
  E acumulado24h >= 100 e acumulado24h < 150
  Entao valor24h = 4

regra33
  Se estado = alerta
  E previsao1h >= 80
  Entao valorPrevisao = 4

```

Fonte: adaptado de Vieira (2007).

Resumindo, uma regra é um par ordenado de símbolos que representa algum conhecimento sobre um determinado assunto (DIEHL, 2000). Uma das grandes vantagens da utilização de regras é a facilidade na demonstração de como se chega à solução através do rastreamento das regras pela máquina de inferência (NOGUEIRA e SILVA, 1997).

Segundo Rabuske (1995), um sistema de produção pode ser formado por uma ou mais bases de regras, separadas de acordo com as conveniências de processamento. O sistema de produção é complementado por uma estratégia de controle que estabelece a prioridade em que

as regras são aplicadas, bem como o critério de desempate quando há mais regras candidatas a aplicação.

### 2.3 A FERRAMENTA JEOPS

O JEOPS surgiu durante a escolha de projetos para uma disciplina de graduação, IA Simbólica, no segundo semestre de 1997 na Universidade Federal de Pernambuco. Então se escolheu desenvolver um motor de inferência para uma linguagem convencional. Java foi a escolhida por motivos de que não havia nenhum mecanismo de inferência integrado à linguagem. Durante a disciplina foi desenvolvido então aquilo que viria a ser o primeiro protótipo de JEOPS (FIGUEIRA, 2000). Mais tarde, aproveitando o fato de que muito já havia sido desenvolvido em JEOPS, o autor decidiu continuar o trabalho mesmo após a conclusão de seu curso de graduação.

#### 2.3.1 Regras em JEOPS

Na criação de regras, prima-se pela separação explícita de todos os campos de uma regra: declarações de variáveis, condições e ações. Regras JEOPS são organizadas dentro de base de regras, tal como métodos em Java são organizados dentro de classes. Uma base de regra contém regras, mas também pode conter métodos e declarações de variáveis, como qualquer classe de Java. As regras JEOPS utilizam objetos. Estes objetos devem, portanto, já estar definidos quando elas forem ser pré-compiladas (FIGUEIRA, 2000, p. 53). O Quadro 2 ilustra um trecho de como são criadas regras na ferramenta.

Quadro 2 – Base de Conhecimento que contém definição de regras.

```
ruleBase Familia {
  rule encontraAncestrais {
    declarations
      Pessoa p;
      Objetivo o;
    localdecl
      Pessoa pai = p.getPai();
      Pessoa mae = p.getMae();
    conditions
      p == o.getAlvo(); o.estaAtivo();
    actions
      o.desativa();
      System.out.println(pai.getNome() + " e " +
        mae.getNome() + " são
ancestrais");
      insert(new Objetivo(pai));
      insert(new Objetivo(mae));
  }
}
```

Fonte: adaptado de Figueira (2000).

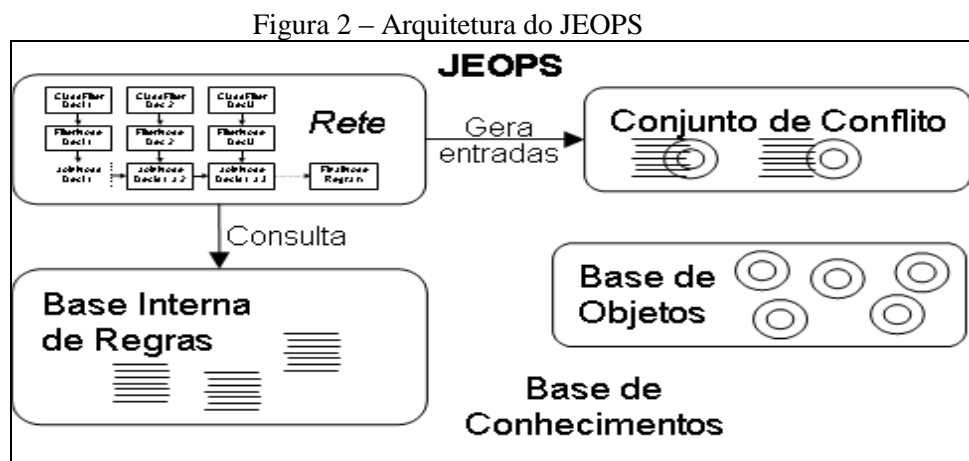


A base de regras do Quadro 2 contém apenas uma regra, *encontraAncestrais*. O campo das declarações contém as variáveis que serão unificadas com os objetos da memória de trabalho. No caso, a regra precisa de um objeto da classe *Pessoa* (p) e um da classe *Objetivo* (o) (FIGUEIRA, 2000).

As condições da regra podem ser qualquer expressão booleana de Java. Em geral, trata-se de comparações ou chamadas a métodos que retornam valores booleanos. Os métodos chamados nas condições de uma regra não devem alterar o estado de seu objeto (FIGUEIRA, 2000, p. 54). As ações da regra podem ser qualquer expressão de Java, além da manipulação dos objetos da base de fatos. Tudo o que pode aparecer como um corpo de método Java pode aparecer neste campo. Além disso, através dos comandos *insert* e *retract*, objetos podem ser inseridos e removidos da memória de trabalho da base, respectivamente. Modificações nos objetos devem ser informadas ao sistema pelo usuário, através da chamada ao método *modified* (FIGUEIRA, 2000, p. 55).

### 2.3.2 Arquitetura interna

A arquitetura de JEOPS segue a linha dos sistemas de produção tradicionais. Existe a separação clara entre a memória de trabalho e a base (interna) de regras. Quando são inseridos na memória de trabalho, os objetos que fazem uma regra se tornar disparável são adicionados junto com a referida regra ao conjunto de conflito. Quando a base de conhecimentos é ativada, ela solicita continuamente um elemento (par <regra, objetos>) do conjunto de conflitos, e executa a regra com os objetos correspondentes. O usuário, no entanto, não precisa ter ciência desta arquitetura, podendo trabalhar apenas com a base de conhecimentos. A Figura 2 ilustra, num alto nível, a arquitetura de JEOPS (FIGUEIRA, 2000, p. 62).



Fonte: adaptado de Figueira (2000).

### 2.3.2.1 A base de conhecimentos

A base de conhecimentos é a fachada de JEOPS que é vista pelo programador. É a base de conhecimentos quem encapsula os demais componentes, de modo a fornecer ao usuário uma porta de entrada única para o motor de inferência (FIGUEIRA, 2000, p. 63). Segundo Figueira (2000), bases de conhecimento JEOPS são subclasses da classe abstrata `jeops.AbstractKnowledgeBase`. O usuário precisa acessar apenas os métodos principais definidos nesta classe para poder utilizá-la. São métodos de inserção e remoção de objetos na memória de trabalho, *reset* e ativação da base de conhecimentos, sendo os seguintes:

- a) `insert(Object)`: insere um objeto na memória de trabalho da base de conhecimentos. Se este objeto puder ser unificado com uma declaração de uma das regras da base, o par <regra, objeto> será inserido imediatamente no conjunto de conflitos;
- b) `retract(Object)`: remove um objeto da memória de trabalho da base de conhecimentos. Se existir algum elemento do conjunto de conflitos que utilize este objeto, ele será imediatamente removido;
- c) `flush()`: remove todos os fatos (objetos) da base de conhecimentos, limpa a história do conjunto de conflitos (se houver). É o *reset* da base;
- d) `run()`: ativa a base de conhecimentos, colocando o motor de inferência para funcionar. Quando da chamada deste método, a base de conhecimentos irá disparar as regras do conjunto de conflitos até que este esteja vazio.

### 2.3.2.2 A base de objetos

A base de objetos é o local onde são armazenadas as referências para os objetos que fazem parte da memória de trabalho da base de conhecimentos. A cada novo objeto que é inserido na base de conhecimentos, antes dele ser enviado à rede Rete, é feita uma consulta à base de objetos para saber se o objeto sendo inserido já está presente na memória de trabalho. Se for o caso, a inserção é desconsiderada (FIGUEIRA, 2000, p. 65).

É a base de objetos também quem trata a relação de herança entre as classes, de modo que se a base de conhecimentos solicita todos os objetos de uma determinada classe, a base de objetos retornará não somente suas instâncias diretas, mas também as indiretas (FIGUEIRA, 2000, p. 65).

### 2.3.2.3 A base interna de regras

A base interna de regras é o componente responsável por responder à base de conhecimentos se as condições de uma regra são verdadeiras, quantas e quais regras foram definidas no arquivo pelo usuário, quais as declarações de cada regra. É também a base interna de regras que é responsável por disparar uma regra, quando solicitada pela base de conhecimentos (FIGUEIRA, 2000, p. 65).

A base interna de regras é gerada a partir da pré-compilação do arquivo de regras. A classe gerada, chamada de `Jeops_RuleBase_<nome da base de regras>` é uma subclasse de uma classe mais geral de JEOPS (`jeops.AbstractRuleBase`), que define uma série de assinaturas de métodos que são implementados pela classe gerada. Estes métodos dão à base de conhecimentos toda a informação de que ela precisa para realizar a unificação dos objetos da memória de trabalho com as declarações das regras (FIGUEIRA, 2000, p. 66).

### 2.3.2.4 A rede Rete

Rete é um algoritmo que apresenta duas vantagens em relação à maneira tradicional de casamento de padrões, na qual cada novo objeto inserido na base é testado com todas as demais combinações de objetos já presentes para se encontrar unificações com as condições das regras. A primeira deve-se ao fato de que, criando um grafo de propagação de novos fatos, as avaliações de uma condição que ocorram em mais de uma regra podem ser feitas apenas uma vez. Outra vantagem do algoritmo é que ele permite que as avaliações que já foram realizadas sejam armazenadas, de modo que não precisem ser refeitas sem necessidade (FIGUEIRA, 2000, p. 66). Segundo Figueira (2000), o JEOPS utiliza apenas uma das facetas de Rete, a da memorização de avaliações feitas anteriormente, deixando de lado a utilização da rede para eliminar avaliações redundantes.

A implementação de Rete do JEOPS não busca unificar condições repetidas entre as regras, tratando cada uma delas individualmente. Para cada regra, é então criada uma “mini-rede” por onde passarão os objetos que são inseridos na base de conhecimentos. Os objetos entram na rede pelos chamados nós de filtro de classe. Para cada declaração da regra é criado um nó de filtro de classe, representando a classe da variável declarada. Desta forma, sempre que um novo objeto é inserido na base de conhecimentos, ele é enviado para todos os nós de filtro de classe da rede. Aqueles que representarem classes do qual o objeto é uma instância propagarão este objeto para seus sucessores. Caso contrário, o objeto não será propagado para o restante da rede (FIGUEIRA, 2000, p. 78).

## 2.4 TRABALHOS CORRELATOS

Nesta seção são mostrados os trabalhos correlatos a este. Dentre os selecionados estão o trabalho de Leandro, Centeno e Krueger (2010) que apresenta um SE para determinação de áreas de fragilidade e o trabalho de Pedrollo e Lanna (2003), sobre um SE difuso para previsão de cheias em tempo atual.

### 2.4.1 Determinação de áreas de fragilidade utilizando um sistema especialista

A ferramenta proposta por Leandro, Centeno e Krueger (2010) tem como objetivo desenvolver um SE para diagnóstico de áreas de fragilidade de bacias hidrográficas utilizando o conhecimento de ciências geodésicas. O sistema permite qualificar e quantificar as transformações ocorridas em determinados ambientes através do monitoramento das variações espaciais da superfície da terra.

Para o desenvolvimento do SE foi utilizada a ferramenta *Java Expert System Shell* (JESS), gerando subsídios para a avaliação de riscos ambientais e permitindo a implantação de um plano de manejo para a unidade de conservação (LEANDRO; CENTENO; KRUEGER, 2010). Segundo Leandro, Centeno e Krueger (2010), para a realização dos primeiros experimentos foi escolhida uma bacia de pequeno porte na região metropolitana de Curitiba, que se localiza dentro de uma área de conservação e que seja afetada por ocupação humana e não possua qualquer tipo de plano de gestão ambiental. A determinação e monitoramentos dos limites da área de estudo são feitos utilizando fotografias aéreas e imagens de satélite.

Como conclusão os autores afirmam que, a agregação de conhecimentos de diversas áreas e a junção de todo um conjunto de informações vindas de varias fontes, como os levantamentos: *Global Positioning System* (GPS), o sensoriamento remoto e hidrologia, podem ser integrados, permitindo gerar um maior número de informações que sirvam de subsidio a outras ciências (LEANDRO; CENTENO; KRUEGER, 2010).

### 2.4.2 Sistema especialista difuso para previsão de cheias em tempo atual

O objetivo principal do trabalho de Pedrollo e Lanna (2003) é a pesquisa do uso de SEs para a previsão hidrológica, sob forma de categorias difusas, de níveis de enchentes fluviais. Propõem-se metodologias para as previsões de níveis de cheia, tanto na forma numérica tradicional quanto na forma de categorias. Utiliza-se um SE baseado em regras e inferências difusas (PEDROLLO; LANNA, 2003).

As metodologias desenvolvidas são aplicadas para previsão na bacia do rio Camaquã localizado no Rio Grande do Sul, para alcances de 1 e 2 dias de antecedência. Os efeitos da variação da densidade da rede são também analisados, verificando-se que sistemas de previsão hidrológica em tempo atual são possíveis. Previsões em tempo atual são previsões de curto prazo baseadas no princípio de que é possível obter-se, no presente, informações suficientes relacionadas às causas que condicionam um evento em um futuro próximo, para, com uso de modelos que relacionam as variáveis envolvidas, predizer, com precisão adequada, o estado de determinada variável que representa este evento (PEDROLLO; LANNA, 2003).

Como conclusão os autores afirmam que os principais recursos matemáticos concebidos para o sistema tiveram por objetivo o aprendizado.

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo são apresentadas as etapas de desenvolvimento do protótipo, onde estão dispostos os RFs e RNFs, o levantamento e aquisição do conhecimento, a especificação do protótipo, incluindo os diagramas de casos de uso e de classes, a implementação, onde são detalhadas as técnicas e ferramentas utilizadas e a operacionalidade do protótipo. Por fim, são apresentados os resultados obtidos e a discussão do mesmo.

#### 3.1 LEVANTAMENTO E AQUISIÇÃO DO CONHECIMENTO

Conforme objetivo definido neste trabalho, que foi desenvolver um sistema para auxiliar os usuários no alerta contra desastres naturais como enchentes e escorregamentos, o protótipo baseia-se no trabalho científico de Vieira (2007) que utilizou o SIG para mapeamento das áreas suscetíveis a escorregamentos em Blumenau e nas informações cedidas pelo CEOPS para apoiar a criação da base de conhecimento e suas regras. Primeiramente foram criadas duas bases de conhecimento:

- a) eventos de enchente;
- b) eventos de escorregamento.

Os fatores que levam estes desastres a ocorrer são provocados pelos elevados índices de precipitações na bacia hidrográfica do Vale do Itajaí, especificamente, na região do município de Blumenau, no caso de escorregamentos. Segundo Vieira (2007), considera-se ainda os seguintes fatores: declividade, tipo de vegetação, geotecnia, cobertura vegetal, densidade populacional e suscetibilidade a escorregamento.

Foram selecionados os fatores quantidade de chuva, tipo de vegetação, densidade populacional e geotecnia para realização do cálculo, atribuindo um peso específico a cada um, considerando os níveis de suscetibilidade: baixo (1), médio (2), alto (3) e muito alto (4).

O cálculo foi adaptado do trabalho de Vieira (2007), para melhor atender os objetivos do sistema. Determinou-se o nível de suscetibilidade a escorregamento (NSE), conforme o seu grau de importância. Considerou-se que o tema geotecnia (GEO), representada pela integração com a geomorfologia, geologia e risco decorrente da ocupação urbana, tem uma contribuição de 40% do nível de suscetibilidade total (VIEIRA, 2007). A quantidade de chuva (CHU), vegetação (VEG) e a densidade populacional (POP) contribuem com 20% cada. Deste modo, o NSE é determinado pela expressão apresentada no Quadro 3.

Quadro 3 – Cálculo NSE

$NSE = (0,4 \cdot GEO) + (0,2 \cdot CHU) + (0,2 \cdot VEG) + (0,2 \cdot POP)$
---

Fonte: adaptado de Vieira (2007).

Para os eventos de enchente seguiu-se a mesma linha, atribuindo valores para as variáveis: *acumulado1h*, *acumulado24h* e *previsao1h*. Considerando os pesos: baixo (1), médio (2), alto (3) e muito alto (4), usados para determinar o nível de risco de enchente (NRE), apresentado no Quadro 4.

Quadro 4 – Cálculo NRE

$$\text{NRE} = (0,4 \cdot \text{Acumulado1h}) + (0,3 \cdot \text{Acumulado24h}) + (0,3 \cdot \text{Previsao1h})$$

Fonte: adaptado de Vieira (2007).

Após identificados os fatores de cada desastre, foram identificadas as condições que levam a ocorrência de um evento, formando as regras. Criou-se uma base de conhecimento para enchentes representada em 96 regras de produção e uma base de conhecimento para escorregamentos com 109 regras. No Quadro 5 são apresentadas algumas das regras de escorregamentos. As regras para enchentes e escorregamentos podem ser vistas no Apêndice A.

Quadro 5 – Regras de produção

```

regra8
  Se vegetacao = capoeira
  E cobertura vegetal = alta
  Entao vegetacaoValor = 2

regra39
  Se declividade >= 0 e declividade < 25
  E geotecnia = cortes e aterros
  Entao geotecniaValor = 2

regra53
  Se populacao >= 71 e populacao <= 105
  Entao populacaoValor = 3

regra104
  Se acumulado1h >= 100 e acumulado1h < 210
  E acumulado24h >= 0 e acumulado24h < 50
  E acumulado168h >= 210
  E previsao1h >= 40 e previsao1h < 80
  Entao valorChuva = 3

```

Fonte: adaptado de Vieira (2007).

As regras apresentadas no Quadro 5 foram inseridas na ferramenta JEOPS, estando completo o SE. Tendo-se as bases definidas e implementadas, foram feitas as telas do sistema no ambiente Eclipse, utilizando os componentes da biblioteca *Swing* do Java, onde implementou-se as telas de entrada e o final, como resultado a previsão de cenário baseado nas informações apresentadas ao SE.

### 3.2 REQUISITOS PRINCIPAIS DO PROTÓTIPO

O protótipo deve:

- a) permitir que o usuário realize consultas dos resultados obtidos anteriormente (RF);
- b) permitir ao usuário escolher qual tipo de desastre realizar a consulta (RF);
- c) permitir ao administrador a alteração das regras na base de conhecimento de desastres naturais (RF);
- d) permitir ao administrador efetuar *login* no sistema (RF);
- e) permitir ao administrador criar uma nova base de conhecimento com extensão *.rules* (RF);
- f) permitir ao usuário visualizar um histórico das regras que foram aceitas depois de realizar o questionário (RF);
- g) permitir ao usuário visualizar informações sobre os questionários (RF);
- h) permitir ao usuário visualizar a tabela de valores usada para calcular os riscos de enchentes e escorregamentos (RF);
- i) permitir ao administrador abrir uma base de conhecimento de desastres naturais (RF);
- j) ser implementado utilizando a linguagem de programação Java (RNF);
- k) ser implementado utilizando o ambiente de desenvolvimento Eclipse (RNF);
- l) utilizar a ferramenta JEOPS para criação das regras (RNF);
- m) utilizar a ferramenta *Enterprise Architect* (EA) para especificação dos diagramas de casos de uso e de classes (RNF).

### 3.3 ESPECIFICAÇÃO

Nesta seção é apresentada a especificação do protótipo com sua matriz de rastreabilidade e detalhamento do caso de uso Responder Questionário, bem como apresentados os diagramas de casos de uso e classes. Todos foram especificados utilizando a ferramenta EA.

#### 3.3.1 Matriz de rastreabilidade

O Quadro 6 apresenta os RFs previstos para o sistema e sua rastreabilidade, ou seja, vinculação com os casos de uso associados.



Quadro 6 – Matriz de rastreabilidade

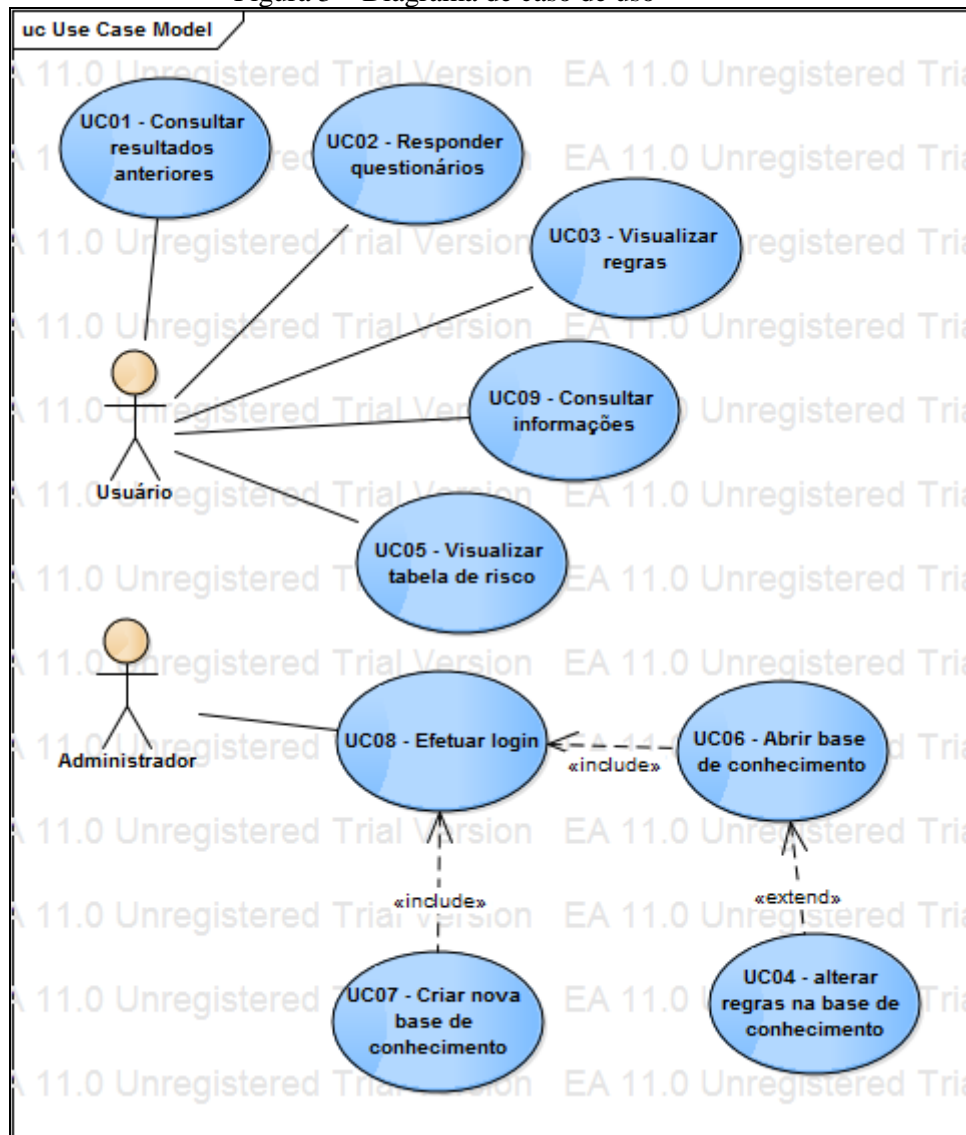
<b>Requisitos Funcionais</b>	<b>Casos de Uso</b>
RF01: O sistema deve permitir que o usuário realize consultas dos resultados obtidos anteriormente.	UC01
RF02: O sistema deve permitir ao usuário escolher qual tipo de desastre realizar a consulta.	UC02
RF03: O sistema deve permitir ao administrador a alteração das regras na base de conhecimento de desastres naturais.	UC04
RF04: O sistema deve permitir ao administrador efetuar <i>login</i> no sistema.	UC08
RF05: O sistema deve permitir ao administrador criar uma nova base de conhecimento com extensão <i>.rules</i> .	UC07
RF06: O sistema deve permitir ao usuário visualizar um histórico das regras que foram aceitas e as que não foram depois de realizar o questionário.	UC03
RF07: O sistema deve permitir ao usuário visualizar informações sobre os questionários.	UC09
RF08: O sistema deve permitir ao usuário visualizar a tabela de valores usada para calcular os riscos de enchentes e escorregamentos.	UC05
RF09: O sistema deve permitir ao administrador abrir uma base de conhecimento de desastres naturais.	UC06

### 3.3.2 Diagrama de casos de uso

Esta subseção apresenta o diagrama de caso de uso do sistema para melhor exemplificação das atividades exercidas por cada um dos atores caracterizados como, administrador e usuário. Na Figura 3 apresenta-se o diagrama de caso de uso especificando as ações que o administrador e o usuário podem realizar no sistema.

O administrador possui total acesso às funcionalidades do sistema, além de ser responsável por criar uma nova base de conhecimento e também adicionar, alterar ou excluir regras numa base já existente. O usuário pode realizar os questionários para enchente ou escorregamento, visualizar as regras que foram utilizadas durante a realização do questionário, consultar informações sobre os questionários e consultar resultados anteriores.

Figura 3 – Diagrama de caso de uso



No Quadro 7, tem-se o detalhamento do caso de uso Responder Questionário. Ele relaciona a função do ator do caso de uso, as pré-condições e pós-condições, além do fluxo necessário para a função de realizar um questionário. O detalhamento dos demais casos de uso podem ser visualizados no Apêndice B.

Quadro 7 – Detalhamento do caso de uso Responder Questionário

<p><b>UC02 – Responder Questionário</b></p> <p><b>Ator:</b> Usuário</p> <p><b>Objetivo:</b> Permite ao usuário realizar o questionário sobre enchentes ou escorregamentos.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Usuário clica no submenu Responder Questionário</li> <li>2. Usuário seleciona o tipo de questionário</li> <li>3. Usuário responde todas as questões</li> </ol>
--

4. Usuário clica no botão OK
5. Sistema mostra resultado final

**Cenário de Exceção:**

No passo 4, caso o usuário não tenha informado algum campo:

- 4.1 Sistema apresenta a mensagem “Selecione uma opção”
- 4.2 Volta ao cenário principal

**Cenário de Exceção:**

No passo 4, caso o usuário informe um número negativo:

- 4.1 Sistema apresenta a mensagem “Números negativos não são aceitos”
- 4.2 Volta ao cenário principal

**Cenário de Exceção:**

No passo 4, caso o usuário informe algum caracter ou letra nos campos:

- 4.1 Sistema apresenta a mensagem “Informe um valor numérico”
- 4.2 Volta ao cenário principal

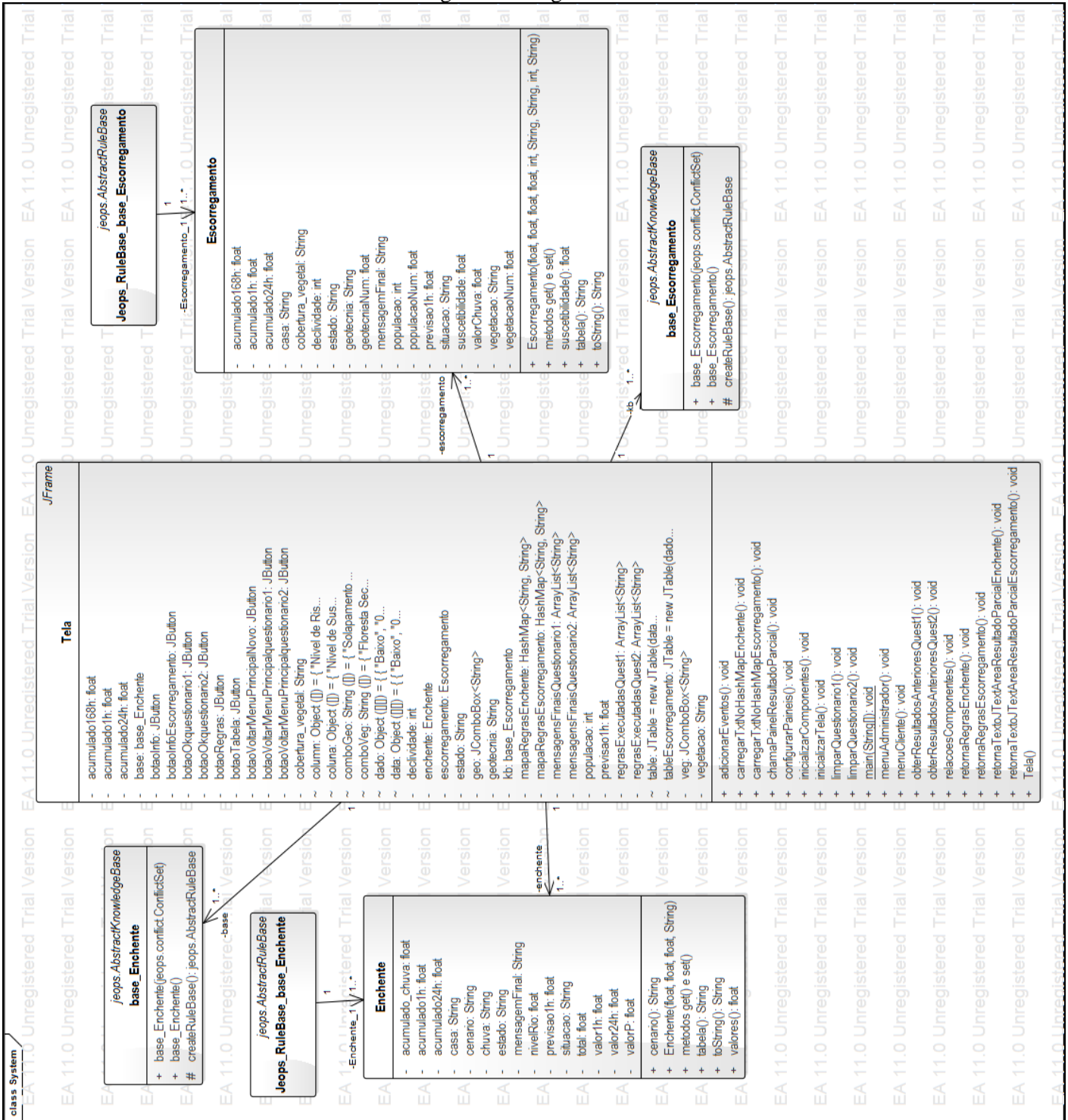
### 3.3.3 Diagrama de classes

A Figura 4 apresenta o diagrama de classes do protótipo. São apresentadas as principais classes do sistema, bem como seus relacionamentos e estrutura. Os atributos e métodos de algumas classes não são mostrados por serem muito grandes.

As classes `Jeops_Rule_base_Enchente` e `Jeops_Rules_base_Escorregamento` são classes que foram geradas pelo JEOPS a partir dos arquivos `base_Enchente.rules` e `base_Escorregamento.rules`, responsáveis pela criação da base de conhecimento e suas regras. Nessas classes são criados objetos das classes `Enchente` e `Escorregamento` para que possam acessar seus métodos que compõem as regras na base de conhecimento. A classe `Tela` é responsável por toda a estrutura do protótipo, como o relacionamento entre os componentes para criar a tela, ações dos botões e ativação das bases de conhecimento.

No método `ActionOkquestionario1`, quando clicado no botão OK do questionário enchente é feita a verificação de todos os campos, se foram corretamente preenchidos e passadas as variáveis como parâmetro para o objeto criado da classe `Enchente`. Após é inserido o objeto na base e ativada a mesma, chamando em seguida o método `chamaPainelResultadoParcial` que chama o painel que traz o resultado e o método `retornaTextoJTextaAreaResultadoParcialEnchente` que mostra o resultado da infêrencia das regras aceitas. O método `retornaRegrasEnchente` mostra as regras na tela. O questionário escorregamento segue o mesmo princípio.

Figura 4 – Diagrama de classes



### 3.4 IMPLEMENTAÇÃO

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

### 3.4.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento do protótipo foi utilizada a linguagem de programação Java versão 1.7 em conjunto com a ferramenta JEOPS versão 2.1, onde é realizada a construção da base de conhecimento. Como ambiente de desenvolvimento, foi utilizado o Eclipse Juno.

#### 3.4.1.1 Criação de regras JEOPS

A criação das regras se dá através da criação de um arquivo com extensão `.rules`. Dentro deste arquivo são construídas as regras de produção, onde se cria objetos da classe que se quer chamar. No Quadro 8 tem-se um exemplo de como as regras são escritas, podendo observar a sintaxe parecida com Java. No quadro está sendo criado um objeto da classe `Enchente` onde seus métodos tornam-se acessíveis para a escrita das regras.

Uma regra em JEOPS é construída da seguinte maneira:

- a) *rule*: define o nome da regra;
- b) *declarations*: criação de um objeto da classe a ser chamada;
- c) *conditions*: corpo de uma regra. Chamada de métodos que irão suportar as condições estabelecidas;
- d) *actions*: ações que serão executadas caso as condições estabelecidas sejam verdadeiras.

Quadro 8 – Criação de uma base de conhecimento

```

public ruleBase base_Enchente {

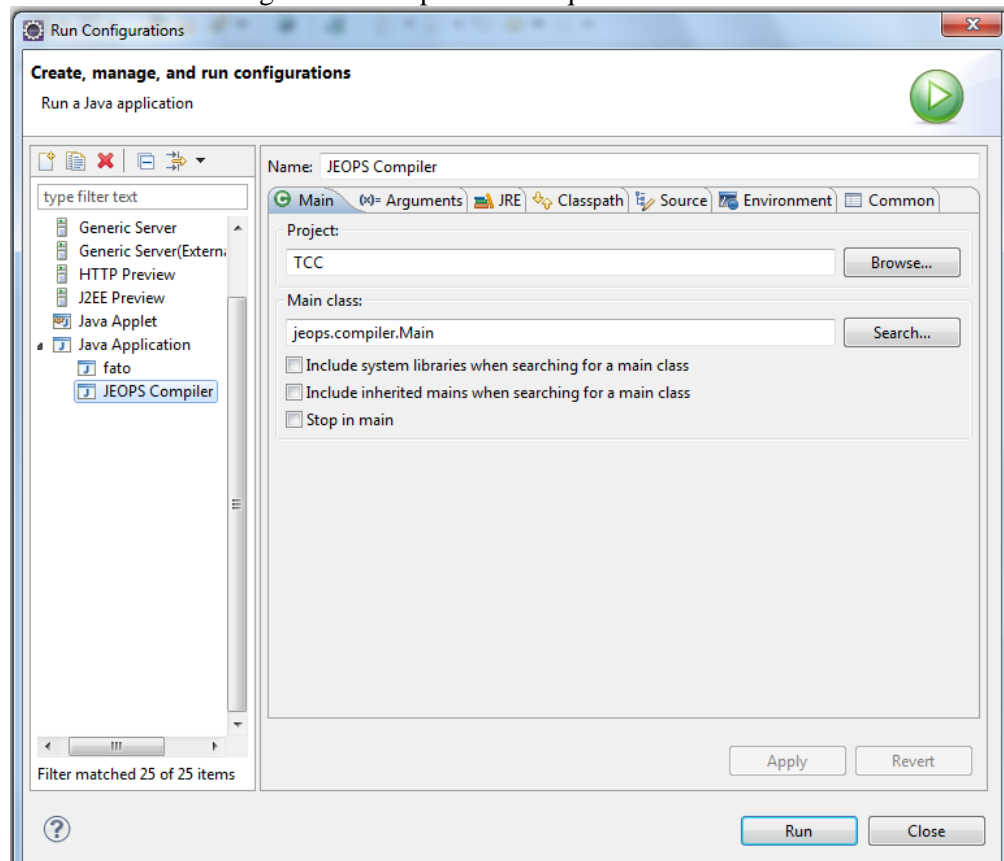
    rule regra1 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        actions
            e.setValor1h(1);
    }

    rule regra2 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        actions
            e.setValor1h(1);
    }

    rule regra3 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        actions
            e.setValor1h(2);
    }
}

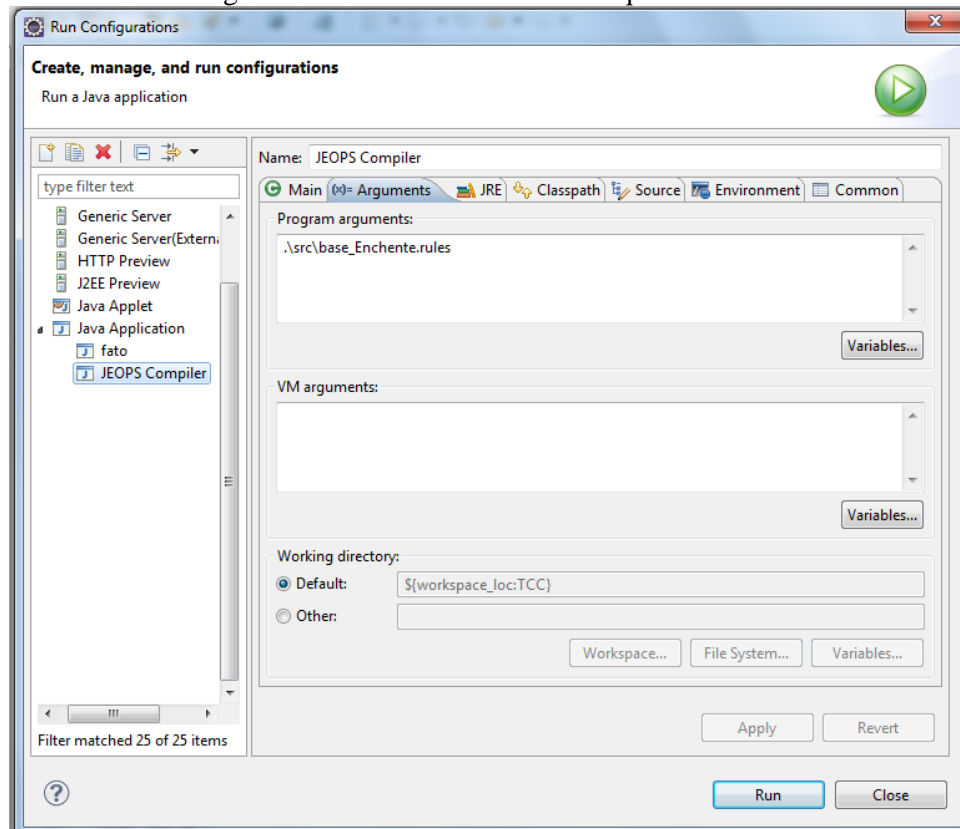
```

Após a criação do arquivo, precisa ser gerado um `.java` a partir dele. Para isto utiliza-se a opção *Run Configurations* do Eclipse, criando um novo *Java Application* chamado *JEOPS Compiler* com as configurações demonstradas na Figura 5.

Figura 5 – Compilando o arquivo `.rules`

Ainda na tela de *Run Configurations*, na aba *Arguments* deve-se setar o caminho em que se encontra o arquivo `.rules` e ao final clicar em *Run* para gerar o arquivo `.java` destas regras, como demonstra a Figura 6.

Figura 6 – Setando o caminho do arquivo



### 3.4.2 Implementação da realização do questionário enchente

No Quadro 9 é mostrado o código para a realização do questionário enchente quando clicado no botão OK.

Quadro 9 – Método: ActionOkquestionario1

```

866 private class ActionOkquestionario1 implements ActionListener {
867     public void actionPerformed(ActionEvent e) {
868         String mensagemErro = "";
869         regrasExecutadasQuest1.clear();
870         textoRegras.setText("");
871         try {
872             mensagemErro = "Resposta 1";
873             acumulado1h = Float
874                 .parseFloat(resposta1questionario1.getText());
875             mensagemErro = "Resposta 2";
876             acumulado24h = Float.parseFloat(resposta2questionario1
877                 .getText());
878             mensagemErro = "Resposta 3";
879             previsao1h = Float
880                 .parseFloat(resposta3questionario1.getText());
881
882             if (opcao1pergunta5questionario1.isSelected()) {
883                 estado = opcao1pergunta5questionario1.getText();
884             }
885             if (opcao2pergunta5questionario1.isSelected()) {
886                 estado = opcao2pergunta5questionario1.getText();
887             }
888             if (opcao3pergunta5questionario1.isSelected()) {
889                 estado = opcao3pergunta5questionario1.getText();
890             }
891             if (opcao4pergunta5questionario1.isSelected()) {
892                 estado = opcao4pergunta5questionario1.getText();
893             }
894         } catch (NumberFormatException nfe) {
895             JOptionPane.showMessageDialog(Tela.this, mensagemErro
896                 + " inválida! Digite um valor numérico!");
897         }
898         enchente = new Enchente(acumulado1h, acumulado24h, previsao1h,
899             estado);

```

Nas linhas 872 a 892 são verificados se tal campo está selecionado, passando seu conteúdo às respectivas variáveis. Depois de feita esta checagem as variáveis são passadas como parâmetro para o objeto `enchente` da classe `Enchente`. Então é criado um novo objeto da classe `base_Enchente` que contém as informações das regras na base de conhecimento. O objeto `base` é uma referência para os métodos da ferramenta JEOPS como é mostrado no Quadro 10.



Quadro 10 – Continuação do método ActionOkquestionario1

```

972         base = new base_Enchente();
973         base.addRuleFireListener(new RuleFireListener() {
974             public void ruleFiring(RuleEvent ex) {
975                 int i = ex.getRuleIndex();
976                 String name = ex.getKnowledgeBase().getRuleBase()
977                     .getRuleNames()[i];
978                 if (regrasExecutadasQuest1.contains(name)) {
979                     //
980                 } else {
981                     regrasExecutadasQuest1.add(name);
982                 }
983             }
984             public void ruleFired(RuleEvent ex) {
985             }
986         });
987         try {
988             if (acumulado1h == 0 || acumulado24h == 0 || previsao1h == 0
989                 || estado == null
990                 || buttonGroupResposta5Questionario1 == null
991                 || buttonGroupResposta6Questionario1 == null) {
992                 JOptionPane.showMessageDialog(Tela.this,
993                     "Todos os campos são obrigatórios.");
994             } else {
995                 if (acumulado1h < 0 || acumulado24h < 0 || previsao1h < 0) {
996                     JOptionPane.showMessageDialog(Tela.this,
997                         "Numeros negativos não são aceitos.");
998                 } else {
999                     base.insert(enchente);
1000                    base.run();

```

Após instanciar uma nova base na linha 972, é criado um novo evento que é disparado na execução das regras. Na linha 976 é passada para a variável `name` a regra que foi executada com seu respectivo nome. O *arraylist* `regrasExecutadasQuest1` guarda as regras que foram aceitas durante a execução. Entre as linhas 978 a 981 é feita uma checagem caso a regra que está sendo passada no momento já esteja contida na variável `regrasExecutadasQuest1` não faz nada, senão adiciona a regra ao *arraylist*.

Entre as linhas 988 a 997 está sendo feita a checagem dos campos, verificando se estão vazios ou negativos, impedindo que isto aconteça na hora da execução.

Nas linhas 999 e 1000 o objeto `enchente` é inserido na base e então a base é ativada pelo método `run`. Caso nenhum campo esteja vazio ele chama o método `chamaPainelResultadoParcial` que mostra um novo painel contendo o resultado final da execução, como mostra o Quadro 11.

Quadro 11 – Métodos chamados ao final do questionário

```

mensagensFinaisQuestionario1.add(enchente
    .getMensagemFinal());

JOptionPane.showMessageDialog(Tela.this, "OK");

chamaPainelResultadoParcial();

retornaTextoJTextAreaResultadoParcialEnchente();
retornaRegrasEnchente();

```

No método `retornaTextoJTextAreaResultadoParcialEnchente` ele adiciona ao `JTextArea` do painel a mensagem final obtida através dos valores selecionados no questionário, o cenário atual da situação e a tabela de valores utilizada para cálculo de risco de enchente, como mostra o Quadro 12.

Quadro 12 – Método: `retornaTextoJTextAreaResultadoParcialEnchente`

```
public void retornaTextoJTextAreaResultadoParcialEnchente() {
    textoResultadoParcial
        .setText(" ----- RESULTADO ----- \n\n");
    textoResultadoParcial.append(enchente.getMensagemFinal() + "\n");
    textoResultadoParcial.append(enchente.cenario() + "\n");
    textoResultadoParcial.append(enchente.tabela() + "\n");
}
```

No Quadro 13 tem-se o método `retornaRegrasEnchentes` que traz as regras que foram aceitas. Nele é feito um *foreach* que pega as regras da variável `regrasExecutadasQuest1` e adiciona ao *hashmap* `mapaRegrasEnchente`.

Quadro 13 – Método: `retornaRegrasEnchente`

```
public void retornaRegrasEnchente() {
    textoRegras.append("\nRegras Utilizadas:\n\n");
    for (String regra : regrasExecutadasQuest1) {
        textoRegras.append(mapaRegrasEnchente.get(regra));
    }
}
```

### 3.4.3 Implementação dos resultados anteriores do questionário enchente

No Quadro 14 tem-se o método `obterResultadosAnterioresQuest1`. Nele é checado se a variável `mensagensFinaisQuestionario1` está vazia, caso contrário é feito um *foreach* e para cada valor contido nele é adicionado ao `TextArea` `textoObterResultados`.

Quadro 14 – Método: `obterResultadosAnterioresQuest1`

```
public void obterResultadosAnterioresQuest1() {
    textoObterResultados
        .append("----- RESULTADOS QUESTIONARIO ENCHENTE ----- \n");

    if (mensagensFinaisQuestionario1.isEmpty()) {
        textoObterResultados
            .append("\nEste questionário ainda não foi realizado.\n");
    } else {
        for (String s : mensagensFinaisQuestionario1) {
            textoObterResultados.append("\n" + s + "\n");
        }
    }
}
```

### 3.4.4 Carregar arquivo de regras para visualização

Para visualizar as regras que foram executadas durante a realização do questionário é chamado o método `carregarTxtNoHashMapEscorregamento`. Este método carrega um arquivo `.txt` que contém todas as regras da base de conhecimento de escorregamento.

No Quadro 15 tem-se o código do método `carregarTxtNoHashMapEscorregamento`.

Quadro 15 – Método: `carregarTxtNoHashMapEscorregamento`

```

1635 public void carregarTxtNoHashMapEscorregamento() throws IOException {
1636
1637     mapaRegrasEscorregamento = new HashMap<String, String>();
1638
1639     try {
1640         InputStream in = this.getClass().getResourceAsStream(
1641             "regrasEscorregamento.txt");
1642         Reader reader = new InputStreamReader(in);
1643         BufferedReader leitor = new BufferedReader(reader);
1644         String line = "", arquivoCopiaEscorregamento = "";
1645
1646         while ((line = leitor.readLine()) != null) {
1647             arquivoCopiaEscorregamento += line.trim() + "\n";
1648         }
1649
1650         arquivoCopiaEscorregamento.replaceAll(" ", "");
1651
1652         String array[] = arquivoCopiaEscorregamento.split("//");
1653
1654         for (int i = 1; i <= array.length; i++) {
1655             mapaRegrasEscorregamento.put("regra" + i, array[i - 1]);
1656         }
1657
1658         keys = mapaRegrasEscorregamento.keySet();
1659
1660         leitor.close();
1661
1662     } catch (IOException e) {
1663         System.out.println("ERRO");
1664     }
1665
1666 }

```

Na linha 1637 é criado um *HashMap* que irá guardar as regras. Entre as linhas 1640 a 1643 é buscado o caminho em que se encontra o arquivo `.txt` e jogado em um *BufferedReader*. Então é verificado cada linha do arquivo e jogado para dentro da variável `arquivoCopiaEscorregamento`.

Na linha 1652 é criado um *array* de *String* onde é passado o arquivo `.txt` contido dentro da variável `arquivoCopiaEscorregamento` separando cada regra quando for achado o símbolo `//`. Nas linhas 1654 e 1656 é feito um *for* para varrer cada regra e passar para o *HashMap* `mapaRegrasEscorregamento`.

No Quadro 16 tem-se um exemplo de um arquivo de regras que é passado no método `carregarTxtNoHashMapEscorregamento`.

Quadro 16 – Arquivo de regras

```

regra55
  Se acumulado1h >= 0 e acumulado1h < 10
    E acumulado24h >= 0 e acumulado24h < 50
    E acumulado168h >= 0 e acumulado168h < 100
  Entao valorChuva = 1
//

regra56
  Se acumulado1h >= 10 e acumulado1h < 30
    E acumulado24h >= 50 e acumulado24h < 100
    E acumulado168h >= 100 e acumulado168h < 150
    E previsao1h >= 0 e previsao1h < 40
  Entao valorChuva = 2
//

regra57
  Se acumulado1h >= 30 e acumulado1h < 50
    E acumulado24h >= 100 e acumulado24h < 150
    E acumulado168h >= 150 e acumulado168h < 210
    E previsao1h >= 40 e previsao1h < 80
  Entao valorChuva = 3
//

```

### 3.4.5 Implementação dos cálculos para determinação do nível de risco

Nos Quadros 17 e 18, tem-se a implementação dos métodos `suscetibilidade` e `valores` respectivamente. Os cálculos seguem o modelo de Vieira (2007) para determinar o nível de risco para enchentes e escorregamentos.

No método `suscetibilidade` é feito o cálculo dos valores recebidos durante a realização do questionário escorregamento, para então fazer a verificação do valor que foi alcançado e determinar o nível de risco, como demonstra o Quadro 17.

Quadro 17 – Método: Suscetibilidade

```

public float suscetibilidade() {
    suscetibilidade = ((0.2f * vegetacaoNum) + (0.2f * populacaoNum)
        + (0.4f * geotecniaNum) + (0.2f * valorChuva));

    if (suscetibilidade >= 0.0 && suscetibilidade <= 1.59) {
        setEstad("Baixo");
        setSituacao("Fora de Perigo");
        setMensagemFinal("A probabilidade de ocorrer um escorregamento em
    }
    if (suscetibilidade >= 1.60 && suscetibilidade <= 2.19) {
        setEstad("Medio");
        setSituacao("Ficar alerta");
        setMensagemFinal("A probabilidade de ocorrer um escorregamento em
    }
    if (suscetibilidade >= 2.20 && suscetibilidade < 2.79) {
        setEstad("Alto");
        setSituacao("Alerta maximo");
        setMensagemFinal("A probabilidade de ocorrer um escorregamento em
    }
    if (suscetibilidade >= 2.79) {
        setEstad("Muito Alto");
        setSituacao("procurar abrigo");
        setMensagemFinal("A probabilidade de ocorrer um escorregamento em
    }
    return suscetibilidade;
}

```

No Quadro 18 tem-se o método `valores`, onde é realizado o cálculo das variáveis utilizadas na realização do questionário enchente. Depois de calculado o valor, é feita a verificação dele para determinar em qual nível de risco se encaixa.

Quadro 18 – Método: `valores`

```
public float valores() {
    total = (0.4f * valor1h) + (0.3f * valor24h) + (0.3f * valorP);
    if (total >= 0 && total < 1.50) {
        setEstad("Normal");
        setSituacao("Fora de Perigo");
        setCenario("BAIXO");
        setMensagemFinal("A probabilidade de ocorrer uma enchente em
            + "\nA situação atual é de estar fora de perigo, onde o nível
    }
    if (total >= 1.50 && total < 2.20) {
        setEstad("Atencao");
        setSituacao("Ficar Alerta");
        setCenario("MEDIO");
        setMensagemFinal("A probabilidade de ocorrer uma enchente em
            + "\nA situação atual é de ficar em alerta, onde o nível
    }
    if (total >= 2.20 && total < 2.80) {
        setEstad("Alerta");
        setSituacao("Alerta Maximo");
        setCenario("ALTO");
        setMensagemFinal("A probabilidade de ocorrer uma enchente em
            + "\nA situação atual é de alerta máximo, onde o nível
    }
    if (total >= 2.80) {
        setEstad("Emergencia");
        setSituacao("Procurar Abrigo");
        setCenario("MUITO ALTO");
        setMensagemFinal("A probabilidade de ocorrer uma enchente em
            + "\nO recomendado é procurar um abrigo, pois o nível
    }
    return total;
}
```

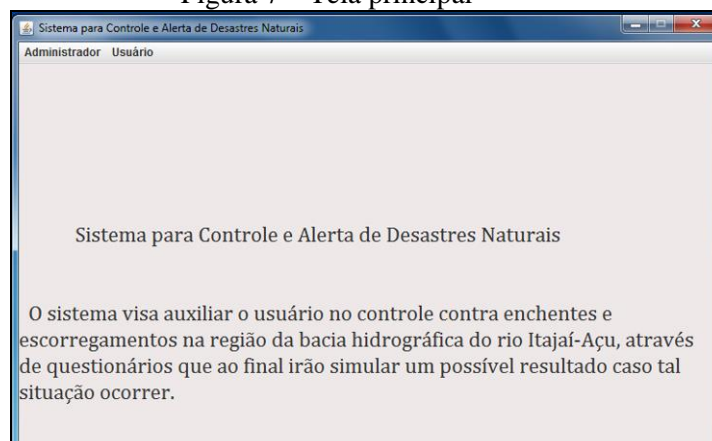
### 3.4.6 Operacionalidade da implementação

Nesta subseção apresentam-se as principais telas do sistema com uma apresentação sobre suas funcionalidades.

#### 3.4.6.1 Tela principal

A tela principal é dividida em dois menus: administrador e usuário como é mostrado na Figura 7.

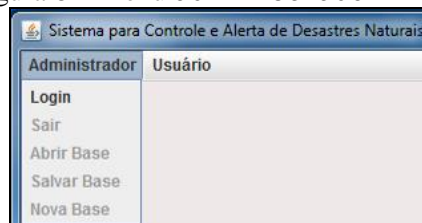
Figura 7 – Tela principal



### 3.4.6.1.1 Administrador

O menu administrador é composto pelos itens *Login*, *Sair*, *Abrir Base*, *Salvar Base* e *Nova Base*. Para estas telas estarem disponíveis é preciso estar logado como administrador clicando no submenu *Login* como mostra a Figura 8.

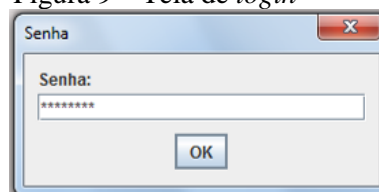
Figura 8 – Menu administrador



### 3.4.6.1.2 Tela de login

Ao clicar no submenu *login* aparecerá a tela para informar a senha de acesso conforme a Figura 9.

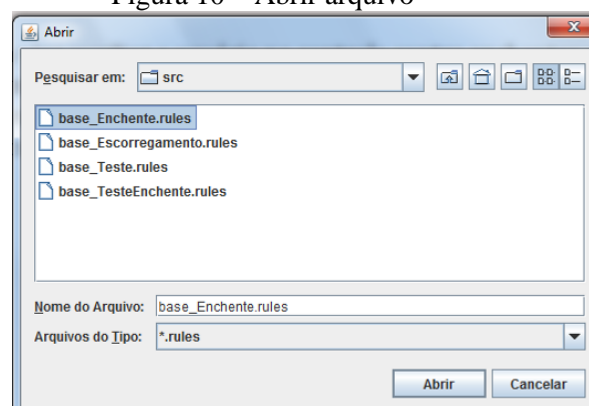
Figura 9 – Tela de login



### 3.4.6.1.3 Abrir base

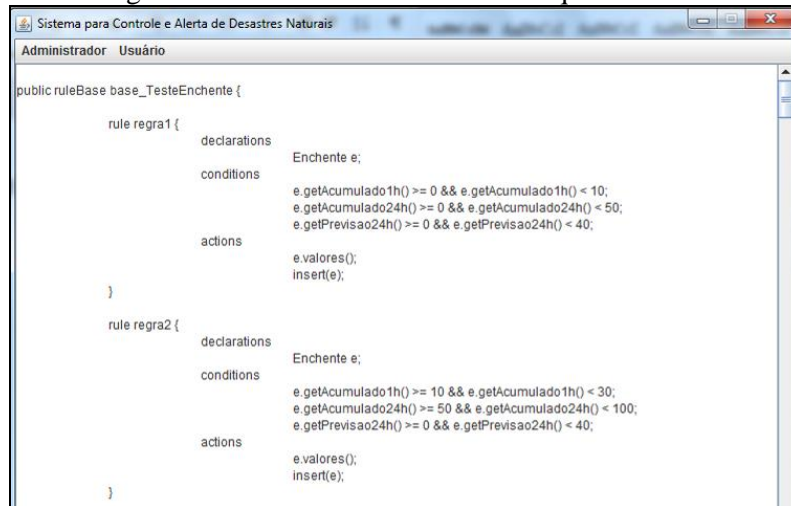
Após efetuar o login no sistema o submenu *abrir base* estará disponível. Clicando nele abrirá uma tela para selecionar um arquivo *.rules* que contém a base de conhecimento, conforme a Figura 10.

Figura 10 – Abrir arquivo



O arquivo de regras aparecerá na tela, conforme a Figura 11, podendo-se adicionar novas regras, alterá-las ou excluí-las. Para salvar alguma alteração feita no arquivo clique no submenu salvar base.

Figura 11 – Mostrando conteúdo do arquivo



```

public ruleBase base_TesteEnchente {

    rule regra1 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
            e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
            e.getPrevisao24h() >= 0 && e.getPrevisao24h() < 40;
        actions
            e.valores();
            insert(e);
    }

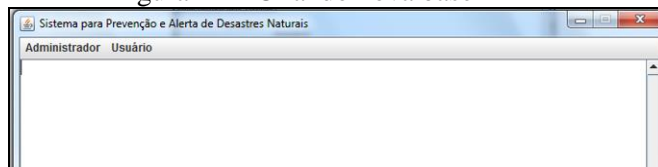
    rule regra2 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getPrevisao24h() >= 0 && e.getPrevisao24h() < 40;
        actions
            e.valores();
            insert(e);
    }
}

```

#### 3.4.6.1.4 Criando nova base

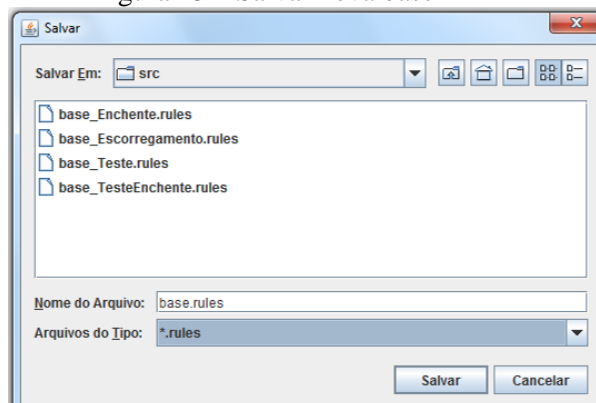
Ao clicar no submenu *nova base*, uma nova tela irá aparecer onde é possível criar uma base de conhecimento conforme a Figura 12.

Figura 12 – Criando nova base



Após criada a nova base clique no submenu *salvar base* que irá abrir uma nova tela para escolher onde salvar o arquivo como mostra a Figura 13.

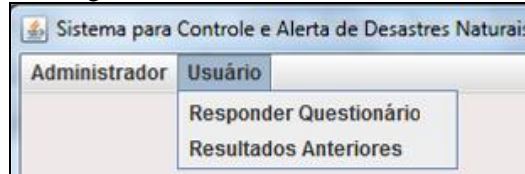
Figura 13 – Salvar nova base



### 3.4.6.2 Usuário

O menu usuário é composto pelos itens Responder Questionário e Resultados Anteriores conforme Figura 14.

Figura 14 – Menu usuário

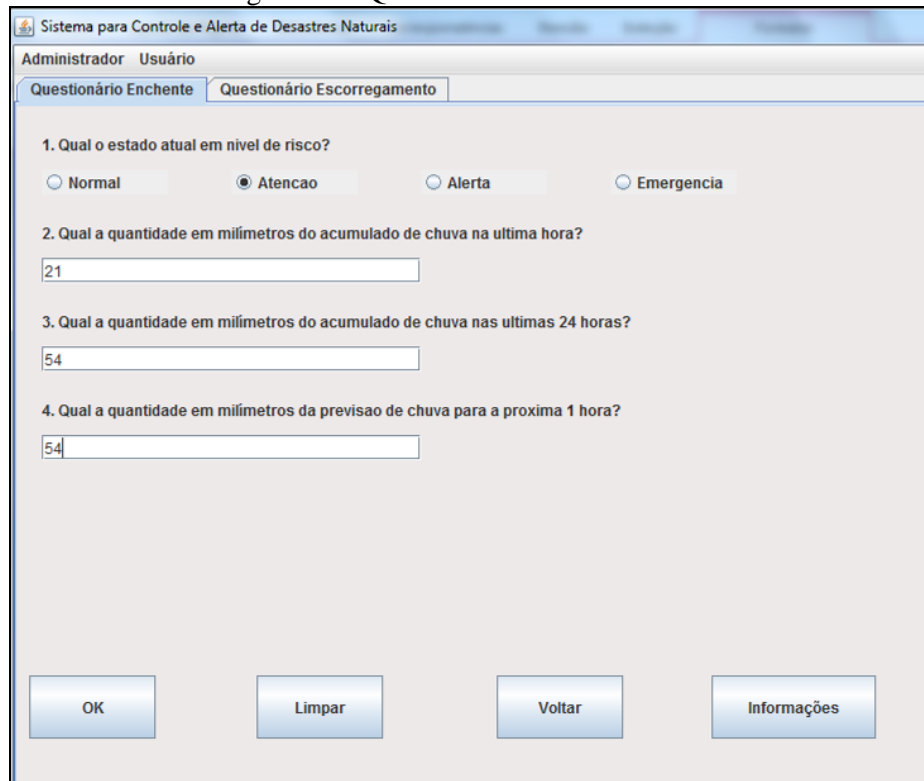


#### 3.4.6.2.1 Responder Questionário

A tela de responder questionário possui dois tipos de questionários: enchentes e escorregamentos, onde são acessados por abas.

A aba do questionário enchente possui quatro questões onde todas devem ser preenchidas e ao final deve-se clicar no botão OK para que seja mostrado o resultado final de acordo com o que foi preenchido, como mostra a Figura 15.

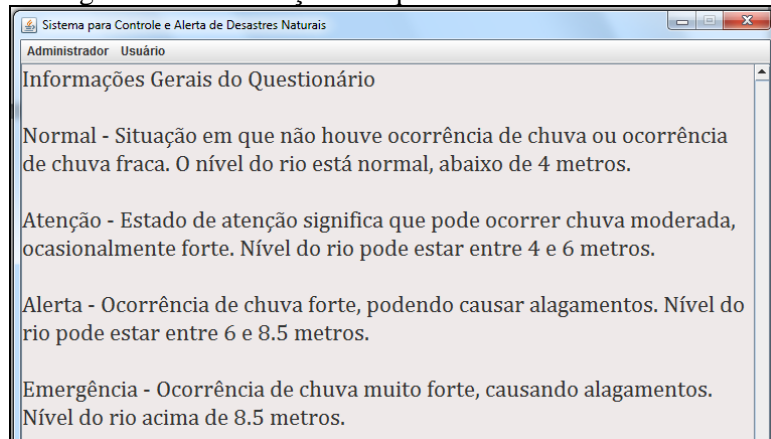
Figura 15 – Questionário enchente

A imagem mostra a interface de um formulário de questionário. No topo, há uma barra de menu com "Administrador" e "Usuário". Abaixo, há duas abas: "Questionário Enchente" (ativa) e "Questionário Escorregamento". O formulário contém quatro perguntas numeradas:  
1. Qual o estado atual em nível de risco?  
Opções:  Normal,  Atencao,  Alerta,  Emergencia  
2. Qual a quantidade em milímetros do acumulado de chuva na ultima hora?  
Campo de texto: 21  
3. Qual a quantidade em milímetros do acumulado de chuva nas ultimas 24 horas?  
Campo de texto: 54  
4. Qual a quantidade em milímetros da previsao de chuva para a proxima 1 hora?  
Campo de texto: 54  
No rodapé, há quatro botões: OK, Limpar, Voltar e Informações.

Para obter informações sobre o questionário clique no botão Informações, que irá abrir uma nova tela explicando o que significa cada opção do questionário conforme mostra a Figura 16.

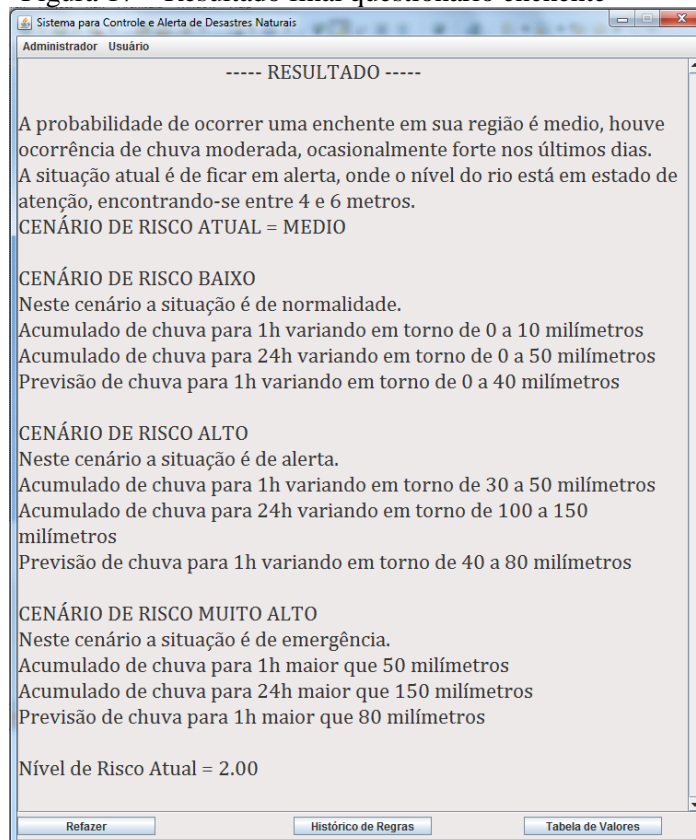


Figura 16 – Informações do questionário enchente



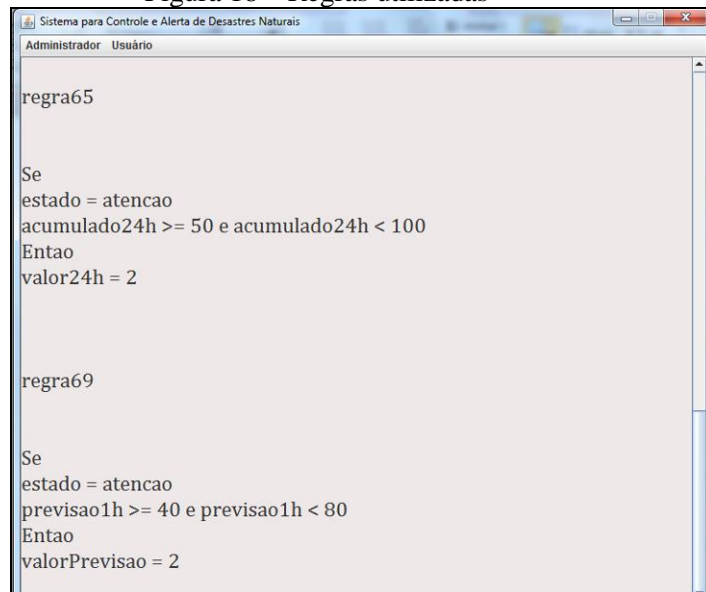
Ao clicar no botão **OK** a tela com o resultado final irá aparecer, informando a situação resultante, os possíveis cenários que podem ocorrer e o nível de risco atual, conforme a Figura 17. Na tela é possível refazer o questionário, visualizar a tabela de risco ou visualizar as regras que foram utilizadas na execução clicando no botão **Histórico de Regras**.

Figura 17 – Resultado final questionário enchente



Na Figura 18 tem-se a tela que mostra as regras aceitas que resultaram no resultado final, elas podem ser visualizadas quando clicado no botão **Histórico de Regras**, disponível na tela do resultado final do questionário.

Figura 18 – Regras utilizadas



Clicando no botão `Tabela de Valores` disponível na tela do resultado final, é possível visualizar uma tabela com o nível de risco de enchentes e seus respectivos valores, conforme a Figura 19.

Figura 19 – Tabela de risco enchente

Nível de Risco	Valores
Baixo	0 a 1.49
Médio	1.50 a 2.19
Alto	2.20 a 2.79
Muito Alto	2.80 a 4.0

A aba do questionário `escorregamento` segue o mesmo princípio do questionário `enchente`. Deve-se responder todas as questões para que o resultado possa ser mostrado. Ao final deve-se clicar no botão `OK`, demonstrado na Figura 20.

Figura 20 – Questionário escorregamento

Sistema para Controle e Alerta de Desastres Naturais

Administrador Usuário

Questionário Enchente Questionário Escorregamento

1. Qual a quantidade em milímetros do acumulado de chuva na ultima hora?  
21

2. Qual a quantidade em milímetros do acumulado de chuva nas ultimas 24 horas?  
54

3. Qual a quantidade em milímetros do acumulado de chuva nas ultimas 168 horas?  
84

4. Qual a quantidade em milímetros da previsao de chuva para a proxima 1 hora?  
61

5. Qual o tipo de geotecnia do local?  
Cortes e Aterros

6. Qual a quantidade de habitantes por hectare?  
21

7. Qual a declividade do local em graus?  
42

8. Qual o tipo de vegetação existente?  
Palmito

9. Qual a quantidade de cobertura vegetal?  
 Inexistente  Baixa  Media  Alta

OK Limpar Voltar Informações

Na Figura 21 tem-se a tela do resultado final para o questionário escorregamento. O resultado é mostrado de acordo com os valores digitados e selecionados nas questões. É possível visualizar as regras que foram executadas para chegar no resultado clicando no botão Histórico de Regras.

Figura 21 – Resultado final questionário escorregamento

Sistema para Controle e Alerta de Desastres Naturais

Administrador Usuário

----- RESULTADO -----

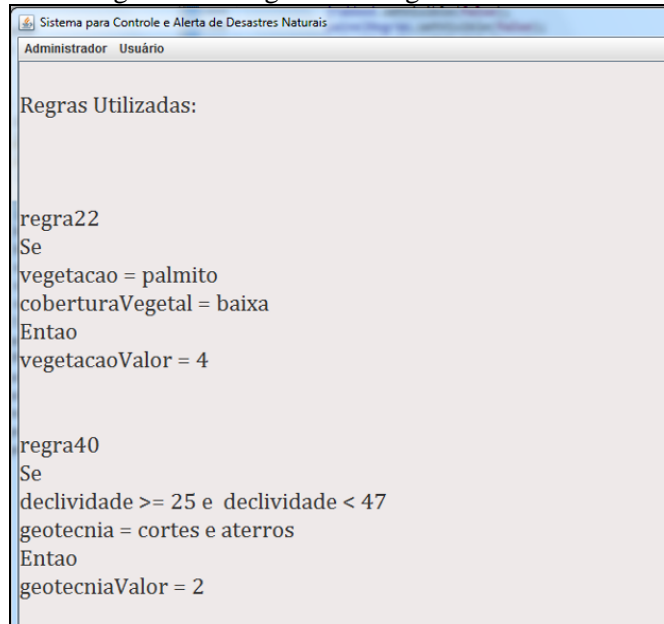
(Alto Risco): alto processo de escorregamentos e solapamentos.  
Apresentam-se evidências de instabilidade de encostas e ocorrência de eventos destrutivos, durante período chuvoso.

Nível de Suscetibilidade a Escorregamento = 2.20

Refazer Histórico de Regras Tabela de Valores

Ao clicar no botão **Histórico** de Regras pode-se visualizar as regras aceitas para chegar no resultado final, conforme a Figura 22.

Figura 22 – Regras escorregamento



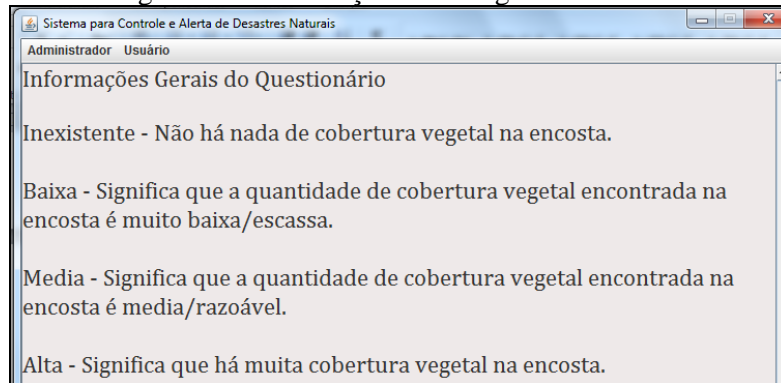
Clicando no botão **Tabela de Valores** disponível na tela do resultado final, tem-se uma tabela com o nível de suscetibilidade de escorregamento e seus respectivos valores, conforme a Figura 23.

Figura 23 – Tabela de suscetibilidade escorregamento

Nível de Suscetibilidade	Valores
Baixo	1.0 a 1.59
Médio	1.60 a 2.19
Alto	2.20 a 2.78
Muito Alto	2.79 a 4.0

Para obter mais informações sobre o questionário clique no botão **Informações** disponível na tela do questionário escorregamento. Nela tem-se uma explicação sobre o significado de cada opção a ser selecionada nas questões, como mostra a Figura 24.

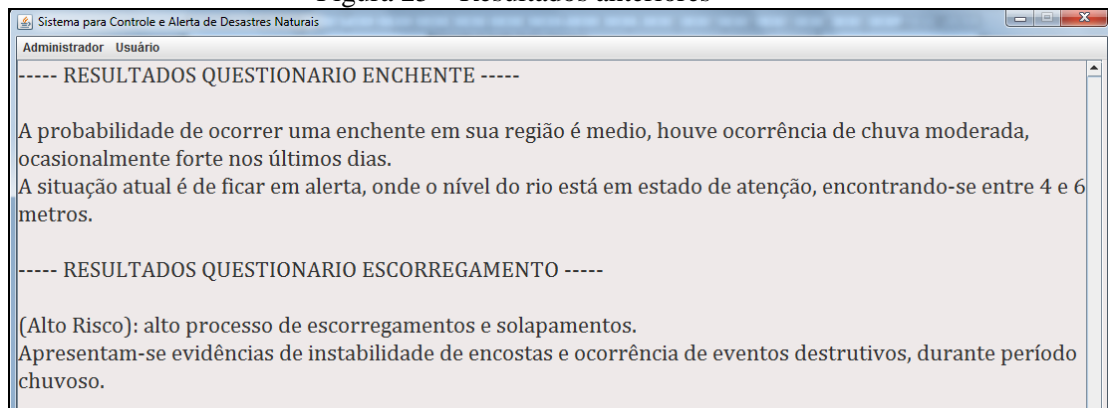
Figura 24 – Informações escorregamento



#### 3.4.6.2.2 Resultados anteriores

Para visualizar todos os resultados obtidos durante a realização dos questionários clique no submenu `resultados anteriores` que fica disponível no menu usuário, conforme a Figura 25.

Figura 25 – Resultados anteriores



### 3.5 RESULTADOS E DISCUSSÃO

O objetivo principal deste trabalho foi desenvolver um SE para auxiliar o usuário a ter um controle contra enchentes e escorregamentos na região da bacia hidrográfica do rio Itajaí-Açu, especificamente para o município de Blumenau, baseado nas simulações de cenários hipotéticos realizados, tanto de cheias como para escorregamentos, verificou-se que tal objetivo foi atingido.

Analisando os trabalhos correlatos, a ferramenta proposta diferencia-se dos mesmos. Enquanto que no trabalho de Leandro, Centeno e Krueger (2010) é utilizada a ferramenta JESS, no presente trabalho foi utilizado a ferramenta JEOPS para representar o conhecimento. Além de que, o objetivo do trabalho de Leandro, Centeno e Krueger (2010), foi desenvolver um SE para diagnóstico de áreas de fragilidade de bacias hidrográficas utilizando o

conhecimento de ciências geodésicas e o foco do presente trabalho foi um SE para alerta contra enchentes e escorregamentos para o município de Blumenau.

No trabalho de Pedrollo e Lanna (2003) o SE foi feito utilizando lógica difusa baseado em regras para a previsão de níveis de enchentes fluviais, enquanto que neste trabalho foi utilizado um SE baseado em regras de produção tanto para enchentes como escorregamentos.

O Quadro 19 traz um comparativo entre o presente trabalho e os correlatos.

Quadro 19 – Comparação entre trabalhos correlatos

Características/Trabalhos	Protótipo desenvolvido	Leandro, Centeno e Krueger (2010)	Pedrollo e Lanna (2003)
Ferramentas utilizadas	JEOPS	JESS	Não identificado
Forma de representação do conhecimento	Regras de produção	Regras de produção	Lógica difusa

Durante o desenvolvimento deste trabalho, foram encontradas algumas dificuldades, principalmente na parte de levantamento de informações para criação das regras de produção, onde juntamente com o CEOPS foram levantados os principais fatores que levam a ocorrência de desastres naturais.

Outra dificuldade em relação a criação das regras é que não foi possível encontrar todas as possibilidades de junção das variáveis, como por exemplo, as variáveis `acumulado1h`, `acumulado24` e `previsao1h` podem conter diversos valores dependendo do que o usuário digitar, sendo assim teria n possibilidades de regras a serem criadas, ficando impossível identificar todas estas possibilidades.

A solução encontrada foi utilizar o processo de calibração que trata-se da flexibilidade do sistema a ir se adequando, ou melhorando os resultados com base na ocorrência de eventos, sendo utilizado através de vários testes realizados no protótipo, ajustando-o da melhor forma.

Foi aplicado, após a finalização do desenvolvimento deste trabalho, um questionário sobre a funcionalidade do protótipo para identificar ocorrências de desastres naturais, sendo aplicado a um integrante do CEOPS que acompanhou o desenvolvimento do mesmo desde o início.

Nas Figuras 26 a 31, são apresentadas as seis perguntas do questionário, utilizadas para avaliar os resultados perante os objetivos do trabalho.

Na Figura 26, apresenta-se o resultado da questão um, mostrando que o protótipo consegue alcançar bons resultados na simulação de uma ocorrência de enchente.

Figura 26 – Questão um

**1. Você acha que o protótipo consegue alcançar bons resultados na simulação de uma ocorrência de enchente? \***

Sim

Não

Na Figura 27, tem-se o resultado da questão dois, mostrando que o protótipo apresenta bons resultados na simulação de uma ocorrência de escorregamento.

Figura 27 – Questão dois

**2. Você acha que o protótipo consegue alcançar bons resultados na simulação de uma ocorrência de escorregamento? \***

Sim

Não

Na avaliação da questão três é mostrado que o protótipo consegue identificar os principais fatores que levam enchentes e escorregamentos a ocorrer, conforme Figura 28.

Figura 28 – Questão três

**3. Você acha que este protótipo consegue demonstrar claramente os fatores que levam enchentes e escorregamentos a ocorrer? \***

Sim

Não

Na avaliação da questão quatro o protótipo foi classificado como “bom” no repasse de informação ao usuário com relação aos desastres naturais, conforme Figura 29.

Figura 29 – Questão quatro

**4. Como você classifica o protótipo no repasse de informação ao usuário com relação aos desastres naturais? \***

Ótimo

Bom

Regular

Ruim

Na Figura 30 é apresentada a avaliação da questão cinco, sendo classificada como “atende totalmente” com relação ao resultado obtido ao final do questionário.

Figura 30 – Questão cinco

**5. Dos resultados apresentados ao final do questionário no protótipo, como você classifica a situação apresentada? \***

Atende Totalmente

Atende Parcialmente

Não Atende

A Figura 31 mostra a operacionalidade da aplicação, na qual foi classificado como “fácil”.

Figura 31 – Questão seis

**6. Como você classifica a operacionalidade do protótipo? \***

Fácil

Médio

Difícil

Optou-se por uma análise qualitativa do protótipo concentrando-se na entrevista de um especialista da área. Conclui-se que o protótipo desenvolvido apresentou bons resultados para simular situações de perigo e alerta de desastres naturais, sendo mais uma aplicação que pode ser utilizada para controle e alerta de desastres naturais.

Com relação aos objetivos específicos propostos todos foram atendidos. Foi possível a construção da base de conhecimento de enchentes e escorregamentos utilizando regras de produção, onde as mesmas foram definidas na ferramenta JEOPS disponibilizadas através de um protótipo.

Com a criação deste protótipo, foi possível obter uma maior clareza em como ocorrem desastres naturais e que fatores levam os mesmos a ocorrer. Mostrando que é possível manter-se informado e diminuindo os riscos de tragédias.



## 4 CONCLUSÕES

Com o avanço desordenado das cidades e o aquecimento global, tem se tornado frequente a ocorrência de catástrofes climáticas que acabam ocasionando grandes transtornos à economia e a sociedade na região. A região do Vale do Itajaí, por si só, é prova disto.

A criação de ferramentas que ajudam a prever o nível das cheias torna-se importante, uma vez que com certa antecedência, a população pode ser avisada sobre o evento e pode tomar medidas para que seu patrimônio humano e físico seja preservado ou que sofra o menor impacto possível. O CEOPS é um exemplo disto, pois ele auxilia a toda a região do Vale do Itajaí em relação às enchentes, conseguindo alertar as pessoas antecipadamente para que se possam evitar maiores desastres.

Com o intuito de proporcionar uma alternativa a mais sobre tais questões, o sistema desenvolvido tem como objetivo simular situações, funcionando como uma ferramenta para previsão de enchentes e escorregamentos, onde foi possível a construção das bases de conhecimento de enchentes e escorregamentos através de regras de produção.

Foi desenvolvido na linguagem Java integrado com a ferramenta JEOPS, responsável pela construção da base de conhecimento e inferência das regras. Para a representação do conhecimento, as regras de produção mostraram-se adequadas para a ferramenta, permitindo representar o conhecimento de forma clara e simples com o auxílio do JEOPS (FIGUEIRA, 2000, p. 114).

De modo geral, este trabalho proporcionou aprendizado tanto na análise do problema em questão como no desenvolvimento da solução, permitindo aplicar os conteúdos adquiridos no decorrer do curso para resolver um problema real onde para Blumenau é uma necessidade, pois enchentes ocorrem com frequência. Contudo, o protótipo elaborado corresponde satisfatoriamente aos objetivos propostos, onde foi possível a construção das bases de conhecimento para criação das regras e os testes realizados mostraram-se adequados conforme questionário aplicado na sessão 3.5. Além disso, a ferramenta poderá ser integrada ao CEOPS, para operar em situações de eventos climáticos adversos.

### 4.1 EXTENSÕES

Para trabalhos futuros, são sugeridas as seguintes extensões:

- a) Incluir uma opção para atualizar a base de conhecimento assim que o arquivo for alterado em tempo de execução;
- b) disponibilizar uma versão do protótipo *online* para que mais usuários possam ter acesso;

- c) adicionar mais tipos de desastres naturais a base de conhecimento, para que se tenha uma maior abrangência;
- d) adicionar uma questão no questionário enchente que permita selecionar o local (bairros ou ruas) que se quer simular uma situação;
- e) adicionar uma questão no questionário de escorregamento relacionada ao tipo de solo do local;
- f) integrar o protótipo com sistemas existentes para obter as entradas de valores nos questionários, evitando a alimentação destes valores por usuários.

## REFERÊNCIAS

ABEL, Mara. **Sistemas especialistas**. 1998. Apostila, 46 f. Universidade Federal do Rio Grande do Sul, Porto Alegre.

BITTENCOURT, Guilherme. **Inteligência artificial: ferramentas e teorias**. 3ª ed. Florianópolis: UFSC, 2006. 371 p.

BITTENCOURT, Guilherme. **Sistemas especialistas**. Disciplina de Inteligência Artificial, Universidade Federal da Bahia, 2008. Disponível em: <<https://disciplinas.dcc.ufba.br/pastas/MATA64/Tutoriais/siesp.pdf>>. Acesso em: 18 mar. 2014.

CASSETI, Valter. **Ambiente e apropriação do relevo**. São Paulo: Contexto, 1991. 147 p.

CASTRO, Antônio Luiz C.; et al. **Manual de desastres naturais**. Brasília: Defesa Civil de Minas Gerais, 2003. 174 p.

CEOPS, **CEOPS**, 2014. Disponível em: <<http://ceops.furb.br>>. Acesso em: 18 mar. 2014.  
CRIPPA, Maurício. **Sistemas especialistas: a engenharia do conhecimento aplicada às organizações**. Florianópolis, 1999. Disponível em: <<http://n27.udesc.br/demo/trabalhos/alunos/mc/index.html>>. Acesso em: 18 mar. 2014.

DIEHL, Vera A. **Protótipo para gerenciamento de programa da qualidade (5s) utilizando sistemas especialistas**. 2000. 71 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Curso de Ciência da Computação, Universidade Regional de Blumenau, Blumenau. Disponível em: <<http://campeche.inf.furb.br/tccs/2000-II/2000-2veraalicediehlvf.pdf>>. Acesso em: 18 mar. 2014.

FEIGENBAUM, Edward. A.; McCORDUCK, Pamela. **The fifth generation: Artificial Intelligence and Japan's computer challenge to the world**. Boston: Addison-Wesley Pub. Co., 1983. 275 p.

FIGUEIRA, Carlos S. **JEOPS: uma ferramenta para o desenvolvimento de aplicações inteligentes em Java**. 2000. 141 f. Tese (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife.

FIGUEIREDO, Ciro J.; et al. **Uso de Sistemas especialistas para a avaliação de um processo agroindustrial**. 2011. Disponível em: <<http://ojs.ingepro.com.br/index.php/ingepro/article/download/344/303>>. Acesso em: 18 mar. 2014.

FLORES, Cecília D. Fundamentos dos Sistemas Especialistas. In: BARONE D. A. C. **Sociedades artificiais: a nova fronteira da inteligência das máquinas**. Porto Alegre: Bookman, 2003.

GENARO, Sérgio. **Sistemas especialistas: o conhecimento artificial**. São Paulo: Rio de Janeiro: LTC, 1987.

GIARRATANO, Joseph; RILEY, Gary. **Expert systems: principles and programming**. 3. ed. Boston: PWS Publishing Company, 1998. 597 p.

GUIDICINI, Guido; IWASA, Oswaldo. Y. **Ensaio de correlação entre pluviosidade e escorregamentos em meio tropical úmido**. São Paulo: Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, 1976. 48 p. (Relatório n. 1080).

HEINZLE, Roberto. **Protótipo de uma ferramenta para criação de sistemas especialistas baseados em regras de produção**. 1995. 145f. Tese (Mestrado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

HENRIQUE, Ary; NEUMANN, Leandro; CÁPUA, Renatha. **Sistema especialista para análise de investimentos do Banco do Brasil**. 2003. 32 f. Tese (Pós-Graduação em Ciência da Computação) - Curso de Pós-Graduação em Ciência da Computação, Universidade Federal Fluminense, Niterói.

KOBIYAMA, Masato; et al. Papel da comunidade e da universidade no gerenciamento de desastres naturais. In: Simpósio Brasileiro de Desastres Naturais, 1., 2004, Florianópolis. **Anais...** Florianópolis: GEDN/UFSC, 2004.

KOBIYAMA, Masato; et al. **Prevenção de desastres naturais**. Curitiba: Organic Training, 2006.

LEANDRO, Diuliana; CENTENO, Jorge; KRUEGER, Cláudia P. Determinação de áreas de fragilidade utilizando um sistema especialista. In: Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação, 3., 2010. Recife. **Anais...** Recife: UFPR, 2010. Disponível em:

<<http://www.lage.ufpr.br/downloads/publicacoes/2010/SISTEMA%20ESPECIALISTA.pdf>>. Acesso em: 07 set. 2013.

LIA, Laboratório de Inteligência Artificial. **Expert SINTA: uma ferramenta para criação de sistemas especialistas**. 1999. Disponível em: <<http://www.lia.ufc.br>>. Acesso em: 18 mar. 2014.

LUGER, George. F. **Structures and strategies for complex problem solving**, 3ª ed., [S.l.]: Addison Wesley, 1998.

MANCHINI, Daniella P. **Sistemas especialistas**. Maringá: UEM, 2005. Disponível em: <<http://www.din.uem.br/ia/intelige/especialistas/especialistas/>>. Acesso em: 18 mar. 2014.

NIKOLOPOULOS, Chris. **Expert systems: introduction to first and second generation and hybrid knowledge based systems**. New York: Marcel Dekker Inc, 1997. 331p.

NOGUEIRA, José H. M.; SILVA, Ricardo B. A. Geração automática de aplicações especialistas usando o Expert SINTA. In: Simpósio Brasileiro de Engenharia de Software, 11., 1997. São Carlos. **Anais...** São Carlos, 1997.

PEDROLLO, Olavo C.; LANNA, Antônio E. Previsão de cheias em tempo atual com sistema especialista difuso. **RBRH - Revista Brasileira de Recursos Hídricos**, Porto Alegre, v. 8, n. 3, p. 45-60, jan/mar. 2003. Disponível em:

<[http://www.abrh.org.br/SGCv3/UserFiles/Sumarios/efced3d0ecf851a2e181c4d045076edd\\_2f2a0f653999ca83795f1e6b1f5900c1.pdf](http://www.abrh.org.br/SGCv3/UserFiles/Sumarios/efced3d0ecf851a2e181c4d045076edd_2f2a0f653999ca83795f1e6b1f5900c1.pdf)>. Acesso em: 07 set. 2013.

PEDROLLO, Olavo C. **Previsão em tempo atual de cheias com uso de sistema especialista difuso**. 2000. 134 f. Tese (Pós-Graduação em Engenharia de Recursos Hídricos e Saneamento Ambiental) – Curso de Pós-Graduação em Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em:

<<http://www.lume.ufrgs.br/bitstream/handle/10183/3041/000285802.pdf?sequence=1>>. Acesso em: 07 set. 2013.

PEREIRA, Marcelo D. **Protótipo de um sistema especialista difuso para seleção de imóveis em uma imobiliária**. 2000. 70f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PY, Mônica Xavier. **Sistemas especialistas: uma introdução**. 2006. Instituto de informática, Universidade federal do Rio Grande do Sul. Disponível em:

<<http://www.inf.ufrgs.br/gppd/disc/cmp135/trabs/mpy/sistemaspecialistas.pdf>>. Acesso em: 18 mar. 2014.

RABUSKE, Renato A. **Inteligência artificial**. Florianópolis: Editora da UFSC, 1995.

SAVARIS, Silvana. V. A. M. **Sistema especialista para primeiros socorros para cães**. 2002. 154f. Tese (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <<http://www.inf.ufsc.br/~barreto/teses/savaris.pdf>>. Acesso em: 18 mar. 2014.

SELBY, Michael J. **Hillslope materials and processes**. 2ed. Oxford: Oxford Univ. Press, 1993. 451p.

TOMINAGA, Lídia K.; SANTORO, Jair; AMARAL, Rosangela. **Desastres naturais: conhecer para prevenir**. São Paulo: Instituto Geológico, 2009.

VIEIRA, Rafaela; et al. Mapeamento das áreas suscetíveis a escorregamentos em Blumenau-SC, utilizando Sistema de Informações Geográficas. In: Simpósio Brasileiro de Sensoriamento Remoto, 13., 2007. Florianópolis. **Anais...** Florianópolis: INPE, 2007.

## APÊNDICE A – Regras de produção

Nos Quadros 20 e 21 são apresentadas as regras de produção do SE.

No Quadro 20 é apresentada a base de conhecimento para enchentes e suas respectivas regras.

Quadro 20 – Base de conhecimento enchente

```

public ruleBase base_Enchente {
    rule regra1 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        actions
            e.setValor1h(1);
    }

    rule regra2 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        actions
            e.setValor1h(1);
    }

    rule regra3 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        actions
            e.setValor1h(2);
    }

    rule regra4 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 100;
        actions
            e.setValor1h(2);
    }

    rule regra5 {
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Normal");
            e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        actions
            e.setValor24h(1);
    }
}

```

```
}  
  
rule regra6 {  
    declarations  
        Enchente e;  
    conditions  
        e.getEstad().equalsIgnoreCase("Normal");  
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;  
    actions  
        e.setValor24h(1);  
}  
  
rule regra7 {  
    declarations  
        Enchente e;  
    conditions  
        e.getEstad().equalsIgnoreCase("Normal");  
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;  
    actions  
        e.setValor24h(2);  
}  
  
rule regra8 {  
    declarations  
        Enchente e;  
    conditions  
        e.getEstad().equalsIgnoreCase("Normal");  
        e.getAcumulado24h() >= 150;  
    actions  
        e.setValor24h(3);  
}  
  
rule regra9 {  
    declarations  
        Enchente e;  
    conditions  
        e.getEstad().equalsIgnoreCase("Normal");  
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;  
    actions  
        e.setValorP(1);  
}  
  
rule regra10 {  
    declarations  
        Enchente e;  
    conditions  
        e.getEstad().equalsIgnoreCase("Normal");  
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;  
    actions  
        e.setValorP(1);  
}  
  
rule regra11 {  
    declarations  
        Enchente e;  
    conditions  
        e.getEstad().equalsIgnoreCase("Normal");  
        e.getPrevisao1h() >= 80;  
    actions  
        e.setValorP(2);  
}
```

```
rule regra12 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
    actions
        e.setValor1h(2);
}

rule regra13 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
    actions
        e.setValor1h(2);
}

rule regra14 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
    actions
        e.setValor1h(3);
}

rule regra15 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado1h() >= 50;
    actions
        e.setValor1h(3);
}

rule regra16 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
    actions
        e.setValor24h(2);
}

rule regra17 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
    actions
        e.setValor24h(2);
}
```



```
rule regra18 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
    actions
        e.setValor24h(3);
}

rule regra19 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getAcumulado24h() >= 150;
    actions
        e.setValor24h(3);
}

rule regra20 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorP(2);
}

rule regra21 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorP(2);
}

rule regra22 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Atencao");
        e.getPrevisao1h() >= 80;
    actions
        e.setValorP(3);
}

rule regra23 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getAcumuladolh() >= 0 && e.getAcumuladolh() < 10;
    actions
        e.setValor1h(2);
}

rule regra24 {
```

```
        declarations
            Enchente e;
        conditions
            e.getEstad().equalsIgnoreCase("Alerta");
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        actions
            e.setValor1h(3);
    }

rule regra25 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
    actions
        e.setValor1h(3);
}

rule regra26 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getAcumulado1h() >= 50;
    actions
        e.setValor1h(4);
}

rule regra27 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
    actions
        e.setValor24h(3);
}

rule regra28 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
    actions
        e.setValor24h(3);
}

rule regra29 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
    actions
        e.setValor24h(4);
}

rule regra30 {
    declarations
```

```
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getAcumulado24h() >= 150;
    actions
        e.setValor24h(4);
}

rule regra31 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorP(2);
}

rule regra32 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorP(3);
}

rule regra33 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Alerta");
        e.getPrevisao1h() >= 80;
    actions
        e.setValorP(4);
}

rule regra34 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getAcumuladolh() >= 0 && e.getAcumuladolh() < 10;
    actions
        e.setValor1h(3);
}

rule regra35 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getAcumuladolh() >= 10 && e.getAcumuladolh() < 30;
    actions
        e.setValor1h(3);
}

rule regra36 {
    declarations
        Enchente e;
```

```
        conditions
            e.getEstad().equalsIgnoreCase("Emergencia");
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        actions
            e.setValor1h(3);
    }

rule regra37 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getAcumulado1h() >= 50;
    actions
        e.setValor1h(4);
}

rule regra38 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
    actions
        e.setValor24h(3);
}

rule regra39 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
    actions
        e.setValor24h(4);
}

rule regra40 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
    actions
        e.setValor24h(4);
}

rule regra41 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getAcumulado24h() >= 150;
    actions
        e.setValor24h(4);
}

rule regra42 {
    declarations
        Enchente e;
    conditions
```

```

        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorP(3);
}

rule regra43 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorP(4);
}

rule regra44 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Emergencia");
        e.getPrevisao1h() >= 80;
    actions
        e.setValorP(4);
}

rule regra45 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Normal");
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 150;
    actions
        e.setValor1h(3);
}

rule regra46 {
    declarations
        Enchente e;
    conditions
        e.getEstad().equalsIgnoreCase("Normal");
        e.getAcumulado1h() >= 150;
    actions
        e.setValor1h(4);
}

rule regra47 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra48 {
    declarations
        Enchente e;

```

```

        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

rule regra49 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra50 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra51 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra52 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra53 {
    declarations
        Enchente e;
    conditions

```

```

        e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra54 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra55 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 150;
        e.getAcumulado24h() >= 10 && e.getAcumulado24h() < 30;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra56 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra57 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra58 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;

```

```

        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra59 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra60 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra61 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra62 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() <= 100;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra63 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() <= 150;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;

```



```

        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
    }

rule regra64 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() <= 210;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
    }

rule regra65 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 50;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
    }

rule regra66 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
    }

rule regra67 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
    }

rule regra68 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;

```

```

        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

rule regra69 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getPrevisao1h() >= 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra70 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra71 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getPrevisao1h() >= 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra72 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra73 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions

```

```

        e.setAcumulad_chuva(e.valores());
        insert(e);
    }

    rule regra74 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 150;
            e.getPrevisao1h() >= 80;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra75 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra76 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getPrevisao1h() >= 80;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra77 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra78 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getPrevisao1h() >= 80;
        actions
            e.setAcumulad_chuva(e.valores());
    }

```

```

        insert(e);
    }

    rule regra79 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 150;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra80 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 150;
            e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra81 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 150;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra82 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

    rule regra83 {
        declarations
            Enchente e;
        conditions
            e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
        actions
            e.setAcumulad_chuva(e.valores());
            insert(e);
    }

```

```
}  
  
rule regra84 {  
  declarations  
    Enchente e;  
  conditions  
    e.getAcumulado1h() >= 50;  
    e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;  
    e.getPrevisao1h() >= 80;  
  actions  
    e.setAcumulad_chuva(e.valores());  
    insert(e);  
}  
  
rule regra85 {  
  declarations  
    Enchente e;  
  conditions  
    e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;  
    e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;  
    e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;  
  actions  
    e.setAcumulad_chuva(e.valores());  
    insert(e);  
}  
  
rule regra86 {  
  declarations  
    Enchente e;  
  conditions  
    e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;  
    e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;  
    e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;  
  actions  
    e.setAcumulad_chuva(e.valores());  
    insert(e);  
}  
  
rule regra87 {  
  declarations  
    Enchente e;  
  conditions  
    e.getAcumulado1h() >= 50;  
    e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;  
    e.getPrevisao1h() >= 80;  
  actions  
    e.setAcumulad_chuva(e.valores());  
    insert(e);  
}  
  
rule regra88 {  
  declarations  
    Enchente e;  
  conditions  
    e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;  
    e.getAcumulado24h() >= 150;  
    e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;  
  actions  
    e.setAcumulad_chuva(e.valores());  
    insert(e);  
}
```

```
rule regra89 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra90 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 150;
        e.getPrevisao1h() >= 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra91 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra92 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra93 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 210;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}
```

```

rule regra94 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra95 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}

rule regra96 {
    declarations
        Enchente e;
    conditions
        e.getAcumulado1h() >= 150;
        e.getAcumulado24h() >= 210;
        e.getPrevisao1h() >= 80;
    actions
        e.setAcumulad_chuva(e.valores());
        insert(e);
}
}

```

No Quadro 21 é apresentada a base de conhecimento para escorregamentos e suas respectivas regras.

#### Quadro 21 – Base de conhecimento escorregamento

```

public ruleBase base_Escorregamento {
    rule regra1 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("floresta secundaria");
            e.getCobertura_vegetal().equalsIgnoreCase("alta");
        actions
            e.setVegetacaoNum(1);
    }

    rule regra2 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 0 && e.getDeclividade() < 25;
            e.getGeotecnia().equalsIgnoreCase("solapamento das
margens");
        actions

```

```
        e.setGeotecniaNum(1);
    }

    rule regra3 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("floresta secundaria");
            e.getCobertura_vegetal().equalsIgnoreCase("media");
        actions
            e.setVegetacaoNum(1);
    }

    rule regra4 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("floresta secundaria");
            e.getCobertura_vegetal().equalsIgnoreCase("baixa");
        actions
            e.setVegetacaoNum(2);
    }

    rule regra5 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capeirao");
            e.getCobertura_vegetal().equalsIgnoreCase("alta");
        actions
            e.setVegetacaoNum(1);
    }

    rule regra6 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capeirao");
            e.getCobertura_vegetal().equalsIgnoreCase("media");
        actions
            e.setVegetacaoNum(1);
    }

    rule regra7 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capeirao");
            e.getCobertura_vegetal().equalsIgnoreCase("baixa");
        actions
            e.setVegetacaoNum(2);
    }

    rule regra8 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capoeira");
            e.getCobertura_vegetal().equalsIgnoreCase("alta");
        actions
            e.setVegetacaoNum(2);
    }
```



```

    }

    rule regra9 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capoeira");
            e.getCobertura_vegetal().equalsIgnoreCase("media");
        actions
            e.setVegetacaoNum(2);
    }

    rule regra10 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capoeira");
            e.getCobertura_vegetal().equalsIgnoreCase("baixa");
        actions
            e.setVegetacaoNum(3);
    }

    rule regra11 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("reflorestamento de
pinus");
            e.getCobertura_vegetal().equalsIgnoreCase("alta");
        actions
            e.setVegetacaoNum(2);
    }

    rule regra12 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("reflorestamento de
pinus");
            e.getCobertura_vegetal().equalsIgnoreCase("media");
        actions
            e.setVegetacaoNum(2);
    }

    rule regra13 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("reflorestamento de
pinus");
            e.getCobertura_vegetal().equalsIgnoreCase("baixa");
        actions
            e.setVegetacaoNum(3);
    }

    rule regra14 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capoeirinha");
            e.getCobertura_vegetal().equalsIgnoreCase("alta");
    }

```

```

        actions
            e.setVegetacaoNum(3);
    }

    rule regra15 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capoeirinha");
            e.getCobertura_vegetal().equalsIgnoreCase("media");
        actions
            e.setVegetacaoNum(3);
    }

    rule regra16 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("capoeirinha");
            e.getCobertura_vegetal().equalsIgnoreCase("baixa");
        actions
            e.setVegetacaoNum(4);
    }

    rule regra17 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("reflorestamento de
eucalipto");
            e.getCobertura_vegetal().equalsIgnoreCase("alta");
        actions
            e.setVegetacaoNum(3);
    }

    rule regra18 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("reflorestamento de
eucalipto");
            e.getCobertura_vegetal().equalsIgnoreCase("media");
        actions
            e.setVegetacaoNum(3);
    }

    rule regra19 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("reflorestamento de
eucalipto");
            e.getCobertura_vegetal().equalsIgnoreCase("baixa");
        actions
            e.setVegetacaoNum(4);
    }

    rule regra20 {
        declarations
            Escorregamento e;
        conditions

```

```
        e.getVegetacao().equalsIgnoreCase("palmito");
        e.getCobertura_vegetal().equalsIgnoreCase("alta");
    actions
        e.setVegetacaoNum(3);
}

rule regra21 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("palmito");
        e.getCobertura_vegetal().equalsIgnoreCase("media");
    actions
        e.setVegetacaoNum(3);
}

rule regra22 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("palmito");
        e.getCobertura_vegetal().equalsIgnoreCase("baixa");
    actions
        e.setVegetacaoNum(4);
}

rule regra23 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("samambaia");
        e.getCobertura_vegetal().equalsIgnoreCase("alta");
    actions
        e.setVegetacaoNum(3);
}

rule regra24 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("samambaia");
        e.getCobertura_vegetal().equalsIgnoreCase("media");
    actions
        e.setVegetacaoNum(3);
}

rule regra25 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("samambaia");
        e.getCobertura_vegetal().equalsIgnoreCase("baixa");
    actions
        e.setVegetacaoNum(4);
}

rule regra26 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("pastagem");
```

```
        e.getCobertura_vegetal().equalsIgnoreCase("baixa");
    actions
        e.setVegetacaoNum(4);
}

rule regra27 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("pastagem");
        e.getCobertura_vegetal().equalsIgnoreCase("inexistente");
    actions
        e.setVegetacaoNum(4);
}

rule regra28 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("pomar");
        e.getCobertura_vegetal().equalsIgnoreCase("baixa");
    actions
        e.setVegetacaoNum(4);
}

rule regra29 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("pomar");
        e.getCobertura_vegetal().equalsIgnoreCase("inexistente");
    actions
        e.setVegetacaoNum(4);
}

rule regra30 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("quintal");
        e.getCobertura_vegetal().equalsIgnoreCase("baixa");
    actions
        e.setVegetacaoNum(4);
}

rule regra31 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("quintal");
        e.getCobertura_vegetal().equalsIgnoreCase("inexistente");
    actions
        e.setVegetacaoNum(4);
}

rule regra32 {
    declarations
        Escorregamento e;
    conditions
        e.getVegetacao().equalsIgnoreCase("solo exposto");
        e.getCobertura_vegetal().equalsIgnoreCase("baixa");
}
```

```

        actions
            e.setVegetacaoNum(4);
    }

    rule regra33 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("solo exposto");
            e.getCobertura_vegetal().equalsIgnoreCase("inexistente");
        actions
            e.setVegetacaoNum(4);
    }

    rule regra34 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("area ocupada");
            e.getCobertura_vegetal().equalsIgnoreCase("baixa");
        actions
            e.setVegetacaoNum(4);
    }

    rule regra35 {
        declarations
            Escorregamento e;
        conditions
            e.getVegetacao().equalsIgnoreCase("area ocupada");
            e.getCobertura_vegetal().equalsIgnoreCase("inexistente");
        actions
            e.setVegetacaoNum(4);
    }

    rule regra36 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 25 && e.getDeclividade() < 47;
            e.getGeotecnia().equalsIgnoreCase("solapamento das
margens");
        actions
            e.setGeotecniaNum(1);
    }

    rule regra37 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 47 && e.getDeclividade() < 60;
            e.getGeotecnia().equalsIgnoreCase("solapamento das
margens");
        actions
            e.setGeotecniaNum(2);
    }

    rule regra38 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 60;

```

```

        e.getGeotecnia().equalsIgnoreCase("solapamento das
margens");
        actions
            e.setGeotecniaNum(2);
    }

    rule regra39 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 0 && e.getDeclividade() < 25;
            e.getGeotecnia().equalsIgnoreCase("cortes e aterros");
        actions
            e.setGeotecniaNum(2);
    }

    rule regra40 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 25 && e.getDeclividade() < 47;
            e.getGeotecnia().equalsIgnoreCase("cortes e aterros");
        actions
            e.setGeotecniaNum(2);
    }

    rule regra41 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 47 && e.getDeclividade() < 60;
            e.getGeotecnia().equalsIgnoreCase("cortes e aterros");
        actions
            e.setGeotecniaNum(3);
    }

    rule regra42 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 60;
            e.getGeotecnia().equalsIgnoreCase("cortes e aterros");
        actions
            e.setGeotecniaNum(3);
    }

    rule regra43 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 0 && e.getDeclividade() < 25;
            e.getGeotecnia().equalsIgnoreCase("cortes, aterros e
abastecimento informal de agua e esgoto");
        actions
            e.setGeotecniaNum(3);
    }

    rule regra44 {
        declarations
            Escorregamento e;
        conditions

```

```

        e.getDeclividade() >= 25 && e.getDeclividade() < 47;
        e.getGeotecnia().equalsIgnoreCase("cortes, aterros e
abastecimento informal de agua e esgoto");
        actions
            e.setGeotecniaNum(3);
    }

    rule regra45 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 47 && e.getDeclividade() < 60;
            e.getGeotecnia().equalsIgnoreCase("cortes, aterros e
abastecimento informal de agua e esgoto");
        actions
            e.setGeotecniaNum(4);
    }

    rule regra46 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 60;
            e.getGeotecnia().equalsIgnoreCase("cortes, aterros e
abastecimento informal de agua e esgoto");
        actions
            e.setGeotecniaNum(4);
    }

    rule regra47 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 0 && e.getDeclividade() < 25;
            e.getGeotecnia().equalsIgnoreCase("cortes, aterros,
erosao no solo, vazamentos de agua e esgoto");
        actions
            e.setGeotecniaNum(3);
    }

    rule regra48 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 25 && e.getDeclividade() < 47;
            e.getGeotecnia().equalsIgnoreCase("cortes, aterros,
erosao no solo, vazamentos de agua e esgoto");
        actions
            e.setGeotecniaNum(3);
    }

    rule regra49 {
        declarations
            Escorregamento e;
        conditions
            e.getDeclividade() >= 47 && e.getDeclividade() < 60;
            e.getGeotecnia().equalsIgnoreCase("cortes, aterros,
erosao no solo, vazamentos de agua e esgoto");
        actions
            e.setGeotecniaNum(4);
    }

```

```

rule regra50 {
    declarations
        Escorregamento e;
    conditions
        e.getDeclividade() >= 60;
        e.getGeotecnia().equalsIgnoreCase("cortes, aterros,
erosao no solo, vazamentos de agua e esgoto");
    actions
        e.setGeotecniaNum(4);
}

rule regra51 {
    declarations
        Escorregamento e;
    conditions
        e.getPopulacao() >= 1 && e.getPopulacao() <= 35;
    actions
        e.setPopulacaoNum(1);
}

rule regra52 {
    declarations
        Escorregamento e;
    conditions
        e.getPopulacao() >= 36 && e.getPopulacao() <= 70;
    actions
        e.setPopulacaoNum(2);
}

rule regra53 {
    declarations
        Escorregamento e;
    conditions
        e.getPopulacao() >= 71 && e.getPopulacao() <= 105;
    actions
        e.setPopulacaoNum(3);
}

rule regra54 {
    declarations
        Escorregamento e;
    conditions
        e.getPopulacao() >= 106;
    actions
        e.setPopulacaoNum(4);
}

rule regra55 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
    actions
        e.setValorChuva(1);
        insert(e);
}

rule regra56 {

```



```

    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorChuva(2);
        insert(e);
    }

    rule regra57 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
            e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
        actions
            e.setValorChuva(3);
            insert(e);
    }

    rule regra58 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 50;
            e.getAcumulado24h() >= 150;
            e.getAcumulado168h() >= 210;
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra59 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
        actions
            e.setValorChuva(1);
            insert(e);
    }

    rule regra60 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
        actions
            e.setValorChuva(1);
    }

```

```

        insert(e);
    }

    rule regra61 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 0 && e.getAcumulado1h() < 10;
            e.getAcumulado24h() >= 150;
            e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
        actions
            e.setValorChuva(1);
            insert(e);
    }

    rule regra62 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
            e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
        actions
            e.setValorChuva(1);
            insert(e);
    }

    rule regra63 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
        actions
            e.setValorChuva(2);
            insert(e);
    }

    rule regra64 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
        actions
            e.setValorChuva(2);
            insert(e);
    }

    rule regra65 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 150;
            e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
        actions
            e.setValorChuva(2);
            insert(e);
    }

```

```
}  
  
rule regra66 {  
  declarations  
    Escorregamento e;  
  conditions  
    e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;  
    e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;  
    e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;  
  actions  
    e.setValorChuva(1);  
    insert(e);  
}  
  
rule regra67 {  
  declarations  
    Escorregamento e;  
  conditions  
    e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;  
    e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;  
    e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;  
  actions  
    e.setValorChuva(2);  
    insert(e);  
}  
  
rule regra68 {  
  declarations  
    Escorregamento e;  
  conditions  
    e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;  
    e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;  
    e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;  
  actions  
    e.setValorChuva(3);  
    insert(e);  
}  
  
rule regra69 {  
  declarations  
    Escorregamento e;  
  conditions  
    e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;  
    e.getAcumulado24h() >= 150;  
    e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;  
  actions  
    e.setValorChuva(3);  
    insert(e);  
}  
  
rule regra70 {  
  declarations  
    Escorregamento e;  
  conditions  
    e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 100;  
    e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;  
    e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;  
  actions  
    e.setValorChuva(1);  
    insert(e);  
}
```

```

rule regra71 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
    actions
        e.setValorChuva(2);
        insert(e);
}

rule regra72 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
    actions
        e.setValorChuva(3);
        insert(e);
}

rule regra73 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50;
        e.getAcumulado24h() >= 150;
        e.getAcumulado168h() >= 0 && e.getAcumulado168h() < 100;
    actions
        e.setValorChuva(4);
        insert(e);
}

rule regra74 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorChuva(2);
        insert(e);
}

rule regra75 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions

```

```

        e.setValorChuva(3);
        insert(e);
    }

    rule regra76 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
            e.getAcumulado168h() >= 210;
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra77 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
            e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
        actions
            e.setValorChuva(3);
            insert(e);
    }

    rule regra78 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getAcumulado168h() >= 210;
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra79 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setValorChuva(2);
            insert(e);
    }

    rule regra80 {
        declarations
            Escorregamento e;

```

```

        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getAcumulado168h() >= 210;
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra81 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 150;
            e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra82 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 150;
            e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
            e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
        actions
            e.setValorChuva(3);
            insert(e);
    }

    rule regra83 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 10 && e.getAcumulado1h() < 30;
            e.getAcumulado24h() >= 150;
            e.getAcumulado168h() >= 210;
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra84 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions

```

```

        e.setValorChuva(2);
        insert(e);
    }

    rule regra85 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
            e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
        actions
            e.setValorChuva(3);
            insert(e);
    }

    rule regra86 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
            e.getAcumulado168h() >= 210;
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra87 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
            e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
        actions
            e.setValorChuva(3);
            insert(e);
    }

    rule regra88 {
        declarations
            Escorregamento e;
        conditions
            e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
            e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
            e.getAcumulado168h() >= 210;
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(4);
            insert(e);
    }

    rule regra89 {
        declarations
            Escorregamento e;

```

```

conditions
    e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
    e.getAcumulado24h() >= 150;
    e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
    e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
actions
    e.setValorChuva(3);
    insert(e);
}

rule regra90 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        e.getAcumulado24h() >= 150;
        e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(4);
        insert(e);
}

rule regra91 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 30 && e.getAcumulado1h() < 50;
        e.getAcumulado24h() >= 150;
        e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorChuva(2);
        insert(e);
}

rule regra92 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorChuva(2);
        insert(e);
}

rule regra93 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;

```



```

        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(3);
        insert(e);
    }

rule regra94 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 210;
        e.getPrevisao1h() >= 80;
    actions
        e.setValorChuva(4);
        insert(e);
    }

rule regra95 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorChuva(2);
        insert(e);
    }

rule regra96 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(3);
        insert(e);
    }

rule regra97 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50;
        e.getAcumulado24h() >= 100 && e.getAcumulado24h() < 150;
        e.getAcumulado168h() >= 210;
        e.getPrevisao1h() >= 80;
    actions
        e.setValorChuva(4);
        insert(e);
    }

rule regra98 {

```

```

    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 150;
        e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorChuva(2);
        insert(e);
    }

rule regra99 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 150;
        e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(3);
        insert(e);
    }

rule regra100 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(3);
        insert(e);
    }

rule regra101 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
150;
        e.getPrevisao1h() >= 80;
    actions
        e.setValorChuva(3);
        insert(e);
    }

rule regra102 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;

```

```

150;         e.getAcumulado168h() >= 100 && e.getAcumulado168h() <
            e.getPrevisao1h() >= 80;
        actions
            e.setValorChuva(3);
            insert(e);
    }

rule regra103 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(3);
        insert(e);
}

rule regra104 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 100 && e.getAcumulado1h() < 210;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getAcumulado168h() >= 210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(3);
        insert(e);
}

rule regra105 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 210;
        e.getPrevisao1h() >= 0 && e.getPrevisao1h() < 40;
    actions
        e.setValorChuva(3);
        insert(e);
}

rule regra106 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 210;
        e.getAcumulado24h() >= 0 && e.getAcumulado24h() < 50;
        e.getAcumulado168h() >= 210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(4);
        insert(e);
}

rule regra107 {

```

```

    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(4);
        insert(e);
}

rule regra108 {
    declarations
        Escorregamento e;
    conditions
        e.getAcumulado1h() >= 50 && e.getAcumulado1h() < 150;
        e.getAcumulado24h() >= 50 && e.getAcumulado24h() < 100;
        e.getAcumulado168h() >= 150 && e.getAcumulado168h() <
210;
        e.getPrevisao1h() >= 40 && e.getPrevisao1h() < 80;
    actions
        e.setValorChuva(3);
        insert(e);
}

rule regra109 {
    declarations
        Escorregamento e;
    conditions
        e.getPopulacao() >= 0;
        e.getGeotecnia() != null;
        e.getDeclividade() >= 0;
        e.getVegetacao() != null;
        e.getCobertura_vegetal() != null;
    actions
        e.setSuscetibilidade(e.suscetibilidade());
}
}

```

## APÊNDICE B – Detalhamento dos casos de uso

Neste apêndice são descritos os casos de uso da ferramenta previstos no diagrama da seção 3.2.2. O Quadro 22 apresenta o caso de uso Consultar Resultados Anteriores.

Quadro 22 – Caso de uso Consultar Resultados Anteriores

<p><b>UC01 – Consultar Resultados Anteriores</b></p> <p><b>Ator:</b> Usuário</p> <p><b>Objetivo:</b> Permite ao usuário consultar os resultados obtidos anteriormente na realização do questionário.</p> <p><b>Pré-Condição:</b> Usuário realiza o questionário.</p> <p><b>Pós-Condição:</b> Usuário visualiza os resultados de todos os questionários realizados.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Usuário clica no submenu Resultados Anteriores</li> <li>2. Sistema mostra os resultados de todos os questionários</li> </ol> <p><b>Cenário de Exceção:</b></p> <p>No passo 2, caso o usuário não tenha feito algum questionário ou nenhum:</p> <ol style="list-style-type: none"> <li>2.1 Sistema apresenta a mensagem “Este questionário ainda não foi realizado”</li> <li>2.2 Volta ao cenário principal</li> </ol>
---

No Quadro 23 tem-se o detalhamento do caso de uso Visualizar Regras.

Quadro 23 – Caso de uso Visualizar Regras

<p><b>UC03 – Visualizar Regras</b></p> <p><b>Ator:</b> Usuário</p> <p><b>Objetivo:</b> Permite ao usuário visualizar as regras utilizadas para mostrar o resultado final.</p> <p><b>Pré-Condição:</b> Sistema mostra o resultado final.</p> <p><b>Pós-Condição:</b> Usuário visualiza o histórico de regras.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Sistema mostra resultado final</li> <li>2. Usuário clica no botão Histórico de Regras</li> <li>3. Sistema mostra as regras utilizadas</li> </ol> <p><b>Cenário Alternativo:</b></p> <p>No passo 3, o usuário deseja voltar a tela de resultado:</p> <ol style="list-style-type: none"> <li>3.1 Usuário clica no botão Voltar</li> <li>3.2 Volta ao cenário principal</li> </ol>
---

O caso de uso apresentado no Quadro 24, descreve o procedimento de como alterar regras numa base de conhecimento.

Quadro 24 – Caso de uso Alterar Regras na Base de Conhecimento

<p><b>UC04 – Alterar Regras na Base de Conhecimento</b></p> <p><b>Ator:</b> Administrador</p> <p><b>Objetivo:</b> Permite ao administrador alterar regras na base de conhecimento.</p> <p><b>Pré-Condição:</b> Administrador deve ter efetuado <i>login</i> no sistema.</p> <p><b>Pós-Condição:</b> Administrador salva base de conhecimento.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Administrador clica no submenu <code>Abrir Base</code></li> <li>2. Administrador seleciona o arquivo que contém a base de conhecimento</li> <li>3. Sistema abre base de conhecimento na tela</li> <li>4. Administrador altera regra</li> <li>5. Administrador salva arquivo</li> </ol> <p><b>Cenário Alternativo:</b></p> <p>No passo 3, o administrador deseja voltar a tela principal:</p> <ol style="list-style-type: none"> <li>3.1 Administrador clica no botão <code>Voltar</code></li> <li>3.2 Volta a tela principal</li> </ol>
--

O Quadro 25 apresenta o caso de uso Visualizar Tabela de Risco.

Quadro 25 – Caso de uso Visualizar Tabela de Risco

<p><b>UC05 – Visualizar Tabela de Risco</b></p> <p><b>Ator:</b> Usuário</p> <p><b>Objetivo:</b> Permite ao usuário visualizar a tabela de valores usada para calcular os riscos de enchentes e escorregamentos.</p> <p><b>Pré-Condição:</b> Usuário responde o questionário.</p> <p><b>Pós-Condição:</b> Usuário visualiza a tabela de risco.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Sistema mostra resultado final</li> <li>2. Usuário clica no botão <code>Tabela de Valores</code></li> <li>3. Sistema mostra a tabela de risco de acordo com o questionário selecionado</li> </ol> <p><b>Cenário Alternativo:</b></p> <p>No passo 3, o usuário deseja voltar a tela de resultado:</p> <ol style="list-style-type: none"> <li>3.1 Usuário clica no botão <code>Voltar</code></li> <li>3.2 Volta ao cenário principal</li> </ol>
--

O Quadro 26 apresenta o caso de uso Abrir Base de conhecimento.

Quadro 26 – Caso de uso Abrir Base de conhecimento

<p><b>UC06 – Abrir Base de Conhecimento</b></p> <p><b>Ator:</b> Administrador</p> <p><b>Objetivo:</b> Permite ao administrador abrir uma base de conhecimento.</p> <p><b>Pré-Condição:</b> Administrador deve ter efetuado <i>login</i> no sistema.</p> <p><b>Pós-Condição:</b> Administrador visualiza a base de conhecimento.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Administrador clica no submenu <i>Abrir Base</i></li> <li>2. Sistema abre tela para selecionar arquivo onde consta a base de conhecimento</li> <li>3. Administrador visualiza a base</li> </ol> <p><b>Cenário Alternativo:</b></p> <p>No passo 3, o administrador deseja voltar a tela principal:</p> <ol style="list-style-type: none"> <li>3.1 Administrador clica no botão <i>Voltar</i></li> <li>3.2 Volta a tela principal</li> </ol> <p><b>Cenário Alternativo:</b></p> <p>No passo 3, caso o administrador altere uma regra:</p> <ol style="list-style-type: none"> <li>3.1 Administrador altera regra</li> <li>3.2 Administrador clica no submenu <i>Salvar</i></li> <li>3.3 Sistema mostra mensagem “Arquivo salvo com sucesso”</li> <li>3.4 Volta ao cenário principal</li> </ol>
--

O caso de uso apresentado no Quadro 27 descreve o procedimento de como criar uma nova base de conhecimento no sistema.

Quadro 27 – Caso de uso Criar Nova Base de Conhecimento

<p><b>UC07 – Criar Nova Base de Conhecimento</b></p> <p><b>Ator:</b> Administrador</p> <p><b>Objetivo:</b> Permite ao administrador criar uma nova base de conhecimento.</p> <p><b>Pré-Condição:</b> Administrador deve ter efetuado <i>login</i> no sistema.</p> <p><b>Pós-Condição:</b> Administrador salva base de conhecimento.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Administrador clica no submenu <i>Nova Base</i></li> <li>2. Sistema abre tela para criação da base</li> <li>3. Administrador cria nova base</li> <li>4. Administrador salva arquivo</li> </ol> <p><b>Cenário Alternativo:</b></p> <p>No passo 3, o administrador deseja voltar a tela principal:</p> <ol style="list-style-type: none"> <li>3.1 Administrador clica no botão <i>Voltar</i></li> <li>3.2 Volta a tela principal</li> </ol>
--

O Quadro 28 apresenta o caso de uso Efetuar *login* descrevendo o processo de autenticação do ator *Administrador*.

Quadro 28 – Caso de uso Efetuar *login*

<p><b>UC08 – Efetuar <i>login</i></b>  <b>Ator:</b> Administrador  <b>Objetivo:</b> Permite ao administrador através da identificação por senha ter acesso às funcionalidades do sistema.  <b>Pré-Condição:</b> Administrador efetua <i>login</i> no sistema.  <b>Pós-Condição:</b> Administrador está logado no sistema.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Administrador preenche o campo senha</li> <li>2. Sistema valida a senha</li> <li>3. Sistema habilita menus</li> </ol> <p><b>Cenário de Exceção:</b>  No passo 2, caso a senha esteja incorreta:</p> <ol style="list-style-type: none"> <li>2.1 Sistema apresenta a mensagem “Senha incorreta”</li> <li>2.2 Volta ao cenário principal</li> </ol>
---

No Quadro 29 tem-se o detalhamento do caso de uso Consultar Informações dos questionários.

Quadro 29 – Caso de uso Consultar Informações

<p><b>UC09 – Consultar Informações</b>  <b>Ator:</b> Usuário  <b>Objetivo:</b> Permite ao usuário consultar informações sobre os questionários.  <b>Pré-Condição:</b> Usuário seleciona o questionário.  <b>Pós-Condição:</b> Usuário clica no botão <i>Informações</i>.</p> <p><b>Cenário Principal:</b></p> <ol style="list-style-type: none"> <li>1. Usuário clica no submenu <i>Responder Questionário</i></li> <li>2. Usuário escolhe um questionário</li> <li>3. Usuário clica no botão <i>Informações</i></li> <li>4. Sistema mostra as informações do questionário</li> </ol> <p><b>Cenário Alternativo:</b>  No passo 4, o usuário deseja voltar a tela do questionário:</p> <ol style="list-style-type: none"> <li>4.1 Usuário clica no botão <i>Voltar</i></li> <li>4.2 Volta ao cenário principal</li> </ol>
--