



PROGRAMA DE PESQUISA E DESENVOLVIMENTO TECNOLÓGICO
CICLO 2003/2004

Nº DO CONTRATO: 4500052325

Nº DO PROJETO: 128

INÍCIO: 28/09/2005

DURAÇÃO: 24 meses

A. Instituição executora:

B. Título do Projeto:

Modernização da Área de Automação de Processos das Usinas Hidroelétricas de Balbina e Samuel

C. Coordenador do Projeto:

Nome:	Alberto José Álvares				
Fone:	(61)-3307-2314	Cel:	(61)-99679435	E-Mail:	alvares@AlvaresTech.com

RELATÓRIO DE PRODUTOS GERADOS: ETAPA 3 – Ano1

ETAPA 3 – ANO 1: CONSTRUÇÃO MODELOS DE COMPONENTES PARA O SISTEMA DE TOMADA DE DECISÃO BASEADO EM 7 CAMADAS

SISTEMA I-KERNEL: UM KERNEL INTELIGENTE PARA O SIMPREBAL SISTEMA DE MANUTENÇÃO PREDITIVA DE BALBINA

(ESPECIFICAÇÃO PRELIMINAR)



Motivação

Este texto se destina a documentar a Especificação do Sub-sistema I-Kernel, um kernel de sistema inteligente a ser desenvolvido em Java, para dar suporte ao desenvolvimento do SIMPREBAL: Sistema Inteligente de Manutenção Preditiva de Balbina (Alvares, 2006).

O sistema SIMPREBAL vem sendo desenvolvido dentro do programa de pesquisa e desenvolvimento tecnológico da Eletronorte, por meio do contrato 4500052325 – projeto 128, e tem como responsável técnico o Prof. Alberto Álvares da UNB.

O sub-sistema I-Kernel vem a ser um dos componentes fundamentais do SIMPREBAL, sendo responsável pela captura dos sinais de monitoramento dos equipamentos, seu processamento inteligente e realimentação do banco de dados com recomendações de manutenção.

Neste documento, apresenta-se a especificação do Sub-sistema I-Kernel.

Compreensão do Problema

O sistema SIMPREBAL, atualmente em desenvolvimento, visa dar suporte às atividades de manutenção da Usina Hidroelétrica de Balbina, gerenciada pela Manaus Energia, utilizando-se de técnicas de sistemas inteligentes para detectar situações de manutenção preditiva e alertar o usuário de potenciais falhas em iminência de acontecer, de tal forma que atividades de manutenção dos equipamentos possam ser consideradas antes que tais falhas aconteçam de fato. A concepção do SIMPREBAL é fundamentada na idéia de manutenção centrada em confiabilidade, a partir de grandezas monitoradas pelo sistema de supervisão e controle da Usina.

Os sinais adquiridos dos equipamentos e do sistema de supervisão e controle da usina devem passar por um processamento que envolve sete camadas (1-sensor, 2-processamento de sinal, 3-monitoração de condição, 4-avaliação de saúde, 5-prognósticos, 6-tomada de decisão e 7-apresentação), de acordo com (Álvares, 2006). Parte deste processamento (camadas 1 e 2, e também parte das camadas 3 e 4), serão processadas em equipamentos de aquisição (Fieldbus, Rockwell) e sistemas de aquisição (Assetview/SMAR). A outra parte do processamento (parte das camadas 3 e 4, além das camadas 5, 6 e 7) deverão ser processadas dentro do SIMPREBAL.

Desta forma, para que o SIMPREBAL possa ser construído, vislumbrou-se a necessidade de um kernel de sistemas inteligentes, capaz de capturar os dados do sistema de supervisão (seja a partir do banco de dados do sistema, seja diretamente dos equipamentos sendo monitorados), processar estes dados utilizando técnicas de sistemas inteligentes, e atualizar os bancos de dados com os dados resultantes desse processamento. Este kernel, deveria ser capaz de acessar diretamente os dados, independente de sua origem e disponibilizá-los na forma de variáveis. Essas variáveis, devem poder ser processadas por um conjunto de regras (convencionais ou fuzzy) e/ou por redes neurais. Este processamento deve ser possível em N camadas, de tal forma a atender a arquitetura OSA-CBM que orienta o SIMPREBAL. O resultado desse processamento deve ser armazenado novamente no banco de dados, e daí poderem ser utilizados para gerar alertas aos controladores do sistema da Usina, na forma de mensagens “popup”, mensagens de e-mail ou algum outro tipo de alerta visual. Da mesma maneira, devem ser armazenados na forma de registros perenes, para compor “logs” dos eventos diagnosticados pelo sistema, funcionando como registros históricos desses diagnósticos, que poderão ser aproveitados futuramente para dotar o sistema de aprendizagem (seja por meio de redes neurais ou não).

Proposta de Solução do Problema

Para promover a solução do problema levantado acima, propõe-se o desenvolvimento de um sub-sistema de aquisição de dados e processamento inteligente que designaremos aqui como I-Kernel. Apesar do sub-sistema I-Kernel visar um apoio à construção do SIMPREBAL, sua concepção é a de um sub-sistema genérico, a ser desenvolvido em linguagem Java, capaz de realizar a aquisição de

dados a partir de bancos de dados e equipamentos acessíveis via OPC (OLE for Process Control), e seu processamento por meio de regras convencionais, regras fuzzy e redes neurais, em N camadas.

De modo a tornar o desenvolvimento do I-Kernel uma atividade mais rápida, ao invés de se desenvolver todo o sistema a partir do início, nossa proposta é a de se integrar diversos componentes já existentes, promovendo um desenvolvimento orientado ao reuso. Da mesma forma, o I-Kernel, dada sua concepção na forma de um componente de software, poderá ser reutilizado em projetos futuros que demandem soluções de processamento inteligentes envolvendo o processamento de regras, regras fuzzy ou redes neurais.

Dentre os componentes que se almeja integrar, estão as ferramentas Jess – Java Expert System Shell, desenvolvido pelo Sandia National Laboratories, para promover o processamento de regras compondo um sistema baseado em regras, o Fuzzy-Jess, desenvolvido pelo National Research Council do Canada's Institute for Information Technology, para o processamento de regras fuzzy, de modo análogo ao Jess, o JDBC para acesso aos bancos de dados e o JNI para o acesso aos servidores OPC que dão acesso direto aos equipamentos. O módulo de redes neurais deverá ser desenvolvido a partir das próprias equações e modelos dos diferentes tipos de redes neurais existentes na literatura científica.

Uma visão geral desta proposta, modelada em termos de um diagrama UML de subsistemas pode ser observada na figura 1 abaixo:

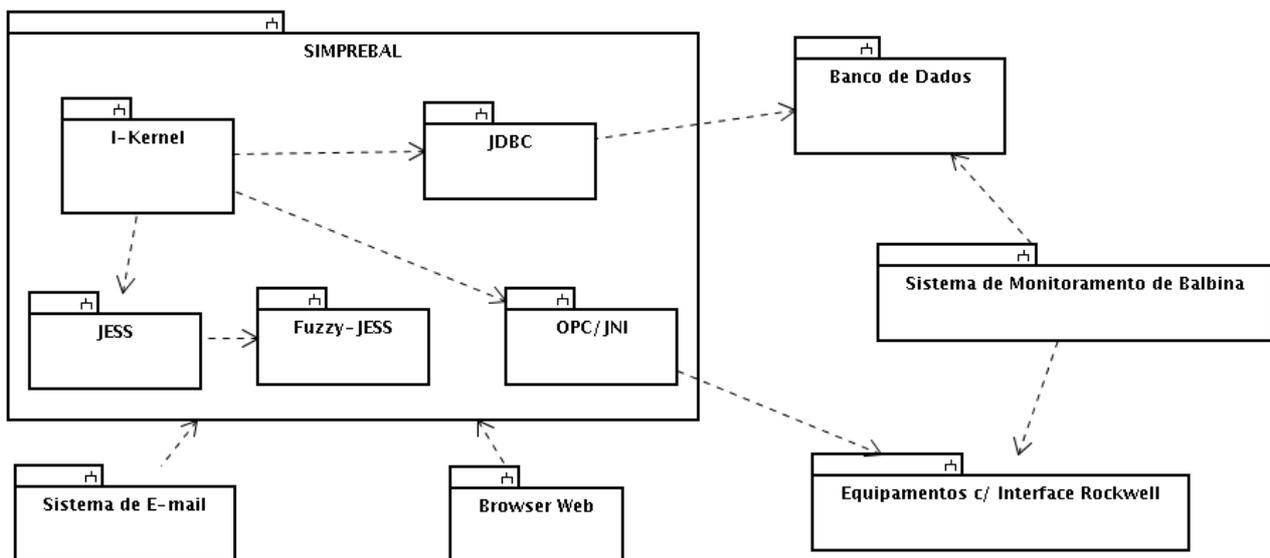


Figura 1: Visão Geral do Projeto SIMPREBAL, demonstrando o I-Kernel e sua interação com outros módulos

De acordo com (Sommerville, 2003), uma maneira útil de se estruturar os requisitos é dividi-los entre requisitos de usuário e requisitos de sistema. Os requisitos de usuário devem capturar as demandas mais gerais dos usuários do sistema. Normalmente, estes requisitos são descritos em linguagem natural e não devem entrar em detalhes mais técnicos. Os requisitos de sistema são descrições mais detalhadas dos requisitos de usuário. Neste documento, estaremos adotando este formato recomendado por Sommerville, listando os requisitos de usuário na forma de uma lista de requisitos em linguagem natural, e os requisitos de usuário serão detalhados de maneira mais técnica, utilizando-se diagramas UML. A metodologia de desenvolvimento a ser utilizada é uma adaptação do processo Unificado de modelagem (Jacobson et.al. 1999).

Requisitos de Usuário

Requisitos Funcionais

Requisito FU1: O sistema deve acessar os dados da Usina de Balbina a partir dos Bancos de Dados utilizados pelo Sistema de Monitoramento de Balbina, OU diretamente dos equipamentos de controle, por meio de um servidor OPC que disponibiliza as informações on-line dos equipamentos.

Requisito FU2: O sistema deve processar esses dados nas seguintes formas:

- Na forma de um sistema especialista baseado em regras
- Na forma de um sistema de regras fuzzy
- Na forma de redes neurais

Requisito FU3: O sistema deve alertar o usuário por meio de mensagens de e-mail quando possíveis falhas puderem ser diagnosticadas.

Requisito FU4: O sistema deve alertar o usuário por meio de um alerta visual, quando possíveis falhas puderem ser diagnosticadas.

Requisito FU5: O sistema deve propiciar a edição de um sinótico contendo certo conjunto de variáveis sendo monitoradas, escolhidas pelo usuário e compondo uma possível tela de apresentação.

Requisito FU6: O sistema deve exibir o valor on-line das variáveis sendo monitoradas que foram selecionadas para compor um determinado sinótico, apresentando-as em uma tela própria previamente desenvolvida.

Requisito FU7: O sistema deve implementar algum mecanismo de aprendizagem, de tal forma que o histórico de falhas e defeitos anteriores possa ser utilizado para prevenir o surgimento de novas falhas.

Requisito FU8: O processamento das informações se dará na forma de um ciclo operacional fechado, que seguirá a seguinte sequência:

- 1 . Verificação dos Dados a serem adquiridos
- 2 . Aquisição de Dados do Banco de Dados
- 3 . Aquisição de Dados via OPC
- 4 . Armazenamento provisório de todos os dados em variáveis do JESS
- 5 . Para cada uma de N camadas possíveis de processamento
 - 5.1 Processamento das regras via JESS
 - 5.2 Processamento das Regras Fuzzy, via Fuzzy-JESS
 - 5.3 Processamento dos dados via Redes Neurais
- 6 . Atualização dos Dados no Banco de Dados
- 7 . Atualização dos Dados via OPC.

Requisitos Não-Funcionais

Requisito NFU1: O sistema deve ser desenvolvido na linguagem Java.

Requisito NFU2: As regras do sistema não devem ser armazenadas diretamente em código-fonte, mas devem ser editáveis e estar disponíveis externamente em um arquivo modificável.

Requisito NFU3: O sistema deve possuir uma interface web de acesso, por meio da qual seja possível ao usuário configurar o sistema, editar regras e parâmetros e monitorar as variáveis sendo processadas pelo sistema.

Requisito NFU4: O sistema deve ser conectável a bancos de dados SQL genéricos, desde que exista um driver JDBC para o respectivo banco de dados.

Requisito NFU5: Para processar as regras na forma de sistemas especialistas, o sistema deve utilizar o pacote JESS.

Requisito NFU6: Para processar as regras fuzzy, o sistema deve utilizar o pacote Fuzzy-JESS.

Requisito NFU7: O sistema deve ser concebido de tal forma que as regras clássicas, regras fuzzy e redes neurais possam ser usadas de modo intercambiável para cada uma das camadas de processamento do SIMPREBAL (3-monitoração de condição, 4-avaliação de saúde, 5-prognósticos, 6-tomada de decisão)

Requisitos de Sistema

Seguindo o processo de desenvolvimento Unificado (Jacobson et.al. 1999), os requisitos do sistema devem ser especificados na forma de casos de uso. Na figura 2, a seguir, temos um diagrama UML de casos de uso, que nos propicia uma visão geral do sistema, segundo a perspectiva do sistema.

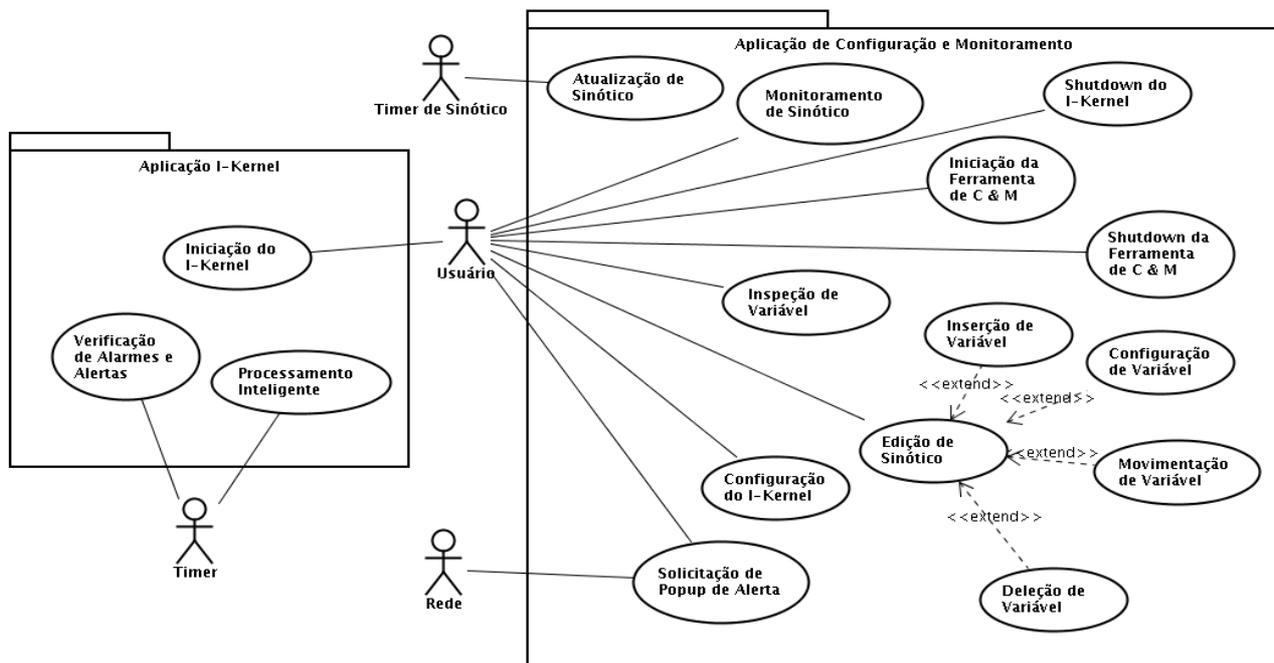


Figura 2 – Casos de Uso do Sistema

Detalhamento dos Casos de Uso

O sistema é composto de duas aplicações que são executadas de maneira independente:

- A aplicação I-Kernel
- A aplicação de Configuração e Monitoramento

A aplicação I-Kernel é um servidor que não possui interface com o usuário. Desta forma, a única coisa que o usuário pode fazer é iniciar a aplicação. Além disso, um timer previamente programado envia “ticks” periódicos que iniciam um ciclo operacional de processamento inteligente e da mesma forma fazem a verificação de alarmes e alertas. A aplicação I-Kernel está detalhada nos casos de uso de 1 a 3.

A aplicação de Configuração e Monitoramento é uma aplicação web, destinada a promover a configuração de parâmetros do I-Kernel, bem como o monitoramento das variáveis sob controle do sistema e a solicitação de shutdown do I-Kernel. Este monitoramento pode ser um monitoramento direto do estado das variáveis, selecionando-se a variável dentre todas aquelas disponíveis, ou pode ser um monitoramento por sinótico. No monitoramento direto de variáveis, escolhe-se a variável a ser

monitorada, e o sistema exibe seu estado diretamente. Este tipo de monitoramento visa efetuar uma inspeção isolada no estado de alguma variável do sistema. No monitoramento por sinótico, é necessário antes a edição de um sinótico, por meio da configuração de uma tela com um conjunto de variáveis que se deseja monitorar, e uma vez que um sinótico tenha sido editado, pode-se proceder ao monitoramento de sinótico. Além destes casos de uso, pode surgir uma solicitação via rede, normalmente originada pela aplicação I-Kernel, para que um popup de alarme/alerta seja exibido para o usuário. Essa solicitação de popup normalmente é gerada a partir do caso de uso de verificação de alarmes e alertas do I-Kernel. A aplicação de Configuração e Monitoramento encontra-se detalhada por meio dos casos de uso de 4 a 13.

Caso de Uso 1: Iniciação do I-Kernel

Neste caso de uso, o usuário solicita ao sistema operacional que a aplicação I-Kernel seja carregada, e o sistema operacional carrega a aplicação e a inicia. Um detalhamento destas operações encontra-se na figura 3 a seguir:

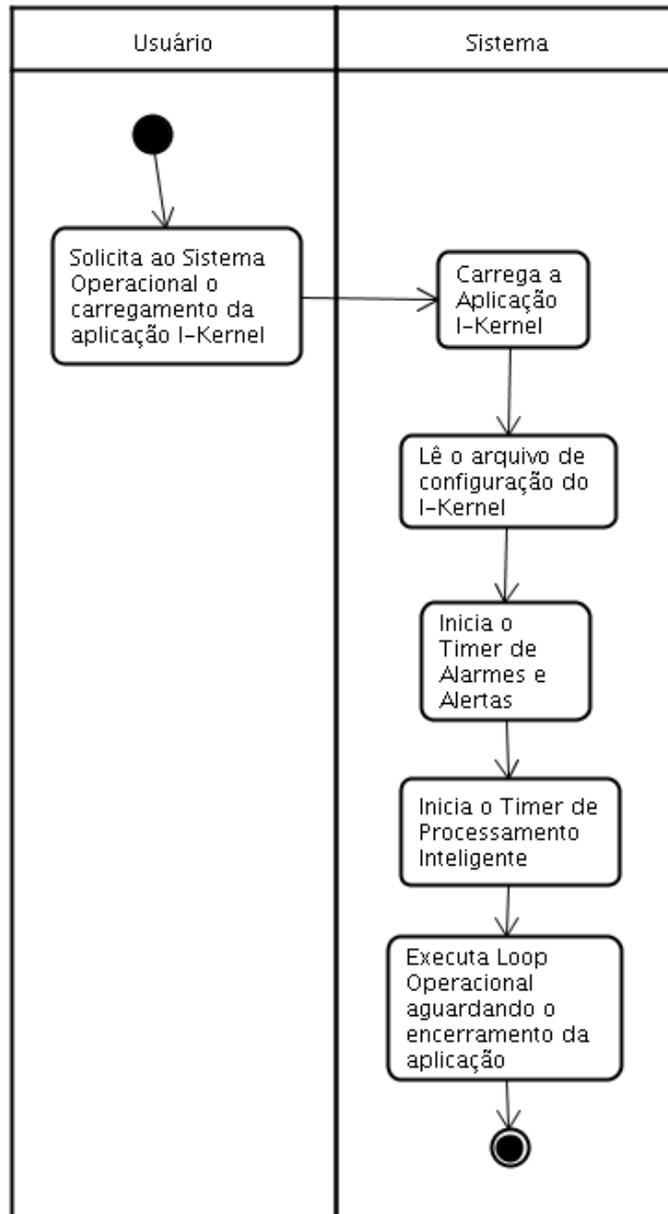


Figura 3 – Caso de Uso: Iniciação do Sistema

Caso de Uso 2: Processamento Inteligente

Nesse caso de uso, o Timer de Processamento Inteligente iniciado durante o caso de uso 1 envia “ticks” de relógio periódicos, iniciando um ciclo de processamento inteligente. Esse caso de uso se encontra detalhado na figura 4 a seguir:

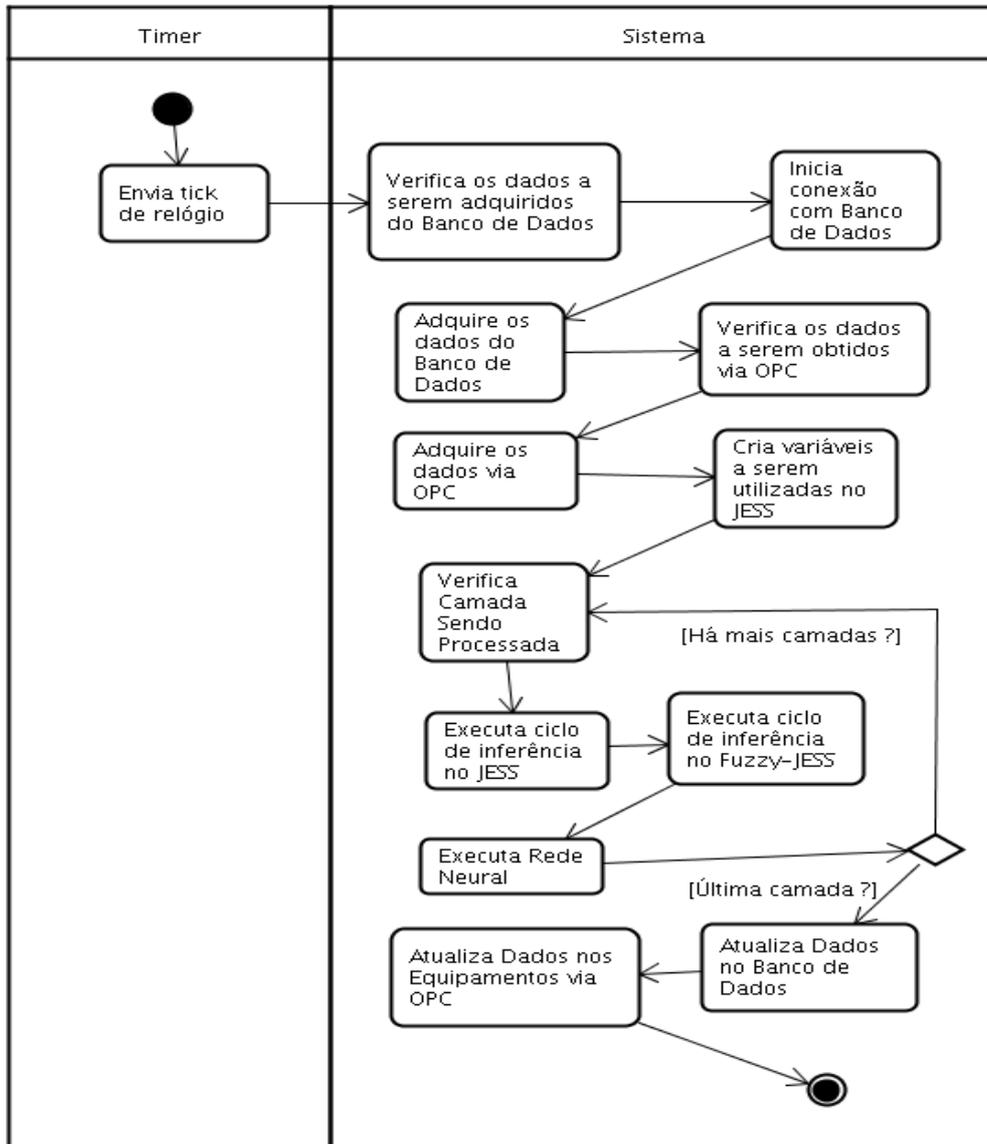


Figura 4 – Caso de Uso: Processamento Inteligente

Caso de Uso 3: Verificação de Alarmes e Alertas

Neste caso de uso, o Timer de Verificação de Alarmes e Alertas envia um tick de relógio, que procede à verificação de um conjunto de variáveis sendo monitoradas, e caso alguma delas esteja marcada, procede ao tipo de alarme ou alerta associado.

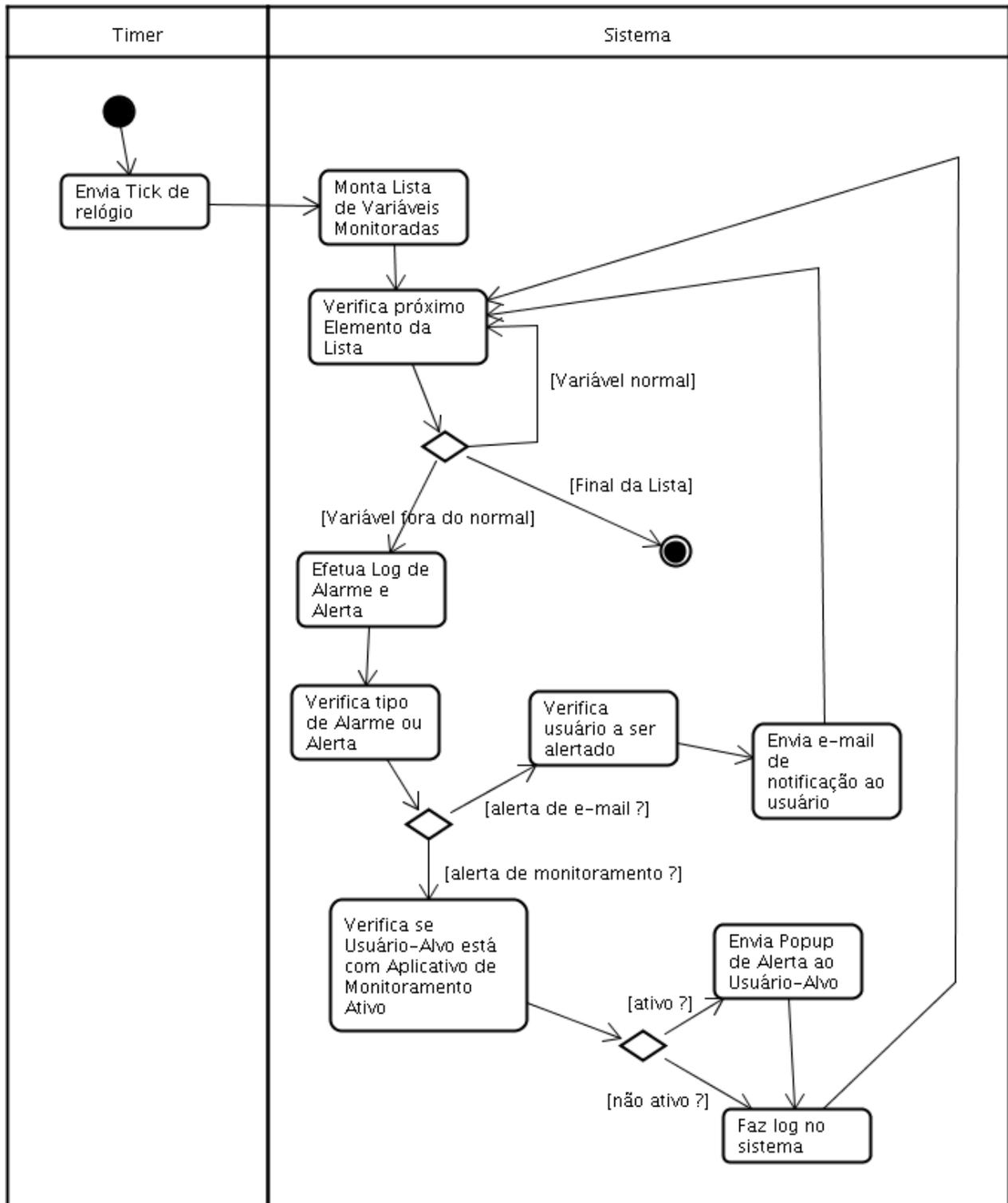


Figura 5 – Caso de Uso: Verificação de Alarmes e Alertas

Cada variável sendo acessada pelo I-Kernel pode ser marcada como uma variável a gerar alerta por e-

mail ou alerta de monitoramento. No alerta por e-mail o sistema deve especificar o endereço de e-mail para o qual uma mensagem, previamente armazenada em um template deve ser enviada. No alerta por monitoramento, o sistema deve especificar um usuário alvo (específico ou qualquer usuário), e levantar a lista de usuários monitorando o sistema, e gerar uma solicitação de popup de alerta com uma mensagem previamente armazenada em um template.

Caso de Uso 4: Shutdown do I-Kernel

Neste caso de uso, uma vez que o sistema de C & M tenha sido iniciado, o usuário solicita o encerramento do sistema I-Kernel. O sistema deve verificar se o I-Kernel está de fato operacional, e se estiver deve encerrar sua operação. Caso o sistema não esteja operacional, deve alertar o usuário de que ele não está operacional.

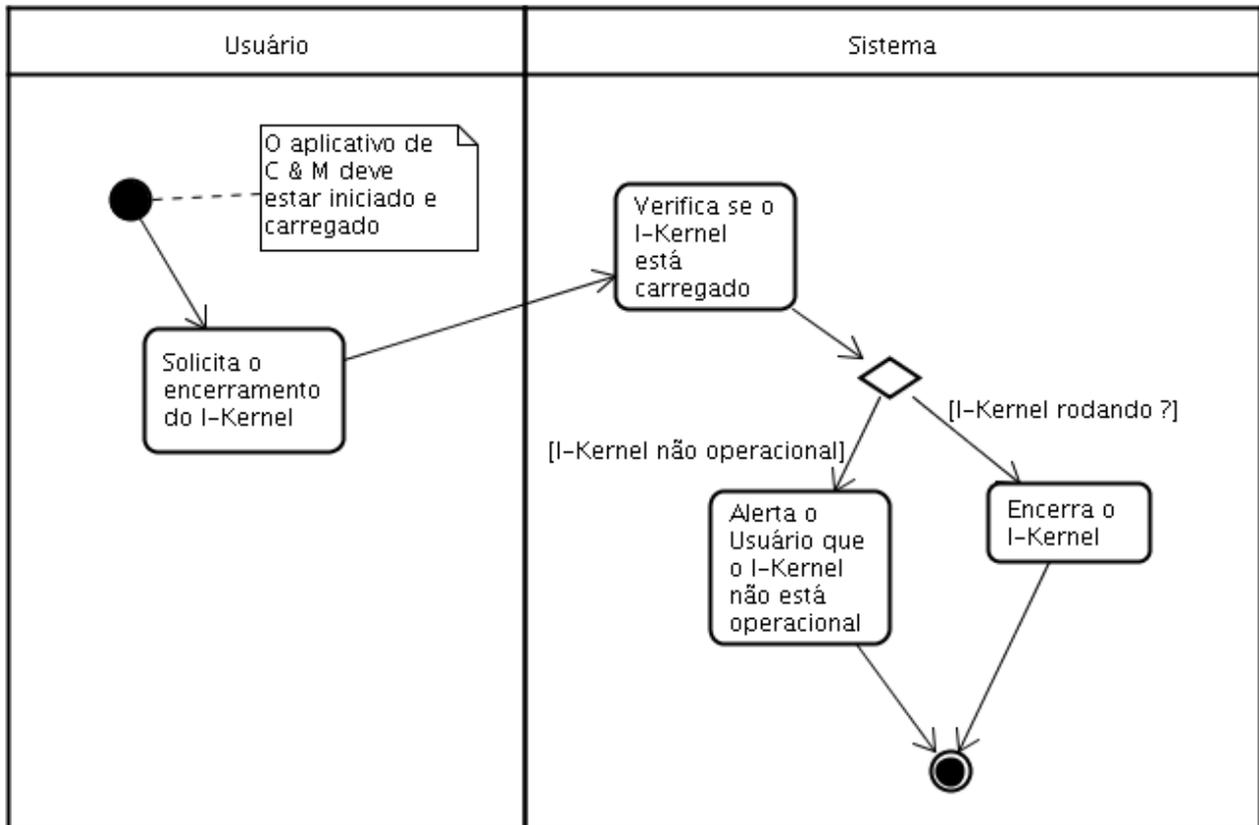


Figura 6 – Caso de Uso: Shutdown do I-Kernel

Caso de Uso 5: Iniciação da Ferramenta de C & M

Neste caso de uso, o Usuário carrega em um browser Web uma URL que corresponde ao endereço da Ferramenta de Configuração e Monitoramento. A partir daí, o sistema deve iniciar a ferramenta de C & M e abrir sua janela principal de interação com o usuário.

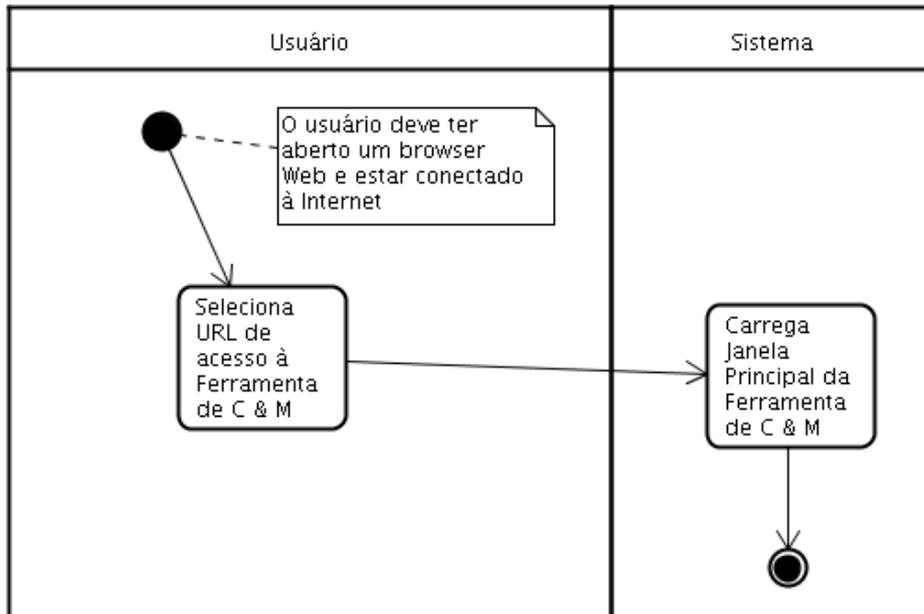


Figura 7 – Caso de Uso: Iniciação da Ferramenta de C & M

Caso de Uso 6: Shutdown da Ferramenta de C & M

Neste caso de uso, uma vez que a Ferramenta de C & M esteja aberta e operacional, o Usuário solicita que a mesma seja encerrada. A partir daí, o sistema deve encerrar a ferramenta.

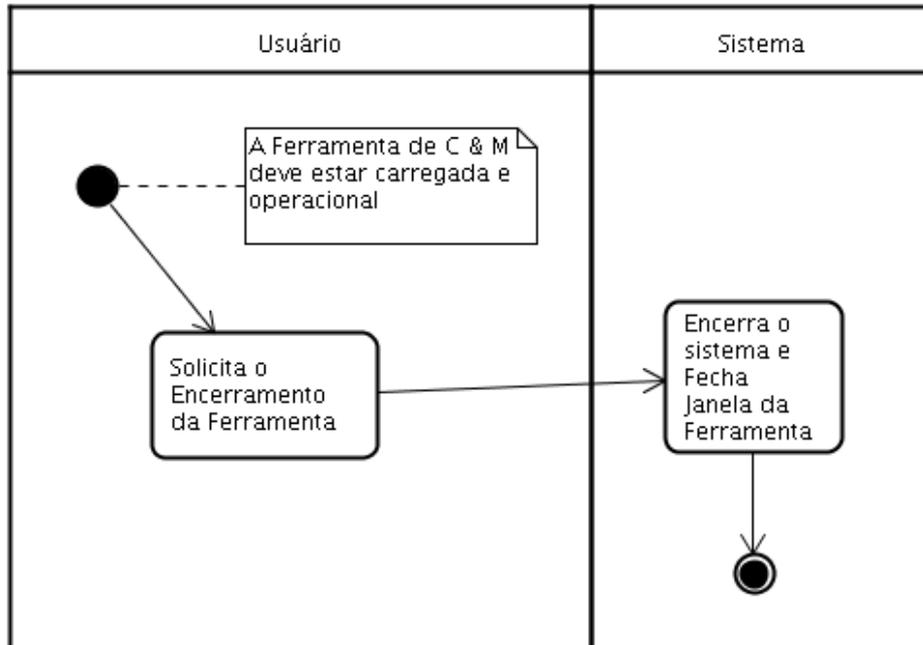


Figura 8 – Caso de Uso: Shutdown da Ferramenta de C & M

Caso de Uso 7: Edição de Sinótico

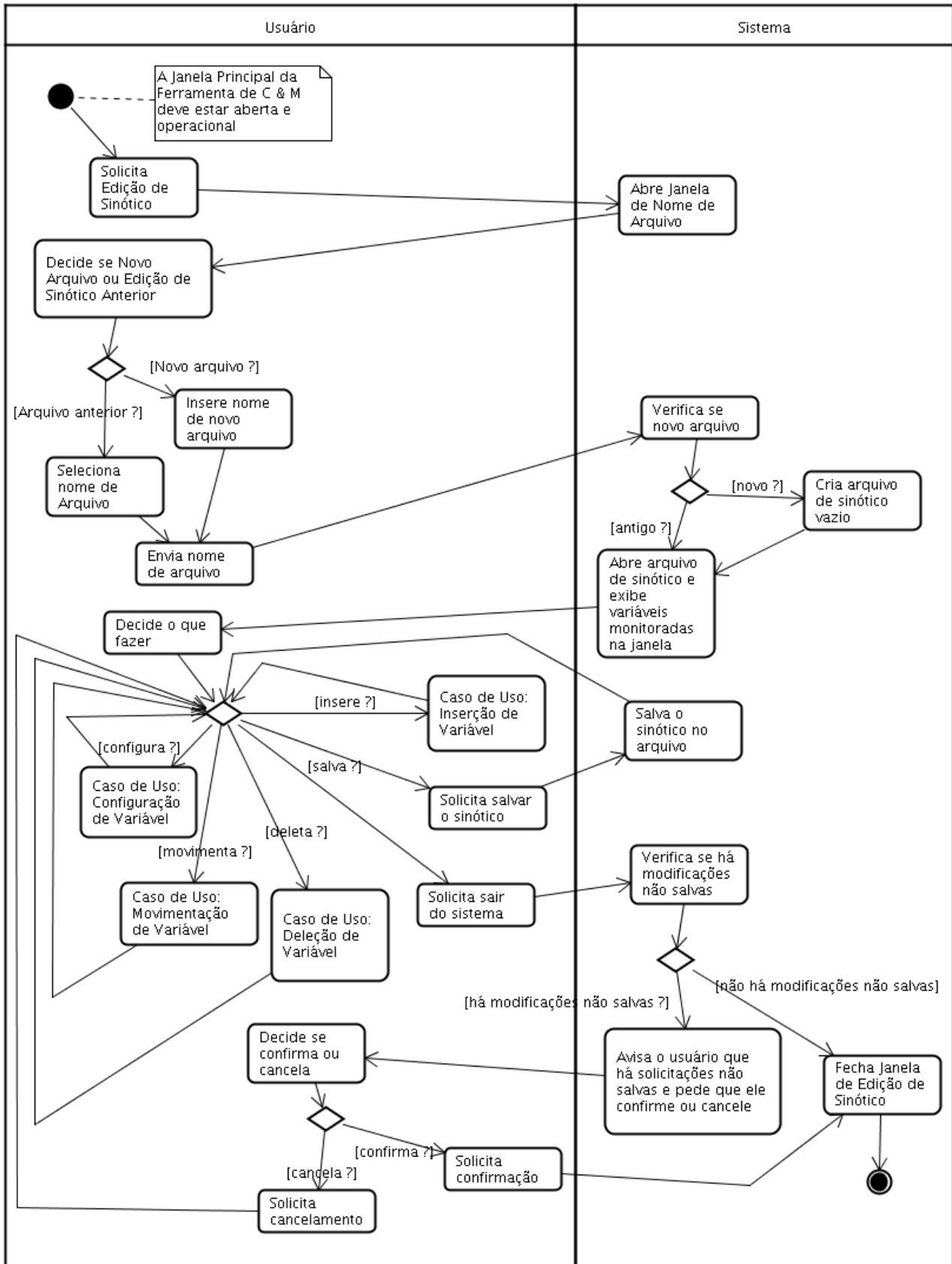


Figura 9 – Caso de Uso: Edição de Sinótico

Este caso de uso é um caso de uso bastante complexo, que se serve de 4 outros casos de uso que estendem sua funcionalidade. É por meio deste caso de uso que o usuário edita um novo sinótico que funcionará como template para o caso de uso de Monitoração de Sinótico.

Basicamente, o usuário informa o nome do sinótico a ser editado (pode ser um novo sinótico ou um sinótico editado anteriormente), e o sistema abre esse sinótico para edição. Com o sinótico aberto, o usuário tem várias opções que podem ser escolhidas em qualquer ordem:

- Inserção de Variável
- Deleção de Variável
- Movimentação de Variável
- Configuração de Variável
- Salvar Sinótico
- Finalizar Edição de Sinótico

Devido a sua complexidade, esse caso de uso prevê 4 pontos de extensão, onde os seguintes casos de uso podem estender as funcionalidades aqui especificadas:

- Caso de Uso: Inserção de Variável
- Caso de Uso: Deleção de Variável
- Caso de Uso: Movimentação de Variável
- Caso de Uso: Configuração de Variável

Caso o usuário tenha feito todas as edições necessárias, ele pode decidir salvar o sinótico e/ou encerrar a edição de sinótico. Os detalhes deste procedimento estão na figura 9 acima.

Caso de Uso 8: Inserção de Variável

Esse caso de uso estende o caso de uso Edição de Sinótico, permitindo que o usuário insira uma nova variável a ser monitorada no sinótico sendo editado. Basicamente o usuário seleciona a inserção de nova variável, posiciona o mouse onde deseja que a mesma seja inserida e clica no mouse, para que a mesma possa ser efetivamente inserida. O sistema insere a nova variável sendo monitorada e redesenha a tela.

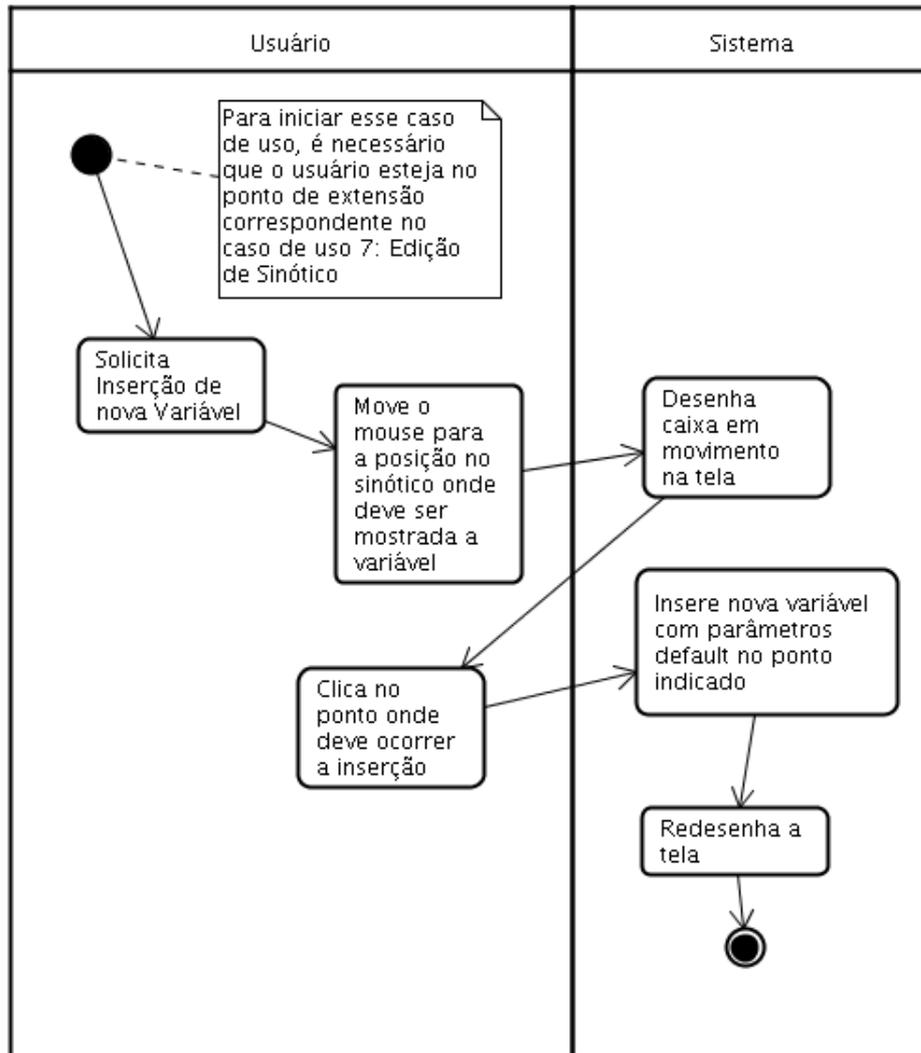


Figura 10 – Caso de Uso: Inserção de Variável

Caso de Uso 9: Deleção de Variável

Este caso de uso estende o caso de uso Edição de Sinótico, permitindo que o usuário delete uma variável sendo monitorada no sinótico sendo editado. Basicamente, o usuário seleciona a variável a ser deletada e pede que ela seja deletada. Ele pode ainda desistir de deletar a variável, e nesse caso o sistema simplesmente volta à situação anterior.

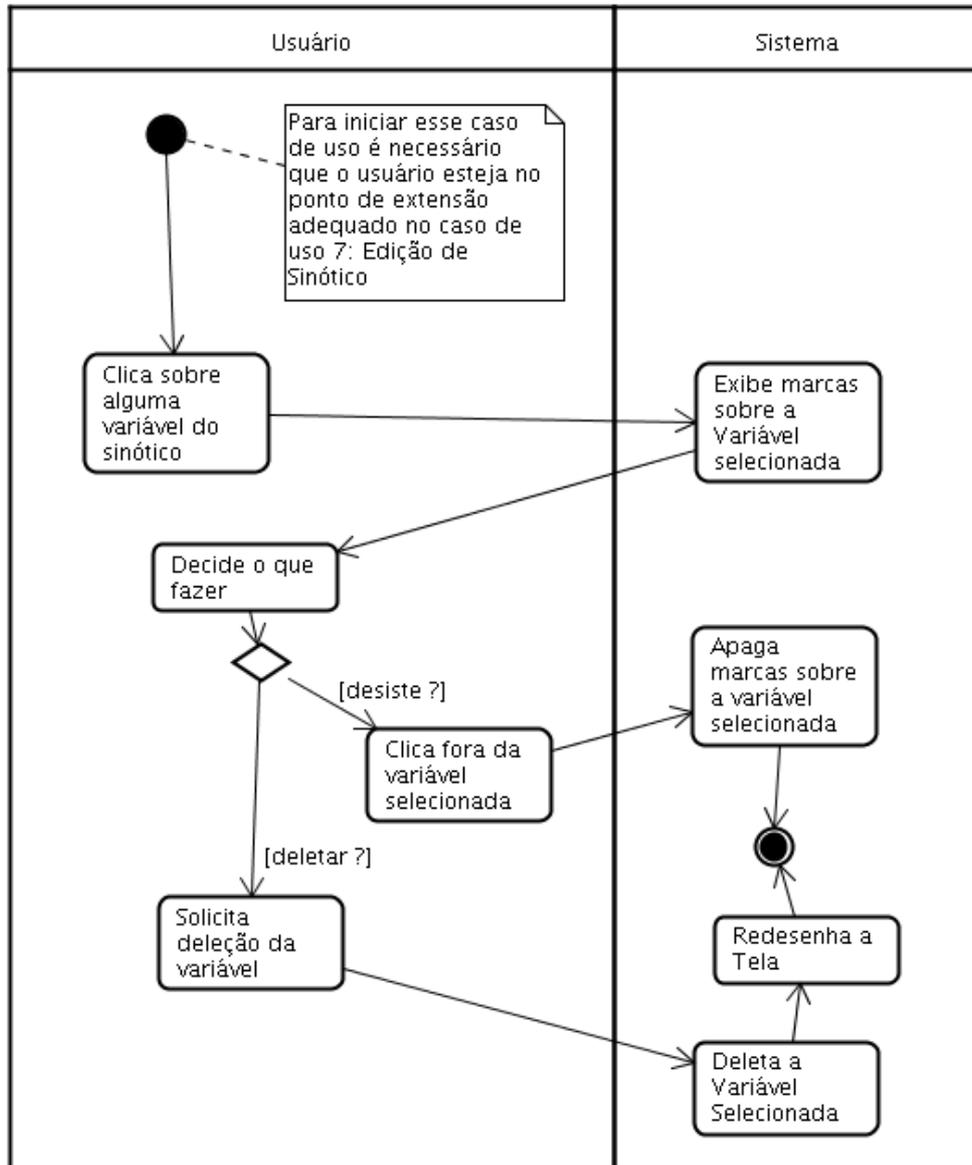


Figura 11: Caso de Uso: Deleção de Variável

Caso de Uso 10: Movimentação de Variável

Este caso de uso estende o caso de uso Edição de Sinótico, permitindo que o usuário movimente uma variável sendo monitorada no sinótico sendo editado. Basicamente, o usuário clica sobre a variável para marcá-la, e depois disso clica nela novamente, sem soltar o botão do mouse e a movimenta para a posição que deseja. Ao soltar o botão do mouse, o sistema movimenta a variável para a posição desejada.

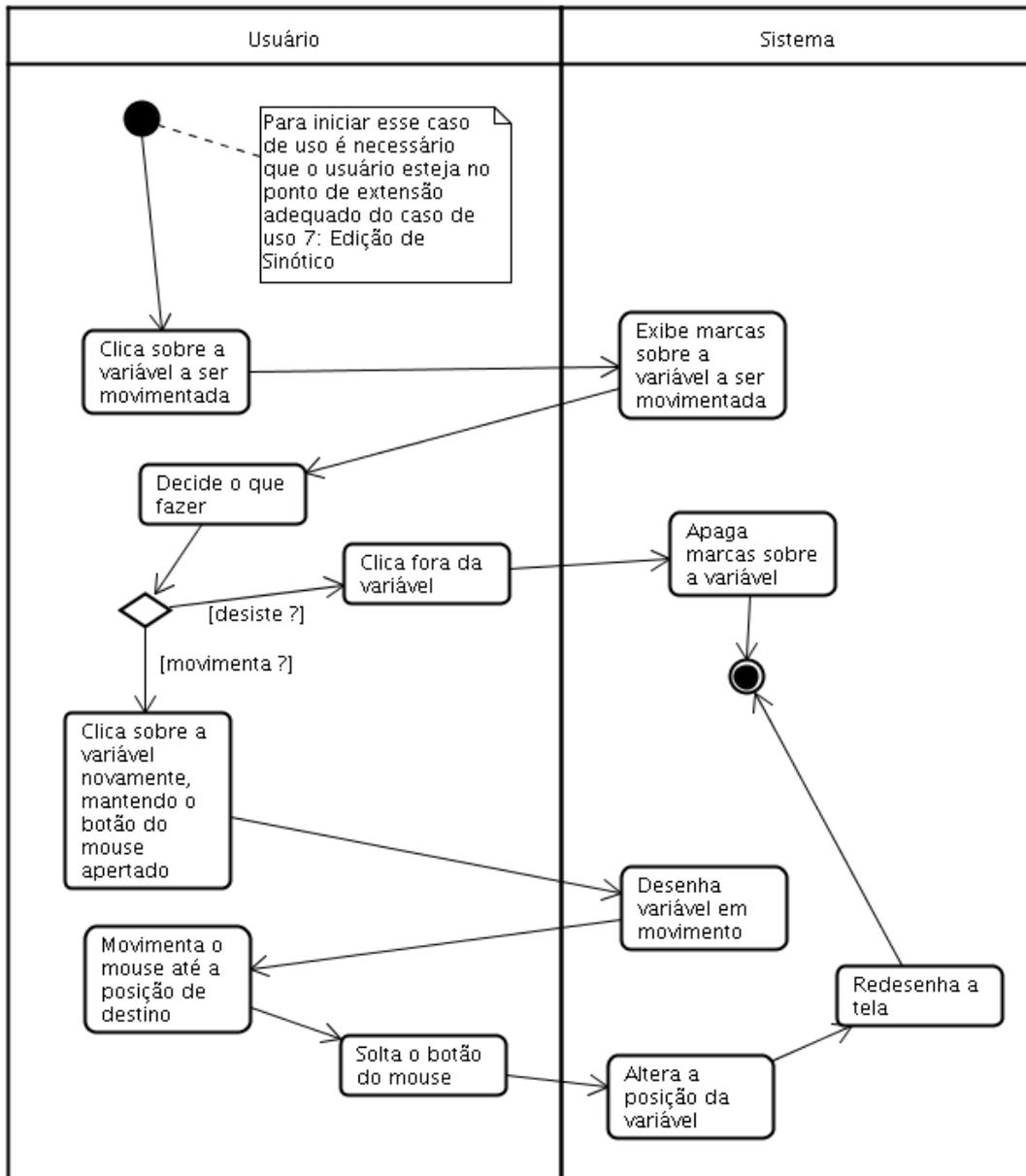


Figura 12 - Caso de Uso: Movimentação de Variável

Caso de Uso 11: Configuração de Variável

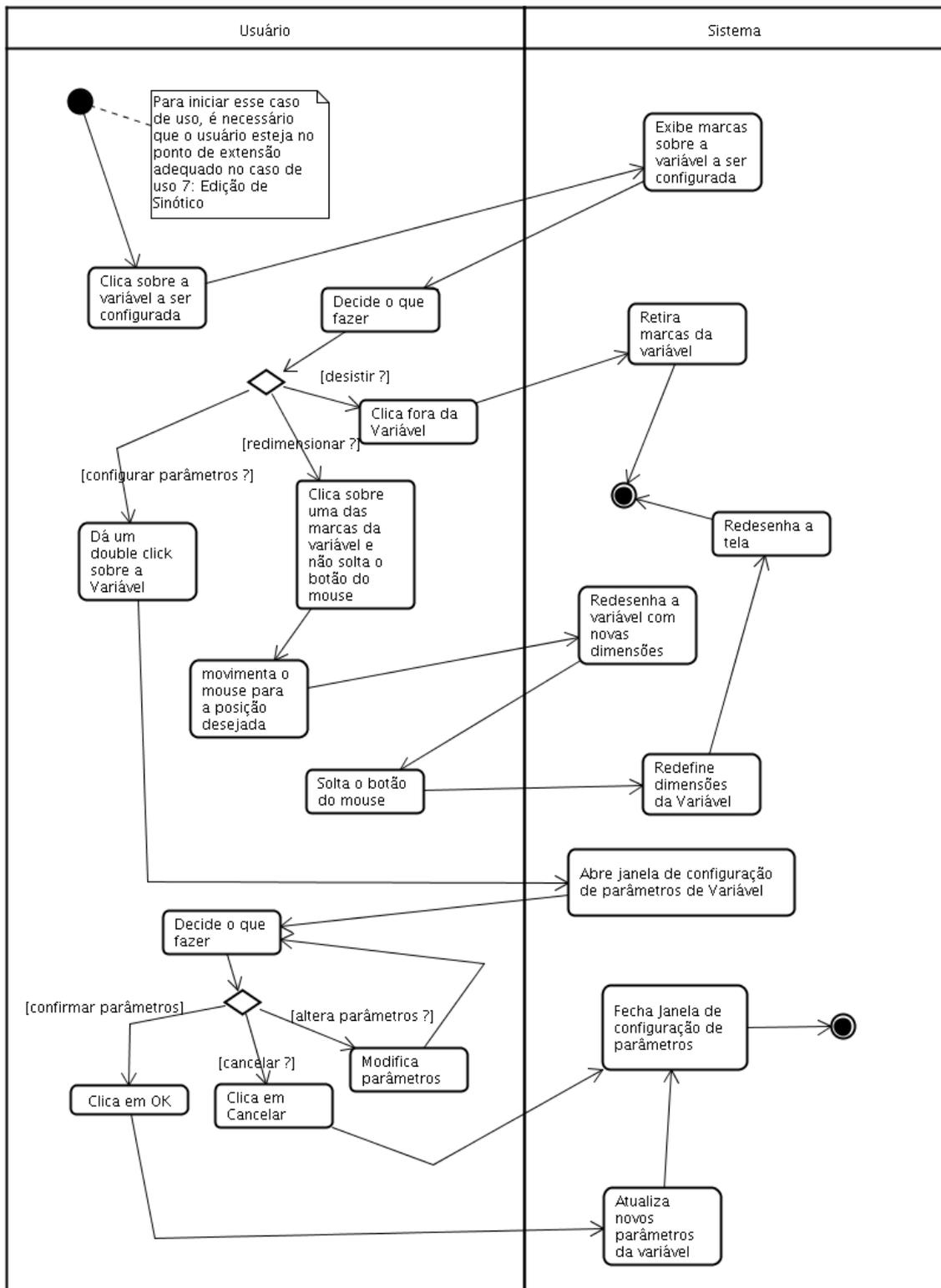


Figura 13 - Caso de Uso: Configuração de Variável

Este caso de uso estende o caso de uso Edição de Sinótico, permitindo que o usuário configure uma variável sendo monitorada no sinótico sendo editado. Basicamente o usuário tem duas opções: ou ele pode redimensionar a variável, ou ele pode modificar os parâmetros de configuração da variável. Se quiser redimensionar a variável, ele clica em uma das marcas da variável e a move para o novo ponto

desejado. O sistema deverá redimensionar a variável considerando esse novo ponto. Alternativamente, o usuário pode dar um “double click” na variável, e uma nova janela com os parâmetros de configuração irá aparecer. Nela, o usuário poderá editar todos os parâmetros da variável que deseja, e após terminar a configuração, ele poderá confirmar as alterações ou cancelar a alteração.

Caso de Uso 12: Monitoramento de Sinótico

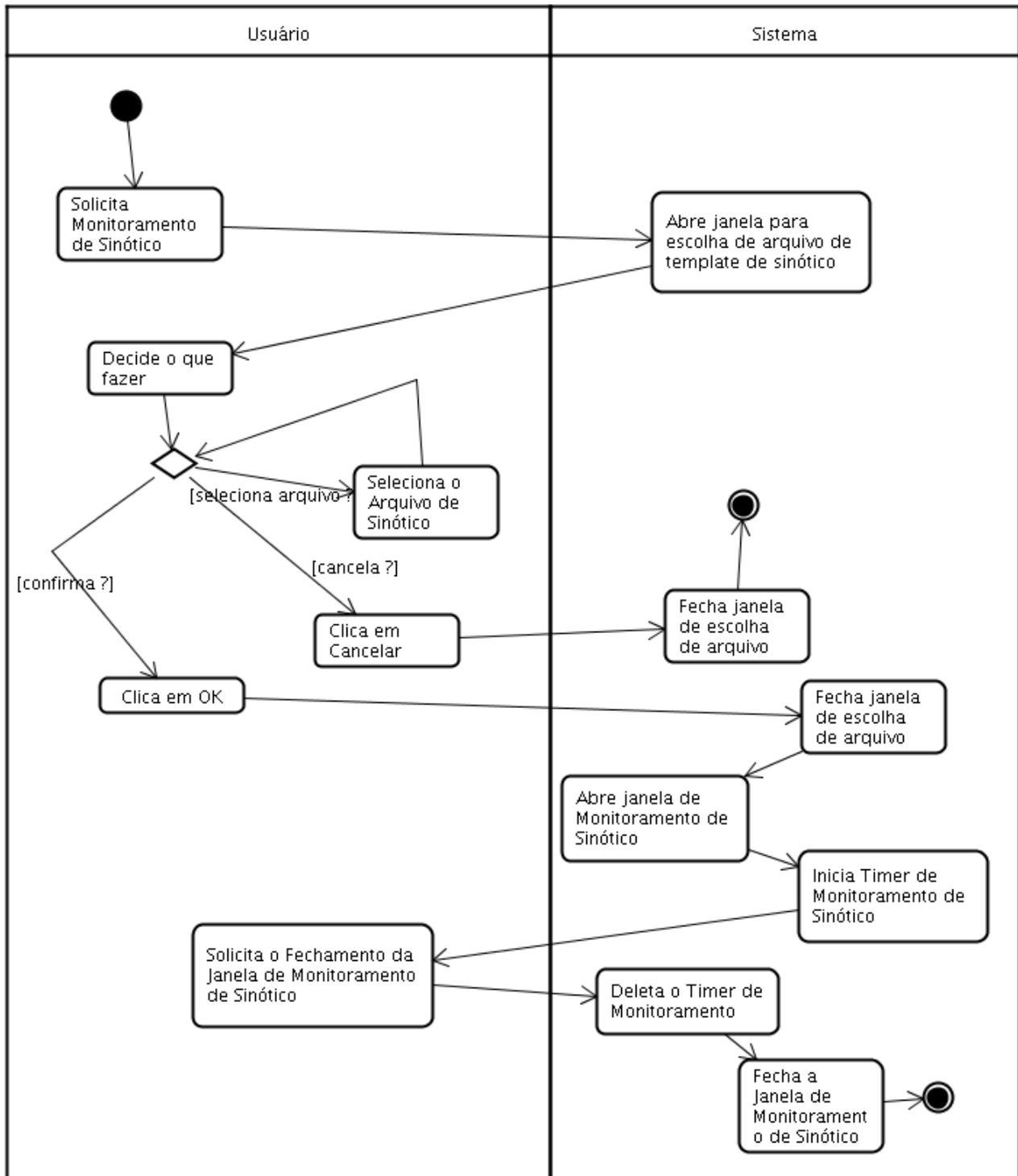


Figura 14 – Caso de Uso: Monitoramento de Sinótico

Neste caso de uso, o template de monitoração de sinótico editado no caso de uso 7: Edição de Sinótico, é utilizado para que o usuário obtenha o monitoramento de um conjunto de variáveis em uma janela definida no template. O usuário basicamente escolhe o arquivo com o template do sinótico, e o sistema abre a janela de monitoramento e inicia o Timer que fará a atualização da tela. Após isso, o sistema fica aguardando o usuário solicitar o fechamento da janela, e quando isso ocorre,

ele deleta o Timer e fecha a janela de monitoramento.

Caso de Uso 13: Atualização de Sinótico

Este caso de uso complementa o caso de uso 12: Monitoramento de Sinótico. Basicamente, quando uma janela de sinótico está sendo monitorada, ela está associada a um Timer de Sinótico que envia ticks periódicos de relógio. A cada tick, o sistema captura as variáveis do sinótico por meio do OPC e do Banco de Dados (conforme tenha sido configurada por meio do caso de uso Configuração de Variável), atualiza seu valor e redesenha tela para espelhar os novos valores.

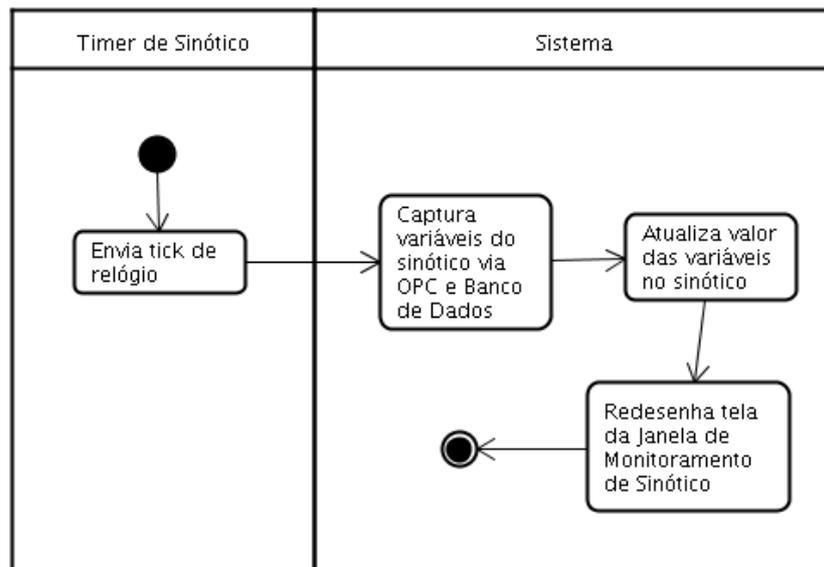


Figura 15 – Atualização do Sinótico

Caso de Uso 14: Inspeção de Variáveis

Nesse caso de uso, o usuário tem a opção de inspecionar uma ou mais variáveis, uma a uma. O usuário seleciona a inspeção de variáveis, e o sistema abre uma janela de inspeção de variáveis. Nela, o usuário pode selecionar qual variável deseja inspecionar, e o sistema busca essa variável no lugar adequado e exibe seu valor. O usuário pode então decidir inspecionar outra variável ou então finalizar a operação.

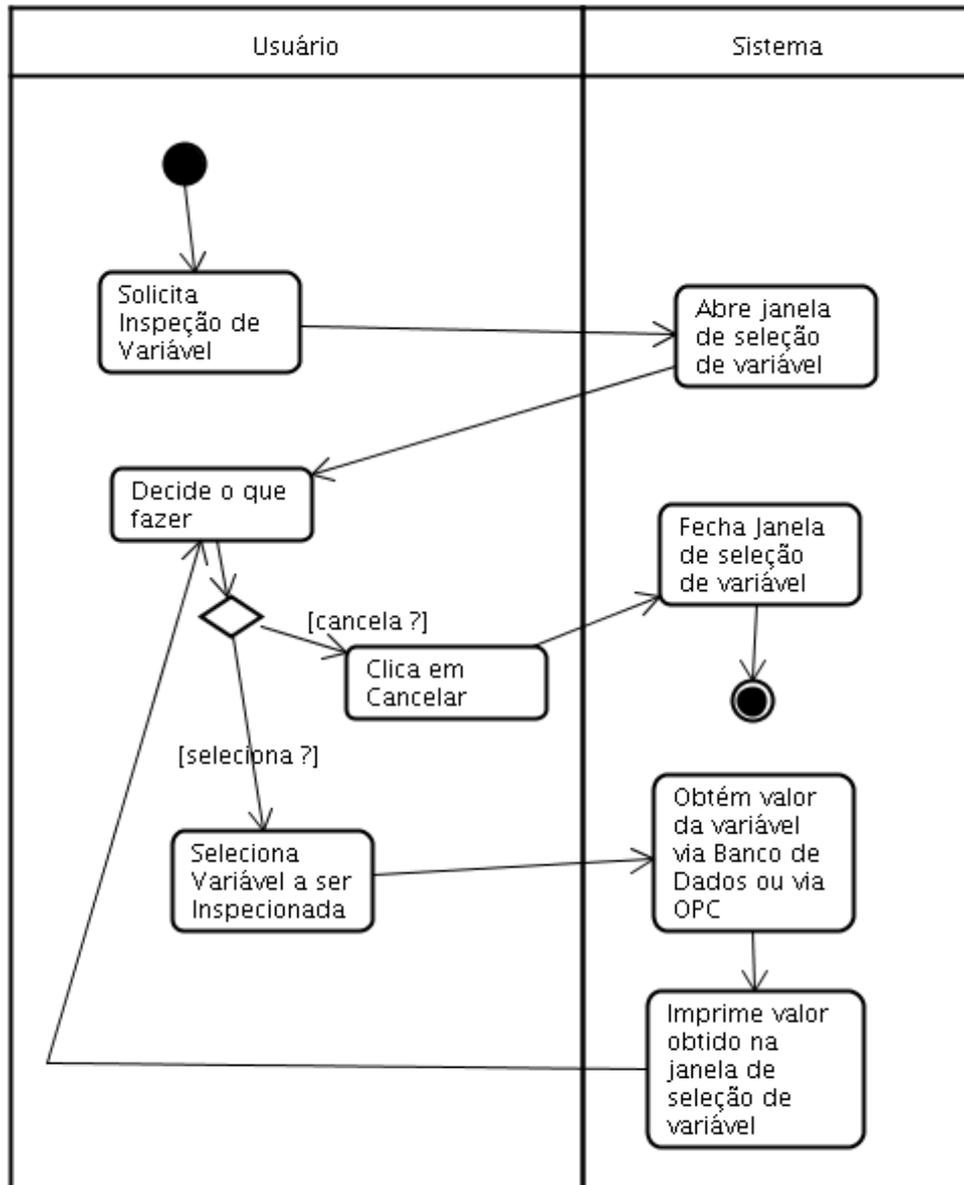


Figura 16 – Caso de Uso: Inspeção de Variável

Caso de Uso 15: Configuração do I-Kernel

Nesse caso de uso, o usuário pode atualizar os parâmetros de configuração do I-Kernel. Basicamente ele solicita ao sistema a configuração do I-Kernel, e o sistema abre uma janela de configuração, contendo todos os parâmetros configuráveis. O usuário pode então efetuar as alterações que desejar, e em seguida pode confirmar as mudanças ou sair cancelando a operação, caso em que nenhuma mudança será implementada.

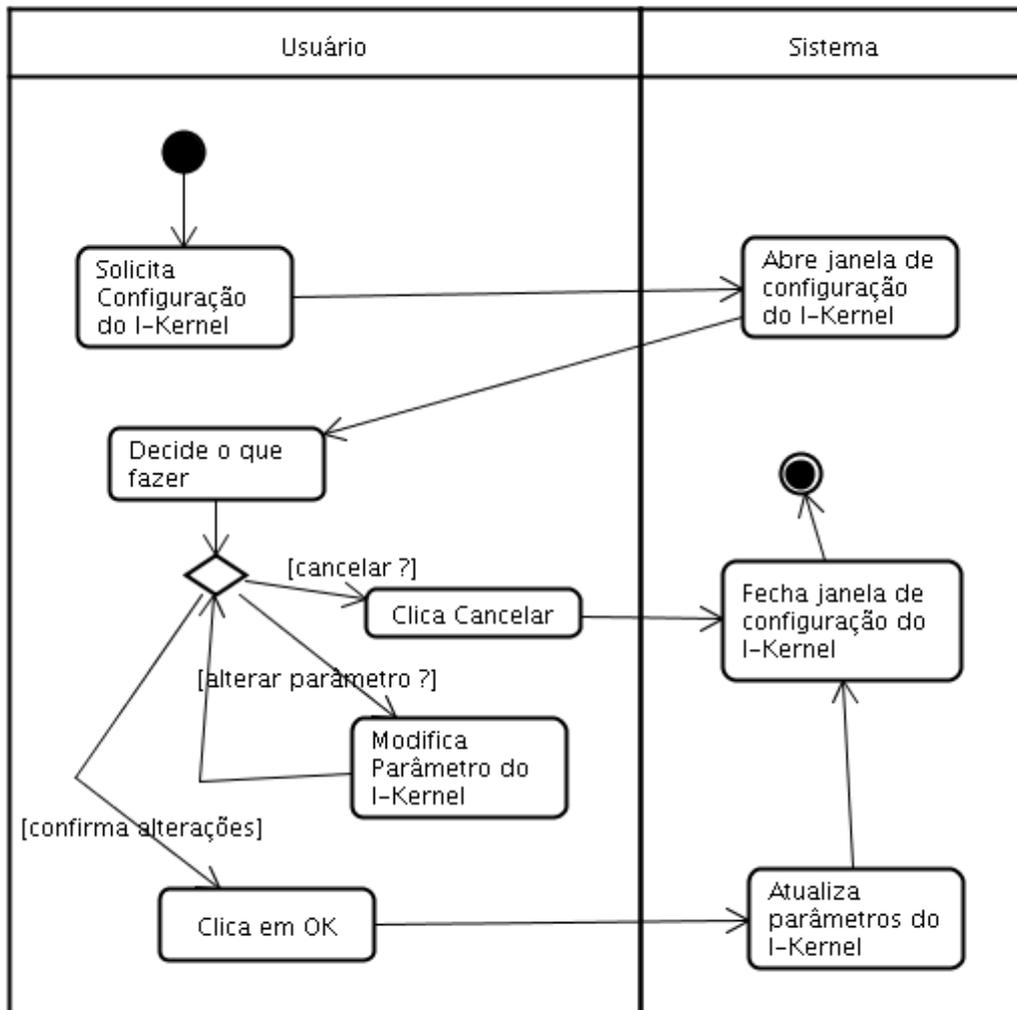


Figura 17 – Caso de Uso 15: Configuração do I-Kernel

Caso de Uso 16: Solicitação de Popup de Alerta

Esse caso de uso é utilizado em conjunto com o caso de uso 3: Verificação de Alarmes e Alertas. No caso de uso 3, o sistema I-Kernel envia uma solicitação via rede para a exibição de um popup de alerta caso alguma variável assim o seja programada. No presente caso de uso, a solicitação chega via rede, e o sistema C&M cria a janela de popup, mostrando a mensagem desejada ao usuário. Após ler a mensagem o usuário pode solicitar o fechamento da janela de popup.

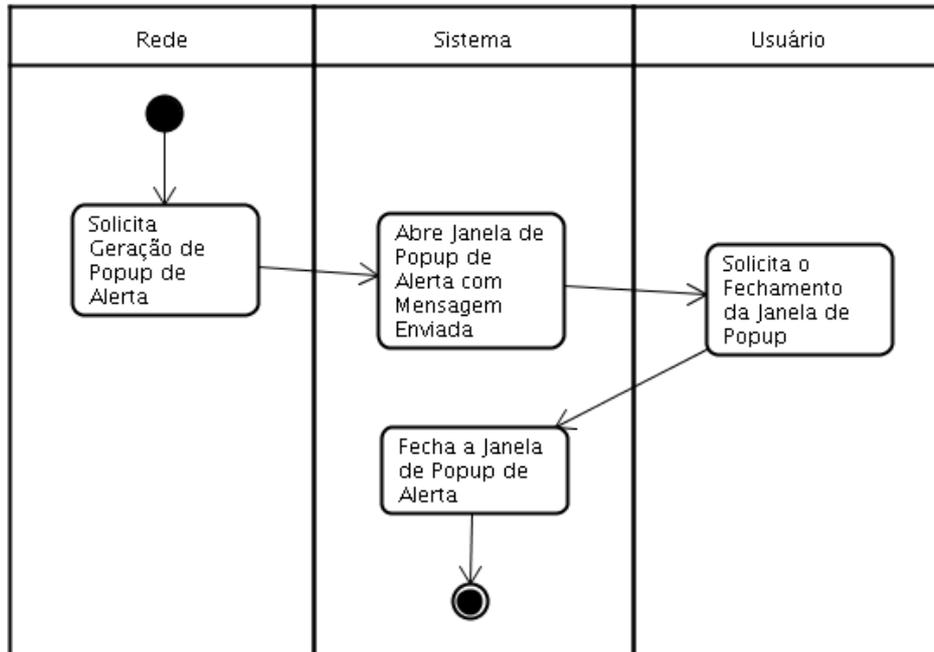


Figura 18 – Caso de Uso: Solicitação de Popup de Alerta

Referências

(Alvares, 2006) Alberto Alvares - SIMPREBAL: Metodologia Do Sistema De Manutenção Preditiva Da Usina De Balbina Baseado Nos Dados Monitorados Do Sistema De Supervisão E Controle Smar E Rockwell, Relatório Técnico de Pesquisa, UNB, Julho de 2006.

(Jacobson et. al. 1999) Ivar Jacobson, Grady Booch, James Rumbaugh - The Unified Software Development Process – Addison Wesley, 1999.

(Sommerville, 2003) Ian Sommerville - Engenharia de Software, 6a. edição, Addison-Wesley/Pearson Education.