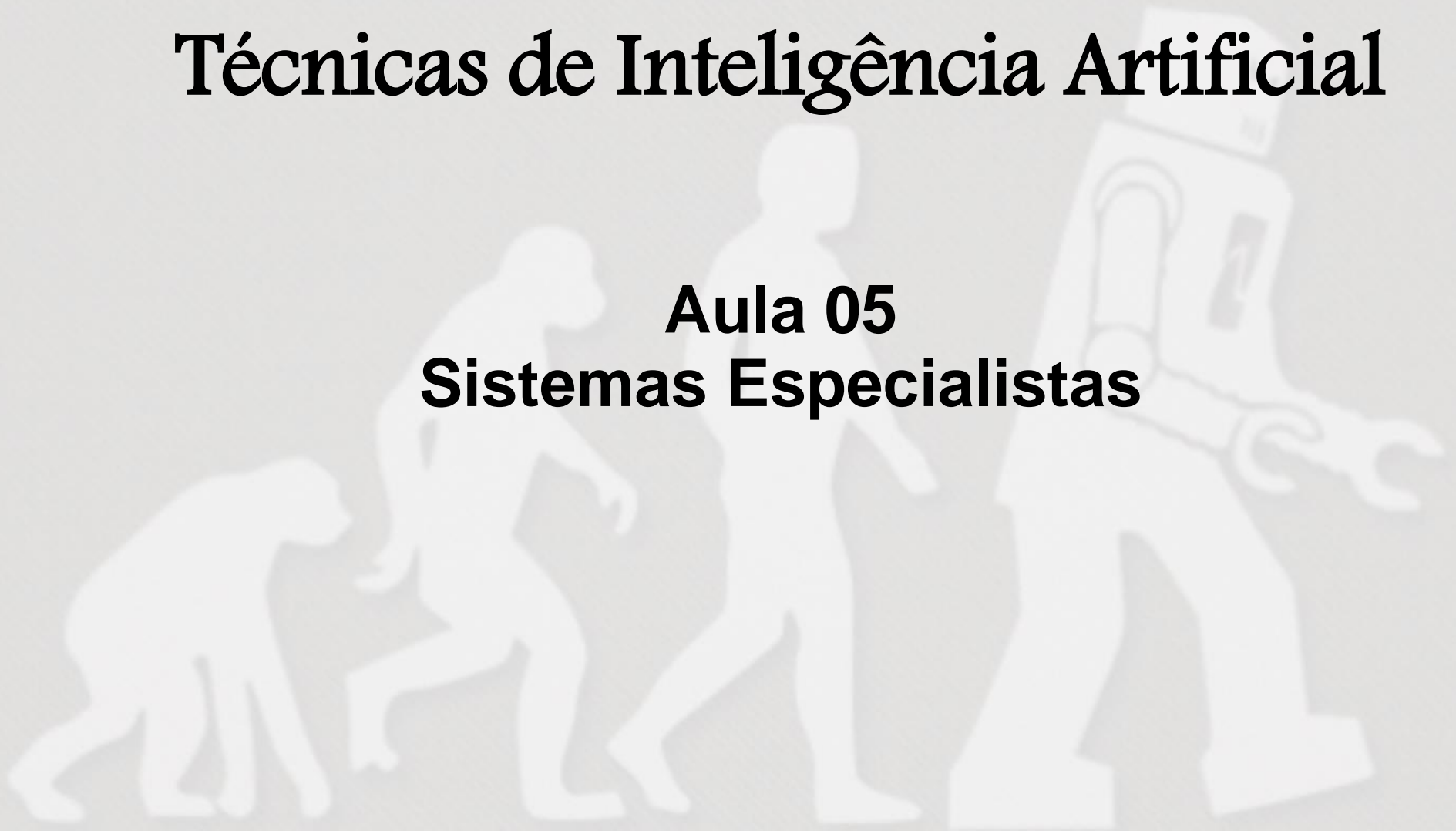


**Universidade do Sul de Santa Catarina
Ciência da Computação**

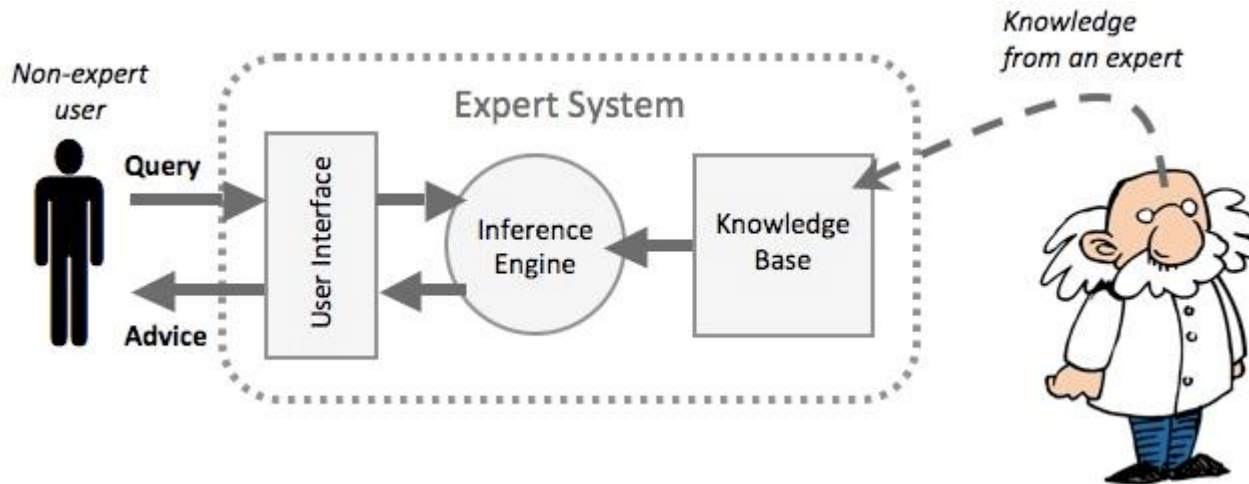
Técnicas de Inteligência Artificial

Aula 05 Sistemas Especialistas

Max Pereira



Sistemas Especialistas



- Pesquisadores de IA queriam desenvolver programas que pudessem “pensar”
- Ou seja, resolver problemas de uma maneira que seria considerada inteligente se fosse feita pelo homem



S.E. ou Sistemas baseados em conhecimento

São sistemas que reproduzem o conhecimento de um especialista adquirido ao longo do tempo.

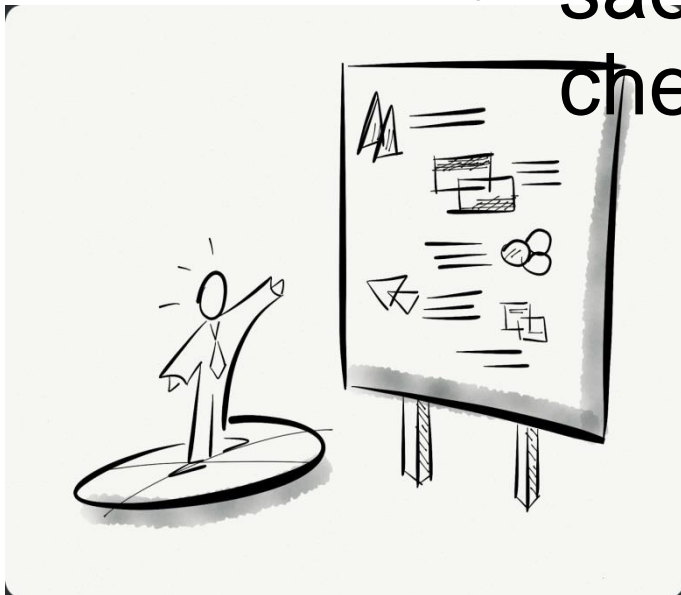
Portanto, o especialista é fundamental para fornecer informações específicas do domínio.



Sistemas especialistas:

Modelam o conhecimento humano em áreas específicas.

- Podem resolver problemas melhor que os humanos;
- aplicam o conhecimento humano a problemas bem compreendidos;
- são capazes de informar como chegaram a decisão;



Aplicações

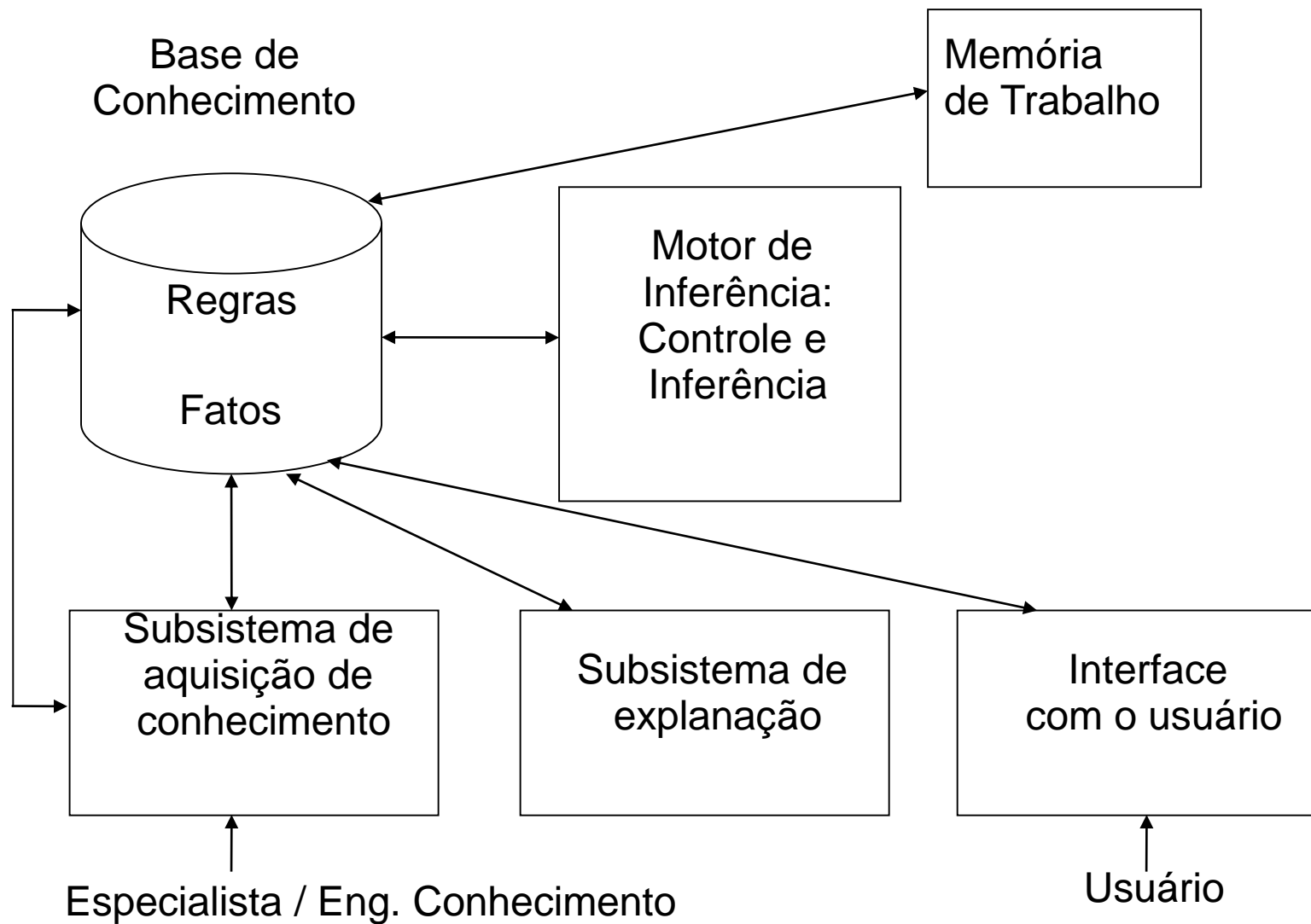
- Setor bancário e financeiro de um modo geral, para análise de crédito
- Análise química-qualitativa de minerais
- Projeto SINTA (Sistemas Inteligentes Aplicados): diagnóstico de pragas e doenças do cajueiro
- Gerência de redes
- Código Penal



- Podem explicar seu raciocínio ou decisões sugeridas. Ex.: Aprovação de crédito;
- Podem exibir um comportamento inteligente. Ex.: Diagnóstico médico;
- Podem esboçar conclusões em relacionamentos complexos. Ex.: Sugerir melhorias em processos produtivos;
- Podem fornecer conhecimento portátil. Ex.: Manutenção de equipamentos



ARQUITETURA



- Especialistas têm dificuldade em explicitar seu modo de raciocínio de uma maneira analítica



- A base de conhecimento pode ser construída com diversos formalismos, estruturas e linguagens:

Regras

Frames ou quadros

Redes semânticas

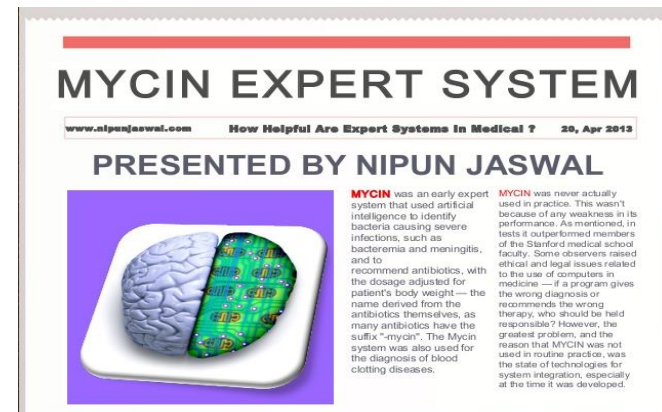
Scripts ou roteiros

Lógica



MYCIN

- Campo de doenças infecciosas
- Projetado para auxiliar no diagnóstico e tratamento de meningite e bacteriemia (infecção bacteriana no sangue)
- Formalismo de Representação do conhecimento: regras
- Uso de fatores de certeza (probabilidade)



Se a infecção é meningite

E organismos não foram vistos na cultura

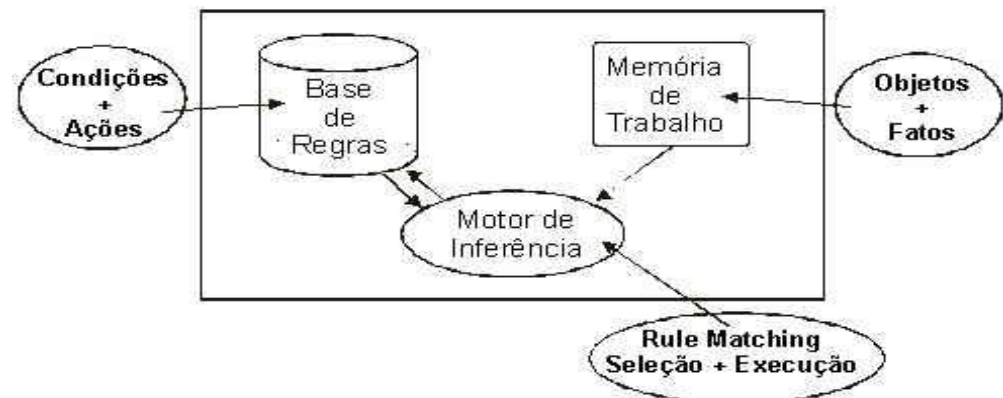
E o tipo de infecção é bacterial

E o paciente não tem lesões na cabeça

E a idade do paciente está entre 15 e 55 anos

ENTÃO

Os organismos que podem estar causando a infecção
são *diplococcus-pneumoniae* (0.75) E
neisseria-meningitidis (0.74)



Máquina de Inferência

- Mecanismo que procura as respostas na BC
- Busca as regras necessárias a serem avaliadas e ordenadas de uma maneira lógica
- Define a busca das regras, comparando a entrada do usuário com as regras da BC buscando “combinações”
- No Prolog este processo é denominado “matching” ou unificação



Forward Chaining

- Regras

1. $A \wedge B \rightarrow C$
2. $A \rightarrow D$
3. $C \wedge D \rightarrow E$
4. $B \wedge E \wedge F \rightarrow G$
5. $A \wedge E \rightarrow H$
6. $D \wedge E \wedge H \rightarrow I$

- Fatos

1. A
2. B
3. F

- Meta

- H

Fatos	Objetivos	Regras Disparadas
A,B,F		

Backward Chaining

- Regras

1. $A \wedge B \rightarrow C$
2. $A \rightarrow D$
3. $C \wedge D \rightarrow E$
4. $B \wedge E \wedge F \rightarrow G$
5. $A \wedge E \rightarrow H$
6. $D \wedge E \wedge H \rightarrow I$

- Fatos

1. A
2. B
3. F

- Meta

- H

Fatos	Objetivos	Regras Disparadas
A,B,F		

Desenvolvimento

Shell (OPS, ExpertSinta, KAS, ...) : é o mais utilizado

Linguagens de programação para IA (Prolog)

Linguagens de programação gerais (OOP)

Linguagens híbridas (componentes de IA): regras + objetos (CLIPS, JESS, NeOpus, JEOPS, etc.)



Pontos positivos

Criação de repositório de conhecimento

Crescimento de produtividade e qualidade

Habilidade de resolver problemas complexos

Flexibilidade e modularidade

Operação em ambientes arriscados

Credibilidade

Habilidade de trabalhar com informações incompletas ou incertas

Fornecimento de treinamento



Pontos negativos

Avaliação de desempenho difícil

É difícil extrair conhecimento especialista

Só trabalham muito bem em domínios estreitos

Engenheiros de Conhecimento são raros e caros

Transferência de conhecimento está sujeito a um grande número de preconceitos



Expert Sinta

É um shell que permite construir SE

www.lia.ufc.br/~bezerra/exsinta

Como utilizar:

1. Estabelecer variáveis
2. Estabelecer objetivos (variáveis de saída)
3. Estabelecer interfaces (perguntas a serem feitas)
4. Criar as regras

Para que serve o Expert Sinta?

É uma ferramenta computacional:

- utiliza técnicas de Inteligência Artificial

- realiza a geração automática de sistemas especialistas.

Modelo de representação do conhecimento:

- Regras de produção

- Probabilidades

Para que serve o Expert Sinta?

Simplifica a implementação de sistemas especialistas

Utiliza máquina de inferência compartilhada

Possui construção automática de telas e menus do tratamento probabilístico das regras de produção

Utilização explicações sensíveis ao contexto da base de conhecimento modelada

Para que serve o Expert Sinta?

O usuário responde a uma sequência de menus, e o sistema se encarrega de fornecer respostas que se encaixem no quadro apontado pelo usuário.

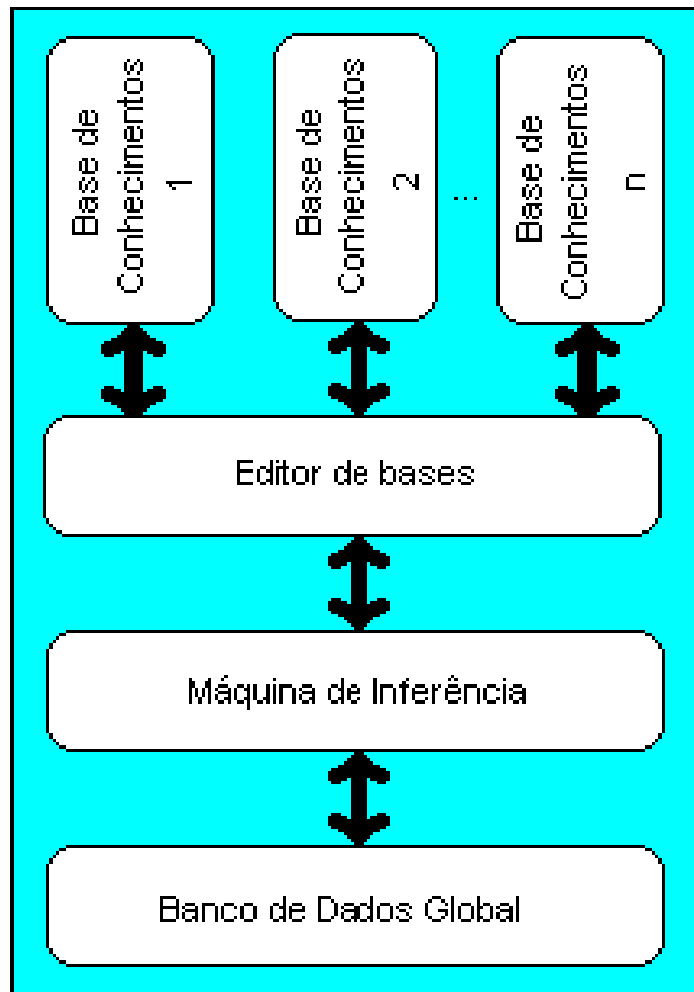
Exemplos:

- sistemas de diagnósticos médicos

- configuração de redes de computadores

Planejando um sistema especialista

Os sistemas especialistas (SE) gerados no Expert SINTA seguem a arquitetura abaixo:



- *base de conhecimentos* representa a informação (fatos e regras) que um especialista utiliza, representada computacionalmente;
- *editor de bases* é o meio pelo qual a *shell* permite a implementação das bases desejadas;
- *máquina de inferência* é a parte do SE responsável pelas deduções sobre a base de conhecimentos;
- *banco de dados global* são as evidências apontadas pelo usuário do sistema especialista durante uma consulta.

Utilizando regras de produção

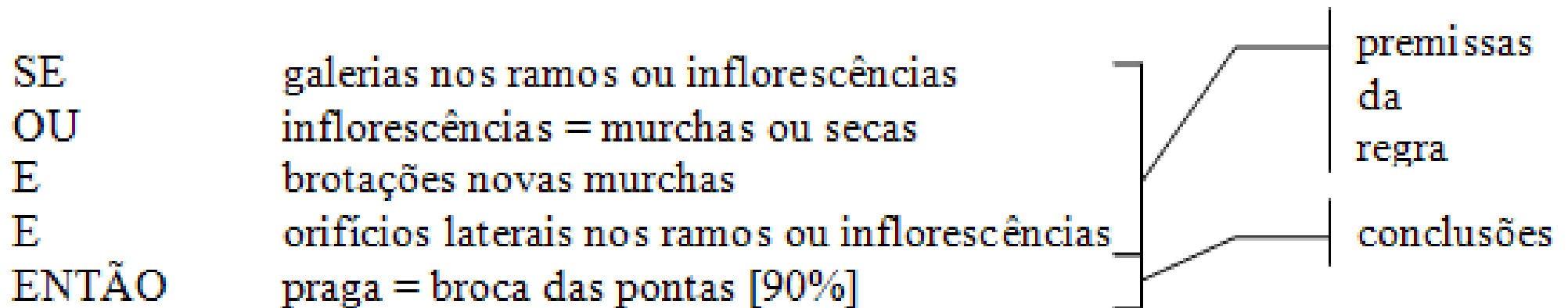
As regras de produção são populares por possuírem as seguintes vantagens:

Modularidade: cada regra, por si mesma, pode ser considerada como uma peça de conhecimento independente;

Facilidade de edição (uma consequência da modularidade): novas regras podem ser acrescentadas e antigas podem ser modificadas com relativa independência;

Transparência do sistema: garante maior legibilidade da base de conhecimentos.

Regra de Produção



Regra de Produção

- A estrutura de cada cauda (premissa) deve obedecer ao seguinte modelo:

<conectivo> <atributo> <operador> <valor>

- Conectivo: NÃO, E, OU (une as premissas)
- Atributo: é uma variável capaz de assumir uma ou múltiplas instâncias no decorrer da consulta à base de conhecimentos.
- Operador: une o atributo e o valor da premissa que define o tipo de comparação a ser realizada. São operadores relacionais: =, >, <=, <>, entre outros;
- Valor: é um item de uma lista a qual foi previamente criada e relacionada a um atributo.

Regra de Produção

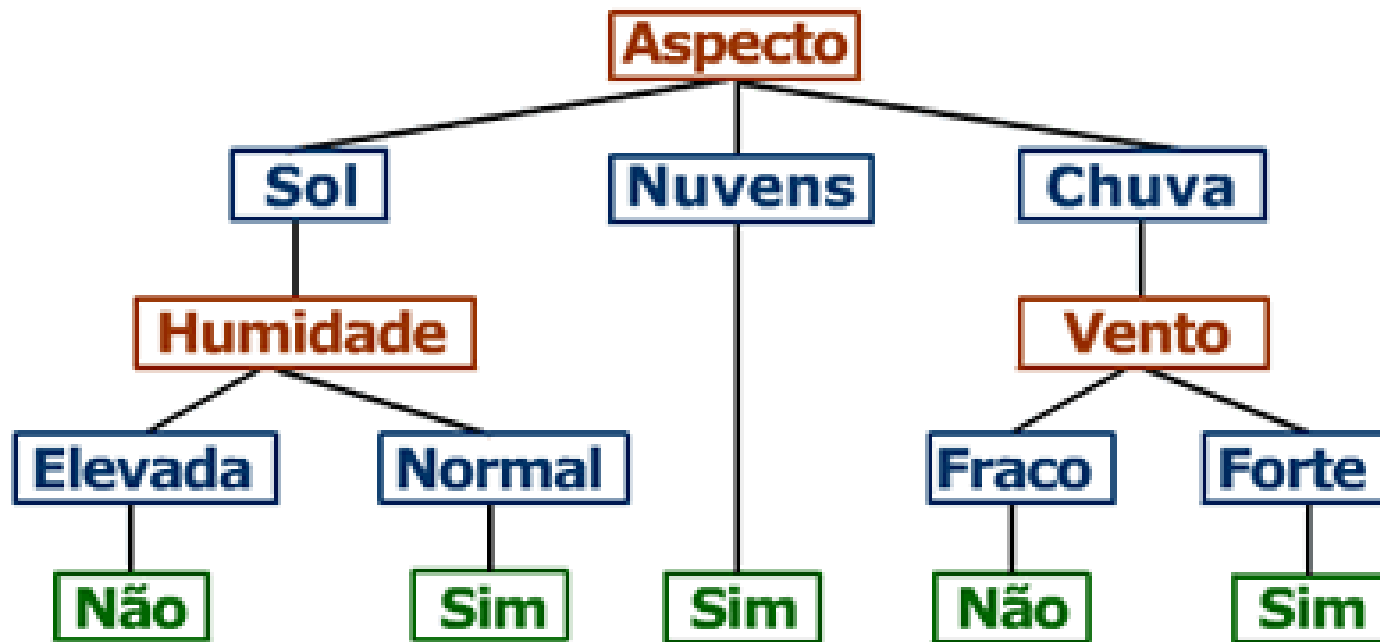
○ A estrutura de cada cauda (premissa) deve obedecer ao seguinte modelo:

<atributo>=<valor><grau de confiança>

- Atributo: é uma variável capaz de assumir uma ou múltiplas instâncias no decorrer da consulta à base de conhecimentos.
- =: é um operador de **atribuição** (o novo valor substituirá o antigo ou será empilhado com os demais).
- Valor: é um item de uma lista a qual foi previamente criada e relacionada a um atributo.
- Grau de confiança: é uma porcentagem indicando a **confiabilidade**. O grau de confiança varia de 0% a 100%.

Base de Conhecimento

Árvore de Decisão para Jogar Ténis



Base de Conhecimento

Regra 1. Se ASPECTO = nuvens então JOGAR = sim

Regra 2. Se ASPECTO = sol e UMIDADE = elevada então JOGAR = não

Regra 3. Se ASPECTO = sol e UMIDADE = normal então JOGAR = sim

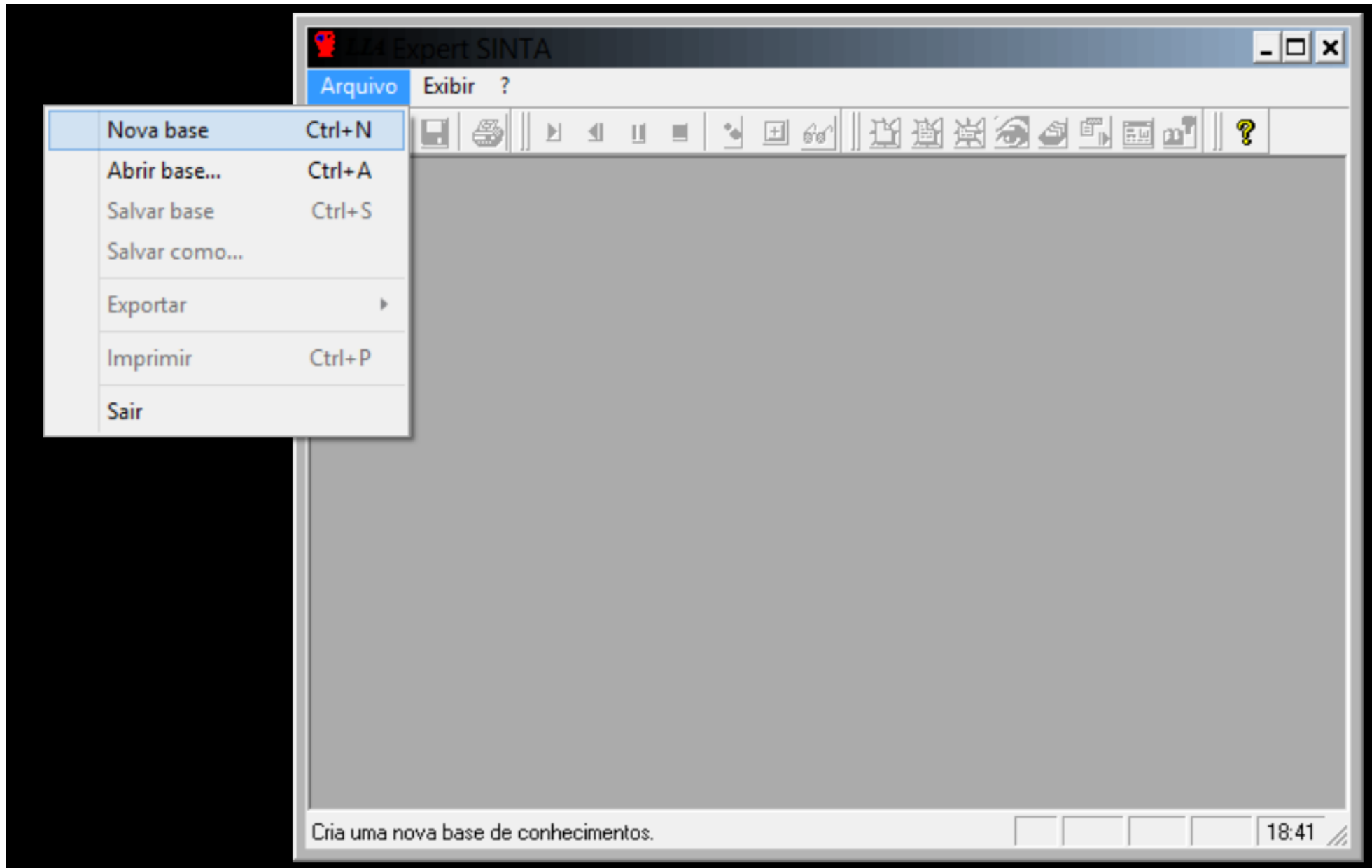
Regra 4. Se ASPECTO = chuva e VENTO = fraco então JOGAR = não

Regra 5. Se ASPECTO = chuva e VENTO = forte então JOGAR = sim

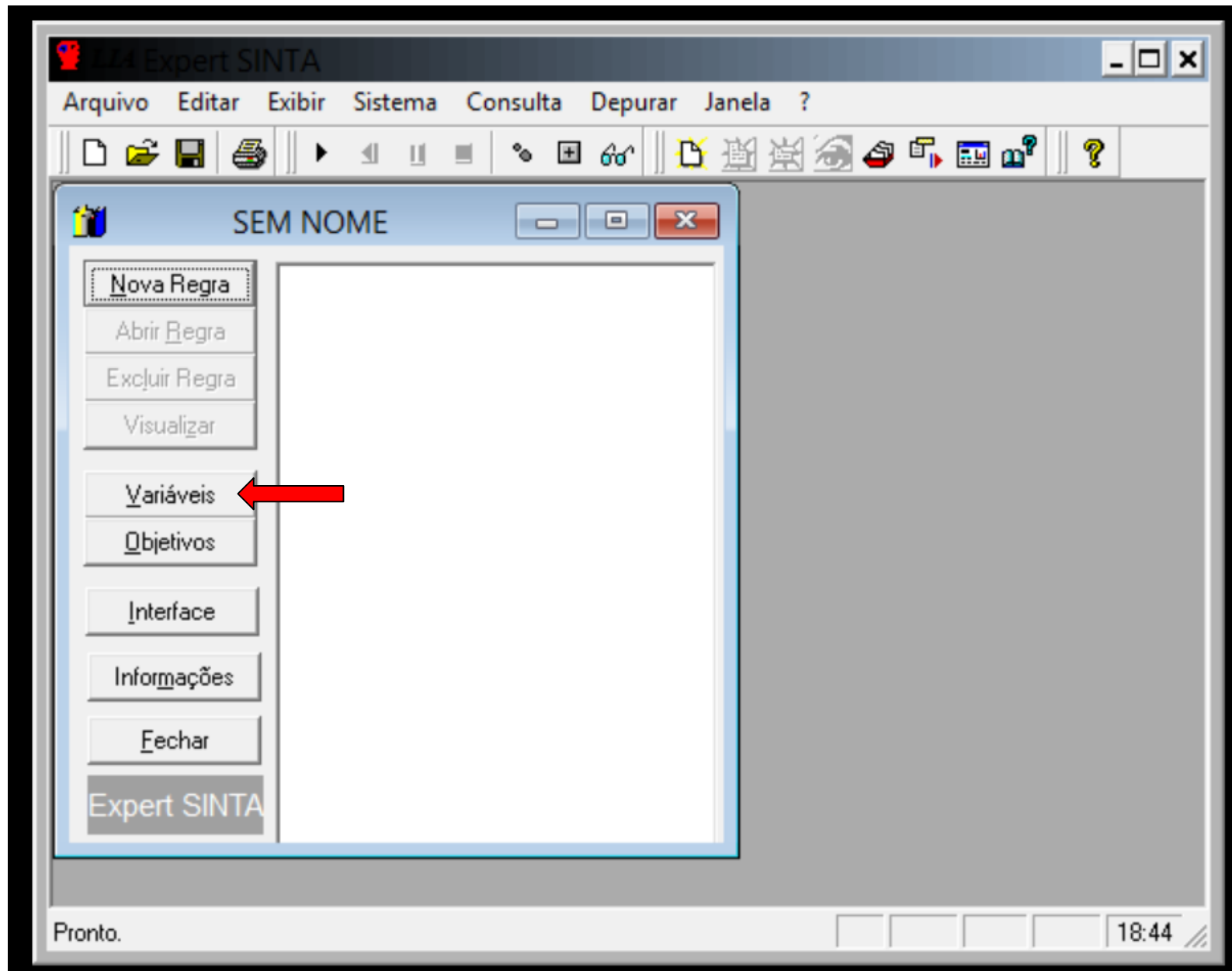
Usando o Expert Sinta

1. Estabelecer variáveis
2. Estabelecer objetivos (variáveis de saída)
3. Estabelecer interfaces (perguntas a serem feitas)
4. Criar as regras

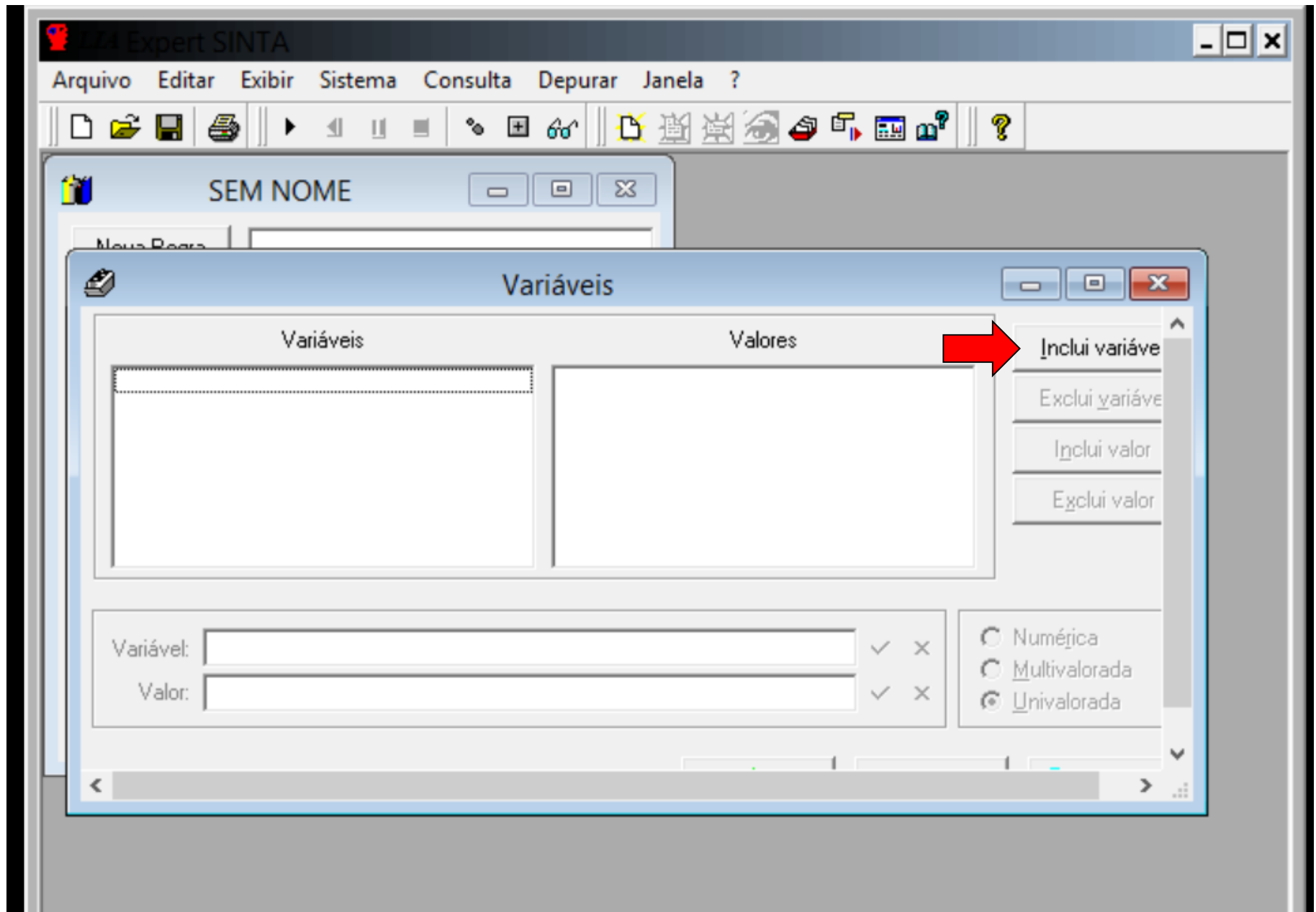
Criando uma nova base



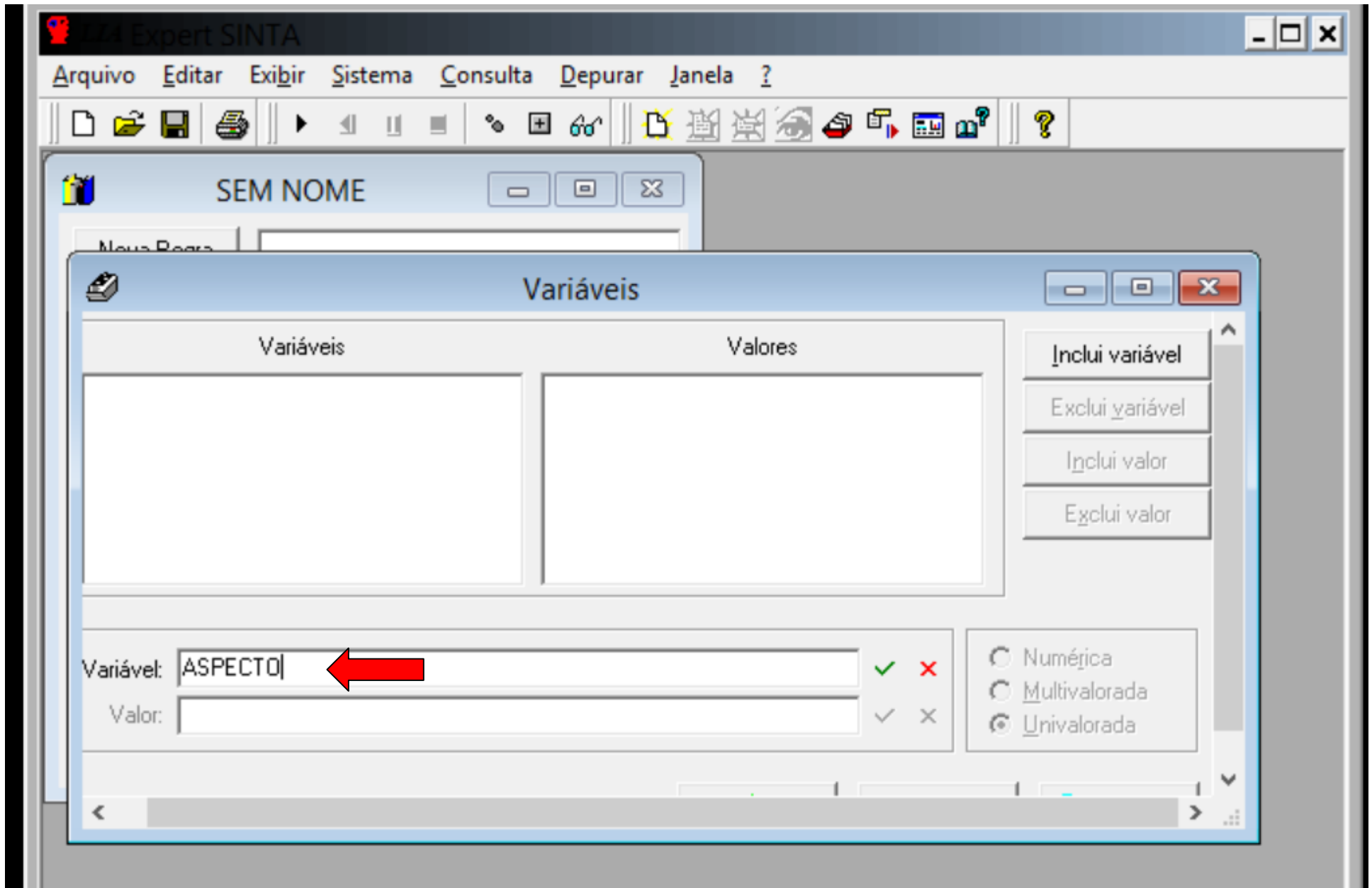
Inserindo as variáveis



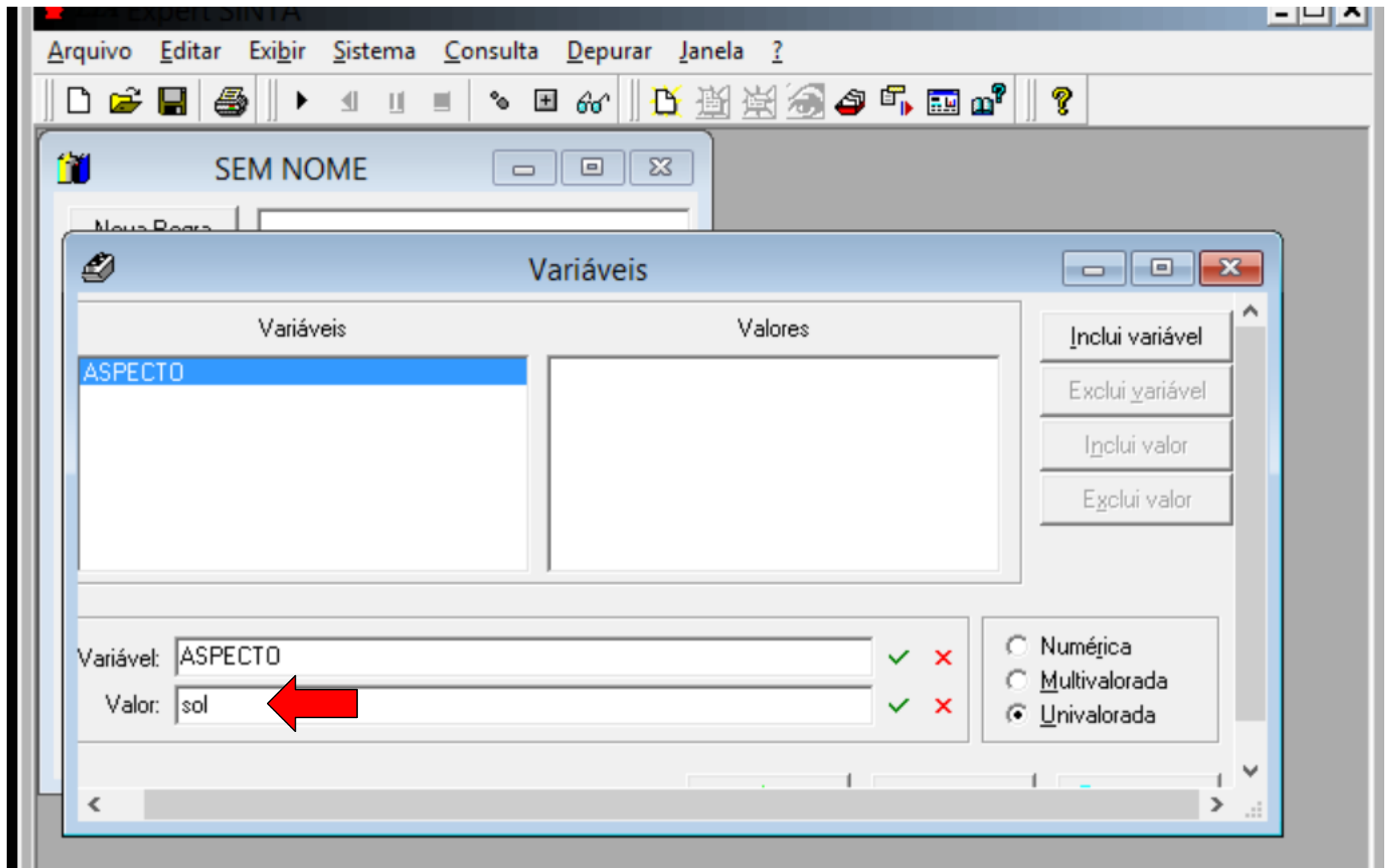
Incluindo uma variável



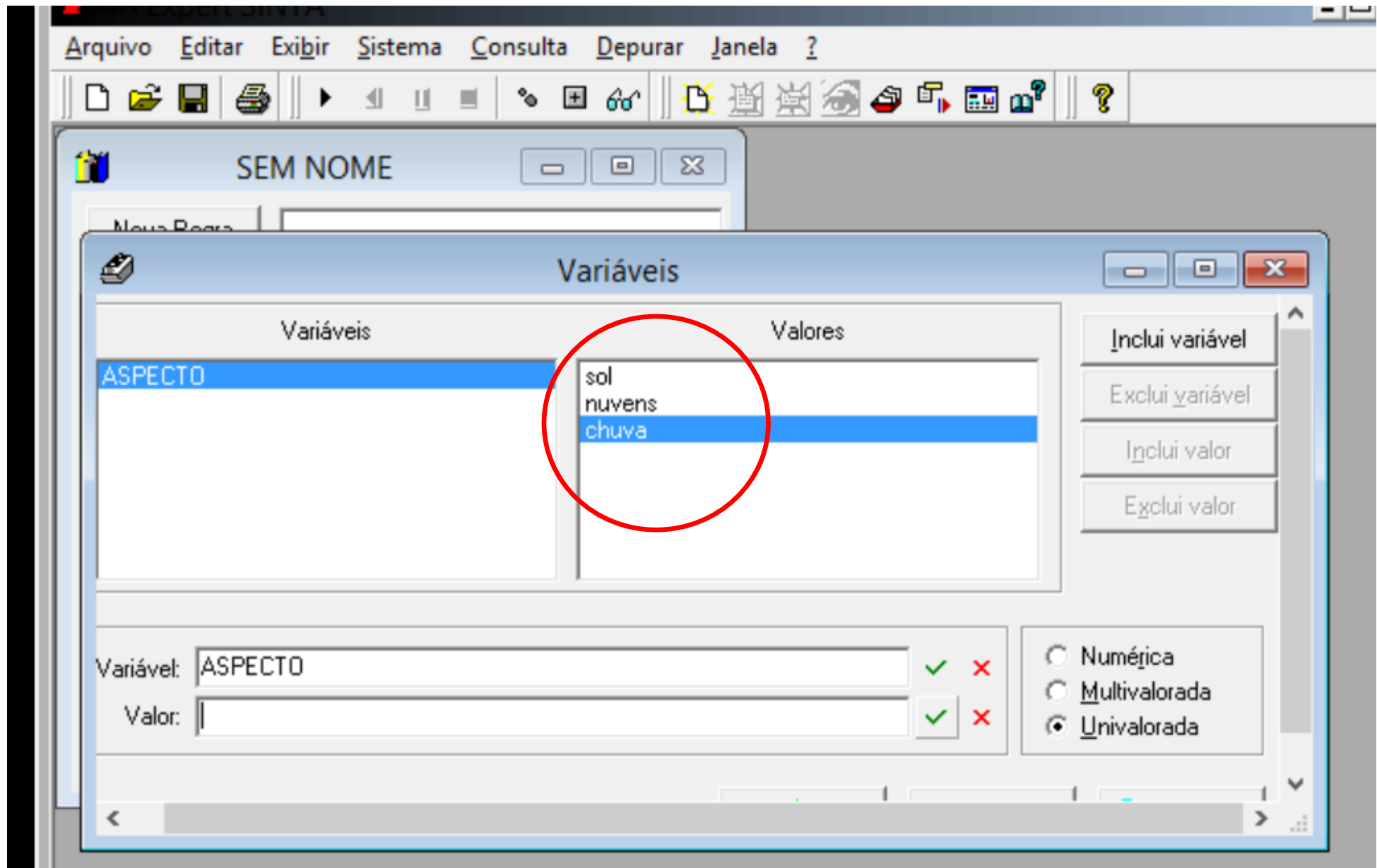
Incluindo uma variável



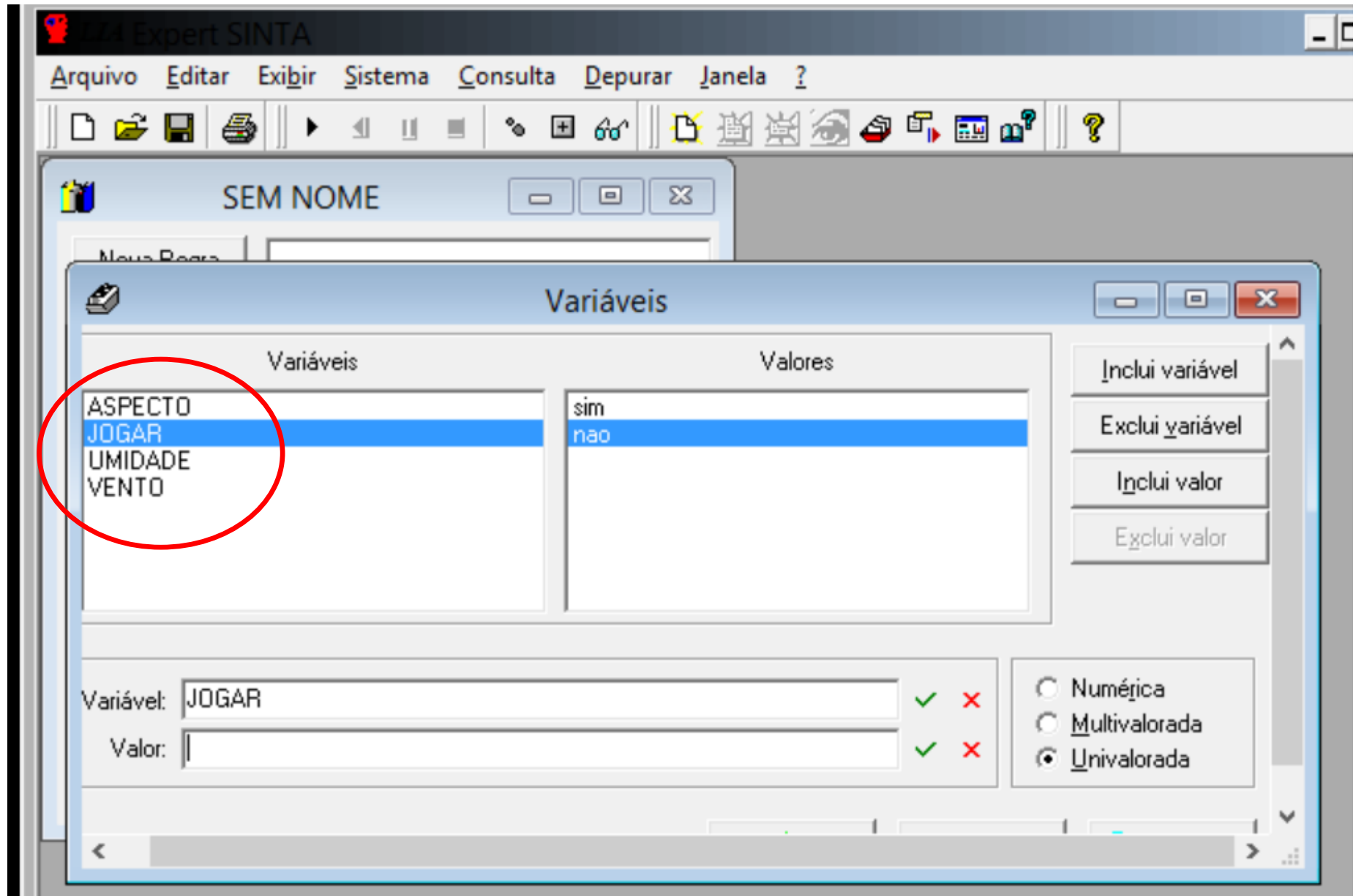
Incluindo valores para uma variável



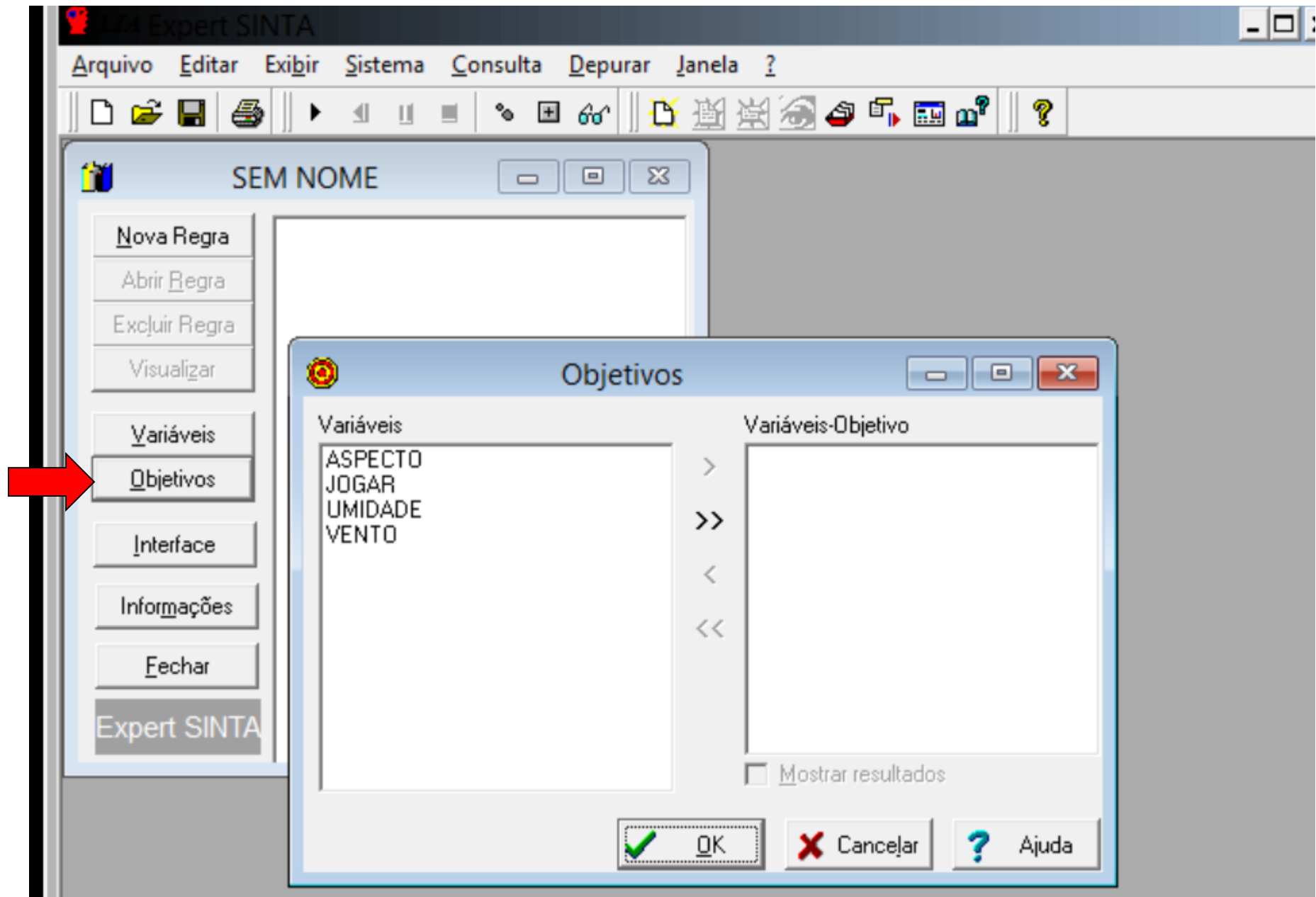
Incluindo valores para uma variável



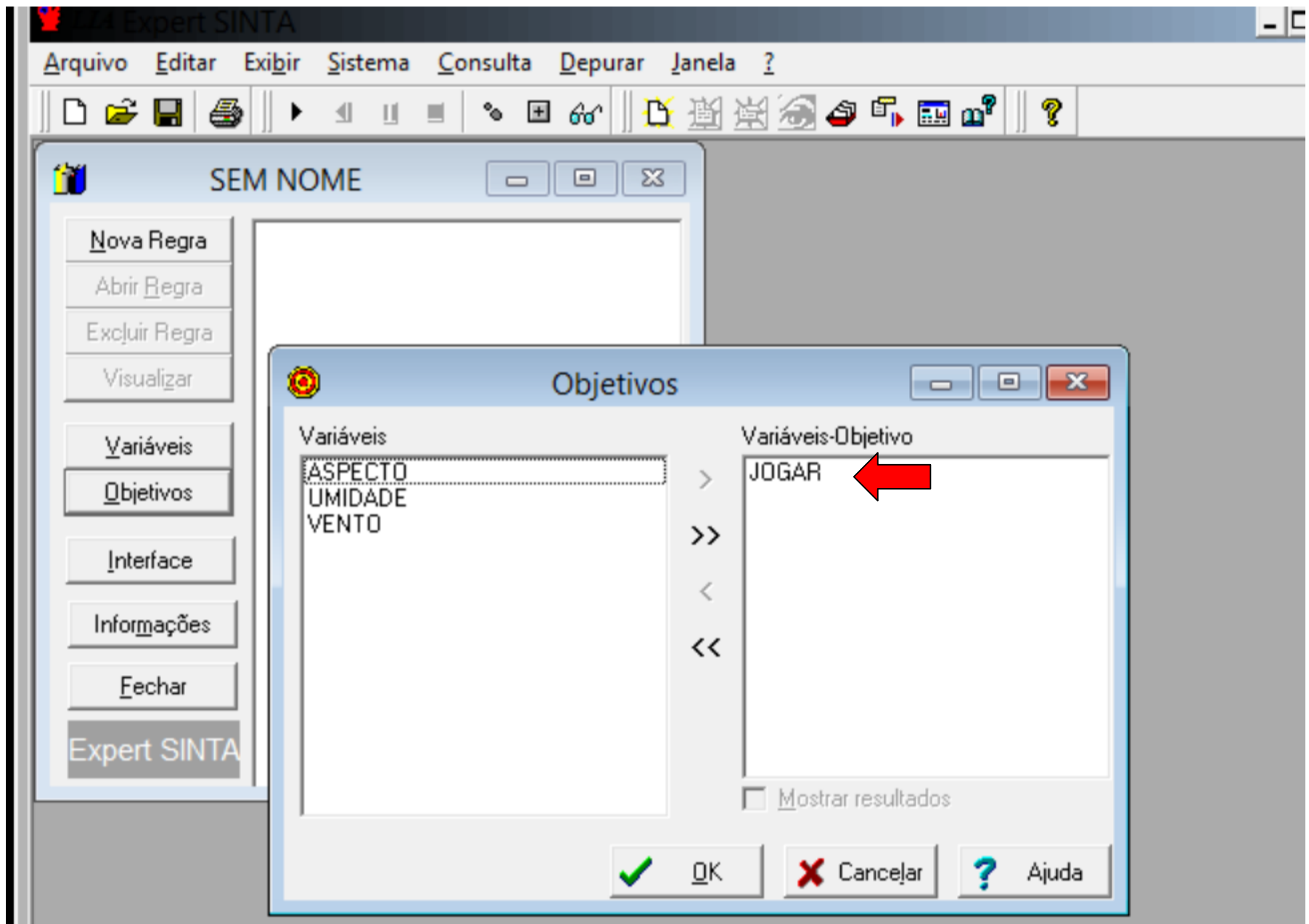
Inclusão de todas as variáveis



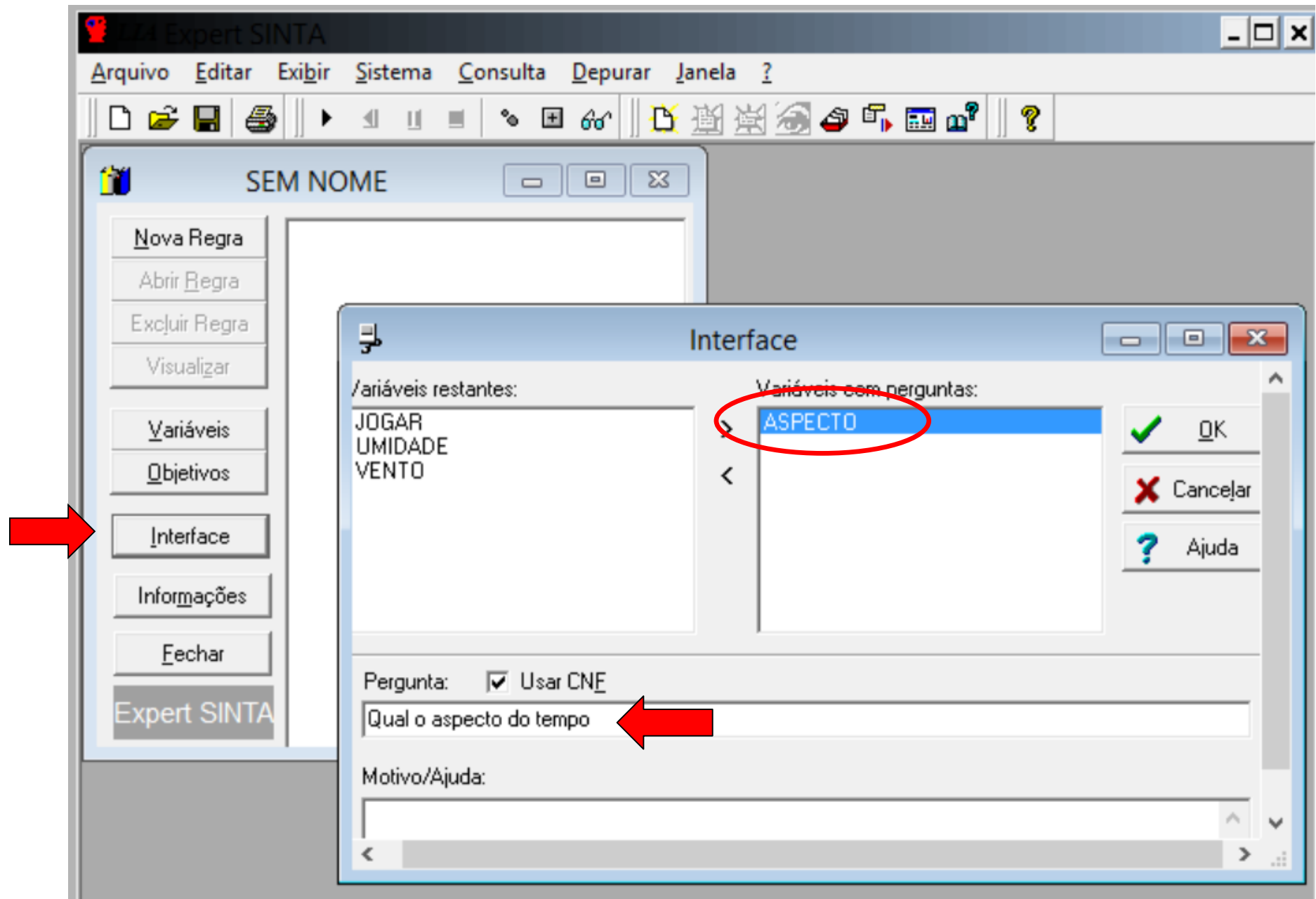
Definindo objetivo(s)



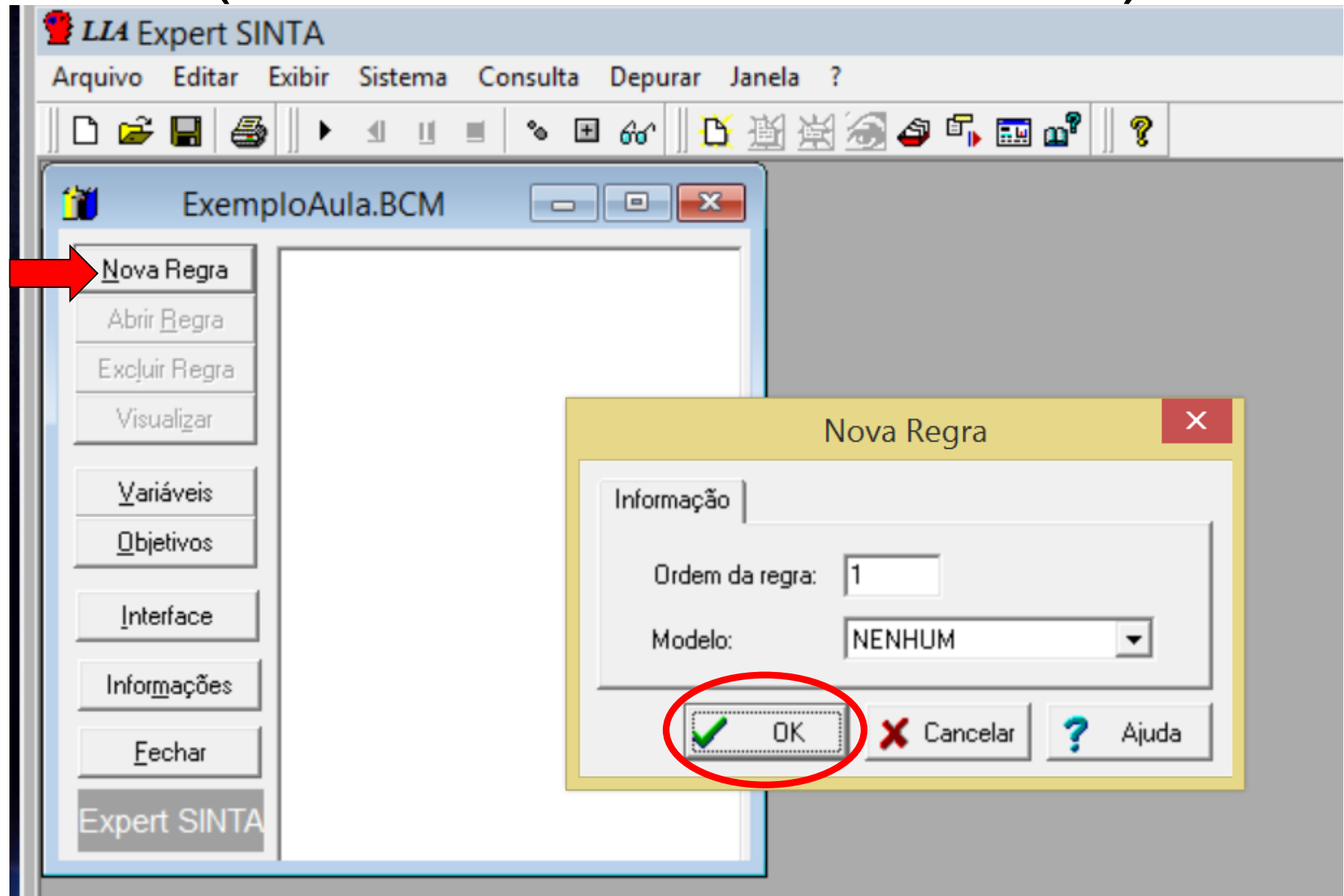
Definindo objetivo(s)



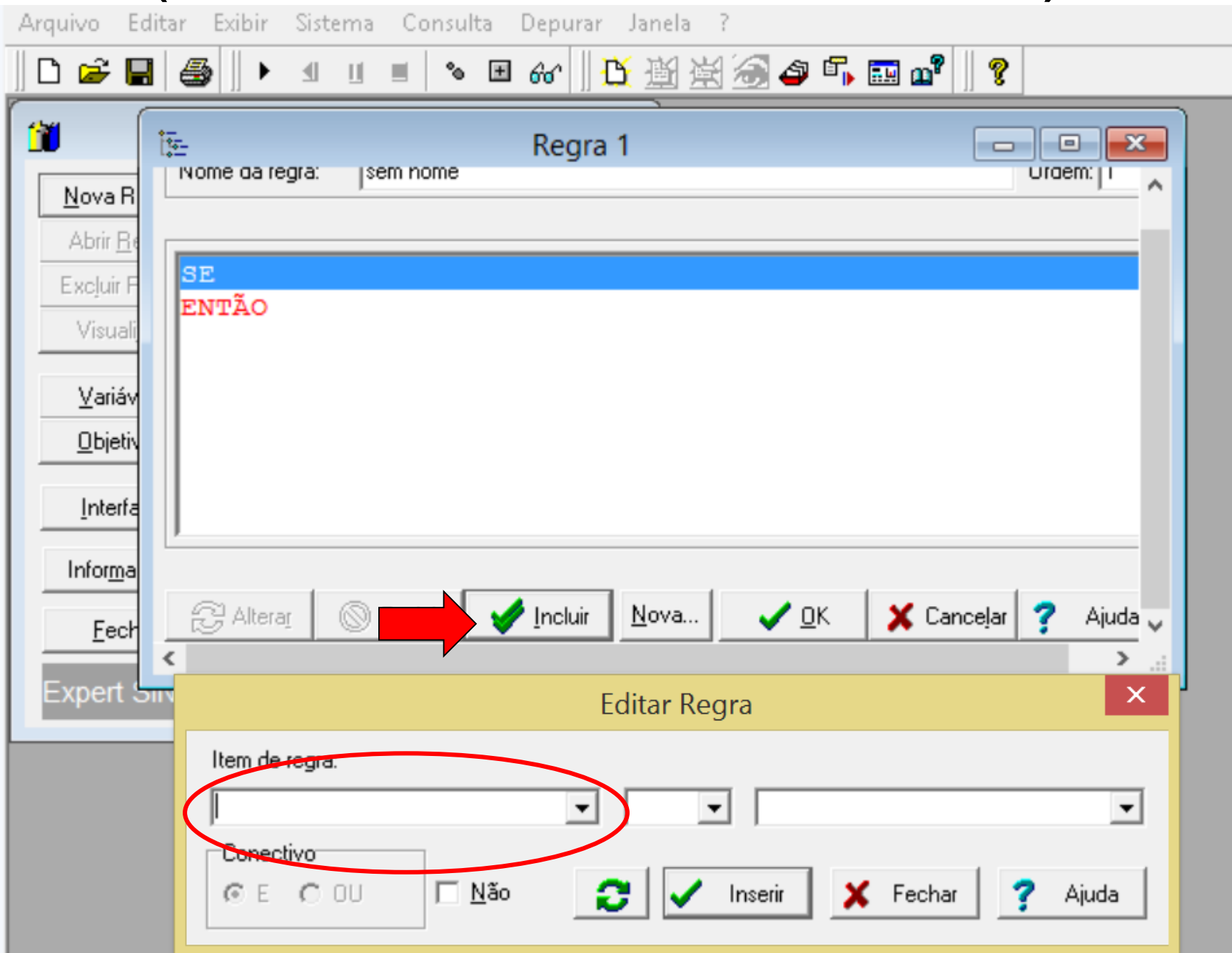
Construindo a interface (perguntas)



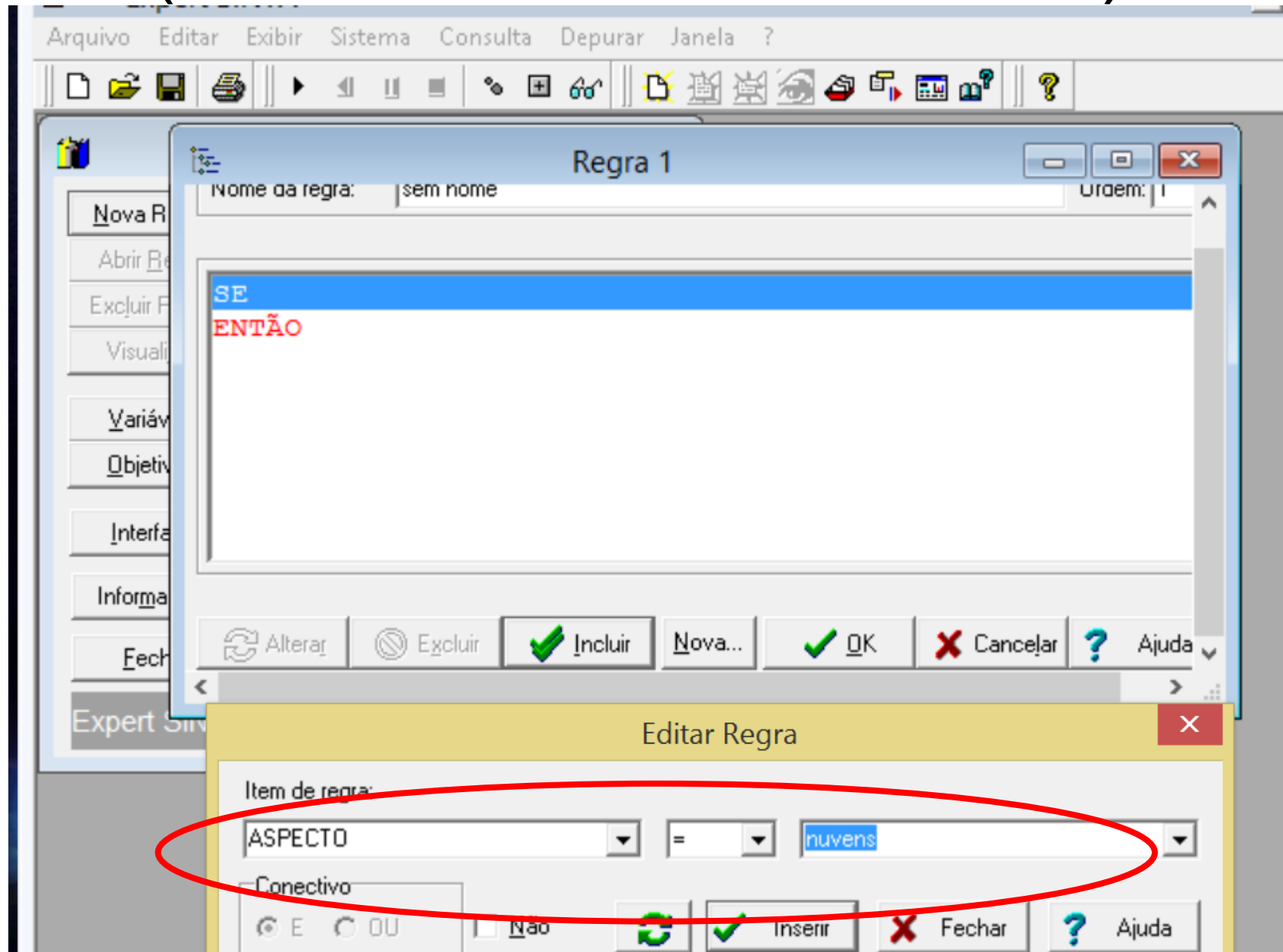
Definindo as regras (Base de Conhecimento)



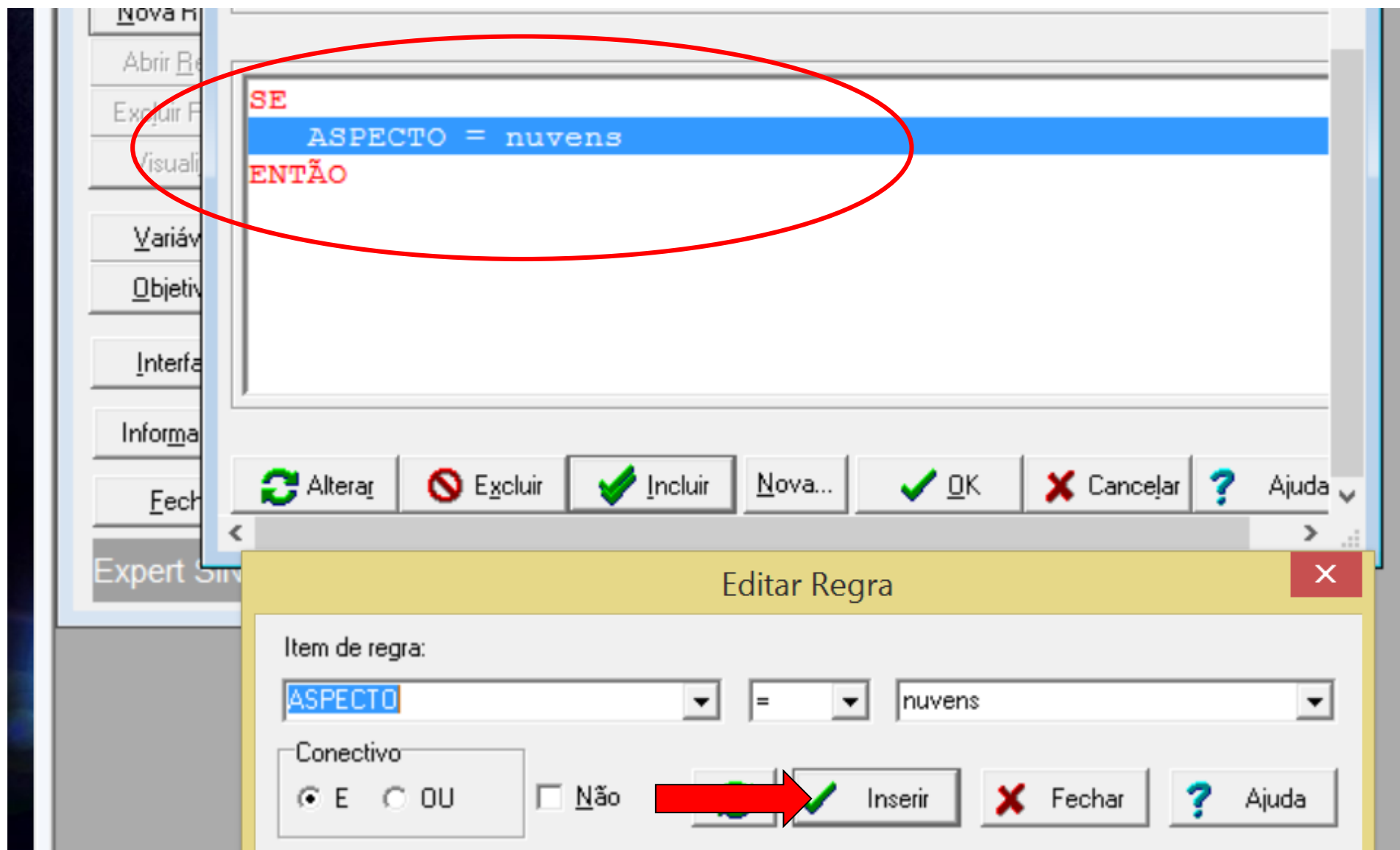
Definindo as regras (Base de Conhecimento)



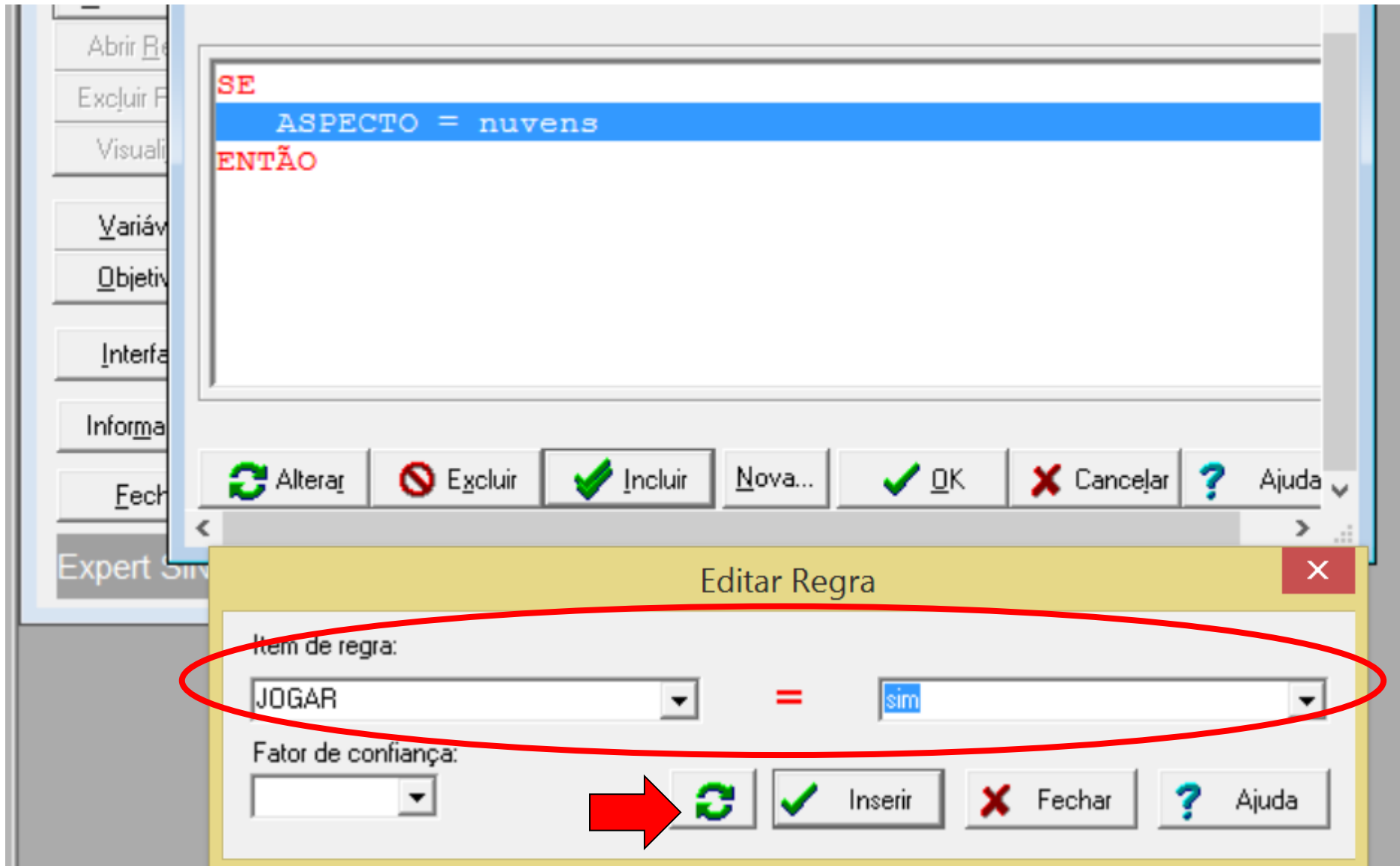
Definindo as regras (Base de Conhecimento)



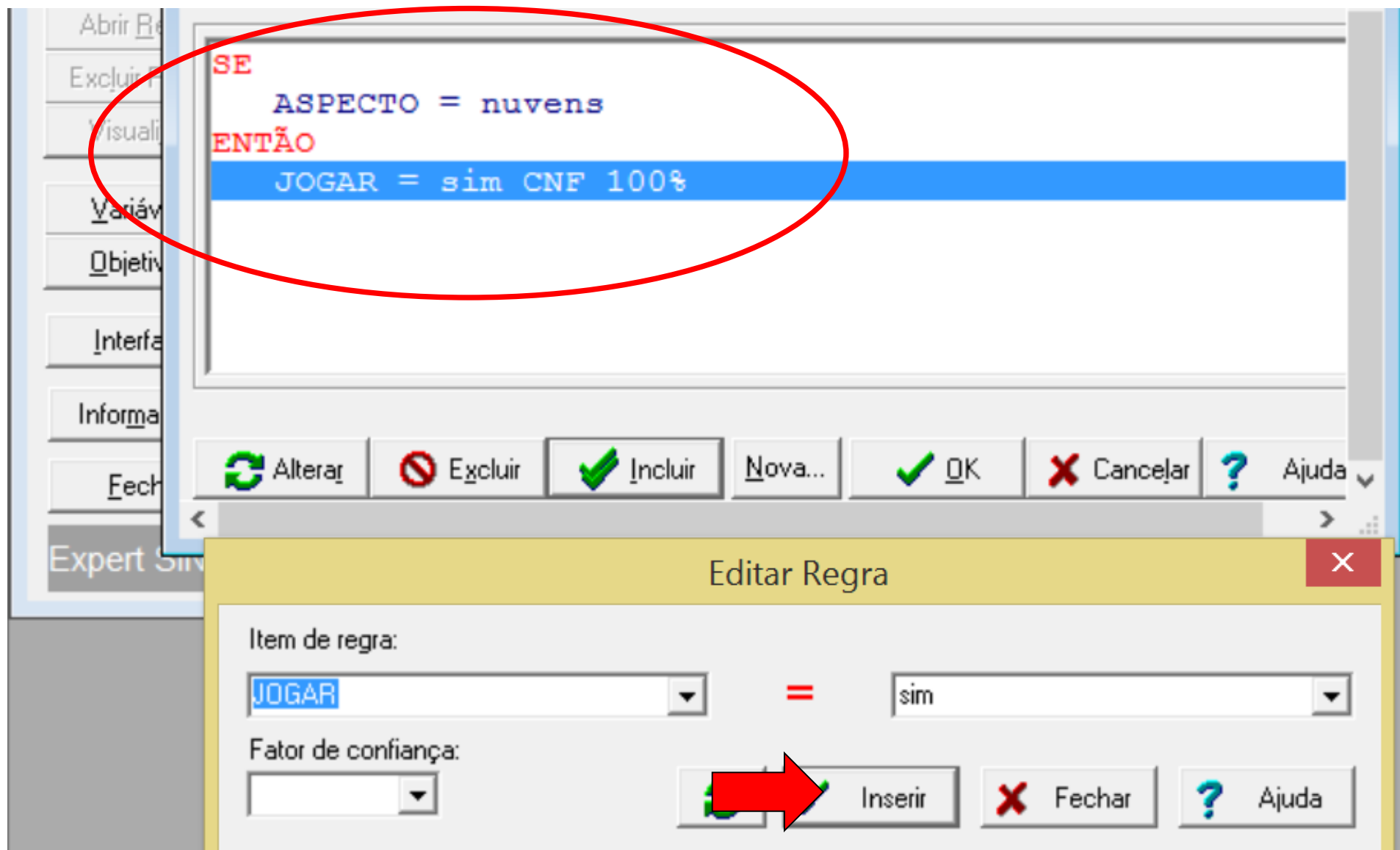
Definindo as regras (Base de Conhecimento)



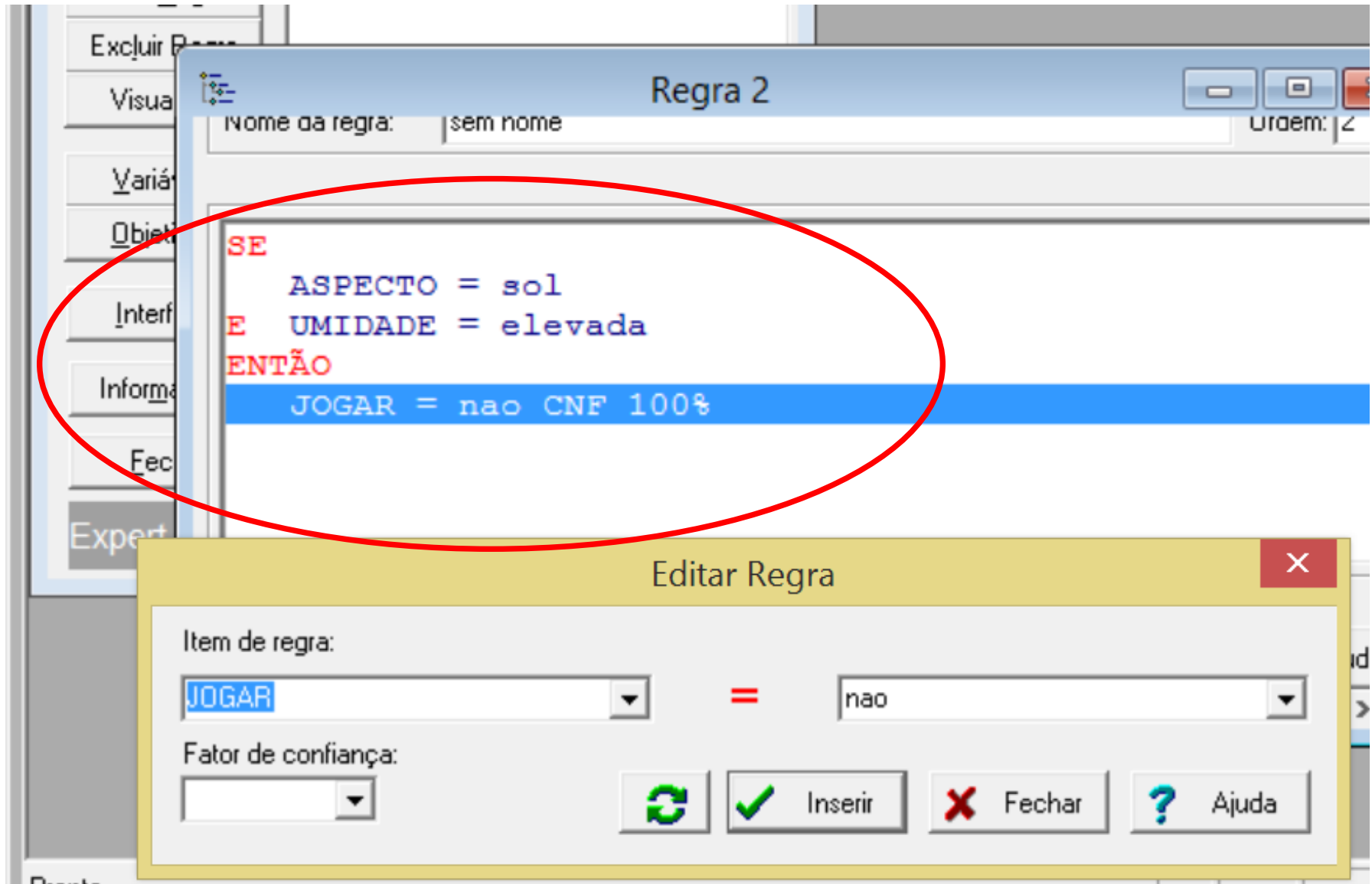
Definindo as regras (Base de Conhecimento)



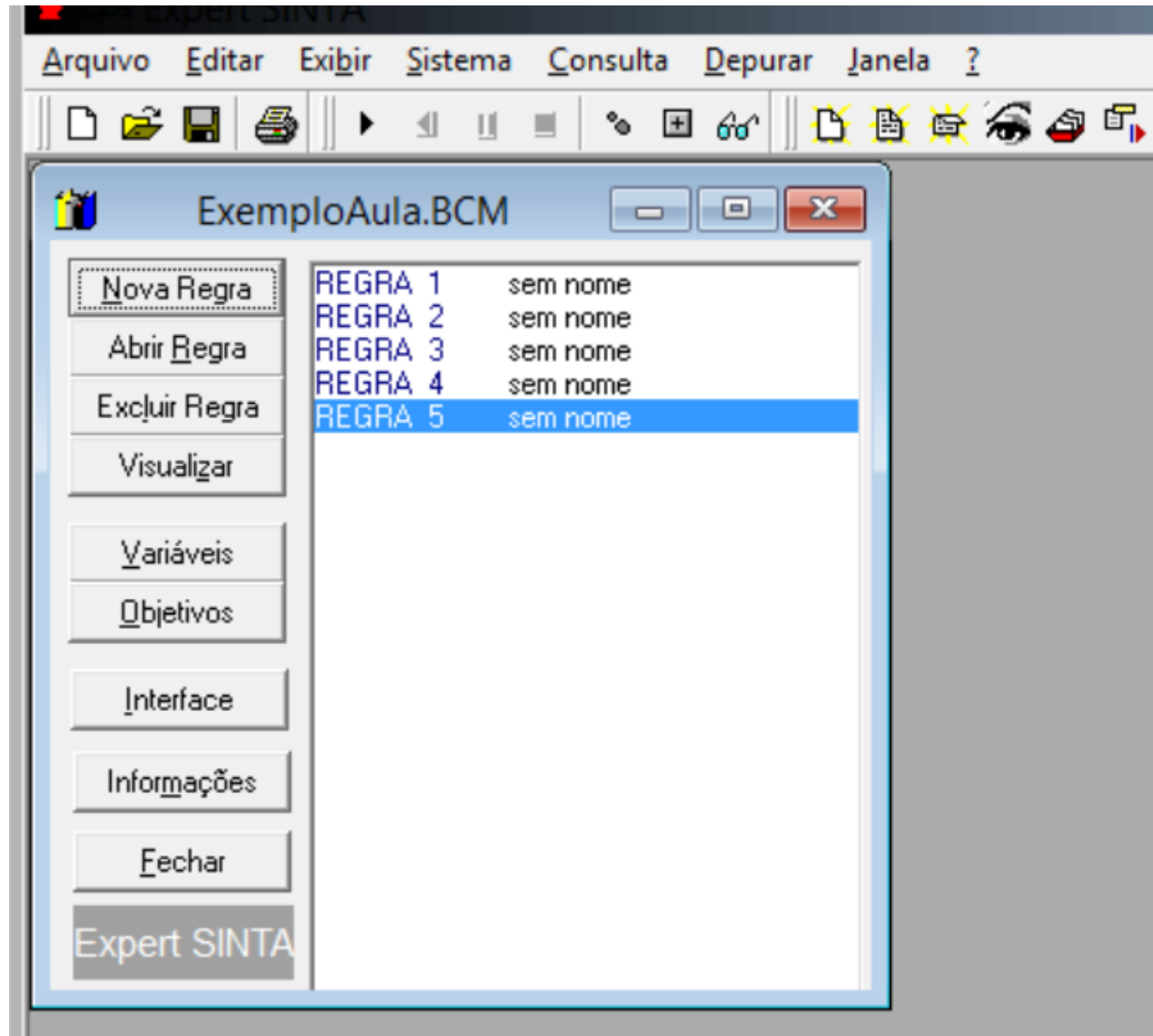
Definindo as regras (Base de Conhecimento)



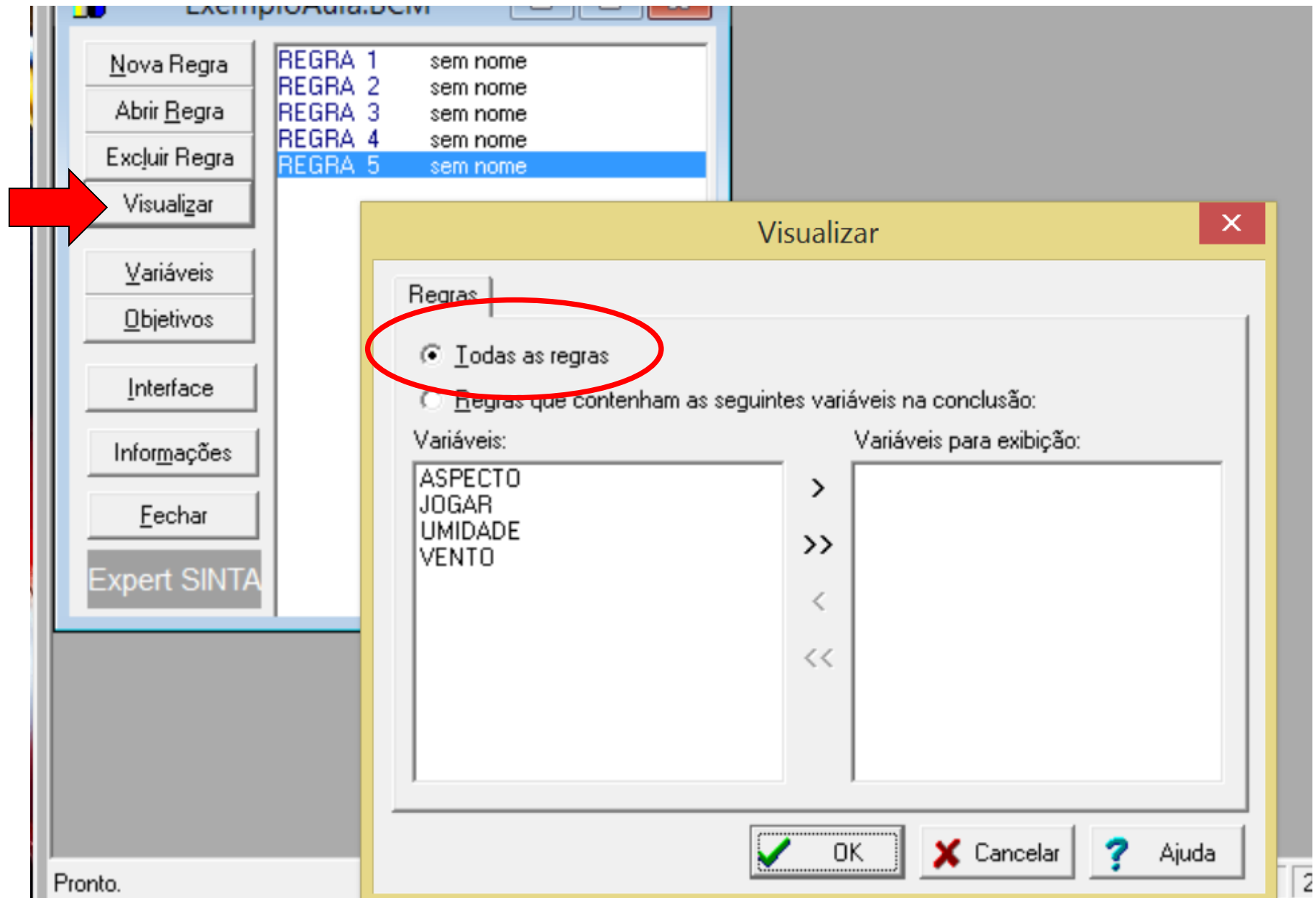
Definindo as regras (Base de Conhecimento)



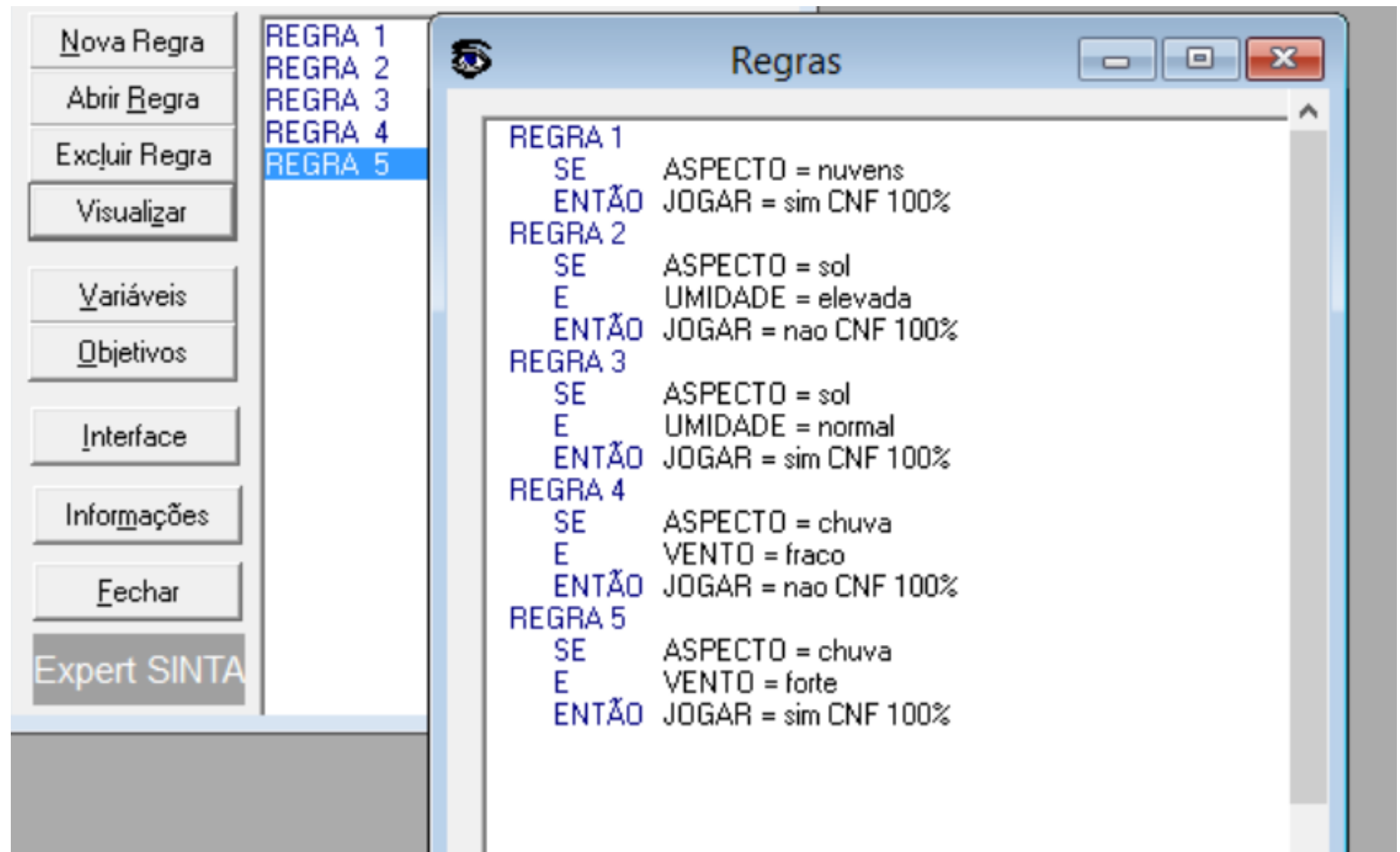
Definindo as regras (Base de Conhecimento)



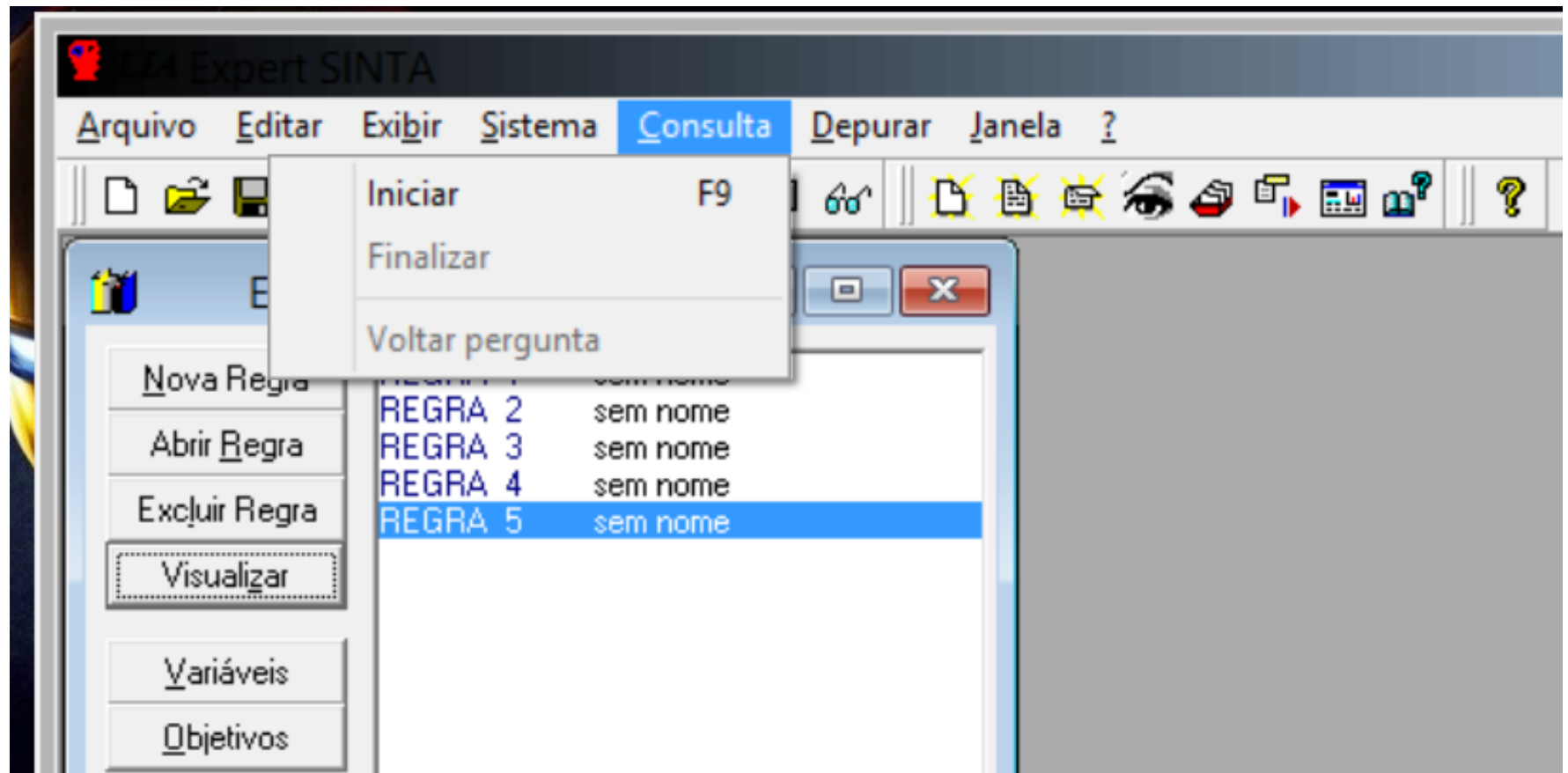
Visualizando as regras



Visualizando as regras



Executando (consulta)



Executando (consulta)

