

# Curso Aprendizado de Máquina

# Introdução – Sistema de Produção

- Um sistema de Produção proporciona uma estruturação dos programas computacionais de Inteligência Artificial de forma a facilitar a descrição e execução dos processos de busca das soluções de problemas.
- Um sistema de produção não é um sistema de produção fabril, como pode-se confundir.

# Características

- Aqui, um sistema de produção tem as seguintes características:
  - Uma base de conhecimento
  - Um conjunto de regras.
  - Uma estratégia de controle ao qual especifique a ordem ao qual as regras serão comparadas com a base de dados, bem como uma forma de solucionar conflitos e ambiguidades que possam surgir conforme as regras vão sendo aplicadas.
  - Um aplicador das regras.

# Solução de Problemas

- Dessa forma, para solucionar um problema, é necessário que seja fornecida uma definição precisa.
- Essa definição pode ser realizada a partir de espaços de estado do problema, e um conjunto de operadores para a movimentação naquele espaço.
- A outra forma de redução do problema é através do chamado Sistema de Produção ao qual representa-se o conhecimento a partir de regras e confronta-se essas regras com os dados e informações do ambiente do problema

# Conhecimento Procedimental VS Conhecimento Declarativo

- A representação declarativa é caracterizada pela qual o conhecimento deve ser especificado mas seu uso não é fornecido.
- Dessa forma, para utilizá-la é necessário especificar o que deve ser realizado com o conhecimento e como isso será feito.
- As declarações de implicações definem os caminhos do raciocínio e as declarações atômicas determinam os pontos de partida desses caminhos.
- Os caminhos dos raciocínios definem os caminhos de execução do programa da mesma forma como são realizados os desenvolvimentos tradicionais “IF-THEN-ELSE”.
- Já a representação procedimental especifica e determina que as informações de controle necessárias ao uso do conhecimento estão inseridas no próprio conhecimento. Para isso é necessário o uso de um interpretador que siga exatamente as instruções fornecidas no conhecimento.
- De forma sucinta, a grande diferença entre as abordagens declarativas e procedimentais esta em onde residem as informações de controle.
- Segundos estudos de IA, existem controvérsias se a representação declarativa ou procedimental é melhor ou pior, não havendo uma resposta precisa sobre o tema.
- O mais comum atualmente é uma abordagem que combinam todas as estruturas de representação de formas mais flexíveis.
- Esclarecemos um pouco mais dos Sistemas de Produção no conteúdo relacionado à Sistemas Especialistas baseados em Regras.

# Sistemas especialistas

- Sistemas especialistas
  - Definição de sistemas especialistas
  - Estrutura de um sistema especialista
  - Técnicas de extração de conhecimento
  - Exemplo de sistemas especialistas
- Raciocínio progressivo e regressivo

# Sistemas inteligentes

- Um sistema computacional pode ser visualizado como uma plataforma de hardware que dá suporte à execução de um software, o qual visa solucionar um problema específico.
- A grande característica/habilidade/capacidade humana que permitiu ao homo sapiens obter tamanha vantagem evolutiva sobre as demais espécies do planeta terra é sua inteligência.
- Logo, conceber uma máquina dotada dessa inteligência, corresponderia a gerar inteligência artificial (IA).

# Inteligência

- Quando uma pessoa tem conhecimento de um fato, certamente ela poderá extrair tudo que souber sobre aquele fato quando bem estender.
- Porém, na inteligência artificial existe um problema a mais quanto ao uso do conhecimento.
- Na IA temos basicamente dois problemas relacionados ao conhecimento:
  - como armazená-lo/representá-lo;
  - como utilizá-lo.



# Sistemas Baseados em Conhecimento

- Para atender a esta demanda surgem os Sistemas Baseados em Conhecimento (SBC), os quais aplicam mecanismos automatizados de raciocínio para a representação e inferência de conhecimento.

Especialista Humano



Sistema Especialista



# Sistema Especialista (SE)

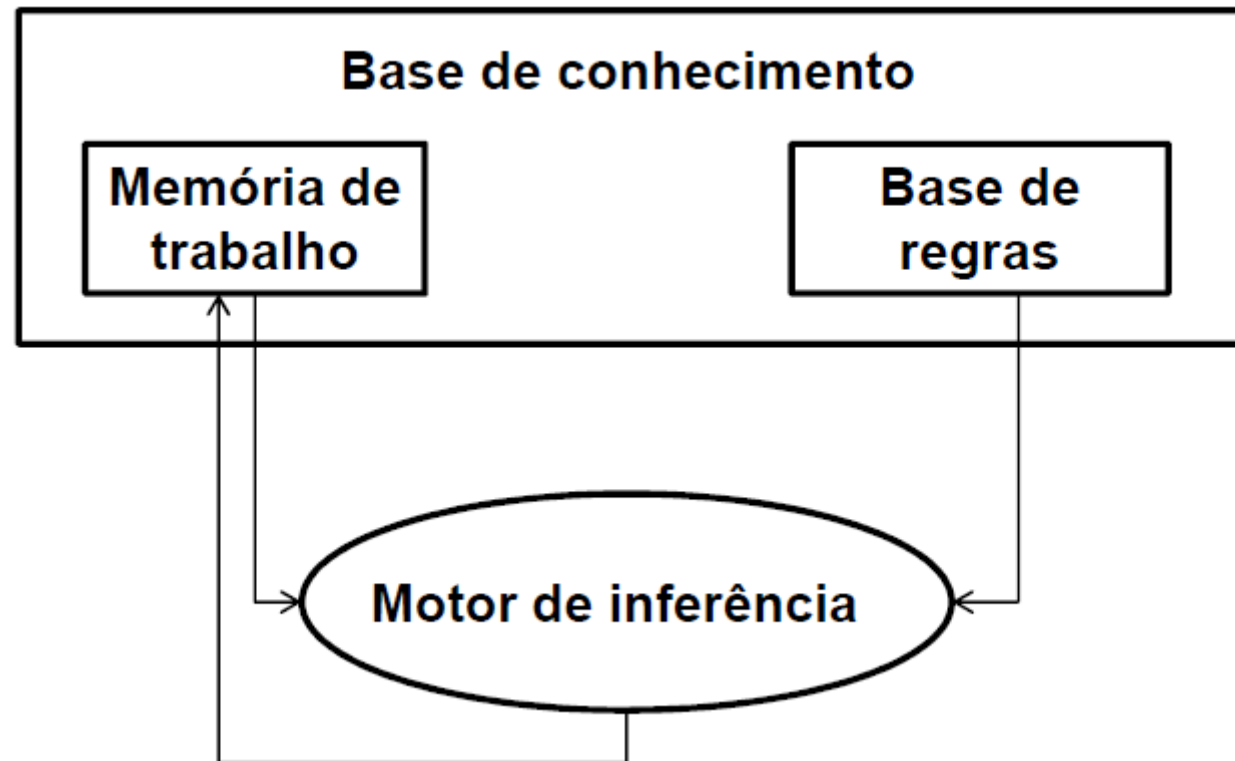
- Logo, um Sistema Especialista (SE) é **definido** como um programa que possibilita a utilização dos conhecimentos de um especialista humano através de uma máquina que permita o armazenamento e o seqüenciamento de informações e a autoaprendizagem.
- Um SE é capaz de processar informações não numéricas, apresentando conclusões sobre um determinado tema desde que devidamente orientado e “alimentado”.
- Outra característica comum nos sistemas especialistas é a existência de um mecanismo de raciocínio incerto que permita representar a incerteza a respeito do conhecimento do domínio.

# Conceitos Básicos

- Especialista
- Quem possui o conhecimento, experiência, métodos e a habilidade de aplicá-los para dar conselhos e resolver problemas.
- Engenheiro de conhecimento
- Guia a aquisição e representação do conhecimento especializado, bem como a implementação e refinamento do SE.

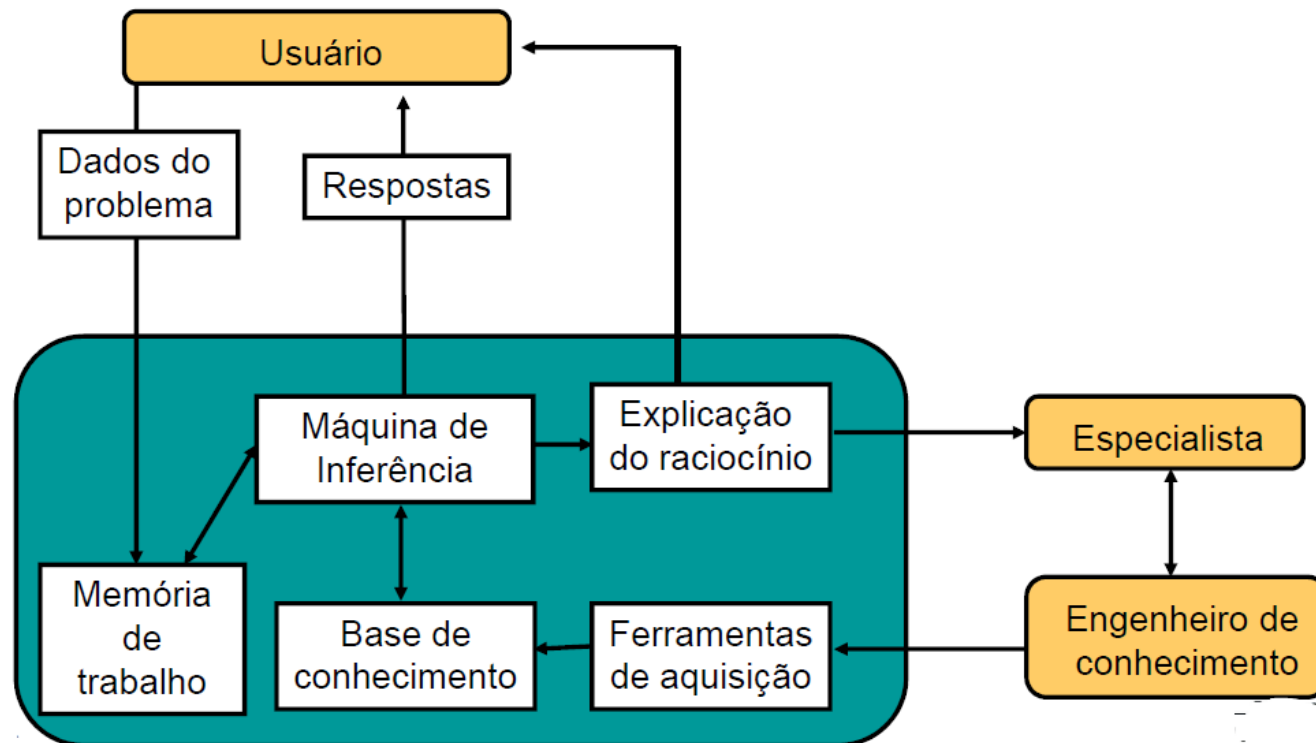
# Arquitetura do SE

- Um sistema especialista é estruturado com base em uma arquitetura similar à apresentada abaixo.



# Os elementos humanos e o SE

- Uma arquitetura mais ampla para um SE, abarcando os atores humanos envolvidos no sistema, é apresentada a seguir.



# Base de Conhecimento Clássica

- Contém: conhecimento, escrito em uma linguagem de representação, necessário para a formulação e solução do problema
- Conhecimento ontológico (regras, redes semânticas, ... )
  - ex. o homem é um animal
- Regras (estrutura de inferência)
  - ex. Todo animal tem uma mãe
- Fatos
  - ex. existem muitos meninos abandonados
- Heurísticas (para resolução de conflitos)
  - ex. prefira a regra disparada mais recentemente

# Memória de Trabalho

- Memória de Trabalho: é volátil e registra...
- descrição do problema em particular (instância)
- hipóteses e decisões intermediárias, sub-
- subobjetivos, etc.
- hipóteses e alternativas que o sistema já tenha produzido

# Motor ou Máquina de Inferência

- O Motor de Inferência controla a atividade do sistema, a qual ocorre em ciclos, cada ciclo consistindo em três fases:
- **Correspondência de dados**, onde as regras que satisfazem a descrição da situação atual são selecionadas;
- **Resolução de conflitos**, onde as regras que serão realmente executadas são escolhidas dentre as regras que foram selecionadas na primeira fase, e ordenadas;
- **Ação**, a execução propriamente dita das regras.



# Subsistema de Explicação

- Objetivo: Explicar o comportamento do SE através de questões como:
- Porque uma certa pergunta foi feita pelo SE ?
- Como a conclusão foi alcançada?
- Porque alguma alternativa foi rejeitada?
- Qual é o plano para alcançar a solução?
- Exemplo:
- Porque é preciso saber o preço?
- Resposta:
  - **REGRA #5**
  - **SE preço  $\leq$  salário E *dia = primeiro***
  - **ENTÃO**
  - **compra é efetuada**

# Juntando tudo temos...

- Em resumo um SE apresenta uma arquitetura com os seguintes componentes básicos: base de conhecimentos, mecanismos de inferência e explanação.
- A **base de conhecimento** reúne o conhecimento do especialista modelado conforme a representação do conhecimento escolhida para modelar o domínio em questão.
- O **mecanismo de inferência** examina o conteúdo da base de conhecimentos, decidindo a ordem em que se tiram as inferências. Assim, o mecanismo de inferência conduz a consulta com o usuário, transferindo os fatos e regras, utilizados durante uma consulta, para a memória de trabalho.
- O **módulo de explanação** é responsável pela descrição do raciocínio do sistema para o usuário.

# Passos para o desenvolvimento de um SE

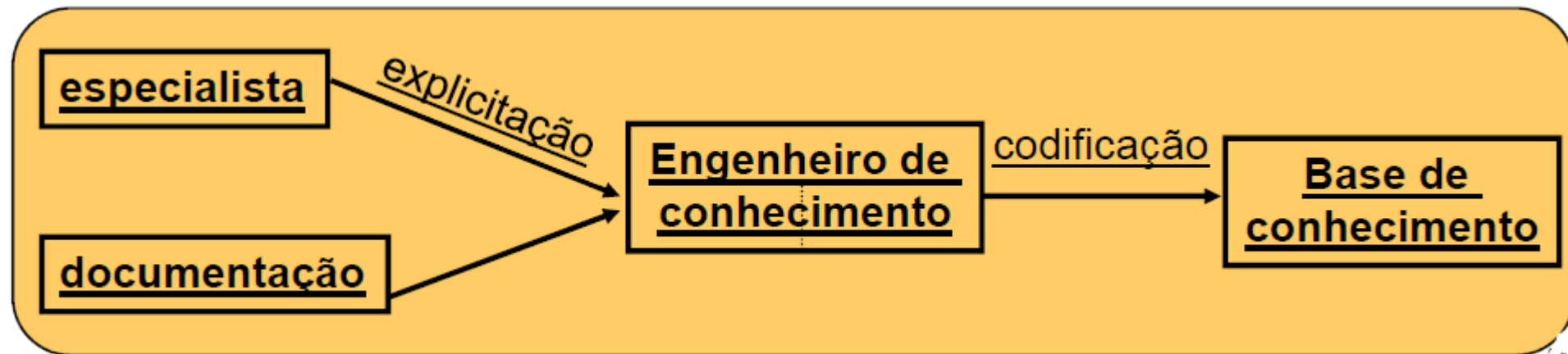
- 1) Construção da base de conhecimento
  - **Aquisição de conhecimento!**
  - Representação de conhecimento (formalização)
- 2) Implementação
  - Codificação
  - Construção do sistema de explicação, interface, etc.
- 3) Refinamento e validação

# Construção da base de conhecimento

- Aquisição/Explicitação de conhecimento acumulação, transferência e transformação de alguma fonte de conhecimento para um computador (base de conhecimento).
- Espécie de engenharia de requisitos mais complexa
- Pode originar-se de várias fontes: especialistas, livros e documentos, filmes, etc.
- Principais fases da aquisição
  - identificar características do problema
  - isolar os conceitos principais e suas relações (ontologia)
  - identificar inferências sobre estes conceitos

# Métodos de aquisição

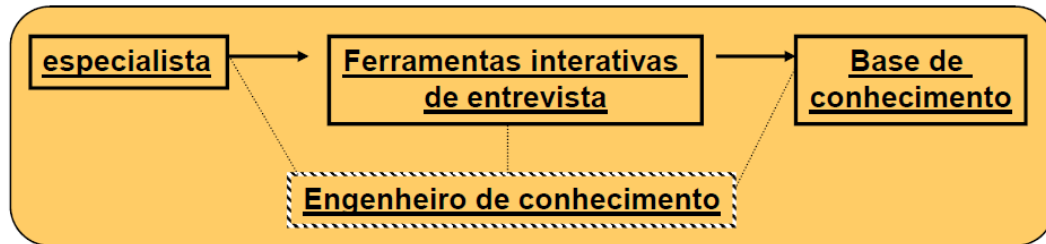
- 3 categorias: Manual, Semi-automático e
- Automático
- Manual
  - Entrevistas (estruturadas ou não estruturadas)
  - Tracking methods (análise de protocolos e observação)



# Métodos de aquisição

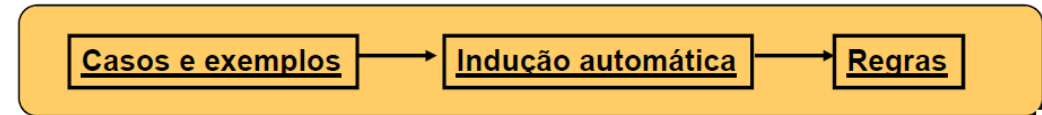
- Semi-automáticos

- ajuda ao especialista (grid repertory analysis)
- ajuda ao engenheiro de conhecimento (editores, documentadores, etc.)



- Automático:

- Aprendizado de máquina



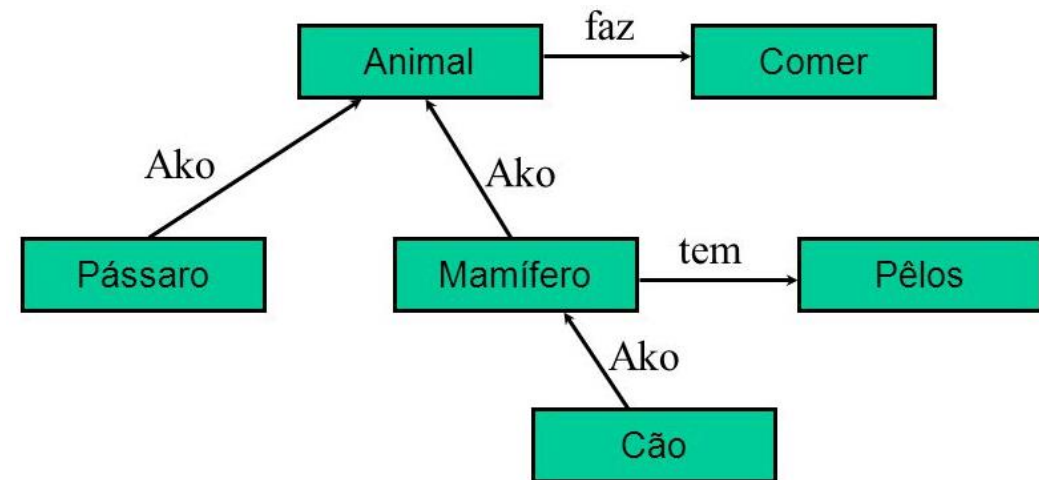
# Métodos de representação de conhecimento

- Ponto importante
- Exemplos de formalismos:
- Lógica : é a base para a maioria dos formalismos de representação de conhecimento, seja de forma explícita, como nos sistemas especialistas baseados na linguagem Prolog, seja mascarada na forma de representações específicas que podem facilmente ser interpretadas como proposições ou predicados lógicos.
- Ex.: Se A então B

# Métodos de representação de conhecimento

- Ponto importante
- Exemplos de formalismos:
- Redes semânticas: é um nome utilizado para definir um conjunto heterogêneo de sistemas.
- Em última análise, a única característica comum a todos estes sistemas é a notação utilizada:
- uma rede semântica consiste em um conjunto de nodos conectados por um conjunto de arcos.
- Os nodos em geral representam objetos e os arcos, relações binárias entre esses objetos.

Ako = relação entre classe e subclasse

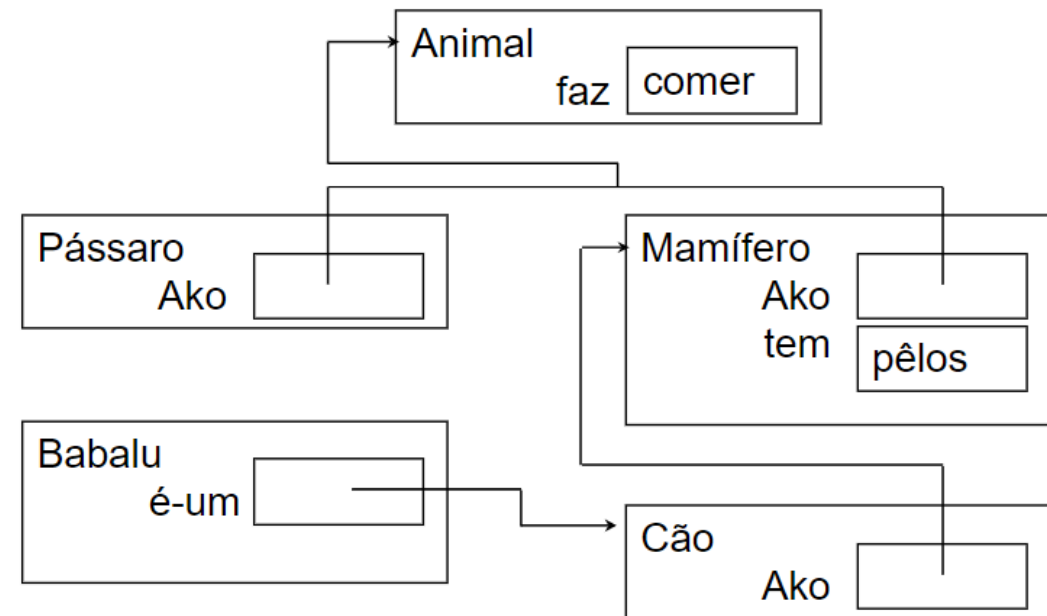


- Exemplos de regras extraídos dessa rede:
  - Pássaro é um animal
  - Cão é um mamífero
  - Mamífero é um animal
  - Cão é animal
  - Animais comem
  - Cão e Pássaros comem
  - Mamíferos tem pelos
  - Cão tem pelos



# Métodos de representação de conhecimento

- Exemplos de formalismos:
- Quadros ( frames ), e sua variação, os roteiros ( scripts ): foram introduzidos para permitir a expressão das estruturas internas dos objetos, mantendo a possibilidade de representar herança de propriedades como as redes semânticas.
- As idéias fundamentais destes métodos foram introduzidas por Marvin Minsky, em seu artigo A framework to represent knowledge (MINSKY, 1975).



Ako = relação entre classe e subclasse

# Motor de Inferência

- As principais características do motor de inferência disponível em um SE dizem respeito às seguintes funcionalidades:
  - método de raciocínio,
  - estratégia de busca,
  - resolução de conflito e
  - representação de incerteza.

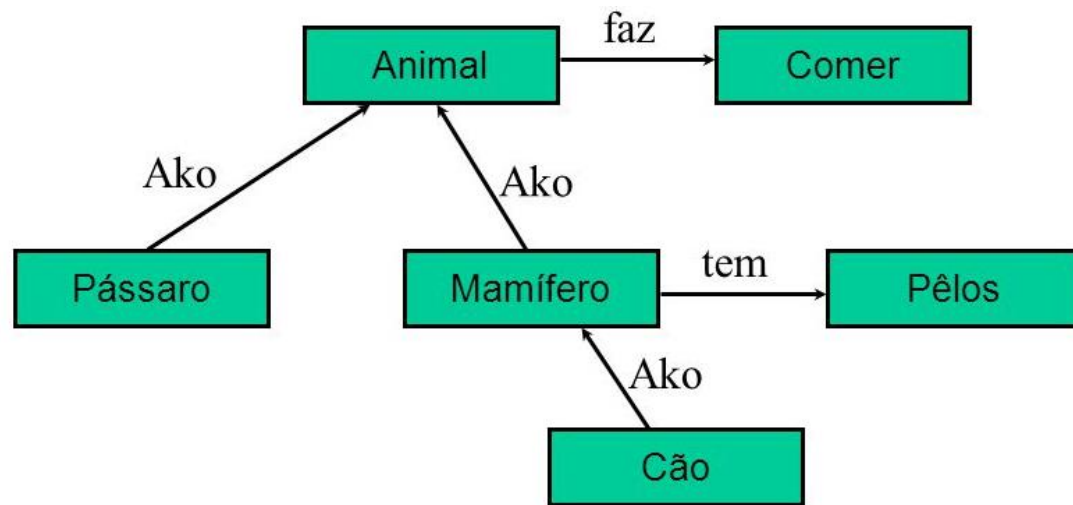
# Motor de Inferência - Modo de raciocínio

- Existem basicamente dois modos de raciocínio aplicáveis a regras de produção:
- encadeamento progressivo ou encadeamento a frente (do inglês, forward chaining ),
- e encadeamento regressivo ou encadeamento para trás (do inglês, backward chaining ).

# Motor de Inferência - Modo de raciocínio

- No encadeamento regressivo, também Chamado encadeamento dirigido por objetivos, o comportamento do sistema é controlado por uma lista de objetivos.
- Um objetivo pode ser satisfeito diretamente por um elemento da memória de trabalho, ou podem existir regras que permitam inferir algum dos objetivos correntes, isto é, que contenham uma descrição deste objetivo em suas partes direitas.
- As regras que satisfazem esta condição têm as instâncias correspondentes às suas partes esquerdas adicionadas à lista de objetivos correntes.
- Caso uma dessas regras tenha todas as suas condições satisfeitas diretamente pela memória de trabalho, o objetivo em sua parte direita é também adicionado à memória de trabalho.
- Um objetivo que não possa ser satisfeito diretamente pela memória de trabalho, nem inferido através de uma regra, é abandonado.
- Quando o objetivo inicial é satisfeito, ou não há mais objetivos, o processamento termina.

# Motor de Inferência - Modo de raciocínio



- Determinar quem Come ?
- O1 = [Animal]
- Qual as instâncias de animal
- O1 = [Pássaro, Mamífero]
- Qual as instâncias de pássaro?
- O1 = [Pássaro, Mamífero]
- Qual as instâncias de mamífero?
- **Logo** O1 → [Pássaro, Cão]

# Motor de Inferência - Estratégia de busca

- Uma vez definido o tipo de raciocínio/encadeamento, o motor de inferência necessita ainda de uma estratégia de busca para guiar a pesquisa na memória de trabalho e na base de regras.
- Este tipo de problema é conhecido como busca em espaço de estados.
- Este tópico foi um dos primeiros estudados em IA, no contexto de solução de problemas (do tipo quebra-cabeças) e jogos por computador (damas, xadrez, etc.).

# Resolução de conflito

- Ao terminar o processo de busca, o motor de inferência dispõe de um conjunto de regras que satisfazem à situação atual do problema, o chamado conjunto de conflito.
- Se esse conjunto for vazio, a execução é terminada; caso contrário, é necessário escolher que regras serão realmente executadas e em que ordem.
- Os métodos de resolução de conflito mais utilizados ordenam as regras de acordo com os seguintes critérios:
  - prioridades atribuídas estaticamente;
  - características da estrutura das regras como complexidade, simplicidade e especificidade;
  - características dos dados associados às regras como o tempo decorrido desde sua obtenção, sua confiabilidade ou seu grau de importância;
  - e, finalmente, seleção ao acaso.

# Um banco de conhecimento

## Ciclos



## Automóvel





# Um banco de conhecimento

## Ciclos



- Ciclo:
  - Se  $\text{num-rodas} < 4$
  - Então **veículoTipo = *ciclo***

## Automóvel



- Automóvel:
  - Se  $\text{num-rodas} = 4$
  - E **motor = sim**
  - Então **veículoTipo = *automóvel***

# Um banco de conhecimento

## Ciclos



- Ciclo:
  - Se **num-rodas**<4
  - Então **veículoTipo**=*ciclo*

- Bicicleta:
  - Se **veículoTipo**=ciclo
  - E **num-rodas**=2
  - E **motor**=não
  - Então **veículo**=*Bicicleta*
- Triciclo:
  - Se **veículoTipo**=ciclo
  - E **num-rodas**=3
  - E **motor**=não
  - Então **veículo**=*Triciclo*
- Motocicleta:
  - Se **veículoTipo**=ciclo
  - E **num-rodas**=2
  - E **motor**=sim
  - Então **veículo**=*Motocicleta*

# Um banco de conhecimento

- CarroSport:
  - Se **veículoTipo=automóvel**
  - E **tamanho=pequeno**
  - E **num-portas=2**
  - Então **veículo=CarroSport**
  - MiniVan:
    - Se **veículoTipo=automóvel**
    - E **tamanho=médio**
    - E **num-portas=3**
    - Então **veículo=MiniVan**
- UtilitárioSport:
  - Se **veículoTipo=automóvel**
  - E **tamanho=grande**
  - E **num-portas=4**
  - Então **veículo=UtilitárioSport**



- Automóvel:
  - Se **num-rodas=4**
  - E **motor=sim**
  - Então **veículoTipo=automóvel**

# Base de Regras para veículos

- Ciclo:
  - Se **num-rodas**<4
  - Então **veículoTipo**=*ciclo*
- Bicicleta:
  - Se **veículoTipo**=ciclo
  - E **num-rodas**=2
  - E **motor**=não
  - Então **veículo**=*Bicicleta*
- Motocicleta:
  - Se **veículoTipo**=ciclo
  - E **num-rodas**=2
  - E **motor**=sim
  - Então **veículo**=*Motocicleta*
- Triciclo:
  - Se **veículoTipo**=ciclo
  - E **num-rodas**=3
  - E **motor**=não
  - Então **veículo**=*Triciclo*
- Automóvel:
  - Se **num-rodas**=4
  - E **motor**=sim
  - Então **veículoTipo**=*automóvel*
- CarroSport:
  - Se **veículoTipo**=automóvel
  - E **tamanho**=pequeno
  - E **num-portas**=2
  - Então **veículo**=*CarroSport*
- UtilitárioSport:
  - Se **veículoTipo**=automóvel
  - E **tamanho**=grande
  - E **num-portas**=4
  - Então **veículo**=*UtilitárioSport*
- MiniVan:
  - Se **veículoTipo**=automóvel
  - E **tamanho**=médio
  - E **num-portas**=3
  - Então **veículo**=*MiniVan*

# Como a máquina de inferência Opera

- Meta-regras
  - Se R1 e R2 podem ser disparadas, escolha R1
  - Se R1 e R2 podem ser disparadas e R1 foi disparada mais recentemente que R2, escolha R2
- Fatos
  - Veículo1: tamanho=pequeno; numportas= 2; motor=sim
  - Veículo2: num-rodas=2; motor=não
- Dos dados à conclusão
  - Parte dos fatos na Base de Regras (BR) e na memória de trabalho, buscando quais regras eles satisfazem, para produzir assim novas conclusões (fatos) e/ou realizar ações.
- Três etapas: Busca, Casamento (unificação), Resolução de conflito

# Encadeamento progressivo Algoritmo

1. Armazena as regras da base de conhecimento (BC) na máquina de inferência (MI) e os fatos na memória de trabalho (MT);

2. Adiciona os dados/conhecimento iniciais (fatos) à memória de trabalho; obs.: esses dados podem ser fornecidos pelo usuário do sistema

3. Compara o antecedente das regras com os fatos na MT.

Todas as regras cujo antecedente “casa” (unifica) com esses fatos podem ser disparadas e são colocadas no conjunto de conflito;

4. Usa o procedimento de resolução de conflito para selecionar uma única regra desse conjunto;

5. Dispara a regra selecionada e verifica o seu conseqüente:

- 5a) se for um fato atualiza a MT
- Repete os passos 3, 4 e 5 até o conjunto de conflito se tornar vazio.
- 5b) se for uma ação
- Chama o procedimento que realiza a ação escolhida atualiza a MT
- Volta para o passo 2 (para obter novos dados do ambiente).

# Encadeamento progressivo

## Busca e Casamento (unificação)

- O algoritmo tenta casar (unificar) as premissas das regras selecionadas com os fatos na memória de trabalho:
- MT1: num-rodas=4, motor=sim, num-portas=3, tamanho=médio
  - MI (regras da BC): Se num-rodas=4 E motor=sim
- Então veículoTipo=automóvel
  - MT2: MT1 + veículoTipo=automóvel
- Geralmente, o antecedente de cada regra selecionada é comparado com os fatos na MT usando busca gulosa (best-first)

# Encadeamento progressivo: Resolução de conflitos

- Resolução de conflitos heurística geral para escolher um subconjunto de regras a disparar
- Exemplos:
  - Não duplicação: não executar a mesma regra com os mesmos argumentos duas vezes.
  - Prioridade de operação: preferir ações com prioridade maior
  - Recency (“recenticidade”): preferir regras que se referem a elementos da Memória de Trabalho criados recentemente.
  - Especificidade: preferir regras que são mais específicas.



# Encadeamento progressivo:

## Exemplo no domínio dos veículos

- Carregar a BR de veículos no MI e atribuir valores iniciais para algumas variáveis, guardando esses fatos na MT.
- Fatos iniciais:
  - num-rodas=4,
  - motor=sim,
  - num-portas=3,
  - tamanho=médio
- Exercício: Aplique os passos estudados e infira um resultado.

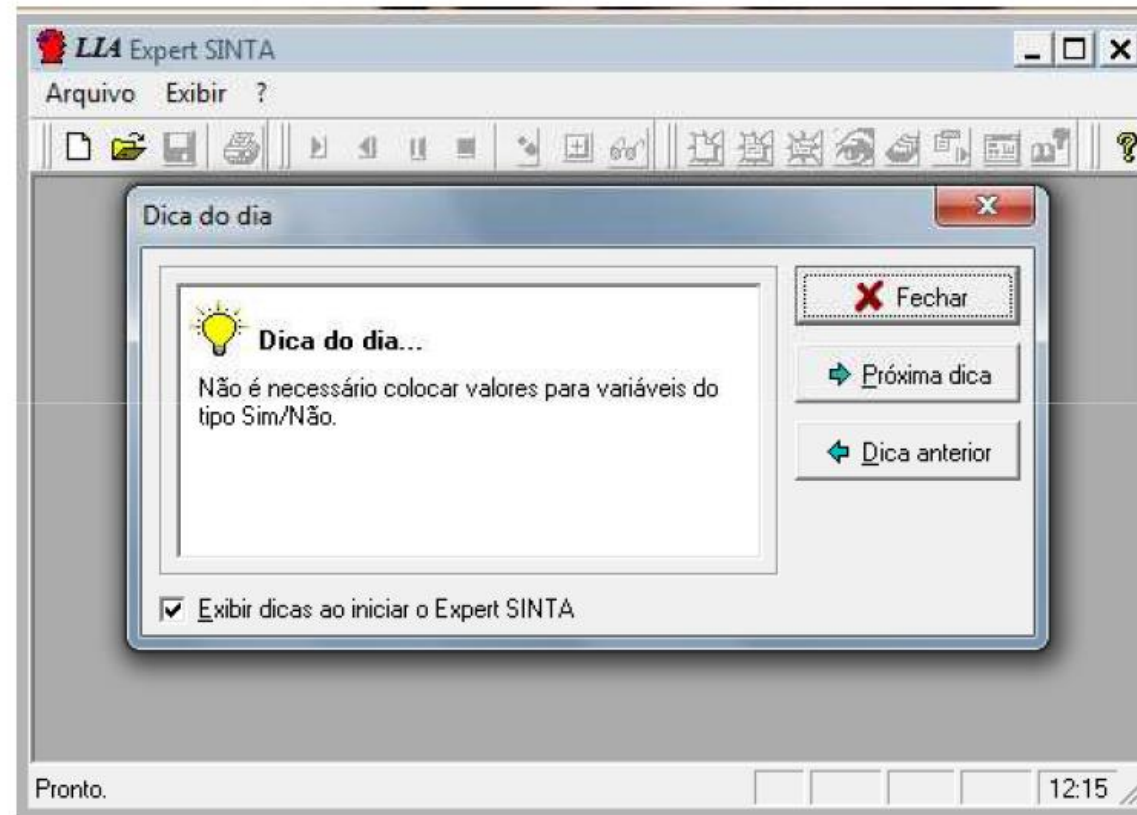
# Expert Sinta

- O Expert SINTA foi criado de modo a permitir ao próprio analista de conhecimento implementar a base desejada.
- O usuário não necessita do menor conhecimento de programação, apenas dever saber como interagir em ambientes visuais.
- Uma base de conhecimento no Expert SINTA envolve os seguintes conjuntos de atributos a serem indicados pelo projetista da base:
  - Variáveis
  - Regras
  - Perguntas
  - Objetivos
  - Informações adicionais
- Como padrão, O Expert SINTA grava as bases de conhecimento geradas em arquivos \*.BCM.
- Download:
- <ftp://ftp.lia.ufc.br/sinta/sinta.zip>

# Tutoriais expert sinta

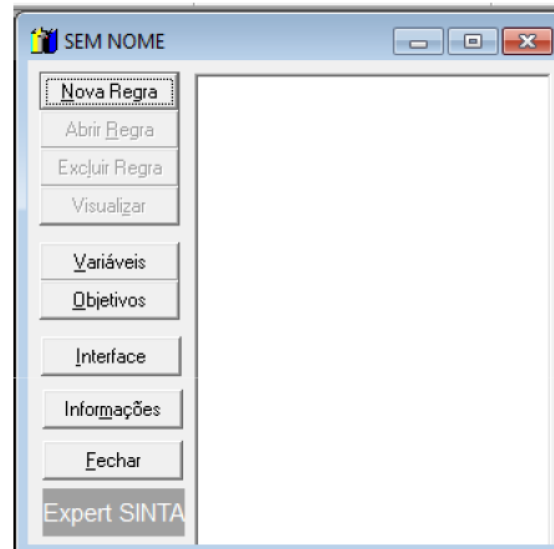
- <https://iaexpert.com.br/index.php/2016/09/13/ferramentas-para-ia-expert-sinta/>
- Tutorial em vídeo
  - IAEXPERT – Aula 1 - <https://www.youtube.com/watch?v=bqzH8kRYmDY>
  - IAEXPERT – Aula 2 - <https://www.youtube.com/watch?v=yMAgPclgEqU>
  - Marcelo Santos- <https://www.youtube.com/watch?v=S4uAHWQBaWE>

# Expert Sinta



Tela Inicial

# Expert Sinta



A janela “Knowledge-in-a-box” (KIB)



Deixe o cursor do mouse momentaneamente sobre cada botão para exibir uma descrição breve.

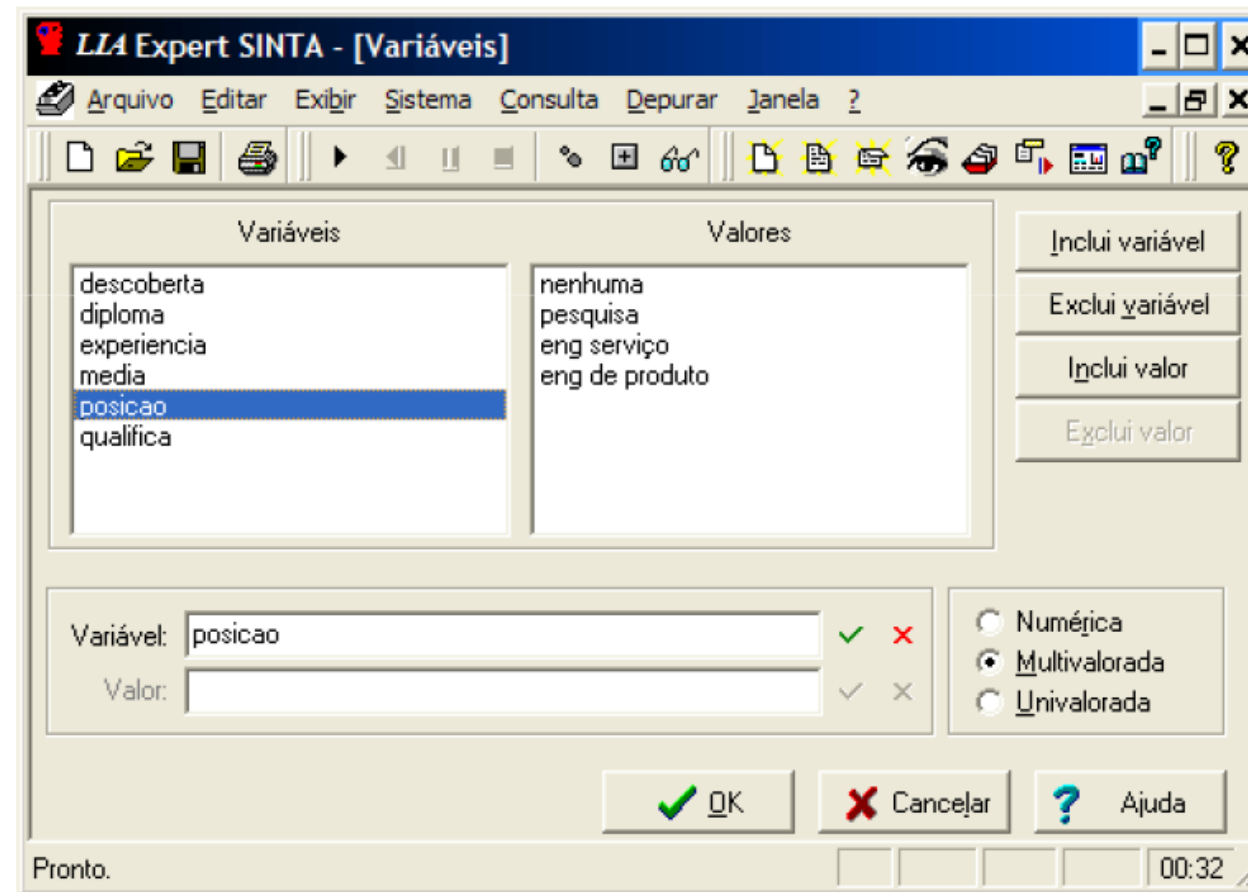
Opções relativas à janela KIB.

# Expert Sinta

- Passos para construção do SE utilizando o Expert SINTA:
- Estabelecer variáveis
- Estabelecer objetivos (variáveis de saída)
- Gerar as regras
- Estabelecer interfaces (perguntas a serem feitas)

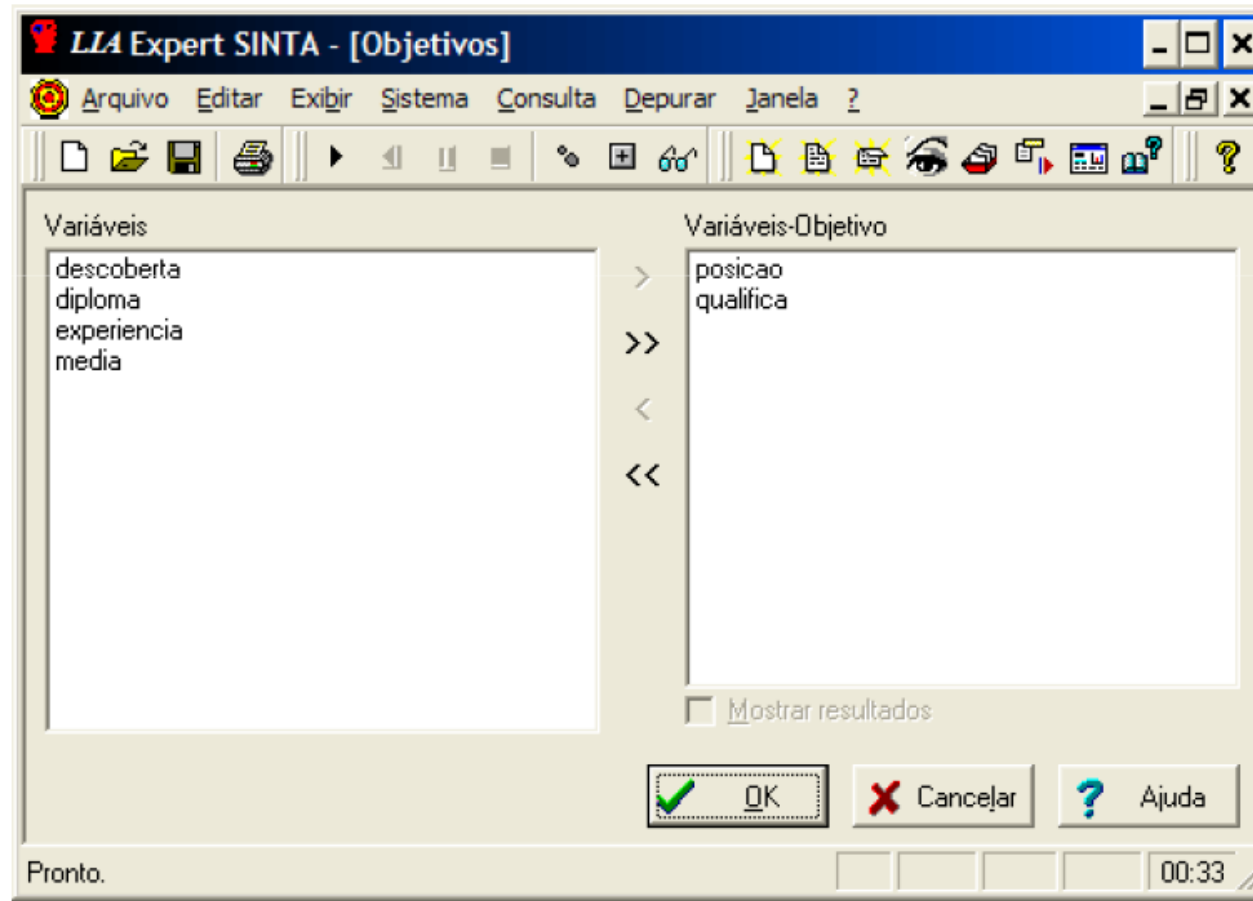
# Expert Sinta

- **Definição de variáveis:** Antes de podermos criar regras, é necessário que todas as variáveis utilizadas, bem como seus respectivos valores, sejam criados.



# Expert Sinta

- **Definição de objetivos:** O objetivo de uma consulta a um especialista é encontrar a resposta para um determinado problema. Assim também o é um sistema especialista.





# Simulando encadeamento para frente

- O Expert SINTA trabalha naturalmente com encadeamento para trás, mas é possível manipular a máquina de inferência de modo a simular encadeamento para frente.
- Para tal, as variáveis que se deseja encontrar inicialmente devem ser colocadas na lista de objetivos.
- Obviamente, como não é desejado que uma janela de resultados apareça a cada variável encontrada, é possível desabilitar essa exibição simplesmente desmarcando a caixa “Mostrar resultados” relativo à variável desejada na janela de edição de objetivos.

# Expert Sinta

## Geração de Regras:.

**Nova Regra**

Informação

Ordem da regra: 1

Modelo: NENHUM

OK Cancelar Ajuda

**Editar Regra**

Item de regra:

Conectivo

E OU Não Inserir Fechar Ajuda

**LLA Expert SINTA - [Regra 1]**

Arquivo Editar Exibir Sistema Consulta Depurar Janela ?

Nome da regra: R1 Ordem: 1

```
SE
  diploma = Não
ENTÃO
  posicao = nenhuma CNF 100%
```

Alterar Excluir Incluir Nova... OK Cancelar Ajuda

Pronto. 00:35

# Fatores de confiança

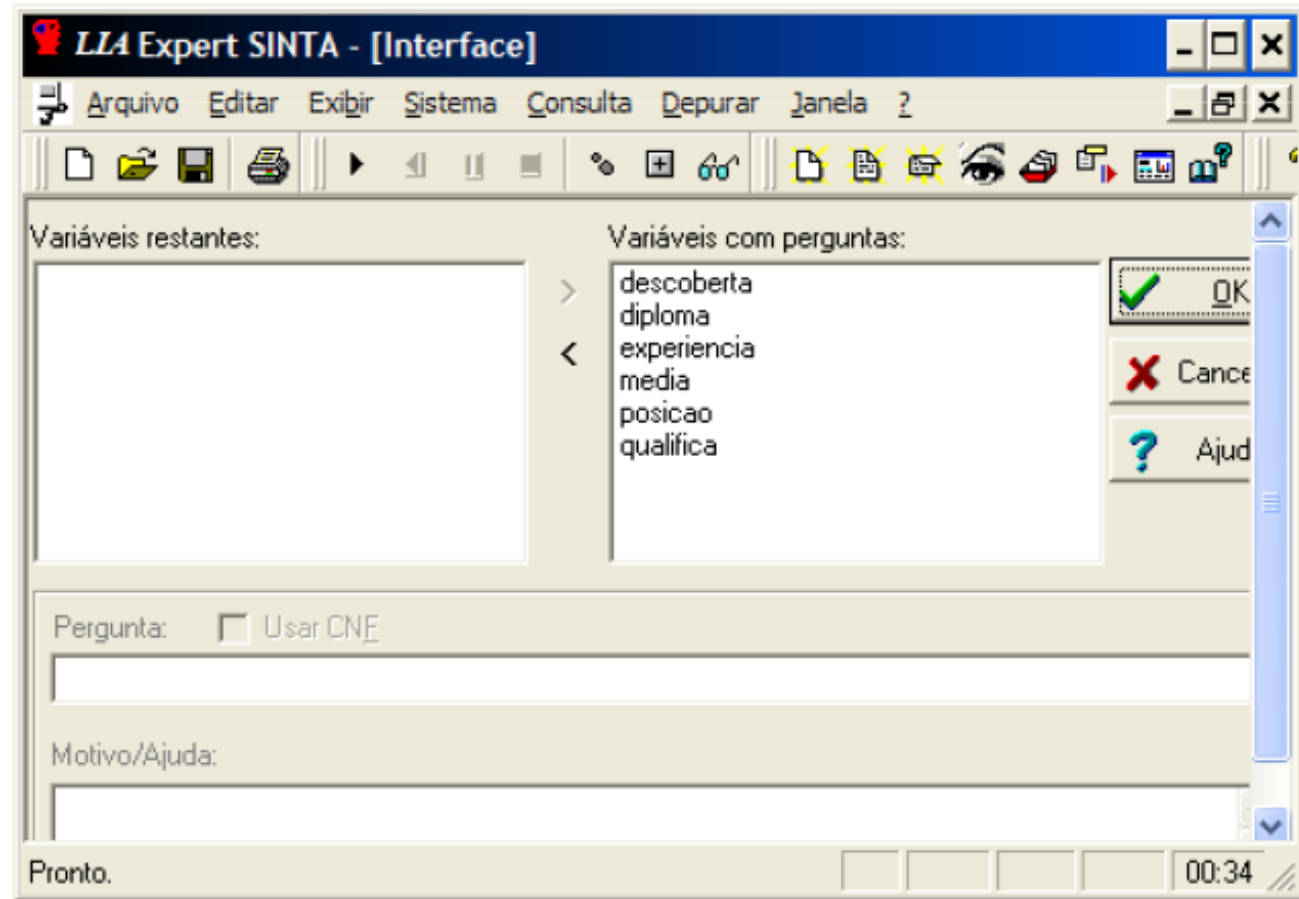
- As variáveis podem receber apenas valores determinísticos (“sim” e “não”).
- Porém, é comum que não queiramos determinar com exatidão o valor que cada variável pode receber.
- O conhecimento humano não é determinístico. Não há especialista que sempre se encontre em condições de afirmar determinada conclusão com certeza absoluta.
- Graus de confiança são freqüentemente atribuídos às suas respostas, principalmente quando existe mais de uma.
- Este torna-se um dos mais fortes pontos críticos na elaboração de uma representação computacional do saber humano.
- Existem duas correntes de pensamento:
  - a que utiliza fórmulas estatísticas rigorosas, com teoria das probabilidades (FREQUENCIALISTA);
  - e aquela que utiliza uma abordagem da teoria das possibilidades sobre os fatores de certeza sem uma base matemática forte (SUBJETIVISTA).
- O Expert SINTA (na versão 1.1) utiliza uma abordagem subjetiva possibilista, ou seja, mais simples.

# Expert Sinta

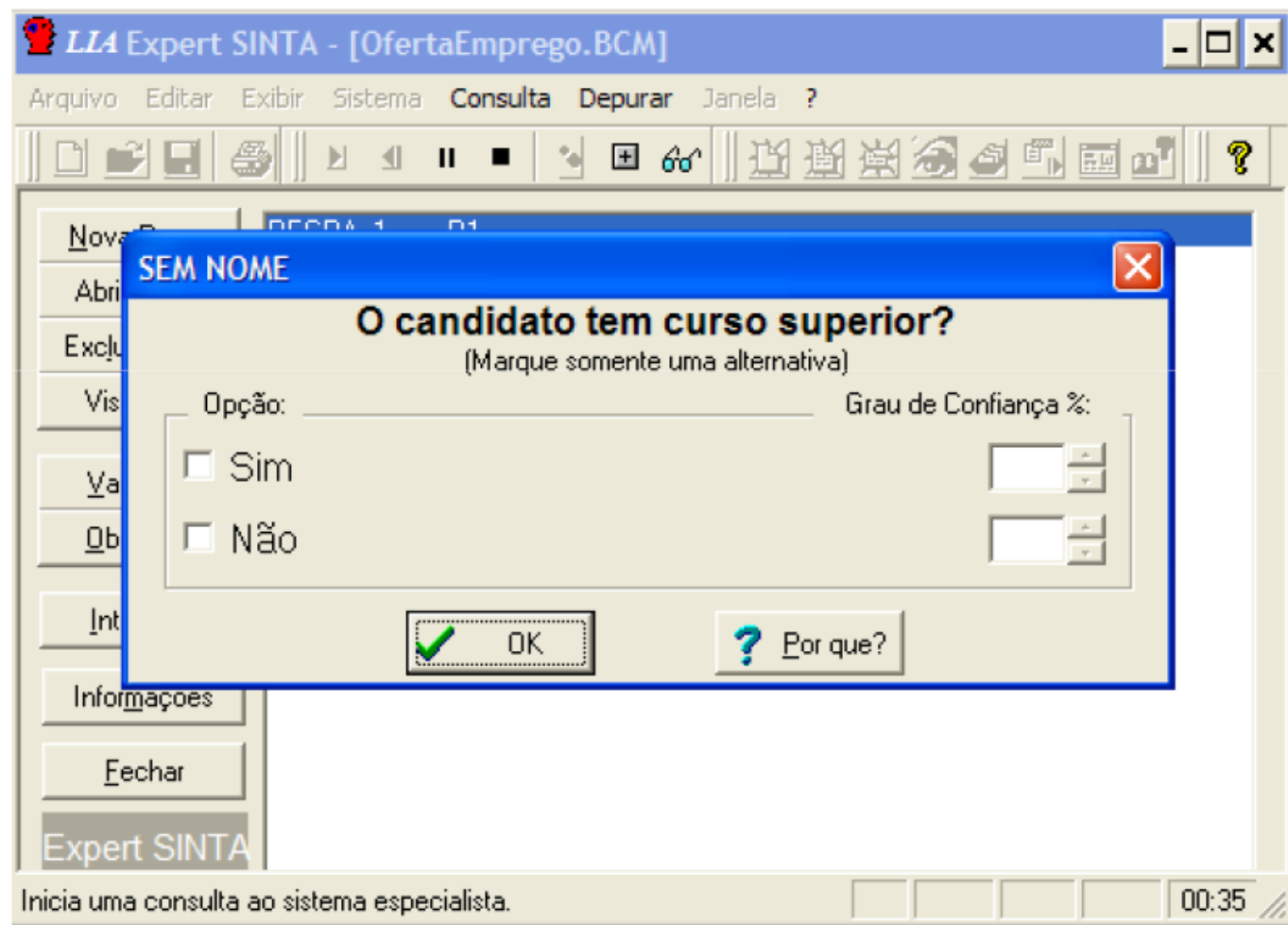
- **Estabelecer interfaces:** Um sistema especialista implementado com o Expert SINTA comunica-se com o usuário final através de menus de múltipla escolha (ou escolha simples, se a variável em questão for univalorada).
- Estes menus são construídos automaticamente pela shell, mas alguns detalhes devem ser fornecidos pelo criador da base.

# Expert Sinta

- A pergunta realizada pela máquina de inferência deve ser personalizada para que seja inteligível.
- Quando o Expert SINTA necessita perguntar o valor de determinada variável, ele cria uma sentença genérica do tipo
- “Qual o valor de x?”, onde x é o nome da variável.



# Expert Sinta - Executando



# Expert Sinta - Resultado

Resultados

posicao

Fechar

Ajuda

Valor	CNF (%)
eng de produto	100

Resultados / Histórico / Todos os valores / O sistema

# Expert Sinta - Exemplo

- **Exemplo:**
- Como exemplo construiremos um SE capaz de diagnosticar possíveis problemas em um carro.
- Iniciaremos pela definição das variáveis:
- carburador tem gasolina (univalorada)
- diagnostico (multivalorada)
- Valores
  - **bateria ou cabos**
  - **Ignicao**
  - **carro sem combustivel**
  - **mangueira bloqueada**
  - **vela de ignicao**
- farois ligam (univalorada)
- motor liga (univalorada)
- tanque vazio (univalorada)



# Expert Sinta - Exemplo

- Objetivos:
  - Diagnostico
- Regras:
- **REGRA 1**
  - **SE**
    - tanque vazio = sim
  - **ENTAO**
    - diagnostico = Carro sem combustivel CNF 100%
- **REGRA 2**
  - **SE**
    - tanque vazio = Nao
    - E carburador tem gasolina = Nao
  - **ENTAO**
    - diagnostico = magueira bloqueada CNF 100%
- **REGRA 3**
- **SE**
  - tanque vazio = Nao
  - E motor\_liga = Nao
  - E farois ligam = nao
- **ENTAO**
  - diagnostico = Bateria ou cabos CNF 100%

# Expert Sinta - Exemplo

- Objetivos:

- Diagnostico

- Regras:

- **REGRA 4**

- **SE**

- tanque vazio = Nao
    - E motor\_liga = Nao
    - E farois ligam = Sim

- **ENTAO**

- diagnostico = Ignicao CNF 100%

- **REGRA 5**

- **SE**

- tanque vazio = Nao
    - E motor\_liga = sim
    - E farois ligam = nao

- **ENTAO**

- diagnostico = vela de ignição CNF 100%

# Expert Sinta Exemplo

- Interface (**VARIAVEIS COM PERGUNTAS**)
  - carburador tem gasolina = O carburador tem gasolina ?
  - Faróis ligam = Os faróis ligam ?
  - motor liga = O motor liga ?
  - tanque vazio = O tanque esta vazio?