

PLP C

Projeto: Solução do teste de Einstein com o uso da linguagem Prolog

PARTE A do Projeto: Pré-Projeto

O problema de Einstein, consiste em uma problema lógico a ser resolvido que deve consistir com os fatos e regras a serem seguidos.

Resolução do problema analiticamente:

Passos a serem seguidos:

- O Inglês vive na casa Vermelha.
- O Sueco tem Cachorros como animais de estimação.
- O Dinamarquês bebe Chá.
- A casa Verde fica do lado esquerdo da casa Branca.
- O homem que vive na casa Verde bebe Café.
- O homem que fuma Pall Mall cria Pássaros.
- O homem que vive na casa Amarela fuma Dunhill.
- O homem que vive na casa do meio bebe Leite.
- O Norueguês vive na primeira casa.
- O homem que fuma Blends vive ao lado do que tem Gatos.
- O homem que cria Cavalos vive ao lado do que fuma Dunhill.
- O homem que fuma Blue Master bebe Cerveja.
- O Alemão fuma Prince.
- O Norueguês vive ao lado da casa Azul.
- O homem que fuma Blends é vizinho do que bebe Água.

(site https://rachacuca.com.br/logica/problemas/teste-de-einstein/#google_vignette)

Pergunta: quem cria os peixes?

Primeiro, pensamos nas coisas que não podem ser mudadas, como:

O inglês deve viver numa casa vermelha.

O sueco tem que criar cachorros.

O dinamarquês bebe chá.

O homem que vive na casa verde bebe café.

Quem fuma Pall Mall cria pássaros.

O morador da casa do meio (casa 3) bebe leite.

O norueguês vive na primeira casa.

Quem fuma Blue Master bebe cerveja.

O alemão fuma Prince.

1. Da pista do norueguês (casa 1) e "vive ao lado da casa azul": a casa 2 é azul obrigatoriamente.

2. Do par verde→branca e da cor já fixada na casa 2 (azul), os únicos pares possíveis são (3→4) ou (4→5). Porém a casa 3 bebe leite e a verde bebe café; logo não pode ser (3→4). Portanto, casa 4 = verde (café) e casa 5 = branca.

3. Entre as cores restantes para 1 e 3 estão amarela e vermelha. Pela pista "amarela → Dunhill" e "inglês → vermelha": como a casa 1 tem o norueguês, ela não pode ser vermelha. Assim, casa 1 = amarela (Dunhill) e casa 3 = vermelha (inglês).

4. "Cavalos ao lado de Dunhill": como Dunhill está na casa 1, os cavalos ficam na casa 2 (único vizinho da 1).

5. Marcas/bebidas que já sabemos:

Casa 1: Dunhill; bebida ainda indefinida.

Casa 3: (vermelha); bebida = leite; marca indefinida.

Casa 4: (verde); café; marca indefinida.

Par obrigatório: Blue Master → cerveja (ainda sem casa).

6. Analisando Blends: quem fuma Blends precisa ter (i) um vizinho com gatos e (ii) um vizinho que bebe água. Se tentarmos pôr Blends na casa 2, então:

Pela vizinhança com 1 e 3, um deles beberia água. Como a casa 3 já é leite, a água ficaria na casa 1.

O outro vizinho teria gatos. Como a casa 2 tem cavalos, os gatos não podem estar nela; restam 1 ou 3; logo os gatos ficam coerentemente na casa 1. Essa escolha é consistente

com todas as demais pistas, então fixamos Blends na casa 2, água na casa 1 e gatos também na casa 1.

7. Como a água está na casa 1, a dupla obrigatória Blue Master + cerveja não pode ser da casa 1. Como na casa 3 já é obrigatoriamente leite e 4 é café (verde + café), sobra a casa 5 para Blue Master + cerveja.

8. Pela relação Prince ↔ alemão, e sabendo que 5 já é Blue Master, 1 é Dunhill e 2 é Blends, a marca Prince só pode ficar em 4 ou 3. Mas na 3, já está ocupada pelo inglês, então o alemão ficará na casa 4, e a marca Prince também.

9. "Pall Mall → pássaros": a única casa ainda sem marca (entre 1: Dunhill, 2: Blends, 4: Prince, 5: Blue Master) é a casa 3, então casa 3 = Pall Mall e cria pássaros.

10. Nacionalidades restantes (além de norueguês na 1, inglês na 3, alemão na 4): dinamarquês e sueco para as casas 2 e 5. Como a casa 5 bebe cerveja (Blue Master), não pode ser o dinamarquês (que bebe chá). Logo casa 2 = dinamarquês, consequentemente a bebida também é preenchida com chá e casa 5 = sueco.

11. O sueco cria obrigatoriamente cachorros, portanto, casa 5 (casa do sueco) = cachorros. Já temos cavalos na 2, pássaros na 3 e gatos na 1. Então os peixes vão para a casa 4, a do alemão. Assim, casa 4 = peixes.

Conclusão:

	Casa 1	Casa 2	Casa 3	Casa 4	Casa 5
Cor	Amarela	Azul	Vermelha	Verde	Branca
Nacionalidade	Norueguês	Dinamarquês	Inglês	Alemão	Sueco
Bebida	Água	Chá	Leite	Café	Cerveja
Cigarro	Dunhill	Blends	Pall Mall	Prince	Blue Master
Animal	Gatos	Cavalos	Pássaros	Peixes	Cachorros

PARTE B do Projeto: Pseudocódigo com Prolog

1. Definição de Paradigma Lógico

É um estilo de programação que é baseado na lógica formal. No prolog, por exemplo, o programa é construído a partir de fatos (verdades absolutas) e regras (lógicas que vêm das verdades absolutas). É ideal para quebra-cabeças lógicos, como o problema de Einstein.

2. Proposta de Solução em Pseudocódigo

FATOS

casa(1). casa(2). casa(3). casa(4). casa(5)

nacionalidade(noruegues). nacionalidade(dinamarques). Nacionalidade(ingles).
nacionalidade(alemao). nacionalidade(sueco).

bebida(agua). bebida(cha). bebida(leite). bebida(cafe). bebida(cerveja).

cigarro(dunhill). cigarro(blends). cigarro(pall_mall). cigarro(prince). cigarro(blue_master).

animal(gatos). animal(cavalos). animal(passaros). animal(peixes). animal(cachorros).

REGRAS À PARTIR DOS FATOS

regra1 :- nacionalidade(ingles, Casa), cor(vermelha, Casa).

regra2 :- nacionalidade(sueco, Casa), animal(cachorros, Casa).

regra3 :- nacionalidade(dinamarques, Casa), bebida(cha, Casa).

regra4 :- cor(verde, CasaVerde), cor(branca, CasaBranca), a_esquerda(CasaVerde, CasaBranca).

regra5 :- cor(verde, Casa), bebida(cafe, Casa).

regra6 :- cigarro(pall_mall, Casa), animal(passaros, Casa).

regra7 :- cor(amarela, Casa), cigarro(dunhill, Casa).

regra8 :- casa(3), bebida(leite, 3).

regra9 :- nacionalidade(noruegues, 1).

regra10 :- cigarro(blends, CasaBlends), animal(gatos, CasaGatos), ao_lado(CasaBlends, CasaGatos).

regra11 :- animal(cavalos, CasaCavalos), cigarro(dunhill, CasaDunhill),
ao_lado(CasaCavalos, CasaDunhill).

regra12 :- cigarro(blue_master, Casa), bebida(cerveja, Casa).

regra13 :- nacionalidade(alemao, Casa), cigarro(prince, Casa).

regra14 :- nacionalidade(noruegues, CasaNoruegues), cor(azul, CasaAzul),
ao_lado(CasaNoruegues, CasaAzul).

regra15 :- cigarro(blends, CasaBlends), bebida(agua, CasaAgua), ao_lado(CasaBlends, CasaAgua).

3. Proposta para resolução do problema

Minha proposta para resolução do problema de Einstein é planejada em 3 etapas:

- a) Definição dos fatos: declaração dos fatos que enumeram as possibilidades para cada categoria, como nacionalidade, bebidas, etc.
- b) Traduzir as pistas do enunciado em regras lógicas: cada uma das 15 pistas será traduzida para uma regra em Prolog.
- c) Consulta e Unificação: implementar um predicado com a solução, fazendo o Prolog gerar combinações possíveis para as 5 casas, e testá-las contra cada restrição. Através do mecanismo backtracking, ele descartará combinações inválidas até encontrar aquela que satisfaça todas as pistas dadas pelo enunciado.

4. Desafios Técnicos Identificados (mais detalhado na Parte B do relatório)

A modelação de regras, como a de vizinhos, principalmente.

As restrições simultâneas que devem ser todas satisfeitas ao mesmo tempo.

A sintaxe e estrutura do Prolog, traduzir a lógica do problema para a sintaxe correta da linguagem.

PARTE C do Projeto: Código completo em Prolog

A partir da minha solução analítica, criei uma implementação em Prolog, que está logo abaixo:

```
ao_lado(X, Y, [X, Y, _, _, _]).
```

```
ao_lado(X, Y, [_ , X, Y, _, _]).
```

```
ao_lado(X, Y, [_ , _ , X, Y, _]).
```

```
ao_lado(X, Y, [_ , _ , _ , X, Y]).
```

```
ao_lado(X, Y, [Y, X, _, _, _]).
```

```
ao_lado(X, Y, [_ , Y, X, _, _]).
```

```
ao_lado(X, Y, [_ , _ , Y, X, _]).
```

```
ao_lado(X, Y, [_ , _ , _ , Y, X]).
```

```
esquerda_de(X, Y, [X,Y,_,_,_]).
```

```
esquerda_de(X, Y, [_ ,X,Y,_,_]).
```

```
esquerda_de(X, Y, [_ ,_,X,Y,_]).
```

```
esquerda_de(X, Y, [_ ,_,_,X,Y]).
```

solucao(Casas) :-

Casas = [Casa1, _, Casa3, _, _],

% Cada casa: casa(Nacionalidade, Cor, Bebida, Cigarro, Animal)

% REGRAS E FATOS

% O norueguês mora na primeira casa.

Casa1 = casa(noruegues, _, _, _, _),

% O homem que mora na casa do centro (3ª) bebe leite.

Casa3 = casa(_, _, leite, _, _),

% O inglês mora na casa vermelha.

member(casa(ingles, vermelha, _, _, _), Casas),

% O sueco cria cachorros.

member(casa(sueco, _, _, _, cachorros), Casas),

% O dinamarquês bebe chá.

member(casa(dinamarques, _, cha, _, _), Casas),

% A casa verde está à esquerda da branca.

esquerda_de(casa(_, verde, _, _, _), casa(_, branca, _, _, _), Casas),

% O morador da casa verde bebe café.

member(casa(_, verde, cafe, _, _), Casas),

% O homem que fuma Pall Mall cria aves.

member(casa(_, _, _, pall_mall, aves), Casas),

% O homem da casa amarela fuma Dunhill.

member(casa(_, amarela, _, dunhill, _), Casas),

% O homem que fuma Blends mora ao lado do que cria gatos.

ao_lado(casa(_, _, _, blends, _), casa(_, _, _, _, gatos), Casas),

% O homem que cria cavalos mora ao lado do que fuma Dunhill.

ao_lado(casa(_, _, _, _, cavalos), casa(_, _, _, dunhill, _), Casas),

% O homem que fuma Blue Master bebe cerveja.

member(casa(_, _, cerveja, blue_master, _), Casas),

% O alemão fuma Prince.

member(casa(alemao, _, _, prince, _), Casas),

% O norueguês mora ao lado da casa azul.

ao_lado(casa(noruegues, _, _, _, _), casa(_, azul, _, _, _), Casas),

% O homem que fuma Blends tem vizinho que bebe água.

ao_lado(casa(_, _, _, blends, _), casa(_, _, agua, _, _), Casas).

% Solução: Quem cria peixes?

quem_tem_peixes(Dono) :-

solucao(Casas),

member(casa(Dono, _, _, _, peixes), Casas).

A seguir é a PARTE D do Projeto, que contém a explicação do código final linha a linha, os relatórios de cada parte do projeto, e prints das saídas do código final. Além disso, disponibilizei as IA's utilizadas durante a PARTE C do projeto e meus prompts, acompanhado de fontes de pesquisa.

PARTE D DO PROJETO: Explicação do código e relatório completo

Explicação do Código: Código em Prolog solucionando o problema de Einstein explicado linha a linha

ao_lado(X, Y, [X, Y, _, _, _]).

ao_lado(X, Y, [_ , X, Y, _, _]).

ao_lado(X, Y, [_ , _, X, Y, _]).

ao_lado(X, Y, [_ , _, _, X, Y]).

ao_lado(X, Y, [Y, X, _, _, _]).

ao_lado(X, Y, [_ , Y, X, _, _]).

ao_lado(X, Y, [_ , _, Y, X, _]).

ao_lado(X, Y, [_ , _, _, Y, X]).

➔ Serve para dizer quando duas casas estão uma do lado da outra.

➔ Cada linha é um jeito diferente de X e Y ficarem juntos na lista das 5 casas.

esquerda_de(X, Y, [X,Y,_,_,_]).

esquerda_de(X, Y, [_ ,X,Y,_,_]).

esquerda_de(X, Y, [_ ,_,X,Y,_,_]).

esquerda_de(X, Y, [_ ,_,_,X,Y]).

➔ Essas linhas que estão escritas esquerda_de correspondem a uma das quatro possíveis posições em uma lista de cinco elementos em que X está imediatamente à esquerda de Y.

solucao(Casas) :-

Casas = [Casa1, _, Casa3, _, _],

- ➔ Solucao(Casas):- Define o predicado principal que unifica Casas da maneira de que as cinco casas satisfaçam todas as pistas.
- ➔ Casas = [Casa1, _, Casa3, _, _], Aqui, eu mostro que Casas é uma lista de 5 elementos, não necessariamente me importando com o nome de todas as variáveis.

% Cada casa: casa(Nacionalidade, Cor, Bebida, Cigarro, Animal)

% REGRAS E FATOS

% O norueguês mora na primeira casa.

Casa1 = casa(noruegues, _, _, _, _),

- ➔ Unifica que a casa 1 pertence à nacionalidade norueguês.

% O homem que mora na casa do centro (3ª) bebe leite.

Casa3 = casa(_, _, leite, _, _),

- ➔ Mesma coisa, mostra que na Casa3, a do meio, a bebida é leite.

% O inglês mora na casa vermelha.

member(casa(ingles, vermelha, _, _, _), Casas),

- ➔ Mostra que há uma casa com o inglês cuja cor é vermelha.

% O sueco cria cachorros.

member(casa(sueco, _, _, _, cachorros), Casas),

- ➔ Mostra que há uma casa com nacionalidade sueco que o animal é cachorros.

% O dinamarquês bebe chá.

member(casa(dinamarques, _, cha, _, _), Casas),

- ➔ Garante a existência de uma casa do dinamarquês cuja bebida é chá.

% A casa verde está à esquerda da branca.

esquerda_de(casa(_, verde, _, _, _), casa(_, branca, _, _, _), Casas),

- ➔ Aqui estou dizendo: existe uma casa com cor verde que está imediatamente à esquerda de uma casa com cor branca na lista Casas.

% O morador da casa verde bebe café.

member(casa(_, verde, cafe, _, _), Casas),

- ➔ Garante que a casa que é verde tenha como bebida o café.

% O homem que fuma Pall Mall cria aves.

member(casa(_, _, _, pall_mall, aves), Casas),

➔ Garante a existência de uma casa que tem o cigarro Pall Mall e o animal aves.

% O homem da casa amarela fuma Dunhill.

member(casa(_, amarela, _, dunhill, _), Casas),

➔ Aqui estou falando que na casa de cor amarela o cigarro é da marca dunhill.

% O homem que fuma Blends mora ao lado do que cria gatos.

ao_lado(casa(_, _, _, blends, _), casa(_, _, _, _, gatos), Casas),

➔ Aqui, eu uso ao_lado para falar que a casa com cigarro blends é vizinha da casa que cria gatos.

% O homem que cria cavalos mora ao lado do que fuma Dunhill.

ao_lado(casa(_, _, _, _, cavalos), casa(_, _, _, dunhill, _), Casas),

➔ Afirma que a casa que se cria cavalos é vizinha da casa em que se fuma dunhill.

% O homem que fuma Blue Master bebe cerveja.

member(casa(_, _, cerveja, blue_master, _), Casas),

➔ Garante que exista uma casa que a bebida seja cerveja e o cigarro Blue Master.

% O alemão fuma Prince.

member(casa(alemao, _, _, prince, _), Casas),

➔ Mostra que existe uma casa com a nacionalidade alemão e que fuma Prince.

% O norueguês mora ao lado da casa azul.

ao_lado(casa(noruegues, _, _, _, _), casa(_, azul, _, _, _), Casas),

➔ Afirma que a casa do lado do norueguês é azul, sabemos que é na segunda posição porque o norueguês mora na primeira casa.

% O homem que fuma Blends tem vizinho que bebe água.

ao_lado(casa(_, _, _, blends, _), casa(_, _, agua, _, _), Casas).

➔ A casa com o cigarro Blends tem como vizinho a casa com água.

% Solução: Quem cria peixes?

quem_tem_peixes(Dono) :-

➔ Define um predicado que responde quem é o dono que cria peixes.

solucao(Casas),

➔ Predicado que encontra uma solução satisfazendo as restrições

member(casa(Dono, _, _, _, peixes), Casas).

- ➔ No resultado Casas procura a casa cujo animal é peixes e unifica a nacionalidade dessa casa com Dono.
- ➔ E então, ao consultar quem_tem_peixes(D), o Prolog retorna o valor de D que satisfaz todas as condições.

Relatório da PARTE A do Projeto: Pré Projeto

Para resolver o problema analiticamente, primeiramente eu separei os fatos, aqueles que não mudariam de jeito nenhum, tipo o fato de que a casa do meio (3) teria a bebida leite, que o inglês vivia na casa vermelha, que o norueguês morava na primeira casa, etc.

Como no problema fala que o norueguês vive na primeira casa e que ele também vive ao lado da casa azul, a casa 2 é obrigatoriamente azul.

Aqui, também fala que a casa verde fica imediatamente à esquerda da casa branca. De primeira, eu pensei que eu poderia simplesmente colocar em qualquer lugar apenas satisfazendo essa obrigação. Mas, a casa do meio, bebe leite, e quem vive na casa verde, bebe café.. Portanto a verde não poderia ficar no meio, e sim na casa 4.

Dessa maneira, eu já havia encontrado todas as cores das casas, aí ficou assim:

Casa 1: Amarela

Casa 2: Azul

Casa 3: Vermelha

Casa 4: Verde

Casa 5: Branca

(Eu concluí que a casa 1 seria amarela porque o inglês mora numa casa vermelha e ele não poderia morar na casa 1 porque quem deve morar lá é o norueguês).

Agora a gente pode montar os fatos de acordo com as cores das casas, como a marca de cigarro Dunhill, que deve ser da casa amarela.

Quem cria cavalos, portanto, está na casa azul, porque “O homem que cria Cavalos vive ao lado do que fuma Dunhill.”

Fatos de quem fuma Blends: Precisa ter (i) um vizinho com gatos e (ii) um vizinho que bebe água.

A casa 3, é obrigatoriamente leite, então a água ficaria na casa 1 e os gatos na casa 1 também. (os gatos não podem ficar na casa 3 porque lá, vai ficar o par pássaros e pall mall)

Como a água está na casa 1, a dupla obrigatória Blue Master + cerveja não pode estar na 1. Como na casa 3 já é obrigatoriamente leite e 4 é café (verde + café), sobra a casa 5 para Blue Master + cerveja.

No momento está assim:

	Casa 1	Casa 2	Casa 3	Casa 4	Casa 5
Cor	Amarela	Azul	Vermelha	Verde	Branca
Nacionalidade	Norueguês		Inglês		Sueco
Bebida	Água		Leite	Café	
Cigarro	Dunhill	Blends	Pall Mall		Blue Master
Animal	Gatos	Cavalos	Pássaros		

O alemão tinha que fumar Prince obrigatoriamente, como 5 já é Blue Master, 1 já é Dunhill, 2 já é Blends e Pall Mall já está na 3, conclui-se que, o alemão e o Prince ficarão na casa 4.

Agora, faltaram duas nacionalidades, daí eu pensei, na 1 já está o norueguês, na 3 o inglês, na 4 o alemão (descobrimos agora), então o dinamarquês e o sueco ficariam ou na 2 ou na 5 obrigatoriamente.

Como a 5 não pode ser o dinamarquês (porque o dinamarquês bebe chá e já definimos que na 5 a bebida é cerveja) portanto, o dinamarquês ficará na casa 2 e automaticamente a casa 2 será preenchida com chá e a 5 com sueco.

O sueco cria cachorros, então, na casa 5 o animal será cachorros.

Sobra peixes, para única casa restante no animal, que é a casa 4, do alemão.

Ficou assim! Concluído.

	Casa 1	Casa 2	Casa 3	Casa 4	Casa 5
Cor	Amarela	Azul	Vermelha	Verde	Branca
Nacionalidade	Norueguês	Dinamarquês	Inglês	Alemão	Sueco
Bebida	Água	Chá	Leite	Café	Cerveja
Cigarro	Dunhill	Blends	Pall Mall	Prince	Blue Master
Animal	Gatos	Cavalos	Pássaros	Peixes	Cachorros

Relatório PARTE B do Projeto: Pseudocódigo com Prolog

Para a Parte B do projeto, eu comecei separando os fatos das regras com o Prolog.

Por exemplo, Casa(1), Casa(2), etc. Assim também com a Nacionalidade, Animal, Bebida, Cigarro e Cor.

E então, eu parti para a parte de deduzir regras a partir das pistas já definidas no problema.

Como por exemplo, a casa do inglês tem que ser necessariamente vermelha, na casa do sueco é criado cachorros, a bebida da casa do dinamarquês é chá, a casa do meio bebe leite ou seja, a casa(3). E assim sucessivamente.

Durante a elaboração dessa parte do projeto, eu tive umas dificuldades (desafios técnicos) encontradas como:

A implementação da relação “vive ao lado de” é fundamental para a conclusão do problema. O desafio foi justamente definir corretamente o que capture essa relação de vizinhança entre duas casas, garantindo que ela seja correta. Ou seja, se a casa 1 está ao lado da casa 2, então a casa 2 deve automaticamente estar ao lado da casa 1, e isso precisa ser representado na lógica do programa para que todas as restrições de vizinhança das pistas funcionem corretamente.

A integração de múltiplas restrições, no caso desse problema, há 15 regras que devem ser satisfeitas ao mesmo tempo. Então, o desafio está em estruturar o código de forma que todas essas regras sejam aplicadas de maneira correta.

A representação da estrutura de dados poderia até ser algo muito difícil de se formular de maneira clara, mas felizmente, com o site (<https://rachacuca.com.br/logica/problemas/teste-de-einstein/>) ficou bem mais fácil de nomear os dados de forma clara para resolução do pseudocódigo.

Relatório da PARTE C do projeto: Implementação do código em Prolog

Aqui, o objetivo foi basicamente traduzir a solução analítica do problema de Einstein para a linguagem Prolog, implementando um programa que utilize o paradigma lógico para encontrar a resposta correta. Envolvendo fatos e regras representando as restrições do problema.

O código foi dividido em 3 partes principais:

Predicados para definir relações entre as casas

Predicados para definir as restrições do problema

Predicados de consultas, usados para obter as respostas das queries.

Implementei dois predicados para modelar as regras de vizinhança, com esquerda_de e ao_lado, os de restrições, com regras e fatos, e o principal solucao(Casas).

Saídas:

```
solucao(Casas).
Casas =
[casa(noruegues,amarela,agua,dunhill,gatos), casa(dinamarques,azul,cha,blends,cavalos),
casa(ingles,vermelha,leite,pall_mall,aves), casa(alemao,verde,cafe,prince,_),
casa(sueco,branca,cerveja,blue_master,cachorros)]

quem_tem_peixes(Dono).
Dono = alemao

quem_tem_peixes(D).
D = alemao
```

IAS UTILIZADAS: CHAT-GPT, DeepSeek.

PROMPTS:

1. "Estou implementando o teste de Einstein em Prolog. Fixei a casa 1 e a 3. Preciso garantir que nacionalidades, cores e outros atributos não se repitam. Usar vários predicados member para cada atributo, como member(casa(ingles, _, _, _), Casas), é suficiente para garantir a unicidade?"

A resposta foi que sim, que para esse problema em específico, as restrições member eram suficientes. **(CHAT-GPT, DeepSeek).**

2. "Meu código em Prolog funciona, mas recebo um aviso 'Singleton variables: [Casa2, Casa4, Casa5]'. O que isso significa? É um problema grave? Como posso corrigir isso sem alterar a lógica do programa?"

A resposta foi que as variáveis estavam declaradas, mas não estavam sendo usadas, por isso o aviso. Então ela me recomendou (para eliminar esse aviso) utilizar o _, para as variáveis não utilizadas. E funcionou. **(DeepSeek).**

3. "Qual é a melhor forma de representar uma casa no Prolog? Devo usar um termo composto como casa(Nacionalidade, Cor, Bebida, Cigarro, Animal) ou listas separadas para cada atributo?"

A IA Recomendou veementemente o uso do termo composto, pois mantém todos os atributos de uma entidade logicamente agrupados, melhorando a legibilidade e a escrita das regras, que frequentemente envolvem múltiplos atributos de uma mesma casa. **(CHAT-GPT e DeepSeek).**

Por fim, a IA foi utilizada como uma ferramenta de consulta para esclarecimentos pontuais, confirmação de conceitos e melhoria de estilo de código.

Fontes utilizadas além das IAs:

<https://querobolsa.com.br/revista/pseudocodigo>

<https://rachacuca.com.br/logica/problemas/teste-de-einstein/>

http://www3.decom.ufop.br/toffolo/site_media/uploads/2011-1/bcc402/slides/10._backtracking.pdf

<https://exame.com/ciencia/enigma-de-einstein-e-um-problema-logico-que-da-no-no-cerebro/>

<https://builtin.com/software-engineering-perspectives/prolog>

<http://www.linhadecodigo.com.br/artigo/1697/descobrindo-o-prolog.aspx>