



Disciplina: **Paradigmas de Linguagens de Programação**
Professor: Dr. Ausberto S. Castro V.
e-Mail: ascv@uenf.br

Data: 27 de abril de 2023

Prática Prolog

Nome Completo: Gabriel Costa Fassarella
Data: 03/05/2023

Arquivo **01-sistespacial.pl**

1. Explicar o significado dos fatos relacionados com **estrela** e **orbita**. Qual é significado das duas primeiras regras?

O fato **estrela** define o nome das estrelas de alguns sistemas, e o fato **orbita** define o nome de alguns planetas que orbitam alguma estrela, ou de alguma lua que orbita algum planeta. A primeira regra define que um planeta é um corpo que se relaciona (**orbita**) com o sol, já a segunda se refere a um satélite, que é definido como um corpo que orbita um planeta P que também orbita o sol.

2. Antes de começar, execute no Editor: **Iniciar** ☐ **Consultar** (ou **F9**). Execute os comandos abaixo listados (capture as telas com os resultados!!!. **Observe as letras maiúsculas ou minúsculas!!!**

- ?- **estrela(sol).**
- ?- **estrela(sirius).**
- ?- **estrela(jupiter).**
- ?- **orbita(lua, sol).**
- ?- **orbita(marte,sol).**
- ?- **planeta(marte).**
- ?- **planeta(X).**
- ?- **satelite(phobos).**
- ?- **satelite(Y).**
- ?- **ssolar(sol).**
- ?- **ssolar(lua).**
- ?- **ssolar(sirius).**
- ?- **ssolar(R).**

```

?- estrela(sol).

?- estrela(sirius).

?- estrela(jupiter).
false.

?- orbita(lua, sol).
false.

?- orbita(marte, sol).

?- planeta(marte).

?- planeta(X).
X = mercurio ;

?- satellite(phobos).
ERROR: Unknown procedure: satellite/1 (DWIM could not correct goal)
?- satellite(Y).
ERROR: Unknown procedure: satellite/1 (DWIM could not correct goal)
?- ssolar(sol).

?- ssolar(lua).
ERROR: Unknown procedure: satellite/1
ERROR: In:
ERROR:    [11] satellite(lua)
ERROR:    [9] toplevel_call(user:user: ...) at c:/program files/swipl/boot/toplevel.pl:1173
ERROR:
ERROR: Note: some frames are missing due to last-call optimization.
ERROR: Re-run your program in debug mode (:- debug.) to get more detail.
      Exception: (11) satellite(lua) ? No previous search
      Exception: (11) satellite(lua) ?
      creep
?- ssolar(sirius).
ERROR: Unknown procedure: satellite/1
ERROR: In:
ERROR:    [11] satellite(sirius)
ERROR:    [9] toplevel_call(user:user: ...) at c:/program files/swipl/boot/toplevel.pl:1173
ERROR:
ERROR: Note: some frames are missing due to last-call optimization.
ERROR: Re-run your program in debug mode (:- debug.) to get more detail.
      Exception: (11) satellite(sirius) ?
      creep
?- ssolar(R).
R = sol .

```

Arquivo 02-familia-A.pl

3. Explicar o significado do fato `mulher(eva)`.? Qual é o significado das regras `filho(X,Y)` e `filha A,B`) definidas no programa?

O fato `eva` significa que `eva` é uma mulher, uma vez que isso é definido no código. Já a regra `filho(X,Y)` define a relação de parentesco entre pais com um filho do sexo masculino, sendo que `X` é filho de `Y`, assim como filha, que define a relação de parentesco entre um pai ou mãe com uma filha do sexo feminino, sendo que `A` é filha de `B`.

4. Execute os comandos (capture as telas!)

- `filho(Filho,De).`
- `filho(paulo,X).`
- `pai(adam,X).`
- `filho(Y,eva).`

```
?- filho(Filho,De) .  
Filho = pedro,  
De = adam .  
  
?- filho(paulo,X) .  
X = adam .  
  
?- pai(adam,X) .  
X = pedro .  
  
?- filho(Y,eva) .  
Y = pedro .
```

5. Arquivo [02-familia-B.pl](#). Incluir no programa alguns fatos e regras para o relacionamento `irmão`, `irmã`, `prima` e `tia`. Faça os testes respectivos para os novos relacionamentos.

```

1 % programa 02-familia-B.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Date: 2023
7 % Aluno: Gabriel Costa Fassarella
8 %-----
9 % FATOS
10
11     mulher(luisa). % significa luisa é uma mulher
12     mulher(eva).
13     mulher(martha).
14     homem(adam).
15     homem(pedro).
16     homem(paulo).
17     pai(adam,pedro). % significa adam é pai de pedro
18     pai(adam,paulo).
19     marido(adam,eva).
20     esposa(eva,adam).
21     mae(eva,luisa).
22     mae(eva,pedro).
23     filha(maria,eva).
24     filha(luisa,adam).
25     filha(suzana,martha).
26     irma(eva,martha).
27
28
29 % Completar as regras:
30     irmao(X,Y):-filho(X,Z),filho(Y,Z), X \= Y.
31     irmao(X,Y):-filho(X,Z),filha(Y,Z), X \= Y.
32
33     irma(X,Y):-filha(X,Z),filho(Y,Z), X \= Y.
34     irma(X,Y):-filha(X,Z),filha(Y,Z), X \= Y.
35
36     tia(X,Y):-irma(X,Z),filho(Y,Z).
37     tia(X,Y):-irma(X,Z),filha(Y,Z).
38     tia(X,Y):-irma(Z,X),filho(Y,Z).
39     tia(X,Y):-irma(Z,X),filha(Y,Z).
40
41     prima(X,Y):-tia(Z,Y),filha(X,Z).

```

```
|      prima(X,Y) .  
X = maria,  
Y = suzana ;  
X = luisa,  
Y = suzana ;  
X = suzana,  
Y = pedro ;  
X = suzana,  
Y = maria ;  
X = suzana,  
Y = luisa ;  
false.
```

```
?-  
|      irmao(X,Y) .  
X = pedro,  
Y = paulo ;  
X = paulo,  
Y = pedro ;  
X = pedro,  
Y = luisa ;  
X = paulo,  
Y = luisa ;  
X = pedro,  
Y = maria ;  
X = pedro,  
Y = luisa ;  
false.
```

```
?- tia(X,Y)  
|      .  
X = eva,  
Y = suzana ;  
X = martha,  
Y = pedro ;  
X = martha,  
Y = maria ;  
X = martha,  
Y = luisa ;  
false.
```

Arquivo 03-casa.pl

6. Execute o programa: **F9** ou no menu: **Start** □ **Consult** . Faça as seguintes consultas (na janela onde aparece o prompt **?-**):

- lugar(sala).
- lugar(quadra).
- lugar(X). % dar <Enter> até aparecer a resposta 'No'
- localizacao(lanterna, Y).
- localizacao(Objetos, cozinha).
- localizacao(Objeto, Onde).
- conectar(cozinha, escritorio).
- conectar(escritorio, celeiro).
- conectar(X, Y).
- lista_coisas(cozinha).
- lista_coisas(escritorio).
- lista_portas(cozinha).
- observar.

```
?- lugar(sala).
```

```
?- lugar(quadra).  
false.
```

```
?- lugar(X).  
X = sala ;  
X = cozinha ;  
X = escritorio ;  
X = corredor ;  
X = 'sala de jantar' ;  
X = celeiro ;  
X = jardim ;  
false.
```

```
?- localizacao(lanterna, Y).  
Y = escrevaninha ;  
false.
```

```
?- localizacao(Objetos, cozinha).  
Objetos = laranja ;  
Objetos = verdura ;  
Objetos = faca ;  
Objetos = panela ;  
Objetos = arroz ;  
Objetos = uva ;  
Objetos = biscoitos ;  
false.
```

```
?- localizacao(Objeto, Onde).
Objeto = escrivaninha,
Onde = escritorio ;
Objeto = laranja,
Onde = cozinha ;
Objeto = lanterna,
Onde = escrivaninha ;
Objeto = 'maq de lavar',
Onde = celeiro ;
Objeto = sabao,
Onde = 'maq de lavar' ;
Objeto = verdura,
Onde = cozinha ;
Objeto = faca,
Onde = cozinha ;
Objeto = panela,
Onde = cozinha ;
Objeto = arroz,
Onde = cozinha ;
Objeto = uva,
Onde = cozinha ;
Objeto = biscoitos,
Onde = cozinha ;
Objeto = computador,
Onde = escritorio ;
Objeto = mesa,
Onde = 'sala de jantar' ;
false.

?- conectar(cozinha, escritorio).

?-
|   conectar(escritorio, celeiro).
false.
```

```
?- conectar(X,Y).
X = escritorio,
Y = corredor ;
X = cozinha,
Y = escritorio ;
X = corredor,
Y = 'sala de jantar' ;
X = cozinha,
Y = celeiro ;
X = 'sala de jantar',
Y = cozinha ;
X = corredor,
Y = escritorio ;
X = escritorio,
Y = cozinha ;
X = 'sala de jantar',
Y = corredor ;
X = celeiro,
Y = cozinha ;
X = cozinha,
Y = 'sala de jantar' ;
false.

?- lista_coisas(cozinha).
laranja
verdura
faca
panela
arroz
uva
biscoitos
```

```

?- lista_coisas(escritorio).
    escritivaninha
    computador

?- lista_portas(cozinha).
    escritorio
    celeiro
    sala de jantar

?- observar.
Voce esta na cozinha
Voce pode ver:
    laranja
    verdura
    faca
    panela
    arroz
    uva
    biscoitos
Voce pode ir para:
    escritorio
    celeiro
    sala de jantar

```

7. Qual é significado da regra `conectar(A,B)`?

Significa os locais que estão ligados por meio de uma porta, ou seja, significa os locais adjacentes nos quais é possível ir por uso da porta que os liga.

8. Adicionar ao programa `03-casa.pl` pelo menos `três lugares` com quatro (04) objetos cada um e fazer os respectivos testes positivos (resposta verdadeira, Yes) e negativos (resposta falsa, No), listando:

```
1 % Programa casa.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Ano: 2023
7 % Aluno: Gabriel Costa Fassarella
8 %-----
9 %     Ver mapa da casa
10 %
11 %----- FATOS -----
12 lugar(sala).
13 lugar(cozinha).
14 lugar(escritorio).
15 lugar(corredor).
16 lugar('sala de jantar').
17 lugar(celeiro).
18 lugar(jardim).
19 lugar(banheiro).
20 lugar(garagem).
21 lugar(deposito).
22
23
24 porta(escritorio, corredor).
25 porta(banheiro, corredor).
26 porta(garagem, corredor).
27 porta(deposito, garagem).
28 porta(cozinha, escritorio).
29 porta(corredor, 'sala de jantar').
30 porta(cozinha, celeiro).
```

```

31 porta('sala de jantar', cozinha).
32
33 localizacao(privada, banheiro).
34 localizacao(chuveiro, banheiro).
35 localizacao(toalha, banheiro).
36 localizacao(sabonete, banheiro).
37 localizacao(carro, garagem).
38 localizacao(pneu, garagem).
39 localizacao(moto, garagem).
40 localizacao(bicicleta, garagem).
41 localizacao(ferramentas, deposito).
42 localizacao(moveis, deposito).
43 localizacao(caixas, deposito).
44 localizacao(armarios, deposito).
45 localizacao(escrivaninha, escritorio).
46 localizacao(laranja, cozinha).
47 localizacao(lanterna, escrivaninha).
48 localizacao('maq de lavar', celeiro).
49 localizacao(sabao, 'maq de lavar').
50 localizacao(verdura, cozinha).
51 localizacao(faca, cozinha).
52 localizacao(panela, cozinha).
53 localizacao(arroz, cozinha).
54 localizacao(uva, cozinha).
55 localizacao(biscoitos, cozinha).
56 localizacao(computador, escritorio).
57 localizacao(mesa, 'sala de jantar').
58

```

- Lugares que tenham conexão entre si

```

?- lista_portas(banheiro).
    corredor

```

```

?- lista_portas(garagem).
    corredor
    deposito

```

```

?- lista_portas(deposito).
    garagem

```

- Objetos de um determinado lugar

```

?- lista_coisas(banheiro).
    privada
    chuveiro
    toalha
    sabonete

?- lista_coisas(garagem).
    carro
    pneu
    moto
    bicicleta

?- lista_coisas(deposito).
    ferramentas
    moveis
    caixas
    armarios

```

- Altere o comando **aqui(X)** para outro lugar e execute o comando **observar**

```

?- estamos_em(X).
X = corredor ;
false.

?- observar.
Voce esta na corredor
Voce pode ver:
Voce pode ir para:
    sala de jantar
    escritorio
    banheiro
    garagem

```

9. Construir:

- uma regra **estou(Lugar)** que permita fazer a consulta: onde você está, quais são as coisas que você esta observando, e aonde você pode ir desde este lugar.

```

estou(Lugar):- estamos_em(Lugar), lista_coisas(Lugar).


```

- Uma única regra que permita saber os objetos que estão na cozinha e são comestíveis?

```
comer(Y):- estou(X), comestivel(Y).
```

- Uma regra: "Encontrar o objeto T e o lugar R tal que existe uma porta da cozinha para R e o objeto T esta em R"

```
107 objt(T,R):- porta(cozinha, R), localizacao(T,R).
108
```

< 

linha: 107 Coluna: 56 Sobrepor ANSI/Windows C:\Users\Gabrie



Warning: Singleton variables: [X]

```
?- objt(T,R).
T = escritaninha,
R = escritorio ;
T = computador,
R = escritorio ;
T = 'maq de lavar',
R = celeiro ;
false.
```

Arquivo [genealogia.pl](#) (Criar o programa!)

10. Construir um programa PROLOG novo **genealogia.pl** que mostre os relacionamentos de SUA família incluindo: pai, mãe, filho, filha, irmãos, tios, primos e avós. Mostre pelo menos cinco testes (telas capturadas para cada teste).

```
1 % Programa casa.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Ano: 2023
7 % Aluno: Gabriel Costa Fassarella
8 %-----
9 %     Ver mapa da casa
10 %
11 %----- FATOS -----
12
13 homem(diogo).
14 homem(gabriel).
15 homem(enzo).
16 homem(luis).
17 homem(lucas).
18 mulher(raquel).
19 mulher(jaqueline).
20 mulher(julia).
21 mulher(ana).
22 mulher(larissa).
23 pai(diogo, enzo).
24 pai(diogo, luis).
25 pai(diogo, julia).
26 pai(enzo, gabriel).
27 pai(enzo, larissa).
28 pai(luis, lucas).
29 mae(raquel, enzo).
30 mae(raquel, luis).
31 mae(raquel, julia).
32 mae(jaqueline, gabriel).
33 mae(jaqueline, larissa).
```

```

34 mae(ana, lucas).
35 casamento(diogo, raquel).
36 casamento(enzo, jaqueline).
37 casamento(luis, ana).
38
39 %----- Regras -----
40
41 filho(X, Y):- homem(X), pai(Y, X).
42 filho(X, Y):- homem(X), mae(Y, X).
43
44 filha(X, Y):- mulher(X), pai(Y, X).
45 filha(X, Y):- mulher(X), mae(Y, X).
46
47 irmao(X, Y):- filho(X, Z), filho(Y, Z), X \= Y.
48 irmao(X, Y):- filho(X, Z), filha(Y, Z), X \= Y.
49
50 irma(X, Y):- filha(X, Z), filho(Y, Z), X \= Y.
51 irma(X, Y):- filha(X, Z), filha(Y, Z), X \= Y.
52
53 tio(X, Y):- irmao(X, Z), filho(Y, Z).
54 tio(X, Y):- irmao(X, Z), filha(Y, Z).
55 tio(X, Y):- homem(X), casamento(X, Z), irma(Z, W), filho(Y, W).
56 tio(X, Y):- homem(X), casamento(X, Z), irma(Z, W), filha(Y, W).
57
58 tia(X, Y):- irma(X, Z), filho(Y, Z).
59 tia(X, Y):- irma(X, Z), filha(Y, Z).
60 tia(X, Y):- mulher(X), casamento(Z, X), irmao(Z, W), filho(Y, W).
61 tia(X, Y):- mulher(X), casamento(Z, X), irmao(Z, W), filha(Y, W).
62
63 primo(X, Y):- tio(Z, Y), filho(X, Z).
64 prima(X, Y):- tio(Z, Y), filha(X, Z).
65
66 avol(X, Y):- homem(X), filho(Z, X), filho(Y, Z).
67 avol(X, Y):- homem(X), filha(Z, X), filho(Y, Z).
68 avol(X, Y):- homem(X), filha(Z, X), filha(Y, Z).
69 avol(X, Y):- homem(X), filho(Z, X), filha(Y, Z).
70
71 avo2(X, Y):- mulher(X), filho(Z, X), filho(Y, Z).
72 avo2(X, Y):- mulher(X), filha(Z, X), filho(Y, Z).
73 avo2(X, Y):- mulher(X), filha(Z, X), filha(Y, Z).
74 avo2(X, Y):- mulher(X), filho(Z, X), filha(Y, Z).

```



```

?- filho(gabriel, enzo).

?- irmaa(larissa, gabriel).
Correct to: "irma(larissa,gabriel)"? yes

?- avol(X,Y).
X = diogo,
Y = gabriel ;
X = diogo,
Y = lucas ;
X = diogo,
Y = larissa ;
false.

?- tia(ana, larissa).

?- prima(larissa, lucas).

```

Arquivo [pets.pl](#)

11. Construir um programa PROLOG [pets.pl](#) para mostrar os relacionamentos entre proprietários, gatos, cães, periquitos etc.

```

1 % Programa casa.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Ano: 2023
7 % Aluno: Gabriel Costa Fassarella
8 %-----
9 %     Ver mapa da casa
10 %
11 %----- FATOS -----
12
13 gato(tufao) .
14 gato(tom) .
15 gato(frajola) .
16 cachorro(bidu) .
17 cachorro(moliver) .
18 cachorro(thorfin) .
19 periquito(piu piu) .
20 periquito(sol) .
21 periquito(romeu) .
22 hamster(mickey) .
23 hamster(jerry) .
24 hamster(minnie) .
25
26 proprietario(pedro) .
27 proprietario(breno) .
28 proprietario(julia) .
29
30 dono(pedro, tufao) .
31 dono(pedro, bidu) .
32 dono(pedro, tom) .
33 dono(breno, moliver) .
34 dono(breno, jerry) .
35 dono(breno, piu piu) .
36 dono(breno, sol) .
37 dono(julia, minnie) .
38 dono(julia, frajola) .
39 dono(julia, minnie) .
40 dono(julia, romeu) .
41 dono(julia, thorfinn) .
42
43 %----- Regras -----
44
45 animais_de(X, Y):- dono(X, Y) .

```

```
?- animais_de(X, Y).  
X = pedro,  
Y = tufao ;  
X = pedro,  
Y = bidu ;  
X = pedro,  
Y = tom ;  
X = breno,  
Y = moliver ;  
X = breno,  
Y = jerry ;  
X = breno,  
Y = sol ;  
X = julia,  
Y = minnie ;  
X = julia,  
Y = frajola ;  
X = julia,  
Y = minnie ;  
X = julia,  
Y = romeu ;  
X = julia,  
Y = thorfinn ;  
false.
```

Arquivo [04-hanoi.pl](#)

12. Executar os comandos

- mover(2,esquerda,centro,direita).
- mover(3,esquerda,centro,direita).

Verifique “manualmente” os resultados (mostre graficamente sequência de resultados)

```
1 % programa 04-hanoi.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Ano: 2023
7 % Nome do aluno: Gabriel Costa Fassarella
8 %-----
9
10 %      X = esquerda      Y = direita
11 mover(1,X,Y,_):-
12     write('Mover disco do topo da torre '),
13     write(X), write(' para a torre '),
14     write(Y), nl.
15
Linha: 7 Coluna: 42      Inserir      ANSI/Windows      C:\Users\Gabriel\Desktop\Costa\Estudos\Uenf\Paradigm
?- chdir('C:/Users/Gabriel/Desktop/Costa/Estudos/Uenf/Paradigmas/Laboratorio-Prolog/
?- consult('04-hanoi').
?- mover(2,esquerda,centro,direita).
Mover disco do topo da torre  esquerda para a torre direita
Mover disco do topo da torre  esquerda para a torre centro
Mover disco do topo da torre  direita para a torre centro
?- mover(3,esquerda,centro,direita).
Mover disco do topo da torre  esquerda para a torre centro
Mover disco do topo da torre  esquerda para a torre direita
Mover disco do topo da torre  centro para a torre direita
Mover disco do topo da torre  esquerda para a torre centro
Mover disco do topo da torre  direita para a torre esquerda
Mover disco do topo da torre  direita para a torre centro
Mover disco do topo da torre  esquerda para a torre centro
```

Arquivo **Geografia.pl**

13. Completar o arquivo com fatos e regras de modo que ao consultar

- Viagem_nacional(X,Y) me indique a rota entre o estado X e o estado Y

```
viagem_nacional(X, Y):- esta_em (X, Z), esta_em(Y, Z), Z \= 'america_do_sul'.|
```

```

?- viagem_nacional(X,Y).
X = saopaulo,
Y = portoalegre ;
X = saopaulo,
Y = rio ;
X = saopaulo,
Y = campos ;
X = saopaulo,
Y = riodejaneiro ;
X = saopaulo,
Y = brasilia ;
X = saopaulo,
Y = recife ;
X = saopaulo,
Y = fortaleza ;
X = saopaulo,
Y = campinas ;
X = saopaulo,
Y = curitiba ;
X = portoalegre,
Y = saopaulo ;
X = portoalegre,
Y = rio ;
X = portoalegre,
Y = campos ;
X = portoalegre,
Y = riodejaneiro ;
X = portoalegre,
Y = brasilia ;
X = portoalegre,
Y = recife ;
X = portoalegre,

```

- Viagem_internacional(A,B), me indique a rota entre o pais A e o pais B

```

59 viagem_internacional(PaisA,PaisB):- limita_com(PaisA,PaisB), limita_com(PaisA,PaisB).

```

```

?- viagem_internacional(X,Y).
X = peru,
Y = brasil ;
X = bolivia,
Y = brasil ;
X = chile,
Y = peru ;
X = peru,
Y = chile ;
X = equador,
Y = peru ;
X = peru,
Y = equador ;
false.

```

14. Em cada arquivo, inserir outros FATOS. Analisar, Testar e comentar os arquivos [rota2.pl](#) e [veste.pl](#).
[rota2.pl](#):

É possível afirmar que os fatos inseridos descrevem uma conexão entre pontos com nomes não determinados.

```

1 % Arquivo: rota2.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Ano: 2023
7 % Aluno: Gabriel Costa Fassarella
8 %-----
9
10 % Fonte:   http://silveiraneto.net/2007/08/29/exemplos-de-prolog/
11
12 edge(1,2).   %conecta os pontos
13 edge(1,4).
14 edge(1,3).
15 edge(2,3).
16 edge(2,5).
17 edge(3,4).
18 edge(3,5).
19 edge(3,6).
20 edge(4,5).
21 edge(4,6).
22 edge(4,7).
23 edge(6,7).

```

```

?- path(2,7,Path) .
Path = [2, 3, 4, 7] ;
Path = [2, 3, 4, 6, 7] ;
Path = [2, 3, 5, 4, 7] ;
Path = [2, 3, 5, 4, 6, 7] ;
Path = [2, 3, 6, 7] ;
Path = [2, 3, 6, 4, 7] ;
Path = [2, 3, 1, 4, 7] ;
Path = [2, 3, 1, 4, 6, 7] .

?- travel(1,6,5,Path) .
Path = [6, 3, 2|5] ;
Path = [6, 4, 3, 2|5] ;
Path = [6, 7, 4, 3, 2|5] ;
Path = [6, 4, 5, 3, 2|5] ;
Path = [6, 7, 4, 5, 3, 2|5] .

```

veste.pl:

```

1 %      Arquivo: veste.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Ano: 2023
7 % Aluno: Gabriel Costa Fassarella
8 %-----
9
10 % Fonte: http://www.facom.ufu.br/~marcelo/PL/tutorial%20de%20prolog.pdf
11
12 % Aqui começa o programa      ...
13 pessoa(ale) .           %      Alexandra
14 pessoa(babe) .         %      Barbara
15 pessoa(rosa) .
16 pessoa(ana) .
17 pessoa(maria) .
18 cor(azul) .
19 cor(preto) .
20 cor(laranja) .
21 cor(amarelo) .
22 cor(vermelho) .
23 cor(lilaz) .
24 cor(verde) .

```

```

26 % este pedaço é importante para mostrar a lógica pois neste trecho estamos restringindo o uso das cores
27 % dito no problema e define uma cor para cada menina
28
29 tem_uma_cor(babe,preto).
30 tem_uma_cor(babe,amarelo).
31 tem_uma_cor(rosa,verde).
32 tem_uma_cor(ale,azul).
33 tem_uma_cor(ana,preto).
34 tem_uma_cor(ana,laranja).
35 tem_uma_cor(maria, amarelo).
36
37 %% aqui pode-se ampliar as restrições
38 nao_pode_vestir(babe,laranja).
39 nao_pode_vestir(ana, amarelo).
40 nao_pode_vestir(maria,verde).
41 nao_pode_vestir(babe,verde).
42 nao_pode_vestir(rosa,azul).
43 nao_pode_vestir(rosa,lilaz).
44

```


?- ache_tudo.

ale veste as cores azul.... epreto

ale veste as cores azul.... elaranja

ale veste as cores azul.... eamarelo

ale veste as cores azul.... evermelho

ale veste as cores azul.... elilaz

ale veste as cores azul.... everde

babe veste as cores preto.... eazul

babe veste as cores preto.... eamarelo

babe veste as cores preto.... evermelho

babe veste as cores preto.... elilaz

babe veste as cores amarelo.... eazul

babe veste as cores amarelo.... epreto

babe veste as cores amarelo.... elilaz

rosa veste as cores verde.... epreto

rosa veste as cores verde.... elaranja

rosa veste as cores verde.... eamarelo

rosa veste as cores verde.... evermelho

ana veste as cores preto.... eazul

ana veste as cores preto.... elaranja

ana veste as cores preto.... evermelho

ana veste as cores preto.... elilaz

ana veste as cores preto.... everde

ana veste as cores laranja.... eazul

ana veste as cores laranja.... epreto

ana veste as cores laranja.... evermelho

ana veste as cores laranja.... elilaz

ana veste as cores laranja.... everde

maria veste as cores amarelo.... eazul

maria veste as cores amarelo.... epreto

maria veste as cores amarelo.... elaranja

maria veste as cores amarelo.... elilaz

```
?- veste(ana, X, Y).  
X = preto,  
Y = azul ;  
X = preto,  
Y = laranja ;  
X = preto,  
Y = vermelho ;  
X = preto,  
Y = lilaz ;  
X = preto,  
Y = verde ;  
X = laranja,  
Y = azul ;  
X = laranja,  
Y = preto ;  
X = laranja,  
Y = vermelho ;  
X = laranja,  
Y = lilaz ;  
X = laranja,  
Y = verde ;  
false.
```

15. CONSTRUIR uma base de conhecimento qualquer (um programa PROLOG), contendo:
- Pelo menos 10 fatos

```

1 %      Arquivo: animais.pl
2 %
3 % Prof. Ausberto S. Castro Vera
4 % Disciplina : Paradigmas de Ling. de Programacao
5 % UENF-CCT-LCMAT-CC
6 % Ano: 2023
7 % Aluno: Gabriel Costa Fassarella
8 %-----
9
10 pelo(julio).
11 pelo(adolfo).
12 pelo(arrascaeta).
13 pelo(ney).
14 pelo(cardajal).
15 pena(rony).
16 pena(arias).
17 cauda(julio).
18 cauda(adolfo).
19 cauda(ney).
20 cauda(arrascaeta).
21 late(cardajal).
22 late(arrascaeta).
23 mia(julio).
24 mia(ney).
25 late(cardajal).
26 late(adolfo).
27 voa(rony).
28 n_voa(arias).
29 bota_ovo(rony).
30 bota_ovo(arias).
31 bota_ovo(dourado).
32 nada(dourado).

```

- Pelo menos 5 regras

```

34 gato(X):- pelo(X), cauda(X), mia(X).
35
36 cachorro(X):- pelo(X), cauda(X), late(X).
37
38 periquito(X):- pena(X), voa(X), bota_ovo(X).
39
40 galinha(X):- pena(X), bota_ovo(X), n_voa(X).
41
42 peixe(X):- bota_ovo(X), nada(X).

```

- Pelo menos 5 consultas (sugestão de testes)

```

?- gato(neey) .

?- cachorro(rony) .
false.

?- galinha(X) .
X = arias ;
false.

?- cachorro(X) .
X = adolfo ;
X = arrascaeta ;
false.

?- peixe(dourado) .

```

- Sugestão:
 - Profissões
 - Ambientes acadêmicos