

## Grupo 12:

Nomes:

- Bruno Henrique Barbosa,
- Guilherme Fenner Hey,
- Gustavo Hammerschmidt,
- Pedro Henrique Silva.

Listas de exercícios 3, 4 e 5

PUCPR

# Lista de Exercícios 3:

## Questão 1(Matching):

A partir da base de dados seguinte, resolva:

carro(joao\_da\_silva, corsa\_sedan, data\_compra(27,05,1997), 14250, placa(sj\_pinhais, 'AJV', 2453)).

carro(carlos\_pereira, cherokee, data\_compra(02,08,2000), 57400, placa(curitiba, 'KCV', 1490)).

carro(ana\_cruz, monza, data\_compra(14,06,2000), 11600, placa(curitiba, 'EAF', 3544)).

carro(carlos\_pereira, silverado, data\_compra(15,08,2001), 46800, placa(curitiba, 'LHR', 1178)).

carro(jose\_emanuel, corsa\_sedan, data\_compra(06,11,2004), 23400, placa(sj\_pinhais, 'AJV', 2273)).

carro(jose\_emanuel, clio, data\_compra(19,12,2004), 25730, placa(sj\_pinhais, 'CKP', 5194)).

a) O modelo de todos os carros comprados por “jose\_emanuel”.

Questionamento: ?- carro(jose\_emanuel, X,\_,\_).

X = corsa\_sedan ;

X = clio.

No

b) O dia, o mês e o ano da compra que “carlos\_pereira” fez da sua cherokee.

Questionamento: ?- carro(carlos\_pereira, cherokee,data\_compra(D,M,A),\_,\_).

D = 2,

M = 8,

A = 2000

Yes.

- c) O nome do comprador e o modelo dos veículos que custaram mais do que 40000 (Quarenta mil).

**Questionamento: ?- carro(X, Y,\_,Z,\_), Z>40000.**

**X = carlos\_pereira,  
Y = cherokee,  
Z = 57400 ;  
X = carlos\_pereira,  
Y = silverado,  
Z = 46800 ;  
true.**

- d) Dado o questionamento a seguir, responda os itens solicitados.

**?- carro(X, \_, data\_compra(\_,\_,2004), \_, placa(curitiba,\_,\_)).**

- d.1) Qual a(s) resposta(s) retornadas pelo Prolog para o questionamento?

**Resposta: False.**

- d.2) Qual o significado desse questionamento? Escreva o significado desse questionamento em Português.

**Resposta: Pessoa(X) que emplacou seu carro do ano 2004 em Curitiba.**

- e) Complete os questionamentos a seguir, de forma que as soluções mostradas possam ser encontradas pelo Prolog

- e.1)

**?- gosta(joao, computador, jogar(esportes(futebol, vôlei, jogo\_eletrônico(paciência))))=  
gosta(joao, computador, jogar(esportes(futebol, volei, jogo\_eletronico(X)))).**

X = paciência

Yes

e.2)

?-  $f(X, a(b, c)) = \underline{f(b, a(Z, c))}$ .

X = b

Z = b

Yes

## Questão 2

Responda as questões a e b.

```
programa( X ) :-  
    carro(X,_,_,P),  
    carro(X,_,_,P1),  
    P \= P1.
```

- a) O que o programa faz? Escreva um parágrafo sucinto em Português explicando o que o programa faz.

**Resposta:** Retorna o Nome de pessoas que possuem dois carros tais quais suas placas, obviamente, não são iguais. Todavia a cláusula  $(P \neq P1)$  está presente para avisar ao Prolog que as duas numerações de placa não são iguais afinal as letras das placas podem ser iguais, já os seus números não, caso contrário o programa retornaria o nome de todas as pessoas da base de dados.

- b) Qual será a resposta para o questionamento: ?- programa( P ).

**Resposta:** Será a mesma resposta com alteração de que agora a variável não é mais X e sim P. P é apenas nome de uma variável a qual valores serão atribuídos e retornados.

- c) Elabore um programa Prolog chamado “compra\_cidade\_ano” que retorne os nomes dos clientes que compraram um veículo em uma dada cidade e em um dado ano. Cidade e ano devem ser fornecidos como parâmetro como no questionamento a seguir:

**Programa:**

**compra\_cidade\_ano(X,Y,Z) :-**

**carro(X, \_,data\_compra(\_,\_,Z),\_,placa(Y, \_,\_))**

**Ex:                               ?- compra\_cidade\_ano(X, curitiba, 2000).**

**X = carlos\_pereira   ;**

**X = ana\_cruz       ;**

**No**

### Questão 3 – Programas Prolog

a) Considere a seguinte base de fatos Prolog:

governou(deodoro\_da\_fonseca,1891,1891).  
governou(flорiano\_peixoto,1891,1894).  
governou(prudente\_de\_moraes,1894,1898).  
governou(compos\_sales,1898,1902).  
governou(rodrigues\_alves,1902,1906).  
governou(afonso\_pena,1906,1909).  
governou(nilo\_pecanha,1909,1910).

Escreva um programa Presidente/2 que retorna o(s) nome(s) do(s) presidente(s) que governou(aram) o Brasil em um dado ano:

**Programa:                       presidente(Y, X) :- governou(X,A,B), A =< Y, Y =< B.**

**Questionamento:           ?- presidente(1891,X).**

**X = deodoro\_da\_fonseca   ;**

**X = Floriano\_peixoto   ;**

**No**

b) Escreva um programa Prolog que calcula o cubo de um dado número.

**Programa:** cubo(X,Y) :- Y is X\*\*3.  
**Exemplo:** ?- cubo(3,X).  
X = 27  
Yes

c) Escreva um programa Prolog que dados dois números, retorna o maior entre eles.

**Programa:** maior(A,B, X) :- X is B, B> A; X is A, A>B.  
**Exemplo:** ?- maior(10, 25,X).  
X = 25  
Yes

d) Escreva um programa Prolog que calcula a distância euclidiana entre dois pontos. representado apenas por duas dimensões, ou seja,  $n = 2$ . Neste exercício você deverá utilizar o predicado Prolog pow/3 e a função Prolog sqrt/1.

**Resposta:**  
power(X,Y) :- 0 =< X -> pow(X,2,Y); pow(-1\*X,2,Y).  
distancia(X1,X2,Y1,Y2,D) :- power(X1-Y1,S), power(X2-Y2,T), D is sqrt((S+T)/2).

Exemplo1: ?- distancia(2, 2, 4, 4, D).  
D = 2  
Yes

Exemplo2: ?- distancia(2, 2, 4, 5, D).  
D = 2.54951  
Yes

e) Escreva um programa que receba o salário do usuário e informe sua classe salarial:

até 700,00 → classe baixa

de 701,00 até 2.500,00 → classe média

de 2.501,00 reais até 4.500,00 → classe média-alta

acima de 4.500,00 → classe alta.

### Resposta:

```
informa_classe(X) :- (X < 700.00 -> writeln('X = classe_baixa.')),!.
```

```
informa_classe(X) :- (701.00 =< X, X =< 2500.00 -> writeln('X = classe_média.')),!.
```

```
informa_classe(X) :- (2501.00 =< X, X =< 4500.00 -> writeln('X = classe_média-  
alta')),!.
```

```
informa_classe(X) :- (4500.00 < X -> writeln('X = classe_alta')).
```

Exemplo:               ?- informa\_classe(1500, X).

                      X = classe\_media

                      Yes

# Lista de Exercícios 4:

**1)** Segundo o seguinte Banco de dados:

i)-fácil(1).	iv) -gizmo(a,1).	vii)-gizmo(d,5).
ii)-fácil(2).	v) -gizmo(b,3).	viii) -gizmo(c,3).
iii)-fácil(3).	vi) -gizmo(a,2).	ix) -gizmo(a,3).
	x) -gizmo(c,4).	

Relate as respostas obtidas das seguintes perguntas:

- a)?- fácil(2). ..... Resp.: **true.**
- b)?- fácil(X). ..... Resp.: **X = 1; X = 2; X = 3.**
- c)?- gizmo(a,X). ..... Resp.: **X = 1; X = 2; X = 3.**
- d)?- gizmo(X,3). ..... Resp.: **X = b; X = c; X = a.**
- e)?- gizmo(d,Y). ..... Resp.: **Y = 5.**
- f)?- gizmo(X,X). ..... Resp.: **false.**



2) Segundo o seguinte Banco de dados:

i)-mais\_difícil(a, 1).

ii)-mais\_difícil(c, X).

iii)-mais\_difícil(b, 4).

iv)-mais\_difícil(d, 2).

Relate as respostas obtidas das seguintes perguntas:

a)?- mais\_difícil(a, X). ..... Resp.: X = 1.

b)?- mais\_difícil(c, X). ..... Resp.: true.

c)?- mais\_difícil(X, 1). ..... Resp.: X = a; X = c; false.

d)?- mais\_difícil(X, 4). ..... Resp.: X = c; X = b; false.

3) Utilize os bancos de dados dos exercícios 4 e 5 para relatar as respostas obtidas das seguintes perguntas:

a)?- gizmo(a, X), fácil(X). ..... Resp.: X = 1; X = 2; X = 3.

b)?- gizmo(c,X), fácil(X). ..... Resp.: X = 3; false.

c)?- gizmo(d,Z), fácil(Z). ..... Resp.: false.

d)?- fácil(Y), gizmo(X,Y). ..... Resp.: Y = 1, X = a; Y = 2, X = a; Y = 3, X = b;

Y = 3, X = c; Y = 3, X = a.

e)?- fácil(X), mais\_difícil(Y,X). ..... Resp.: X = 1, Y = a; X = 1, Y = c; X = 2, Y = c;

X = 2, Y = d; X = 3, Y = c; false.

f)?- mais\_difícil(Y,X), fácil(X). ..... Resp.: Y = a, X = 1; Y = c, X = 1; Y = c, X = 2;

Y = c, X = 3; Y = d, X = 2.

4) Segundo o seguinte Banco de dados:

i) $a(a1,1).$	iv) $b(1,b1).$	vii) $c(X,Y) :- a(X,N), b(N,Y).$
ii) $a(A,2).$	v) $b(2,B).$	viii) $d(X,Y) :- a(X,N), b(Y,N).$
iii) $a(a3,N).$	vi) $b(N,b3).$	ix) $d(X,Y) :- a(N,X), b(N,Y).$

Relate as respostas obtidas das seguintes perguntas:

- a)  $?- a(X,2).$  ..... Resp.: **true**;  $X = a3.$   
b)  $?- b(X, kalamazu).$  ..... Resp.:  $X = 2$ ; **false**.  
c)  $?- c(X, b3).$  ..... Resp.:  $X = a1$ ; **true**; **true**;  $X = a3$ ;  $X = a3.$   
d)  $?- c(X, Y).$  ..... Resp.:  $X = a1, Y = b1$ ;  $X = a1, Y = b3$ ; **true**;  $Y = b3$ ;  
 $X = a3, Y = b1$ ;  $X = a3$ ;  $X = a3, Y = b3.$   
e)  $?- d(X,Y).$  ..... Resp.:  $X = a1$ ;  $Y = 2$ ;  $Y = 2$ ;  $X = a3, Y = 1$ ;  
 $X = a3, Y = 2$ ;  $X = a3$ ;  $X = 1, Y = b3$ ;  $X = 2, Y = b1$ ;  $X = 2$ ;  $X = 2, Y = b3$ ;  $Y = b3.$

5) [Linguagem Prolog] Dada a seguinte base de fatos:

$p(a).$      $q(a,1).$      $r(1,1).$      $r(3,5).$   
 $p(b).$      $q(a,2).$      $r(1,2).$      $r(3,6).$   
           $q(b,3).$      $r(2,3).$      $r(4,7).$   
           $q(b,4).$      $r(2,4).$      $r(4,8).$

Qual é o resultado para as queries seguintes?

- a)  $p(X), q(X,Y), r(Y,Z).$   
b)  $!, p(X), q(X,Y), r(Y,Z).$   
c)  $p(X), !, q(X,Y), r(Y,Z).$   
d)  $p(X), q(X,Y), !, r(Y,Z).$   
e)  $p(X), q(X,Y), r(Y,Z), !.$

Respostas:

- a)  $X = a, Y = Z, Z = 1$ ;  
 $X = a, Y = 1, Z = 2$ ;  
 $X = a, Y = 2, Z = 3$ ;  
 $X = a, Y = 2, Z = 4$ ;  
 $X = b, Y = 3, Z = 5$ ;  
 $X = b, Y = 3, Z = 6$ ;  
 $X = b, Y = 4, Z = 7$ ;  
 $X = b, Y = 4, Z = 8.$

b)  $X = a, Y = Z, Z = 1 ;$   
 $X = a, Y = 1, Z = 2 ;$   
 $X = a, Y = 2, Z = 3 ;$   
 $X = a, Y = 2, Z = 4 ;$   
 $X = b, Y = 3, Z = 5 ;$   
 $X = b, Y = 3, Z = 6 ;$   
 $X = b, Y = 4, Z = 7 ;$   
 $X = b, Y = 4, Z = 8.$

c)  $X = a, Y = Z, Z = 1 ;$   
 $X = a, Y = 1, Z = 2 ;$   
 $X = a, Y = 2, Z = 3 ;$   
 $X = a, Y = 2, Z = 4.$

d)  $X = a, Y = Z, Z = 1 ;$   
 $X = a, Y = 1, Z = 2.$

e )  $X = a, Y = Z, Z = 1.$

6) Considere o seguinte programa Prolog, que deve retornar como terceiro argumento o maior entre dois números:

```
max(X, Y, X) :- X >= Y.
max(X, Y, Y) .
```

a) Elabore um questionamento mostrando que este programa está incorreto.

Questionamento:  $?- \text{max}(2, 1, X).$

Output:  $X = 2 ;$

$X = 1.$

Resposta: Quando  $x$  é maior ou igual a  $y$ , o programa testa a primeira condição e a segunda o que, obviamente, invalida todo o conceito de um número ser maior ou igual a outro. Para corrigir o erro, é necessário usar o comando cut.

b) Modifique o programa de forma que ele passe a funcionar corretamente.

Forma correta do programa:

```
max(X, Y, X) :- X >= Y, !.
```

```
max(X, Y, Y).
```

7) O programa seguinte classifica um número em negativo, zero ou positivo.

`classe(N, negativo) :- N < 0.`

`classe(0, zero).`

`classe(N, positivo) :- N > 0.`

Como você tornaria este programa mais eficiente utilizando o comando cut?

**Resposta:**

`classe(N, negativo) :- N < 0, !.`

`classe(N, positivo) :- N > 0, !.`

`classe(0, zero).`

# Lista de exercícios 5 :

A partir da base de fatos da Lista de Exercícios 2, escreva os seguintes programas Prolog:

- 1) Fazer a leitura do teclado do nome de uma família e imprimir todos os membros dessa família. Caso nenhuma pessoa seja encontrada, então fornecer uma mensagem de aviso.

**finder(Sobrenome) :-**

```
    pessoa(N, Sobrenome, nascimento(A,B,C), origem(X),profissao(Y)),nl,
    write('Nome: '),write(N),nl,
    write('Sobrenome: '),write(Sobrenome),nl,
    write('Nascimento: '),write(A),write('/'),write(B),write('/'),write(C),nl,
    write('Origem: '),write(X),nl,
    write('Profissão: '),write(Y),nl,nl,
    fail.
```

**finder(Sobrenome) :-**

```
    not(pessoa(_, Sobrenome,_,_,_)),
    write('Nenhuma pessoa dessa família foi encontrada.').
```

**encontra\_membros :-**

```
    write('Por favor, entre com o nome da família: '),nl,
    read(Sobrenome),nl,
    write('Os membros dessa família são: '),nl,nl,
    finder(Sobrenome).
```

Ex:

?- encontra\_membros.

Por favor, entre com o nome da família:

|: silva\_pinheiro.

Os membros dessa família são:

Nome: carlos

Sobrenome: silva\_pinheiro

Nascimento: 1/4/1962

Origem: guarulhos

Profissão: mecanico

Nome: ana\_claudia

Sobrenome: silva\_pinheiro

Nascimento: 18/7/1966

Origem: castro

Profissão: do\_lar

Nome: silvia

Sobrenome: silva\_pinheiro

Nascimento: 27/12/1998

Origem: sao\_paulo

Profissão: nenhuma

Nome: carolina

Sobrenome: silva\_pinheiro

Nascimento: 27/12/1998

Origem: sao\_paulo

Profissão: nenhuma

Nome: claudia

Sobrenome: silva\_pinheiro

Nascimento: 15/4/2003

Origem: curitiba

Profissão: nenhuma

false.

## 2) Verificar se a profissão de uma pessoa está correta.

Programa:

```
verifica_profissao(N, S, P) :-
```

```
    pessoa(N, S, _, profissao(P)) -> true,
```

```
    write(N),write(' é '),write(P).
```

```
verifica_profissao(N, S, P) :-
```

```
    not(pessoa(N, S, _, profissao(P))),
```

```
    write(N),write(' não é '),write(P).
```

EX:

```
?- verifica_profissao(joao, pereira_santos, engenheiro).
```

```
joao é engenheiro
```

```
true.
```

## 3) Retorna a diferença em dia, mês e ano da idade de duas pessoas.

Ex.) ?- diferenca\_idade(joao, pereira\_santos, patricia,  
nogueira\_carvalho).

Programa:

```
opp(D1,M1,A1,D2,M2,A2,Delta) :-
```

```
    A1 =:= A2, M1 =:= M2, D2 =:= D1,
```

```
    Delta is 0, !.
```

```
opp(D1,M1,A1,D2,M2,A2,Delta) :-
```

```
    A1 =:= A2, M1 =:= M2, D1 > D2,
```

```
    Delta is (D1-D2), !.
```

**opp(D1,M1,A1,D2,M2,A2,Delta) :-**

**A1 ::= A2, M1 ::= M2, D2 > D1,**

**Delta is (D2-D1), !.**

**opp(D1,M1,A1,D2,M2,A2,Delta) :-**

**A1 ::= A2, M2 > M1,**

**Delta is round(((M2-M1)\*153/5)+D2-D1), !.**

**opp(D1,M1,A1,D2,M2,A2,Delta) :-**

**A1 ::= A2, M1 > M2,**

**Delta is round(((M1-M2)\*153/5)+D1-D2), !.**

**opp(D1,M1,A1,D2,M2,A2,Delta) :-**

**A2 > A1,**

**Delta is round((365-((((M1-1)\*153/5))+D1))+((A2-A1-1)\*1461/4)+((M2-1)\*153/5)+D2), !.**

**opp(D1,M1,A1,D2,M2,A2,Delta) :-**

**A1 > A2,**

**Delta is round((365-((((M2-1)\*153/5))+D2))+((A1-A2-1)\*1461/4)+((M1-1)\*153/5)+D1).**

**dias\_pra\_data(Delta, Ano, Mes, Dia) :-**

**Ano is (Delta//365),**

**Mes is ((Delta mod 365)//30),**

**Dia is ((Delta mod 365) mod 30).**

**print\_ano(Ano) :- Ano ::= 1, write(Ano),write(' ano, '),!.**



```
print_ano(Ano) :- write(Ano), write(' anos, ').
```

```
print_mes(Mes) :- Mes >= 1, write(Mes),write(' mês e '),!.
```

```
print_mes(Mes) :- write(Mes), write(' meses e ').
```

```
print_dia(Dia) :- Dia >= 1, write(Dia),write(' dia. '),!.
```

```
print_dia(Dia) :- write(Dia), write(' dias. ').
```

```
diferença_idade(N1,S1,N2,S2) :-
```

```
    pessoa(N1,S1,nascimento(D1,M1,A1),_,_),
```

```
    pessoa(N2,S2,nascimento(D2,M2,A2),_,_),
```

```
    opp(D1,M1,A1,D2,M2,A2,Delta),
```

```
    dias_pra_data(Delta, Ano, Mes, Dia),
```

```
    write('A diferença de idade é de '),
```

```
    print_ano(Ano),
```

```
    print_mes(Mes),
```

```
    print_dia(Dia).
```

Observações sobre o código:

- O programa funciona sobre a base de dados fornecida para o exercício.

- O cálculo dos anos e meses são estimativas aproximadas baseadas nas seguintes equações:  $(\text{diferença entre os anos}) * 1461 / 4$  para a diferença entre os anos (necessário saber qual dos anos é o maior ou usar o comando `abs/1` para evitar o surgimento de um dado negativo); e meses  $* 153/5$  para o cálculo dos meses. Ambas equações retornarão o valor de, seja anos, seja meses, em dias. As equações podem retornar números float , para tanto, foi utilizado o comando `round/1` para arredondar o dado. Essa imprecisão pode resultar em uma inconsistência com a verdadeira quantidade de dias entre as datas de no máximo (no pior cenário) 10 dias de diferença, quanto mais extremas(para dias de 1 a 3, 29 a 31; e, para meses, 1 a 2, 11 a 12) as datas, maior é a inconsistência entre elas.

Questionamento :   ?- `diferença_idade(joao, pereira_santos, patricia, nogueira_carvalho)`.

Resposta :

A diferença de idade é de 14 anos, 3 meses e 16 dias.

true.