

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Lucas Machado da Palma	12200640
Luiz Henrique Urias	12200647

TRABALHO 2B

– Base de Dados –

:- dynamic(nome/1).
nome('Suzana Vieira').
nome('Mary Jane').
nome('Pablo Monte').
nome('Lucifer Petry').
nome('Bruce Wayne').
nome('Oliver Queen').
nome('Natasha Romanov').
nome('Logan Silva').
nome('Silva e Silva').
nome('Barry Allen').

:- dynamic(nasc/4).
nasc('Bruce Wayne', 22, 02, 1939).
nasc('Suzana Vieira', 10, 20, 1959).
nasc('Mary Jane', 15, 07, 1950).
nasc('Lucifer Petry', 06, 06, 1966).
nasc('Oliver Queen', 15, 10, 1979).
nasc('Natasha Romanov', 10, 06, 1978).
nasc('Logan Silva', 29, 12, 1969).
nasc('Silva e Silva', 22, 02, 1982).
nasc('Barry Allen', 22, 02, 1986).
nasc('Pablo Monte', 25, 11, 1982).

:- dynamic(professor/1).
professor('Joao Tenorio').
professor('Aline Vemcurt').
professor('Tereza Montegrando').
professor('Clorisvaldo de Aumeida').
professor('Pedro Pedrozo').

:- dynamic(orientou/2).
orientou('Joao Tenorio', 'Barry Allen').
orientou('Tereza Montegrando', 'Pablo Monte').
orientou('Tereza Montegrando', 'Suzana Vieira').
orientou('Clorisvaldo de Aumeida', 'Mary Jane').
orientou('Joao Tenorio', 'Lucifer Petry').
orientou('Joao Tenorio', 'Bruce Wayne').
orientou('Clorisvaldo de Aumeida', 'Oliver Queen').
orientou('Joao Tenorio', 'Natasha Romanov').
orientou('Pedro Pedrozo', 'Logan Silva').
orientou('Pedro Pedrozo', 'Silva e Silva').

:- dynamic(curso/1).
curso('Ciencias da Computacao').
curso('Letras').
curso('Numeros').
curso('Artes Marciais').
curso('Sistemas').
curso('Engenharia Mecanica').

:- dynamic(cursou/4).
cursou('Bruce Wayne', 'Artes Marciais', 1980, 1985).
cursou('Oliver Queen', 'Artes Marciais', 1982, 1987).

cursou('Suzana Vieira', 'Letras', 1982, 1987).
cursou('Mary Jane', 'Engenharia Mecanica', 1990, 2000).
cursou('Pablo Monte', 'Numeros', 1990, 1994).
cursou('Pablo Monte', 'Sistemas', 1995, 2000).
cursou('Lucifer Petry', 'Letras', 1966, 1972).
cursou('Natasha Romanov', 'Ciencias da Computucao', 1995, 2000).
cursou('Logan Silva', 'Artes Marciais', 1970, 1977).
cursou('Silva e Silva', 'Numeros', 2000, 2006).
cursou('Barry Allen', 'Sistemas', 1997, 2001).

:- dynamic(empresa/1).
empresa('LABSEC').
empresa('UFSC').
empresa('Google').
empresa('Microsoft').
empresa('Wayne Enterprises').
empresa('Queen Industries').
empresa('Samsumg').

:- dynamic(telefone/1).
telefone(33333333).
telefone(44444444).
telefone(76547636).
telefone(87568444).
telefone(87468923).
telefone(98769867).
telefone(49654348).
telefone(87565439).
telefone(98766543).
telefone(35899864).

:- dynamic(cidade/1).
cidade('Florianopolis').
cidade('Gotham').
cidade('Starling').
cidade('NewYork').
cidade('Central').

:- dynamic(telefonede/2).
telefonede('Suzana Vieira', 33333333).
telefonede('Pablo Monte', 44444444).
telefonede('Bruce Wayne', 76547636).
telefonede('Natasha Romanov', 87468923).
telefonede('Barry Allen', 98769867).
telefonede('Silva e Silva', 49654348).
telefonede('Oliver Queen', 87468923).
telefonede('Lucifer Petry', 87565439).
telefonede('Mary Jane', 98766543).
telefonede('Logan Silva', 35899864).

:- dynamic(cidadede/2).
cidadede('Pablo Monte', 'Florianopolis').
cidadede('Logan Silva', 'Florianopolis').
cidadede('Suzana Vieira', 'Central').
cidadede('Bruce Wayne', 'Gotham').
cidadede('Natasha Romanov', 'NewYork').
cidadede('Barry Allen', 'Central').

cidade('Silva e Silva', 'NewYork').
cidade('Oliver Queen', 'Starling').
cidade('Lucifer Petry', 'Florianopolis').
cidade('Mary Jane', 'Central').

:- dynamic(trabalhou/2).
trabalhou('Pablo Monte', 'LABSEC').
trabalhou('Pablo Monte', 'Microsoft').
trabalhou('Pablo Monte', 'Google').
trabalhou('Logan Silva', 'Google').
trabalhou('Suzana Vieira', 'Google').
trabalhou('Bruce Wayne', 'Wayne Enterprises').
trabalhou('Natasha Romanov', 'Queen Industries').
trabalhou('Barry Allen', 'UFSC').
trabalhou('Silva e Silva', 'UFSC').
trabalhou('Oliver Queen', 'Queen Industries').
trabalhou('Lucifer Petry', 'LABSEC').
trabalhou('Mary Jane', 'Samsung').

:- dynamic(referencia/2).
referencia('Oliver Queen', 'Bruce Wayne').
referencia('Logan Silva', 'Bruce Wayne').
referencia('Silva e Silva', 'Pablo Monte').
referencia('Lucas Silva', 'Pablo Monte').
referencia('Mary Jane', 'Pablo Monte').
referencia('Silva e Silva', 'Lucifer Petry').
referencia('Mary Jane', 'Barry Allen').
referencia('Natasha Romanov', 'Barry Allen').
referencia('Natasha Romanov', 'Pablo Monte').
referencia('Natasha Romanov', 'Mary Jane').
referencia('Bruce Wayne', 'Natasha Romanov').

:- dynamic(inst/1).
inst('UFSC').
inst('UDESC').
inst('UNISUL').
inst('UFSM').
inst('QueenSchool').
inst('Harvard').
inst('Vida').

:- dynamic(estudouEm/2).
estudouEm('Pablo Monte', 'UNISUL').
estudouEm('Logan Silva', 'UNISUL').
estudouEm('Suzana Vieira', 'UNISUL').
estudouEm('Suzana Vieira', 'UFSC').
estudouEm('Mary Jane', 'UDESC').
estudouEm('Lucifer Petry', 'UFSM').
estudouEm('Oliver Queen', 'QueenSchool').
estudouEm('Silva e Silva', 'Vida').
estudouEm('Barry Allen', 'Harvard').
estudouEm('Natasha Romanov', 'UFSC').

:- dynamic(xp/5).
xp('Pablo Monte', 'Facilitador', 'LABSEC', 2001, 2007).
xp('Pablo Monte', 'Testador', 'Microsoft', 2008, 2010).

xp('Pablo Monte', 'Gerente Organizacional dos Facilitadores', 'Google', 2011, 2014).
xp('Logan Silva', 'Programador', 'Google', 2000, 2009).
xp('Logan Silva', 'Fisiculturista', 'UFSC', 2010, 2015).
xp('Suzana Vieira', 'Secretaria', 'Google', 1980, 2012).
xp('Bruce Wayne', 'Bilionário', 'Wayne Enterprises', 2010, 2015).
xp('Natasha Romanov', 'Arquiteta de Projetos', 'Queen Industries', 1997, 2014).
xp('Barry Allen', 'Facilitador', 'UFSC', 2000, 2009).
xp('Silva e Silva', 'Facilitador', 'UFSC', 2001, 2008).
xp('Oliver Queen', 'CEO', 'Queen Industries', 2013, 2014).
xp('Lucifer Petry', 'Comentarista', 'LABSEC', 2005, 2007).
xp('Mary Jane', 'Diretora de Marketing', 'Microsoft', 1970, 2005).

listlen([], 0).
listlen([H|T], N) :- listlen(T, N1), N is N1 + 1.
conta(P, Contador):-findall(1, P, Lista), listlen(Lista, Contador).
acumula([], 0).
acumula([H|T], N):-acumula(T, N1), N is N1 + H.
acumulaMedia([], 0).
acumulaMedia([H|T], K):-acumulaMedia(T, K1), totalEstudado(H,L), K is K1 + L.
max([X],X).
max([X|Xs],X) :- max(Xs,Y), X >= Y.
max([X|Xs],N) :- max(Xs,N), N > X.
min([X], X).
min([X|Xs], X):- min(Xs,Y), X =< Y.
min([X|Xs], N):- min(Xs,N), N < X.
inserir(Elem, Lista, [Elem|Lista]).

-- TRABALHO PROLOG B PARTE 2 --

1) Escreva uma regra para a criação apropriada de um grafo (escolha uma das formas vistas em aula) de todas as pessoas (nós ou vértices) e suas respectivas referências (arestas).

```
connected(X,Y) :- referencia(X,Y).  
connected(X,Y) :- referencia(Y,X).
```

Para tanto usamos a meta 'referencia' da antiga base de dados, como conversado com o professor anteriormente.

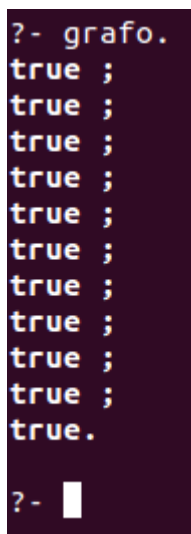
```
referencia('Oliver Queen', 'Bruce Wayne').  
referencia('Logan Silva', 'Bruce Wayne').  
referencia('Silva e Silva', 'Pablo Monte').  
referencia('Lucas Silva', 'Pablo Monte').  
referencia('Mary Jane', 'Pablo Monte').  
referencia('Silva e Silva', 'Lucifer Petry').  
referencia('Mary Jane', 'Barry Allen').  
referencia('Natasha Romanov', 'Barry Allen').  
referencia('Natasha Romanov', 'Pablo Monte').  
referencia('Natasha Romanov', 'Mary Jane').  
referencia('Bruce Wayne', 'Natasha Romanov').
```

2) Faça a persistência deste grafo em um arquivo '.pl'.

Para isso usamos um arquivo externo chamado grafo.pl que apenas monta uma representação das ligações entre as pessoas da base de dados através de suas referências. Como informado pelo professor em sala, não há a necessidade de uso posterior do que será gravado no arquivo, por esse motivo, uma representação textual é mais do que suficiente.

A forma de representação escolhida, foi a 'Representação de grafos(II)' cláusulas na forma amigável slide número 11 da apostila de aula, com uma pequena modificação, pois não usamos vírgulas e sim uma nova linha entre cada ligação.

```
grafo:-referencia(P,O), open('grafo.pl', append, Stream), write(Stream,P), write(Stream,' - '), write(Stream,  
O), nl(Stream), close(Stream).
```



```
?- grafo.  
true ;  
true ;  
true ;  
true ;  
true ;  
true ;  
true ;  
true ;  
true ;  
true ;  
true.  
?- 
```

Resultado no arquivo 'grafo.pl'.

```

Oliver Queen - Bruce Wayne
Logan Silva - Bruce Wayne
Silva e Silva - Pablo Monte
Lucas Silva - Pablo Monte
Mary Jane - Pablo Monte
Silva e Silva - Lucifer Petry
Mary Jane - Barry Allen
Natasha Romanov - Barry Allen
Natasha Romanov - Pablo Monte
Natasha Romanov - Mary Jane
Bruce Wayne - Natasha Romanov

```

3) Dadas duas pessoas P1 e P2, escreva uma regra que devolva uma lista de um conjunto de outras pessoas (uma ligada a outra) que possam conectá-las. Ou seja, apresente caminhos possíveis no grafo de P1 a P2 (pode não haver caminho possível ou as duas pessoas serem ligadas diretamente) .

```

path(A,B,Path) :- travel(A,B,[A],Q), reverse(Q,Path).
travel(A,B,P,[B|P]) :-connected(A,B).
travel(A,B,Visited,Path) :- connected(A,C),C \== B,\+member(C,Visited),travel(C,B,[C|Visited],Path).

```

```

:- dynamic(referencia/2).
referencia('Oliver Queen', 'Bruce Wayne').
referencia('Logan Silva', 'Bruce Wayne').
referencia('Silva e Silva', 'Pablo Monte').
referencia('Lucas Silva', 'Pablo Monte').
referencia('Mary Jane', 'Pablo Monte').
referencia('Silva e Silva', 'Lucifer Petry').
referencia('Mary Jane', 'Barry Allen').
referencia('Natasha Romanov', 'Barry Allen').
referencia('Natasha Romanov', 'Pablo Monte').
referencia('Natasha Romanov', 'Mary Jane').
referencia('Bruce Wayne', 'Natasha Romanov').

```

```

?- path('Pablo Monte', 'Bruce Wayne', Caminho).
Caminho = ['Pablo Monte', 'Mary Jane', 'Barry Allen', 'Natasha Romanov', 'Bruce Wayne'] ;
Caminho = ['Pablo Monte', 'Mary Jane', 'Natasha Romanov', 'Bruce Wayne'] ;
Caminho = ['Pablo Monte', 'Natasha Romanov', 'Bruce Wayne'] ;
false.

```

?- █

4) Mostre o menor número de passos em um caminho entre P1 e P2.

```

tamanho(Nome1, Nome2, N, P):-path(Nome1, Nome2, P), listlen(P, N).
tamanhos(Nome1, Nome2, V):- findall(N, tamanho(Nome1, Nome2, N,_), Lista), min(Lista, V).
menorCaminho(N,M,C):-tamanhos(N,M,T),tamanho(M,N,T,C).

```

```

?- menorCaminho('Pablo Monte', 'Bruce Wayne', MenorCaminho).
MenorCaminho = ['Bruce Wayne', 'Natasha Romanov', 'Pablo Monte'] ;
false.

```

?- █

Como pode ser verificado na imagem acima, o menor caminho de pessoas referenciadas entre 'Pablo Monte' e 'Bruce Wayne' é de tamanho 3 e respeita a ordem estabelecida na lista. Quando perguntado por mais opções retorna falso, o que não ocorreria se houvessem mais de um caminho do mesmo tamanho.

