

# Trabalho 1 de PLP: Prolog

Felipe Sampaio de Souza, RA: 619523

Marcio Lima Inácio, RA: 587265

## 1 Código fonte e documentação

---

```
%% desparentize(+ListaIn, ?ListaOut)
% Elimina sublistas de ListaIn.
% Ignora elementos indesejáveis: variáveis.
desparentize([], []) :- !.
desparentize([Cabeca | Cauda], [Cabeca | CaudaOut]) :- atomic(Cabeca),
                                                    desparentize(Cauda, CaudaOut), !.
desparentize([Cabeca | Cauda], ListaOut) :- is_list(Cabeca),
                                                    desparentize(Cabeca, Aux1),
                                                    desparentize(Cauda, Aux2),
                                                    append(Aux1, Aux2, ListaOut), !.
desparentize([Cabeca | Cauda], [Cabeca | CaudaOut]) :- compound(Cabeca),
                                                    desparentize(Cauda, CaudaOut), !.
desparentize(_ | Cauda, ListaOut) :- desparentize(Cauda, ListaOut).

%% tira_nao_comuns(+Lista1, +Lista2, ?ListaOut)
% Elimina elementos de Lista1 que nao estao em Lista2.
tira_nao_comuns([], _, []) :- !.
tira_nao_comuns([Cabeca | Cauda], Lista2, [Cabeca | CaudaOut]) :- member(Cabeca, Lista2),
                                                                    tira_nao_comuns(Cauda, Lista2, CaudaOut), !.
tira_nao_comuns(_ | Cauda, Lista2, ListaOut) :- tira_nao_comuns(Cauda, Lista2, ListaOut).

%% quantas_vezes(+Elem, +ListaIn, ?N)
% Conta quantas vezes Elem ocorre em ListaIn.
quantas_vezes(_, [], 0) :- !.
quantas_vezes(X, [X | Cauda], N) :- quantas_vezes(X, Cauda, Aux), N is Aux + 1, !.
quantas_vezes(X, _ | Cauda, N) :- quantas_vezes(X, Cauda, N).

%% elimina_ocorrencias(+Elem, +ListaIn, ?ListaOut)
% Elimina todas as ocorrencias de Elem em ListaIn.
elimina_ocorrencias(_, [], []) :- !.
elimina_ocorrencias(X, [X | Cauda], Lout) :- elimina_ocorrencias(X, Cauda, Lout), !.
elimina_ocorrencias(X, [Cabeca | Cauda], [Cabeca | CaudaOut]) :- elimina_ocorrencias(X, Cauda, CaudaOut).

%% monta_pares(+ListaIn, ?ListaOut)
% Cria uma lista de pares no seguinte formato:
% [Elemento, N]
% Para cada elemento em ListaIn
% Em que N e o numero de vezes que Elemento ocorre em ListaIn
monta_pares([], []) :- !.
monta_pares([Cabeca | Cauda], [[Cabeca, N] | CaudaOut]) :- quantas_vezes(Cabeca, [Cabeca | Cauda], N),
                                                            elimina_ocorrencias(Cabeca, Cauda, ListaAux),
                                                            monta_pares(ListaAux, CaudaOut).

%% conta_atomos(+Lista1, +Lista2, ?ListaOut)
% Cria uma lista de pares no seguinte formato
% [Elemento, N]
% Para cada elemento em comum entre Lista1 e Lista2
% Em que N e o numero de vezes que Elemento ocorre em ambas listas.
conta_atomos(Lista1, Lista2, ListaOut) :- desparentize(Lista1, Aux1),
                                                    desparentize(Lista2, Aux2),
                                                    tira_nao_comuns(Aux1, Aux2, ListaFinal1),
                                                    tira_nao_comuns(Aux2, Aux1, ListaFinal2),
                                                    append(ListaFinal1, ListaFinal2, ListaFinal),
                                                    monta_pares(ListaFinal, ListaOut).
```

```

%% main()
% Predicado principal.
% Le duas listas do usuario.
% Retorna para o usuario uma lista de pares descrito em conta_atomos.
main() :- write('Digite a primeira lista: '), nl,
          read(Lista1),
          write('Digite a segunda lista: '), nl,
          read(Lista2),
          conta_atomos(Lista1, Lista2, ListaOut),
          write('A lista de pares e: '), nl,
          write(ListaOut).

```

---

## 2 Execução

```

Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- main().
Digite a primeira lista:
|: [a, b, Z, [a, x], [], [5,z], [], par(c,d)].
Digite a segunda lista:
|: [ a, 4.6, x, [], x, b, [], par(c,d)].
A lista de pares é:
[[a,3],[b,2],[x,3],[[],4],[par(c,d),2]]
true.

2 ?-

```

Figure 1: Primeiro exemplo de execução

```

2 ?- main().
Digite a primeira lista:
|: [2, [], [h, e, l, o, i, s, [a]], [], plp(t1)].
Digite a segunda lista:
|: [plp(X), [[[]]], a, [[2]]].
A lista de pares é:
[[2,2],[[],3],[a,2],[plp(t1),2]]
true.

2 ?-

```

Figure 2: Segundo exemplo de execução