



# Universidade do Vale do Itajaí

## Campus KobraSol

**Paradigmas de Linguagens de Programação**

Lista de Exercícios M2

*Aluno:* Mauricio Macário de Farias Junior

*Professora :* Fernanda dos Santos Cunha

São José, 2018, 27 de Maio

**Exercícios 1, 4 e 7 estão com o código em anexo.**

**Exercício 2. Funcionamento das Listas [H|T] – teórico**

Preveja os resultados das seguintes questões em Prolog:

a) ?- [a|[b,c,d]] = [a,b,c,d].

R: True, Pois as duas estruturas são iguais.

b) ?- [H|T] = [a, b, c, d, e].

R: H = a

T = [a,b,c,d,e]

c) ?- [H|T] = [a, [b,c,d]].

R: H = a

T = [[b,c,d]]

d) ?- [H|T] = [].

R: False, não tem o que atribuir

e) ?- [One, Two | T] = [apple, sprouts, fridge, milk].

R: One = apple

Two = sprouts

T = [fridge, milk]

**Exercício 3. Funcionamento das Listas [H|T] – teórico**

Resolva as igualdades, dizendo quais são os valores finais das variáveis.

a) ?- lista([a,[b],c,[d]]) = lista([\_|[X|X]]).

R: Retornará false, pois não consegue atribuir, já que no segundo predicado é necessário um header igual ao corpo.

b) ?- lista([[a],[b],C])=lista([C,B,[a]]).

R: C = [a]

B = [b]

c) ?- lista([c,c,c])=lista([X|[X|\_]]).

R: X = c

d) ?- lista([a,[b,c]])=lista([A,B,C]).

R: Retornará false, pois o primeiro predicado tem 2 parâmetros e o segundo tem 3.

e) ?- [1,2,3,4,5,6,7]=[X,Y,Z|D].

R: X = 1

Y = 2

Z = 3

D = [4,5,6,7]

### **Exercício 5. Efeito do Cut**

Suponha a seguinte base de fatos em Prolog

`dados(um).`

`dados(dois).`

`dados(tres).`

a) Qual o resultado da seguinte pergunta?

`cut_teste_a(X) :- dados(X).`

`cut_teste_a('ultima_clausula').`

?- `cut_teste_a(X), write(X), nl, fail.`

R: O "`cut_teste_a(x)`" vai sempre achar um resultado, que então o "`write(x)`", irá printar na tela, após isso será pulado para a proxima linha e a pergunta irá falhar e repetir o processo com todos os dados e por ultimo com o ultimo predicado.

Saida:

um

dois

tres

ultima\_clausula

false

b) Qual o resultado do seguinte programa com um Cut no final da primeira clausula?

`cut_teste_b(X):- dados(X), !.`

`cut_teste_b('ultima_clausula').`

?- `cut_teste_b(X), write(X), nl, fail.`

R: Apenas irá printar o primeiro dado

Saida:

um

false

c) Qual o resultado do seguinte programa com um Cut no meio dos dois objetivos?

```
cut_teste_c(X,Y) :- dados(X), !, dados(Y).
```

```
cut_teste_c('ultima_clausula').
```

```
?- cut_teste_c(X,Y), write(X-Y), nl, fail.
```

R: Irá printar todas as combinação com o primeiro dado(X)

Saida:

um-um

um-dois

um-tres

false

### **Exercício 6. Cuts Verdes e Vermelhos**

Explique a função dos ‘cuts’ incluídos no programa abaixo.

```
imaturo(X):- adulto(X), !, fail.
```

```
imaturo(X).
```

```
adulto(X):- pessoa(X), !, idade(X, N), N>=18.
```

```
adulto(X):- tartaruga(X), !, idade(X, N), N>=50.
```

R: No predicado “imaturo(X)” irá fazer com que nenhuma outra possibilidade ou predicado seja testado, e portanto seja retornado apenas um false por conta do fail a seguir, por conta disso o proximo predicado só vai ser chamado se o predicado adulto(X) falhar, e esse predicado simplesmente irá retornar True, ou seja, o primeiro cut basicamente serve para negar a saida, os cuts incluídos nos predicados “adulto(X)” são para garantir que o outro predicado não seja testado também, pois se um “X” é uma pessoa, não será uma tartaruga, por isso não há necessidade do teste.