

**Disciplina:** Paradigmas de Programação  
**Professor:** Maicon Rafael Zatelli  
**Entrega:** Moodle

## Trabalho IV - Programação Lógica - Prolog

**Atenção:** Faça um ZIP com todos os arquivos de solução. Use o nome do arquivo de maneira a entender qual problema você está resolvendo. Por exemplo, problema1.pl, problema2.pl e assim por diante. Responda todas as questões referentes ao problema 1, no arquivo do problema 1. Faça o mesmo para as questões referentes ao problema 2, e assim por diante.

- A legibilidade, organização do código, bem como o uso de comentários também serão considerados na avaliação.
  - Note que a somatória dos pontos passa de 10, porém mesmo acertando todos os problemas, a nota máxima será 10.
  - Para cada questão, **crie exemplos de consultas** feitas e **anote a saída** do seu programa para cada consulta. Coloque o resultado no mesmo arquivo do código fonte, ao final, e em comentário.
1. Altere o arquivo `familia.pl` disponível juntamente com este enunciado. Assuma que nenhum nome de pessoa é repetido, isto é, cada nome de pessoa no arquivo é um nome único (um identificador). Crie novas pessoas e altere idades, se for necessário. Faça os itens abaixo:
    - A (1.0):** Crie uma regra `casados(X,Y)`, onde a pessoa X é casada com a pessoa Y se elas possuem algum filho em comum. Note que nenhuma pessoa é casada com ela mesma.
    - B (1.0):** Crie uma regra `cunhados(X,Y)`, onde X é cunhado ou cunhada de Y.
    - C (1.0):** Crie regras `sobrinho(X,Y)` e `sobrinha(X,Y)`, onde X é sobrinho ou sobrinha de Y, dependendo da regra.
    - D (1.0):** Crie uma regra `casadoMaisVezes(X)`, que deve retornar verdadeiro se X pode ter sido casado(a) mais de uma vez, ou seja, possui filhos com diferentes pessoas.
    - E (1.0):** Crie uma consulta para retornar a lista ordenada (e sem repetições) de todos os avôs ou avós com idade maior de 50 anos.
    - F (1.0):** Crie uma consulta para retornar a lista ordenada (e sem repetições) de todas as pessoas que não são avôs ou avós e que possuem filhos com idade entre 30 (inclusive) e 90 (inclusive) anos.
    - G (1.0):** Crie uma consulta para retornar, para cada avô (masculino), a lista de seus descendentes, ou seja, agrupe os descendentes por avô.
  2. **(1.5)** Crie uma regra `repetidos(L1,L2)`, a qual recebe uma lista L1 como entrada e deve devolver uma lista L2 com apenas os elementos que aparecem mais de uma vez em L1, conforme exemplo abaixo. **Não utilize** nenhuma função pronta do Prolog para esta tarefa.  
  

```
?- repetidos([1,2,3,2,1,1,1,2,2,4,5,3],L).  
L = [3, 1, 2].
```

  

```
?- repetidos([1,2,3,4,5],L).  
L = [].
```
  3. **(2.0)** Crie uma regra `rotacionarDireita(L1, N, L2)`, a qual recebe uma lista L1 e um inteiro N como entrada e deve devolver uma lista L2 rotacionada à direita N vezes. Veja o exemplo abaixo. **Não utilize** nenhuma função pronta do Prolog para esta tarefa.  
  

```
?- rotacionarDireita([1,2,3,4,5,6,7],1,L).  
L = [7, 1, 2, 3, 4, 5, 6].
```

  

```
?- rotacionarDireita([1,2,3,4,5,6,7],2,L).  
L = [6, 7, 1, 2, 3, 4, 5].
```