Pontifícia Universidade Católica de Minas Gerais
***Campus* de Poços de Caldas**

Trabalho Teórico-Prático

Curso: Bacharelado em Ciência da Computação
Disciplina: Paradigmas de Linguagens de Programação
Período: 04 **Ano Letivo:** 2013 **Semestre:** 01
Professora: M. Sc. Luciana De Nardin
Data de Entrega/Apresentação: 04 e 05 de junho de 2013
Metodologia de Trabalho: em equipes de no máximo 02 (dois) alunos

Contexto

Para implementar a "aplicação" do ***Trabalho Teórico-Prático***, deve-se definir uma interface em uma linguagem imperativa (C ou Pascal, por exemplo) ou linguagem orientada a objetos (Java, por exemplo). Nesta interface, deverá ser possível cadastrar um conjunto de fatos e regras de produção e, estes dados deverão ser escritos em um arquivo cuja extensão seja ".pro".

A seguir, nesta mesma aplicação deverá ser possível cadastrar a consulta que se deseja fazer. A partir disso, uma interface de comunicação entre o Prolog e a linguagem escolhida deverá ser capaz de realizar a consulta desejada a base cadastrada, e escrever as respostas formatadas em um arquivo-texto. Por sua vez, a interface desenvolvida deverá ser capaz de realizar a leitura formatada do arquivo-texto gerado e apresentar uma visão gráfica das respostas constantes desse arquivo.

Desenvolvimento
Um exemplo de programa Prolog pode ser visto a seguir. Nesse exemplo, encontra-se definida uma base de fatos que descreve uma árvore genealógica familiar, através de construções simbólicas. Em seguida, o programa Prolog, através de regras de produção (RP), gera respostas a consultas feitas ao motor de inferência, escrevendo *strings* formatadas num arquivo em disco.

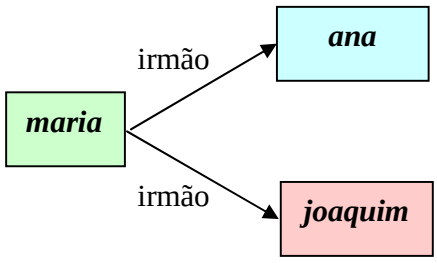
```
pai_de(joao, maria).
pai_de(joao, ana).
pai_de(joao, joaquim).
pai_de(joaquim, pedro).
pai_de(pedro, luiza).
pai_de(pedro, carla).

irmao_de(X,Y) :- pai_de(Z,X), pai_de(Z,Y), X \= Y.
```

```
cria(ARQ) :- irmao_de(maria,X),
             tell(ARQ),
             write('<PARENTE>'), nl,
             write('   <PESSOA = maria>'), nl,
             write('   <IRMAO = '), write(X), write('>'), nl,
             write('</PARENTE>'), nl,
             flush,
             tell(user).
```

Ob servando-se a base de fatos descrita, a RP ***cria(ARQ)*** produzirá a seguinte saída formatada no arquivo-texto identificado por ***ARQ***.

```
<PARENTE>
   <PESSOA = maria>
   <IRMAO = ana>
</PARENTE>
<PARENTE>
   <PESSOA = maria>
   <IRMAO = joaquim>
</PARENTE>
```

O passo final consiste da implementação de um programa em uma linguagem imperativa ou orientada a objetos, que seja capaz de interpretar as *strings* formatadas no arquivo-texto e apresentar uma visão gráfica das respostas na tela. Considerando que a semântica das respostas descreve uma relação de ***irmandade*** entre três símbolos (*maria* e *ana*; *maria* e *joaquim*), poder-se-ia pensar numa visão gráfica que apresentasse essa relação na tela. Uma sugestão pode ser vista a seguir.



Considerações finais

O programa deve ter uma aplicação prática. Dentre as aplicações que podem ser implementadas pode-se destacar um sistema de rotas urbanas ou uma árvore genealógica, por exemplo.