UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO
RELATÓRIO - PROJETO DE LÓGICA PARA COMPUTAÇÃO

Professor: Dr. SERGIO QUEIROZ

Grupo: Márcio Arruda (mba5),
Renata Amorim (rksa),
Vinícius Giles (vgcp).

1. Introdução

Nesse projeto foi desenvolvido um sistema especialista para recomendar lugares para a prática de escalada utilizando Prolog. Para isso foi preciso obedecer algumas regras, tais como, um lugar de escalada é composto por um nome, uma lista de estações do ano em que se pode escalar, o tipo de escalada (esportiva ou para diversão) e um nível de dificuldade: Fácil, Médio ou Difícil, regras para definir a forma física, entre outras que vamos discutir no decorrer do relatório.

2. Criação do programa em Prolog

%Predicados

1. escalador (Pessoa, NumMarinheiros).
Nesse predicado definimos as pessoas que estão cadastradas e o número de marinheiros que a pessoa realizou.
Pessoa: a pessoa cadastrada.
NumMarinheiros: quantidade de marinheiros.
Os escaladores são:

se.pl
```
%escalador(Pessoa,NumMarinheiros)
escalador(pedro, 12).
escalador(joao, 8).
escalador(luiza, 18).
escalador(ana, 23).
escalador(roberto, 27).
escalador(carmen, 10).
escalador(eliana, 19).
escalador(jose, 6).
escalador(sabrina, 9).
escalador(rayane, 30).
escalador(priscila, 26).
```

2. local (NomeDoLocal, [lista de estacoes], Tipo, NivelDoLocal).
Nesse predicado cadastramos os locais de escalada, onde registramos o nome, as estações que pode ser praticada a escala, o tipo e o nível.
NomeDoLocal: nome do local de escalada.
[lista de estacoes]: uma lista que contém as estações que pode ser praticada a escalada, exemplo: outono, inverno.

Tipo: tipo de escalada (diversão ou esporte).
NivelDoLocal: Nível de dificuldade de escalada no local (fácil, médio, difícil).

Os locais de escalada são:

se.pl
```
%locais de escalada
local('Serra do Cipo', [inverno, primavera], diversao, dificil).
local('Pao de Acucar', [outono, inverno, primavera, verao], esporte, dificil).
local('Sao Bento do Sapucai', [inverno, primavera], esporte, dificil).
local('Pindamonhangaba', [inverno, primavera], esporte, medio).
local('Chapada da Diamantina', [inverno, primavera], diversao, medio).
local('Lapinha', [inverno, primavera], esporte, medio).
local('Serra Caiada', [primavera, verao], diversao, facil).
local('Pedra da Boca', [primavera, verao], diversao, facil).
local('Lapa do Seu Antao', [outono, inverno, primavera, verao], esporte, facil).
```

%Regras

1. estacao (Dia, Mes, X).

Nessa regra definimos qual estação é de acordo com a data dada, utilizamos os períodos de acordo com as estações do ano atual (2016). Para isso fizemos várias checagens onde as estações são:

- Outono: de 20 de março a 20 de junho.

- Inverno: de 20 de junho a 22 de setembro.

- Primavera: de 22 de setembro a 21 de dezembro.

- Verão: de 21 de dezembro a 20 de março.

Exemplo:

estacao (Dia, Mes, verao) :- Dia > 21, Dia =<31, Mes >= 12, Mes =< 12;

Nessa regra buscamos localizar todos os locais que atendem ao determinado Tipo que a pessoa solicitou (diversao, esporte), Estacao e Nivel (facil, medio, dificil) para que possam ser retornados.

Exemplo:

locaisAdequados(Local, NivelDoLocal, Estacao, Tipo, dificil) :- local(Local, ListaEstacoes, Tipo, _), local(Local, ListaEstacoes, Tipo, NivelDoLocal), contemEstacao(Estacao, ListaEstacoes).

%onde temos nesse exemplo a variável anônima, pois quem é do nível difícil pode praticar todos os níveis.

2. formaFisica (E, X).

Nessa regra definimos a forma física do escalador, chamando o predicado escalador (Pessoa, NumMarinheiros) e de acordo com a quantidade de marinheiros realizados atribuímos a forma física. Caso tenha feito menos de 10 marinheiros é forma física fraca, se fez pelo menos 10 marinheiros até 24 sua forma física é média e se fez a partir de 25 é atlética.

Exemplo:

formaFisica(E, atletica) :- escalador(E, M), M >= 25.

Nessa regra também tivemos que cuidar da exceção da Pessoa do escalador não está cadastrado, então assumimos que se não está cadastrado a Pessoa tem forma física fraca.

3. nivelDificuldade(Pessoa, HorasPratica, X).

Nessa regra veremos o nível de dificuldade que a Pessoa pode praticar a partir de sua forma e da quantidade de horas que tem de prática. Para o nível fraco qualquer Pessoa pode praticar, porém para o médio precisa ter pelo menos 16 horas de prática e ter forma física média, para o nível difícil precisar ter pelo menos 40 horas de prática e ter forma física atlética. Logo, observamos que mesmo que a pessoa tenha uma forma física boa, mas a quantidade de horas seja ruim, irá limitar os locais de prática, o inverso também acontece.

Exemplo:

nivelDificuldade(Pessoa, HorasPratica, facil) :-

formaFisica(Pessoa, fraca), HorasPratica >= 16;

HorasPratica < 16.

4. recomendarEscalada(Pessoa, HorasPratica, Tipo, DiaEntrada, MesEntrada, ListaRecomendada).

Chegamos então na nossa regra principal, nela vamos ter o retorno esperado. O retorno dessa regra é uma lista de todos os locais que com aquela entrada são possíveis de praticar a escalada em forma de duplas (Local, NivelDoLocal).

A regra:

se.pl [modified]

```
%principal
recomendarEscalada(Pessoa, HorasPratica, Tipo, DiaEntrada, MesEntrada, ListaRecomendada) :-
        %calcula a forma fisica da pessoa
        formaFisica(Pessoa, FormaFisica),

        %calcula o nivel de acordo com horas de pratica. saida: Nivel = facil || medio || dificil
        nivelDificuldade(HorasPratica, FormaFisica, NivelDoEscalador),

        % calcula a Estacao de acordo com Dia e Mes (saida: outono || inverno || primavera || verao)
        estacao(DiaEntrada, MesEntrada, Estacao),

        % pega os locais do tipo e nivel do escalador. saida: Locais
        findall((Local,NivelDoLocal), locaisAdequados(Local, NivelDoLocal, Estacao, Tipo, NivelDoEscalador), ListaRecomendada).
```

3. Testes

SWI-Prolog (Multi-threaded, version 7.2.0)

File Edit Settings Run Debug Help

```
1 ?- [se].
true.

2 ?- recomendarEscalada(eliana, 18, diversao, 13, 2, ListaRecomendada).
ListaRecomendada = [ ('Serra Caiada', facil), ('Pedra da Boca', facil)] ;
ListaRecomendada = [ ('Serra Caiada', facil), ('Pedra da Boca', facil)] ;
false.
```

Teste com escalador já cadastrado (Eliana, que tem forma física média), 18 horas de prática, tipo diversão, no dia 13 do mês 2 (verão), com nível de dificuldade média.

SWI-Prolog (Multi-threaded, version 7.2.0)

File Edit Settings Run Debug Help

```
3 ?- recomendarEscalada(roberto, 13, esporte, 29, 5, ListaRecomendada).
ListaRecomendada = [ ('Lapa do Seu Antao', facil)] ;
false.

4 ?-
```

Teste com escalador já cadastrado (Roberto, forma física atlética), com 13 horas de prática, tipo esporte, no dia 29 do mês 5 (outono), com nível de dificuldade fácil.

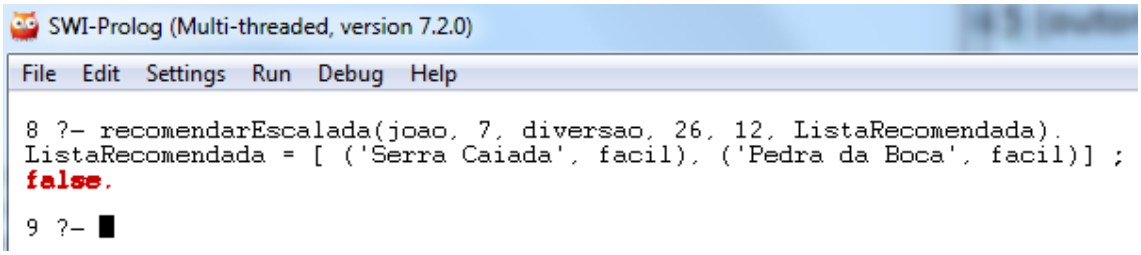
SWI-Prolog (Multi-threaded, version 7.2.0)

File Edit Settings Run Debug Help

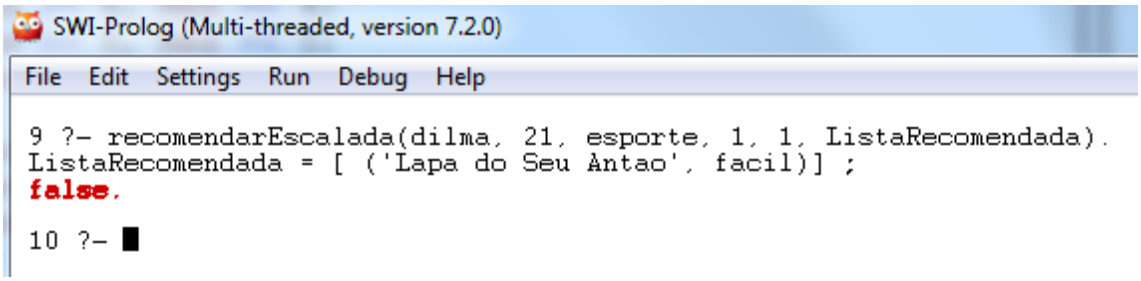
```
7 ?- recomendarEscalada(priscila, 43, esporte, 12, 8, ListaRecomendada).
ListaRecomendada = [ ('Pao de Acucar', dificil), ('Sao Bento do Sapucai',
 dificil), ('Pindamonhangaba', medio), ('Lapinha', medio), ('Lapa do Seu
Antao', facil)] ;
false.
```


Av. Jornalista Anibal Fernandes, s/n | Cidade Universitária | CEP: 50740-560 | Recife - PE | Brasil | Telefone: + 55 81 2126-8430 | Fax: + 55 81 2126-8438

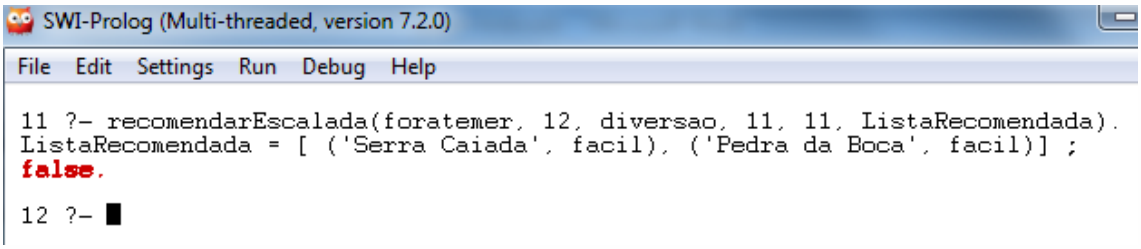
Teste com escalador já cadastrado (Priscila, forma física atlética), com 43 horas de prática, tipo esporte, no dia 12 do mês 8 (inverno), com nível de dificuldade difícil.



Teste com escalador já cadastrado (João, forma física fraca), com 7 horas de prática, tipo diversão, no dia 26 do mês 12 (verão), com nível de dificuldade fácil.

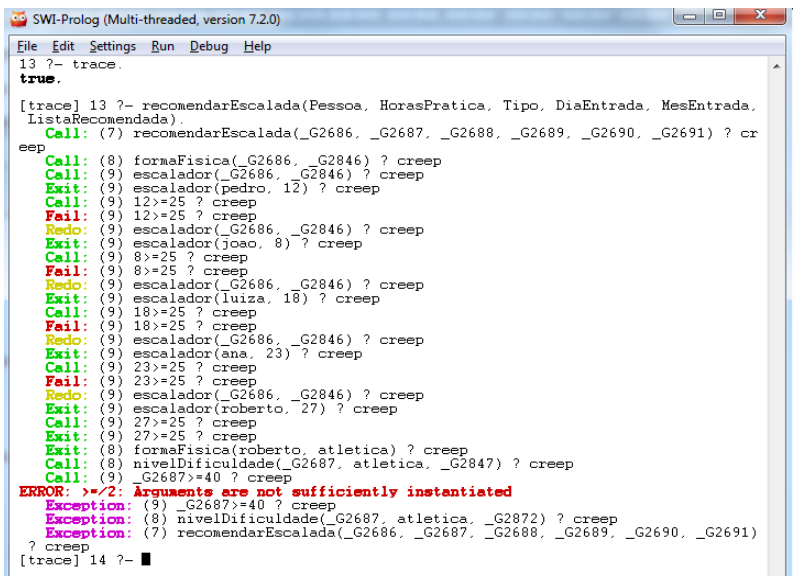


Teste com escalador não cadastrado (Dilma, forma física fraca), com 21 horas de prática, tipo esporte, no dia 1 do mês 1 (verão), com nível de dificuldade fácil.

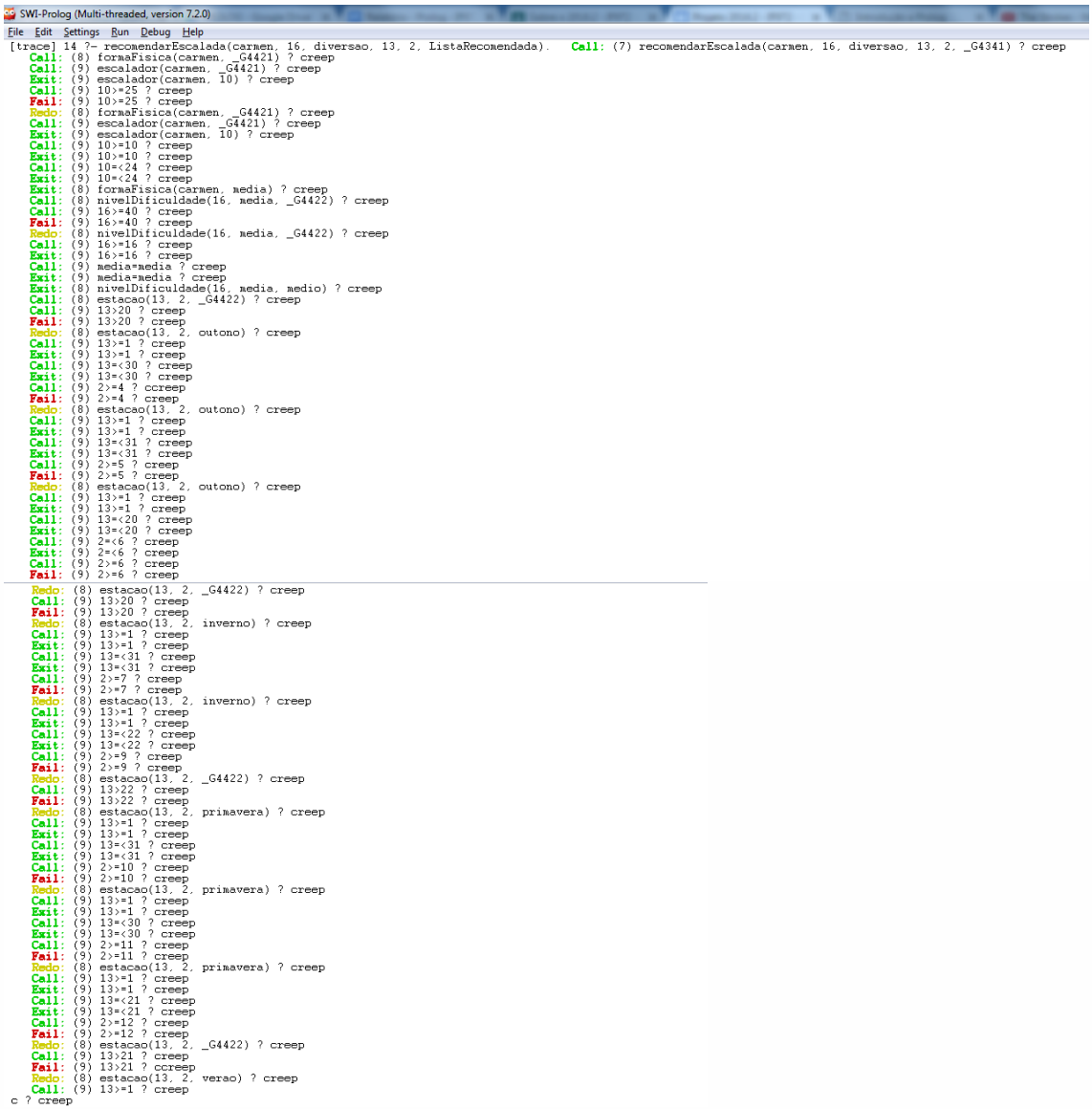


Teste com escalador golpista não cadastrado (Foratemer, forma física fraca), com 12 horas de prática, tipo diversão, no dia 11 do mês 11 (primavera), com nível de dificuldade fácil.

4. Trace



Nesse trace a call começa checando os escaladores,começa perguntando se o NumMarinheiros é maior ou igual a 25, ele vai falhando e seguindo para o próximo, até achar um caso que seja verdade, ele acaba achando no escalador Roberto que tem o NumMarinheiros de 27, em seguida ele da um exit com a resposta da formaFisica de Roberto que é atlética, porém quando vamos para a próxima call com nivelDificuldade, HorasPraticas não está definida e ele não consegue instanciar o argumento, gerando o Erro e várias exceções.



[...]

```
[...]
   Call: (9) 13=<31 ? creep
   Exit: (9) 13=<31 ? creep
   Call: (9) 2>=1 ? creep
   Exit: (9) 2>=1 ? creep
   Call: (9) 2=<1 ? creep
   Fail: (9) 2=<1 ? creep
   Redo: (8) estacao(13, 2, verao) ? creep
   Call: (9) 13>=1 ? creep
   Exit: (9) 13>=1 ? creep
   Call: (9) 13=<29 ? creep
   Exit: (9) 13=<29 ? creep
   Call: (9) 2>=2 ? creep
   Exit: (9) 2>=2 ? creep
   Call: (9) 2=<2 ? creep
   Exit: (9) 2=<2 ? creep
   Exit: (8) estacao(13, 2, verao) ? creep
^  Call: (8) findall((_G4413, _G4414), locaisAdequados(_G4413, _G4414, verao, diversao, medio), _G4341) ? creep
   Call: (13) locaisAdequados(_G4413, _G4414, verao, diversao, medio) ? creep
   Call: (14) local(_G4413, _G4447, diversao, medio) ? creep
   Exit: (14) local('Chapada da Diamantina', [inverno, primavera], diversao, medio) ? creep
   Call: (14) local('Chapada da Diamantina', [inverno, primavera], diversao, _G4414) ? creep
   Exit: (14) local('Chapada da Diamantina', [inverno, primavera], diversao, medio) ? creep
   Call: (14) contemEstacao(verao, [inverno, primavera]) ? creep
   Call: (15) contemEstacao(verao, [primavera]) ? creep
   Call: (16) contemEstacao(verao, []) ? creep
   Fail: (16) contemEstacao(verao, []) ? creep
   Fail: (15) contemEstacao(verao, [primavera]) ? creep
   Fail: (14) contemEstacao(verao, [inverno, primavera]) ? creep
   Redo: (14) local('Chapada da Diamantina', [inverno, primavera], diversao, _G4414) ? creep
   Fail: (14) local('Chapada da Diamantina', [inverno, primavera], diversao, _G4414) ? creep
   Redo: (14) local(_G4413, _G4447, diversao, medio) ? creep
   Fail: (14) local(_G4413, _G4447, diversao, medio) ? creep
   Redo: (13) locaisAdequados(_G4413, _G4414, verao, diversao, medio) ? creep
   Call: (14) local(_G4413, _G4447, diversao, facil) ? creep
   Exit: (14) local('Serra Caiada', [primavera, verao], diversao, facil) ? creep
   Call: (14) local('Serra Caiada', [primavera, verao], diversao, _G4414) ? creep
   Exit: (14) local('Serra Caiada', [primavera, verao], diversao, facil) ? creep
   Call: (14) contemEstacao(verao, [primavera, verao]) ? creep
   Call: (15) contemEstacao(verao, [verao]) ? creep
   Exit: (15) contemEstacao(verao, [verao]) ? creep
   Exit: (14) contemEstacao(verao, [primavera, verao]) ? creep
   Exit: (13) locaisAdequados('Serra Caiada', facil, verao, diversao, medio) ? creep
   Redo: (15) contemEstacao(verao, [verao]) ? creep
   Call: (16) contemEstacao(verao, []) ? creep
   Fail: (16) contemEstacao(verao, []) ? creep
   Fail: (15) contemEstacao(verao, [verao]) ? creep
   Fail: (14) contemEstacao(verao, [primavera, verao]) ? creep
   Redo: (14) local('Serra Caiada', [primavera, verao], diversao, _G4414) ? creep
   Fail: (14) local('Serra Caiada', [primavera, verao], diversao, _G4414) ? creep
   Redo: (14) local(_G4413, _G4447, diversao, facil) ? creep
   Exit: (14) local('Pedra da Boca', [primavera, verao], diversao, facil) ? creep
   Call: (14) local('Pedra da Boca', [primavera, verao], diversao, _G4414) ? creep
   Exit: (14) local('Pedra da Boca', [primavera, verao], diversao, facil) ? creep
   Call: (14) contemEstacao(verao, [primavera, verao]) ? creep
   Call: (15) contemEstacao(verao, [verao]) ? creep
   Exit: (15) contemEstacao(verao, [verao]) ? creep
   Exit: (14) contemEstacao(verao, [primavera, verao]) ? creep
   Exit: (13) locaisAdequados('Pedra da Boca', facil, verao, diversao, medio) ? creep
   Redo: (15) contemEstacao(verao, [verao]) ? creep
   Call: (16) contemEstacao(verao, []) ? creep
   Fail: (16) contemEstacao(verao, []) ? creep
   Fail: (15) contemEstacao(verao, [verao]) ? creep
   Fail: (14) contemEstacao(verao, [primavera, verao]) ? creep
   Redo: (14) local(_G4413, _G4447, diversao, facil) ? creep
   Fail: (14) local(_G4413, _G4447, diversao, facil) ? creep
   Redo: (13) locaisAdequados(_G4413, _G4414, verao, diversao, medio) ? creep
   Fail: (13) locaisAdequados(_G4413, _G4414, verao, diversao, medio) ? creep
^  Exit: (8) findall((_G4413, _G4414), user:locaisAdequados(_G4413, _G4414, verao, diversao, medio), [ ('Serra Caiada', facil), ('Pedra da Boca', facil)]) ? creep
   Exit: (7) recomendarEscalada(carmen, 16, diversao, 13, 2, [ ('Serra Caiada', facil), ('Pedra da Boca', facil)]) ? creep
ListaRecomendada = [ ('Serra Caiada', facil), ('Pedra da Boca', facil)] .

[trace] 15 ?- 
```

Verificou-se que o programa busca definir Carmem a partir das regras escalador e formaFisica. Por exemplo, a variável X que indica a forma física é inicialmente definida como **_G4421**, e após algumas chamadas, é identificada uma forma física média. Logo após, a regra nivelDificuldade define o nível da escaladora como médio. E assim seguiu-se com as demais cláusulas.

Analisou-se também que em alguns pontos onde ocorre FAIL é devido ao que foi apresentado não ser verdade. Depois deu continuidade a próxima iteração onde ocorreu o REDO em que foi feito o backtracking.

5. Dificuldades e limitações

Uma das maiores dificuldades na aprendizagem da linguagem Prolog é se acostumar com o paradigma de programação lógica, que é bem diferente do usual como o paradigma imperativo. Outras dificuldades foram tidas com o mecanismo de trace, que muitas vezes informou erros no código que tiveram que ser solucionados, também com a ajuda dos monitores da disciplina. O prazo para concluir o projeto também foi muito difícil de cumprir, devido ao pouco tempo que tivemos para realiza-lo.

6. Referências:

1. Lógica para Ciência da Computação, João Nunes de Souza, Editora Campus, 2002.
2. http://www.suapesquisa.com/geografia/estacoesdoano.htm
3. http://www.inf.puc-rio.br/~inf1621/logica.pdf
4. Página do SWI-Prolog: http://www.swi-prolog.org/
5. http://www.learnprolognow.org/
6. http://www.linhadecodigo.com.br/artigo/1697/descobrindo-o-prolog.aspx