

# Roteiro III

Alexsandro Santos Soares  
prof.asoares@gmail.com

Programação Lógica  
Faculdade de Computação  
Universidade Federal de Uberlândia

11 de dezembro de 2021

## 1 Objetivos

Este roteiro tem por objetivos:

- Introduzir alguns comandos úteis do intérprete SWI-Prolog;
- Praticar a ideia de recursão.

## 2 Alguns comandos úteis

Os comandos seguintes podem ser digitados na linha de comando do SWI-Prolog.

```
?- listing.
```

Exibe uma listagem do conteúdo corrente da base de dados do Prolog (fatos e predicados).

```
?- make.
```

Procura alterações nas datas dos arquivos consultados pelo Prolog e os reconsulta se ocorrer alguma alteração.

```
?- help(Topico).
```

Abre o manual no tópico especificado, ex. `help(member)`.

```
?- apropos(Palavra).
```

Abre o manual e faz uma busca usando a palavra digitada, ex. `apropos(append)`.

```
?- halt.
```

Fecha o SWI-Prolog, sem qualquer pedido de confirmação.

### 3 Uso de regras

**Ex. 1** Considere a imagem a seguir descreve uma situação onde os objetos estão uns sobre os outros:



- (a) Instrua o interpretador de Prolog sobre a situação mostrada na imagem, usando para isto o predicado `sobre/2`.
- (b) Expresse por meio de fórmulas lógicas (e em seguida na notação do Prolog) a seguinte regra de conhecimento:

Se um objeto está sobre outro,  
então este objeto está acima do outro.

Se um objeto A está sobre B e B está acima de C,  
então A está acima de C.

Note que o predicado `acima_de` é transitivo.

- (c) Coloque a seguinte questão ao interpretador Prolog: “O cupcake está acima da mesa?”.
- (d) Coloque a seguinte questão ao interpretador Prolog: “Os lápis estão acima da mesa?”.
- (e) Coloque a seguinte questão ao interpretador Prolog: “Quais objetos estão acima do novelo?”.
- (f) Coloque a seguinte questão ao interpretador Prolog: “Quais objetos estão acima da caneca?”.
- (g) Usando diretamente o predicado `acima_de` escreva um outro predicado `abaixo_de` que descreva transitivamente os objetos que estão abaixo de outros.
- (h) Coloque a seguinte questão ao interpretador Prolog: “Quais objetos estão abaixo do cupcake?”
- (i) Coloque a seguinte questão ao interpretador Prolog: “Quais objetos estão abaixo dos lápis?”

## 4 Recursão

**Ex. 2** A hierarquia militar do Exército brasileiro compreende as patentes a seguir, mostradas do mais baixo para o mais alto posto:

- Soldado
- Taifeiro
- Cabo
- Terceiro-sargento
- Segundo-sargento
- Primeiro-sargento
- Subtenente
- Aspirante
- Segundo-tenente
- Primeiro-tenente
- Capitão
- Major
- Tenente-coronel
- Coronel
- General-de-brigada
- General-de-divisão

- General-de-exército
  - Marechal
- (a) Escreva um conjunto de fatos `próximo_posto` que é verdadeiro quando o primeiro posto é imediatamente mais baixo que o segundo. Por exemplo, o posto de capitão é imediatamente mais baixo que o de major.

A revista em quadrinhos *Recruta Zero* criada pelo cartunista norte-americano Mort Walker foi muito popular no Brasil até por volta de 2010.



Figura 1: Da esquerda para a direita: Roque, Otto, Sargento Tainha, Recruta Zero, Quindim, Platão, Tenente Escovinha, Tenente Mironga, Cuca, O Capelão, Capitão, Dona Tetê, General Dureza e Dentinho.

Nesta revista os personagens militares possuem os seguintes postos:

- Soldado Zero
- Soldado Platão
- Soldado Dentinho
- Soldado Cosme
- Soldado Roque
- Soldado Quindim
- Soldado Blips
- Cabo Ky
- Sargento Cuca
- Sargento Tainha
- Sargento Louise Lorota
- Tenente Escovinha
- Tenente Mironga

- Capitão Durindana
  - Major Peroba
  - General de exército Amos Dureza
- (b) Escreva um conjunto de fatos `militar/2` que retrate os personagens militares da revista *Recruta Zero*. O primeiro argumento é o nome do personagem e o segundo é o seu posto.
- (c) Usando o fato `próximo_posto` escreva um predicado `menor_patente/2` que é verdadeiro se a patente no primeiro argumento é menor do que aquela no segundo argumento. Note que `menor_patente` é transitivo.
- (d) Use o predicado `menor_patente` para escrever o predicado `maior_patente/2` que é verdadeiro se a patente no primeiro argumento é *maior* do que aquela no segundo argumento.
- (e) Escreva o predicado `subordinado/2` que recebe dois nomes de militares e é verdadeiro sempre que a patente associada ao primeiro nome é inferior àquela associada ao segundo nome.
- (f) Coloque a seguinte questão ao interpretador Prolog: “O recruta Zero está subordinado ao sargento Tainha?”
- (g) Coloque a seguinte questão ao interpretador Prolog: “Quais são os nomes dos militares que são superiores ao tenente Mironga?”

**Ex. 3** Imagine que a base de conhecimentos seguinte descreva um labirinto. Os fatos determinam quais pontos estão conectados, ou seja, os pontos que se pode alcançar com exatamente um passo. Além disto, imagine que todos os caminhos são ruas de mão única, tal que você somente pode caminhar por elas em uma direção. Assim, você poderá ir do ponto 1 para o ponto 2, mas não poderá ir de 2 para 1.

```
conectado(1,2).
conectado(3,4).
conectado(5,6).
conectado(7,8).
conectado(9,10).
conectado(12,13).
conectado(13,14).
conectado(15,16).
conectado(17,18).
conectado(19,20).
conectado(4,1).
conectado(6,3).
conectado(4,7).
conectado(6,11).
conectado(14,9).
```

```
conectado(11,15).
conectado(16,12).
conectado(14,17).
conectado(16,19).
```

Escreva um predicado `caminho/2` que diga até onde se pode chegar partindo de um determinado ponto, seguindo as conexões na base de conhecimento. Pode-se ir do ponto 5 para o ponto 10? Em quais outros pontos se pode chegar partindo do ponto 1? E quais pontos podem ser alcançados saindo do ponto 13?

- Ex. 4** Imagine que você resolva passear mundo afora e possua a seguinte base de conhecimento sobre opções de transporte entre cidades:

```
deCarro(auckland,hamilton).
deCarro(hamilton,raglan).
deCarro(valmont,saarbruecken).
deCarro(valmont,metz).

deTrem(metz,frankfurt).
deTrem(saarbruecken,frankfurt).
deTrem(metz,paris).
deTrem(saarbruecken,paris).

deAviao(frankfurt,bangkok).
deAviao(frankfurt,singapore).
deAviao(paris,losAngeles).
deAviao(bangkok,auckland).
deAviao(losAngeles,auckland).
```

Escreva um predicado `viagem/2` que determine se é possível viajar de um lugar a outro usando qualquer meio de transporte disponível: carro, trem e avião. Por exemplo, seu programa deveria responder `true` para a consulta `viagem(valmont,raglan)`.

- Ex. 5** Usando o predicado `viagem/2` para consultar a base de dados anterior pode-se deduzir que é possível ir de Valmont para Raglan. No caso de estar planejando uma viagem, isto é uma boa informação, mas o que você realmente deseja saber é *como* exatamente ir de Valmont a Raglan. Escreva um predicado `viagem/3` que diga a você como viajar de um lugar para o outro. Exemplos de consultas:

```
?- viagem(valmont,paris,vai(valmont,metz,vai(metz,paris))).
true

?- viagem(valmont,losAngeles,X).
X = vai_de(valmont,metz,vai_de(metz,paris,vai_de(paris,losAngeles)))
```

- Ex. 6** Estenda o predicado `viagem/3` tal que ele não somente diga a você em quais cidades intermediárias você passará, mas também *como* ir de

uma cidade a outra, ou seja, de carro, trem ou avião.

## 5 Sugestões de leitura

- Luiz A. M. Palazzo. *Introdução à programação Prolog*  
<http://puig.pro.br/Logica/palazzo.pdf>
- Eloi L. Favero. *Programação em Prolog: uma abordagem prática*  
<http://www3.ufpa.br/favero>
- Wikilivro sobre Prolog em  
<http://pt.wikibooks.org/wiki/Prolog>
- Patrick Blackburn, Johan Bos and Kristina Striegnitz. *Learn Prolog Now!*  
<http://www.learnprolognow.org>