



# PROLOG

Rafael D. Ribeiro, M.Sc.  
[rafaeldiasribeiro@gmail.com](mailto:rafaeldiasribeiro@gmail.com)  
<http://www.rafaeldiasribeiro.com.br>

- Leitura e escrita de termos:
  - Prolog oferece um predicado pré- definido `read` para a entrada de dados.
  - A meta `read(X)` é satisfeita quando `X` unifica com o próximo termo lido no dispositivo de entrada.
  - É preciso colocar um ponto final para sinalizar o fim do termo.



# Strawberry PROLOG

path

```
pequeno:- read(N),N<30,write(N).
```

```
?-pequeno.
```

Write a term

Write here:

OK

Cancel

Output

Saving.

Compiling the f:  
C:\Users\RDRIBE:  
0 errors, 0 warn

25Yes.

# Strawberry PROLOG

```
path  
  
soma:- read(A),read(B), C is A+B,write(C).  
?-soma.
```

Write a term

Write here:

OK Cancel

Write a term

Write here:

OK Cancel

Saving.

```
Compiling the file  
C:\Users\RDRIBEIR  
0 errors, 0 warni
```

50Yes.



# Strawberry PROLOG

```
path  
soma:- read(A),read(B), C is A+B,write(C).  
?-soma.
```

Write a term

Write here:

OK Cancel

Write a term

Write here:

OK Cancel

Saving.

Compiling the file  
C:\Users\RDRIBEIR  
0 errors, 0 warni

50Yes.



# Strawberry PROLOG

## OPERADOR OU (;)

$\text{happy3}(X) \text{ :- attractive}(X), (\text{rich}(X) ; \text{famous}(X)) .$

This says  $X$  is  $\text{happy3}$  if  $X$  is attractive, and  $X$  is either rich or famous:

$\text{happy3}(X) \Leftarrow \text{attractive}(X) \wedge (\text{rich}(X) \vee \text{famous}(X)) .$



## Strawberry PROLOG

✓ path

```
consulta(X):- 1 is X mod 2; X<3.
```

```
?-consulta(1), write("atende").|
```

```
Saving.
```

```
Compiling the  
C:\Users\RDRI  
0 errors, 0 w
```

```
atendeYes.
```



## OPERADORES INFIXOS (I)

E se eu quisessemos realizar consultas utilizando uma linguagem mais próxima da nossa?

? - Quem tem carro.

? joana faz Oque?



## OPERADORES INFIXOS (II)

O programador Prolog pode definir seus próprios operadores, como no exemplo: **tem** e **faz**.

A definição de novos operadores é realizada pela inserção de um tipo especial de cláusula chamada **diretiva**.

As diretivas devem aparecer antes de qualquer expressão que contenha o operador criado.



## DIRETIVAS (I)

Exemplo de diretiva:

`:- op (500, xfx, tem).`

Onde:

**500** indica a prioridade do operador;

**xfx** indica que o operador (f) deve ser colocado entre dois argumentos (x);

**tem** indica o nome do operador.



## DIRETIVAS (II)

Tendo a diretiva, criamos a base de conhecimento:

pedro tem carro.

joana tem dinheiro.

joao tem problemas.

joao tem dívidas.



E fizemos as consultas:

?- Quem tem carro.

Quem = pedro.



## DIRETIVAS (III)

**Curiosidade:** os “comandos” que usamos no Prolog, são diretivas pré-definidas na sua implementação:

```
:- op (1200, xfx, ‘:-’);  
:- op (1200, fx [‘:-’, ‘?-’]).  
:- op (1100, xfx, ‘;’)  
:- op (1000, xfx, ‘,’) /* etc...*/
```

Note que alguns operadores (ex.: ‘:-’), possuem definição infixa (xfx) e prefixa (fx).



## PRÁTICA 7

1. Crie duas diretivas novas com operador infix, e uma base de conhecimento usando estas diretivas.
2. Em seguida crie consultas que se assemelhem a linguagem humana.  
Exemplos:

? - fulado conhece Quem.

? - ciclano mora Onde.



# OPERADORES DE COMPARAÇÃO

Lista completa dos operadores de comparação:

OPERADOR	PRIORIDADE	TIPO	SIGNIFICADO
>	700	xfx	maior que
<	700	xfx	menor que
>=	700	xfx	maior ou igual a
<=	700	xfx	menor ou igual a
== ou ===	700	xfx	valores iguais
!= ou \==	700	xfx	valores diferentes

# BASE DE DADOS RELACIONAL (I)

Como criar uma base de dados relacional em Prolog?

Como pesquisar resultados dentro dessa base?

nasceu(pedroBarnack,joinville,1978).

nasceu(muriloPereira,itajai,1980).

nasceu(rafaelRosario,joinville,1980).

nasceu(janineBoos,joinville,1985).

nasceu(douglasSouza,curitiba, 1970).



## BASE DE DADOS RELACIONAL (II)

Como perguntamos:

Nome e ano de nascimento dos curitibanos?

?- nasceu(Quem,curitiba,Ano).

Quem nasceu em joinville a partir de 1980?

?- nasceu(Quem,joinville,Ano),Ano >= 1980.

Nome e ano de nascimento de quem nasceu antes de 1980 (sem trazer a cidade)?

?- nasceu(Quem,\_,Ano), Ano < 1980.



## PRÁTICA 8

Crie um banco de dados que contenhas as seguintes informações:

- Nome do Professor e disciplina que leciona;
- Disciplina e horário (dia da semana, aula: primeiro ou segundo horário);

Construa as consultas para responder:

1. Qual o horário do professor X (disciplina, dia e aula)?
2. Quais professores lecionam na terça-feira?
3. Quais matérias o professor X leciona no primeiro horário?



## PRÁTICA 8 - RESPOSTA

1. `horario_prof(P,D,S,A) :- leciona(P,D), horario(D,S,A).`  
? - `horario_prof(rafael,Disc,DiaSemana,Aula) .`
2. ?- `lecciona(X,Disciplina) , horario(Disciplina,3,Aula).`
3. ?- `lecciona(eduardo,Disc), horario(Disc,_,aula1).*/`

### **Base de dados:**

`lecciona(rafael,ia).`

`lecciona(eduardo,redes).`

`horario(ia,2,aula1).`

`horario(ia,5,aula1).`

`horario(redes,3,aula1).`

`horario(redes,4,aula2).`



## PRÁTICA 8 – RESPOSTA 2 (I)

:- op(500, xfx, **leciona**).

:- op(450, xfx, **eh\_leccionada\_na**).

:- op(400, xfx, **no\_horario**).

:- op(300, fx, **qual\_o\_horario\_do\_professor**)

**qual\_o\_horario\_do\_professor**(Prof, Materia,  
DiaSemana, SeqAula) :-

Prof **leciona** Materia, Materia **eh\_leccionada\_na**  
DiaSemana **no\_horario** SeqAula.



## PRÁTICA 8 – RESPOSTA 2 (II)

### **Base de dados:**

rafael **leciona** ia.

eduardo **leciona** redes.

ia **eh\_leccionada\_na** segunda\_feira **no\_horario**  
primeira\_aula.

ia **eh\_leccionada\_na** quinta\_feira **no\_horario**  
primeira\_aula.

redes **eh\_leccionada\_na** terca\_feira **no\_horario**  
primeira\_aula.

redes **eh\_leccionada\_na** quarta\_feira **no\_horario**  
segunda\_aula.



## PRÁTICA 8 – RESPOSTA 2 (III)

1. ? - qual\_o\_horario\_do\_professor(eduardo, Materia, DiaSemana, SeqAula).
2. ?- Professor leciona Materia, Materia eh\_leccionada\_na terca\_feira no\_horario Qualquer.
3. ?- Professor leciona Materia, Materia eh\_leccionada\_na DiaSemana no\_horario primeira\_aula. /\*ou \_ no lugar de DiaSemana\*/

