

# GSI010 - Programação Lógica

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7
```

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

Nomes:

- fato: `progenitor(maria, josé).`

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: `progenitor(maria, josé).`
- ▶ regra: `filho(X, Y) :- progenitor(Y, X).`

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
  - ▶ cabeça (da regra): **filho(X, Y)**

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
  - ▶ cabeça (da regra): **filho(X, Y)**
  - ▶ corpo (da regra): **progenitor(Y, X)**

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
  - ▶ cabeça (da regra): **filho(X, Y)**
  - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
  - ▶ cabeça (da regra): **filho(X, Y)**
  - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)
- ▶ 2 predicados: progenitor/2, filho/2



## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
  - ▶ cabeça (da regra): **filho(X, Y)**
  - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)
- ▶ 2 predicados: progenitor/2, filho/2
  - ▶ marcado por ponto final

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
  - ▶ cabeça (da regra): **filho(X, Y)**
  - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)
- ▶ 2 predicados: progenitor/2, filho/2
  - ▶ marcado por ponto final
- ▶ functor: **filho(X,Y)**

## Revisão aula 2

```
1 progenitor(maria , josé).  
2 progenitor(joão , josé).  
3 progenitor(joão , ana).  
4 progenitor(josé , júlia).  
5 progenitor(josé , íris).  
6 progenitor(íris , jorge).  
7  
8 filho(X,Y) :- progenitor(Y,X).
```

### Nomes:

- ▶ fato: **progenitor(maria, josé).**
- ▶ regra: **filho(X, Y) :- progenitor(Y, X).**
  - ▶ cabeça (da regra): **filho(X, Y)**
  - ▶ corpo (da regra): **progenitor(Y, X)**
- ▶ cláusula: um fato ou uma regra (são 7 nessa base)
- ▶ 2 predicados: progenitor/2, filho/2
  - ▶ marcado por ponto final
- ▶ functor: **filho(X, Y)**

# Aula de hoje: recursão

- ▶ técnica de programação
- ▶ divisão de um problema grande em pedaços menores
- ▶ algumas linguagens são puramente recursivas: Prolog e Lisp

# Recursão

Problemas:

- ▶ fatorial:  $n! = n * (n - 1)!$ ,  $0! = 1$
- ▶ fibonacci:  $f_n = f_{n-1} + f_{n-2}$ ,  $f_1 = 1$ ,  $f_2 = 2$
- ▶ estruturas de dados em árvores
- ▶ grafos, redes sociais

# Recursão

Estratégia de resolução:

1. Definir caso base
  - ▶ fatorial:  $0! = 1$
  - ▶ fibonacci:  $f_1 = 1, f_2 = 2$
  - ▶ em árvores: nó é folha
2. Definir operação para subproblemas
3. Combinar resultado de subproblema com parte maior

Função recursiva: função que faz chamada a ela mesma

# Definições recursivas

- ▶ Predicados em Prolog podem ser definidos recursivamente
- ▶ Um predicado é definido recursivamente se uma ou mais regras em sua definição refere-se a ela mesma

## Exemplo: versão não recursiva

```
1 filho(ana , brigitte) .  
2 filho( brigitte , caroline) .  
3 filho( caroline , donna) .  
4 filho( donna , emilia) .  
5
```



## Exemplo: versão não recursiva

```
1 filho(ana , brigitte) .  
2 filho( brigitte , caroline) .  
3 filho( caroline , donna) .  
4 filho( donna , emilia) .  
5  
6 descende(X,Y):—
```

## Exemplo: versão não recursiva

```
1 filho(ana , brigitte) .  
2 filho( brigitte , caroline) .  
3 filho( caroline , donna) .  
4 filho( donna , emilia) .  
5  
6 descende(X,Y):- filho(X,Y) .  
7 descende(X,Y):- filho(X,Z) , filho(Z,Y) .
```

## Exemplo: versão não recursiva

```
1 filho(ana , brigitte) .  
2 filho( brigitte , caroline) .  
3 filho( caroline , donna) .  
4 filho( donna , emilia) .  
5  
6 descende(X,Y):- filho(X,Y) .  
7 descende(X,Y):- filho(X,Z) , filho(Z,Y) .
```

Algumas respostas não são corretas:

```
1 ?- descende(ana , donna) .  
2 false .
```

## Exemplo: versão recursiva

```
1 filho(ana,brigitte).  
2 filho(brigitte,caroline).  
3 filho(caroline,donna).  
4 filho(donna,emilia).  
5
```

## Exemplo: versão recursiva

```
1 filho(ana,brigitte).  
2 filho(brigitte,caroline).  
3 filho(caroline,donna).  
4 filho(donna,emilia).  
5  
6 descende(X, Y):- filho(X,Y).  
7 descende(X, Y):- filho(Z, Y), descende(X, Z).
```

## Exemplo: versão recursiva

```
1 filho(ana,brigitte).  
2 filho(brigitte,caroline).  
3 filho(caroline,donna).  
4 filho(donna,emilia).  
5  
6 descende(X, Y):- filho(X,Y).  
7 descende(X, Y):- filho(Z, Y), descende(X, Z).
```

```
1 ?- descende(ana, donna).  
2 true ;  
3 false.
```

# Árvore de busca

- Desenhe a árvore de busca para

```
1 ?— descende(ana, donna).
```

# Pesquisa na base de conhecimento

O Prolog:

- ▶ pesquisa base de conhecimento de cima para baixo
- ▶ processa cláusulas da esquerda para a direita
- ▶ retroce para se recuperar de escolhas ruins

Diferenças entre:

```
1 descende(X,Y):- filho(X,Y).  
2 descende(X,Y):- filho(X,Z), descende(Z,Y).
```

```
1 descende(X,Y):- filho(X,Z), descende(Z,Y).  
2 descende(X,Y):- filho(X,Y).
```

```
1 descende(X,Y):- descende(Z,Y), filho(X,Z).  
2 descende(X,Y):- filho(X,Y).
```



# Exemplo: fatorial

## Naturalmente recursivo

$$\begin{aligned}\text{fatorial}(5) &= 5 * \text{fatorial}(4) \\ &= 5 * (4 * \text{fatorial}(3)) \\ &= 5 * (4 * (3 * \text{fatorial}(2))) \\ &= 5 * (4 * (3 * (2 * \text{fatorial}(1))))\end{aligned}$$

Em Prolog?

# Referências

- ▶ Luis, A. M. Palazzo, Introdução à programação prolog, Educat, 1997
- ▶ Slides profs. Elaine Faria, Hiran Nonato e Gabriel Coutinho - UFU
- ▶ Slides da Profa. Solange - ICMC - USP