Linguagem Prolog

Prof. Dr. Silvio do Lago Pereira

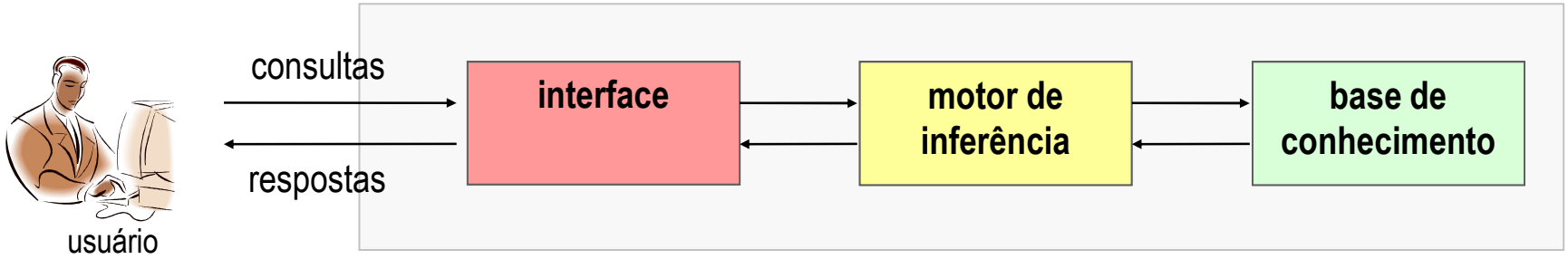
Departamento de Tecnologia da Informação

Faculdade de Tecnologia de São Paulo

Introdução

Prolog (*Programming in Logic*)

é uma linguagem de programação declarativa para processamento simbólico.



Essencialmente, a programação em Prolog consiste em:

- identificar os **objetos** em um contexto de discurso
- identificar **relações** (ou predicados) de interesse entre estes objetos
- declarar **fatos** e **regras** a respeito destas relações
- **consultar** o sistema acerca das relações declaradas

Elementos básicos

fato
regra
consulta

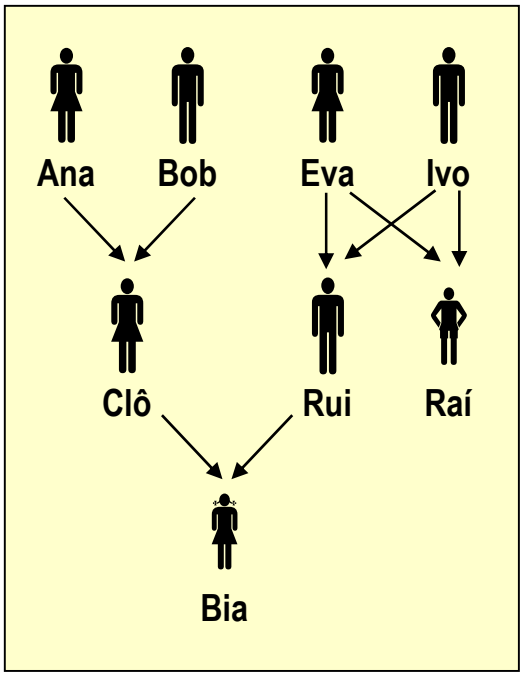
Fato

Um fato

estabelece um relacionamento **incondicional** entre objetos de um contexto.

Alguns fatos no contexto ao lado são:

- `mulher(ana).`
- `homem(bob).`
- `casal(ana,bob).`
- `gerou(ana,clô).`
- `gerou(bob,clô).`
- `irmão(rui,rai).`
- `tio(raí,bia).`



Devemos evitar declarar fatos que podem ser deduzidos de outros!

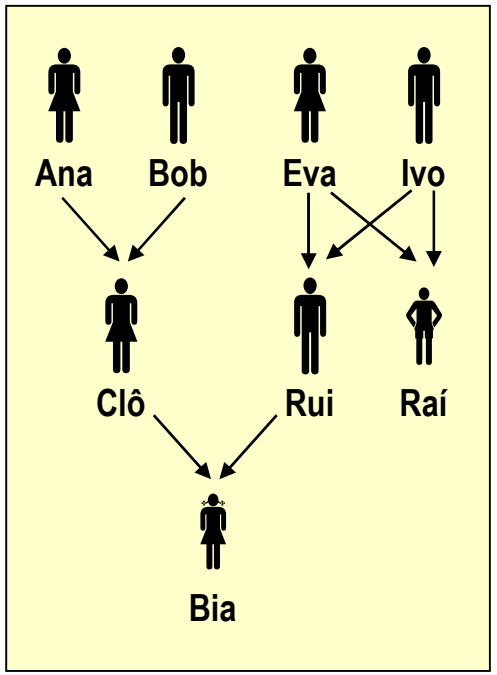
Regra

Uma regra

estabelece um relacionamento **condicional** entre objetos de um contexto.

Algumas regras no contexto ao lado são:

```
casal(X,Y) :- gerou(X,Z), gerou(Y,Z), X\=Y.
mãe(X,Y)   :- mulher(X),  gerou(X,Y).
filho(X,Y) :- homem(X),   gerou(Y,X).
avó(X,Y)   :- mãe(X,Z),   gerou(Z,Y).
```

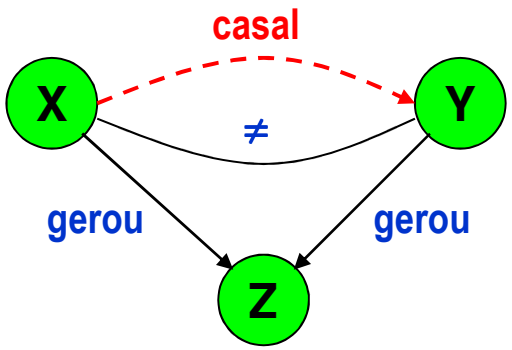


Regras são definidas em termos de fatos e regras já definidos!

Regra

Um grafo de relacionamento

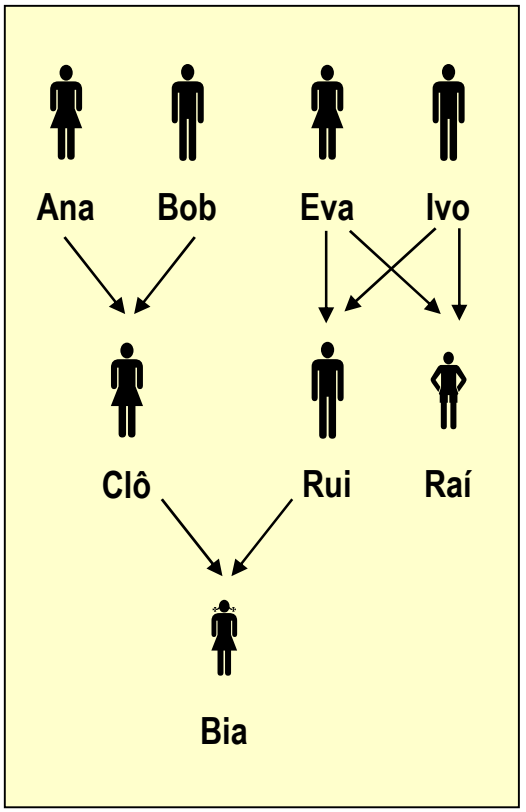
permite visualizar graficamente uma regra, facilitando a sua definição/codificação.




```
casal(X,Y) :- gerou(X,Z), gerou(Y,Z), X\=Y.
```

conclusão

condições



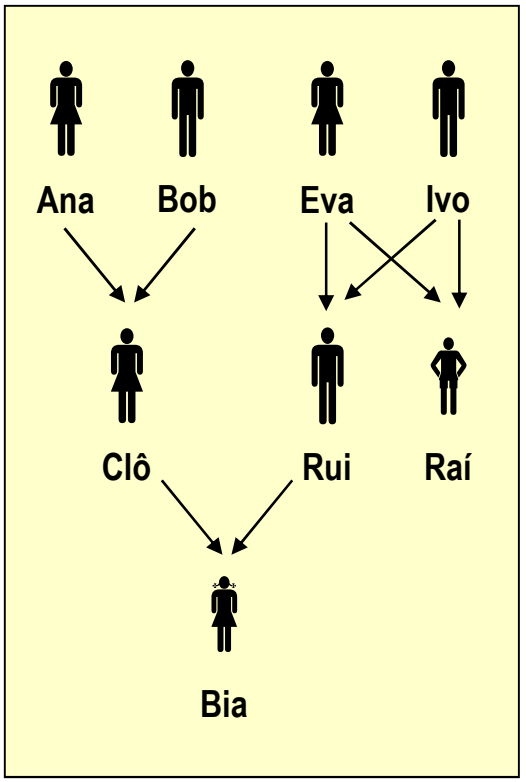
Programa

Um programa

é um conjunto de fatos e regras.

```
% gerou(X,Y) : X gerou Y
gerou(ana,clô).
gerou(bob,clô).
gerou(eva,rui).
gerou(eva,raí).
gerou(ivo,rui).
gerou(ivo,raí).
gerou(clô,bia).
gerou(rui,bia).

% casal(X,Y) : X e Y formam um casal
casal(X,Y) :- gerou(X,Z), gerou(Y,Z), X\=Y.
```



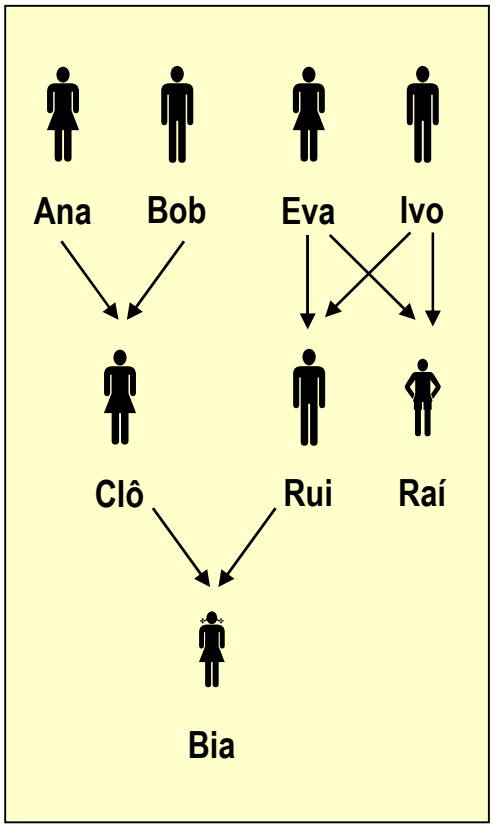
Consulta

Uma consulta

permite obter informações que podem ser deduzidas de um programa.

Algumas consultas que poderíamos fazer:

```
?- gerou(ana,clô).
yes
?- gerou(ana,rui).
no
?- gerou(Q,clô).
Q = ana ;
Q = bob
yes
?- gerou(clô,Q).
Q = bia
yes
```

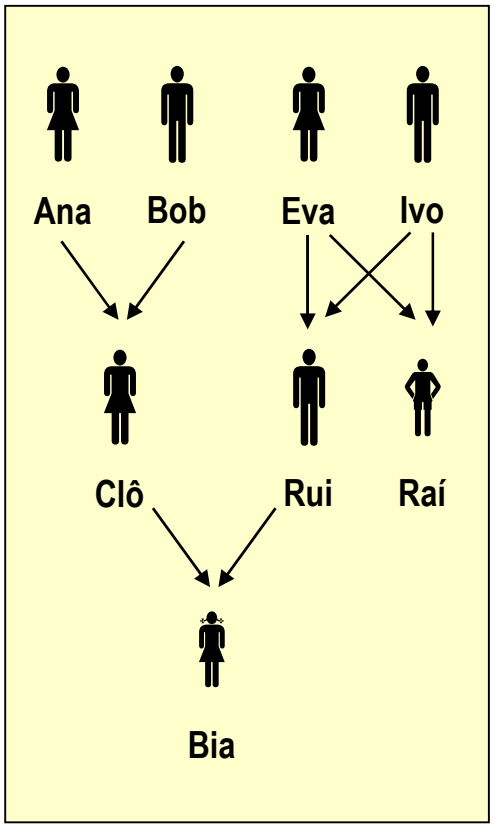


Consulta

Uma variável compartilhada

permite propagar objetos entre literais de uma consulta.

```
?- gerou(X,Y), gerou(Y,Z).
X = ana, Y = clô, Z = bia ;
X = bob, Y = clô, Z = bia ;
X = eva, Y = rui, Z = bia ;
X = ivo, Y = rui, Z = bia ;
no
```

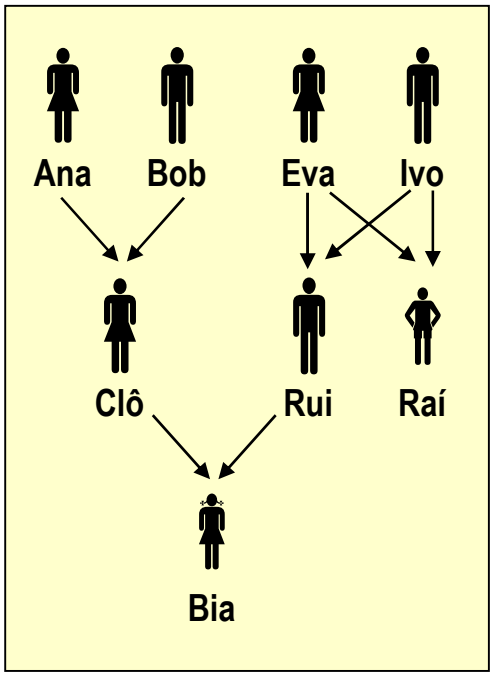


Consulta

Uma variável anônima (indicada pelo sinal de sublinha)

permite indicar objetos cujos valores são irrelevantes para obtenção de uma resposta.

```
% Ana gerou alguém?
?- gerou(ana, _ ).
yes
% Bia gerou alguém?
?- gerou(bia, _ ).
no
% Quem é avó/avô de Raí?
?- gerou(Q, _ ), gerou( _ , raí).
Q = ana <= ERRO
```



Cada ocorrência da variável anônima é distinta e, portanto, não compartilhada!

Eliminação de respostas duplicadas

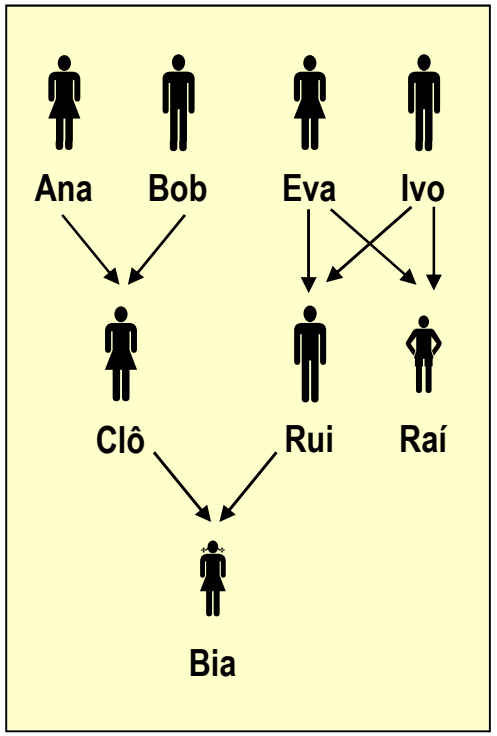
Predicados predefinidos em Prolog

- `findall(R,C,L)`: encontra todas respostas R da consulta C e as guarda na lista L
- `sort(L,S)`: ordena a lista L, removendo duplicatas, e devolve o resultado em S

```
?- irmão(rui,R).
R = raí ;
R = raí ;
no

?- findall(R, irmão(rui,R), L).
L = [raí, raí]
yes

?- findall(R, irmão(rui,R), L), sort(L,S).
L = [raí, raí]
S = [raí]
yes
```



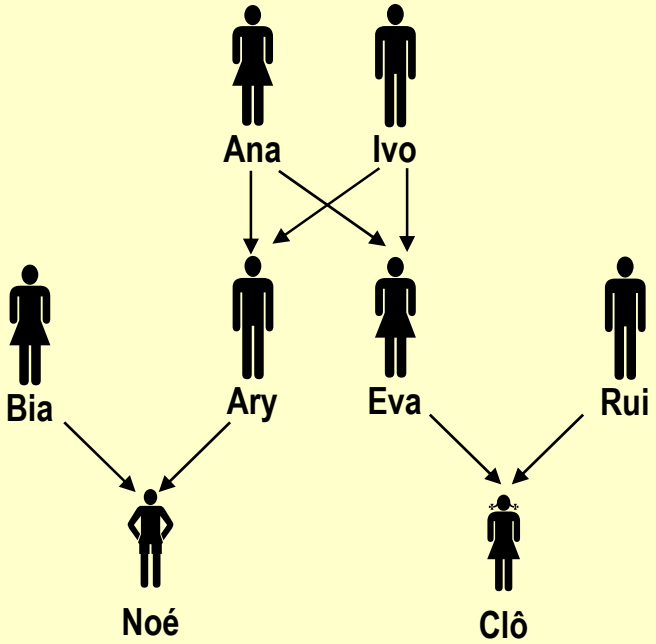
Exercício

Exercício 1. Genealogia

Com base no contexto ao lado, defina as relações a seguir e consulte o sistema.

- homem
- mulher
- gerou
- casal
- filha
- filho
- irmão
- irmã

- mãe
- pai
- avó
- avô
- tia
- tio
- prima
- primo



Exercício

Exercício 2. Carona

- *Ana e Raí moram em Santana, Bia mora no Tatuapé, Edu mora no Mandaqui, Gil mora na Penha e Eva mora na Vila Carrão.*
- *Santana e Mandaqui ficam na zona norte e Tatuapé, Penha e Vila Carrão ficam na zona leste.*
- *Ana e Gil têm carro.*
- *Uma pessoa pode dar carona à outra se ela tem carro e ambas moram em bairros que ficam na mesma zona.*

Represente estas informações, usando os predicados a seguir:

- `mora_em(Pessoa,Bairro)`
- `fica_na_zona(Bairro,Zona)`
- `tem_carro(Pessoa)`
- `pode_dar_carona_a(Pessoa,Pessoa)`

Exercício

Exercício 4. Negação por falha finita

```
pessoa(ana).
pessoa(bia).
baixa(ana).
alta(X) :- not(baixa(X)), pessoa(X).
```

Considerando o programa acima:

- O que o sistema Prolog responde à consulta:

```
?- alta(R).
```

- Que modificação é necessária para que a consulta seja respondida corretamente?

Processamento numérico

cálculo

aritmética

comparação

Operadores aritméticos

Embora Prolog seja especialmente voltado para processamento simbólico, ele também oferece algumas facilidades para processamento numérico, por meio do predicado predefinido `is/2`.

```
?- X = 1 + 3.
X = 1+3
yes

?- X is 1 + 3.
X = 4
yes

?- X is 7 // 2.
X = 3
yes

?- X is 4 ^ 0.5.
X = 2
yes
```

Operação aritmética	Operador
Soma	+
Subtração	-
Multiplicação	*
Divisão real	/
Divisão inteira	//
Resto da divisão inteira	mod
Potenciação	^

Operadores relacionais

Para realizar comparações entre expressões aritméticas, podemos usar os operadores relacionais listados na tabela a lado.

```
?- 1+3 = 3+1.
no

?- 1+3 =:= 3+1.
yes

?- X = 2+3, Y = 1+4, X =< Y.
X = 2+3
Y = 1+4
yes

?- ana @> amélia.
yes
```

Comparação	Operador
Igual	=:=
Diferente	=\=
Menor	<
Menor ou igual	=<
Maior	>
Maior ou igual	>=

Para comparar valores alfanuméricos, usamos ==, \=, @<, @=<, @>, @>=.

Exercício

Exercício 5. Área do imóvel

```
medida(cozinha, 2.0, 3.0).
medida(sala,    4.0, 5.5).
medida(quarto,  3.0, 3.5).
medida(banheiro,1.5, 2.0).
```

Complemente o programa acima com a definição do predicado **`tamanho(Cômodo,Área)`** e faça as seguintes consultas:

- Qual área da cozinha?
- Que cômodos são maiores que a cozinha?
- Que cômodos são menores que a cozinha?
- É verdade que o tamanho da cozinha é o dobro do tamanho do banheiro?
- É verdade que a sala é o maior cômodo da casa?

Exercício

Exercício 6. Área e população dos países (em milhões)

```
% país(Nome,ÁreaKm2,População).

país(brasil, 8,  196).
país(china,  9, 1330).
país(eua,    9,  304).
país(índia,  3, 1147).
```

Com base no programa acima, faça as seguintes consultas:

- Qual o número de habitantes dos EUA?
- Qual a densidade demográfica do Brasil?
- Qual a diferença entre as populações da China e da Índia?
- Que países são maiores que o Brasil?
- Que países são menos populosos que a Índia?

Modelo relacional

tabela

consulta

Modelo relacional

- O modelo relacional de banco de dados (BD)
  - Representa os dados em um BD usando um conjunto de **tabelas**.
  - As linhas das tabelas são denominadas **tuplas** e suas colunas, **atributos**.
  - As **consultas** são representadas em termos de operações da álgebra relacional (projeção, seleção, etc.).
- Programação em lógica é uma poderosa extensão do modelo relacional.
  - Um conjunto de fatos para um predicado corresponde a uma tabela.
  - Fatos representam tuplas e seus argumentos representam atributos da tabela.
  - Uma regra corresponde a uma vista no modelo relacional, isto é, uma tabela virtual que reúne atributos já armazenados em outras tabelas.
  - As operações de álgebra relacional podem ser representadas por meio de consultas (ou regras).

Modelo relacional: tabelas

Primeira Forma Normal (1FN)

Todos os atributos devem ser atômicos.

Dados

Código	Nome	Salário	Dependentes
107	Ana	5.500,00	Bia, Lia
290	Rui	7.850,00	Raí
368	Eva	2.390,00	---
405	Ivo	4.700,00	Clô, Ary, Noé

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Dependentes

Código	Nome
107	Bia
107	Lia
290	Raí
405	Clô
405	Ary
405	Noé

Dados na 1FN

Modelo relacional: tabelas

Tabelas representadas em Prolog

```
% func(Código,Nome,Salário)

func(107,ana,5500).
func(290,rui,7850).
func(368,eva,2390).
func(405,ivo,4700).

% dep(Código,Nome)

dep(107,bia).
dep(107,lia).
dep(290,raí).
dep(405,clô).
dep(405,ary).
dep(405,noé).
```

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Dependentes

Código	Nome
107	Bia
107	Lia
290	Raí
405	Clô
405	Ary
405	Noé

Modelo relacional: consultas

Projeção

Seleciona um conjunto de atributos (colunas) de uma tabela.

Exemplo: Quais os nomes e salários dos funcionários?

```
?- func(_,N,S).
N = ana, S = 5500 ;
N = rui, S = 7850 ;
N = eva, S = 2390 ;
N = ivo, S = 4700
yes
?- forall(func(_,N,S), writeln([N,S])).
[ana, 5500]
[rui, 7850]
[eva, 2390]
[ivo, 4700]
yes
```

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Modelo relacional: consultas

Seleção

Seleciona um conjunto de tuplas (linhas) de uma tabela, de acordo com uma condição

Exemplo: Quem recebe salário entre 3 e 6 mil reais?

```
?- func(C,N,S), S>=3000, S=<6000.
C = 107, N = ana, S = 5500 ;
C = 405, N = ivo, S = 4700
yes

?- forall((func(C,N,S), S>=3000, S=<6000),
          writeln([C,N,S])).

[107, ana, 5500]
[405, ivo, 4700]
yes
```

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Modelo relacional: consultas

Um relacionamento entre tabelas

É estabelecido com o uso variáveis compartilhadas.

Funcionários

Código	Nome	Salário
107	Ana	5.500,00
290	Rui	7.850,00
368	Eva	2.390,00
405	Ivo	4.700,00

Exemplo: Quem são os dependentes de Ivo?

```
?- forall((func(C,ivo,_), dep(C,N)), writeln(N)).
clô
ary
noé

yes
```

Dependentes

Código	Nome
107	Bia
107	Lia
290	Raí
405	Clô
405	Ary
405	Noé

Exercício

Exercício 7. Funcionários

```
% func(Código,Nome,Salário)

func(107,ana,5500).
func(290,rui,7850).
func(368,eva,2390).
func(405,ivo,4700).

% dep(Código,Nome)

dep(107,bia).
dep(107,lia).
dep(290,raí).
dep(405,clô).
dep(405,ary).
dep(405,noé).
```

Com base no programa a lado, faça as seguintes consultas:

- Qual o salário de Eva?
- Qual o código do Rui?
- Quem é dependente de Ana?
- De quem Raí é dependente?
- Quem não tem dependente?
- Quem recebe salário de no máximo R$ 5.500,00?
- Quem depende de funcionário com salário de no máximo R$ 5.500,00?

Fim