 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Birigui</p>	<p><b>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA</b> <b>Campus Birigui</b> <b>Bacharelado em Engenharia de Computação</b></p>	
<b>Disciplina:</b> Inteligência Artificial	<b>Lista 1</b>	
<b>Professor:</b> Prof. Dr. Murilo Vargas da Silva	<b>Data:</b> 25/09/2023	
<b>Nome do Aluno:</b> Henrique Akira Hiraga	<b>Prontuário:</b> BI300838X	

## LISTA 1 – PROLOG

1. Digite o Programa 1.1, incluindo as regras que definem as relações avô e irmão, e realize as seguintes consultas:

### Programa 1.1: *Uma árvore genealógica.*

```

pai (adão, cain) .
pai (adão, abel) .
pai (adão, seth) .
pai (seth, enos) .

```

#### 1.3 Regras

*Regras* nos permitem definir novas relações em termos de outras relações já existentes. Por exemplo, a regra

```
avô(X,Y) :- pai(X,Z), pai(Z,Y) .
```

define a relação *avô* em termos da relação *pai*, ou seja, estabelece que *X* é avô de *Y* se *X* tem um filho *Z* que é pai de *Y*. Com essa regra, podemos agora realizar consultas tais como

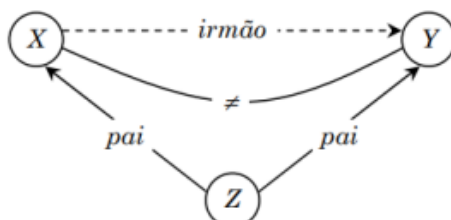
```

?- avô(X, enos) .
X = adão

```

Fatos e regras são tipos de *cláusulas* e um conjunto de cláusulas constitui um *programa lógico*.

Como mais um exemplo, vamos definir a relação *irmão* em termos da relação *pai*, já existente. Podemos dizer que duas pessoas distintas são irmãs se ambas têm o mesmo pai. Essa regra é representada pelo seguinte grafo:



Em Prolog, essa regra é escrita como:

```
irmão(X,Y) :- pai(Z,X), pai(Z,Y), X\=Y.
```

Código utilizado para realizar as consultas:

```
pai(adão,cain).
pai(adão,abel).
pai(adão,seth).
pai(seth,enos).
avô(X,Y):-pai(X,Z),pai(Z,Y).
irmão(X,Y):-pai(Z,X),pai(Z,Y),X\=Y.
```

a) Quem são filhos de Adão?

```
?- pai(adão,X).
X = cain ;
X = abel ;
X = seth.
```

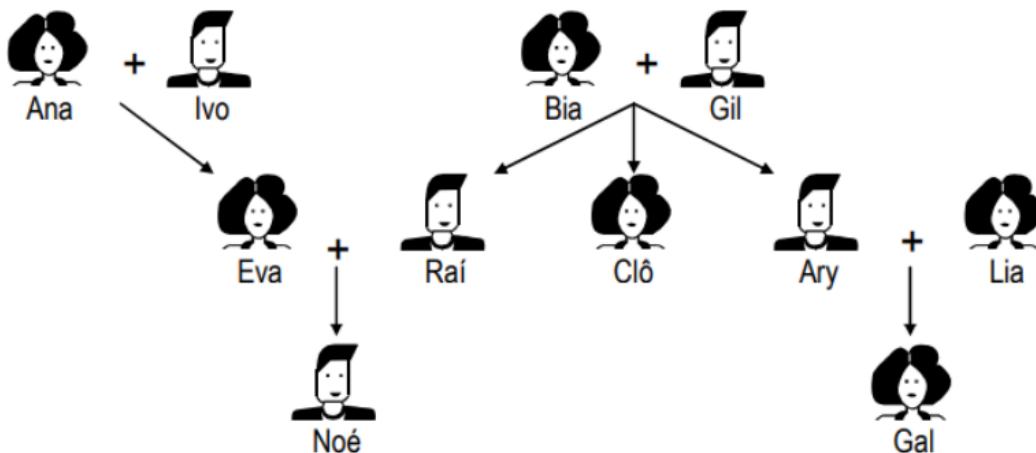
b) Quem são os netos de Adão?

```
?- avô(adão,X).
X = enos.
```

c) Quem são os tios de Enos?

```
?- irmão(seth,X).
X = cain ;
X = abel ;
false.
```

2. Considere a árvore genealógica a seguir:



Código utilizado para as consultas:

```

mãe(ana,eva) .
mãe(bia,rai) .
mãe(bia,clô) .
mãe(bia,ary) .
mãe(eva,noé) .
mãe(lia,gal) .
pai(ivo,eva) .
pai(rai,noé) .
pai(ary,gal) .
pai(gil,rai) .
pai(gil,clô) .
pai(gil,ary) .
mulher(ana) .
mulher(eva) .
mulher(bia) .
mulher(clô) .
mulher(lia) .
mulher(gal) .
homem(ivo) .
homem(rai) .
homem(noé) .
homem(gil) .
homem(ary) .
gerou(X,Y):-mãe(X,Y);pai(X,Y) .
irmão(X,Y):-pai(Z,X),pai(Z,Y),X\=Y,homem(X) .
irmã(X,Y):-pai(Z,X),pai(Z,Y),X\=Y,mulher(X) .
filho(X,Y):-gerou(Y,X),homem(X) .
filha(X,Y):-gerou(Y,X),mulher(X) .
tio(X,Y):-pai(Z,Y),mãe(M,Y),irmão(X,Z);irmão(X,M),Z\=M.
tia(X,Y):-pai(Z,Y),mãe(M,Y),irmã(X,Z);irmã(X,M),Z\=M.
primo(X,Y):-filho(X,Z),tio(Z,Y) .
prima(X,Y):-filha(X,Z),tio(Z,Y) .
avô(X,Y):-homem(X),gerou(X,Z),gerou(Z,Y) .

```

- a) Usando fatos, defina as relações pai e mãe. Em seguida, consulte o sistema para ver se suas definições estão corretas.

```

?- pai(X,noé) .
X = rai .

```

```

?- mãe(X,noé) .
X = eva .

```

- b) Acrescente ao programa os fatos necessários para definir as relações homem e mulher. Por exemplo, para estabelecer que Ana é mulher e Ivo é homem, acrescente os fatos mulher(ana) e homem(ivo).

```
?- mulher(eva).  
true.
```

```
?- homem(rai).  
true.
```

- c) Usando duas regras, defina a relação gerou(X,Y) tal que X gerou Y se X é pai ou mãe de Y. Faça consultas para verificar se sua definição está correta. Por exemplo, para a consulta gerou(X,eva) o sistema deverá apresentar as respostas X = ana e X = ivo.

```
?- gerou(X,eva).  
X = ana ;  
X = ivo.
```

- d) Usando relações já existentes, crie regras para definir as relações filho, filha, tio, tia, primo, prima, avô e avó. Para cada relação, desenhe o grafo de relacionamentos (Efetue leitura da apostila da USP), codifique a regra correspondente e faça consultas para verificar a corretude.

```
?- filho(noé,Y).  
Y = eva ;  
Y = rai.
```

```
?- filha(eva,Y).  
Y = ana ;  
Y = ivo.
```

```
?- tio(X,noé).  
X = ary ;  
false.
```

```
?- tia(X,noé).  
X = clô ;  
false.
```

```
?- primo(X,gal).  
X = noé ;  
false.
```

```
?- prima(X,noé).  
X = gal ;  
false.
```

```
?- avô(X,noé).  
X = ivo ;  
X = gil ;  
false.
```

```
?- avó(X,noé).  
X = ana ;  
X = bia ;  
false.
```

3. Codifique as regras equivalentes às seguintes sentenças:

a) Todo mundo que tem filhos é feliz.

```
feliz(X):-tem_filhos(X).  
tem_filhos(X):-pai(X,_);mãe(X,_).
```

b) Um casal é formado por duas pessoas que têm filhos em comum.

```
casal(X,Y):-tem_filhos_em_comum(X,Y).  
tem_filhos_em_comum(X,Y):-pai(X,Z),pai(Y,Z);mãe(X,Z),mãe(Y,Z).
```