

1) KARIERA.PRO

```
% Versión 1
% formalizacion del sistema kariera con resoluciones redundantes
% problemas : es posible eliminar reglas?
%           se puede evitar la redundancia?
```

domains

```
alguien, sexo, tipo = symbol
```

predicates

```
casable(alguien,alguien)
conyuge(alguien,alguien)
individuo(alguien)
progenitor(alguien,alguien)
sexo(alguien,sexo)
tipo(alguien,tipo)
```

clauses

```
casable(A,B) :- sexo(A,varon), sexo(B,mujer),
                tipo(A,X), tipo(B,Y), X=Y.
```

% alternativa pslm

```
casable(A,B) :- conyuge(A,B).
```

```
tipo(h5,t1).
tipo(h7,t2).
```

```
tipo(A,X) :- progenitor(A,B), tipo(B,t1), sexo(A,varon), X=t2.
tipo(A,X) :- progenitor(A,B), tipo(B,t1), sexo(A,mujer), X=t3.
tipo(A,X) :- progenitor(A,B), tipo(B,t2), sexo(A,varon), X=t3.
tipo(A,X) :- progenitor(A,B), tipo(B,t2), sexo(A,mujer), X=t1.
tipo(A,X) :- progenitor(A,B), tipo(B,t3), sexo(A,varon), X=t1.
tipo(A,X) :- progenitor(A,B), tipo(B,t3), sexo(A,mujer), X=t2.
```

```
tipo(A,X) :- conyuge(A,B), tipo(B,Z), Z=X.
```

```
conyuge(A,B) :- progenitor(C,A), progenitor(C,B), A<>B.
```

```
progenitor(h1,h5).
progenitor(h1,m5).
progenitor(m1,h6).
progenitor(m1,m6).
progenitor(h2,h1).
progenitor(h2,m1).
progenitor(m2,h1).
```

```
progenitor(m2,m1).
progenitor(h3,h7).
progenitor(h3,m7).
progenitor(m3,h8).
progenitor(m3,m8).
progenitor(h4,h3).
progenitor(h4,m3).
progenitor(m4,h3).
progenitor(m4,m3).
```

```
sexo(h1,varon).
sexo(h2,varon).
sexo(h3,varon).
sexo(h4,varon).
sexo(h5,varon).
sexo(h6,varon).
sexo(h7,varon).
```

```
sexo(X,mujer) :- individuo(X), not(sexo(X,varon)).
```

```
individuo(A) :- progenitor(A,_).
individuo(A) :- progenitor(_,A).
individuo(A) :- conyuge(A,_).
```

```
*****
```

```
% Versión 2 - Con aprendizaje de máquina
% formalizacion del sistema kariera con resoluciones redundantes
```

```
domains
```

```
alguien, sexo, tipo = symbol
```

```
database
```

```
asertipo(alguien,tipo)
repetido(alguien,alguien)
```

```
predicates
```

```
casable(alguien,alguien)
conyuge(alguien,alguien)
individuo(alguien)
preliminar
progenitor(alguien,alguien)
sexo(alguien,sexo)
tipo(alguien,tipo)
```

```
clauses
```

```
casable(A,B) :-
    sexo(A,varon), sexo(B,mujer),
```

tipo(A,X), tipo(B,X),
not(repetido(A,B)), assert(repetido(A,B)).

casable(A,B) :- conyuge(A,B),
not(repetido(A,B)), assert(repetido(A,B)).

casable(A,B) :- conyuge(B,A),
not(repetido(A,B)), assert(repetido(A,B)).

asertipo(h5,t1).
asertipo(m5,t1).
asertipo(h7,t2).
asertipo(m7,t2).

preliminar :- sexo(A,varon),
progenitor(A,B), asertipo(B,t3), assert(asertipo(A,t1)),
fail.

preliminar :- sexo(A,mujer),
progenitor(A,B), asertipo(B,t2), assert(asertipo(A,t1)),
fail.

preliminar :- sexo(A,varon),
progenitor(A,B), asertipo(B,t1), assert(asertipo(A,t2)),
fail.

preliminar :- sexo(A,mujer), progenitor(A,B), asertipo(B,t3),
assert(asertipo(A,t2)),fail.

preliminar :- sexo(A,varon), progenitor(A,B), asertipo(B,t2),
assert(asertipo(A,t3)),fail.

preliminar :- sexo(A,mujer), progenitor(A,B), asertipo(B,t1),
assert(asertipo(A,t3)),fail.

tipo(A,X) :- asertipo(A,X).

tipo(A,X) :- conyuge(A,B), asertipo(B,X).

tipo(A,X) :- conyuge(B,A), asertipo(B,X).

tipo(A,X) :- progenitor(A,B), tipo(B,t1), sexo(A,varon), X=t2.

tipo(A,X) :- progenitor(A,B), tipo(B,t1), sexo(A,mujer), X=t3.

tipo(A,X) :- progenitor(A,B), tipo(B,t2), sexo(A,varon), X=t3.

tipo(A,X) :- progenitor(A,B), tipo(B,t2), sexo(A,mujer), X=t1.

tipo(A,X) :- progenitor(A,B), tipo(B,t3), sexo(A,varon), X=t1.

tipo(A,X) :- progenitor(A,B), tipo(B,t3), sexo(A,mujer), X=t2.

conyuge(h1,m1).

conyuge(h3,m3).

conyuge(h5,m5).

conyuge(h6,m6).

conyuge(h7,m7).

conyuge(h8,m8).

progenitor(h1,h5).

progenitor(h1,m5).

progenitor(m1,h6).

progenitor(m1,m6).
progenitor(h2,h1).
progenitor(h2,m1).
progenitor(m2,h1).
progenitor(m2,m1).
progenitor(h3,h7).
progenitor(h3,m7).
progenitor(m3,h8).
progenitor(m3,m8).
progenitor(h4,h3).
progenitor(h4,m3).
progenitor(m4,h3).
progenitor(m4,m3).

sexo(h1,varon).
sexo(h2,varon).
sexo(h3,varon).
sexo(h4,varon).
sexo(h5,varon).
sexo(h6,varon).
sexo(h7,varon).
sexo(A,mujer) :- individuo(A), not(sexo(A,varon)).

individuo(h1).
individuo(h2).
individuo(h3).
individuo(h4).
individuo(h5).
individuo(h6).
individuo(h7).
individuo(h8).
individuo(m1).
individuo(m2).
individuo(m3).
individuo(m4).
individuo(m5).
individuo(m6).
individuo(m7).
individuo(m8).