



EJERCICIO 1 - PROLOG

PAULA ANDREA PARRA CASTAÑO



PROGRAMACION III

PROFESOR: RAMIRO ANDRES BARRIOS

25 DE FEBRERO DE 2025
UNIVERSIDAD TECNÓLOGICA DE PEREIRA

EJERCICIO 1:

Hechos:

```
File Edit Selection View Go ... ← →
actividad1.pl X
actividad1.pl
1  es_padre(abraham,herbert).
2  es_padre(abraham,homero).
3  es_padre(clancy,marge).
4  es_padre(clancy,patty).
5  es_padre(clancy,selma).
6  es_padre(homero,bart).
7  es_padre(homero,lisa).
8  es_padre(homero,maggie).
9
10 es_madre(mona,herbert).
11 es_madre(mona,homero).
12 es_madre(jacqueline,marge).
13 es_madre(jacqueline,patty).
14 es_madre(jacqueline,selma).
15 es_madre(marge,bart).
16 es_madre(marge,lisa).
17 es_madre(marge,maggie).
18 es_madre(selma,ling).
19
```

Reglas:

```
20
21 es_abuelo(X,Y) :- es_padre(X,Z) , es_padre(Z,Y). % ,=and ;=or
22 es_abuelo(X,Y) :- es_padre(X,Z) , es_madre(Z,Y).
23 es_abuela(X,Y) :- es_madre(X,Z) , es_padre(Z,Y).
24 es_abuela(X,Y) :- es_madre(X,Z) , es_madre(Z,Y).
25
26 son_hermanos(X,Y) :- (es_padre(M,X) , es_padre(M,Y)) , X \= Y. % parentesis para separar condiciones , \= diferente
27
28
29 es_tio(X,Y) :- son_hermanos(X,P),
30               (es_padre(P,Y) ; es_madre(P,Y)).
31
32 son_primos(X,Y) :- (es_padre(P,X) ; es_madre(P,X)) ,
33                   (es_padre(M,Y) ; es_madre(M,Y)) ,
34                   son_hermanos(P,M).
```

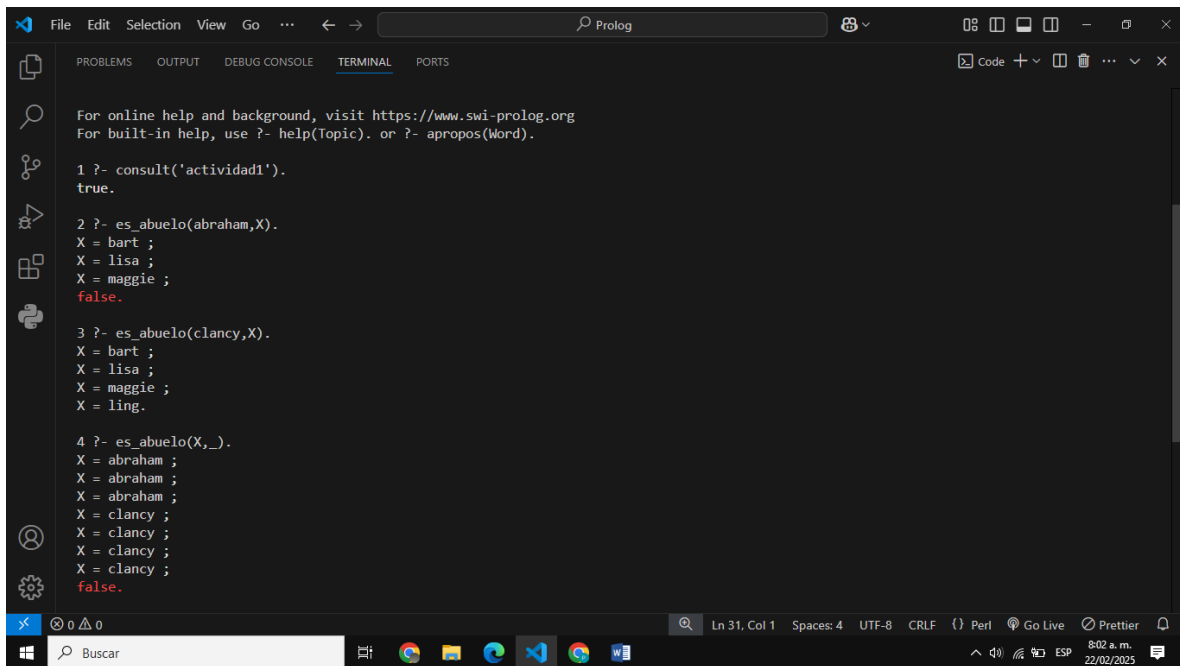
Preguntas:

?- Abuelos, nietos

Se utiliza un hecho para realizar la consulta, se pone un funtor llamado es_abuelo o es_abuela con parámetros X, Y como datos a utilizar. Se relaciona un si es_padre o si es_madre para que el programa indague entre los padres hombres y mujeres y la Z hace referencia al hijo. Si el hijo de la persona que buscamos tiene otro hijo asegura que es abuelo.

Con estas cuatro reglas se puede saber:

- 🚦 Quienes son abuelos y abuelas.
- 🚦 Quienes son nietos y nietas.



```
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- consult('actividad1').
true.

2 ?- es_abuelo(abraham,X).
X = bart ;
X = lisa ;
X = maggie ;
false.

3 ?- es_abuelo(clancy,X).
X = bart ;
X = lisa ;
X = maggie ;
X = ling.

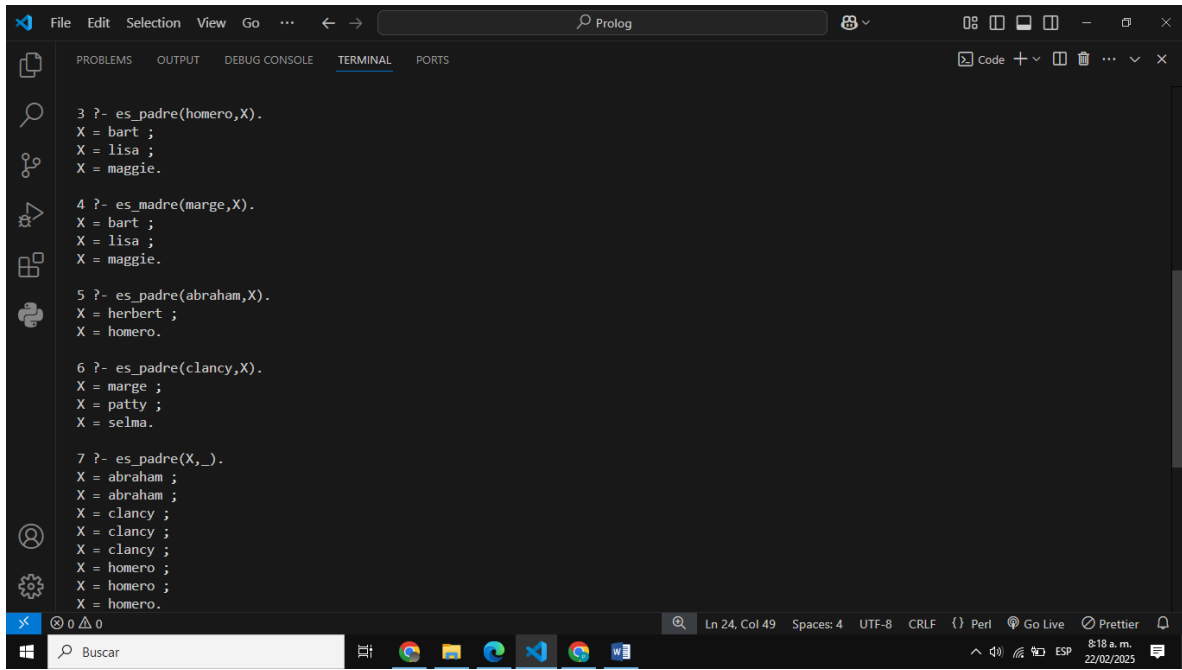
4 ?- es_abuelo(X,_).
X = abraham ;
X = abraham ;
X = abraham ;
X = clancy ;
X = clancy ;
X = clancy ;
X = clancy ;
false.
```

?- Hijos, padres

Se utilizan los funtores es_padre y es_madre. Es una consulta desde los hechos.

Se puede preguntar:

- 🚦 Quien es el papá o la mamá de alguien.
- 🚦 Quien es el hijo de quien.
- 🚦 Quienes son hijos.
- 🚦 Quienes son padres.



```
3 ?- es_padre(homero,X).  
X = bart ;  
X = lisa ;  
X = maggie.  
  
4 ?- es_madre(marge,X).  
X = bart ;  
X = lisa ;  
X = maggie.  
  
5 ?- es_padre(abraham,X).  
X = herbert ;  
X = homero.  
  
6 ?- es_padre(clancy,X).  
X = marge ;  
X = patty ;  
X = selma.  
  
7 ?- es_padre(X,_).  
X = abraham ;  
X = abraham ;  
X = clancy ;  
X = clancy ;  
X = clancy ;  
X = clancy ;  
X = homero ;  
X = homero ;  
X = homero.
```

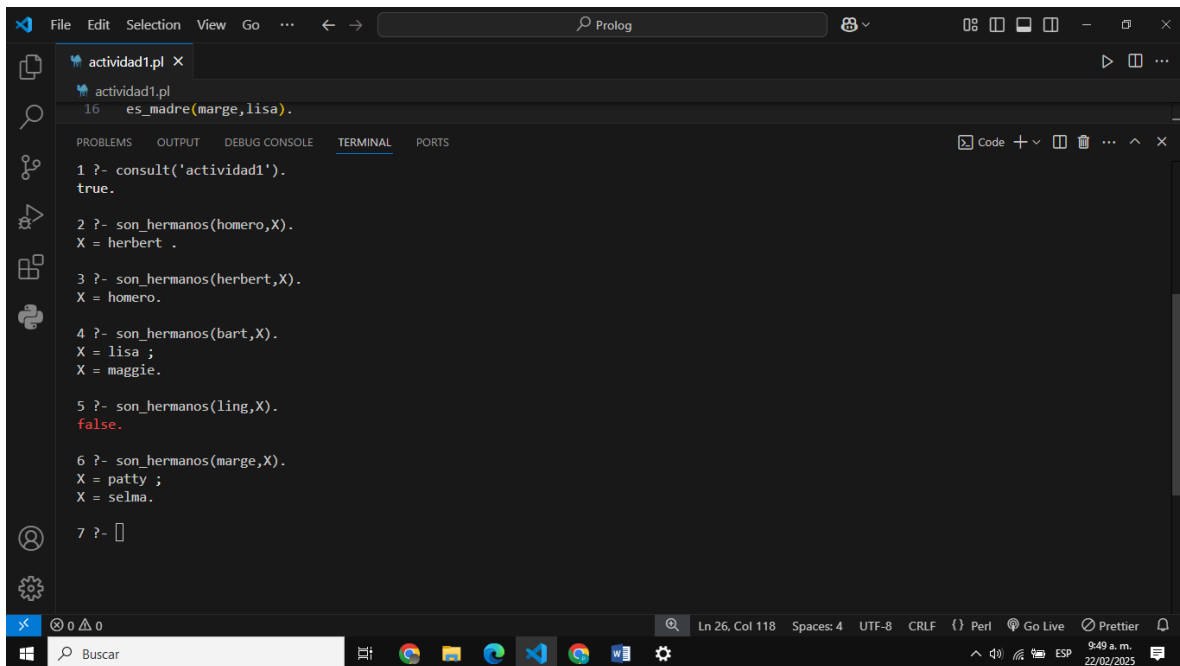
? - hermanos

`son_hermanos(X,Y) :- (es_padre(M,X) , es_padre(M,Y)) , X \= Y.`

Se utiliza esta regla para determinar quien comparte padre y que X y Y no sean iguales para determinar quien comparte el mismo padre. Los paréntesis se utilizan para separar condiciones.

Se puede preguntar:

- Quienes son hermanos
- Si alguien tiene hermano



The screenshot shows a Prolog IDE window titled 'Prolog'. The main editor displays a file named 'actividad1.pl' with the following code:

```
16 es_madre(marge,lisa).
```

The terminal window shows the following queries and results:

```
1 ?- consult('actividad1').
true.

2 ?- son_hermanos(homero,X).
X = herbert .

3 ?- son_hermanos(herbert,X).
X = homero.

4 ?- son_hermanos(bart,X).
X = lisa ;
X = maggie.

5 ?- son_hermanos(ling,X).
false.

6 ?- son_hermanos(marge,X).
X = patty ;
X = selma.

7 ?- 
```

The status bar at the bottom indicates 'Ln 26, Col 118', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Perl'. The system tray shows the time as 9:49 a.m. on 22/02/2025.

? – tíos

(Realicé la prueba por Swish porque estaba arrojando error por VSC)

La regla **es_tio** tiene como parametros(**tio,sobрино**) donde se cumple si el **tio** tiene un **hermano** y ese **hermano** tiene un **hijo**, si es asi la regla se cumple y menciona los datos a las preguntas.

Se puede preguntar:

- Quienes son tíos
- Si alguien tiene tío
- Quienes son los sobrinos

The image shows the SWISH Prolog IDE interface. The main editor contains the following Prolog code:

```

10 es_madre(mona,herbert).
11 es_madre(mona,homero).
12 es_madre(jacqueline,marge).
13 es_madre(jacqueline,patty).
14 es_madre(jacqueline,selma).
15 es_madre(marge,bart).
16 es_madre(marge,lisa).
17 es_madre(marge,maggie).
18 es_madre(selma,ling).
19
20
21 es_abuelo(X,Y) :- es_padre(X,Z) , es_padre(Z,Y). % ,=and ;=or
22 es_abuelo(X,Y) :- es_padre(X,Z) , es_madre(Z,Y).
23 es_abuela(X,Y) :- es_madre(X,Z) , es_padre(Z,Y).
24 es_abuela(X,Y) :- es_madre(X,Z) , es_madre(Z,Y).
25
26 son_hermanos(X,Y) :- (es_padre(M,X) , es_padre(M,Y)) , X \= Y. % parentesis par
27
28
29 es_tio(X,Y) :- son_hermanos(X,P),
30               (es_padre(P,Y) ; es_madre(P,Y)).
31
32
33
34
35

```

The right sidebar shows the execution results for three queries:

- es_tio(herbert,X).**
 - X = bart
 - X = lisa
 - X = maggie
- es_tio(selma,X).**
 - X = bart
 - X = lisa
 - X = maggie
 - false
- es_tio(X,_).**
 - X = herbert

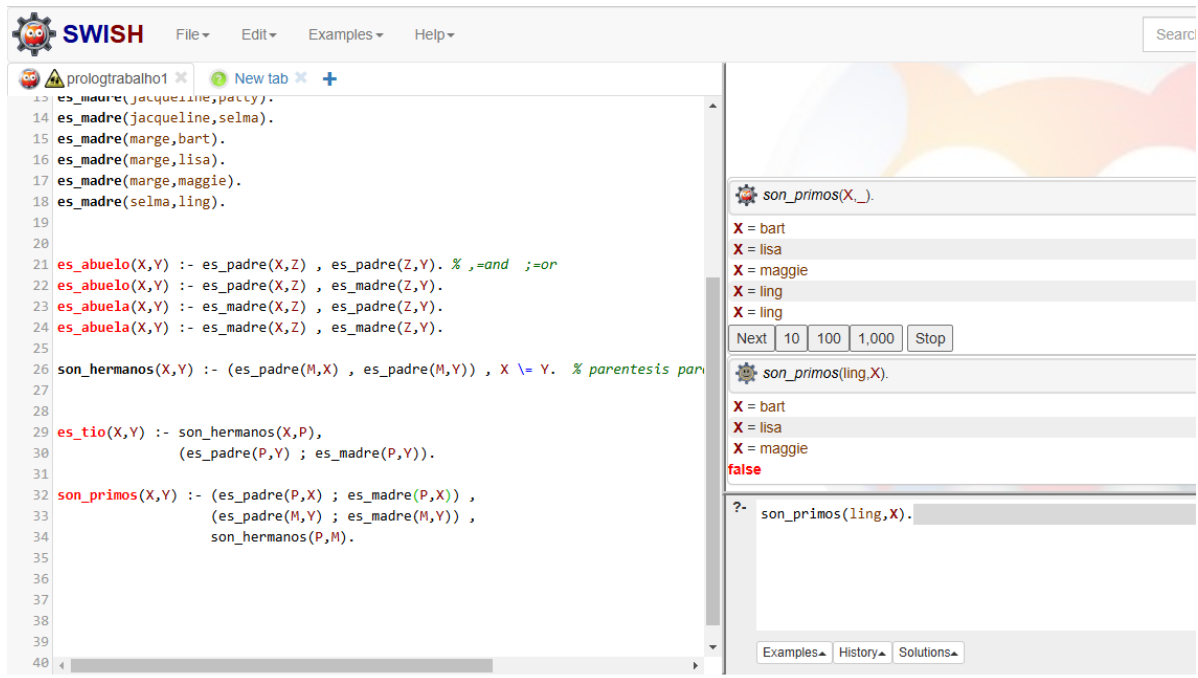
At the bottom of the sidebar, there are buttons for "Next", "10", "100", "1,000", and "Stop".

? - Quiénes son primos

La regla `son_primos(primoX, primoY)` se cumple si sus **papás** o **mamás** tienen **hermanos** y esos **hermanos** tienen **hijos**, ahí se cumple la condición.

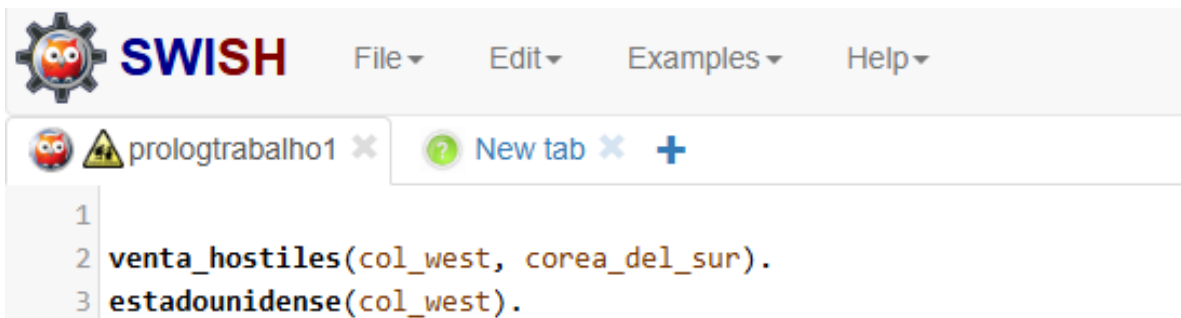
Se puede preguntar:

- Quiénes son primos
- Si alguien tiene primo



EJERCICIO 2:

Hechos:



Reglas:

```
4  
5 criminal(X) :-  
6     estadounidense(X), venta_hostiles(X,_).  
7  
8
```


Preguntas:

Se utiliza el funtor `venta_hostiles` para determinar el **vendedor** y el **pais hostil**, además se agrega otro hecho de **estadounidense**. En la regla se determina si el **estadounidense** es el mismo **vendedor** del pais hostil es un **criminal**.



 `criminal(paula).`

false

 `criminal(col_west).`

true

?- `criminal(col_west).`