

Solución con listas y estructuras

% padre

```

padre(abraham, hijos([helbert, homero])).
padre(homero, hijos([bart, lisa, maggie])).
padre(clancy, hijos([marge, patty, selma])).

```

% madres

```

madre(mona, hijos([homero])).
madre(jacqueline, hijos([marge, patty, selma])).
madre(marge, hijos([bart, lisa, maggie])).
madre(selma, hijos([ling])).

```

%Verifica si una persona es padre/madre de otra

```

es_padre_de(Padre, Hijo) :-
    (padre(Padre, hijos(Hijos)) ; madre(Padre, hijos(Hijos))),
    member(Hijo, Hijos).

```

% Abuelos

```

abuelos(Abu, Nieto) :-
    es_padre_de(Abu, Hijo),
    es_padre_de(Hijo, Nieto).

```

Consultas

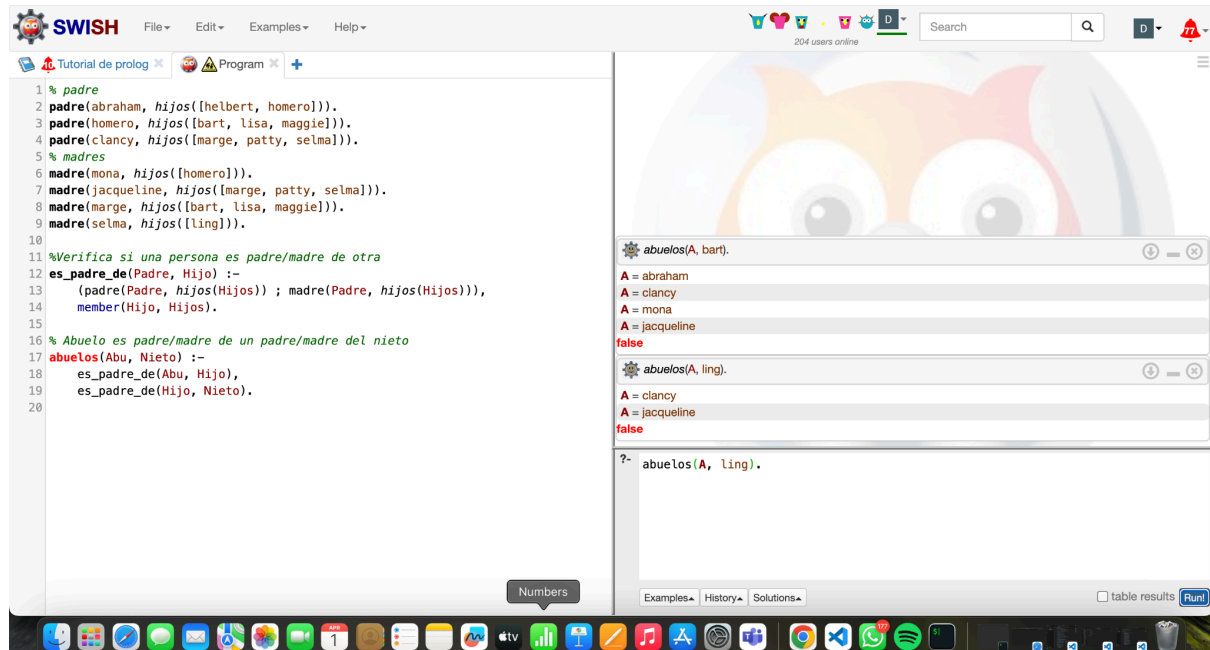
```

?- abuelos(A, bart).
?- abuelos(A, lisa).

```

?- abuelos(A, maggie).

?- abuelos(A, ling).



Link de GitHub:

<https://github.com/DanielCuellar7/Programacion-utp/tree/main/prolog>

Soporte de trabajo en equipo: Llamada en Meet dónde trabajamos ambos integrantes, probando, estudiando y aportando para realizar la tarea.

