

## Competencia

Identificar los conceptos básicos necesarios para implementar las primeras secuencias de programación lógica utilizando el lenguaje de programación Prolog.

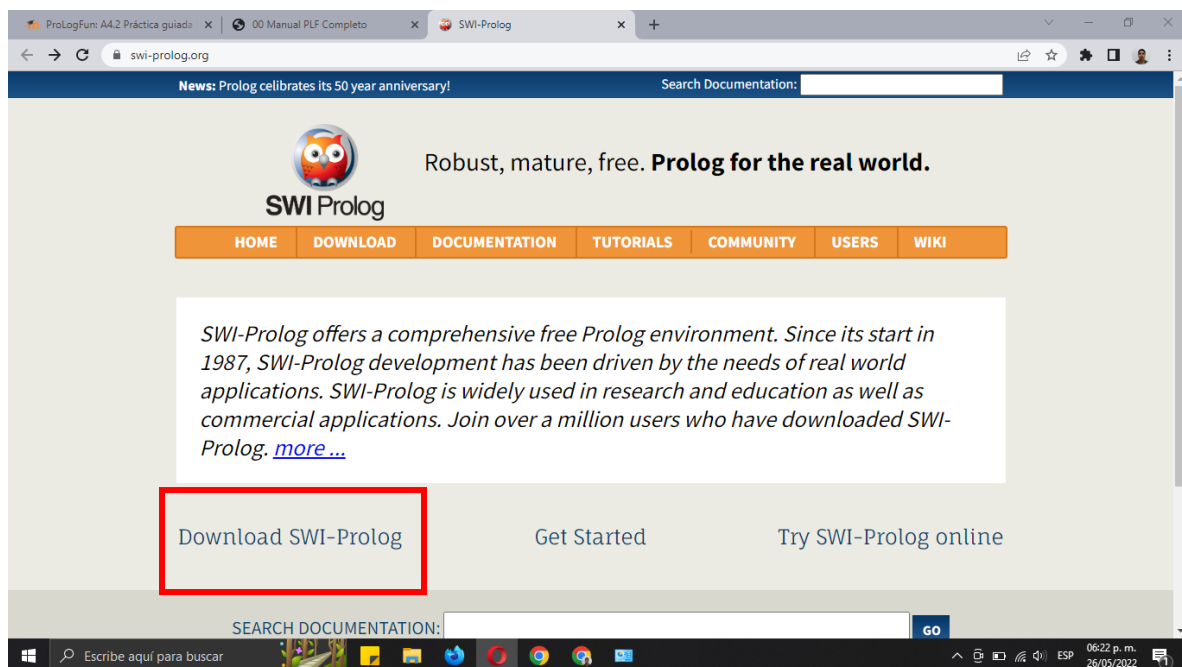
## Introducción

La programación lógica estudia el uso de la lógica para el planteamiento de problemas y el control sobre las reglas de inferencia para alcanzar la solución automática. Se puede ver como una deducción controlada. Lógica de hechos y reglas para representar conocimiento.

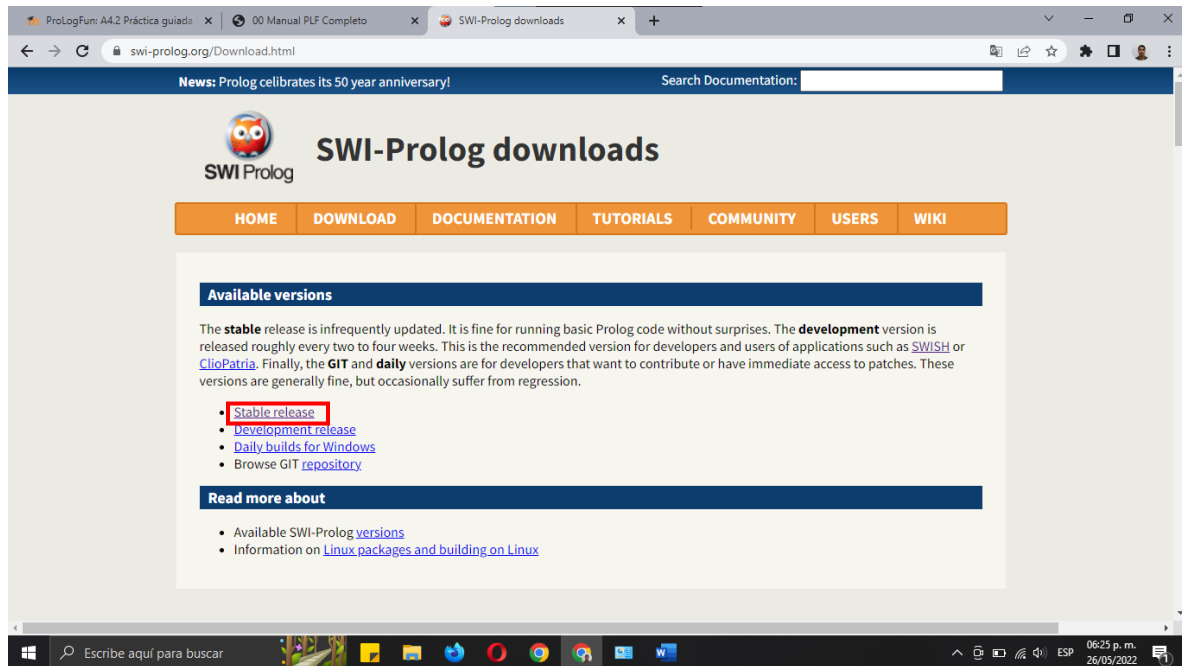
En este documento vamos a ver desde la instalación paso por paso del SWI-Prolog, como programar en este y también vamos a realizar varias actividades en las cuales vamos a necesitar crear nuevas relaciones.

## Instalación de Prolog.

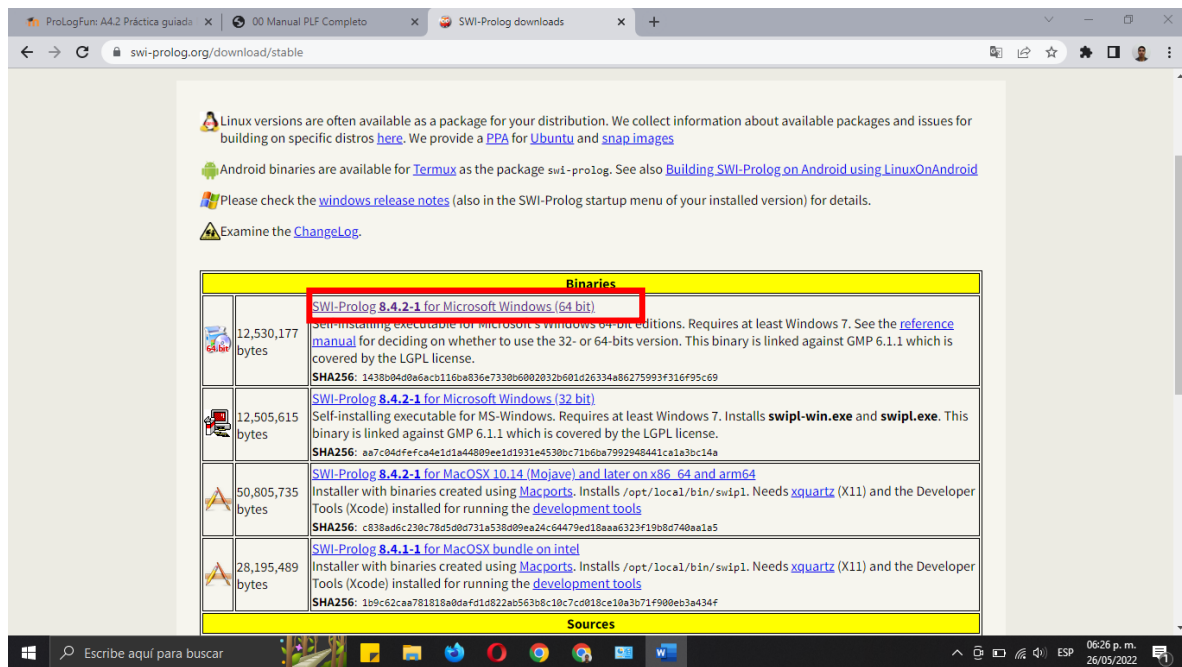
Entramos al link: <https://www.swi-prolog.org/> y le damos a Download SWI-Prolog.



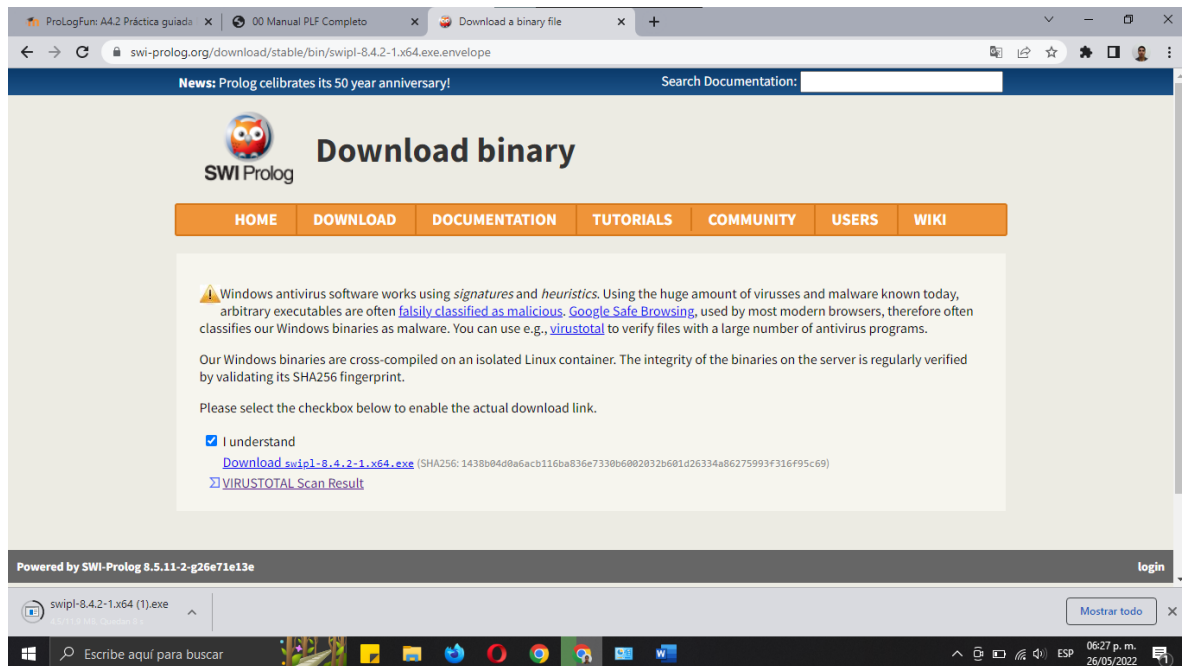
Después ingresamos a la opción Stable release



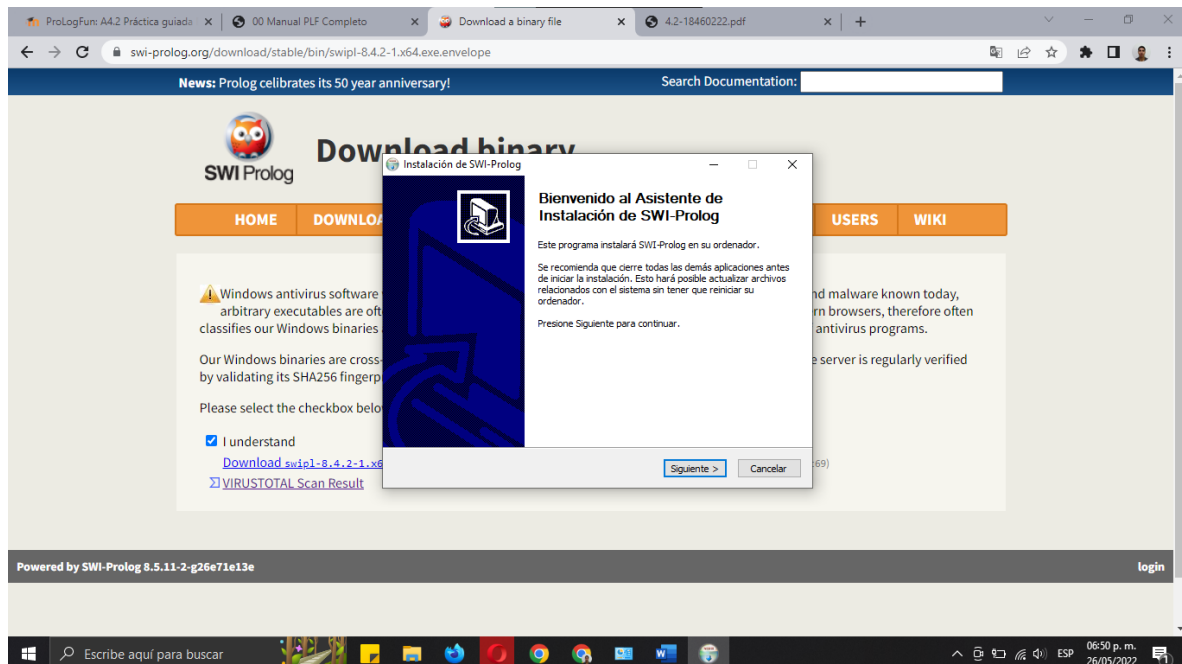
Ahora, en mi caso descargare la versión de 64 bits para Windows.



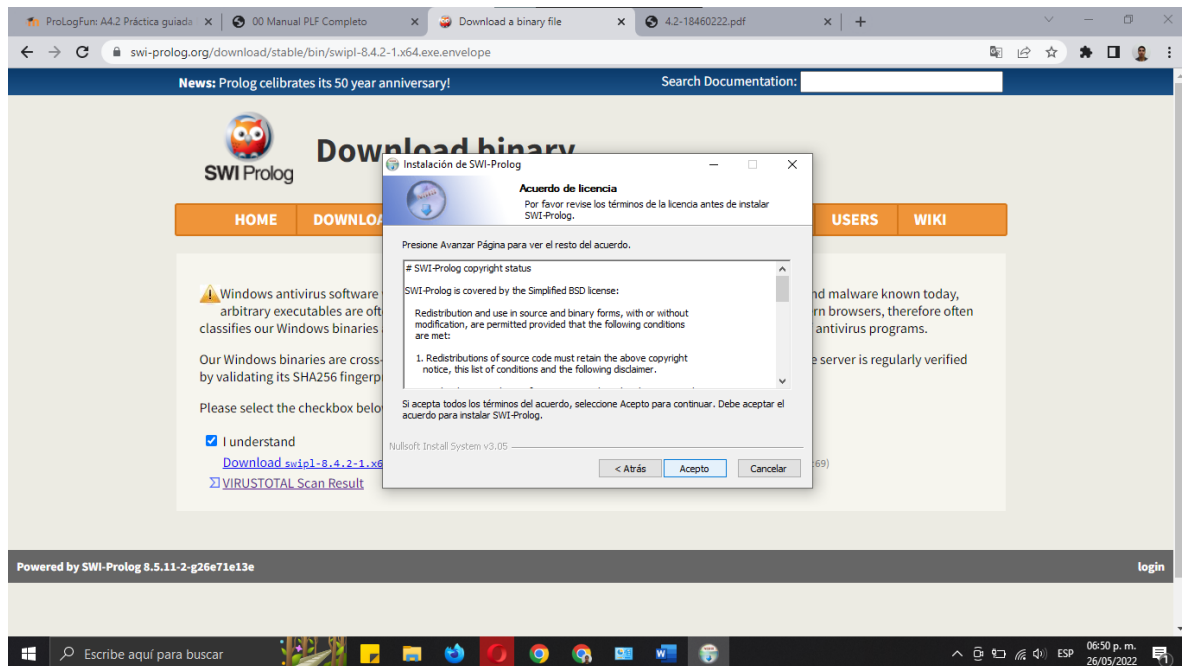
Una vez clickiamos la opción a descargar nos aparecerá un checkbox en el cual una vez crackeado nos habilitará la opción para descargar el swi-prolog.



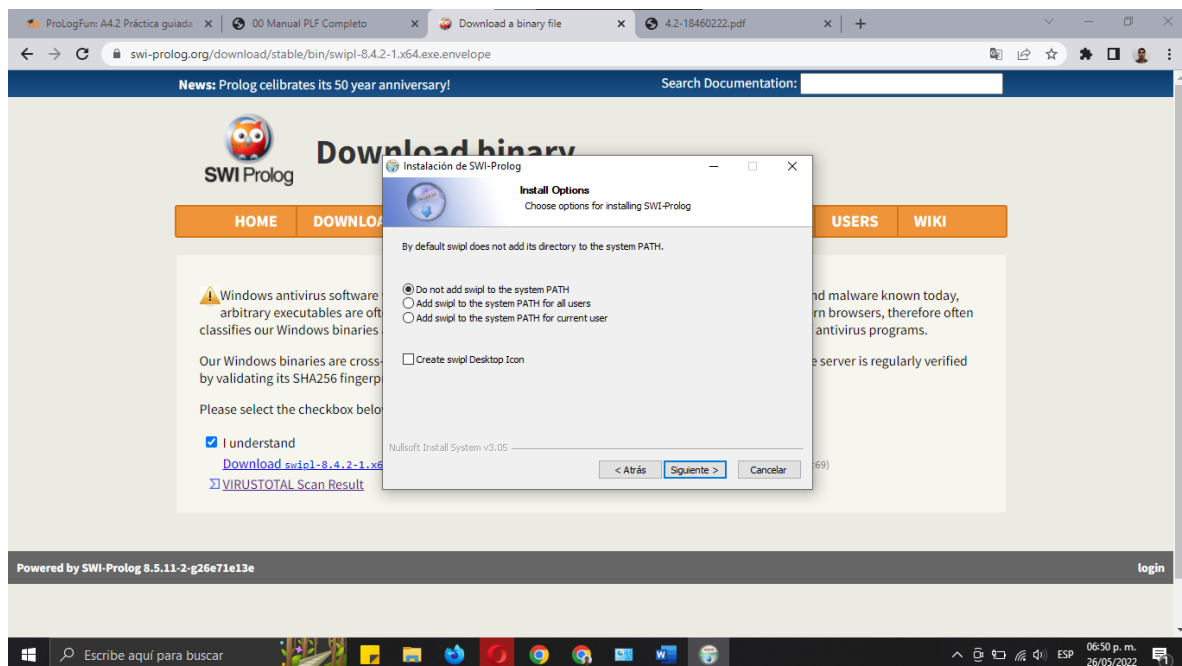
Ya que hemos descargado el swi-prolog toca instalarlo, para esto iremos a donde se descargo el archivo y le daremos doble click, una vez hecho esto nos aparecerá la siguiente ventana, simplemente le damos en siguiente.



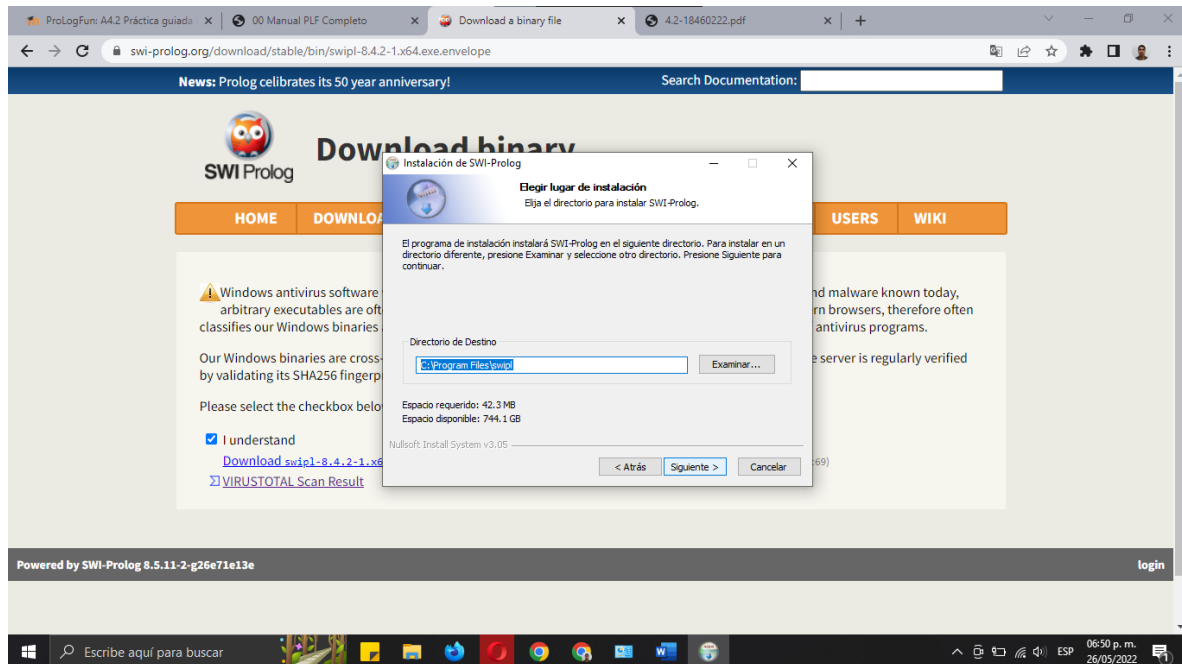
Después, nos aparecerá la ventana con los términos y condiciones, aquí luego de leerlos y aceptarlos le damos en acepto.



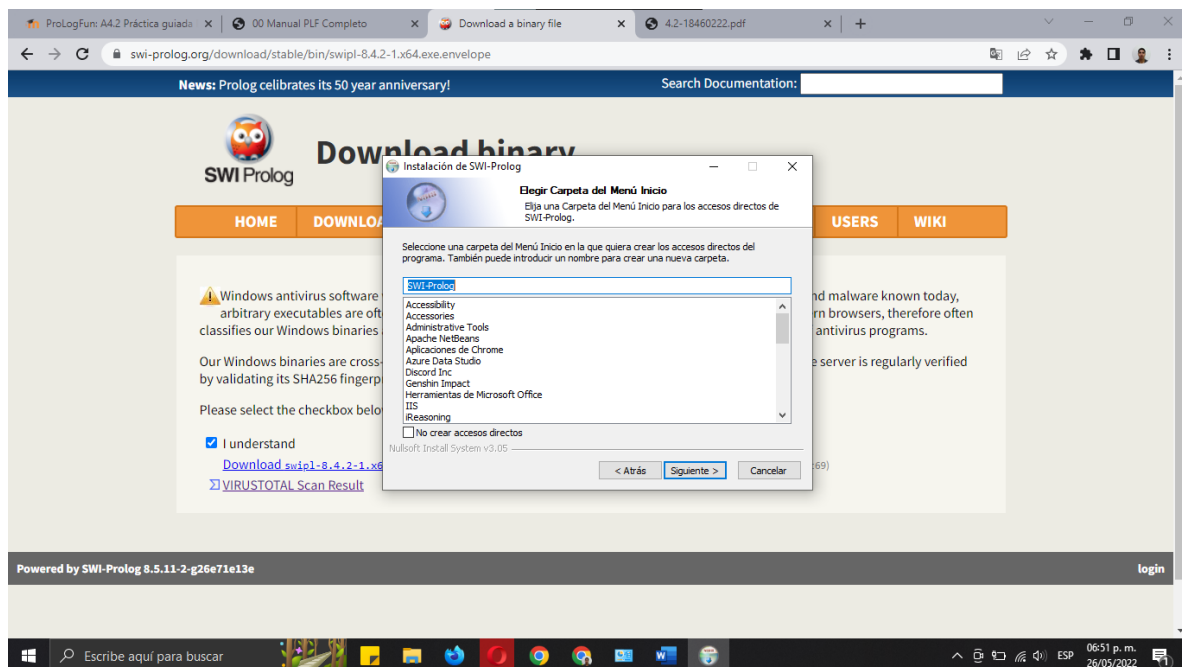
Aquí vamos a ver que puso Luis :v

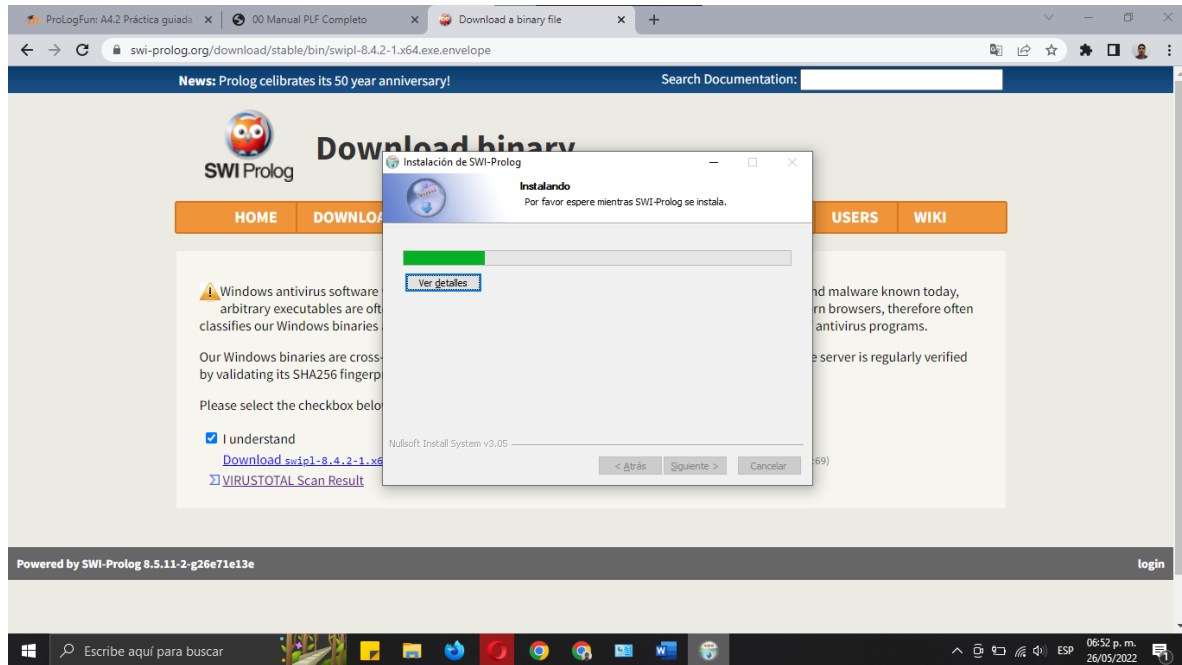


Ingresamos la ubicación donde queremos instalar el SWI-Prolog, una vez hecho esto daremos en siguiente.



En la siguiente ventana seleccionaremos una carpeta donde del menú de inicio donde se crearán los accesos directos del programa.






## Creación del archivo baseconocer.pl.

```
baseconocer.pl
File Edit Browse Compile Prolog Pce Help
baseconocer.pl
esHombre(jose).
esHombre(carlos).
esHombre(juan).
esHombre(omar).
esHombre(jorge).
esMujer(eva).
esMujer(rosa).
esMujer(maria).
esMujer(esther).
papa(jose,carlos).
papa(jose,juan).
papa(jose,rosa).
papa(juan,omar).
papa(juan,jorge).
papa(omar,maria).
mama(eva,carlos).
mama(eva,juan).
mama(eva,rosa).
mama(eva,omar).
mama(eva,jorge).
mama(rosa,esther).
papa/2: (not loaded) unreferenced
Line: 12
```

Carga del archivo baseconocer.pl y realizar las consultas.



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?- [baseconocer].
true.

?- esHombre(eva).
false.

?- esHombre(juan).
true.

?- papa(jose,carlos).
true.

?- mama(eva,esther).
false.

?- papa(X,carlos).
X = jose.

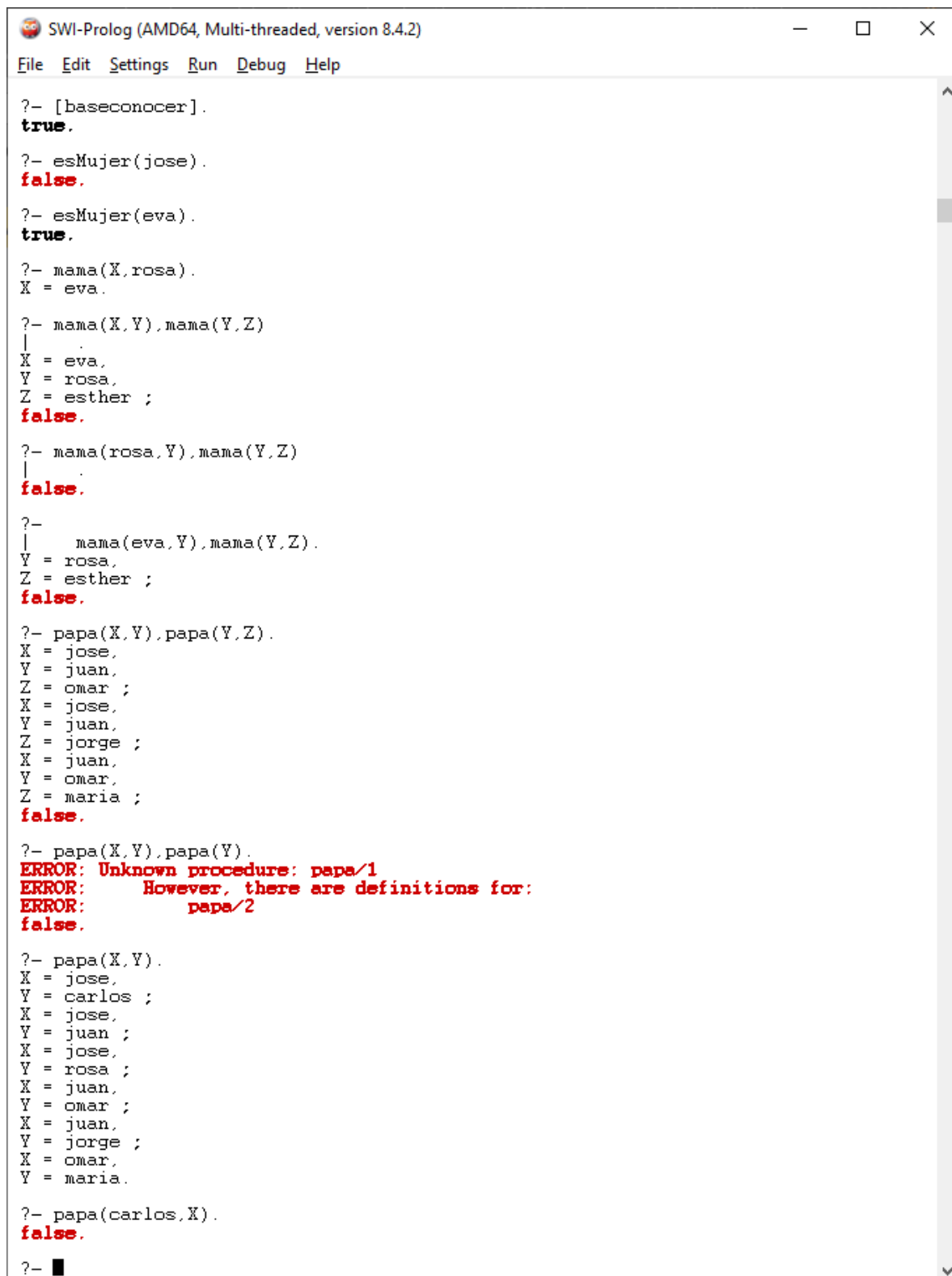
?- papa(jose,X).
X = carlos ;
X = juan ;
X = rosa.

?- papa(jose,juan),papa(juan,omar).
true.

?- papa(jose,X),papa(X,Y).
X = juan,
Y = omar ;
X = juan,
Y = jorge ;
false.

?-
```

## Actividad 1:



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

?- [baseconocer].
true.

?- esMujer(jose).
false.

?- esMujer(eva).
true.

?- mama(X,rosa).
X = eva.

?- mama(X,Y),mama(Y,Z)
|
| .
X = eva,
Y = rosa,
Z = esther ;
false.

?- mama(rosa,Y),mama(Y,Z)
|
| .
false.

?-
|
|   mama(eva,Y),mama(Y,Z) .
Y = rosa,
Z = esther ;
false.

?- papa(X,Y),papa(Y,Z).
X = jose,
Y = juan,
Z = omar ;
X = jose,
Y = juan,
Z = jorge ;
X = juan,
Y = omar,
Z = maria ;
false.

?- papa(X,Y),papa(Y).
ERROR: Unknown procedure: papa/1
ERROR:   However, there are definitions for:
ERROR:       papa/2
false.

?- papa(X,Y).
X = jose,
Y = carlos ;
X = jose,
Y = juan ;
X = jose,
Y = rosa ;
X = juan,
Y = omar ;
X = juan,
Y = jorge ;
X = omar,
Y = maria.

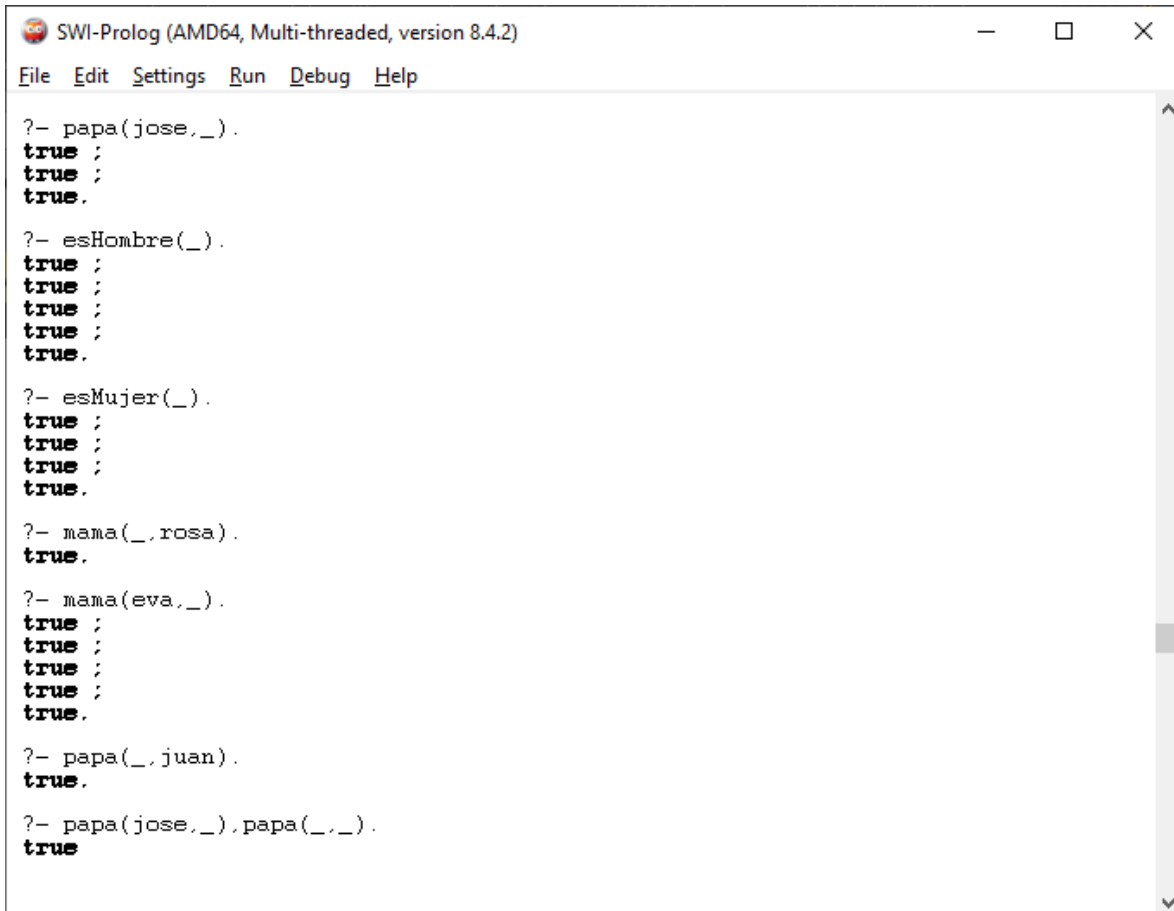
?- papa(carlos,X).
false.

?- █
```



## Actividad 2:

La variable anónima es utilizada cuando se requiere saber si existe un objeto que cumpla un objetivo. Se puede considerar como una variable libre, sin ningún cuantificador, al aplicarla nos regresará una respuesta simple de si o no. Como por ejemplo, está el que nos devuelve si un sujeto es padre de alguien y de ser el caso cuantas veces cumple esta condición, además de otros ejemplos como si alguien es mujer, hombre.



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

?- papa(jose,_).
true ;
true ;
true.

?- esHombre(_).
true ;
true ;
true ;
true ;
true.

?- esMujer(_).
true ;
true ;
true ;
true.

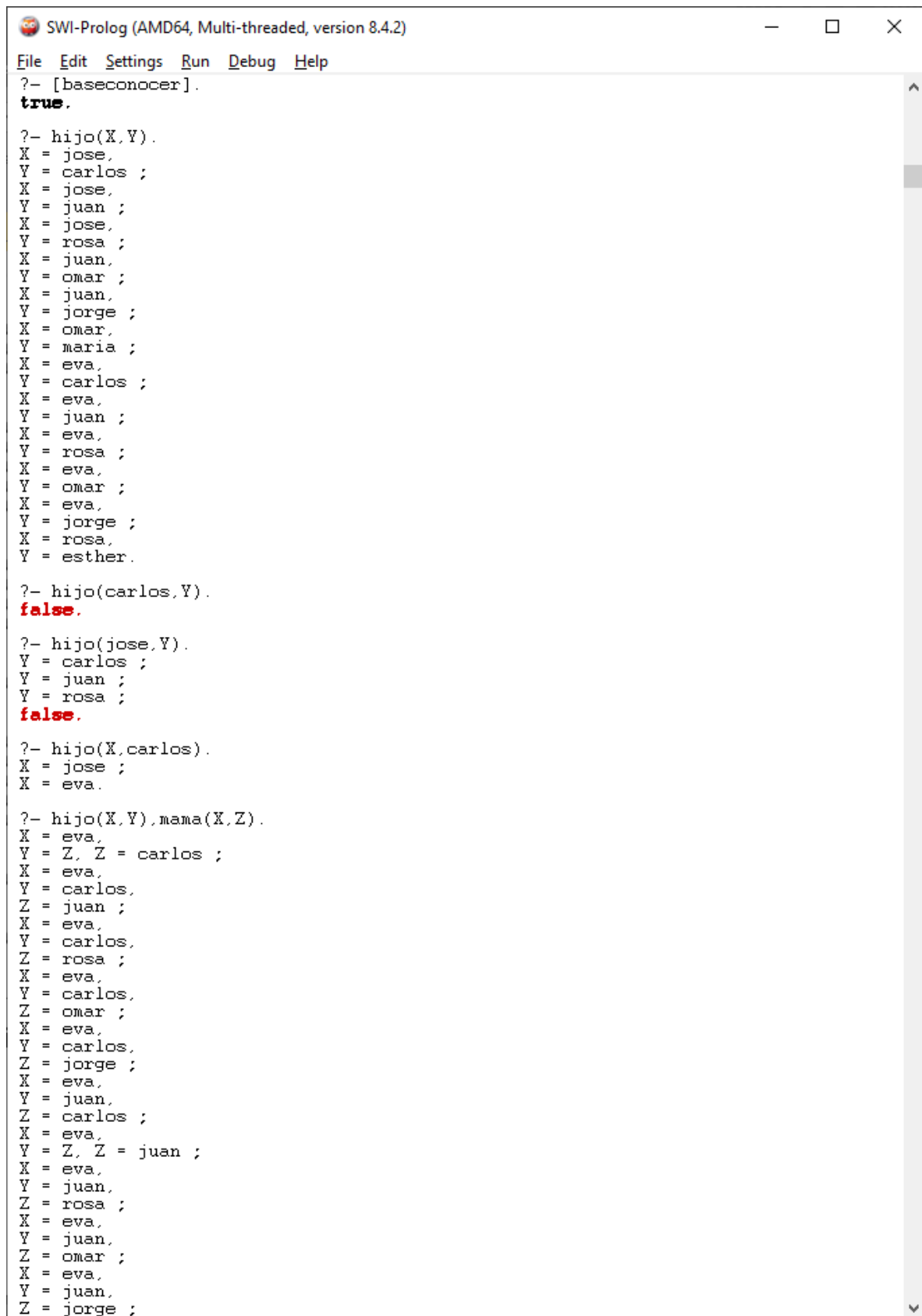
?- mama(_,rosa).
true.

?- mama(eva,_).
true ;
true ;
true ;
true ;
true.

?- papa(_,juan).
true.

?- papa(jose,_),papa(_,_).
true
```

## Actividad 3:



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
?- [baseconocer].
true.

?- hijo(X,Y).
X = jose,
Y = carlos ;
X = jose,
Y = juan ;
X = jose,
Y = rosa ;
X = juan,
Y = omar ;
X = juan,
Y = jorge ;
X = omar,
Y = maria ;
X = eva,
Y = carlos ;
X = eva,
Y = juan ;
X = eva,
Y = rosa ;
X = eva,
Y = omar ;
X = eva,
Y = jorge ;
X = rosa,
Y = esther.

?- hijo(carlos,Y).
false.

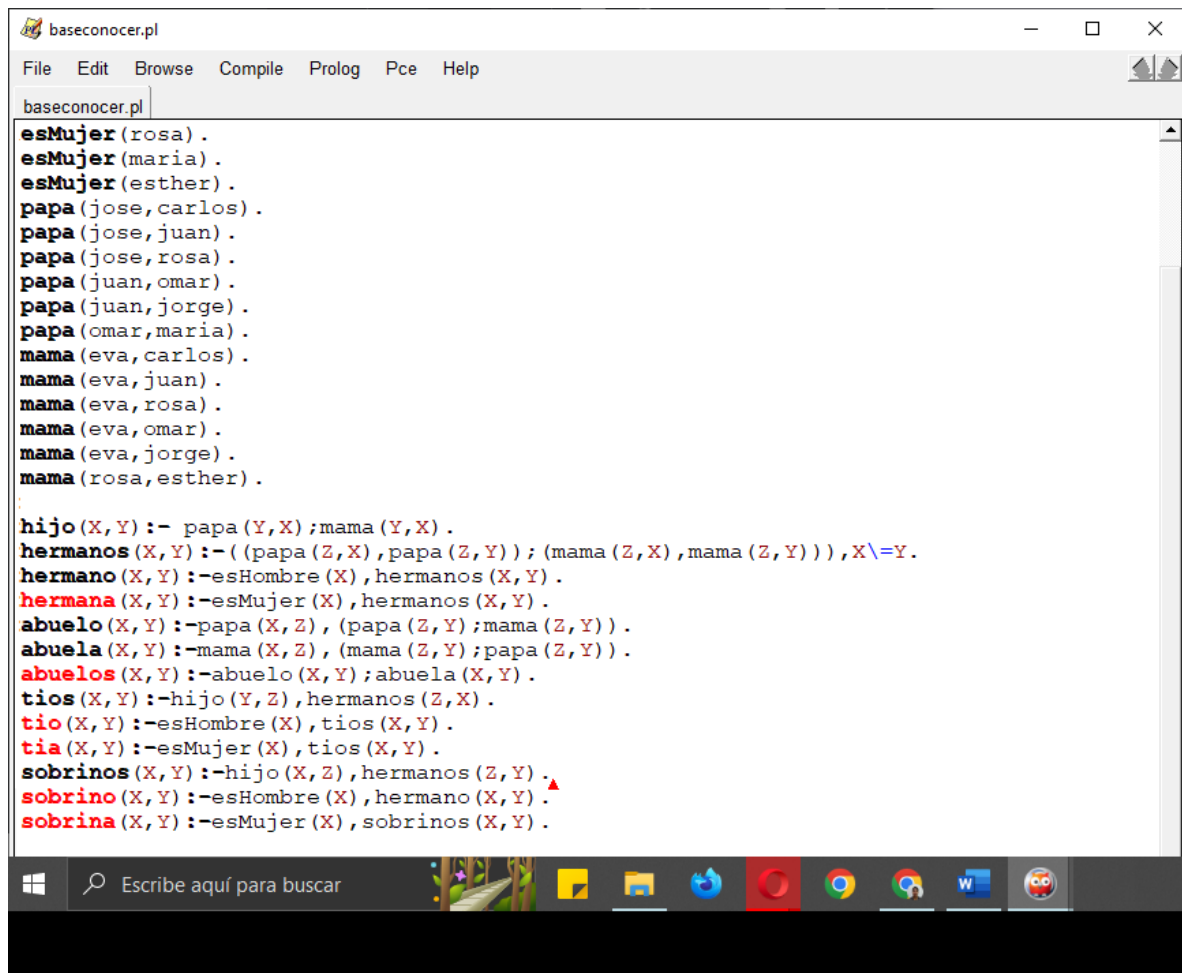
?- hijo(jose,Y).
Y = carlos ;
Y = juan ;
Y = rosa ;
false.

?- hijo(X,carlos).
X = jose ;
X = eva.

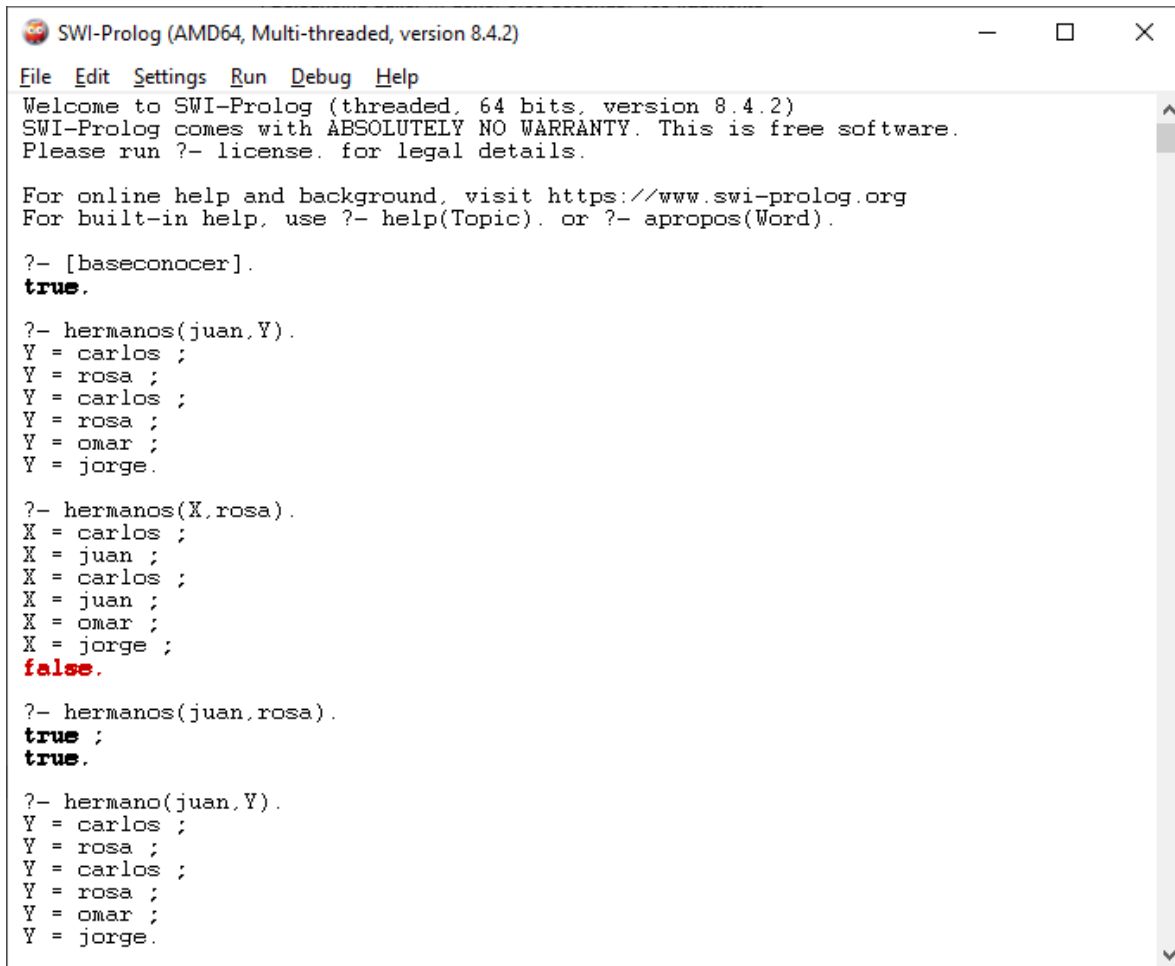
?- hijo(X,Y),mama(X,Z).
X = eva,
Y = Z, Z = carlos ;
X = eva,
Y = carlos,
Z = juan ;
X = eva,
Y = carlos,
Z = rosa ;
X = eva,
Y = carlos,
Z = omar ;
X = eva,
Y = carlos,
Z = jorge ;
X = eva,
Y = juan,
Z = carlos ;
X = eva,
Y = Z, Z = juan ;
X = eva,
Y = juan,
Z = rosa ;
X = eva,
Y = juan,
Z = omar ;
X = eva,
Y = juan,
Z = jorge ;
```

## Actividad 4:

Para este ejemplo se definieron las relaciones de hermanos, hermana, hermano, abuelos, abuela, abuelo, tíos, tía, tío, sobrinos, sobrino y sobrina, las cuales son relaciones que ya podíamos obtener con la información previa y que se podían demostrar mediante pruebas.



```
baseconocer.pl
File Edit Browse Compile Prolog Pce Help
baseconocer.pl
esMujer(rosa).
esMujer(maria).
esMujer(esther).
papa(jose, carlos).
papa(jose, juan).
papa(jose, rosa).
papa(juan, omar).
papa(juan, jorge).
papa(omar, maria).
mama(eva, carlos).
mama(eva, juan).
mama(eva, rosa).
mama(eva, omar).
mama(eva, jorge).
mama(rosa, esther).
.
hijo(X, Y) :- papa(Y, X) ; mama(Y, X).
hermanos(X, Y) :- (papa(Z, X), papa(Z, Y)) ; (mama(Z, X), mama(Z, Y)), X \= Y.
hermano(X, Y) :- esHombre(X), hermanos(X, Y).
hermana(X, Y) :- esMujer(X), hermanos(X, Y).
abuelo(X, Y) :- papa(X, Z), (papa(Z, Y) ; mama(Z, Y)).
abuela(X, Y) :- mama(X, Z), (mama(Z, Y) ; papa(Z, Y)).
abuelos(X, Y) :- abuelo(X, Y) ; abuela(X, Y).
tios(X, Y) :- hijo(Y, Z), hermanos(Z, X).
tio(X, Y) :- esHombre(X), tios(X, Y).
tia(X, Y) :- esMujer(X), tios(X, Y).
sobrinos(X, Y) :- hijo(X, Z), hermanos(Z, Y).
sobrino(X, Y) :- esHombre(X), hermano(X, Y).
sobrina(X, Y) :- esMujer(X), sobrinos(X, Y).
```

A screenshot of the SWI-Prolog (AMD64, Multi-threaded, version 8.4.2) window. The window has a title bar with standard Windows controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Settings', 'Run', 'Debug', and 'Help'. The main text area contains the following text:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

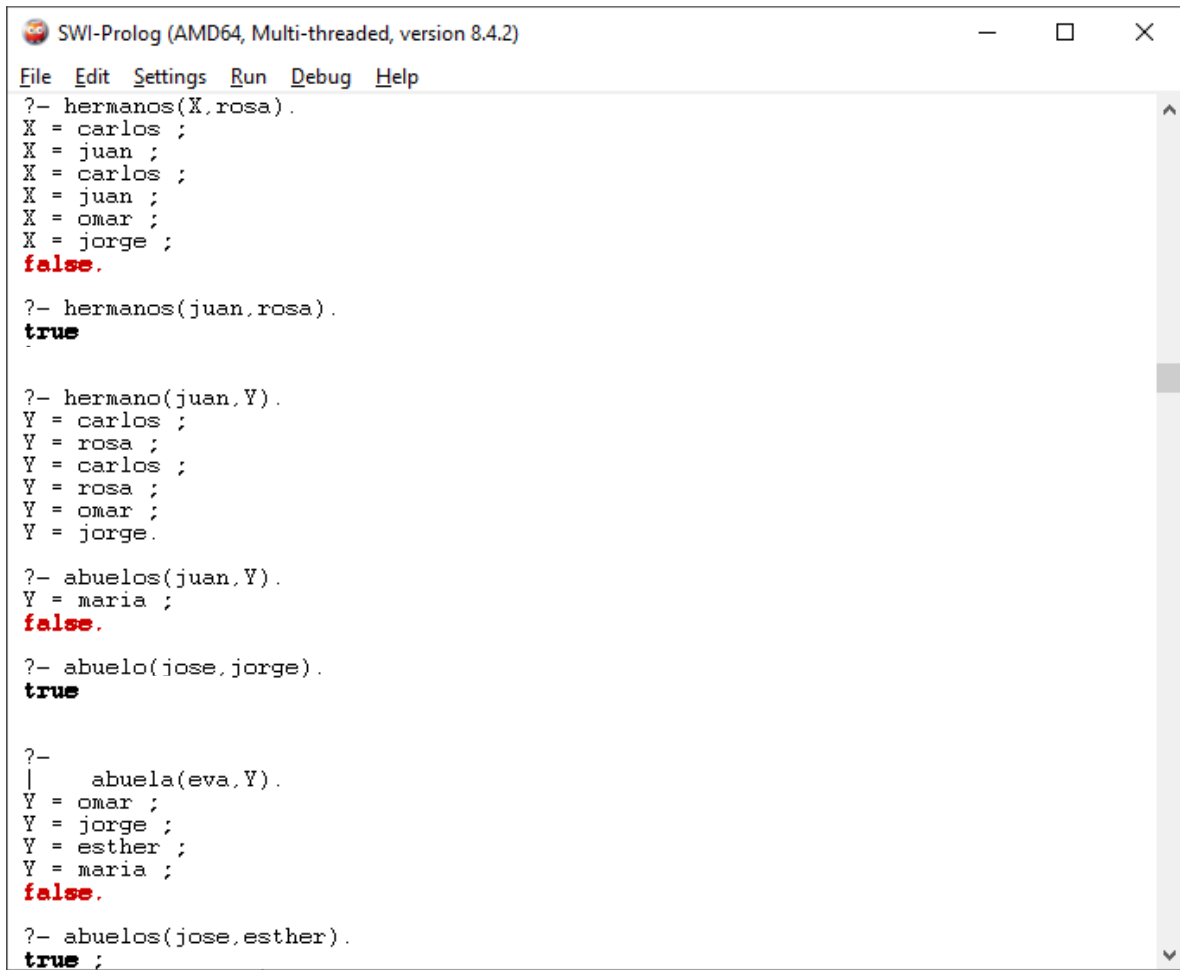
?- [baseconocer].
true.

?- hermanos(juan,Y).
Y = carlos ;
Y = rosa ;
Y = carlos ;
Y = rosa ;
Y = omar ;
Y = jorge.

?- hermanos(X,rosa).
X = carlos ;
X = juan ;
X = carlos ;
X = juan ;
X = omar ;
X = jorge ;
false.

?- hermanos(juan,rosa).
true ;
true.

?- hermano(juan,Y).
Y = carlos ;
Y = rosa ;
Y = carlos ;
Y = rosa ;
Y = omar ;
Y = jorge.
```



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
?- hermanos(X,rosa).
X = carlos ;
X = juan ;
X = carlos ;
X = juan ;
X = omar ;
X = jorge ;
false.

?- hermanos(juan,rosa).
true

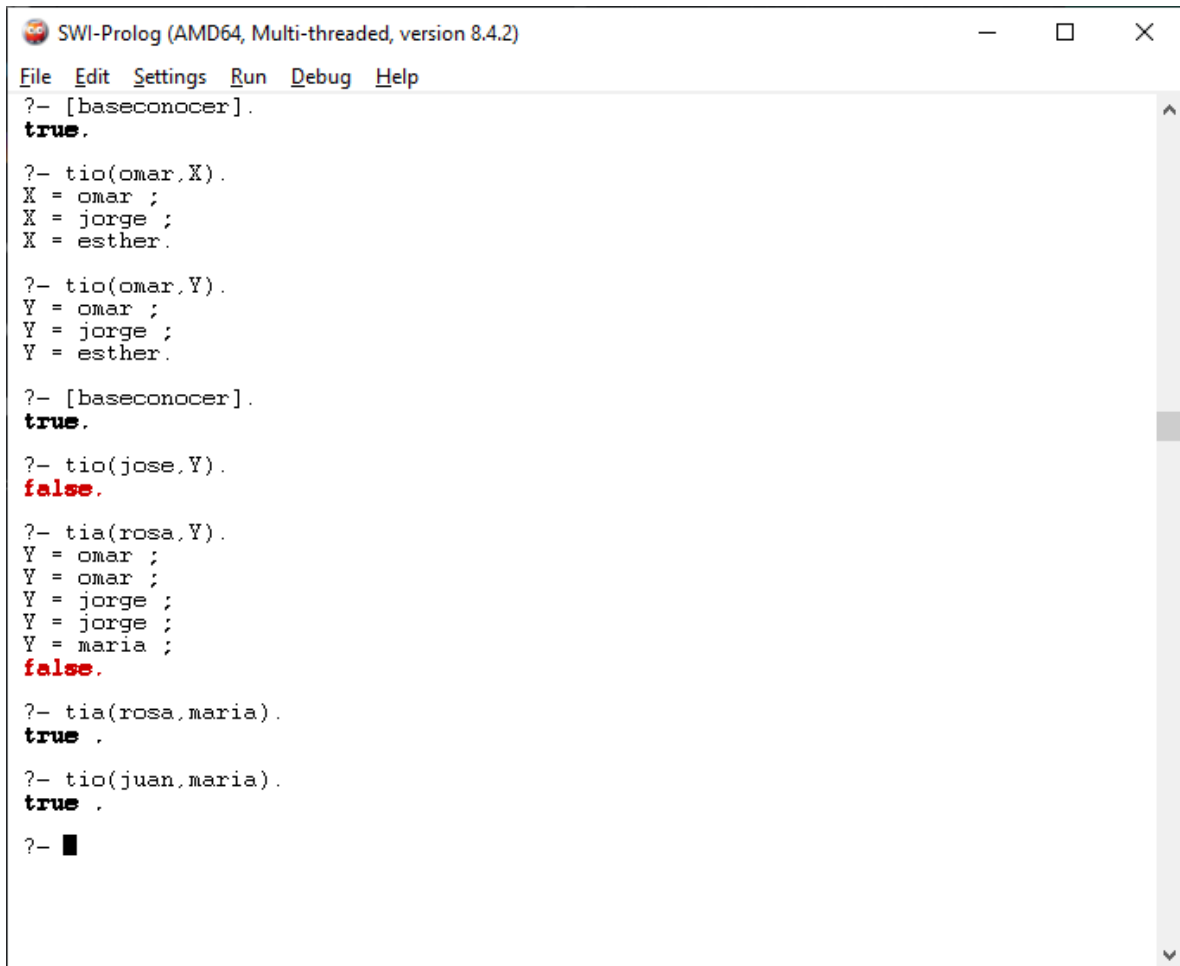
?- hermano(juan,Y).
Y = carlos ;
Y = rosa ;
Y = carlos ;
Y = rosa ;
Y = omar ;
Y = jorge.

?- abuelos(juan,Y).
Y = maria ;
false.

?- abuelo(jose,jorge).
true

?-
|   abuela(eva,Y).
Y = omar ;
Y = jorge ;
Y = esther ;
Y = maria ;
false.

?- abuelos(jose,esther).
true ;
```



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
?- [baseconocer].
true.

?- tio(omar,X).
X = omar ;
X = jorge ;
X = esther.

?- tio(omar,Y).
Y = omar ;
Y = jorge ;
Y = esther.

?- [baseconocer].
true.

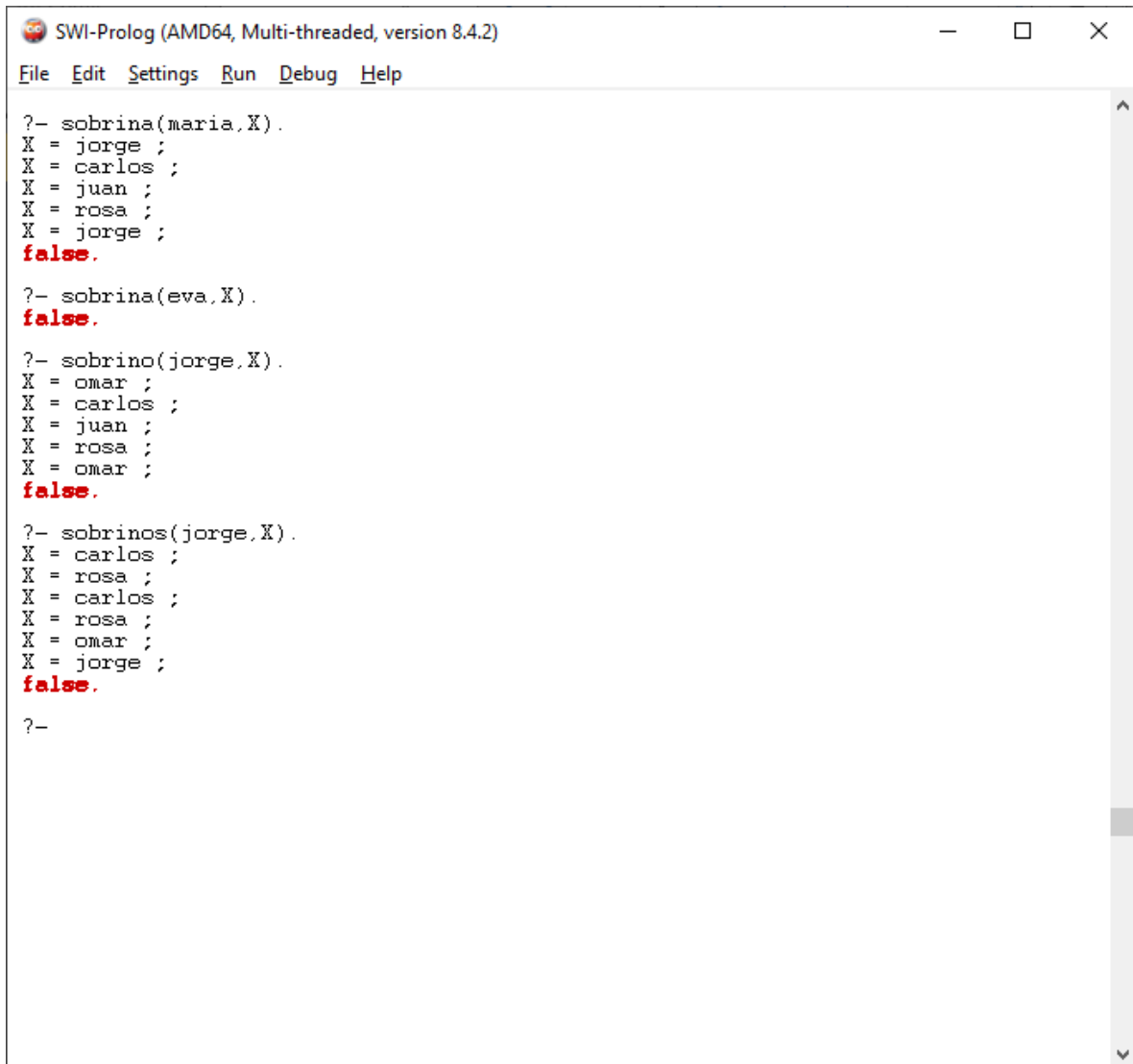
?- tio(jose,Y).
false.

?- tia(rosa,Y).
Y = omar ;
Y = omar ;
Y = jorge ;
Y = jorge ;
Y = maria ;
false.

?- tia(rosa,maria).
true .

?- tio(juan,maria).
true .

?- █
```



```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help

?- sobrina(maria,X).
X = jorge ;
X = carlos ;
X = juan ;
X = rosa ;
X = jorge ;
false.

?- sobrina(eva,X).
false.

?- sobrino(jorge,X).
X = omar ;
X = carlos ;
X = juan ;
X = rosa ;
X = omar ;
false.

?- sobrinos(jorge,X).
X = carlos ;
X = rosa ;
X = carlos ;
X = rosa ;
X = omar ;
X = jorge ;
false.

?-
```

## Conclusión

Como pudimos observar la programación en Prolog es bastante sencilla, ya que en a diferencia de otros lenguajes como Java, Python, c++, entre otros, no busca resolver un problema. Además, Gracias a todo lo que vi en la unidad pasada me sirvió como base apara entender un poco mejor las cláusulas, tanto en como leerlas, formarlas y entenderlas.

## Referencias

[1]] M. M. E. M. Durán, «Moodle,» [En línea]. Available: [https://moodle.colima.tecnm.mx/pluginfile.php/19727/mod\\_assign/intro/P08%20Principios%20basicos%20PL-Prolog.pdf](https://moodle.colima.tecnm.mx/pluginfile.php/19727/mod_assign/intro/P08%20Principios%20basicos%20PL-Prolog.pdf). [Último acceso: 26 mayo 2022].

[2] J. C. Cubero y F. Berzal, 2011. [En línea]. Available: <https://elvex.ugr.es/decsai/intelligent/workbook/ai/PROLOG.pdf>. [Último acceso: 26 mayo 2022].