



Nombre de la práctica:	PROGRAMACIÓN DE ENUNCIADOS DE LÓGICA PROPOSICIONAL Y DE LÓGICA DE PREDICADOS (PROLOG)
No. de práctica:	1
Alumna:	LAURA KAREN NAVA GÁLVEZ

INTRODUCCIÓN

La lógica predictiva es un sistema deductivo basado en un lenguaje lógico matemático formal. Incluye proposiciones lógicas, predicados y cuantificadores.

Más expresiva de la Lógica proposicional.

¿Qué se afirma? (predicado o relación)

¿De quién se afirma? (objeto)

Es un sistema formal diseñado para estudiar la inferencia en los lenguajes de primer orden.¹ Los lenguajes de primer orden son, a su vez, lenguajes formales con cuantificadores que alcanzan sólo a variables de individuo, y con predicados y funciones cuyos argumentos son sólo constantes o variables de individuo. La lógica de primer orden tiene el poder expresivo suficiente para definir a prácticamente todas las matemáticas.

Hechos

Los hechos son utilizados para declarar verdades desde el principio de la ejecución de un programa, y el conjunto de estos hechos se denomina como base de conocimientos

Expresión atómica (declaración, cláusula o proposición) que se formula de la forma $\text{P}(\text{t}_{\{1\}}, \dots, \text{t}_{\{n\}})$, indicando que se verifica el predicado P sobre los objetos $\text{t}_{\{1\}}, \dots, \text{t}_{\{n\}}$

Reglas

Conjunto de proposiciones lógicas escritas como cláusulas de Horn que permiten inferir el valor de verdad de nuevas proposiciones, permitiendo ampliar la base de conocimientos, a la vez que son utilizadas para definir el dominio del problema.

Consultas



Proposición construida con el propósito de ser demostrada/desmentida o de encontrar el conjunto de valores que la convierten en verdadera. En las consultas se especifica el problema a resolver. Partiendo de que los humanos son mortales y de que Pepito es humano (ejemplos anteriores), deducimos que Pepito es mortal.

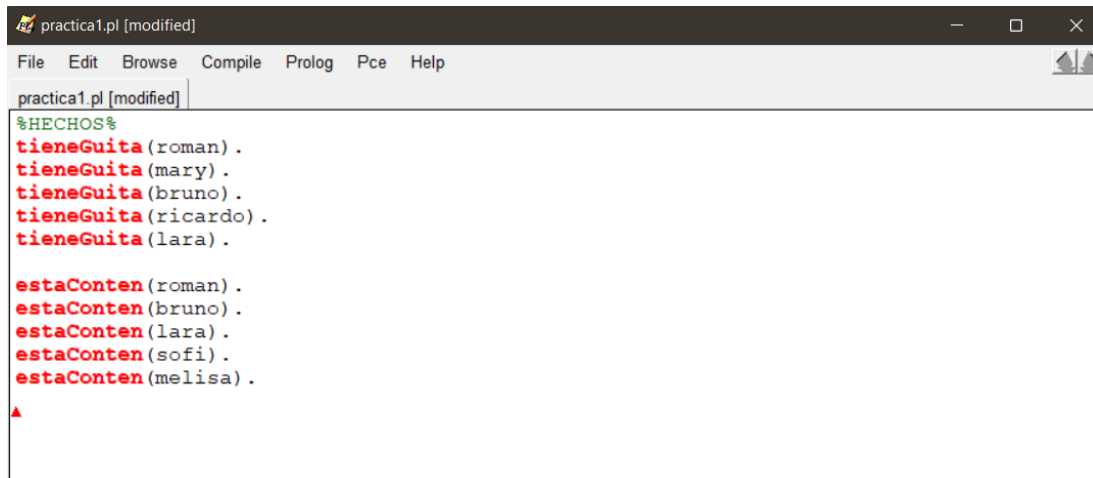
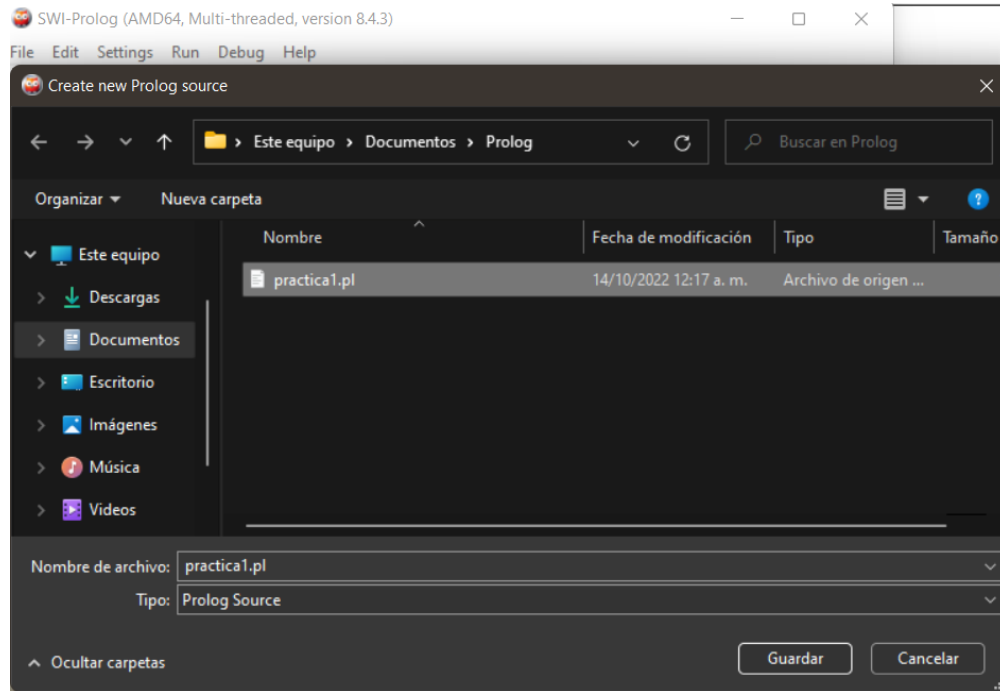
Recursión

La recursividad puede ser tratada de una manera más eficaz si se piensa en que hace el algoritmo recursivo que se piensa aplicar, en vez de como hacerlo. Para esto se usará la modularidad, la cual se basa en separar el problema en otros más pequeños y hallar la solución a estos para luego unirlos, como es usual en la programación lógica. (Microsoft® Word 2013, 2017)

OBJETIVO

- Aprender a utilizar prolog con unos breves ejercicios de predicados.
- Identificar las partes de un programa realizado en prolog.
- Comprender el funcionamiento del uso de predicados en prolog.

DESARROLLO DE LA ACTIVIDAD PRÁCTICA



```
practica1.pl
File Edit Browse Compile Prolog Pce Help
practica1.pl
%HECHOS%
tieneGuita(roman).
tieneGuita(mary).
tieneGuita(bruno).
tieneGuita(ricardo).
tieneGuita(lara).

estaConten(roman).
estaConten(bruno).
estaConten(lara).
estaConten(sofi).
estaConten(melisa).

%REGLAS%
escuMusic(X) :- estaConten(X).
tieneRad(X) :- escuMusic(X).
tocaGuita(X) :- escuMusic(X), tieneGuita(X).
```

RESULTADO (Consultas)

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
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).

?-
% c:/users/karen/documents/prolog/practical compiled 0.00 sec, -2 clauses
?- tieneGuita(X).
X = roman .

?- tieneGuita(lara).
true.

?- tieneGuita(sofi).
false.

?- tieneGuita(ricardo).
true.

?- estaConten(X).
X = roman .

?- estaConten(melisa).
true.

?- estaConten(bruno).
true.

?- estaConten(mary).
false.

?- estaConten(ricardo).
false.

?-
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help

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

?-
% c:/Users/karen/Documents/Prolog/practical.pl compiled 0.00 sec, 13 clauses
% c:/users/karen/documents/prolog/practical compiled 0.00 sec, -2 clauses
?- escuMusic(roman).
true.

?- escuMusic(ricardo).
false.

?- escuMusic(lara).
true.

?- tieneRad(roman).
true.

?- tieneRad(mary).
false.

?- tieneRad(sofi).
true.

?- tocaGuita(roman).
true.

?- tocaGuita(melisa).
false.

?- tocaGuita(sofi).
false.

?- tocaGuita(lara).
true.

. ?- █
```

DISCUSIÓN

Finalmente investigando y ocupando las herramientas proporcionadas por el docente se pudo lograr hacer el programa en prolog con sus distintos hechos y reglas para al final hacer las consultas necesarias, respondiendo ¿Está tocando la guitarra? ¿Quién está contento?

Al usar prolog por primera vez y tratar de entender la lógica de los predicados era un poco confuso como plasmar lo que se pedía, pero una vez indagando y poniendo en practica lo visto se comprendido de una mejor manera.



CUESTIONARIO

1.- De los siguientes enunciados, realiza la lógica de predicados con sus respectivas consultas:

Base de conocimiento:

Regla 1: Si está contento entonces escucha música.

Regla 2: Si tiene radio entonces escucha música.

Regla 3: Si escucha música y tiene una guitarra entonces toca la guitarra.

Hecho 1: Tiene una guitarra.

Hecho 2: Está contento.

Consultas:

> ¿Está tocando la guitarra?

> ¿Quién está contento?



REFERENCIAS

López Ostenero Fernando, García Serrano Ana María (2014), Teoría de los Lenguajes de Programación, Editorial Universitaria Ramón Areces, ISBN: 9788499611396.

Gortazar Bellas Francisco, Martínez Unanue Raquel, Fresno Fernández Victor, Lenguajes de Programación y Procesadores (2012), Editorial Universitaria Ramón Areces, ISBN: 9788499610702.

Llorens F. & Castel de Haro J. (2001), Prácticas de Lógica Prolog, Lógica de primer orden, lógica computacional y aplicación de lógica, Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante España, fecha de consulta: Agosto 2019, disponible en:

http://www.dccia.ua.es/logica/prolog/docs/prolog.pdfSy4VAwzYotm940cpMyQlgmsxnr bAltFes380x_BmksU1tZKzAk1wkzSGsjQ