

## LÓGICA 1º AÑO

Clase N.º 6: Lógica de Predicados.

Contenido: Cláusulas de Horn. Introducción a Prolog.

Hola, ¿Cómo están? ¡Bienvenidas y bienvenidos a la sexta clase! Como hemos visto, la Lógica de predicados permite la utilización de variables ya que podemos formalizar oraciones que hablan sobre individuos, sobre las propiedades de esos individuos, y sobre como esos individuos se relacionan entre sí. Es decir, se distingue:

- Qué se afirma (predicado o relación)
- De quién se afirma (objeto/persona)

Dentro del programa de la tecnicatura superior en Cs de datos e IA, se encuentra la Lógica ¿por qué?, ¿de qué manera la lógica opera en Cs de datos e IA? Para ello, les invitamos a hacer un breve recorrido por los distintos paradigmas de programación.









Programar viene a ser el proceso de crear un conjunto de instrucciones que le dicen a una computadora como realizar algún tipo de tarea. Pero no solo la acción de escribir un código para que la computadora o el software lo ejecute. Incluye, además, todas las tareas necesarias para que el código funcione correctamente y cumpla el objetivo para el cual se escribió. Para crear un programa y que la computadora lo interprete y ejecute, las instrucciones deben escribirse en un lenguaje de programación.

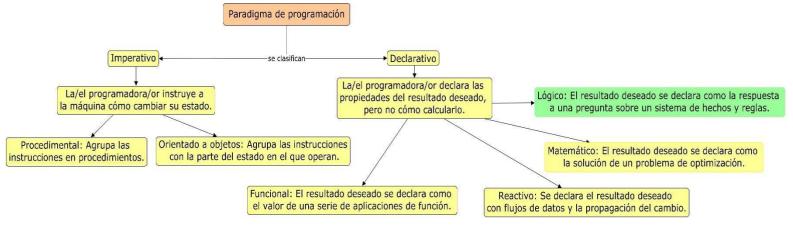
Un lenguaje de programación es un lenguaje con reglas gramaticales que le proporciona a una persona, en este caso la/el programadora/or, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático¹ (SI), de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina programa informático.

¹ Sistema que permite almacenar y procesar información; es el conjunto de partes interrelacionadas: hardware, software y personal informático. El hardware incluye computadoras o cualquier tipo de dispositivo electrónico, que consisten en procesadores, memoria, sistemas de almacenamiento externo, etc. El software incluye al sistema operativo, firmware (es el software que tiene directa interacción con el hardware, siendo así el encargado de controlarlo para ejecutar correctamente las instrucciones externas) y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos. Por último, el componente humano incluye al personal técnico que apoya y mantienen el sistema (analistas, programadores, operarios, etc.) y a los usuarios que lo utilizan.





Ahora bien, existen diversos lenguajes de programación, con sus características y estructuras, es por eso que los idiomas se pueden clasificar en múltiples paradigmas:



Como es de intuirse, abordaremos el paradigma de programación declarativo lógico puesto que la programación lógica gira en torno a la lógica de predicados. Ahora bien, el lenguaje de programación lógica por excelencia es Prolog pero... ¿qué es Prolog? **Prolog** (o PROLOG), proveniente del francés **PRO**grammation en **LOG**ique, es un lenguaje de programación lógica usado habitualmente en el campo de la Inteligencia artificial.

Los programas en Prolog se componen de **cláusulas de Horn** que constituyen reglas del tipo "modus ponendo ponens", es decir, "Si es verdad el antecedente, entonces es verdad el consecuente". No obstante, la forma de escribir las cláusulas de Horn es al contrario de lo habitual. Primero se escribe el consecuente y luego el antecedente. El antecedente puede ser una conjunción de condiciones que se denomina secuencia de objetivos. Cada objetivo se separa con una coma y puede considerarse similar a una instrucción o llamada a procedimiento de los lenguajes imperativos.





## Cláusula de Horn<sup>2</sup>:

Una fórmula lógica es una cláusula de Horn si es una cláusula (disyunción de literales<sup>3</sup>) con, como máximo, un literal positivo.

Esto es un ejemplo de una cláusula de Horn:

$$\neg p \lor \neg q \lor \cdots \lor \neg t \lor u$$

Una fórmula como esta también puede reescribirse de forma equivalente como una implicación:

$$(p \wedge q \wedge \cdots \wedge t) \rightarrow u$$

Veamos el siguiente ejemplo:

"A es mujer y B es padre de A entonces A es hija de B". En términos lógicos representa la siguiente implicación:

$$(mujer(A) \land padre(B,A)) \rightarrow hija(A,B)$$

En Prolog, como mencionamos anteriormente, se escribe primero el consecuente luego el antecedente, de esta forma tenemos que: "A es hija de B si A es mujer y B es padre de A". Aplicándolo a la sintaxis3 de una cláusula de Horn en Prolog, tiene el siguiente aspecto:

Donde el símbolo ":-" separa la cabeza (consecuente) del cuerpo (antecedente). Existen distintos tipos de cláusulas:

<sup>&</sup>lt;sup>3</sup> Valor constante formado por una secuencia de caracteres.



<sup>&</sup>lt;sup>2</sup> Se llaman así por el lógico Alfred Horn, el primero en señalar la importancia de estas cláusulas en 1951.



- ❖ Una cláusula sin cuerpo (sin antecedente) es un **Hecho:** Un hecho es un mecanismo para representar propiedades o relaciones de los objetos que se están representando. Los hechos declaran los valores que van a ser verdaderos o afirmativos para un predicado en todo el programa. Los hechos siguen la siguiente sintaxis: **nombre\_predicado(argumentos).** Los hechos se dividen en 2 tipos: propiedades y relaciones.
  - ➤ **Propiedades:** las propiedades se caracterizan por llevar un solo argumento y de esta manera expresan una propiedad de los objetos. Por ejemplo:

```
color(azul).
color(verde).
padre(juan).
padre(pablo).
```

➤ **Relaciones:** las relaciones se caracterizan por llevar más de un argumento y de esta manera expresan la relación entre varios objetos. Por ejemplo:

```
edad(juan, 30).
edad('pablo', 50).
padrede( juan , maria ).
padrede( pablo , juan ).
```

❖ Una cláusula con cabeza y cuerpo es llamada **Regla**: Cuando la verdad de un hecho depende de la verdad de otro hecho o de un grupo de hechos se usa una regla. Permiten establecer relaciones más elaboradas entre objetos donde se declaran las condiciones para que un predicado sea cierto, combinando hechos para dar el valor de verdad del predicado. La sintaxis base para una regla es la siguiente:

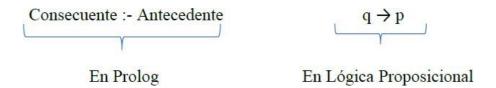
cabeza :- cuerpo





Donde la cabeza de una regla establece la relación con variables y el cuerpo es la conjunción o disyunción entre propiedades y relaciones con variables. Además, la forma como se debe leer esta sintaxis es de la siguiente manera: "La cabeza es verdad si el cuerpo es verdad". De esta manera se obtendrá el valor de verdad de la cabeza con el valor que se obtenga en el cuerpo, si el cuerpo resulta falso, la cabeza será falsa. Por ejemplo:

El símbolo ":-" cumple la función de "entonces". Recuerden que en Prolog se escribe primero el consecuente y luego el antecedente



Las reglas se pueden dividir en dos tipos:

➤ Conjunciones: Se usa una coma para separar los hechos del cuerpo de la regla. Este 'separador' se traduce como un AND lógico, concatenado cada hecho con un AND.

Por ejemplo:

```
hermano(X, Y) :- padre(Z), padrede(Z, X), padrede(Z, Y).
```

➤ **Disyunciones:** Se usa un **punto y coma** para separar los hechos del cuerpo de la regla. Este 'separador' se traduce como un **OR** lógico, concatenado cada hecho con un **OR**. Por ejemplo:

```
familiarde(A,B) :- padrede(A,B); hijode(A,B); hermanode(A,B).
```





- ❖ Una cláusula sin cabeza es una Pregunta o Consulta: Las consultas, llamadas comandos, comienzan por "?-". Es el mecanismo para extraer conocimiento del programa, en el que la consulta se deduce de: propiedades + relaciones + reglas. Ante una consulta, Prolog intenta hacer un matching sobre la base de conocimiento:
  - Mismo predicado
  - Mismo número de argumentos
  - Mismos argumentos: términos / variables

```
Hecho:

1 amigos(pedro, antonio).

Consulta:

= ?- amigos(pedro, antonio).
```

Ahora bien, para responder a las preguntas o consultas formuladas por quien programe, Prolog consulta una **base de conocimiento**. Esta base conocimiento representa el programa como tal, programa que se compone únicamente de **las propiedades, las relaciones y las reglas**, que, con el uso de la lógica, expresan el conocimiento deseado por el programa.

Nuestro punto de partida para empezar a utilizar y armar bases de conocimientos, será interpretar críticamente las propiedades, relaciones y reglas de distintas bases de conocimientos para luego realizarle preguntas al programa y razonar las respuestas. ¡Comencemos!







Copia y pega la siguiente base de conocimiento, que establece un árbol genealógico, en el simulador de Prolog:

```
hombre (jorge).
hombre (roberto).
hombre (pedro).
hombre (diego).
hombre (vicente).
mujer(elena).
mujer(victoria).
mujer (angelica).
mujer(sofia).
mujer(carolina).
es_padre(jorge, roberto).
es padre(jorge, victoria).
es padre (roberto, pedro).
es padre (roberto, sofia).
es padre (diego, vicente).
es_padre(diego, carolina).
es_madre(elena, roberto).
es_madre(elena, victoria).
es madre (angelica, pedro).
es madre (angelica, sofia).
es madre (victoria, vicente).
es madre (victoria, carolina).
es abuela(X, Y) := mujer(X), es madre(X, Z), es madre(Z, Y).
es_abuela(X, Y) :- mujer(X), es_madre(X, Z), es_padre(Z, Y).
es abuelo(X, Y): - hombre(X), es padre(X, Z), es madre(Z, Y).
es_abuelo(X, Y) :- hombre(X), es_padre(X, Z), es_padre(Z, Y).
es_{max} = mujer(X), es_{max} = mujer(Z, X), es_{max} = mujer(Z, Y).
es hermana(X, Y) :- mujer(X), es padre(Z, X), es padre(Z, Y).
es hermano(X, Y) :- hombre(X), es madre(Z, X), es madre(Z, Y).
```





```
es_hermano(X, Y) :- hombre(X), es_padre(Z, X), es_padre(Z, Y).
es_hijo(X, Y) :- hombre(X), es_padre(Y, X).
es_hijo(X, Y) :- hombre(X), es_madre(Y, X).
es_hija(X, Y) :- mujer(X), es_padre(Y, X).
es_hija(X, Y) :- mujer(X), es_madre(Y, X).
```

- 1.1: Identifica e indica las propiedades, relaciones y reglas que posee la base de conocimiento.
- 1.2: Haz las siguientes preguntas en Prolog e indica sus respuestas. Luego completa los renglones, según corresponda.

. ¿Quién es la madre de Sofia?	
Consulta en Prolog:	
Resultados:	
. ¿Victoria es abuela de Pedro?	
Consulta en Prolog:	
Resultados:	
. ¿Quién es el hijo de Elena?	
Consulta en Prolog:	
Resultados:	
. ¿Roberto y Diego son hermanos?	
Consulta en Prolog:	
Resultados:	







Copia y pega la siguiente base de conocimiento, que establece cómo aprobar un examen, en el simulador de Prolog:

```
persona(maria).
persona(alfredo).
persona(tomas).
persona (susana).
persona (juan).
lee libro(alfredo).
lee libro(maria).
asiste a clases (alfredo).
asiste a clases (maria).
realiza ejercicios (alfredo).
realiza ejercicios (maria).
sabe todo(tomas).
tiene tutor(maria, juan).
tiene_tutor(juan,susana).
aprueba examen(X):- preparado para examen(X).
preparado_para_examen(X):- sabe_todo(X).
preparado para examen(X):- asiste a clases(X),
realiza ejercicios(X), lee libro(X).
preparado para examen(X):-
preparado para examen(Y), tiene tutor(Y,X).
```

- 2.1: Identifica e indica las propiedades, relaciones y reglas que posee la base de conocimiento.
- 2.2: Haz las siguientes preguntas en Prolog e indica sus respuestas. Luego completa los renglones, según corresponda.

a. ¿Aprueba el examen Juan?	
Consulta en Prolog:	
Resultados:	



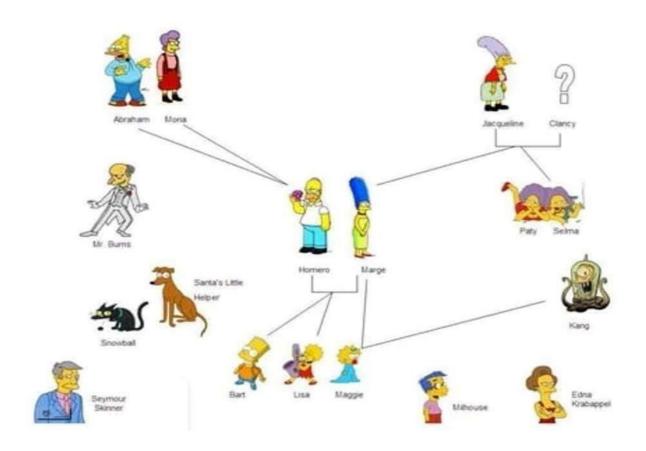


o. ¿Está Susana preparada para el examen?
Consulta en Prolog:
Resultados:
c. ¿Quién es el sabio del grupo?
Consulta en Prolog:
Resultados:
d. ¿Quiénes aprueban el examen?
Consulta en Prolog:
Resultados:





3.a Considerando la lámina anexa tomada de la serie de televisión "Los Simpsons", elabora un programa en Prolog que incluya las propiedades, por ejemplo, el género de las personas; y las relaciones entre las personas -en este caso progenitor/progenitora-:



3.a Asimismo, basadas en las relaciones y las propiedades, establece las reglas para:

hijo\_de(X,Y):hija\_de(X,Y):es\_hermano(X,Y):es\_hermana(X,Y):-





Llegamos al fin de la clase 6, hemos visto uno de los lenguajes de programación por excelencia: Prolog. A partir de esta clase, las actividades serán más interactivas porque se van a basar en programar, ¡las y los esperamos en la siguiente clase!



- Simulador de Prolog para realizar las siguientes actividades:
   <a href="https://swish.swi-prolog.org/">https://swish.swi-prolog.org/</a>
- Playlist: ProLog. Fredy Geek (2021). YouTube.
   <a href="https://www.youtube.com/playlist?list=PLGfF3KgbxaiwLDxZaSuec">https://www.youtube.com/playlist?list=PLGfF3KgbxaiwLDxZaSuec</a>
   2zxNZC7zmQkI
- Anexo: PDF con los procesos que realiza Prolog para dar las respuestas:
   <a href="https://drive.google.com/file/d/1k4rnDSYEWwh5zLqX059DIn10HU">https://drive.google.com/file/d/1k4rnDSYEWwh5zLqX059DIn10HU</a>
   ON3nan/view?usp=drive\_link
- PPT resumido de Prolog:
   <a href="https://docs.google.com/presentation/d/1yCBS9ftGhXRgwG4tbZRR">https://docs.google.com/presentation/d/1yCBS9ftGhXRgwG4tbZRR</a>
   mnPY7NiHuVGjYDB6lydH4T0/edit?usp=drive\_link

