**Programación Lógica**

* Pertenece al grupo de los **lenguajes declarativos.**
* Planteamiento de la solución a un problema basado en un **QUE** y no en un COMO.
* Paradigma de **mayor abstracción**.
* Basado en el **pensamiento lógico del ser humano**.
* Basado en el *cálculo de predicados de primer orden*.

**Características generales**

* Programas no determinísticos, no se construye un algoritmo, cuando se ejecuta un query, Prolog prueba lo que sea necesario para cumplir esa consulta.
* No se utilizan declaraciones de tipos de datos.
* Ambiente de traducción basado en interpretación.
* **Modularización**: dentro de las cláusulas tendremos parámetros de entrada y/o salida, pero sin distinción en su declaración.
* Abstracción de control basada en recursividad.
* Abstracción de datos basada en la estructura de la lista y en describir relaciones entre objetos.
* Internamente se generan estructuras jerárquicas de búsqueda. (Árboles de Búsqueda).
* Automatización de procesos de búsqueda de soluciones, mayores abstracciones, pero menos eficiencia.

**Cálculo de predicados de primer orden**

Lógica de primer orden

* Mundo construido por **objetos.**
  + **Objetos físicos, conceptuales o abstractos.**
  + Los objetos poseen propiedades y relaciones.
  + Relaciones: pueden tener uno o más objetos.
  + Algunas relaciones son **funciones**.

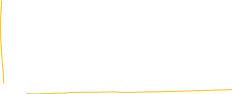
Símbolos

* ***Términos***
  1. Objetos en el mundo (carro, fábricas, etc.)
  2. Nombres, constante.
  3. Variables
* ***Predicados***
  1. Símbolos (constantes).
  2. Unarios: propiedades de objetos
     1. Color, forma, altura, …
  3. N-arios: relaciones entre objetos
     1. Mayor que, entre, mostrar, etc.
* Funciones
  1. Mapeo de objetos a objetos
  2. Padre-de, suma, …

Conjunciones



**Sintaxis CP1**



Oraciones 🡪 OracionAtomica



| Oración Conectivo Oración

Disyunciones



| Cuantificador Variable … Oración



¬ Oración



( Oración )



OracionAtomica 🡪 Predicado(Termino…)



| Termino = Termino



Termino 🡪 Funcion(Termino…)

| Constante | Variable



**Constante:** A | X1 | John

**Variable:** a | x | s

**Predicado:** Antes | TieneColor | Lloviendo

**Función:** MadreDe | PiezIzqDe

*Ejemplo de oraciones atómicas*

* Mujer (María)



* Muere (Pedro)

*Ejemplo de oraciones complejas*

* Izquierda (**x**, A) 🡪 Todos los objetos están a la izquierda de A.



* [Perro(**x**) => Muerde(**x**)] 🡪 Si x es un perro entonces x muerde, todos los perros muerden.



* [Persona(**x**) => Amigo (**x**, y)] 🡪 Todas las personas tienen algún amigo.



**En la Lógica de primer orden**

* Un programa es un conjunto de oraciones lógicas que modelan un conocimiento de un dominio.



* La ejecución consiste en la realización de pruebas de verdad o deducciones o consecuencias de las reglas, dándole así un significado al programa.

**Aplicaciones**

Prolog es un lenguaje de objetivo general.

* Inteligencia artificial
* Bases de datos relacionales
* Procesamiento de lenguaje natural
* Sistemas basados en el conocimiento

**PROLOG**

Lenguaje de propósito general, *Programation et Logique* (Universidad Marsella, 70s).

Creadores son Alan Colmenaeur y Philipe Roussel, trabajos previos de Robert Kowalski.

**Clausulas de Horn**

* Los programas en Prolog se componen de cláusulas de Horn que constituyen oraciones de tipo:
  + Si es verdad el antecedente, entonces es verdad el consecuente
* A V B
* Forma:



* + A1 A2 A3 A4 … => b



* + Representa una implicación.
* En Prolog, se escribe el consecuente antes que el antecedente, el cuerpo define el antecedente.
  + **b**:- a1, a2, a3, …, an.



**Programa en Prolog**

1. *Declaraciones de* ***HECHOS*** acerca de los objetos y sus relaciones.
   1. Clausulas de Horn sin cuerpo.
2. *Declaraciones de* ***REGLAS*** *acerca* de los objetos y sus relaciones.
   1. Clausulas de Horn con cuerpo.
3. **Consultas, interrogaciones o queries (Ejecución del programa)**
   1. Clausulas de Horn sin cabeza.
   2. Provocan el proceso de resolución y unificación de reglas y/o hechos para instanciar variables y dar resultados.

**Hechos**

pred(arg1, arg2, …, argn)

* Donde *pred* es el nombre de una relación y los argumentos pueden ser constantes, variables o estructuras.
* Se escriben como oraciones atómicas.

*Ejemplos de Hechos*

femenino (María).

suma (0,1,1).

**Consultas**

* Comprobación de la veracidad de un hecho, o bien, relaciones entre objetos.
* Se denomina **metas**.
* Se platean directamente a nivel del intérprete.
* Se responden siguiendo el principio de la **correspondencia de patrones**.

*Ejemplo de consultas*

papa (juan, maría): true

suma (1,2,3): false

**Negación como falla**: se considera falso si no se puede probar, mundo cerrado.

**Consultas cuantificadas**

* Uso de variables para cuantificar la consulta
* Si no existen variables, la consulta va a regresar un valor booleano, pero si le ponemos variables, entonces regresa el valor que encuentra para esa variable.

*Ejemplos de consultas cuantificadas*

masculino(X). X = Juan

suma (X, Y, 1). X = 0, Y = 1

**Conjunciones**

*Ejemplos de conjunciones*

Papa(juan, X), femenino(X).



X = maría

**Definición de reglas**

Útiles para cuando un hecho depende del cumplimiento de otros hechos.

* Todas las variables son locales a cada clausula o regla.

*Ejemplo de reglas*

hijo (X, Y):- padre(X, Y), masculino(X, Y).