

---

# Programación Lógico

*Lucas Fanchin | Bruno Motto*

---

## Conceptos Básicos de Prolog

### 1. Hechos y Reglas:

- **Hechos:** Son afirmaciones sobre el mundo. Ejemplo:  
`hamburguesa(lechuga, tomate, carne).` significa que hay una hamburguesa con esos ingredientes.
- **Reglas:** Son relaciones que derivan hechos a partir de otros hechos. Se definen usando `:- (si)`. Ejemplo: `comestible(X) :- hamburguesa(X, _, _).` significa que algo es comestible si es una hamburguesa.

### 2. Consultas:

- Las consultas preguntan si ciertos hechos o reglas son verdaderos. Se escriben en el intérprete de Prolog. Ejemplo: `?- hamburguesa(lechuga, tomate, carne).` pregunta si existe una hamburguesa con esos ingredientes.

### 3. Predicados:

- Los predicados son funciones lógicas que pueden tener varios argumentos, y se definen como `predicado/n` donde `n` es el número de argumentos.  
Ejemplo: `hamburguesa/3`.

### 4. Variables:

- Las variables en Prolog empiezan con mayúscula o con un guión bajo (`_`).  
Ejemplo: `X, Y, _Variable`.

### 5. Unificación:

- Prolog resuelve consultas a través de la unificación, que es el proceso de hacer coincidir estructuras de datos.

## Consultas en Prolog

### 1. Consultas Básicas:

- Se utilizan para preguntar sobre los hechos y reglas definidos en la base de datos. Una consulta básica verifica si un hecho específico es verdadero.
- Ejemplo: Para preguntar si "Tom es un gato".  
`?-gato(tom).`

### 2. Consultas con Variables:

- Las consultas pueden incluir variables para buscar todos los valores que satisfacen la condición dada.
- Ejemplo: Para encontrar todos los gatos:  
`?-gato(X).`

### 3. Consultas con Condiciones:

- Se pueden hacer consultas que impliquen condiciones más complejas utilizando reglas definidas.
- Ejemplo: Para encontrar quiénes son padres de María:  
`?-padre(X, maria).`

### 4. Consultas con Conjunciones (and):

- Las consultas pueden incluir múltiples condiciones que deben ser verdaderas simultáneamente (conjunción).
- Ejemplo: Para encontrar quién es hermano de Pedro y también es hermano de Juan (se usa la coma):  
`?-hermano(X, pedro), hermano(X, juan).`

### 5. Consultas con Disyunciones (or):

- Las consultas pueden incluir condiciones que son verdaderas si al menos una de ellas es verdadera (disyunción).
- Ejemplo: Quién es padre de María o madre de María (se usa punto y coma):  
`?-padre(X, maria) ; madre(X, maria).`

### 6. Consultas de Existencia:

- Consultar si existe algún miembro que cumpla con ciertas condiciones.
- Ejemplo: Para preguntar si existe algún gato (se usa `_`)  
`?-gato(_).`

### 7. Consultas para Listar Todos los Datos:

- Se puede solicitar que Prolog enumere todos los resultados posibles que cumplan una condición.
- Ejemplo: Para listar todos los gatos, usarías `findall`:  
`?-findall(X, gato(X), ListaDeGatos).`

### 8. Consultas en Posiciones Específicas:

- Si estás trabajando con listas o estructuras más complejas, puedes consultar elementos en posiciones específicas.
- Ejemplo: Supongamos un menú que consiste en una entrada, un plato principal y un postre, el predicado `nth1(pos, Menu, n_a_buscar)` permite acceder a elementos en posiciones específicas directamente.  
`?-menu(Menu), nth1(3, Menu, helado).`

Esto debería listar todos los menús que tienen `helado` como postre.

## Símbolos Comunes en Prolog

1. `:-` (si):
  - Utilizado para definir reglas.  
Ejemplo: `padre(X, Y) :- hombre(X), progenitor(X, Y).` significa que X es el padre de Y si X es hombre y X es progenitor de Y.
2. `.` (punto):
  - Utilizado para finalizar hechos, reglas y consultas.  
Ejemplo: `hombre(juan).`
3. `,` (coma):
  - Representa una conjunción lógica (y).  
Ejemplo: `hombre(X), progenitor(X, Y)` significa que ambas condiciones deben ser verdaderas.
4. `;` (punto y coma):
  - Representa una disyunción lógica (o).  
Ejemplo: `hombre(juan); mujer(maria)` significa que al menos una de las condiciones debe ser verdadera.
5. `=` (unificación):
  - Utilizado para unificar dos términos.  
Ejemplo: `X = 3` unifica X con 3.
6. `\=` (no unificación):
  - Utilizado para indicar que dos términos no pueden unificarse.  
Ejemplo: `X \= 3` significa que X no es igual a 3.
7. `is` (evaluación aritmética):
  - Utilizado para evaluar expresiones aritméticas.  
Ejemplo: `X is 3 + 4` asigna a X el valor 7.

## Representando información y relaciones en Prolog

### → Términos

Los términos son los componentes básicos que incluyen constantes, variables, y estructuras.

```
% Constante: 'tom' es una constante.  
% Variable: 'X' es una variable.  
% Estructura: 'gato(tom)' es una estructura con el functor 'gato' y un argumento 'tom'.  
  
gato(tom).  
animal(X) :- gato(X).
```

### → Lógica de Predicados

Se usa para expresar relaciones entre objetos mediante predicados, que son funciones que devuelven verdadero o falso.

```
% Predicado 'amigo' que relaciona dos individuos.  
amigo(juan, maria).  
amigo(maria, pedro).  
  
% Consulta para verificar si Juan y María son amigos.  
?- amigo(juan, maria).  
% Resultado: true.
```

### → Fórmulas atómicas

Representan afirmaciones simples o relaciones entre entidades del dominio del problema. Tienen la forma p(t1, ..., tn), donde p es un predicado y ti son términos. Son la base de las expresiones lógicas en prolog.

```
% Fórmulas atómicas:  
hermano(juan, pedro).  
padre(juan, maria).
```

### → Orden de Precedencia de los Conectores Lógicos

Determina cómo se evalúan las expresiones lógicas.

```
% Conectores lógicos: ',' (AND), ';' (OR).  
% Orden de precedencia: ',' tiene mayor precedencia que ';'.  
  
% Ejemplo:  
padre(juan, maria), madre(ana, maria); hermano(pedro, maria).  
  
% Esto se interpreta como:  
(padre(juan, maria), madre(ana, maria)) ; hermano(pedro, maria).
```

### → Semántica de las Fórmulas Lógicas

La semántica de las fórmulas lógicas se refiere al significado de las expresiones lógicas y cómo se evalúan en términos de verdad.

```
% La regla significa que 'X es un humano si X es una persona'.  
  
humano(X) :- persona(X).  
persona(juan).  
  
% Consulta:  
?- humano(juan).  
% Resultado: true.
```

## → Inferencia Lógica

Es el proceso de derivar nuevas verdades a partir de hechos y reglas conocidas.

```
% Reglas:  
mortal(X) :- humano(X).  
humano(socrates).  
  
% Consulta para inferir:  
?- mortal(socrates).  
% Resultado: true.
```

## → Cláusulas Definidas

En Prolog son reglas que tienen la forma de implicaciones, donde la cabeza es una fórmula atómica y el cuerpo es una conjunción de fórmulas atómicas.

```
% Significa que 'X es un abuelo de Z si X es el padre de Y y Y es el padre de Z'.  
  
abuelo(X, Z) :- padre(X, Y), padre(Y, Z).  
  
padre(juan, pedro).  
padre(pedro, luis).  
  
% Consulta:  
?- abuelo(juan, luis).  
% Resultado: true.
```

## → Recursividad

Se basa en definir relaciones en términos de ellas mismas.

En una definición recursiva, es necesario considerar dos casos: - Caso Básico: Momento en que se detiene el proceso o cómputo. - Caso Recursivo: Suponiendo que ya se ha solucionado un caso más simple, se descompone el caso actual hasta llegar al caso más simple.

```
sucesor(X, Y) :- progenitor(X, Y).  
sucesor(X, Y) :- progenitor(X, Z), sucesor(Z, Y).
```