



PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática
Programador Universitario



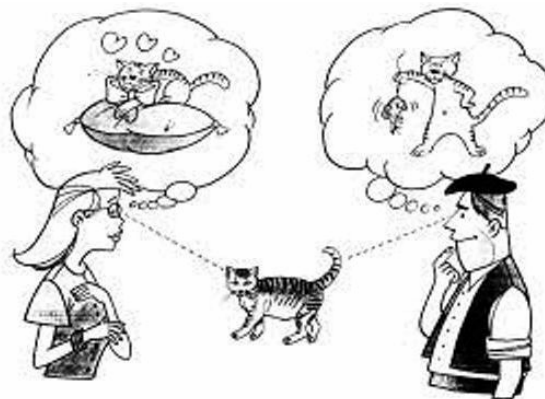
UNIDAD I

PARADIGMAS DE PROGRAMACIÓN

Tipos de Paradigmas

PARADIGMA

Un enfoque particular o filosofía que afecta la **forma en que un individuo percibe la realidad** y la forma en que responde a esa percepción



PARADIGMA

PARADEIGMA

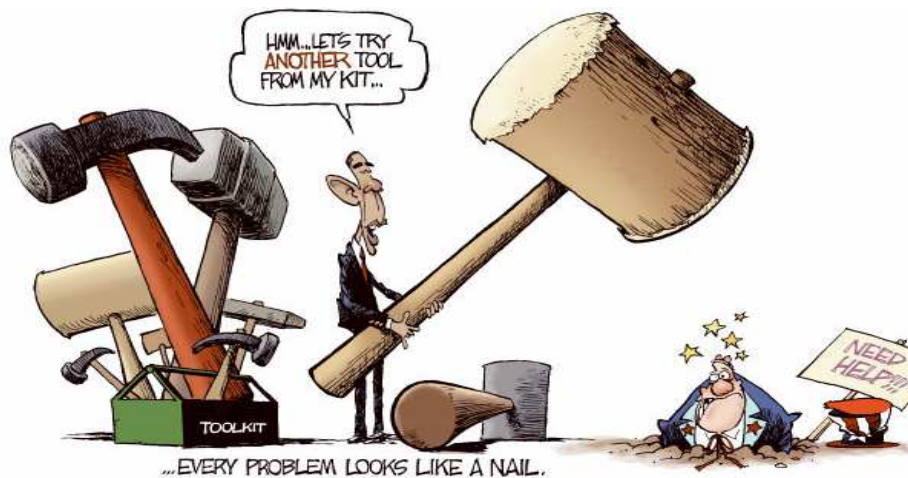
Modelo o ejemplo representativo

Paradigma de Programación

Modelo básico de diseño e implementación que permite producir programas conforme con directrices específicas

3

PARADIGMA



4

Tipos de Paradigmas

- **Paradigma Basado en reglas:** se utilizan para el desarrollo de sistemas expertos, en el que el núcleo del mismo son las reglas de producción "If-then".
- **Paradigma Lógico:** Basado en asertos y reglas lógicas.
- **Paradigma Funcional:** Basado en funciones, formas funcionales para crear funciones y mecanismos para aplicar los argumentos.
- **Paradigma de Programación Heurística:** modela el problema de una forma adecuada aplicando heurísticas, teniendo en cuenta la forma de representación de las mismas, las estrategias de búsqueda y métodos de resolución.
- **Paradigma de Programación Paralela**
- **Paradigma Imperativo**
- **Paradigma Orientado a Objetos**
- ...

5

Paradigma de Programación

Definición*

Colección de **modelos conceptuales** que juntos **organizan el proceso de diseño** y **determinan la estructura de un programa**

* Operational Versus Definitional: A Perspective on Programming Paradigms, Ambler, A. Burnett, M. Zimmerman, B. IEEE Computer.1992.

6

Lenguajes de Programación

- **Herramienta** para implementar un programa en un determinado paradigma.
- El lenguaje de programación que soporta un determinado paradigma debe **reflejar adecuadamente los modelos conceptuales** de ese paradigma.
- Un lenguaje que soporta correctamente un paradigma es difícil de distinguir del propio paradigma, por lo que se identifica con él.

7

Tipos de Paradigmas de Programación

Categorías*

Los que soportan técnicas de programación de **bajo nivel**

Los que soportan métodos de **diseño de algoritmos**

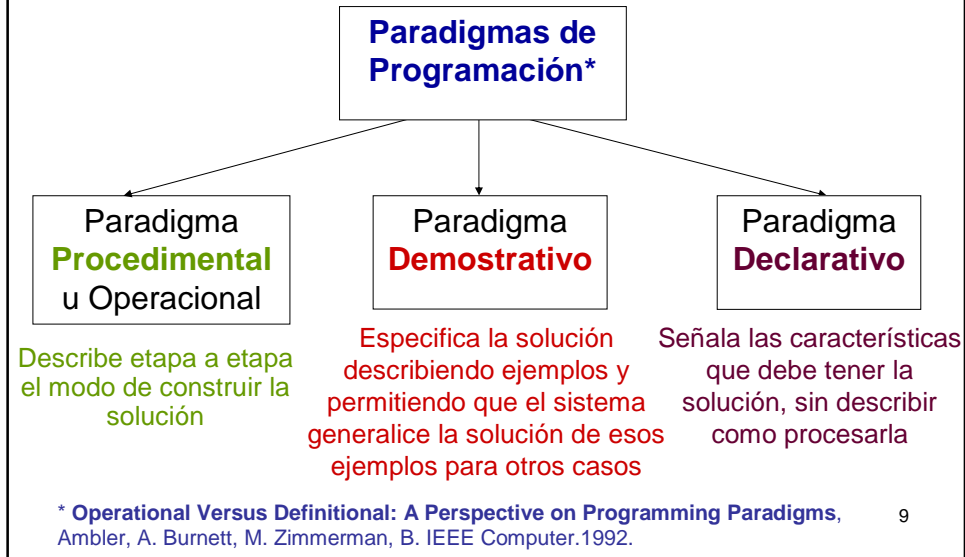
Los que soportan soluciones de programación de **alto nivel**



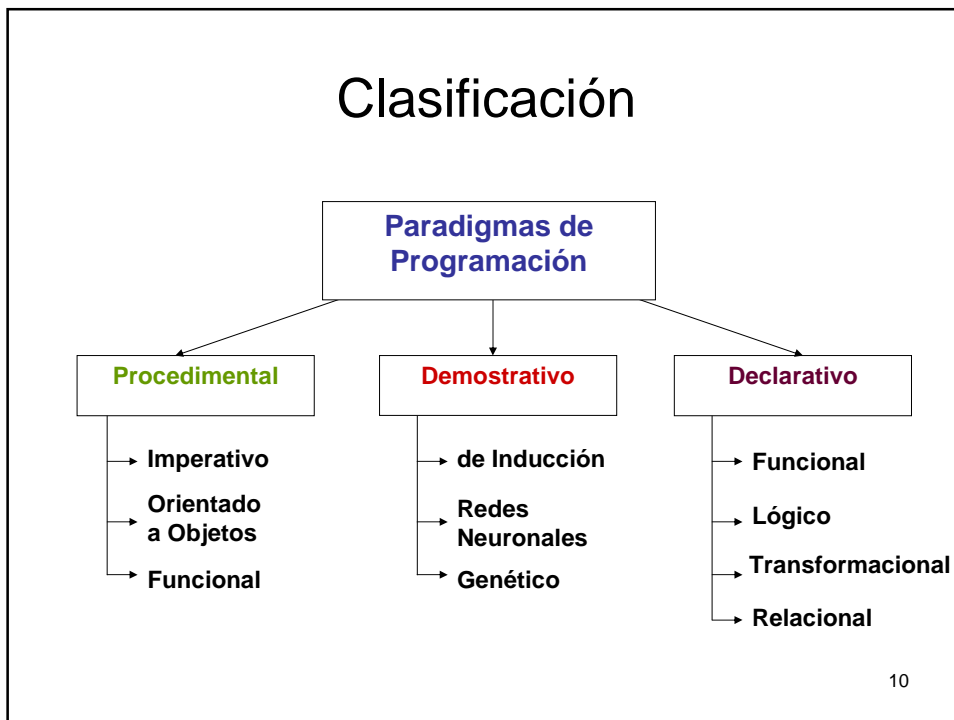
* The Paradigms of Programming
Robert W. Floyd
1978 ACM Turing Award Lecture

8

Programación de **Alto Nivel**



Clasificación





PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática
Programador Universitario



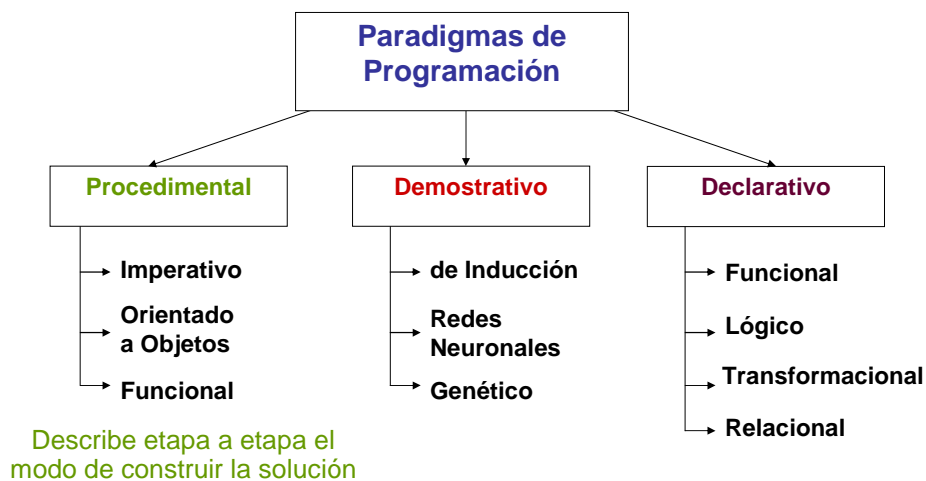
UNIDAD I

PARADIGMAS DE PROGRAMACIÓN

Tipos de Paradigmas

Procedimentales

Clasificación



Paradigmas Procedimentales

Especifican la **solución** como un **conjunto de secuencias computacionales** que se ejecutan paso a paso

Problema

Determinar si el valor computado es una solución correcta del problema

3

Paradigmas Procedimentales

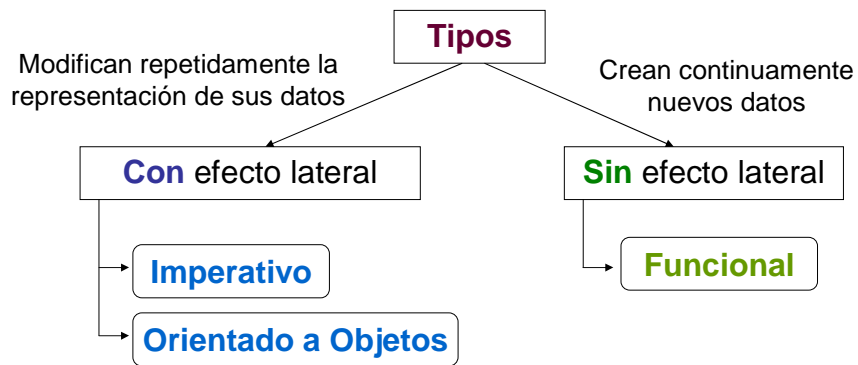


Margaret Hamilton
Directora de la División
de Ingeniería de Software
del MIT
Proyecto de Apollo 11

Foto: Instrumentation
Laboratory - MIT 1969

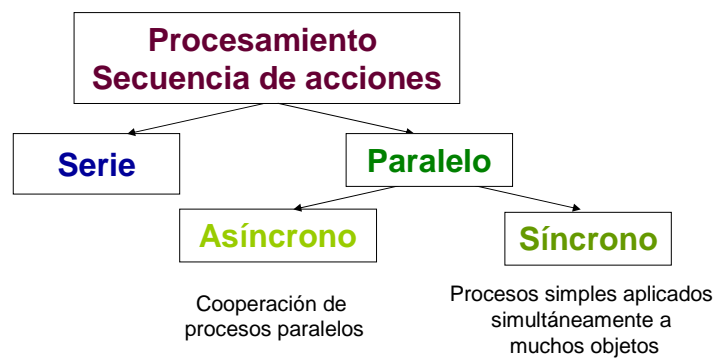
4

Paradigmas Procedimentales



5

Paradigmas Procedimentales



6



PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática
Programador Universitario



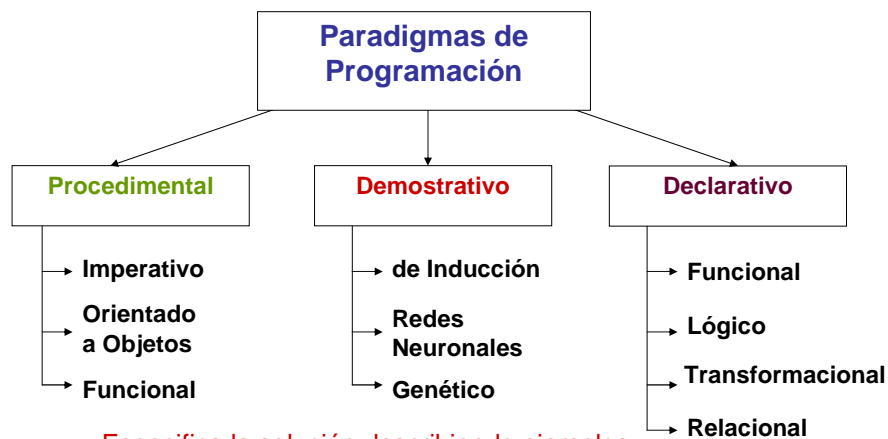
UNIDAD I

PARADIGMAS DE PROGRAMACIÓN

Tipos de Paradigmas

Demostrativos

Clasificación



Especifica la solución describiendo ejemplos y permitiendo que el sistema generalice la solución de esos ejemplos para otros casos

Paradigmas Demostrativos

Se presentan **soluciones a problemas similares** y permite que el **sistema generalice una solución** procedimental a partir de estas demostraciones

- Los sistemas que hacen **inferencias** intentan generalizar usando razonamiento basado en conocimiento.
- **Programación asistida**: un sistema observa acciones que el programador ejecuta, y si son similares a acciones pasadas intentará inferir cuál es la próxima acción.
- **Paradigmas**:
 - Inducción
 - Redes Neuronales
 - Genéticos

3

The Neural Net Tank Urban Legend*



Fotos con
tanques de guerra

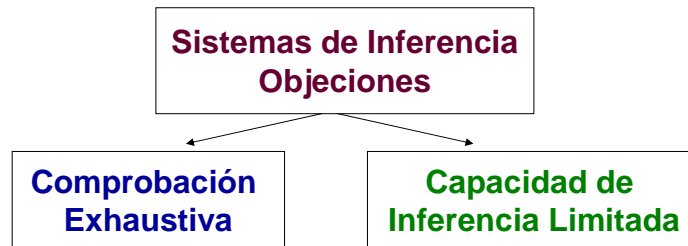
Fotos sin
tanques de guerra



*www.gwern.net/Tanks

4

Paradigmas Demostrativos





PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática
Programador Universitario



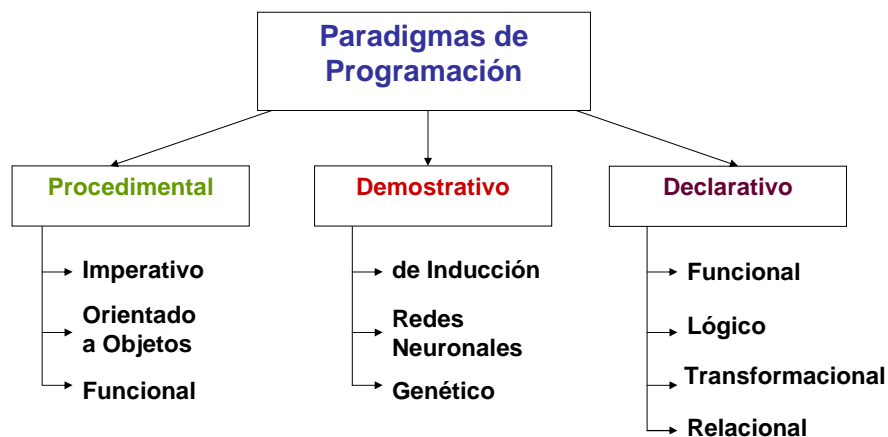
UNIDAD I

PARADIGMAS DE PROGRAMACIÓN

Tipos de Paradigmas

Declarativos

Clasificación



Señala las características que debe tener la solución, sin describir como procesarla

Paradigmas Declarativos

Un programa se construye señalando **hechos**, **reglas**, **restricciones**, **ecuaciones**, **transformaciones** y otras **propiedades** que debe tener el conjunto de valores que constituyen la solución

- A partir de esta información, el **sistema debe** ser capaz de **derivar un esquema de evaluación** que nos permita computar una solución.
- **NO** describen las diferentes etapas a seguir para alcanzar una solución.

3

Paradigmas Declarativos

- Programar consiste en centrar la atención en la lógica dejando de lado el control, que se asume automático, al sistema.
- **Característica Fundamental** → uso de la **lógica** como **lenguaje de programación**.
- Un **programa** es un **conjunto de fórmulas lógicas** que resultan ser la especificación del problema que se pretende resolver, y la computación se entiende como una forma de inferencia o deducción en dicha lógica.

4

Paradigmas Declarativos

Un detective tiene que resolver un crimen, para ello dispone de los siguientes datos:

- O el crimen se cometió de noche en la más absoluta oscuridad o el principal sospechoso es ciego.
- Pero, o el principal sospechoso no es ciego o miente al declarar que no vio nada.
- Si el principal sospechoso miente entonces el detector de mentiras está estropeado.
- El caso es que el citado detector no puede estar estropeado jamás (es infalible).

¿Miente o no miente el sospechoso? ¿Es ciego o no lo es?
¿Se cometió el crimen de día o de noche?

5

Paradigmas Declarativos

P = El crimen se cometió de noche

Q = El principal sospechoso es ciego

R = El principal sospechoso miente

S = El detector de mentiras está estropeado

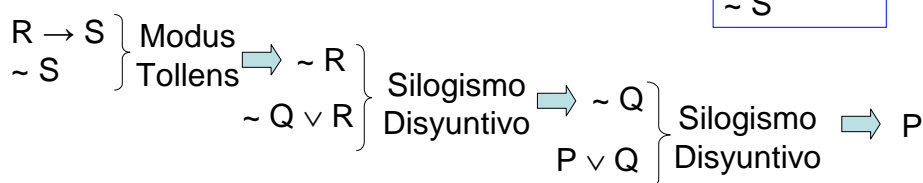
PLANTEO

$P \vee Q$

$\sim Q \vee R$

$R \rightarrow S$

$\sim S$



¿Miente o no miente el sospechoso? ¿Es ciego o no lo es? ¿Se cometió el crimen de día o de noche?

6

Paradigmas Declarativos

- **NO** dirigen la secuencia de control \therefore no son soluciones de tipo serie o paralelo.
- En la práctica **se debe proporcionar un esquema que incluya el orden de evaluación** que compute una solución, por lo que los paradigmas resultantes y los lenguajes que los soportan no son verdaderamente declarativos sino **pseudo-declarativos**.
- **Paradigmas:**
 - Funcional
 - Lógico
 - de Transformación
 - Relacional

7



PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática
Programador Universitario



UNIDAD I

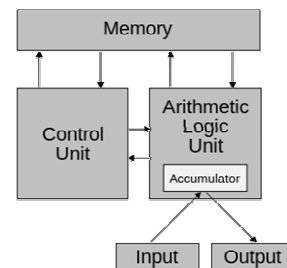
PARADIGMAS DE PROGRAMACIÓN

Paradigmas Procedimentales

Imperativo

Paradigma Imperativo

- Su nombre se debe al papel dominante que desempeñan las **sentencias imperativas**.
- El paradigma está representado por la **arquitectura Von Neumann**, ya que usa un modelo de máquina para conceptualizar la solución.
- Su esencia es el cálculo iterativo, paso a paso, de valores de nivel inferior y su asignación a posiciones memoria.



*Conceptos de Lenguajes de Programación, C. Ghezzi – M. Jazayeri, Ed. Diaz de Santos, Madrid

2

Paradigma Imperativo

- Programar en este paradigma consiste en:
 - Determinar qué datos son requeridos para el cálculo
 - Asociar a esos datos unas direcciones de memoria
 - Efectuar, paso a paso, una secuencia de transformaciones en los datos almacenados
 - El resultado final representa el resultado correcto
- En su **Forma Pura**, este paradigma únicamente soporta **sentencias simples** que modifican la memoria y efectúan bifurcaciones condicionales o incondicionales.

3

Paradigma Imperativo

Características principales

- **Concepto de celda de memoria (*variable*) para almacenar valores:** Las celdas tienen nombres que permiten referenciarlas y sobre las que se producen efectos laterales y definiciones de alias.
- **Operaciones de asignación:** Cada valor calculado debe ser almacenado. La sentencia de asignación es de suma importancia en el paradigma.
- **Repetición:** Un programa imperativo normalmente realiza su tarea ejecutando repetidamente una secuencia de pasos elementales.

4

Paradigma Imperativo

Recibe también el nombre de **Paradigma Algorítmico** ya que se entiende que el concepto de algoritmo es privativo de la programación procedimental porque su característica principal es la secuencia computacional

Un programa se define por la ecuación (según N. Wirth)

Algoritmos + Estructuras de datos = Programas

5

Lenguajes Imperativos

- Orientados a **Expresiones**:

- Fortran
- Algol
- Pascal
- C, etc.

PASCAL

- Expresión Condicional

`v := x > y`

- Sentencia Condicional

```
if x > y then v:= true
else v:= false;
```

- Orientados a **Sentencias**:

- Cobol
- PL/1.

```
continue
call funcion()
```

6

Paradigma Imperativo

**Encontrar el MCD
entre a y b**

ALGORITMO: **MaximoComunDivisor**

ENTRADA: a, b: entero

SALIDA: mcd: entero

VAR.AUX.: temp: entero

A1. LEER (a, b)

A2. **Encontrar MCD** →

A3. ESCRIBIR (mcd)

A4. PARAR

A2. **Encontrar MCD**

A2.1. MIENTRAS (b \neq 0)

temp \leftarrow a

a \leftarrow b

b \leftarrow MOD(temp, b)

A2.2. mcd \leftarrow a



PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática
Programador Universitario



UNIDAD I

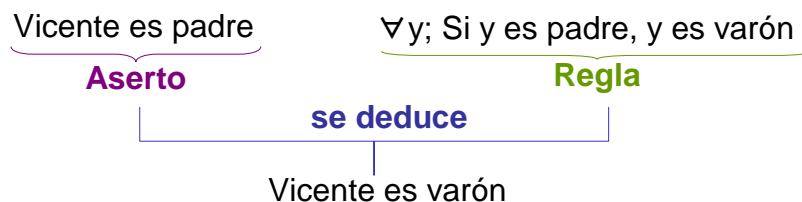
PARADIGMAS DE PROGRAMACIÓN

Paradigmas Declarativos

Paradigma Lógico

Paradigma Lógico

Asume un **conjunto de hechos** (o asertos) y un **conjunto de reglas** que permiten deducir nuevos hechos



Paradigma Lógico

- Fue creado por **Robert Kowalski** en el Imperial College de Londres y lo implementó en base a las **cláusulas de Horn**, que son un subconjunto de la lógica de predicados de primer orden.

Forma restrictiva de lógica de predicados con una sola conclusión en cada cláusula

Una fórmula lógica es una **Cláusula de Horn** si es una cláusula (disyunción de literales) con un literal positivo como máximo

$$\neg p \vee \neg q \vee \dots \vee \neg t \vee u$$
$$(p \wedge q \wedge \dots \wedge t) \rightarrow u$$

3

Paradigma Lógico

- La notación causal de la lógica de predicados combina variables, constantes y expresiones para expresar proposiciones condicionales

`(natural(X) ∧ divisor(2,X)) → par(X)`

- HECHOS**: fórmulas atómicas (p, q, t, u)

Juan es marido de Luisa

El número 0 es par

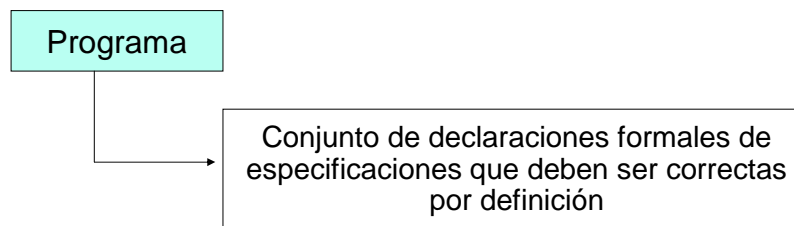
- REGLAS**: fórmulas de la forma $(p \wedge q \wedge \dots \wedge t) \rightarrow u$

$\forall x, x$ es un número natural y es par \rightarrow el siguiente de x es impar

4

Paradigma Lógico

La **programación lógica** es una instrucción de componentes lógicos de un algoritmo, siendo el sistema el que proporciona la secuencia de control



5

Paradigma Lógico

- En este tipo de programación, la evaluación empieza definiéndose una **meta** e intentando probar que ella se ajusta a un hecho o se deduce de alguna regla.

Ej: ¿El número 6 es par?

- Una **meta** se deduce de una regla si todos los antecedentes de la regla se verifican con la transformación existente; para lo cual, estos antecedentes se convierten en nuevas metas que deben equipararse con hechos o resolverse vía otras reglas. El proceso termina cuando todas las submetas han sido probadas. Y la solución final viene determinada al aplicar los resultados intermedios obtenidos a las variables de la meta inicial.

6

Paradigma Lógico

Un detective tiene que resolver un crimen, para ello dispone de los siguientes datos:

- O el crimen se cometió de noche en la más absoluta oscuridad o el principal sospechoso es ciego.
- Pero, o el principal sospechoso no es ciego o miente al declarar que no vio nada.
- Si el principal sospechoso miente entonces el detector de mentiras está estropeado.
- El caso es que el citado detector no puede estar estropeado jamás (es infalible).

¿Miente o no miente el sospechoso? ¿Es ciego o no lo es? ¿Se cometió el crimen de día o de noche?

7

Paradigma Lógico

- **Problema** → definir el mecanismo de búsqueda adecuado para seleccionar las reglas

Búsqueda DFS

Deep First Search
con un proceso de
Backtracking

8

Paradigma Lógico

- **Pseudodeclarativo:**

- **Ordena** cuidadosamente tanto los hechos como las reglas, para aumentar la eficiencia de la selección y su terminación.
- Usa **mecanismos de corte (!)**, que son dispositivos no lógicos que **inhiben el mecanismo de backtracking** y fueron introducidos para **aumentar la eficiencia** de los mecanismos de búsqueda.

9

Paradigma Lógico

Un **programa lógico** se configura como un **conjunto de hechos** (asertos o proposiciones) y de **reglas lógicas** previamente establecidas, que obtienen conclusiones en base a una serie de preguntas o cuestiones lógicas

Un programa en este paradigma se define por la ecuación:

Lógica + Control + Estructuras de datos = Programa

Donde:

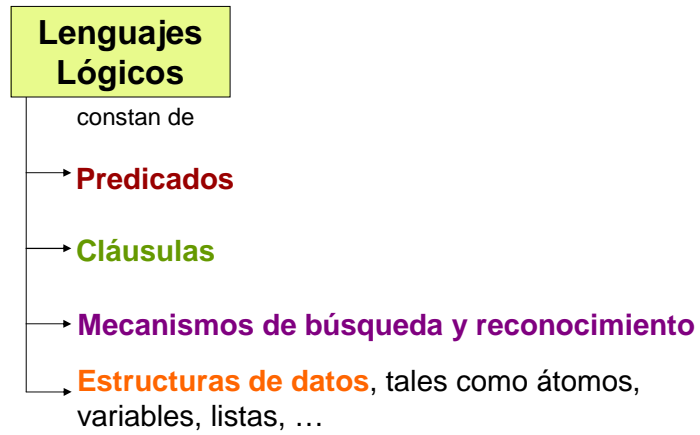
Lógica - Está constituida por los asertos y reglas lógicas.

Control - Inherente al sistema.

Estructuras de Datos - Son los elementos que soportan la base de conocimiento y cuestiones lógicas (variables, constantes, listas,...)

10

Paradigma Lógico



11



PARADIGMAS DE PROGRAMACIÓN

Licenciatura en Informática
Programador Universitario



UNIDAD I

PARADIGMAS DE PROGRAMACIÓN

PROLOG

Paradigma Lógico

Paradigma Lógico

Lenguajes Lógicos

constan de

- **Predicados**
- **Cláusulas**
- **Mecanismos de búsqueda y reconocimiento**
- **Estructuras de datos**, tales como átomos, variables, listas, ...

Lenguaje PROLOG

a) Predicados (hechos): se utilizan para expresar propiedades de los objetos

`predicado(argumento).`

Los objetos deben ir entre () separados por ,

Debe comenzar con minúscula

`predicado(arg1,arg2,..., argn).`

```
programador(luis).    % Luis es programador
hijo(luis,juan,ana).  % Luis es hijo de Juan y Ana
casado(juan,ana).     % Juan está casado con Ana
almuerza(jorge, X).  % Jorge almuerza cualquier cosa
```

3

Lenguaje PROLOG

b) Cláusulas: Definen reglas lógicas que permiten inferir otros conceptos al aplicarlas a una Base de Conocimiento.

`<parte-izq-regla> :- <parte-der-regla>.`

SI <parte-der-regla> ENTONCES <parte-izq-regla>

```
hombre(X) :- varon(X), adulto(X).
progenitor(X,Y) :- padre(X,Y); madre(X,Y).
vuela(X):- pajaro(X), \+ pinguino(X), \+ avestruz(X)
```

Conjunción (y)

,

Disyunción (o)

;

Negación

\+

4

Lenguaje PROLOG

```
hombre(X) :- varon(X), adulto(X).  
progenitor(X,Y) :- padre(X,Y); madre(X,Y).  
vuela(X):- pajaro(X), \+ pinguino(X), \+ avestruz(X)
```

¿Cómo escribimos la relación abuelo?

```
abuelo(X,Z) :- hombre(X), progenitor(X,Y), progenitor(Y,Z).
```

```
 $\forall X, Z \exists Y :$   
(X es hombre  $\wedge$  X es progenitor de Y  $\wedge$  Y es progenitor de Z)  $\Rightarrow$  X es abuelo de Z
```

5

Lenguaje PROLOG

c) Mecanismos de Búsqueda: la búsqueda en la Base de Conocimiento se efectúa de arriba a abajo, usando el algoritmo de **DFS** (Deep First Search) con aplicación de **Backtracking**. Las reglas se resuelven de izquierda a derecha.

```
numero(0):-write(0).  
numero(N):-N1 is N-1, numero(N1), write(N).
```

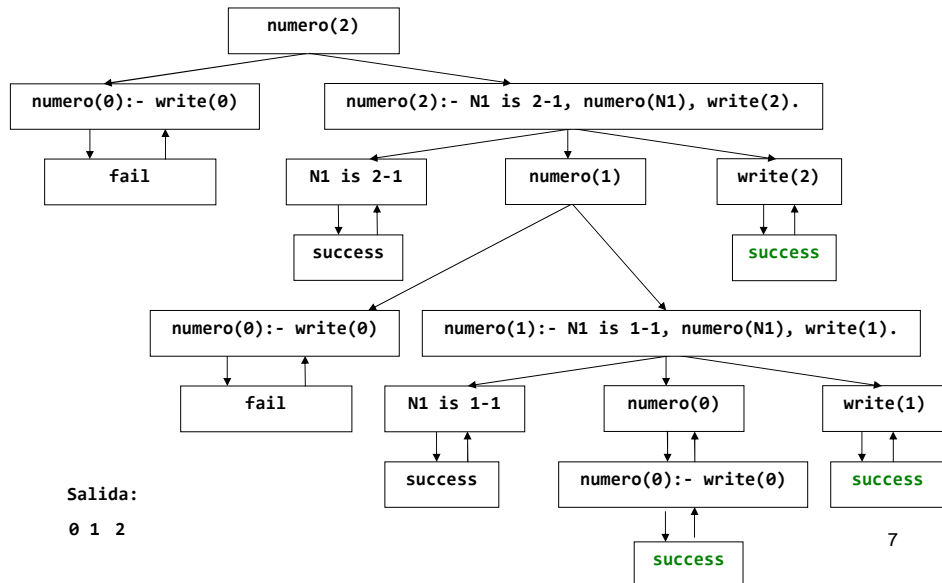
Consulta

```
?.-numero(2).  
012
```

6

Mecanismos de Búsqueda

```
numero(0):-write(0).
numero(N):-N1 is N-1, numero(N1), write(N).
```



Lenguaje PROLOG

Base del Conocimiento

```
empleado(juan).
empleado(luis).
secretaria(rosa).
jefe(pedro).
jefe(julio).
supervisor(X,Y) :- jefe(X),empleado(Y).
supervisor(X,Y) :- jefe(X),secretaria(Y).
```

¿Cuál será la respuesta de Prolog a la pregunta ?

? - supervisor(X,Y).

8

La respuesta será

```
X=pedro, Y=juan  
X=pedro, Y=luís  
X=julio, Y=juan  
X=julio, Y=luís  
X=pedro, Y=rosa  
X=julio, Y=rosa
```

Para suprimir el mecanismo de Backtracking rescribimos las reglas con el operador ! (cut)

```
supervisor(X,Y) :- jefe(X), empleado(Y), !.  
supervisor(X,Y) :- jefe(X), secretaria(Y), !.
```

La respuesta será

```
X=pedro, Y=juan
```

9

Lenguaje PROLOG

d) Estructuras de datos:

Constantes: Átomos, literales ó Valores enteros

```
luís, silla, manzana  
'hola'  
20, 50, 123
```

Variables

```
X, Y, N1, Hola
```

Listas

```
[a, b, c, d]  
[rojo, verde, azul]
```

10

Lenguaje PROLOG

Lista → `[cabeza | resto]`

```
vocales([a, e, i, o, u]).
```

```
? - vocales(X).  
X = [a, e, i, o, u]
```

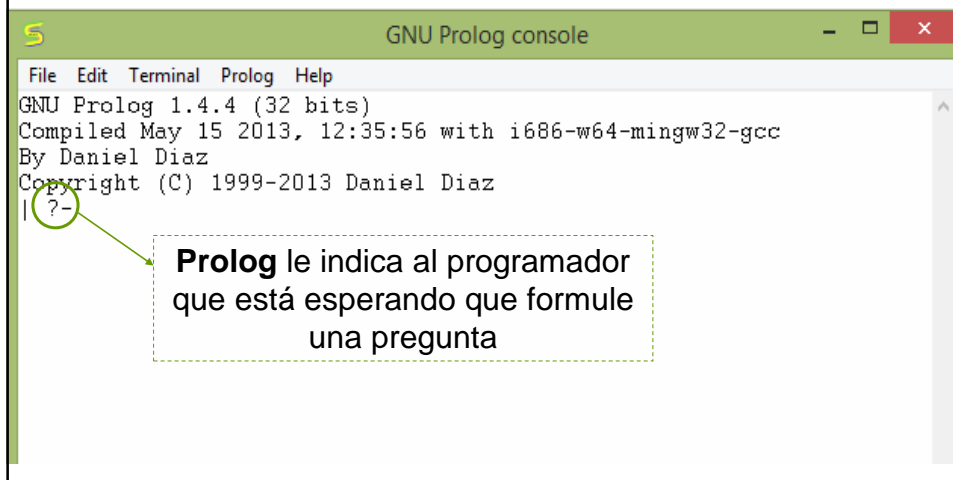
```
? - vocales([H | T]).  
H = a  
T = [e, i, o, u]
```

```
? - vocales ([X | _ ]).  
X = a
```

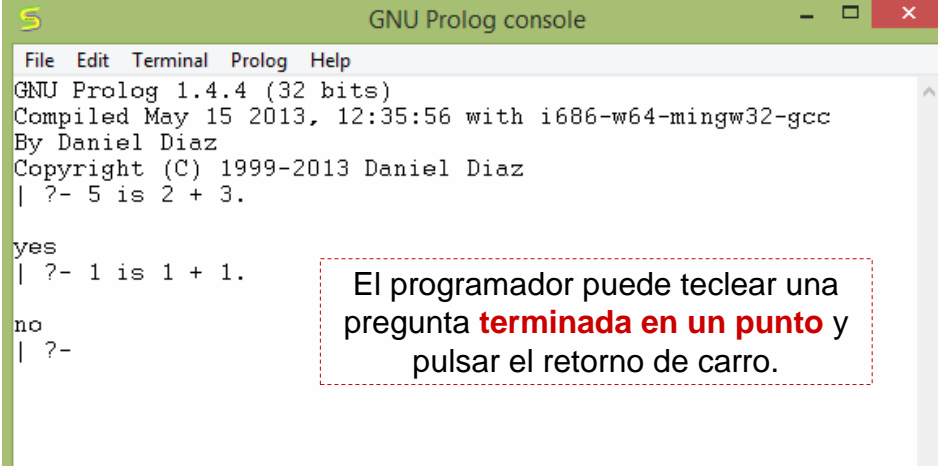
11

Lenguaje PROLOG

Prolog es un lenguaje conversacional



Lenguaje PROLOG



The screenshot shows the GNU Prolog console window. The title bar is green and says "GNU Prolog console". The menu bar includes File, Edit, Terminal, Prolog, and Help. The console text is as follows:

```
GNU Prolog 1.4.4 (32 bits)
Compiled May 15 2013, 12:35:56 with i686-w64-mingw32-gcc
By Daniel Diaz
Copyright (C) 1999-2013 Daniel Diaz
| ?- 5 is 2 + 3.

yes
| ?- 1 is 1 + 1.

no
| ?-
```

A red dashed box highlights the text: "El programador puede teclear una pregunta **terminada en un punto** y pulsar el retorno de carro."

13

Lenguaje PROLOG

```
?- 5 is 2+.3.
uncaught exception:
  error(syntax_error('user_input:15
(char:7) . or operator expected after
expression'),read_term/3)
```

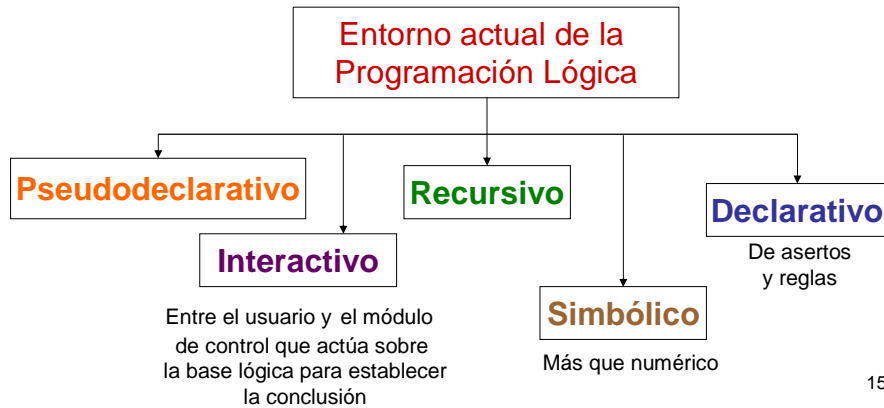
Las preguntas son términos Prolog y deben ajustarse a una **sintaxis formal concreta**. Si la pregunta en cuestión no es un término correcto, habremos cometido un error sintáctico.

Prolog detecta tales errores y nos avisará que no entiende la pregunta formulada.

14

Lenguaje PROLOG

En la actualidad PROLOG cuenta con estrategias y características que **NO** se corresponden con el paradigma declarativo.



15