

PROGRAMACIÓN LÓGICA Y FUNCIONAL

SESIÓN 10

CICLO: AGOSTO 2021



Universidad
Tecnológica
del Perú

CONTENIDO

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA



Pautas de trabajo

- Los días que tengamos clases debemos conectarnos a través de Zoom.
- La participación de los estudiantes se dará través del **chat de Zoom.**
- En Canvas encontrarán la clase de hoy, el pdf de la sesión 10, Laboratorio 10

RECORDANDO LA SESIÓN ANTERIOR

¿ QUE SE ENTIENDE POR PROGRAMACION LÓGICA?

Levantemos la mano para participar



Logro del Aprendizaje

Al finalizar la presente sesión el estudiante tiene conocimiento de la semántica de la Programación lógica



Utilidad

¿Porqué será importante tener conocimientos de la semántica de programación lógica?



SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA

- **Interpretación:** Sea R un programa en lógica con sus vocabulario V de átomos, F de símbolos funcionales y P de significado predicativos Sea D un conjunto (dominio) dado.

Una interpretación $I(D)$ sobre el programa R asigna a cada elemento de V , F , Y P los correspondientes elementos de D .

- **Satisfacción lógica:** Una interpretación $I(D)$ satisface a una fórmula, si su aplicación resulta verdadera.

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA

- ◉ **Consecuencia lógica o deducción:** Una fórmula f es consecuencia lógica o se deduce, de un conjunto de fórmulas R , si todo dominio D , toda interpretación $I(D)$ que satisface a R , satisface también a f .

$$R := f.$$

Permite determinar la forma en que los programas son evaluados.

- ◉ **Inferencia lógica:** Conjunto inicial de fórmulas son sentencias válidas y se las llama *axiomas*. Los axiomas junto a las reglas de inferencia constituyen *sistemas de formas*.

Elemento de derivación sintáctica que a partir de conjunto de fórmulas permite derivar nuevas fórmulas.

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA



SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA



Universidad
Tecnológica
del Perú

- **Regla básica de inferencias:** De las fórmulas A y $A \rightarrow B$ se puede inferir B . Un paso de inferencia corresponde a la aplicación de una regla para inferir una nueva fórmula.

$$\frac{A \quad A \rightarrow B}{B}$$

- **Demostración:** Será sucesión de F_1, F_2, \dots, F_n de fórmulas del lenguaje.

F_1 es axioma o se obtiene de fórmulas anteriores por aplicar una regla de inferencia.

- **Teorema:** Una fórmula F es un teorema si existe una demostración en la que F es el último término de la sucesión.

$$: \neg F$$

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA

- ◉ **Deducción lógica:** Sea F una fórmula y R un conjunto de fórmulas. Se dice que F es lógicamente deducible lógicamente a partir de R .

$$R := F \qquad R: - F$$

Si existe una sucesión de fórmulas F_1, F_2, \dots, F_n tal que $F = F_n$ y cada F_i es:

- ◉ Un axioma.
- ◉ O una fórmula de R .
- ◉ O deducible de una fórmula precedente de la sucesión.

Las fórmulas de R se llaman *hipótesis*.

- ◉ **Consistencia:** Un sistema formal en el que exista el símbolo de negación " \sim " se dice que es consistente si no existe una fórmula F en el sistema tal que pueda deducirse F y $\sim F$.

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA

- **Compleitud:** Sea P un programa en lógica y
 P Cláusulas del programa
 Q Regla de inferencia
 $P : - p$ (p es deducible de P)
 $P := p$ (P es consecuencia lógica de p)
- **Regla de resolución:** Sean A_1, A_2, \dots, A_n y B_1, B_2, \dots, B_m símbolos predicativos, la regla provee:
 $\sim (A_1, \dots, A_k, \dots, A_n)$
 $\frac{AK \leftarrow B_1, \dots, B_m}{\sim (A_1, \dots, A_k - 1, B_1, \dots, B_m, A_k + 1, \dots, A_n)}$

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA

La aplicación de la regla induce una nueva cláusula producto de la cancelación del predicado A_k .

Casos particulares de la regla:

$$\frac{\sim A}{A \leftarrow B} \quad \frac{\sim A}{\frac{A}{\square}}$$

- Teorema: La regla de resolución es completa.
- Teorema: Prueba por contradicción o reducción al absurdo Sea F una fórmula y R un conjunto de fórmulas, F es deducible lógicamente a partir de R si el conjunto formado por R con los $\sim F$ es inconsistente:

$R: -F$ si R unión $(\sim F)$ es inconsistente = Principio de resolución de Robinson.

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA

- **Cláusulas de Horn:** Pueden tener como máximo una conclusión y en las cuales pueden re expresarse cualquier cláusula de forma general. Pueden adoptar las siguientes formas:
 - $B \leftarrow A_1, A_2, \dots, A_n$ (Implicaciones)
 - $B \leftarrow$ (Afirmaciones)
 - $\leftarrow A_1, A_2, \dots, A_n$ (Negaciones)
- Cuando una *implicación* no tiene antecedente (condición) se denomina afirmación. (nada condiciona al consecuente).
- Si no tiene consecuente (conclusión) es en realidad una negación.
- Una cláusula sin antecedente ni consecuente es vacía [] y representa una contradicción.
- Regla genérica, si el predicado A_1, A_2, \dots , son ciertos *implican* B es una regla o sentencia.

SEMÁNTICA DE LA PROGRAMACIÓN LÓGICA

- **Sustitución:** Es un conjunto de asignaciones de tipo $X := t$ donde X es una variable y t un término. No puede existir más de una asignación.
Ej.: $\{X := \text{juan}, Y := \text{maría}\}$
- **Aplicaciones de sustitución:** Dada una sustitución x y un predicado P , la aplicación de x a P produce un nuevo predicado Px y que corresponde al predicado inicial P , donde toda variable asignada en x es cambiada por el término correspondiente.
- **Unificador:** Dada dos expresiones del lenguaje, Ej.: predicados $P1$ y $P2$, se llama unificadora una sustitución x que cumple:
 $E1x = E2x$ en la aplicación de la sustitución da la misma expresión.

PROGRAMAS LÓGICOS

- Un programa lógico es un conjunto de sentencias o cláusulas.

$B \leftarrow (\text{hecho})(\text{cláu. Horn - afirmación incondicional})$

$B \leftarrow A1, A2, \dots, An \quad (\text{regla}) \quad (\text{afirmación condicionada})$

B se denomina "cabeza" y
los antecedentes "cuerpo"

- El conjunto de todas las cláusulas que tienen como cabeza el mismo predicado, constituye su definición.

PROGRAMAS LÓGICOS



- Otra clausula que interesa es la denominada "clausula objetivo" o "goal".
- Los compiladores de lenguajes lógicos son interpretes capaces de llevar a cabo el proceso de inferencia con el fin de que los programas lógicos se ejecuten.
- Cada alternativa de respuesta da lugar a una ramificación adicional que deberá ser explorada para obtener todas las soluciones alternativas al problema.

PALABRAS CLAVES PROGRAMACIÓN LÓGICA



Sintaxis

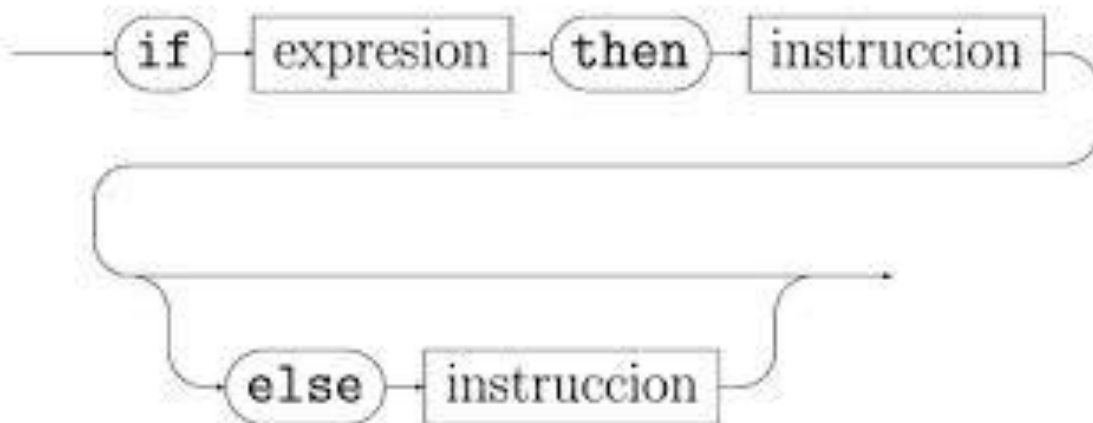
Conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.



Semántica

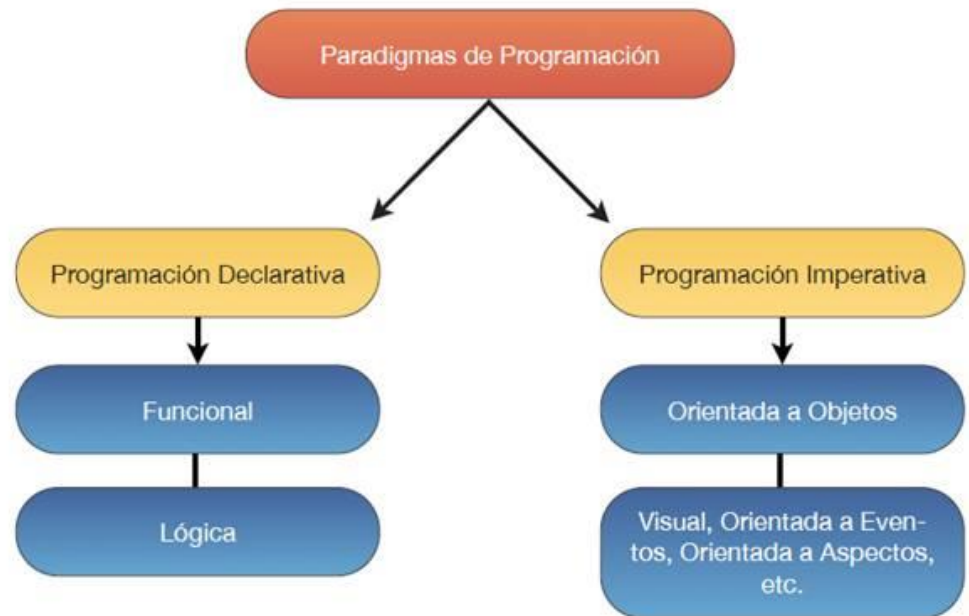
Es la interpretación del significado de una expresión.

if then else



Paradigmas de programación

Son la forma que determinan los métodos y las herramientas que un programador usará en la construcción de un software.



Programación imperativa o procedural

Consiste en determinar qué datos son requeridos para el cálculo, asociar a esos datos unas direcciones de memoria, y efectuar paso a paso una secuencia de transformaciones en los datos almacenados, de forma que el estado final represente el resultado correcto.



Programación declarativa

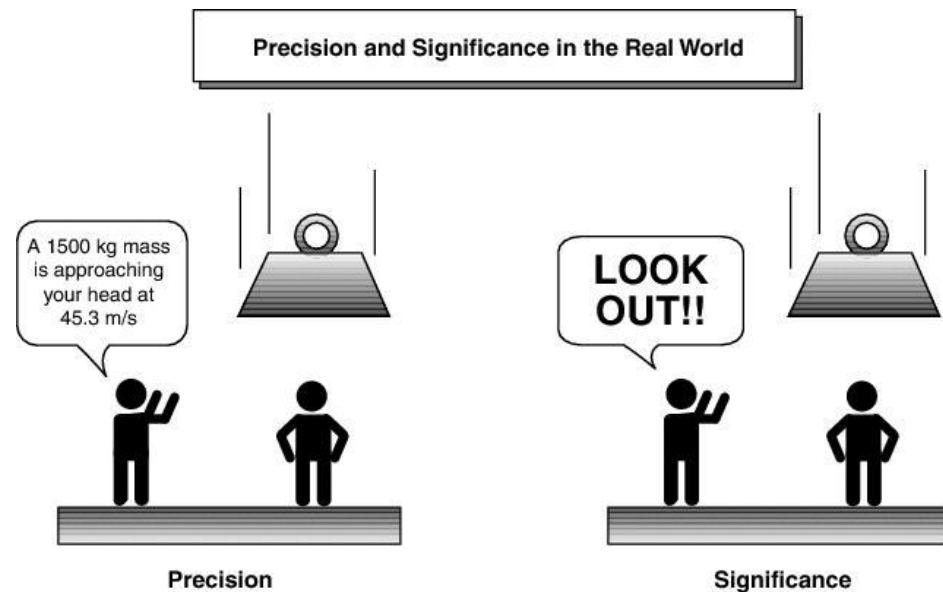
Es un estilo de programación en el que el programador especifica qué debe computarse y, además, cómo debe hacerse.

Programa = Lógica + Control

El componente lógico determina el significado, mientras que el componente de control solo determina su eficiencia.

Así, la tarea de programar se centra en la lógica, puesto que se asume el control automático a la máquina.

La característica fundamental es el uso de la lógica como lenguaje de programación.



Programación lógica

Consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño de lenguajes de programación; no debe confundirse con la disciplina de la lógica computacional.

Válido	Inválido	
Siempre verdadero	A veces V o F	Siempre falso
Satisfacible	Insatisfacible	

Ventajas y desventajas de la programación lógica

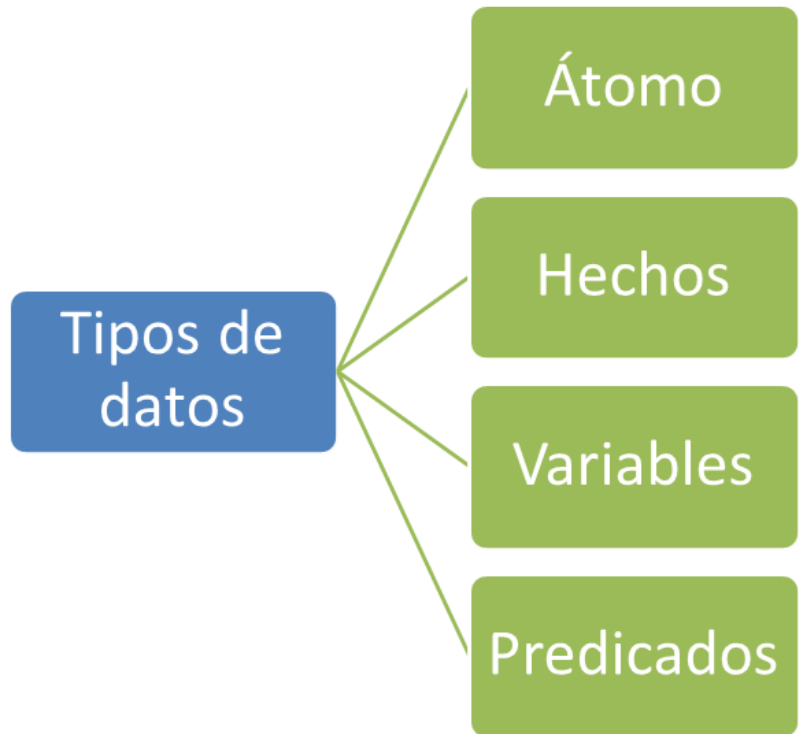
Ventajas

- Simplicidad.
- Cercanía a las especificaciones del problema realizada con lenguajes formales.
- Sencillez, potencia y elegancia.
- Metodología rigurosa de especificación.
- Sencillez en la implementación de estructuras complejas.

Desventajas

- Poco eficientes.
- Poco utilizado en aplicaciones reales.

En la programación lógica necesitamos especificar los pasos del programa como en un lenguaje de programación iterativo, sino que vamos a ir construyendo nuestro programa con ciertos elementos, como átomos, predicados, hechos y variables.



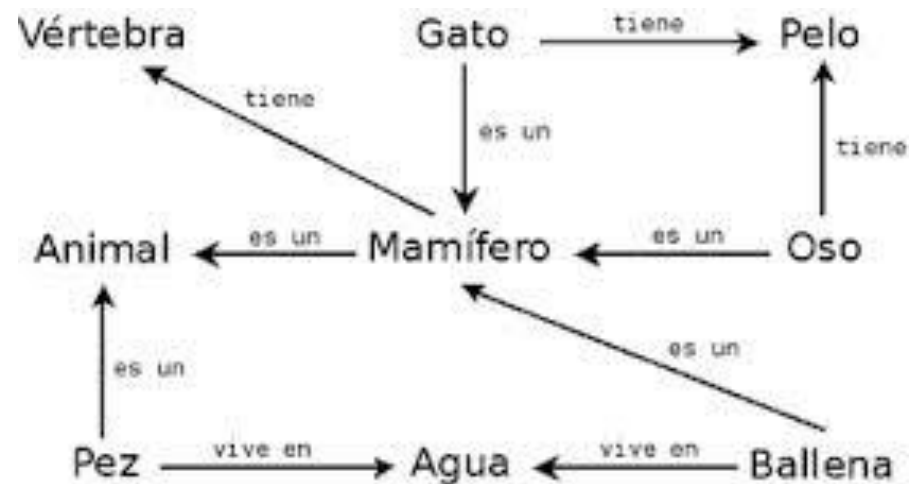
Tipo de datos simples

Átomos: Es una definición genérica de un objeto del mundo que queremos representar.

Predicados: Nos permite especificar características de los objetos de nuestro mundo o las relaciones entre ellos.

Hechos: Es algo que está ocurriendo en nuestro mundo, característica o relación entre objetos.

Variables: Representa el valor de un átomo en la programación declarativa.



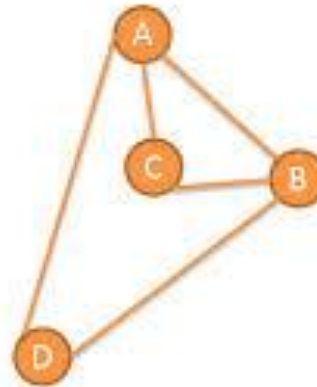
Tipos de datos estructurados

Son árboles, grafos y demás estructuras de datos no lineales a partir de listas.

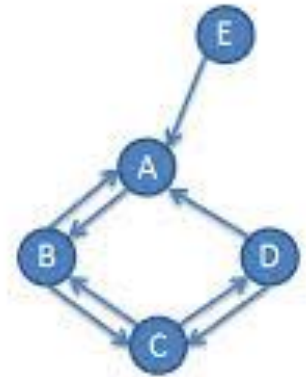
Las listas en Prolog se dividen en dos partes:

- **Cabeza:** Contiene el primer elemento de la lista, ya sea un átomo o una estructura
- **Cola:** Contiene el resto de los elementos de una lista.

Grafo No Dirigido



Grafo Dirigido



Match de un objeto

La operación más importante sobre los términos es la de *matching* (empatamiento).

Dados dos términos cualesquiera decimos que "empatan" si se cumple lo siguiente :

- Son idénticos.
- Las variables en ambos términos pueden instanciarse a objetos de tal modo que, después de la sustitución de las variables por estos objetos, los términos puedan ser idénticos.



PRÁCTICA CALIFICADA 02



**Universidad
Tecnológica
del Perú**