

UNIVERSIDAD PRIVADA-DE-TACNA



INGENIERIA DE SISTEMAS

TITULO:

INFORME DE LABORATORIO No 01

CURSO:

BASE DE DATOS II

DOCENTE(ING):

Patrick Cuadros Quiroga

Integrantes:

Orlando Antonio Mostacero Ortiz	(2015052775)
Orestes Ramirez Ticona	(2015053236)
Marlon Xavier Villegas Arando	(2015053890)
Nilson Laura Atencio	(2015053846)
Roberto Zegarra Reyes	(2010036175)

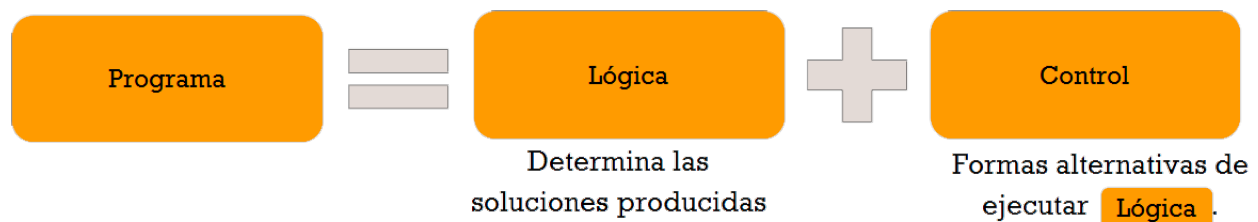
Índice

1. Paradigma No 03 – Programación Logica	1
2. Paradigma No 04 – Programación Funcional	4

1. Paradigma No 03 – Programación Lógica

1. Programación Lógica:

La programación lógica es un paradigma que se encuentra dentro del paradigma de la programación funcional. Aunque no es tan conocido como otros paradigmas de programación, es realmente interesante. Se basa en la declaración de hechos y reglas que permiten ir creando lo que para nosotros sería el conocimiento. Aunque inicialmente sea un poco complejo de entender, la programación lógica trabaja de forma muy similar a los humanos en cuanto al manejo de información y conocimientos se refiere. Veamos un ejemplo para entender mejor cómo es la declaración de las reglas y hechos Paradigma de programación basado en la lógica de primer orden. La Programación Lógica estudia el uso de la lógica para el planteamiento de problemas y el control sobre las reglas de inferencia para alcanzar la solución automática. La Programación Lógica, junto con la funcional, forma parte de lo que se conoce como Programación Declarativa, es decir la programación consiste en indicar como resolver un problema mediante sentencias, en la Programación Lógica, se trabaja en una forma descriptiva, estableciendo relaciones entre entidades, indicando no como, sino que hacer, entonces se dice que la idea esencial de la Programación Lógica es



Asume que partimos de un conjunto de hechos y reglas conocidos. Solamente es la declaración del componente lógico de un algoritmo. El sistema desarrolla el componente de control de secuencia. En este paradigma, la evaluación asume que cuando se selecciona una regla, es porque ésta es la única posibilidad o la necesaria para resolver el problema. Es decir, se encuentra una solución si un conjunto de reglas adecuado y las sustituciones a dichas reglas producen un conjunto de reglas aterrizadas (sin variables libres), suficientes para deducir el resultado de los hechos conocidos. La programación lógica intenta resolver lo siguiente: Dado un problema S, saber si la afirmación A es solución o no del problema o en que casos lo es. Además queremos que los métodos sean implantados en máquinas de forma que la resolución del problema se haga de forma automática. La programación lógica: construye base de conocimientos mediante reglas y hechos

Características

2. Los programas para los lenguajes de programación lógicos son un conjunto de hechos y reglas.
3. La sintaxis de los lenguajes de programación lógicos es notablemente diferente de los lenguajes de programación imperativos.
4. Unificación de términos.

5. Mecanismos de inferencia automática.
6. Recursión como estructura de control básica.
7. Visión lógica de la computación.
8. La aplicación de las reglas de la lógica para inferir conclusiones a partir de datos.
9. El programa se transforma en un conjunto de declaraciones formales de especificaciones que deben ser correctas por definición.
10. No tiene un algoritmo que indique los pasos que detallen la manera de llegar a un resultado.
11. Las salidas son funcionalmente dependientes de las entradas.

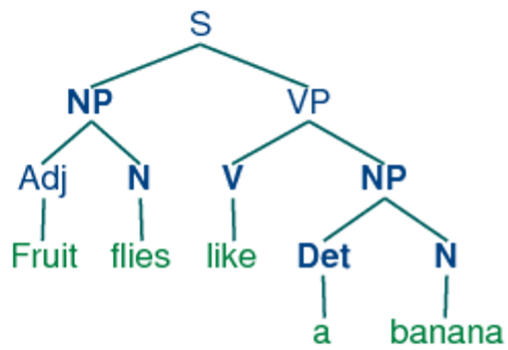
Ventajas y Desventajas del uso de este paradigma

Ventajas

12. Puede mejorarse la eficiencia modificando el componente de control sin tener que modificar la lógica del algoritmo.
 13. Relaciones multipropósito.
 14. Simplicidad.
 15. Generación rápida de prototipos e ideas complejas.
 16. Sencillez en la implementación de estructuras complejas.
 17. Potencia.
- Desventajas
18. Altamente ineficiente.
 19. Pocas áreas de aplicación
 20. No existen herramientas de depuración efectivas.
 21. En problemas reales, es poco utilizado.
 22. Si el programa no contiene suficiente información para contestar una consulta responde false.

Ejemplo

De forma tradicional imperativa tendríamos que especificar como lo haremos



```

oracion(O) :- sintagma_nominal(SN),
               sintagma_verbal(SV),
               append(SN,SV,O).
sintagma_nominal(SN) :- nombre(SN).
sintagma_nominal(SN) :- articulo(A),
               nombre(N),
               append(A,N,SN).
sintagma_verbal(SV) :- verbo(V),
               sintagma_nominal(SN),
               append(V,SN,SV).

articulo([el]).
nombre([gato]).
nombre([perro]).
nombre([pescado]).
nombre([carne]).
verbo([come]).

```

```

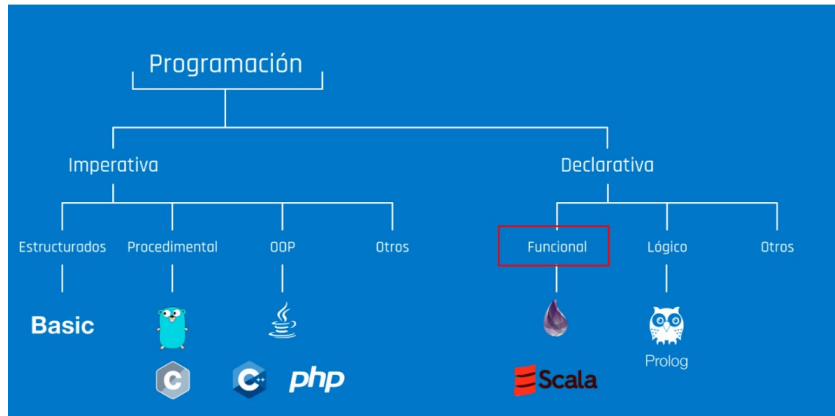
7 ?- oracion([perro,come,pescado]).
true .

```

2. Paradigma No 04 – Programación Funcional

1. Programacion funcional:

La Programación Funcional es una forma en la cual podemos resolver diferentes problemáticas ya que estaremos trabajando principalmente con funciones, evitaremos los datos mutables, así como el hecho de compartir estados entre funciones.



Las funciones serán tratadas como ciudadanos de primera clase. Las funciones podrán ser asignadas a variables además podrán ser utilizadas como entrada y salida de otras funciones

Ejemplo

De forma tradicional imperativa tendríamos que especificar como lo haremos

```
List<Integer> numeros = Arrays.asList(18, 6, 4, 15, 55, 78, 12, 9, 8);  
  
//Imperativo  
int contador = 0;  
for(int i=0; i<numeros.size(); i++) {  
    int numero = numeros.get(i);  
    if(numero > 10) {  
        contador++;  
    }  
}  
System.out.println(contador);
```

Pero de forma declarativa o programación funcional tendríamos

```
List<Integer> numeros = Arrays.asList(18, 6, 4, 15, 55, 78, 12, 9, 8);  
  
//Declarativo  
Long result = numeros.stream().filter( n -> n>10).count();  
System.out.println(result);
```