Instituto Tecnológico Superior de Jerez

Ingeniería en Sistemas Computacionales

Programación Lógica y Funcional

10 semestre

Tema 1.- Conceptos Fundamentales

Mapa Conceptual.- Paradigma de programación

Alumno: Esteban Faustino Muñoz Hidalgo

E-mail: faustino10.96@gmail.com

No. De Control: S15070115

MTI. Salvador Acevedo Sandoval

07/02/2020.

1.- ¿Qué es un paradigma de programación?

R= Un paradigma de programación indica un método de realizar cómputos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa.

2.- ¿Qué paradigmas de programación existen?

R= Paradigma imperativo, Paradigma declarativo, Paradigma lógico, Paradigma funcional, Paradigma orientado a objetos, Paradigma dirigido por eventos, Paradigma orientado a aspectos.

3.- ¿Cuáles son las características que definen a cada uno de ellos?

R= Paradigma imperativo: Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.

Este paradigma también se conoce como principio de ocultación de procedimientos y datos.

Consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible y manejable.

Se presenta históricamente como una evolución de la programación estructurada para solucionar problemas de programación más grandes y complejos de lo que ésta puede resolver.

Paradigma declarativo: Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos.

Paradigma lógico: El problema se modela con enunciados de lógica de primer orden.

Paradigma funcional: Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida.

Los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones.

La no existencia de asignaciones de variables y la falta de construcciones estructuradas como la secuencia o la iteración.

Existen dos grandes categorías de lenguajes funcionales: los funcionales puros y los híbridos.

En contraste, los lenguajes funcionales puros tienen una mayor potencia expresiva, conservando a la vez su transparencia referencial.

Haskell es un lenguaje de programación funcional, en este lenguaje podemos encontrar las características más significativas del paradigma funcional.

Paradigma orientado a objetos: El comportamiento del programa es llevado a cabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.

Abstracción: denota las características esenciales de un objeto, donde se capturan sus comportamientos.

Encapsulamiento: significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.

Modularidad: propiedad que permite subdividir una aplicación en partes más pequeñas.

Polimorfismo: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. **Java** es un lenguaje de programación orientada a objetos, en este lenguaje podemos encontrar muchas de las características que conforman a la programación orientada a objetos.

Paradigma dirigido por eventos: El flujo del programa está determinado por sucesos externos (por ejemplo, una acción del usuario).

Paradigma orientado a aspectos: Apunta a dividir el programa en módulos independientes, cada uno con un comportamiento bien definido.

4.- Ejemplos de código de dichos paradigmas:

R=

Fibonacci en Haskell

```
fibonacci 0 = 0

fibonacci 1 = 1

fibonacci n = fibonacci ( n - 1) + fibonacci ( n - 2 )
```

Fibonacci en Java

```
public class fibonacci{
    static int fibonacci ( int n ){
        int f = -1; // -1 indica que la función falló
        if ( n == 0 || n == 1)
            f = n;
```

```
        else if ( n > 1 )
            f = fibonacci ( n - 1 ) + fibonacci ( n - 2 );
        return f;
    }
}
```

5.- Aplicaciones reales donde se implementan dichos paradigmas:

Programación funcional

* Diseño de computadoras.

* Concurrencia y sistemas paralelos.

* Sistemas altamente seguros.

6.- Ventajas y Desventajas de cada paradigma:

R=

Paradigma funcional

Ventajas:

Ausencia de efectos colaterales.

Proceso de depuración menos problemático.

Pruebas de unidades más confiables.

Mayor facilidad para la ejecución concurrente

Desventajas:

Falta de estandarización.

Bajo rendimiento de los programas.

Paradigma imperativo

Ventajas:

Al aplicar el paradigma imperativo, un problema complejo debe ser dividido en varios subproblemas más simples, y estos a su vez en otros subproblemas más simples.

En caso de que un módulo necesite de otro, puede comunicarse con éste mediante una interfaz de comunicación que también debe estar bien definida.

Es fácil de mantener y modificar.

Es más fácil de escribir y depurar.

Facilidad de controlar es decir descompone un problema en estructuras jerárquicas, de modo que se puede considerar cada estructura desde dos puntos de vista.

Desventajas

No se dispone de algoritmos formales de modularidad, por lo que a veces los programadores no tienen claras las ideas de los módulos.

El Paradigma imperativo requiere más memoria y tiempo de ejecución.

Paradigma orientado a objetos

Ventajas

Permite crear sistemas más complejos.

Agiliza el desarrollo de software.

Proporciona conceptos y herramientas con las cuales se modela y representa el mundo real tan fielmente como sea posible.

Fomenta la reutilización y extensión del código.

Desventajas

Complejidad para adaptarse.

Mayor cantidad de código

7.- ¿Qué es una función (desde el punto de vista matemático)?

R= Una función matemática es una relación que se establece entre dos conjuntos, a través de la cual a cada elemento del primer conjunto se le asigna un único elemento del segundo conjunto o ninguno.

8.- ¿Qué es la programación funcional?

R= La programación funcional nos es más que un paradigma de programación, es decir, es una forma en la cual podemos resolver diferentes problemáticas. Cuando

nos encontramos desarrollamos software utilizando este paradigma, estaremos trabajando principalmente con funciones, evitaremos los datos mutables, así como el hecho de compartir estados entre funciones.

Con este paradigma 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.

A las funciones que puedan tomar funciones como parámetros y devolver funciones como resultado serán conocidas como función de orden superior.

La programación funcional es un paradigma declarativo. Nos enfocaremos en "qué" estamos haciendo y no en "cómo" se está haciendo (enfoque imperativo). Esto quiere decir que nosotros expresaremos nuestra lógica sin describir controles de flujo; no usaremos ciclos o condicionales.

9.- ¿Qué es una expresión matemática?

R= Una expresión algebraica es una expresión matemática en la que se combinan números y letras.

$$3a + 2$$

Los números se denominan “coeficiente” y las letras “parte literal”.

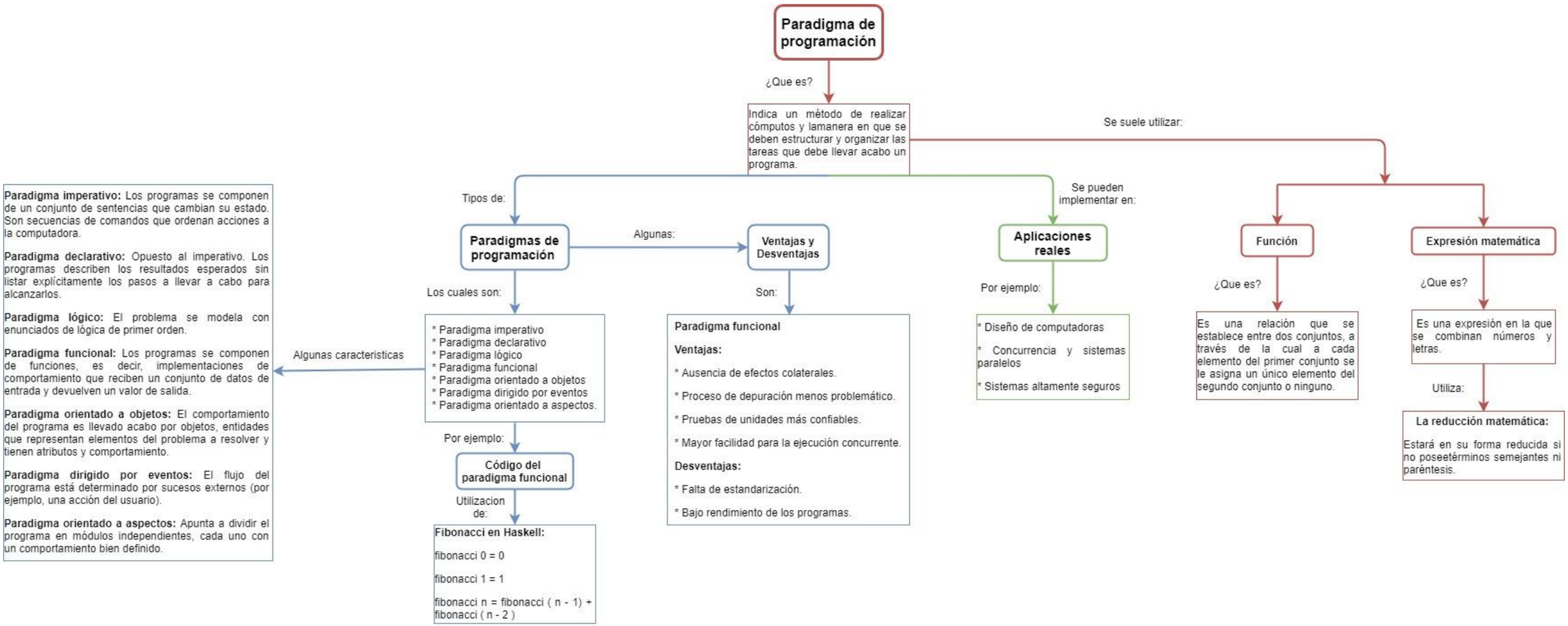
La letra “a” representa una incógnita, es decir una variable de la que desconocemos su valor y que hay que calcular. El número que acompaña a la letra la va multiplicando.

$$3a = 3 \times a$$

10.- ¿Qué es la reducción, simplicacion o evaluación de una expresión matemática?

R= En muchas ecuaciones tenemos términos que son semejantes, es decir, que poseen el mismo factor literal y muchas también poseen constantes, términos que no tienen una variable y que también son considerados semejantes entre ellos.

Una expresión algebraica estará en su forma reducida si no posee términos semejantes ni paréntesis.


Paradigma de programación
¿Que es?
Indica un método de realizar cómputos y lamanera en que se deben estructurar y organizar las tareas que debe llevar acabo un programa.
Se suele utilizar:
Tipos de:
Paradigmas de programación
Algunas:
Ventajas y Desventajas
Se pueden implementar en:
Aplicaciones reales
Función
Expresión matemática
Los cuales son:
* Paradigma imperativo
* Paradigma declarativo
* Paradigma lógico
* Paradigma funcional
* Paradigma orientado a objetos
* Paradigma dirigido por eventos
* Paradigma orientado a aspectos.
Algunas caracteristicas
Paradigma imperativo: Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.
Paradigma declarativo: Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos.
Paradigma lógico: El problema se modela con enunciados de lógica de primer orden.
Paradigma funcional: Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida.
Paradigma orientado a objetos: El comportamiento del programa es llevado acabo por objetos, entidades que representan elementos del problema a resolver y tienen atributos y comportamiento.
Paradigma dirigido por eventos: El flujo del programa está determinado por sucesos externos (por ejemplo, una acción del usuario).
Paradigma orientado a aspectos: Apunta a dividir el programa en módulos independientes, cada uno con un comportamiento bien definido.
Por ejemplo:
Código del paradigma funcional
Utilizacion de:
Fibonacci en Haskell:
fibonacci 0 = 0
fibonacci 1 = 1
fibonacci n = fibonacci (n - 1) + fibonacci (n - 2)
Son:
Paradigma funcional
Ventajas:
* Ausencia de efectos colaterales.
* Proceso de depuración menos problemático.
* Pruebas de unidades más confiables.
* Mayor facilidad para la ejecución concurrente.
Desventajas:
* Falta de estandarización.
* Bajo rendimiento de los programas.
Por ejemplo:
* Diseño de computadoras
* Concurrencia y sistemas paralelos
* Sistemas altamente seguros
¿Que es?
Es una relación que se establece entre dos conjuntos, a través de la cual a cada elemento del primer conjunto se le asigna un único elemento del segundo conjunto o ninguno.
¿Que es?
Es una expresión en la que se combinan números y letras.
Utiliza:
La reducción matemática:
Estará en su forma reducida si no poseetérminos semejantes ni paréntesis.

(Torres, 2019) (4rsoluciones, 2019) (definicion, 2019) (codigofacilito, 2019)

Bibliografía

4rsoluciones. (06 de 02 de 2020). *4rsoluciones*. Obtenido de 4rsoluciones: http://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/

codigofacilito. (06 de 02 de 2020). *codigofacilito*. Obtenido de codigofacilito: https://codigofacilito.com/articulos/programacion-funcional

definicion. (06 de 02 de 2020). *definicion*. Obtenido de definicion: https://definicion.de/funcion-matematica/

Torres, M. (06 de 02 de 2020). *michelletorres*. Obtenido de michelletorres: https://michelletorres.mx/paradigmas-de-programacion/