

INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

Instituto Tecnológico Superior de Jerez



Jerez de García Salinas

07/02/2020

Ivan Gamboa Ultreras

16070125

gamboita9@gmail.com

Ingeniería en Sistemas Computacionales

Programación Lógica y Funcional

8^{vo} Semestre

Actividad:

Mapa Conceptual: Paradigmas

ISC Salvador Acevedo Sandoval

1. ¿Qué es un paradigma de programación?

Un paradigma de programación indica un método de realizar cálculos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa. (Rodríguez, 2019)

Un paradigma de programación representa un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas. Es decir, un paradigma es una filosofía para la creación de programas. (cienciasfera, 2020)

Colección de patrones conceptuales (estructuras o reglas) que juntos modelan el proceso de diseño y que determinan en última instancia la estructura de los programas realizados. (uv, 2020)

2. ¿Qué paradigmas de programación existen?

Paradigma Imperativa: “La cual corresponde a el tipo de programación más usual. Describe cómo debe realizarse el cálculo, no el porqué. En otras palabras, es aquel tipo de programación donde se debe detallar paso a paso toda instrucción escrita desde un simple mensaje hasta un cálculo matemático para obtener un valor.” (studentplace98, 2018)

Paradigma Declarativo: “Describe que se debe calcular, sin explicitar el cómo. Los lenguajes de programación que aplican este paradigma están llenos de funciones y operaciones recursivas puramente matemáticas que se ejecutan para resolver una tarea.” (studentplace98, 2018)

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

Funcional:

- La computación se realiza mediante la evaluación de expresiones
- Definición de funciones
- Funciones como datos primitivos
- Valores sin efectos laterales, no existe la asignación
- Programación declarativa

Lógico:

- Definición de reglas
- Unificación como elemento de computación
- Programación declarativa

Imperativo:

- Definición de procedimientos
- Definición de tipos de datos
- Chequeo de tipos en tiempo de compilación
- Cambio de estado de variables
- Pasos de ejecución de un proceso

Orientada a Objetos:

- Definición de clases y herencia
- Objetos como abstracción de datos y procedimientos
- Polimorfismo y chequeo de tipos en tiempo de ejecución

4. Ejemplos de dichos paradigmas**Funcional:**

- LISP
- Scheme
- Haskell
- Scala
- Clojure

Orientada a Objetos:

- Java
- C++
- Ruby
- Python
- C#
- PHP

Lógico:

- Prolog
- Mercury
- Oz.

5. Aplicaciones reales donde se implementan dichos paradigmas**6. Ventajas y desventajas de cada 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

Lógico

Ventajas

- Descripciones independientes de la implementación (unificación semántica).
- Puede mejorarse la eficiencia modificando el componente de control sin tener que modificar la lógica del algoritmo.
- Base de conocimiento fácilmente escalable.
- Relaciones multipropósito.
- Expresión simple y precisa de los problemas.
- Generación rápida de prototipos e ideas complejas.
- Sencillez en la implementación de estructuras complejas.
- Potencia.

Desventajas

- Dependiendo del problema a solucionar, la implementación y el motor de inferencia, puede llegar a ser extremadamente ineficiente.
- Pocas y muy específicas áreas de aplicación.
- Existen muy pocas herramientas de depuración, en su mayoría poco efectivas.
- En problemas reales, es poco utilizado.
- Si el programa no contiene suficiente información para responder una consulta la respuesta puede ser una que se preste para malentendidos.
- Inferencia limitada por su base de conocimiento.

Imperativo

Ventajas

- Su relativa simplicidad y facilidad de implementación de los compiladores e intérpretes
- La capacidad de reutilizar el mismo código en diferentes lugares en el programa sin copiarlo.
- Una forma fácil seguir la pista de flujo del programa.
- La capacidad de ser muy modular o estructurado.
- Necesita menos memoria solamente.

Desventajas

- Los datos son expuestos a la totalidad del programa, así que no hay seguridad para los datos.
- Dificultad para relacionarse con los objetos del mundo real.
- Difícil crear nuevos tipos de datos reduce la extensibilidad.
- Se da importancia a la operación de datos en lugar de los datos mismos

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)?

El concepto general de función, aplicación o mapeo se refiere a una regla que asigna a cada elemento de un primer conjunto un único elemento de un segundo conjunto. Las funciones son relaciones entre los elementos de dos conjuntos. (wikipedia, 2020)

8. ¿Qué es la programación funcional?

“La programación funcional se deriva del paradigma funcional, un modelo matemático basado en el cálculo λ -cálculo que posibilita la construcción de soluciones simples basadas en funciones como núcleo básico de la programación” (BURITICÁ, 2013)

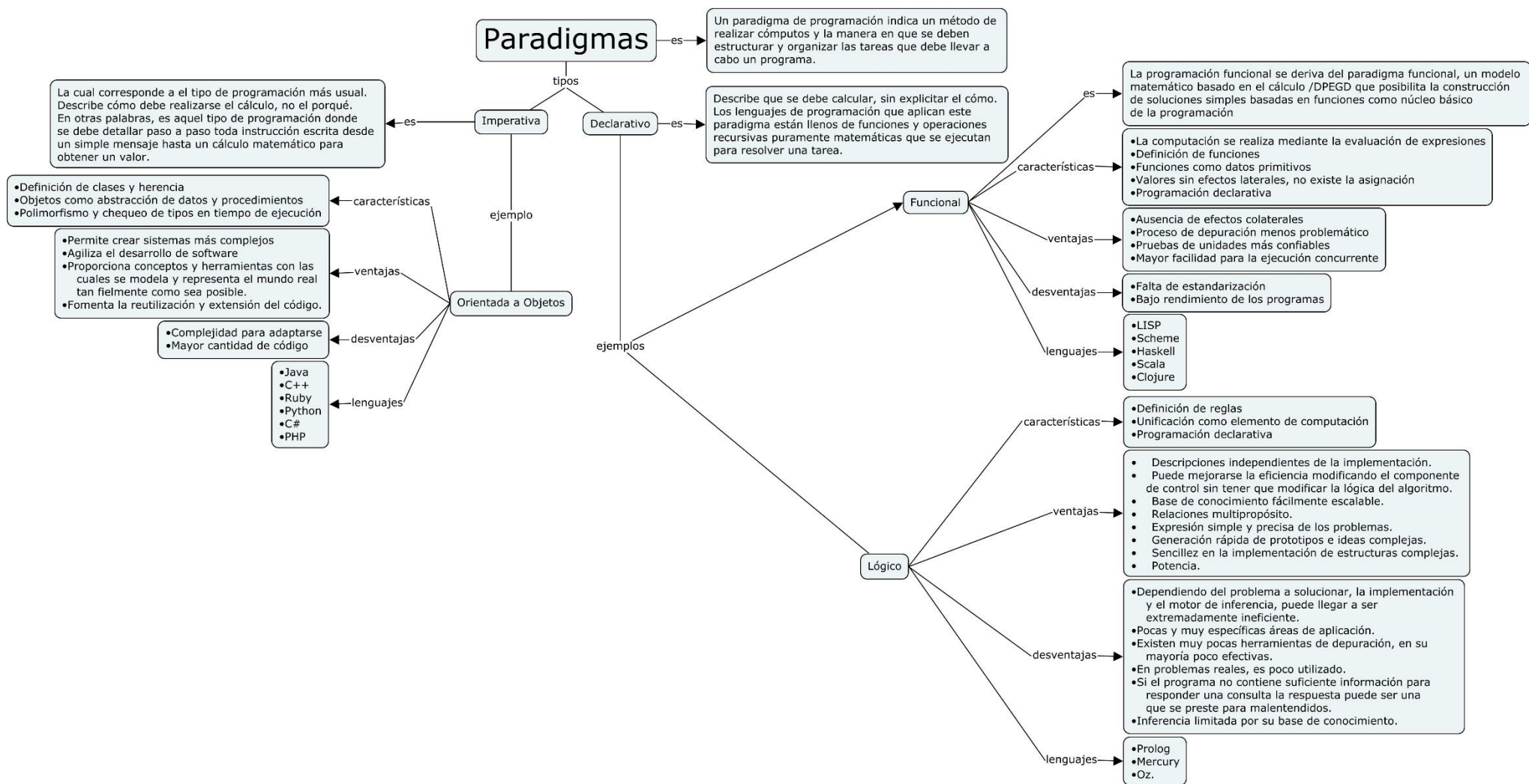
9. ¿Qué es una expresión matemática?

Es una ecuación que relaciona constantes o variables matemáticas y que se expresa mediante una igualdad matemática. (wikipedia, 2020)

Una expresión matemática es una combinación de símbolos que pueden designar números (constantes), variables, operaciones, símbolos de agrupación y otros signos de puntuación. (aaamath, 2020)

10. ¿Qué es la reducción (simplificación o evaluación) de expresiones matemáticas?

“Evaluar una expresión algebraica es el proceso de calcular el valor numérico de una expresión algebraica, cuando se sustituye el valor dado de la variable (letra) utilizando paréntesis y usando el orden de las operaciones.” (matematica7, 2020)



Referencias

aaamath. (28 de Enero de 2020). Obtenido de <https://www.aaamath.com/aa51.htm>

BURITICÁ, O. I. (23 de 11 de 2013). *redalyc*. Obtenido de <https://www.redalyc.org/pdf/2570/257031319008.pdf>

cienciasfera. (01 de 02 de 2020). Obtenido de http://cienciasfera.com/materiales/informatica/tecnologiainformacion/tema21/12_paradigmas_de_la_programacin.html

dccia. (09 de 02 de 2011). Obtenido de <http://www.dccia.ua.es/dccia/inf/asignaturas/LPP/2010-2011/clases-cristina/s2.pdf>

matematica7. (01 de 02 de 2020). Obtenido de <https://www.matematica7.com/evaluar-expresiones-algebraicas.html>

Rodríguez, C. V. (2019). *Paradigmas de programación*. Departamento de informática de la Universidad de Valladolid.

studentplace98. (19 de 12 de 2018). Obtenido de <https://studentplace98.blogspot.com/2018/12/fundamentos-de-programacion-paradigmas.html>

uv. (01 de 02 de 2020). Obtenido de <http://informatica.uv.es/guia/asignatu/LP/teoria/apuntes/cuatr1/tema1.pdf>

wikipedia. (28 de Enero de 2020). Obtenido de [https://es.wikipedia.org/wiki/F%C3%B3rmula_\(expresi%C3%B3n\)#En_Matem%C3%A1ticas](https://es.wikipedia.org/wiki/F%C3%B3rmula_(expresi%C3%B3n)#En_Matem%C3%A1ticas)

wikipedia. (20 de 01 de 2020). Obtenido de https://es.wikipedia.org/wiki/Funci%C3%B3n_matem%C3%A1tica