



Instituto Tecnológico Superior de Jerez

Jerez de García Salinas, Zacatecas.

Fecha: 31/01/2020

Alumno: Adán Ruiz Villalobos

Núm. Control: 16070137

Correo: adnruiz1@gmail.com

Ing. Sistemas Computacionales

Materia: Administración de Bases de Datos

Semestre: 8°

Actividad: Mapa Conceptual

Docente: I.S.C. Salvador Acevedo Sandoval

1. ¿Qué es un paradigma de programación?
Puede ser definido como un método para resolver un problema o para realizar alguna tarea. Un paradigma de programación es un enfoque para resolver problemas utilizando un lenguaje de programación o se puede decir que es un método para resolver un problema usando herramientas y técnicas que están disponibles siguiendo algún enfoque.
2. ¿Qué paradigmas de programación existen?
 - Imperativo: Esta se divide en tres grandes categorías:
 - Procesal
 - POO
 - Procesamiento Paralelo
 - Declarativo
 - Logic Programmin Paradigm
 - Database Processing Approach
3. ¿Cuáles son las características que define a cada uno de ellos?
 - a. Paradigma de programación imperativo: Uno de los más antiguos, presenta una relación estrecha con la arquitectura de la máquina. Consiste en varias declaraciones y después de la ejecución de todo el resultado se almacena.
 - Paradigma de programación procesal: Enfatiza el procedimiento en términos del modelo de maquina subyacente. Tiene la capacidad de reutilizar código.
 - Paradigma de programación POO: Se escribe como una colección de clases y objetos destinados a la comunicación. La entidad más pequeña y básica es el objeto y todo tipo de cálculo se realiza solo en los objetos. Se hace más hincapié en los datos en lugar del procedimiento. Puede manejar casi todo tipo de problemas de la vida real que hoy están en escena.
 - b. Paradigma de programación declarativo: La programación declarativa es un estilo de construcción de programas que expresa la lógica de la computación sin hablar de su flujo de control. A menudo considera los programas como teorías de alguna lógica. Puede simplificar la escritura de programas paralelos. La atención se centra en lo que hay que hacer, más bien en cómo se debe hacer, básicamente, enfatizar en lo que realmente está haciendo el código de código. Simplemente declara el resultado que queremos más bien cómo se ha producido.

- **Logic programming paradigms:** Se puede denominar como modelo abstracto de computación. Resolvería problemas lógicos como rompecabezas, series, etc. En la programación lógica tenemos una base de conocimiento que conocemos antes y, junto con la pregunta y la base de conocimiento que se da a la máquina, produce resultados. En los lenguajes de programación normales, dicho concepto de base de conocimiento no está disponible, pero al usar el concepto de inteligencia artificial, aprendizaje automático, tenemos algunos modelos como el modelo de Percepción que usa el mismo mecanismo.
- **Functional programming paradigms:** Los paradigmas de programación funcional tienen sus raíces en las matemáticas y son independientes del lenguaje. El principio clave de este paradigma es la ejecución de una serie de funciones matemáticas. El modelo central para la abstracción es la función destinada a un cálculo específico y no la estructura de datos.

4. Ejemplos de dichos paradigmas.

a. Paradigma de programación imperativo:

- C
- Fortan.
- Basic.

1. Paradigma de programación procesal:

- a. C
- b. C++
- c. Java
- d. ColdFusion
- e. Pascal

2. Paradigma de programación POO:

- a. Smula
- b. Java
- c. C++
- d. Onjective-C
- e. Visual Basic .Net
- f. Python

- g. Ruby
- 3. Paradigma de programación paralela:
- 5. Aplicaciones reales donde se implementen dichos paradigmas.
- 6. Ventajas y desventajas de cada paradigma.
 - a. Paradigma de programación imperativo:
 - Ventajas: Muy simple de implementar. Contiene bucles, variables, etc.
 - Desventajas: El problema complejo un puede resolverse. Menos eficiente y menos productivo. La programación paralela no es posible.
 - 1. Paradigma de programación procesal:
 - 2. Paradigma de programación POO:
 - a. Data security
 - b. Inheritance
 - c. Code reusability
 - d. Flexible and abstraction is also present
 - b. Paradigma de programación declarativa:

A pesar de lo anterior existen algunas ventajas en el uso de la programación declarativa. Entre las ventajas se destaca que la solución de un problema se puede realizar con un nivel de abstracción considerablemente alto, sin entrar en detalles de implementación irrelevantes, lo que hace a las soluciones más fácil de entender por las personas.

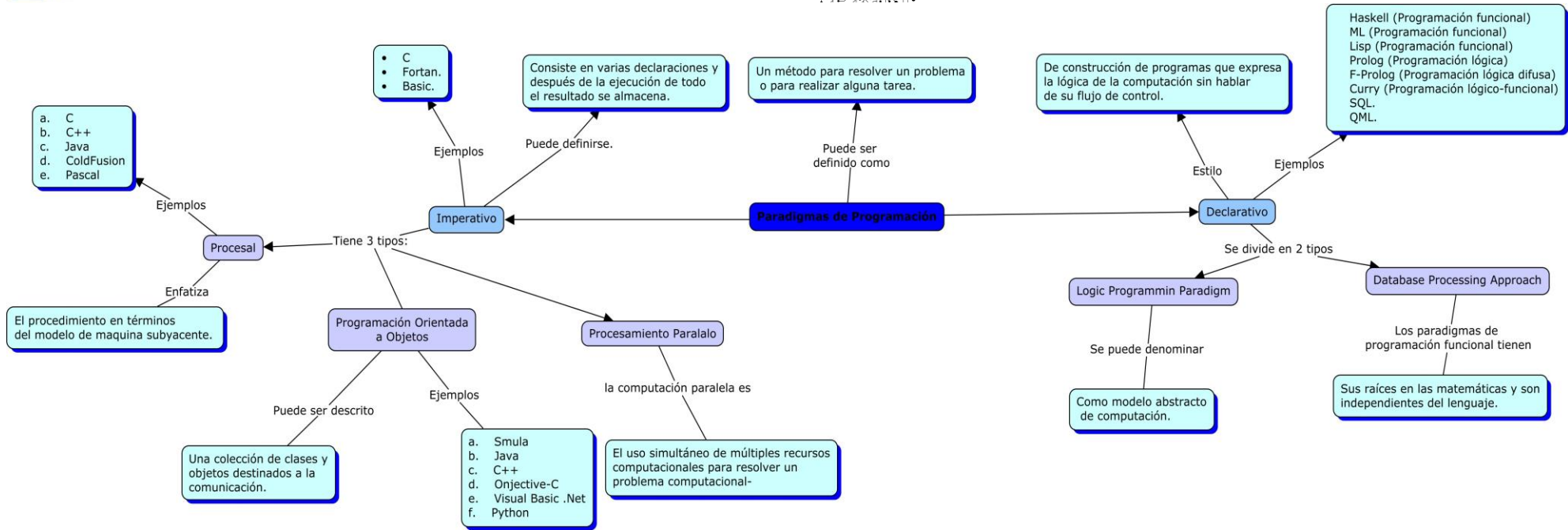
(Rani, 2018)

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

Una función (f) es una relación entre un conjunto dado X (llamado dominio) y otro conjunto de elementos Y (llamado codominio) de forma que a cada elemento x del dominio le corresponde un único elemento $f(x)$ del codominio (los que forman el recorrido, también llamado rango o ámbito). (htt141)
- 8. ¿Qué es la programación funcional?

La programación funcional nos es más que un paradigma de programación, es decir, es una forma en la cual podemos resolver diferentes problemáticas. (Pérez., 2019)
- 9. ¿Qué es una expresión matemática?

Números, símbolos y operadores (como $+$ y \times) agrupados para mostrar el valor de algo.





Referencias

(s.f.). Obtenido de https://www.profesorenlinea.cl/matematica/Funciones_matematicas.html

Pérez., E. I. (04 de 10 de 2019). *CodigoFacilito*. Obtenido de <https://codigofacilito.com/articulos/programacion-funcional>

Rani, B. (09 de 2018). *GeekForGeeks*. Obtenido de <https://www.geeksforgeeks.org/introduction-of-programming-paradigms/>