

Instituto Tecnológico Superior de Jerez
Jerez De García Salinas, Zac.

Alumno:

Castillo Landeros Miguel Eduardo

Núm. Control:

17070001

Ing. En Sistemas Computacionales.

Materia:

Programación Lógica y Funcional.

Semestre:

8

Actividad:

Cuestionario y Mapa Conceptual.

Docente:

ISC Salvador Acevedo Sandoval

Fecha:

7 de febrero 2020.

Actividad 1: Cuestionario

1. ¿Qué es un paradigma de programación?

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite producir programas con un conjunto de normas específicas, tales como la estructura.

2. ¿Qué paradigmas de programación existen?

- a. **Imperativo.**
- b. **Declarativo.**
- c. **Lógico.**
- d. **Funcional.**
- e. **Orientado a objetos.**

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

- a. **Imperativo.** Los programas se componen de un conjunto de sentencias que cambian su estado. Son secuencias de comandos que ordenan acciones a la computadora.
- b. **Declarativo.** Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos.
- c. **Lógico.** El problema se modela con enunciados de lógica de primer orden.
- d. **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.
- e. **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.

4. Ejemplo de dichos paradigmas

- a. **Imperativo.** Basic, C, PHP.
- b. **Declarativo.** F-prolog, curry, etc.,
- c. **Lógico.** Prolog
- d. **Funcional.** F#, Lisp, Clojure, Scheme, etc.,
- e. **Orientado a objetos.** C++, Java, etc.,

5. Aplicaciones reales donde se implementan dichos paradigmas

Desarrollo de Inteligencia artificial.

6. Ventajas y desventajas de cada paradigma

a. **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. Necesita menos memoria.

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. Se da importancia a la operación de datos en lugar de los datos mismos

b. **Declarativo.**

Ventajas: Consiste en que el indicar a la computadora "qué" tarea es la que tiene que hacer, en lugar de "cómo" hacerla nos protege de cambios en el contexto tecnológico.

Desventajas:

c. **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.

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.

d. **Funcional.**

Ventajas: Debido a que las funciones puras no cambian ningún estado y dependen completamente de la entrada, son fáciles de entender. Las funciones puras toman argumentos una vez y producen una salida inmutable.

Desventajas: Los valores inmutables combinados con la recursividad pueden conducir a una reducción en el rendimiento. En algunos casos, escribir funciones puras provoca una reducción en la legibilidad del código.

e. **Orientado a objetos.**

Ventajas: Son más fáciles de entender los códigos. La ejecución del programa es rápida y sencilla, todo se encuentra en una sola ventana.

Desventajas: Los programas no pueden ser moldeados enteramente por la programación orientada a objetos. Si se fuerza el lenguaje puede perder algunos objetos y características.

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

De manera más abstracta, el concepto general de función, aplicación o mapeo se refiere en matemáticas a una regla que asigna a cada elemento de un primer conjunto un único elemento de un segundo conjunto.

8. **¿Qué es la programación funcional?**

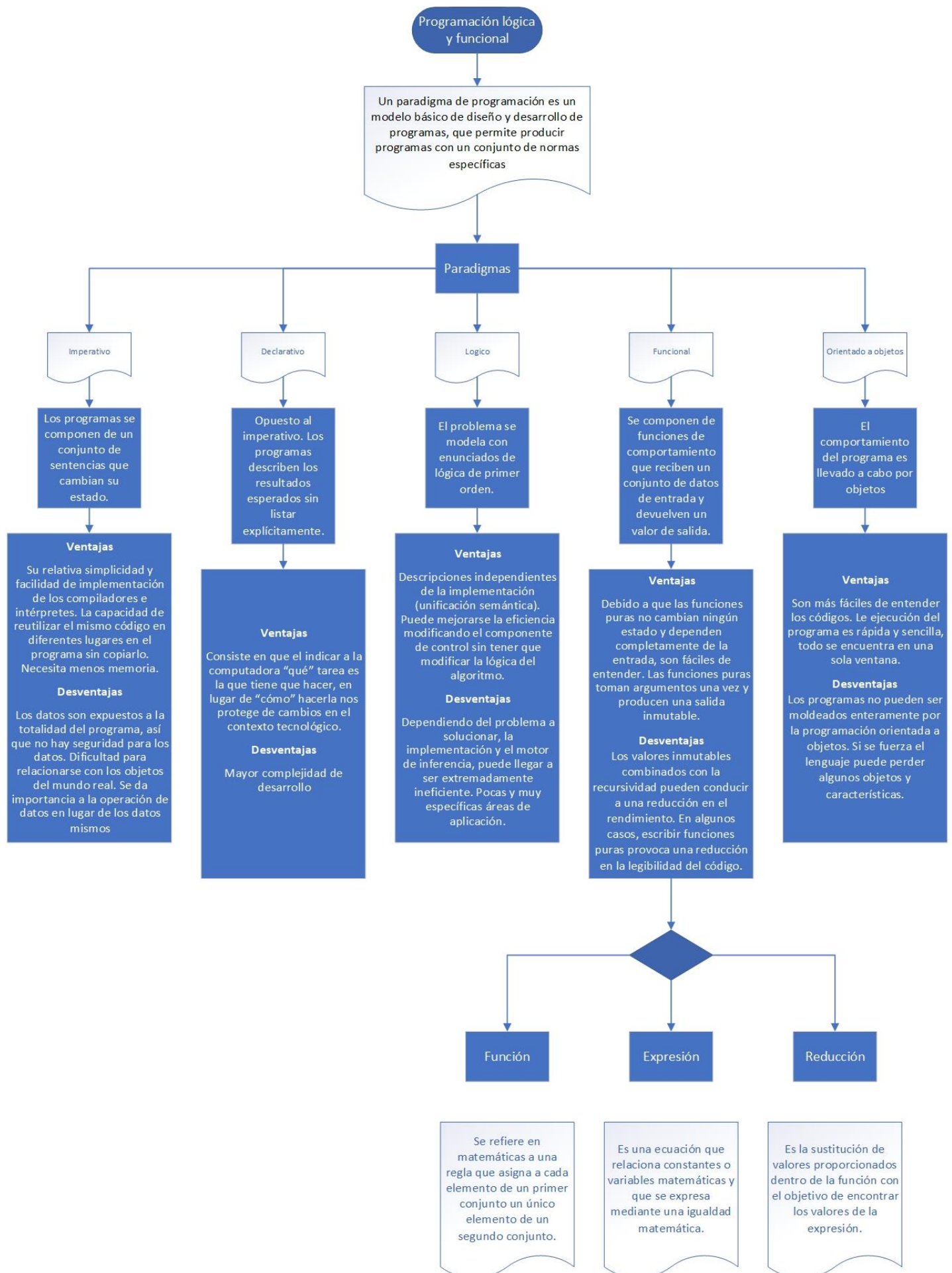
La programación funcional es un paradigma de programación declarativa basado en el uso de funciones matemáticas, en contraste con la programación imperativa, que enfatiza los cambios de estado mediante la mutación de variables.¹ La programación funcional tiene sus raíces en el cálculo lambda

9. **¿Qué es una expresión matemática?**

Es una ecuación que relaciona constantes o variables matemáticas y que se expresa mediante una igualdad matemática.

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

Es la sustitución de valores proporcionados dentro de la función con el objetivo de encontrar los valores de la expresión.



Bibliografía:

<https://www.monografias.com/trabajos107/paradigmas-programacion/paradigmas-programacion.shtml>

<https://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/>

https://www.ecured.cu/Programación_lógica

<http://ceciliaurbina.blogspot.com/2010/09/lenguajes-funcionales.html>

https://www.ecured.cu/Programación_declarativa

<https://medium.com/@Loopa/paradigmas-de-programación-programación-imperativa-y-programación-declarativa-4c4a4182fd87>

<https://es.slideshare.net/JFREDYOLAYARAMOS/paradigma-imperativo-39302522>

https://www.ecured.cu/Programación_Declarativa

<https://sg.com.mx/revista/24/programacion-declarativa>

https://ferestrepoca.github.io/paradigmas-de-rogramacion/proglogica/logica_teoría/lang.html

<https://pcweb.info/programacion-funcional-conceptos-ventajas-desventajas-y-aplicaciones/>

<https://codigofacilito.com/articulos/programacion-funcional>