

Paradigmas de Programación.

Programing Paradigms

Autor 1: Kevin Smith Muñoz Domicó

Risaralda, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: smith.munoz@utp.edu.co

Resumen— Un paradigma de programación representa un enfoque particular o filosofía para la construcción del software. No es mejor uno que otro sino que cada uno tiene ventajas y desventajas. También hay situaciones donde un paradigma resulta más apropiado que otro.

Palabras clave— Programación, Paradigmas, Funcional, Imperativo, Declarativo, Lógico, Orientado, Estructurada, Modular.

Abstract— a programming paradigm represents a particular approach or philosophy for software construction. One is not better than another but each has advantages and disadvantages. There are also situations where one paradigms is more appropriate than another.

Key Word — Programming, Paradigms, Functional, Imperative, Declarative, Logical, Oriented, Structured, Modular.

I. INTRODUCCIÓN

En este documento encontraras que es un paradigma de programación y cuáles son los principales paradigmas de programación.

II. CONTENIDO

A lo largo de la historia, el término «paradigma» fue objeto de muchas interpretaciones. En su origen griego, significaba «modelo», «ejemplo» o «patrón». Sobre este punto de partida, podemos hablar de un paradigma como un conjunto de creencias, prácticas y conocimientos que guían el desarrollo de una disciplina durante un período de tiempo. En diversas ramas de la ciencia, un conjunto de ideas en vigencia puede ser reemplazado drásticamente por otro que entre en conflicto con él y se demuestre más acertado. La programación tiene sus propios paradigmas, pero el término «paradigma de programación» no necesariamente representa un modelo único que deba ser respetado hasta que aparezca otro mejor. De hecho, actualmente muchos paradigmas coexisten en armonía.

Un paradigma de programación es un estilo de desarrollo de programas. Es decir, un modelo para resolver problemas computacionales. Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis

¿Cuáles son los principales paradigmas de programación?

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

PROGRAMACIÓN 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

PROGRAMACIÓN ESTRUCTURADA

Ventajas

- Reducción de los costos de mantenimiento
- Reducción del esfuerzo en las pruebas y depuración
- Los bloques de código son casi auto-explicativos, lo que reduce y facilita la documentación.

- Se incrementa el rendimiento de los programadores

Desventajas

- Reducción de los costos de mantenimiento
- Reducción del esfuerzo en las pruebas y depuración
- Los bloques de código son casi auto-explicativos, lo que reduce y facilita la documentación.
- Se incrementa el rendimiento de los programadores

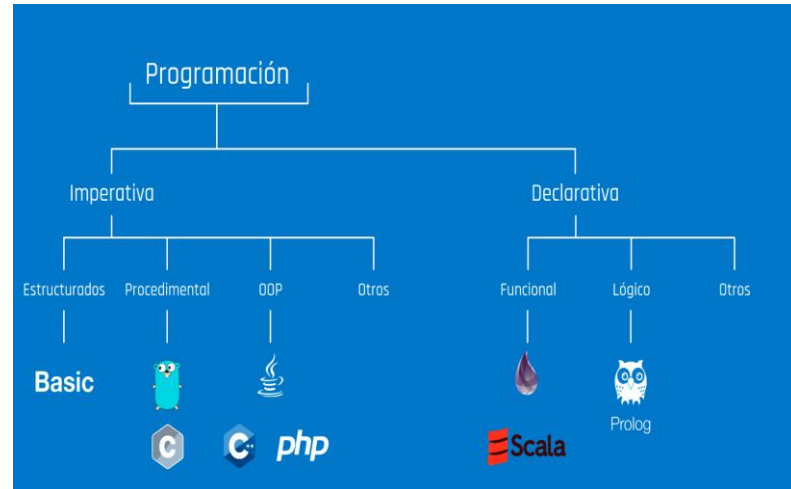
PROGRAMACIÓN MODULAR

Ventajas

- Al aplicar la programación modular, 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
- La programación modular requiere más memoria y tiempo de ejecución



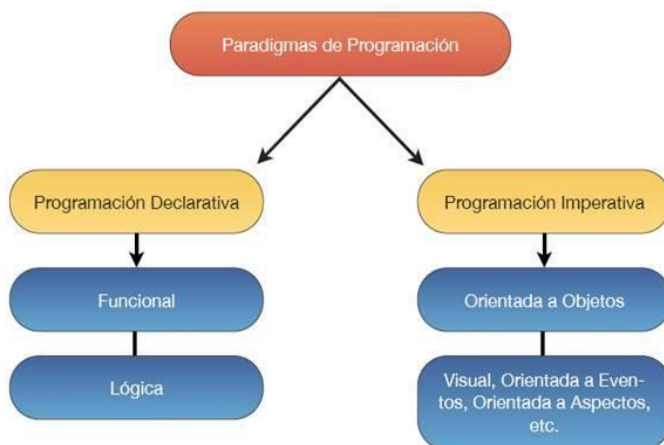
<https://studentplace98.blogspot.com/2018/12/fundamentos-de-programacion-paradigmas.html>

III. Referencias

<http://desarrollo--software.blogspot.com/2013/01/caracteristicas-de-los-paradigmas-de.html>

[https://sandramarramirez.fandom.com/es/wiki/Paradigmas de Programaci%C3%B3n](https://sandramarramirez.fandom.com/es/wiki/Paradigmas_de_Programaci%C3%B3n)

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



<https://desarrollo.espino.info/la-alegria-de-programar/los-paradigmas-de-programacion.html>