Paradigmas de Programación

Programming Paradigms

Autor: Jorge Alberto Bedoya Montoya

*Universidad tecnológica de Pereira, Pereira, Colombia*

Correo-e: Jorge.bedoya1@utp.edu.co

***Resumen*—** La programación es el proceso iterativo (repetición) de escribir o editar código fuente. Dicha edición implica probar, analizar, perfeccionar, y coordinar con otros programadores en caso de ser un programa desarrollado en conjunto.

Estos lenguajes pueden usarse para crear programas cuya función sea controlar el comportamiento físico y lógico de una máquina, programas para expresar algoritmos con precisión, o para facilitar la comunicación humana entre otros.

Los paradigmas son propuestas adoptadas por una comunidad de programadores cuyo núcleo es incuestionable en cuanto a resolver problemas claramente.

***Palabras clave—Paradigma, resolver, datos, problema, lenguaje.***

***Abstract*—** Programming is the iterative (repetition) process of writing or editing the source code. This edition implies testing, analyzing, perfecting and coordinating with other programmers if it is a jointly developed program.

These languages ​​can allow to create programs whose function is to control the physical and logical behavior of a machine, to detect algorithms with precision or facilitate human communication among others.

Paradigms are specific problems for a community of programmers whose core is unquestionable in terms of solving problems clearly.

**Key Word —** **Paradigm, solving, data,** **problem, language.**

1. INTRODUCCIÓN

A lo largo de la historia se han ido diseñando y evolucionando variados lenguajes de programación en el panorama actual de la computación. Caracterizándose por la evaluación y construcción de reglas que permiten dar soluciones a determinados problemas.

En la actualidad los lenguajes de programación no son tan diferentes unos a otros, sin embargo, podemos conocer sus diferencias según su enfoque. De esta manera ya podríamos adentrarnos un poco más al concepto de “Paradigmas de Programación”.

En este documento se hablará sobre su concepto, tipos y evolución, lenguajes usados y algunos ejemplos que permitirán comprender de manera más clara y precisa los términos técnicos de los lenguajes y paradigmas de programación.

1. CONTENIDO

La palabra paradigma proviene del griego “paradeigma”, que significa modelo o patrón que servirá como ejemplo a seguir ante una situación dada.

Teniendo en cuenta esto, un paradigma de programación se definiría como la manera en que un programador proporciona, determina y ejecuta un programa para resolver un problema.

**¿Qué son los Paradigmas de Programación?**



Fig. . ¿Qué son los paradigmas de Programación? [1]

La diferencia entre estos paradigmas se hace a través de las técnicas que estos permiten o prohíben.

**Tipos:**

|  |  |
| --- | --- |
| Paradigma Operacional. | Paradigma imperativo |
|  | Paradigma orientado a objetos. |
| Paradigma Declarativo. | Paradigma Funcional. |
|  | Paradigma Lógico. |
|  | Paradigma Transformacional. |
|  | Paradigma Relacional. |
| Paradigma Demostrativo. | Paradigma de Inducción |
|  | Paradigma de redes de neuronas. |
|  | Paradigma Genético. |

Esquema . Tipos paradigmas de programación.

Paradigma Estructurado.

Paradigma guiado por eventos.

**Tipos de lenguajes:**

Lenguajes desarrollados para calculo numérico, Tales como Fortran, Mathematica y Matlab.

Lenguajes para sistemas, tales como C, C ++ y ensamblador.

Lenguajes para aplicaciones de Inteligencia Artificial, tales como Prolog y Lisp.



Fig. . Clasificación de Paradigmas. [2]

**Cronología de los lenguajes de programación.**

|  |  |  |  |
| --- | --- | --- | --- |
| Fortran | Algol | Lisp | Cobol |
| 1957 | **1958** | **1960** | **1960** |
|  |  |  |  |
| APL | **Simula** | **Basic** | **Prolog** |
| 1962 | **1962** | **1964** | **1970** |
|  |  |  |  |
| C | **Pascal** | **Modula** | **Smalltalk** |
| 1972 | **1975** | **1975** | **1983** |
|  |  |  |  |
| ADA | **C ++** | **Perl** | **Python** |
| 1983 | **1986** | **1887** | **1991** |
|  |  |  |  |
| Ruby | **Java** | **PHP** | **C #** |
| 1993 | **1995** | **1995** | **2000** |

Tabla . Cronología de los lenguajes de programación.

**Compiladores:** Es un programa que se caracteriza por leer un programa escrito en lenguaje fuente, este lo traduce a un lenguaje objeto y genera una lista de posibles errores que encuentre en su lenguaje fuente.

**Interpretes:** Es como un compilador, solo que la salida es una ejecución. El programa entrada se interpreta y ejecuta a la vez, su principal ventaja es que permite una fácil depuración, sus desventajas son lentitud de ejecución y mayor consumo de recursos.

Al usar un interprete este traduce el programa cuando lo lee, convirtiendo su código directamente en acciones mientras que si es compilado puede correr por sí solo, ya que ha sido transformado en lenguaje máquina.

**Paradigma Operacional:** Estos son lenguajes centrados en la acción como un conjunto de secuencias paso a paso que convierten los datos de entrada iniciales en datos de salida finales, incluye el paradigma imperativo y orientado a objetos.

**Paradigma Imperativo:**

Lenguajes: Fortran, cobol, basic, C, ada, pascal.

Ecuación: Algoritmos + Estructuras de Datos = Programas.

**Paradigma orientado a objetos:** Evolución al concepto objeto del lenguaje que sigue el paradigma imperativo.

Lenguajes: Smalltalk, Eiffel, C ++, java, Object pascal.

Ecuación: Objetos + Mensajes = Programas.

**Paradigma Demostrativo:** Este lenguaje se construye mediante hechos, reglas y restricciones que constituyan y permitan una solución.

Ejemplos: lenguajes funcionales, lógicos, transformacionales y relacionales.

**Paradigma Demostrativo:** En este tipo no es necesario especificar operaciones y restricciones para obtener la solución, también llamada programación por ejemplos.

No se especifican operaciones o procedimientos para dar una solución. En lugar de esto se le brinda al sistema problemas similares y a partir de este generaliza procedimentalmente una solución.

Ejemplos: lenguajes de inducción, redes de neuronas y programación genética.

REFERENCIAS

1. Mosqueira Rey, E. EDteam. (2019) Figura. *¿Qué son los paradigmas de Programación?* [**https://ed.team/comunidad/que-son-los-paradigmas-de-programacion**](https://ed.team/comunidad/que-son-los-paradigmas-de-programacion)
2. Wordpress.com (2016) Figura. *Paradigmas de programación* [**https://objetosweb.wordpress.com/2016/02/01/paradigmas-de-programacion/**](https://objetosweb.wordpress.com/2016/02/01/paradigmas-de-programacion/)
3. Refseek Departamento de investigación (s. f) *Estructura de datos y de la información.*

**quegrande.org/apuntes/EI/1/EDI/teoria/07-08/tad\_-\_introduccion.pdf**

1. Queesela.net (s. f) Paradigma

[**https://queesela.net/paradigma/**](https://queesela.net/paradigma/)

1. González Oscar. Academia.edu (s. f). *Programación 1* [**https://www.academia.edu/40311039/Programaci%C3%B3n\_I**](https://www.academia.edu/40311039/Programaci%C3%B3n_I)