PARADIGMAS DE LA PROGRAMACIÓN

PROGRAMMING PARADIGMS

Autor 1: Kenneth Kaled

Giraldo Marulanda

Autor 2: Kevin Alonso

Llanos Morales

*Ingeniería de sistemas y computación,*

*Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e:

k.giraldo1@utp.edu.co

k.llanos@utp.edu.co

**Resumen— los paradigmas de programación son los métodos los cuales son utilizados para realizar un programa de manera efectiva**

***Palabras clave—paradigma, programación, método***

***Abstract*— los paradigmas de programación son los métodos los cuales son utilizados para realizar un programa de manera efectiva**

***Keyword* —paradigm, programming, method**

**1. INTRODUCCIÓN**

Un paradigma de programación consiste en un método para llevar a cabo cálculos y la forma en la que deben estructurarse y organizarse las tareas que debe realizar un programa.

Se trata de una propuesta tecnológica adoptada por una comunidad de programadores, y desarrolladores en el que el objetivo principal trata de resolver uno o varios problemas claramente delimitados; la resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de software.

Representa un enfoque particular o filosofía para diseñar soluciones. Los paradigmas difieren unos de otros, en los conceptos y la forma de abstraer los elementos involucrados en un problema, así como en los pasos que integran su solución del problema, en otras palabras, el cómputo. Tiene una estrecha relación con la formalización de determinados lenguajes en su momento de definición. Es un estilo de programación empleado.

Un paradigma de programación está delimitado en el tiempo en cuanto a aceptación y uso, porque nuevos paradigmas aportan nuevas o mejores soluciones que lo sustituyen parcial o totalmente.

# **2. CONTENIDO**

1. Introducción

2. Contenido

3¿Qué es un paradigma?

3.1 lista de paradigmas

3.1.1 Imperativo

3.1.2 Declarativo

3.1.3 Lógico

3.1.4 Funcional

3.1.5 Orientados a objetos

3.1.6 Dirigidos por eventos

3.1.7 Orientados a aspectos

4. Ejemplos

5. Webgrafía

# **3. ¿Qué es un paradigma?**

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.

## **3.1 Lista de paradigmas**

### 3.1.1 Paradigma imperativo

Los primeros lenguajes imperativos fueron los lenguajes de máquina de los equipos originales. En estos lenguajes, las instrucciones eran muy simples, lo que hizo más fácil la aplicación de hardware, pero dificultaron la creación de programas complejos. FORTRAN, desarrollado por John Backus en IBM a partir de 1954, fue el primer lenguaje de programación principal para eliminar los obstáculos presentados por el código de la máquina en la creación de programas complejos.

año en que se creó: 1954

lenguaje utilizados: FORTRAN, COBOL, BASIC, ADA Y PASCAL.

### 3.1.2 Paradigma declarativo

Programación declarativa, en contraposición a la programación imperativa, es un paradigma de programación que está basado en el desarrollo de programas especificando o "declarando" un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución. La solución es obtenida mediante mecanismos internos de control, sin especificar exactamente cómo encontrarla (tan solo se le indica a la computadora qué es lo que se desea obtener o qué es lo que se está buscando). No existen asignaciones destructivas, y las variables son utilizadas con transparencia referencial.

### 3.1.3 Paradigma lógico

### Programación lógica. Consiste en la aplicación del corpus de conocimiento sobre [lógica](https://www.ecured.cu/L%C3%B3gica) para el diseño de [lenguajes de programación](https://www.ecured.cu/Lenguajes_de_programaci%C3%B3n); no debe confundirse con la disciplina de la [lógica computacional](https://www.ecured.cu/index.php?title=L%C3%B3gica_computacional&action=edit&redlink=1). La programación lógica es un tipo de [paradigmas de programación](https://www.ecured.cu/Paradigmas_de_programaci%C3%B3n) dentro del paradigma de [programación declarativa](https://www.ecured.cu/Programaci%C3%B3n_declarativa).

creador: alain colmerauer es el creador de la programación lógica. lenguaje prolog fue concebido por alain colmerauer en la universidad de aix marseille, francia, donde el idioma se implementó por primera vez en 1973

Año en que se creó: 1973

Lenguajes utilizados: ´prolog´

### 3.1.4 Programación funcional

Basada en la definición los predicados y es de corte más matemático, está representado por Scheme (una variante de Lisp) o Haskell. Python también representa este paradigma.

Este tipo de paradigma está basado en el uso de funciones matemáticas, que a su vez están constituidos únicamente por definiciones de funciones, entendiendo estas no como subprogramas clásicos de un lenguaje imperativo, sino como funciones puramente matemáticas.

### 3.1.5 Programación orientado a objetos

Programación orientada a objetos (POO). Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

se creò: 1967

creado : Kristen Nygaard y Ole-johan Dahl

### 3.1.6 Programación orientada a eventos

La Programación orientada a eventos es un paradigma de programación en el que la estructura y la ejecución de los programas van determinados por los sucesos o acciones que ocurren en el sistema, definidos por el usuario o por el propio sistema.

La programación orientada a eventos es muy fácil de usar y es adecuada para aquellas personas que tienen poco conocimiento en programación. Con los lenguajes orientados a eventos se pueden realizar en poco tiempo aplicaciones sencillas y muy funcionales, utilizando interfaces gráficas en las que se insertan componentes o controles a los que se le programan eventos. Dichos eventos permiten al usuario realizar una serie de acciones lógicas para un determinado programa.

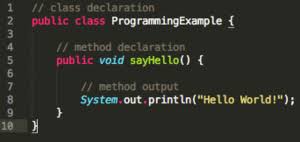
### 3.1.7 Programación orientada a aspectos

La Programación Orientada a Aspectos (POA): es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de conceptos. Gracias a la POA se pueden capturar los diferentes conceptos que componen una aplicación en entidades bien definidas, de manera apropiada en cada uno de los casos y eliminando las dependencias inherentes entre cada uno de los módulos.

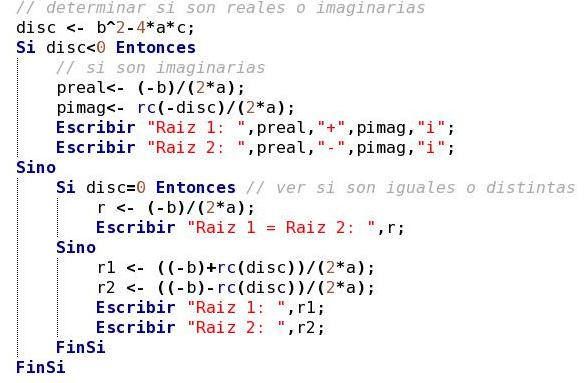
De esta forma se consigue razonar mejor sobre los conceptos, se elimina la dispersión del código y las implementaciones resultan más comprensibles, adaptables y reusables.

4. Ejemplos

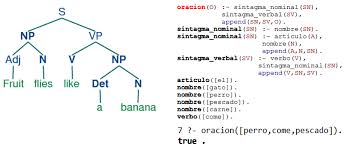
paradigma imperativa



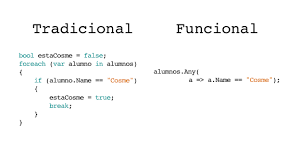
paradigma declarativa



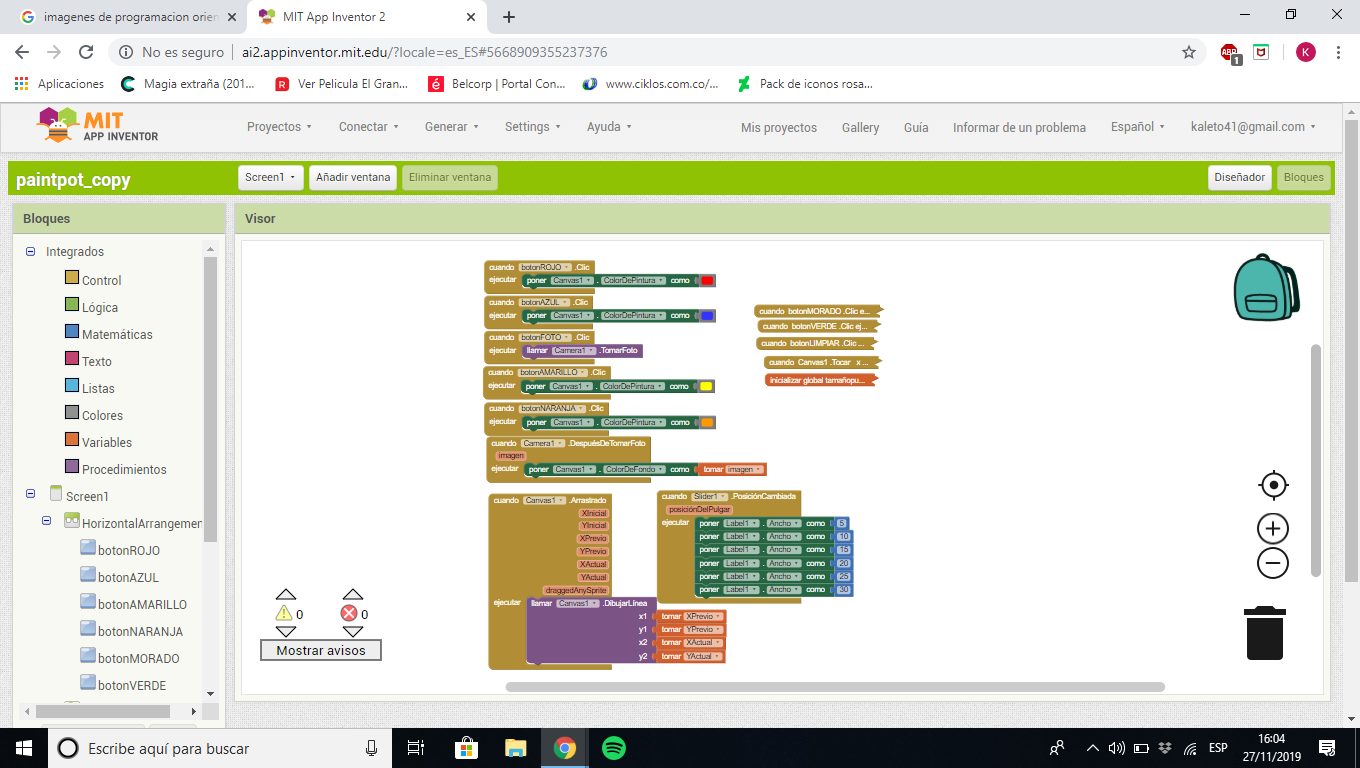
paradigma lògica



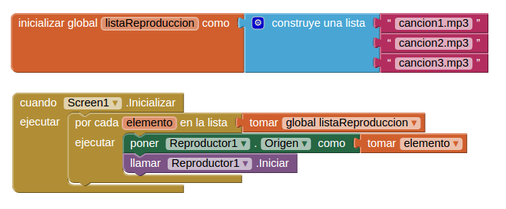
programaciòn funcional



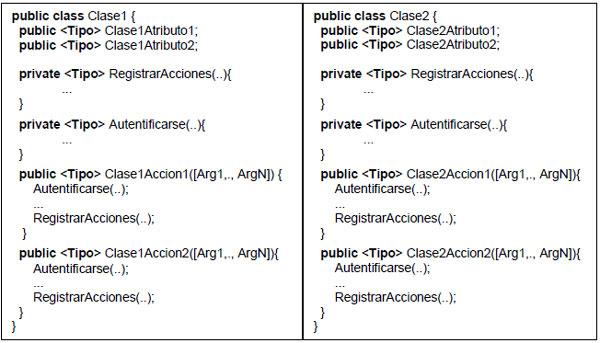
programciòn orientada a objetos



programaciòn orientada a eventos



programciòn orientada a aspectos



5. Biografías

3.1.1 [https://kevinldp.wordpress.com/paradigma-imperativ](https://kevinldp.wordpress.com/paradigma-imperativo/)/

3.1.2 <https://www.ecured.cu/Programaci%C3%B3n_declarativa>

3.1.4

<http://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional#Caracter%C3%ADsticas>

3.1.5

<https://www.ecured.cu/Programaci%C3%B3n_Orientada_a_Objetos>

3.1.6

<http://contenidos.sucerman.com/nivel3/dispositivos/unidad1/leccion2.html>

3.1.7

<https://www.ecured.cu/Programaci%C3%B3n_orientada_a_aspectos>