# PARADIGMAS DE PROGRAMACION

Autor: Emmanuel Vélez López

*Risaralda, Universidad Tecnológica De Pereira, Pereira, Colombia*

Correo-e: emmanuel.velez@utp.edu.co

**Resumen** — El trabajo consiste en determinar que es un paradigma de programación en el abarcaremos varios temas como sus ventajas sus desventajas y sus lenguajes esto permiten implementar, algo que tiene una relación directa con su sintaxis.

**Palabras clave—** Programas, Paradigmas, Datos, Lenguajes

**Abstract—** The work consists in determining that it is a programming paradigm in which we will cover several topics such as its advantages, its disadvantages and its languages, which allow it to be implemented, something that has a direct relationship with its syntax.

**Key Word—** Programs, Paradigms, Data, Languages

## I. INTRODUCCIÓN

Un paradigma de programación provee (y determina) la visión y métodos de un programador en la construcción de un programa o subprograma. Diferentes paradigmas resultan en diferentes estilos de programación y en diferentes formas de pensar la solución de problemas (con la solución de múltiples “problemas” se construye una aplicación).

II. CONTENIDO

1. QUE ES
2. VENTAJAS
3. DESVENTAJAS
4. LENGUAJES
5. EJEMPLO

1. QUE ES :

Un paradigma de programación consiste en un método para llevar a cabo cómputos y la forma en la que deben estructurarse y organizarse las tareas que debe realizar un programa.3​ Se trata de una propuesta tecnológica adoptada por una comunidad de programadores, y desarrolladores cuyo núcleo central es incuestionable en cuanto que únicamente 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.

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.

1. VENTAJAS :

* Ausencia de efectos colaterales
* Proceso de depuración menos problemático
* Pruebas de unidades más confiables
* Mayor facilidad para la ejecución concurrente
* La estructura de los programas es clara, puesto que las instrucciones están más ligadas o relacionadas entre sí.
* Los programas son más fáciles de entender, pueden ser leídos de forma secuencial.
* Un programa escrito de acuerdo a los principios de programación estructurada no solamente tendrá una mejor estructura sino también una excelente presentación.

1. DESVENTAJAS :

* Falta de estandarización
* Bajo rendimiento de los programas
* El principal inconveniente de este método de programación es que se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar problemático el manejo de su código fuente
* Se obtiene un único bloque de programa, que cuando se hace demasiado grande puede resultar difícil su manejo

**Programación imperativa o por procedimientos:**

Es el más usado en general, se basa en dar instrucciones al ordenador de cómo hacer las cosas en forma de algoritmos, en lugar de describir el problema o la solución. Las recetas de cocina y las listas de revisión de procesos, a pesar de no ser programas de computadora, son también conceptos familiares similares en estilo a la programación imperativa; donde cada paso es una instrucción.

**Programación declarativa:**

Está basada en describir el problema declarando propiedades y reglas que deben cumplirse, en lugar de instrucciones. Hay lenguajes para la programación funcional, la programación lógica, o la combinación lógico-funcional. 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).

**Programación multiparadigma:**

Es el uso de dos o más paradigmas dentro de un programa.

**Programación reactiva:**

Este paradigma se basa en la declaración de una serie de objetos emisores de eventos asíncronos y otra serie de objetos que se "suscriben" a los primeros (es decir, quedan a la escucha de la emisión de eventos de estos) y \*reaccionan\* a los valores que reciben. Es muy común usar la librería Rx de Microsoft (Acrónimo de Reactive Extensions), disponible para múltiples lenguajes de programación.

1. LENGUAJES :

El lenguaje Lisp se considera multiparadigma. Al igual que Python, que es orientado a objetos, reflexivo, imperativo y funcional.​ Según lo describe Bjarne Stroustrup, esos lenguajes permiten crear programas usando más de un estilo de programación. El objetivo en el diseño de estos lenguajes es permitir a los programadores utilizar el mejor paradigma para cada trabajo, admitiendo que ninguno resuelve todos los problemas de la forma más fácil y eficiente posible. Por ejemplo, lenguajes de programación como C++, Genie, Delphi, Visual Basic, PHP o D6​ combinan el paradigma imperativo con la orientación a objetos. Incluso existen lenguajes multiparadigma que permiten la mezcla de forma natural, como en el caso de Oz, que tiene subconjuntos (particularidad de los lenguajes lógicos), y otras características propias de lenguajes de programación funcional y de orientación a objetos. Otro ejemplo son los lenguajes como Scheme de paradigma funcional o Prolog (paradigma lógico), que cuentan con estructuras repetitivas, propias del paradigma imperativo.



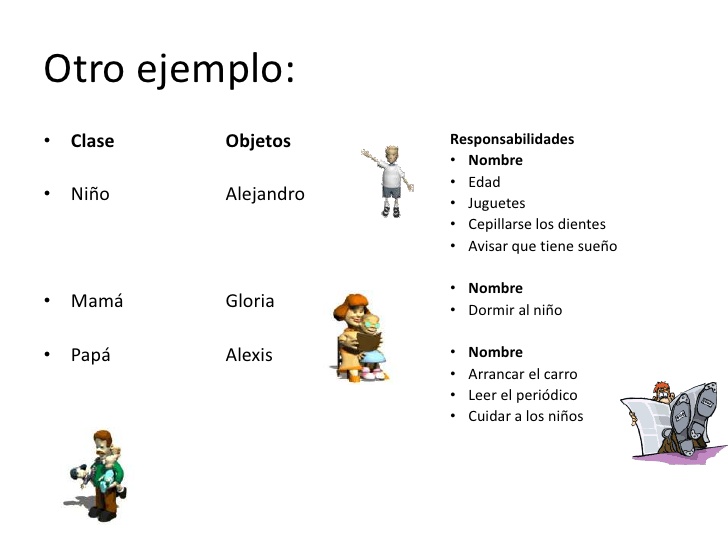
1. EJEMPLOS :

Tenemos a 10 personas en un cuarto y les proponemos una tarea, que salgan del cuarto en el que se encuentran, el único hecho es que el cuarto no tiene puertas o ventanas.

Una persona puede intentar por fuerza bruta romper una pared. Otra personas puede intentar escarbar para atravesar el muro, otra puede buscar un acertijo, etc.

En realidad tenemos algo que observar:

El problema está definido, es salir del cuarto. Cada quien puede abordar el problema como quiera, puede ser que si necesitamos agrupar las soluciones que intentan, el que intenta patear la pared y el que intenta golpear están utilizando el mismo paradigma: la fuerza bruta los que están intentando escarbar atraves del muro y los que intentan encontrar un acertijo están intentando utilizar otro paradigma y así podríamos clasificar las diferentes soluciones.



## III. CONCLUSIONES

En conclusión la mayoría de los lenguajes de programación son puramente textuales, es decir, utilizan secuencias de texto que incluyen palabras, números y puntuación, de manera similar a los lenguajes naturales escritos. Por otra parte, hay algunos lenguajes de programación que son más gráficos en su naturaleza, utilizando relaciones visuales entre símbolos para especificar un programa.



## IV. REFERENCIAS

<https://labsys.frc.utn.edu.ar/ppr-2009/Unidad%20I%20-%20POO%20Avanzada/Apunte/Unidad%20I%20-%20POO%20Avanzada.pdf>

<https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n#Paradigma_de_programaci%C3%B3n>

<https://ed.team/blog/paradigmas-de-programacion>