**Universidad Tecnológica de México**

**Verduzco López Rubén**

**Ing. Sistemas Computacionales**

**Programación Lógica**

**Jesús Barajas Natividad**

**Tarea 1.- Paradigmas**

****

**Paradigma**

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: estructura modular, fuerte cohesión, alta rentabilidad, etc. Los paradigmas pueden ser considerados como patrones de pensamiento para la resolución de problemas. Desde luego siempre teniendo en cuenta los lenguajes de programación, según nuestro interés de estudio. 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.

El concepto de paradigma puede referirse, de manera cotidiana, a una creencia u opinión compartida colectivamente. Sin embargo, este uso del concepto puede generar diversos errores e imprecisiones, al presentarse en ámbitos teóricos y científicos, ya que se refiere prácticamente a cualquier idea o creencia que tenga un sujeto o grupo de sujetos. Es recomendable no emplear este concepto en usos de la vida cotidiana, y dejarlo para discusiones de carácter epistemológico. En su lugar, cotidianamente se pueden usar cualesquiera de los conceptos arriba mencionados.

**Tipos de paradigmas de programación:**

Podemos clasificar los paradigmas de programación en:

Paradigma imperativo, heurístico, concurrente, funcional, lógico, paradigma basado en objetos.

**Paradigma imperativo:** Son aquellos que facilitan los cálculos por medio de cambios de estado, entendiendo como estado la condición de una memoria de almacenamiento. Los lenguajes estructurados en bloques, se refieren a los ámbitos anidados, es decir los bloques pueden estar anidados dentro de otros bloques y contener sus propias variables.

Los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones, entendiendo éstas no como subprogramas clásicos de un lenguaje imperativo, sino como funciones puramente matemáticas, en las que se verifican ciertas propiedades como la ''transparencia referencial'' (el significado de una expresión depende únicamente del significado de sus subexpresiones), y por tanto, la carencia total de ''efectos laterales''.

**Algunos de los lenguajes imperativos son:**

• C

• C++

• C#

• Basic

• Java

• Perl

**Paradigma heurístico:** Define un modelo de resolución de problemas en el que se incorpora algún componente heurístico, sobre la base de una representación más apropiada de la estructura del problema, para su resolución con técnicas heurísticas.

Se puede definir como "aquel tipo de programación computacional que aplica para la resolución de problemas reglas de buena lógica (reglas del pulgar).

Denominadas heurísticas, las cuales proporcionan entre varios cursos de acción uno que presenta visos de ser el más prometedor, pero no garantiza necesariamente el curso de acción más efectivo."

**Paradigma funcional:** Sus orígenes provienen del Cálculo Lambda (¿o? -cálculo), una teoría matemática elaborada por Alonzo Church como apoyo a sus estudios sobre compatibilidad. Un lenguaje funcional es, a grandes rasgos, un azúcar sintáctico del Cálculo Lambda.

El paradigma funcional está basado en el modelo matemático de composición funcional. En este modelo, el resultado de un cálculo es la entrada del siguiente, y así sucesivamente hasta que una composición produce el valor deseado.

No existe el concepto de celda de memoria que es asignada o modificada. Más bien, existen valores intermedios que son el resultado de cálculos anteriores y las entradas a cálculos subsiguientes. Tampoco existen sentencias imperativas y todas las funciones tienen transparencia referencial.

**Paradigma lógico:** La Programación Lógica es un Paradigma de Programación basado en la Lógica.

Los programas construidos un lenguaje lógico están construidos únicamente por expresiones lógicas, es decir, que son ciertas o falsas, en oposición a una expresión interrogativa (una pregunta) o expresiones imperativas (una orden). Un ejemplo de lenguaje lógico es Prolog (Programación lógica).

Prolog, proveniente del inglés Programming in Logic, es un lenguaje lógico bastante popular en el medio de investigación en Inteligencia Artificial. Prolog es un lenguaje muy diferente, tanto de los imperativos como Fortran, Pascal, C etc., como de los funcionales como Lisp.

**La programación lógica está dentro del paradigma de la programación declarativa.**

**La 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.

**Conclusiones:** La palabra paradigma en principio no tiene relación con la informática, el problema de estas frases en realidad es que no tienen sentido si no tienen un contexto, la palabra paradigma se podría aplicar a cualquier área y en cualquier contexto, ese es el verdadero problema.

Pues bien, habiendo quedado claro que es un paradigma ya podemos pasar a explicar que son los paradigmas de programación, que son ni más ni menos una serie de “normas” que todo programador ha de seguir a la hora de programar.

**Referencias:**

<https://blog.michelletorres.mx/paradigmas-de-programacion/>

<https://duglasm.wordpress.com/2012/04/01/paradigmas-de-programacion/>

<https://www.cienciasfera.com/materiales/informatica/tecnologiainformacion/tema21/12_paradigmas_de_la_programacin.html>