

CONSIDERACIONES SOBRE LA EVOLUCIÓN DEL PENSAMIENTO HUMANO A PARTIR DE LOS PARADIGMAS DE PROGRAMACIÓN

Considerations about the evolution of the human knowledge based on the Programming paradigms

RESUMEN

El presente artículo aborda el tema de los paradigmas de programación explicándolos de una manera simple y presentándolos en una línea de tiempo conceptual que puede interpretarse como una necesidad en la evolución del pensamiento humano. Se hace una breve descripción no solo de los paradigmas sino de sus aplicaciones que a la postre también se han asimilado y entendido como paradigmas y se presentan algunas reflexiones acerca no solo de la incidencia que ha tenido la evolución del pensamiento en conjunto con las nuevas necesidades del ser humano sino su proyección en el tiempo y lo que podría sobrevenir en el mundo de la programación de computadores en un futuro no muy lejano.

OMAR IVAN TREJOS B.

Ingeniero de Sistemas, M. Sc.
Profesor Titular

Universidad Tecnológica de Pereira
omartrejos@utp.edu.co

PALABRAS CLAVES: Pensamiento humano, paradigmas de programación,

ABSTRACT

This article shows the programming paradigms in a simple way and inside a conceptual timeline looking for them as a needing of the human knowledge evolution. We present a brief description of the paradigms and their applications and we establish some considerations about the relation between the evolution of the human thinking and the new problems of the world. In the same way, we present a prospection of the programming languages in the time and what we can wait in the world of the computers programming in the next future.

KEYWORDS: Human thinking, programming paradigms.

1. INTRODUCCIÓN

El presente artículo es producto del proyecto de investigación “Aplicación del Modelo 4Q de preferencias de pensamiento en el perfilamiento de los estudiantes de 1er semestre de Ingeniería de Sistemas de la UTP y en el ajuste de la metodología de enseñanza en beneficio del aprendizaje” que fue aprobado por la Vicerrectoría de Investigaciones de la Universidad Tecnológica de Pereira bajo el código interno 6-10-6 en el cual se busca aprovechar el conocimiento que se tiene del modelo 4Q de preferencias de pensamiento formulado por el Dr. Ned Herrmann en el proceso de perfilar a los estudiantes de 1º semestre de Ingeniería de Sistemas de la misma universidad y, a partir de allí, encontrar elementos que permiten mejorar los resultados del aprendizaje.

Establecer relaciones entre la evolución del pensamiento y los paradigmas de programación permitirá encontrar algunas relaciones causales entre cada uno de los paradigmas y la manera como éstos han ido mutando en virtud de los nuevos problemas que han ido apareciendo y en los cuales la tecnología ha permitido resolver. Cuando se conocen los antecedentes de procesos de orden tecnológicos entonces se hace más fácil pensar prospectivamente y mirar hacia el futuro con el ánimo de saber qué viene en camino y cómo nos debemos de preparar desde la Ingeniería de Sistemas para esas nuevas fronteras de conocimiento.

A manera de hipótesis, podríamos decir dos cosas: primero que es posible establecer una relación entre la evolución del pensamiento humano a través de su Historia con los paradigmas de programación y también que, basados en dicha relación, es posible mirar prospectivamente hacia donde avanza la programación y que nuevos retos nos esperan. Es claro que el espíritu que motiva este artículo sostiene que en realidad no solo existe tal relación sino que los paradigmas son consecuencia de dicha evolución.

El esquema que sigue el presente artículo comienza por describir el concepto de lo que es un paradigma en algunas de las diferentes interpretaciones que podemos encontrar, luego define lo que significa un paradigma de programación, establece una relación con la evolución del pensamiento humano y continúa con las primeras formas que se usaron para resolver problemas computacionales. Se sigue con el modelo de la programación orientada a objetos y con los lenguajes basados en matemáticas específicas.

2. QUE ES UN PARADIGMA.

Etimológicamente hablando, la palabra “paradigma” viene del griego παράδειγμα, la palabra “paradigma” viene del griego παράδειγμα que significa “ejemplo” o “modelo”. Esta primera aproximación nos permite conocer un perfil inicial no solo de lo que significa sino de hacia dónde apunta su orientación de significado. En términos generales, un paradigma puede considerarse

como un modelo o un patrón que se acepta en una determinada disciplina científica o en cualquier otro contexto epistemológico [1].

Algunos autores como Kuhn definen la palabra “paradigma” como “Un conjunto completo de valores, creencias y técnicas que son compartidas por todos los miembros de una comunidad específica”. Esta definición no s hace pensar en que, teniendo en cuenta la evolución del pensamiento y los cambios que en éstos se van dando, un paradigma también puede concebirse como un modelo de acuerdo al cual se construyen otros objetos.

En la evolución del pensamiento humano, los paradigmas han estado presentes en diferentes aristas de la concepción del mundo. La religión, la formación socioeconómica, las relaciones entre las sociedades y diferentes criterios de orden político, económico, social, cultural e, incluso, tecnológico han ido marcando la manera como el ser humano concibe el mundo. Podría uno imaginarse que el hombre primitivo miraba a la naturaleza y a sus demás semejantes de una manera antes de conocer el fuego y la rueda y que esa mirada cambió drásticamente luego de conocerlos. ¿Qué podemos decir de la concepción que tenía el ser humano en épocas posteriores?

La concepción estética del mismo cuerpo y su asociación con el arte y otras expresiones de tipo cultural y social son muestras de que los paradigmas con que el ser humano concibe el mundo han cambiado profundamente. La misma tecnología electrónica a proporcionado nuevas herramientas para que la vida del hombre, y su forma de vivirla, sea significativamente diferente. ¿Se adquiriría en estos tiempos un televisor sin control remoto? ¿Nos negaríamos a utilizar un cajero automático un día a las once de la noche y en un momento en el cual precisamos un dinero?

Es, en este punto, en donde vale la pena tener una pequeña definición de lo que significa Programar, de su relación con las matemáticas y, de nuevo, de los nexos con el pensamiento humano y su evolución. Ese es precisamente el tema del siguiente ítem.

3. QUÉ ES PROGRAMAR

En términos generales puede decirse que la programación de computadores es un camino para resolver un determinado conjunto de problemas a través de unos modelos y métodos específicos. Es un camino puesto que, frente a los problemas que pueden resolverse con la programación de computadores, siempre habrá caminos alternos que no necesariamente involucren tecnología.

Tal como se establece, la programación de computadores posibilita resolver un conjunto específico de problemas: los problemas computacionales, es decir, aquellos en los cuales la solución necesita, según el contexto, la participación de dispositivos de alta tecnología o sea

aquellos que pueden recibir información, procesar, almacenar y elaborar respuestas. Los modelos y métodos con los que cuenta la programación de computadores son, precisamente, los que se derivan de los paradigmas de programación que no son otra cosa que una manera determinada de resolver un problema [2].

La programación de computadores, por definición, es el arte de construir programas a través de un conjunto de instrucciones que son entendibles y que pueden ser ejecutadas por el computador, como solución a un problema de determinado. Programar implica entonces conocer un problema, definir el objetivo a resolver, plantear su solución, escribir un algoritmo, codificar en un lenguaje de programación, hacer pruebas pertinentes, compilar y ejecutar el programa y, finalmente, evaluar los resultados que deben satisfacer los objetivos planteados.

Tanto en la evolución del pensamiento del ser humano como en la evolución de los paradigmas de programación, han sido las matemáticas la gran base para que se pueda mutar ajustando las soluciones, y encontrando nuevas, acorde con los cambios que va teniendo la humanidad en su devenir y en su existencia y en rima con las nuevas necesidades que van surgiendo bien por simple evolución o bien porque las nuevas soluciones también traen en sí mismas nuevos problemas.

4. LOS PRIMEROS PARADIGMAS

Construidos los primeros computadores, las necesidades comenzaron a aparecer. El contacto con la máquina fue directo y aún no habían aparecido los “mediadores” que hoy hacen mucho más cómoda la vida del programador. El concepto de “programación” no estaba tan definido como en la actualidad e incluso el mismo oficio de programador era bastante difuso pues sencillamente cada computador, del tamaño de una casa, venía con las personas que lo manejaban, lo configuraban y podía sacarle el máximo provecho.

Como no eran de aplicación genérica, como los computadores de hoy, entonces tampoco era posible pensar en un abanico muy amplio de posibilidades y de soluciones a implementar a través de estas nuevas máquinas [3]. Casi que los “super computadores” (se usa el término *super* por su tamaño) tenía aplicaciones tan específicas que por esa razón no se pensaba, ni lejanamente, que un día estaríamos invadidos en nuestras casas y en todo nuestro entorno por estos aparatos.

El concepto de computador personal no se consideraba posible y los oficiantes eran tan específicos que había que pensar en su bienestar permanente para que, de pronto, esta estructura electrónica tan admirable no se fuera a quedar sin quien la operara. Son los tiempos prehistóricos de la informática y, lo sorprendente, es que esto sucedió apenas hace sesenta años, hacia 1945. Ese tipo de programación, en la que unos hombres muy especializados tenían que activar y desactivar un tablero

de más de veinte mil interruptores para lograr que el computador entregara los resultados esperados, se conoció como *Programación en Lenguaje de Máquina*.

El ser humano concebía a la tecnología como esa aplicación de la ciencia que estaba permitiendo que procesos que duraban cinco años se pudieran procesar en tan solo seis u ocho meses; algo que, comparativamente, era bastante favorable para las empresas y, especialmente, para los bancos. Si bien la Programación en Lenguaje de Máquina no era un paradigma como tal, sirvió como un modelo para empezar a concebir la tecnología como parte de la solución de los problemas sin pensar en que un día dependeríamos tanto de ellas.

La manipulación de un sistema como el binario en donde en vez de escribir el número 65 se escribía el número 01000001, trajo consigo riesgos de error que poco a poco se fueron cometiendo con las consecuencias, a nivel de información, que tuvieron precios altos. Manejar grandes volúmenes de información representada en unos y ceros y saber que era tan exageradamente fácil cometer errores, junto con el hecho de que cada operador o técnico de computadores en esos tiempos era casi “intocable” laboralmente, comenzó a hacer que el ser humano y la sociedad empresarial, particularmente, buscaran solución a tales malestares y fue entonces cuando se comenzó a pensar en que se necesitaba un mediador que hiciera un poco más fácil la relación con la estructura interna de esa máquina llamada computador.

Fue cuando apareció la posibilidad de construir pequeños programas basados en un lenguaje que se conoció como Lenguaje Ensamblador. A este tipo de programación se le conoció como *Programación a Bajo Nivel* pues de todas formas las instrucciones, a pesar de que no eran del todo entendibles, por lo menos lo eran mucho más que un conjunto de bits, que fue el nombre que se le puso a los unos y a los ceros.

La Programación a Bajo Nivel como tal tampoco se considera como un paradigma formal de programación pero el hecho de que permitiera un cambio significativo en la relación técnica con la máquina por parte de los operadores (que ya se empezaban a llamar como “programadores”) la convirtió en una forma de ver el mundo de esas nuevas tecnologías.

El ser humano comenzaba a ver que era posible resolver muchos nuevos problemas con el computador y que las instrucciones, finalmente, terminaban siendo manejables. Una orden como ST (que significaba “almacene”, del inglés storage) era mucho más entendible que 10001010 como pudo haber sido la orden original. El conjunto de órdenes (instrucciones) no era tan amplio como para no poderse manejar y se fue encontrando, poco a poco, más comodidad en las órdenes del Assembler que en los bits del Binario.

Es allí en donde se comienza a pensar en el concepto de Paradigma de Programación, es decir, en una forma estándar de resolver los problemas en los cuales la solución se podía implementar a través de un computador y que además fuera aceptada por los profesionales del mundo de la programación de aquellos días.

Aparece el concepto de Lógica de Programación y con él viene de la mano el concepto de Paradigma. En realidad lo que aparece en la escena es el primer paradigma cuyo nombre, debido a que se basa en unas estructuras de pensamiento, es *Paradigma Estructurado*. Este paradigma permitió ver en la programación la solución a muchos más problemas de los que hasta ese momento se habían resuelto. Se seguía aproximando más la relación entre el ser humano y la sociedad con el mundo de los computadores.

El paradigma estructurado abrió las puertas para la *Programación Imperativa* que es un tipo de programación en la cual se le dan órdenes al computador y éste las ejecuta literalmente. Los Lenguajes de Programación alcanzaron un nivel altamente comercial y se inició el proceso de crear aplicaciones (programas) y venderlos como productos comerciales, algo que hasta el momento no era tan fácil. Esto llevó a que la estructura electrónica de un computador fuera mucho más pequeña, más cómoda, más económica y más alcanzable.

El paradigma estructurado comenzó a resolver una muy buena cantidad de problemas pero, como todo en el ser humano, el mismo hombre comenzó a encontrar sus propios límites y comenzó a pensar en que este paradigma tenía unas fronteras desde las cuales algunas cosas era muy difícil de imitar. Fue cuando comenzó a planear la forma de ampliar las soluciones que había encontrado para aproximarse mucho más a la realidad, el gran anhelo de la sociedad informática.

5. APROXIMANDOSE A LA REALIDAD

El espíritu de la programación de computadores radica en poder construir soluciones que simulen la realidad que vivimos por fuera del mundo electrónico y que estas soluciones puedan ser ejecutadas por computadores. Sobre esta premisa surge una interpretación muy interesante que generó una aproximación a la realidad tal como la vivimos. El mundo tangible está construido a partir de objetos tales como un vaso, una puerta, una mesa, un código, etc.

En su más simple descripción un objeto tiene unas características y unos usos, es decir, el objeto puede ser descrito a partir de sus atributos y está destinado para algunas funciones específicas. Por ejemplo, un vaso, tiene un peso, una altura, un material que lo conforma, una capacidad, unas medidas geométricas, una forma geométrica, una densidad en su material, una transparencia, una determinada capacidad de resistir al calor y otros atributos muy específicos. Efectivamente

fue este principio lo que llevó a que se formulara un paradigma que ha ido modificando poco a poco la relación de la sociedad con la tecnología y, particularmente, con el mundo de la programación. Se llamó *Programación Orientada a Objetos*.

Concebida así la programación se hizo mucho más fácil aproximarse al mundo real pues, en los paradigmas anteriores, era necesario hacer una cantidad de reinterpretaciones para que ese mundo real tuviera una equivalencia en el mundo electrónico que posibilitaba la informática. Lo más interesante es que, con la POO (sigla que se ha acuñado para describir abreviadamente la Programación Orientada a Objetos) no solo se pudo describir el mundo tangible que nos rodea sino también el mundo intangible, el mundo de la imaginación, el mundo de los juegos, de las fantasías, de todas aquellas concepciones que el ser humano difícilmente había podido “construir” en el computador.

Con la POO aparecieron nuevos conceptos de programación tales como la herencia, el encapsulamiento y el polimorfismo que, a la luz del paradigma imperativo, eran conceptos difíciles de implementar (aunque ya se han desarrollado mecanismos para simularlos). Estos conceptos nuevos fueron apareciendo producto de las nuevas necesidades del ser humano y de ese intento franco que ha ido realizando el ser humano de aproximar el mundo informático al mundo electrónico de manera que éste simule a aquel.

Los lenguajes visuales, tan cómodos para el programador, se tomaron rápidamente el mercado y sus programas, mucho más cómodos para el usuario, se convirtieron en el factor diferenciador de tiempos pasados y tiempos modernos. La idea de los íconos constituyó un elemento comunicativo de un poder inimaginable tanto que ello permitió que grandes fortunas económicas y verdaderos imperios se construyeran a partir de su incidencia y aceptación en la sociedad.

La programación visual es una muy buena derivación, y si se quiere aplicación, de la POO que ha permitido, ante todo, simular visualmente el mundo real para bien, comodidad y dependencia del usuario. La dependencia del ser humano por la tecnología se hizo realidad y tal entorno es lo que ha hecho que se disparen las ventas de computadores en el mundo pues en últimas es como si el ser humano tuviera su entorno contenido en un portátil.

Es así como aparece una forma de pensar la programación que constituyó un paso adelante en el mundo de la informática. Aparece la *Programación Declarativa* como respuesta a este nuevo reto.

En la programación declarativa se condensan instrucciones de muy alto nivel, procesos completos reducidos a una sola orden con la cual el computador es capaz de interactuar sin mayores problemas, facilidades

inimaginables por el ser humano o que su concepción, hasta ese momento, involucraba un nivel de complejidad tal que exigía altas destrezas para su ejecución. La programación, otrora labor de grandes especialistas, estaba cada vez más a la mano de las personas del común, del ciudadano de a pie y sus resultados, los programas, eran más entendibles y mucho más fáciles de mantener.

Dentro de esta nueva forma de pensamiento que aprovechaba lo mejor de otros paradigmas pero tenía sus propias características, se empiezan a generar vertientes que hacen más efectiva la programación y que mantienen la filosofía de lo declarativo. Es aquí en donde la *Programación Funcional* irrumpe en lo técnico pues su fundamentación matemática ya se tenía hacía más de cincuenta años.

En la programación funcional la concepción general de solución a un problema parte de tres principios sencillos: la simplificación del objetivo, la reutilización del código y la reducción del tiempo en pruebas. La simplificación del objetivo se basa en la estructuración de una solución a partir de una unidad que es la que cambia el panorama y que se conoce con el nombre de función. Una función es una pequeña unidad que requiere unos argumentos para que cumpla un pequeño objetivo.

El cálculo Lambda desarrollado por Alonzo Church fue la base para que, muchos años después, la programación funcional fuera una realidad. La programación funcional hace uso de diferentes recursos tomados de otros paradigmas pero incorpora nuevos elementos para que la programación sea cada vez más fácil y más efectiva, espíritu principal de los cambios que se dan en las tendencias de la programación de computadores. La construcción de programas es mucho más simple y el planteamiento de soluciones se hace más ágil y menos complejo. ¿En un panorama tan promisorio como éste, que nuevos horizontes tendría la programación de computadores? Ese es precisamente el tema de nuestras próximas secciones.

6. MODELACIÓN MATEMÁTICA

Basados en estas formas de pensar lógicamente, que finalmente es lo que va marcando el camino a partir de los paradigmas, y teniendo en cuenta todas las posibilidades que cada vez se iban abriendo con la tecnología y su participación en la solución de problemas de la sociedad, se empezó a pensar en la necesidad de rescatar, para bien de esa misma sociedad, modelos matemáticos que habían sido diseñados hace mucho tiempo y que solo hasta ese momento, como ha sucedido con muchos de los aportes de la matemática, se veían como camino de aplicación y aprovechamiento tecnológico.

La primera forma de programación de la que hablaremos es la *Programación Lógica*. Este tipo de programación lo

podemos sintetizar de la siguiente forma: en un lenguaje tradicional (o sea aquel lenguaje de programación que se basa en paradigmas como el imperativo) la programación se reduce a la construcción de soluciones de problemas a partir de la disposición apropiada de una serie de instrucciones y sentencias. La buena disposición de las estructuras existentes junto con el aprovechamiento de los recursos, que tiene el lenguaje determinado, permiten que la solución finalmente sea efectiva y cumpla con el objetivo de resolver el problema.

En la programación lógica se acude a una forma de trabajo descriptiva de manera que se pueda establecer una relación entre entidades a través de lo cual se indica qué se va a hacer y no precisamente el cómo, que es la esencia del modelo imperativo. Según su creador, Robert Kowalski de la Universidad de Edimburgo, todo algoritmo se puede construir a partir de la especificación del conocimiento en un lenguaje de programación formal (que involucra la lógica de 1º orden) y un problema que se quiera plantear se puede resolver a través de un mecanismo de inferencia (que actúa como control) actuando sobre aquel.

El segundo modelo de programación, aceptado formalmente como otro paradigma de programación, lo constituye la *Programación por Restricciones*. En este paradigma las relaciones entre las variables son expresadas en los términos en que lo permitan las ecuaciones que se asocian con la solución. Sin duda, este es un paradigma que representa un interesante desarrollo en la programación de computadores y ha sido reconocido como uno de los grandes pasos dados en la investigación en computación.

En este paradigma como su nombre lo indica, se especifican un conjunto de instrucciones, que deben ser satisfechas por cualquier vista que tenga la solución que se plantea al problema en lugar de establecer, de manera determinada, los pasos para llegar a la solución. En su más simple esencia, en la programación con restricciones se busca un determinado estado en el cual se puedan satisfacer, al tiempo, una gran cantidad de restricciones. En este sentido se parte del principio de que un problema está definido como un estado específico de la realidad en donde existe un número determinado de variables con valores que no se conocen.

La tercera forma de programación que aparece en escena corresponde a la *Programación Concurrente*. En este estilo de programación se le da prioridad a la ejecución simultánea de diferentes tareas interactivas. Dichas tareas bien pueden ser un conjunto de procesos o diferentes hilos de ejecución que hayan sido creados por un mismo programa. Las tareas mencionadas pueden ser ejecutadas en una sola unidad de proceso pero también pueden ser ejecutadas en varios computadores (léase varios procesadores).

La programación concurrente enfatiza mucho más en la interacción entre tareas que en la especificación de proceso como tal. Por razones suficientemente entendibles, desde lo técnico, este estilo de programación (que algunos consideran un paradigma informal) tiene una relación directa con la programación paralela que algunos autores llaman programación distribuida. Esto ha llevado a asociar mucho más la programación lógica con la Inteligencia Artificial, la programación por restricciones con procesos matemáticos de alto nivel y la programación concurrente con redes y computación distribuida.

7. APROVECHANDO LAS NUEVAS TECNOLOGÍAS

Si la programación de computadores ha estado ligada a la evolución del pensamiento humano y si éste ha tenido un nexo con la tecnología, es apenas lógico pensar que la aparición de nuevos medios de información y comunicación han incidido en la aparición de nuevas formas de resolver problemas que si bien parten de paradigmas existentes, de una u otra forma se han ido convirtiendo en modelos para resolver problemas de la sociedad y ello los ha llevado, de manera indirecta, a ponerlos en el nivel de nuevos paradigmas [5].

La alta penetración de Internet en todos los rincones de la sociedad y su posible aprovechamiento en muchas situaciones problemáticas de ella, ha llevado a que se acuñe el término Programación para la Web que no es más que la aplicación de los conceptos teóricos junto con las posibilidades tecnológicas de la Web para implementar soluciones eficientes con programación.

Es la misma programación para la web la que ha llevado, con la explosión comercial de los dispositivos móviles (léase aparatos celulares, iphone, ipad, etc.), a que se piense en un modelo muy específico, por sus características, y que se ha coincidido en llamar Programación para Dispositivos Móviles. En este modelo, considerado por algunos ya como un paradigma, se destaca fundamentalmente el aprovechamiento de los servicios inalámbricos de conexión y comunicación como base para el desarrollo de la programación.

Como se puede ver estas dos tendencias, consideradas por algunos como paradigmas, han sido el resultado de encontrar soluciones eficientes y prácticas en el mundo de los dispositivos modernos y, viviendo la gran penetración que estos han tenido en la sociedad, se hace imposible pensar que el campo de la programación no haya sido permeado por dichas innovaciones tecnológicas.

8. LAS NUEVAS TENDENCIAS

Tendencias modernas como la Programación Literaria, la Programación Inteligente, la Programación Virtual, la Programación para la Animación y la Programación 3D forman parte de ese mundo de posibilidades que pueden

llegar a cristalizarse como paradigmas formales y que, actualmente, son solo planteamientos de cara a un inmediato futuro. La programación literaria aborda el problema del código como base primigenia para la generación de programas y, partiendo del principio de la documentación estándar, pretende excluirlo de forma que entre el programador y la máquina el nexo sea solamente la solución planteada en términos totales del programador y no como sucede hoy que dichas soluciones deben escribirse en términos de lenguajes de programación para poder ser ejecutados apropiadamente.

La programación inteligente busca resolver el problema de que los programas se modifiquen y se ajusten a ellos mismos de manera que en la medida en que los problemas que resuelven vayan mutando, ellos también lo vayan haciendo [6]. Tal vez sea una aparente utopía pero varios casos concretos prácticos han demostrado que esto es totalmente posible. El avance de la tecnología tendrá siempre elementos propios en su insondable mundo [7].

La programación virtual busca que los programas no sean unidades de código como tales sino soluciones que, sin tener una presencia específica, puedan aportar una solución concreta a los problemas. No ha de olvidarse que estas tendencias solo están en proceso de “cocción” y que lo que ahora podría parecerse un planteamiento imposible, tal vez la misma tecnología nos lo desmienta de cara a un inmediato futuro.

La programación para la animación y la programación 3D buscan lograr el proceso de la imitación de la realidad en toda su plenitud al punto de que el ser humano pueda interactuar, como ya lo está haciendo, con las creaciones que a partir de allí se hagan. La aparición de la tecnología MRI (Medical Resonance Image), los nuevos dispositivos y conceptos como la Realidad Virtual y Realidad Aumentada han permitido que se abra una gran cantidad de posibilidades en esta línea que antes eran inimaginables.

9. DISCUSION.

Como se ha expuesto es natural que la evolución del pensamiento, y exactamente la relación del ser humano con la tecnología, ha ido siempre en el sentido de buscar nuevas y mejores soluciones a partir de los avances tecnológicos existentes.

La programación orientada a objetos, como paradigma, ha sido la que en la actualidad se acepta más pues su aproximación a la realidad y a la interpretación de ella ha permitido construir soluciones que simulan dicha realidad, creando entornos virtuales frente a los cuales el usuario común pareciera estar en su mundo real. Todo esto pareciera contradecirse con la industria armamentista y con el mundo de la guerra pero reflexiones al respecto y evaluaciones de lo que ello ha significado serán tema de otro artículo.

10. CONCLUSIONES

A manera de conclusión podemos decir que sigue siendo muy vigente el mundo de la programación de los computadores con sus soluciones y sus posibilidades, amén de las nacientes e insatisfechas necesidades del ser humano. Conocer los paradigmas nos permite saber el camino que ha recorrido el pensamiento del hombre y confirmar, una vez más, que como mamífero racional nunca deja de buscar soluciones a los problemas.

Conocer dichos paradigmas, o por lo menos tener una idea clara de ellos, posibilita tener herramientas para realizar una mirada prospectiva y que, de una u otra forma, podamos saber hacia dónde tiende el mundo de las necesidades de la sociedad y que nuevas puertas podrán estar próximas a ser abiertas. Cabe destacar el importantísimo papel de las matemáticas en el desarrollo y consolidación de los paradigmas de programación. Si bien este artículo no es ni un recorrido histórico plagado de fechas ni un análisis profundo de la evolución matemática, no se puede negar la importancia de tener en cuenta que a todo paradigma le subyace una teoría matemática que lo fortalece.

11. REFERENCIAS BIBLIOGRÁFICAS.

- [1] A NEW APPROACH TO COMPUTER SCIENCE, (2005), Burns Brendan, Univ. of Massachussets, USA
- [2] CONCEPTS, TECHNICS AND MODELS OF COMPUTER PROGRAMMING, (2003), Van Roy Peter, Swedish Institute of Computer Science
- [3] HOW TO DESIGN PROGRAM, An Introduction to Programming and Computing, (2003), Matthias Felleisen, Robert Bruce Findler, Shriram Krishnamurthi, The MIT Press,, Cambridge, Massachussets, USA
- [4] AN INTRODUCTION TO THE HISTORY OF MATHEMATICS, Howard Eves (1990), Editorial Saunders, ISBN 0-03-029558-0, USA
- [5] FUNDAMENTOS DE PROGRAMACIÓN, Trejos Buriticá, Omar Ivan (2004), Fundamentos de Programación, Manizales (Colombia), Editorial Papiro
- [6] PROGRAMACIÓN LÓGICA, Alonso Jiménez, José, (2006), Grupo de Lógica Computacional, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Sevilla, España, Creative Commons, Stanford, California, USA
- [7] THE EARLY YEARS OF THE LOGIC PROGRAMMING, Kowalski, Robert, (1988), Communications of the ACM, January 1988, Volumen 31, Número 1, ACM Digital Library, Computing Machinery Association