Revista de investigación de sistemas e informática
15(2): 87-105 (2022)
DOI: https://doi.org/10.15381/risi.v15i2.24781

ISSN: 1815-0268 (Impreso) / ISSN: 1816-3823 (Electrónico)
Facultad de Ingeniería de Sistemas e Informática - UNMSM

Concepto de computabilidad en Alan Turing

Alan Turing's concept of computability

Miguel Salinas Molina
masepistemologia@gmail.com

Universidad Nacional Mayor de San Marcos,
Facultad de Matemáticas. Lima. Perú

RECIBIDO: 17/11/2022 - ACEPTADO: 14/12/2022 - PUBLICADO: 30/12/2022

RESUMEN

Reflexionamos sobre el concepto de computabilidad, conceptos definidos en la década de los años 30 del siglo XX, basado en la conocida tesis de Church-Turing; que dice que una función matemática recursiva es equivalente a lo que ejecuta la máquina de Turing que contiene la idea de algoritmo, noción intuitiva de secuencia de instrucciones. Desde una perspectiva crítica (por lo tanto filosófica), consideramos que la tesis de Church-Turing no contiene plenamente la teoría de computabilidad que profesó Alan Turing, que las ideas de Turing tienen un sentido más amplio.

Palabras clave: Computabilidad, Recursividad, Algoritmo, Tesis Church-Turing, Teoría de Turing.

ABSTRACT

We reflect on the concept of computability, concepts defined in the 1930s, based on the well-known Church-Turing thesis; which says that a recursive mathematical function is equivalent to what is executed by the Turing machine that contains the algorithm ides, an intuitive notion of sequence instructions. From a critical perspective (therefore philosophical), we consider that the Church-Turing thesis does not fully contain the theory of computability professed by Alan Turing, that Turing's ideas have a broader meaning.

Keywords: Computability, Recursion, Algorithm, Church-Turing Thesis, Turing theory.

I. EL PROBLEMA

La tesis de Church-Turing se formuló en base a las propuestas de ilustres matemáticos: Alonzo Church, definió un cálculo de funciones matemáticas que son recursivas, y Alan Turing, formalizo un procedimiento que ejecuta una máquina abstracta elemental, denominada autómata. Ambas propuestas constituyen, en cierto sentido una equivalencia en cuanto al significado de computar funciones matematicas, y juntas son un acercamiento formal de algoritmo.

Nuestro análisis tiene como inicio, el instante en que Alonzo Church y Alan Turing publican en 1936, sus documentos (separadamente y casi simultáneamente), el primero define el cálculo efectivo y el segundo el procedimiento efectivo, para luego en los años 1950 Stephen Kleene alumno y amigo de Church definirá la tesis Church-Turing.

El trabajo científico de Alan Turing abarcó diversos temas, que por circunstancias del celo en la protección del conocimiento científico por parte de los diversos gobiernos británicos, no permitieron que se divulgaran oportunamente. De otro lado, la tesis sostenida por Kleene fue formulada el siglo pasado, cuando aún no existía la computadora, siendo esta definición necesaria para los primeros pasos de la naciente ciencia de la computación.

II. DEFINICIONES PRELIMINARES.

Afirmamos que una definición de solo funciones matemáticas reduce el significado de computabilidad, siendo necesario clarificar el concepto de computabilidad en la tesis de Church-Turing, formulada por Stephen Kleene[1] en los años 50 del siglo pasado, esta dice: El cálculo efectivo (expresada mediante funciones recursivas) es equivalente a un procedimiento efectivo (instrucciones simples).

Ambos conceptos (cálculo efectivo y procedimiento efectivo) son considerados equivalentes en el sentido que refieren a la función matemática computable.

El concepto de computabilidad en las matemáticas suele relacionarse a la obtención de un valor mediante la ejecución de un cálculo, como funciones de operaciones aritméticas. Dado el desarrollo de las matemáticas en el área de las funciones y de la teoría de los números, permitieron constituir la ciencia de computación. Así lo expresa Roberto Torretti[2]:

> En los siglos XIX y XX la matemática prolifera y florece como quizás ningún otro que hacer del espíritu. Movidos por la misma riqueza y audacia de sus invenciones... Su reflexión es lo que se llama filosofía, y así la entienden; pero la conducen como matemáticos que son, aunando libertad y rigor, fantasía ubérrima y precisión pedante, en el estilo propio de su disciplina (Torretti, 1998: XI).

Ferrater Mora[3] define el concepto de computabilidad mediante funciones del tipo recursivo, incluye términos utilizados por otro ilustre matemático Alfred Tarski[4] en relación al concepto decidible. Afirma que un procedimiento es una secuencia de operaciones que permite la demostración de un teorema, y la teoría es decidible si sus funciones son recursivas:

> Se llama decidible a un cálculo C cuando puede forjarse un método o un procedimiento mecánico mediante el cual sea posible decidir – en una serie de operaciones finita – si una fórmula bien formada de C es o no un teorema de C… si existe en un cálculo C o una teoría formalizada T un procedimiento o método de decisión es llamado problema de decisión. Si se encuentra tal procedimiento o método, el cálculo o la teoría formalizada reciben el nombre de decidibles; si no, el de indecidible… Las anteriores definiciones no tienen carácter formal. Para una definición formal suficiente del término 'decidible' aplicando a una teoría formalizada T usaremos la formulación de A. Tarski (*Undecidable Theories*, 1953). Una teoría es llamada decidible si el conjunto de todas sus funciones válidas es recursivo; de lo contrario, es llamada indecidible. (Ferrater Mora, 2004, 786).

Precisa el concepto de decisión en el mismo sentido al formulado por Hilbert[5] en el congreso inter-

1 Cole Kleene, nace el 5 de enero de 1909 en Hartford, Connecticut y muere en Madison Wisconsin el 25 de Enero de 1994. Lógico y matemático. Se especializó en la teoría de las funciones recursivas. Escribió diferentes artículos y libros, destacando *Introducción a la matemática (1952) y lógica matemática (1967.)*

2 Roberto Torretti, autor del libro *El Paraíso de Cantor*, es un documento que contiene diversos temas que están expuestos en la tradición conjuntista en la filosofía de la matemática.

3 Ferrater Mora, filósofo y escritor, autor del Diccionario de Filosofía, libro referente de temas filosóficos.

4 Alfred Tarski, (14 de enero de 1901- 26 de octubre de 1983) lógico, matemático y filósofo polaco.

5 David Hilbert, nace en Königsberg, Prusia Oriental el 23 de enero de 1862, muere en Göttingen, Alemania el 14 de febrero de 1943, matemático. Fue uno de los fundadores de la teoría de la demostración, la lógica matemática y la distinción entre matemática y metamatemática. Un ejemplo de su liderazgo es la presentación en 1900 de un conjunto de problemas que establecieron el curso de gran parte de la investigación matemática del siglo XX.

nacional de matemáticas, celebrado en la ciudad de Paris en 1900 (al plantear 23 problemas de investigación). El programa sugerido por Hilbert trata sobre la búsqueda para la unificación axiomática de las matemáticas, esta sería conocido como *Entscheidungsproblem,* que significa: Problema de Decisión. Hilbert acota al respecto:

> La compatibilidad de los axiomas aritméticos. Cuando estamos inmersos en la investigación de los fundamentos de una ciencia, debemos establecer un sistema de axiomas que contiene una descripción exacta y completa de las relaciones que subsisten entre las ideas elementales de esta ciencia. Los axiomas para configurar son al mismo tiempo, las definiciones de las ideas elementales, y ninguna declaración en el ámbito de la ciencia cuya fundamentación nos están poniendo a prueba se considera correcto a menos que pueda derivarse de esos axiomas por medio de un número finito de pasos lógicos... quiero designar los siguiente como los más importantes de las numerosas preguntas que se le puede pedir en lo que respecta a los axiomas: probar que no son contradictorios, es decir, que un número definido de pasos lógicos basados en los mismos nunca pueden conducir a resultados contradictorios. (Hilbert, 1900:31-32).

III. TESIS DE CHURCH

El concepto de computabilidad expresado como la tesis de Church[6], fue publicada en el artículo *An Unsolvable Problem of Elementary Number Theory,* de 1936 (Un problema insoluble de la teoría del número elemental), define que toda función es efectivamente calculable si es una función recursiva.

Church presenta como procedimiento para obtener el valor de una función mediante la función recursiva a la que llamo calculo lambda, así lo expresa en la introducción del documento “Hay una clase de problemas de la teoría elemental de números que pueden estar en la forma requerida a encontrar una función de cálculo efectivo *f* de n enteros positivos” (Church, 1936:345).

Indica que la finalidad de su documento es proponer una definición de calculabilidad efectiva, en la que demuestra que no todos los problemas de la teoría de números elementales son solubles.

Church afirma que cualquier función lambda definible de números enteros positivos, proporciona un algoritmo, en el proceso de reducción de fórmulas. La evaluación mediante las funciones recursivas es un cálculo efectivo. Por lo tanto el cálculo lambda expresa adecuadamente el concepto de algoritmo.

> Esto es claro que, en el caso de cualquier función definible de enteros positivos, el proceso de reducción de fórmulas a formas normales provee un algoritmo para los cálculos efectivos de valores particulares de la función (Church, 1936:349).

En tal sentido Church define el cálculo efectivo, como un algoritmo para obtener los valores en una función de enteros positivos. Concluye que puede ser obtenido de cualesquiera de los dos métodos: (1) mediante la definición de una función efectivamente calculable si existe un algoritmo para el cálculo de sus valores, (2) por la definición de una función de calculo efectivo si existe un entero positivo tal como F(m)=n. La función recursiva toma en cuenta el concepto de invariantes[7] extendiendo lo expuesto en su documento.

Se interpreta lo expuesto como un criterio que supera la definición matemática que refiere a la calculabilidad, Al respecto, Roberto Torretti afirma: “Por eso, no parece justo describir la tesis de Church como una conjetura matemática que aguarda ser demostrada. La veo más bien como una decisión de aceptar la computabilidad como criterio de calculabilidad” (Torretti, 1998:376).

Kleene en su libro *Introduction to Metamathematics* dedica dos capítulos (XII y XIII) para sustentar la evidencia de la tesis de Church, sostiene que el concepto de calculabilidad efectiva de una función es de naturaleza intuitiva, por lo tanto la tesis no puede ser demostrada, es una hipótesis: “La tesis puede ser considerada como una hipótesis acerca de la noción intuitiva de calculabilidad efectiva, o, una definición matemática de calculabilidad efectiva” (Kleene, 1052:289).

IV. TEORÍA DE TURING

El concepto de algoritmo está implícitamente en la teoría de Turing, cuando define el concepto de máquina, en el sentido de ser un dispositivo que

6 Alonzo Church, nace el 14 de junio de 1903 en Washington y muere el 11 de agosto de 1995 en Hudson, Ohio. Matemático y lógico norteamericano, contribuyó en crear la base de la computación teórica. Se diplomó en la Universidad de Princeton en 1924. Su obra más conocida es el cálculo lambda y su trabajo de 1936 muestra la existencia de problemas indecidibles.

7 La teoría de invariantes permite obtener información de un objeto a través del estudio de otro objeto más sencillo. Las conjeturas de isomorfismos nos relacionan diferentes invariantes y facilitan el cálculo de los mismos.

ejecuta instrucciones básicas. Así lo dice Wittgenstein[8]: "Si el cálculo nos aparece como una actividad maquinal, entonces la máquina es el ser humano que realiza el cálculo" (Wittgenstein, 1978:195), esta expresión resulta equivalente a decir que las máquinas de Turing son personas que calculan.

El concepto de computabilidad que manifiesta Turing la encontramos en sus diversas publicaciones realizadas a lo largo de su vida científica, hemos escogido ocho documentos, porque consideramos que explican suficientemente la teoría de Turing[9]. Los documentos son los siguientes:

1. On computable numbers, with application to the Entscheidungsproblem (1936).
2. Systems of logic based on ordinals (1938),
3. Intelligent Machinery (1948),
4. Computing machinery and intelligence (1950),
5. Can Digital Computers Think (1951),
6. The Chemical Basis of Morphogenesis (1952)
7. The Chess (1953)
8. Solvable and Insolvable Problems (1954).

Afirmamos que la definición de computabilidad en Turing está en relación a la noción intuitiva de algoritmo, que trata sobre operar lógicamente dispositivos "físicos". Los conceptos utilizados son abstractos y corresponden a la ejecución en términos físicos siguiendo reglas, mediante instrucciones.

La opinión de Turing sobre la "inteligencia" en las computadoras ha propiciado debates en el campo de la filosofía, contribuyendo con resultados interesantes en el campo de la teoría de los lenguajes de la computación[10]. En cuanto a la definición dada por Turing sobre de "inteligencia" observamos que muchos desarrollaron diversas interpretaciones al respecto, pero si consideramos las publicaciones de nuestro matemático en su conjunto, observamos un cuerpo teórico en relación al concepto de computabilidad, al que denominamos Teoría de Turing.

Sostenemos como hipótesis que Turing realizó sus investigaciones en la ciencia de la computación, en relación entre máquina e instrucciones, habiendo tratado diversos temas que son parte de un cuerpo teórico, iniciándose en una solución negativa al *Entscheidungsproblem*, sobre funciones computables hasta llagar a sus estudios sobre aspectos relacionados al funcionamiento del cerebro del hombre, en la búsqueda y mejora de lo considerado como computable.

V. LA MAQUINA DE TURING

En 1936, Turing publica su trabajo en la Sociedad de Matemáticas de Londres, titulado: *On Computable Numbers, with application to the Entscheidungsproblem,* definiendo un dispositivo abstracto, que fue nombrado máquina de Turing. Éste es un concepto matemático basado en un procedimiento secuencial que se ejecuta en la máquina. Turing conoció casi instantáneamente el artículo, publicado por Church en 1936, observando la equivalencia de los resultados con los que él había obtenido sobre el *Entscheidungsproblem;* en el mismo sentido iniciado por Gödel[11]. Así lo indica Coello[12]:

> Turing tenía lista su investigación en abril de 1936, pero debió retrasarla, porque al mismo tiempo Church había llegado a la misma conclusión, en forma diferente, utilizando un cálculo basado en funciones recursivas. Así que Turing decidió hacer una nota de forma que se incluya como referencia el trabajo de Church. Ambos habían llegado al mismo resultado simultáneamente pero de formas diferentes. (Coello, 2004: 112).

La máquina de Turing suele ser descrita, como un dispositivo "físico", que no es posible fabricarla debido a la definición abstracta de sus componentes. La máquina está constituida por una unidad de control (determina el estado en que se encuentra la

---

8 Ludwig Josef Johann Wittgenstein (Viena, Austria, 26 de abril de 1889, Cambridge, Reino Unido, 29 de abril de 1951) filósofo austríaco, nacionalizado británico. Publicó el libro: *Tractatus logico-philosophicus*. Discípulo de Bertrand Russell en el Trinity College de Cambridge, donde llegó a ser profesor.

9 Los documentos corresponden a los trabajos de investigación de Alan Turing, publicados en los años: 1936 (define la máquina Turing), 1938 (su tesis doctoral en matemáticas), 1948 (informe cuando trabajaba en NPL) y 1950 (presenta el conocido Test de Turing), 1951 (respuesta al debate sobre el tema de inteligencia en las máquinas, 1952 (estudio sobre morfogénesis), 1953 (estudio sobre el ajedrez) y 1954 es un regreso al tema tratado en 1936).

10 Para mayor información consultar: Gramáticas de Lenguajes Formales por Chomsky N. http://www.chomsky.info/articles.htm. así también el documento sobre autómatas de la Universidad de Murcia, España. http://perseo.dif.um.es/~roque/talf/Material/apuntes.pdf

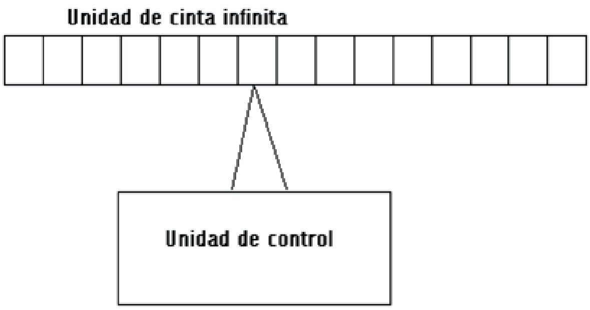
11 Kurt Gödel, (República Checa, 28 de abril de 1906-Princeton, Estados Unidos; 14 de enero de 1978) lógico, matemático y filósofo austríaco. Se le considera uno de los lógicos más importantes de todos los tiempos. Se le conoce por sus dos teoremas de la incompletitud, publicados en 1931, que resuelve negativamente el problema planteado por Hilbert *Entscheidungsproblem*.

12 Carlos A Coello, nace el 18 de octubre de 1967 en Tonalá Chiapas, México. Ingeniero civil con maestría en Ciencias de la Computación en la universidad de Tulane, Lousiana, EEUU y doctorado en Ciencia de la Computación en la misma universidad.

máquina y actúa sobre la memoria), memoria que es una cinta de longitud infinita en ambos sentidos (izquierda y derecha), contiene celdas una a continuación de otra (cada celda puede contener un símbolo).

Figura 1

Componentes de la máquina de Turing



La máquina de Turing ejecuta un procedimiento de manera precisa. Está definida de forma que es posible verificar el resultado de su funcionamiento mediante la utilización de lápiz y papel, con la necesaria actitud para no equivocarnos, explicado en las instrucciones del procedimiento.

Turing adiciona una nota a su primer trabajo de 1936, trata sobre el documento publicado por Church unos días antes, acerca de la idea de cálculo efectivo, manifestando que los resultados mostrados son equivalentes, pero que son diferentes definiciones.

> En un reciente documento de Alonzo Church ha introducido un concepto de "calculo efectivo", lo que equivale a mí "computabilidad", pero es definida muy diferente. Church también llega a conclusiones similares acerca de la *Entscheidungsproblem*. La prueba de equivalencia entre "computabilidad" y "calculo efectivo" se expone en un anexo al presente documento. (Turing, 1936:2).

De otro lado, la máquina de Turing contiene definiciones que permitieron elucidar el desarrollo de las computadoras, así tenemos la opinión de Claude Shannon[13] quien publica en 1956, su artículo *A Universal Turing Machine With Two Internal State* en la que demuestra que una máquina de Turing se puede reducir el número de estados hasta el mínimo de dos y que los estados eliminados pueden ser sustituidos por símbolos que estarían en la unidad de cinta (memoria). Esta demostración matemática relaciona las reglas que tiene la máquina con el almacenamiento en la memoria, planteando la idea que un programa puede ser un archivo almacenado.

Von Neumann[14], en su libro publicado póstumamente *The Computer and The Brain*, en 1958, acota como metáfora sobre la similitud de la computadora con el funcionamiento del cerebro. En el caso de las computadoras refiere a los códigos y su rol en el control del funcionamiento. Precisa que estos códigos adquieren la forma de ser completos, en el sentido que permiten el funcionamiento de la máquina e indica que la máquina de Turing se refiere al concepto de código corto, como código elemental de la máquina.

> El lógico ingles A. M Turing, mostró en 1937… que es posible desarrollar sistemas de códigos de instrucciones para una máquina computadora… Entonces el sistema de instrucciones que hizo una imitación de máquina como el comportamiento de otra es conocido como código corto (von Neumann, 1958:71)

VI. MAQUINA DE TURING Y EL ORACULO

Turing en su tesis doctoral en 1938, en la universidad de Princeton, la que fuera supervisada por Church, introduce la idea de un mecanismo del tipo “caja negra”, al que denomina “Oráculo”, éste dispositivo se añade a la definición de máquina. El mencionado mecanismo tiene la capacidad de determinar si un proceso terminará el cálculo que está por ejecutar sin haberlo realizado. El Oráculo resuelve el problema de la máquina que no pueda detenerse, como consecuencia de una ejecución al infinito, dado que no encuentra la solución.

El trabajo de Turing titulado *Systems of logic based on ordinal*, es un documento clásico en la teoría de la recursividad, en el cual el dispositivo “Oráculo”, convierte su máquina en una máquina-O. En el documento no explica cómo sería el dispositivo. Conocedores del interés del trabajo realizado por Church (las funciones recursivas), debe haber sido el propósito de Turing el resaltar las características de la recursividad y aislar los problemas que tiene la máquina de Turing sobre lo que no puede calcular.

13 Claude Elwood Shannon (nace el 30 de abril de 1916 y fallece el 24 de febrero del 2001). Ingeniero electricista y matemático estadounidense conocido por sus estudios de la información. Es conocida su publicación de 1948 sobre *A Mathematical Theory of Communications*.

14 John von Neumann zu Margitta, (28 de diciembre de 1903 - 8 de febrero de 1957) matemático húngaro-estadounidense. Padre de la teoría de juegos, publicó *Theory of games and economic behavior* junto a Oskar Morgenstern, en 1944. Pionero de la computadora.

Turing indica que las máquinas con Oráculo (máquinas-O) pueden ser descritas por tablas del mismo tipo de las que usó para su máquina original a la que denominó máquina-A (definida en 1936), de manera que para la definición de la máquina-O le asigna números a las configuraciones internas, que son estados en el mismo sentido al de la máquina-A.

> Con la ayuda del Oráculo podremos formar un nuevo tipo de máquina (llamada o-máquina), teniendo en éste uno de sus procesos fundamentales que resuelve un determinado problema teórico numérico. Más aún estas máquinas son el comportamiento en esta vía. Los movimientos de la máquina son determinados como usuales por una tabla, excepto en el caso de movimientos desde una cierta configuración interna **o.** Si la máquina está en la configuración **o** y si la secuencia de símbolos marcados con ***I*** son la formula bien formada **A**, entonces la máquina va de la configuración interna **p** o **t** según como esta es o no verdad que **A**. (Turing, 1938:173).

En la sustentación de la tesis doctoral de Turing, demuestra lo que calcula una máquina de Turing en relación a las funciones recursivas, basada en números ordinales. Ésta contribuye a la formulación de la Tesis de Church-Turing que dice: Si una función de enteros positivos es calculable si es recursiva, implica que ninguna máquina puede realizar un procesamiento que se encuentre más allá del alcance de una máquina de Turing. Todo cálculo que puede ser ejecutado en la máquina es expresable en funciones recursivas.

El Oráculo en la máquina de Turing, una máquina-O, contiene una potencia de cálculo mayor (puede determinar que es computable o no), para algunos, suele ser interpretada como una máquina con capacidades relacionadas al computo de la inteligencia del hombre.

VII. MAQUINA DE TURING QUE APRENDE

Turing argumenta sobre la máquina “inteligente” y lo manifiesta en el documento *Intelligent Machinery*, escrito en los tiempos en que laboraba en el *National Physical Laboratory* de Londres, en 1948, donde hace una crítica a las ideas que considera que atacan el concepto que él manejaba de “inteligencia” en las máquinas. Su documento crítica a cinco inclinaciones, las rechaza por considerarlas superadas, así las enuncia:

> (Objeción a) Una indisposición para admitir la posibilidad de lo que se pueda hacer, es que el hombre pueda tener un rival con poder intelectual… (Objeción b) Una creencia religiosa en la que cualquier idea a construir una máquina es una clase de promesa irreverente… (Objeción c) El límite máximo de caracteres de máquina que se pueden usar, según tiempos recientes (hasta 1940). Estimuló a creer que la máquina es limitada ante las exigencias extremas …(Objeción d) El teorema de Gödel y sus resultados relacionados (Gödel, Church, Turing) muestran que si uno usa las máquinas para determinar la verdad o falsedad de teoremas matemáticos, no está dispuesta a tolerar un inesperado resultado errado, entonces cualquier máquina estará en el mismo caso de no proporcionar una respuesta a todo. De otro lado la inteligencia humana parece ser capaz a encontrar métodos e incrementar el poder de tratamiento con tales problemas. (Objeción e) Hasta donde una máquina pueda mostrar inteligencia y es contemplada de que si es posible, será como reflejo de la inteligencia de su creador (Turing, 1948: 1).

Turing refuta cada una de las objeciones, afirmando que es posible la “inteligencia” en las máquinas, desde una perspectiva matemática, las máquinas abstractas guardan relación con el concepto de algoritmo, contienen un procedimiento de instrucciones que permiten operar lógicamente los dispositivos de la máquina.

Sostiene que un procedimiento puede ser cambiado, siendo un concepto temprano sobre la modificación de la secuencia de ejecución de las instrucciones del algoritmo, indicando que la máquina “aprende” para no repetir una situación no deseada, esto se logra modificando la secuencia de instrucciones en el programa, para este fin el programa debe tener elementos almacenados en la memoria de la máquina, de forma que puedan ser modificados en el momento de que se esta ejecutando.

Sobre las objeciones mencionadas por Turing, (a) y (b) nos dice que son de naturaleza emocional, por lo tanto no requiere mayor sustentación en su contra.

En la objeción (c), Turing la descarta con el ejemplo de la máquina ENIAC, indicando sus capacidades de operación en cuanto a velocidad y almacenamiento, el argumento se hace más fuerte si consideramos las capacidades actuales de las computadoras. En la objeción (d), referida a los argumentos del teorema de Gödel y que la máquina no debería cometer errores, afirma como criterio de refutación y además como una línea de acción del pensamiento del diseño de las computadoras y programas, que errar no es un requisito para negar la posibilidad de "inteligencia". Sostiene como ejemplo del aprendizaje, que se basa en el error, como elemento importante para incorporar experiencia.

Finalmente en la objeción (e) sostiene que se contradice con el siguiente ejemplo: Si un profesor ayuda a sus alumnos con métodos de enseñanza y luego abandona la comunicación con el pupilo, siendo el pupilo quien presenta los resultados y no el profesor, la decisión le corresponde al pupilo y no al profesor, en clara alusión a que la máquina presenta el resultado y no el creador del programa.

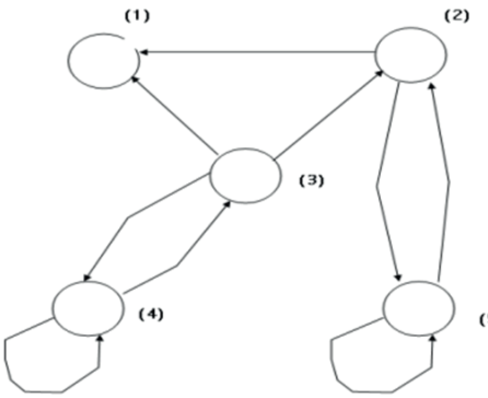
Siguiendo con el documento de 1948, *Intelligent Machinery,* Turing define una máquina abstracta, que la denomina máquina desorganizada, sus estados y las reglas de incidencia son definidas como funciones, representadas mediante una matriz, de forma que el programa contiene reglas que están en la mencionada matriz, si se altera los valores de la matriz, se modifica la relación entre los estados, cambiando la secuencia de las instrucciones de la máquina. Esta definición de cambio de relación entre los estados, es una modificación a la máquina original de Turing.

Turing nos muestra un ejemplo sobre la máquina desorganizada, presenta la relación de cinco estados y refiere que es una idealización, porque en la práctica las computadoras están definidas con más y múltiples estados.

La máquina desorganizada está constituida por estados, y los relaciona mediante funciones i(r) y j(r), que indican la salida hacia el estado r. Así nos presenta el siguiente el diagrama.

Figura 2

Máquina desorganizada (Turing, 1948: 6)



Donde las conexiones corresponden a las relaciones de incidencia, expresadas en las siguientes funciones:

Figura 3

Relaciones en la máquina desorganizada (Turing, 1948: 6)

r (estado)	i(r)	j(r)
1	3	2
2	3	5
3	4	5
4	3	4
5	2	5

Turing incorpora adicionalmente los conceptos de *control* y *active* (controlado y activo). El controlando proporciona información mientras el concepto de activo es de sentido contrario, no proporciona información, así nos da algunos ejemplos de máquinas:

> Un teléfono es continuo y controlado (proporciona información).
>
> Una calculadora mecánica es discreta y controlada.
>
> Un cerebro es probablemente continuo y controlada, muy similar a un mecanismo discreto (proporciona información).
>
> La ENIAC, ACE, etc. son discretos y controlados (proporcionan información).

Menciona de su máquina como un mecanismo inteligente, con estructura lógica (en relación a un sentido físico), que realiza lo que puede ejecutar un hombre usando un lápiz y papel. Define que las máquinas de Turing no tienen límites en la duración de la ejecución y tampoco en la capacidad de la memoria.

> Estas máquinas que aquí llamamos máquinas de computación lógica, son de principal interés cuando nosotros consideramos en principio que una máquina puede estar designada a hacer, permitiendo lo ilimitado del tiempo y lo ilimitado en la capacidad de almacenamiento (Turing, 1948:4)

Turing afirma que es posible hablar de una máquina que "aprende", porque puede cambiar la relación de sus estados. Así resulta que lo manifestado en 1948 es un concepto temprano de lo que sería expresado en la inteligencia artificial, y que sería un tema interesante y también filosóficamente controversial.

Hoy el concepto de estado en la máquina y que puede ser modificado en la ejecución de un procedimiento, tiene diversas aplicaciones, como en los algoritmos genéticos, que son programas que modifican su comportamiento inicial, temas estudiados en las redes neurales en la computación[15].

VIII. TEST DE TURING

Turing publica en 1950 un artículo titulado *Computing Machinery and Intelligence*, en la revista de filosofía *Mind* de octubre de 1950, refiriendo a la posibilidad de "inteligencia" en las máquinas, dejando el significado de inteligencia artificial, como un concepto de simulación mediante la ejecución de un procedimiento, proponiendo un juego de imitación en la que participan tres personas: un hombre, una mujer y un interrogador.

> El interrogador está en una habitación separado de los otros dos (hombre y mujer), y el objetivo es determinar cuál es la mujer y cuál es el hombre. Para este fin el interrogador hace preguntas a los otros dos, la mujer se presenta como una ayuda al interrogador y el hombre procura enredar y confundir al interrogador. Las preguntas del interrogador pueden ser: Diga cuan largo es su pelo. Las respuestas de la mujer pueden ser: Yo soy la mujer, no le crea a él, esto no ayuda mucho porque el hombre puede hacer la misma afirmación. (Turing, 1950).

El documento conocido como Test de Turing, sostiene que es posible el diálogo, entre la computadora y personas, tal como lo enuncio en su ejercicio para identificar el sexo de las personas. Expone un dialogo de preguntas y respuestas, que tiene la restricción de limitar las formas coloquiales, reduciéndolas a mantener una jerarquía de preguntas y respuestas, precisando en el ejemplo que la participación de la computadora es como interrogador, limitando el juego de imitación en que la automatización participa solo haciendo preguntas.

En el año 1950, Turing manifiesta que dentro de los próximos cincuenta años[16] será posible construir programas y computadoras con mayor capacidad de procesamiento y almacenamiento, por lo tanto, podrán manejar mayor cantidad de preguntas y situaciones, que permitan la identificación del sexo de los jugadores, tal como lo propuesto en el Test. Asimismo en el documento refuta las ideas que cuestionan la posibilidad de "inteligencia" como lo hizo en su documento de 1948, denominándolas objeciones.

Roberto Perazzo[17] menciona un hecho ocurrido en 1991, referente a la exposición de programas de computadora, basados en juegos de imitación, realizado en el Museo de Computación de Boston, Estados Unidos, nos dice que al final del evento se registro que las máquinas lograron engañar a cinco de los diez jueces que actuaron como interrogadores. (Perazzo, 1994: 101).

Si bien los argumentos expuestos por Turing aún no son contrastables plenamente, tampoco pueden ser descartados, según lo expuesto, se presenta el concepto de "inteligencia" para las computadoras, como simulación en que las máquinas logran confundir a jueces.

Turing investigó la computabilidad, en el sentido del procedimiento efectivo en relación a su máquina, elucidando detalles mínimos de lo que entendemos por algoritmo. Copeland,[18] nos dice que Turing nunca hizo mención directa a la Tesis de Church-Turing, pero interpreta cierta referencia implícita de Turing en su ensayo: *The Chess*, que fue publicado en 1953 en la colección *Faster Than Thought*, en la sección titulada *Digital Computers Applied to Games.* El párrafo al que se refiere es el siguiente:

> Si uno puede explicar con bastante ambigüedad en el inglés y con la ayuda de símbolos matemáticos, si es requerido, como un cálculo que se debe hacer, entonces siempre es posible programar cualquier computadora digital para hacer ese cálculo, siempre que la capacidad de almacenamiento sea adecuada (Turing en Copeland, 204:567).

Copeland señala lo mencionado por Turing: "no es el tipo de asunto que admite la reducción clara de la prueba" (Turing en Copeland, 204:567-568). Consideramos que la interpretación de Copeland es en cierto sentido entendible al referir un método matemático, suponiendo que es así y lo dice de la

15 Carlos Coello Coello lo menciona en *Breve historia de la computación y sus pioneros*, pagina 117.

16 En una discusión presentada por la BBC de Londres, el 10 de enero de 1952, entre Richard Braithwaite, Geoffrey Jefferson, Max Newman y Alan Turing, el tema refería a que las máquinas de cálculo automático pueden "pensar". Turing afirma que dentro de 100 años esto será posible, aumentando la fecha al año 2052.

17 Roberto Perazzo físico y notable investigador argentino, autor del libro *De Cerebros, Mentes y Máquinas.*

18 Jack Copeland (nacido en 1950) es profesor de filosofía en la Universidad de Canterbury (Nueva Zelanda). Recibió su D.Phil y B.Phil en filosofía de la Universidad de Oxford en 1979 por investigaciones sobre lógica modal y no-clásica. Es director del Archivo Turing para la Historia de la computación en Canterbury (Nueva Zelanda) desde 1985. Es considerado un experto en los temas de Alan Turing.

siguiente manera: “como un cálculo que está hecho por un obediente empleado en concordancia a un método matemático” (Copeland, 204:568).

El concepto que postula Turing sobre máquinas inteligentes, no refiere a una inteligencia en forma como se presenta en el hombre, la interpreta como una simulación mediante algoritmos, que se manifiestan en las máquinas en correspondencia “física” a lo que se ejecuta mediante un programa.

Constatamos el desarrollo de la ciencia y tecnología, hoy materializada en la computadora, siendo denominada por algunos, como uno de los inventos tecnológicos más importantes del siglo XX. En opinión de Turing, en 1950, manifestó que se llegará a tener computadoras con mayor capacidad y que puedan engañar a personas como si la máquina “pensara”, tal como ocurre con programas de computadora que “juegan”, siendo el caso ejemplar programas de ajedrez, donde ganan partidas de juego a maestros.

IX. COMPUTABILIDAD Y VIDA ARTIFICIAL

Turing conocía de programación de computadoras, fue usuario de las mismas, así lo menciona Copeland en referencia a una carta de Turing a un colega del *National Physical Laboratory*, a principios de 1951, en que muestra su alegría por la llegada de la computadora para sus estudios en diversos patrones que se presentan en la naturaleza como la filotaxis[19] de la serie de Fibonacci[20], como sucesión numérica que está presente en diversas estructuras biológicas.

> Nuestra nueva máquina [la Ferranti Mark I] empieza a llegar el lunes. Estoy esperando con uno de los primeros trabajos para hacer algo sobre 'embriología química'. En particular creo que uno puede dar cuenta de la aparición de los números de Fibonacci en relación con el abeto-conos (Turing, 1951:1).

La “embriología química” estudia el desarrollo de la estructura anatómica del embrión, es el resultado de la difusión de productos químicos que reaccionan entre sí, formando patrones. En su documento *The Chemical Basis of Morphogenes,* de 1952. Hace una modelización matemática de lo que llegaría a ser conocido por los estudiosos de la morfogénesis en la biología como los patrones de Turing.

Según Damian Strier[21] en su tesis para optar el grado de Doctor en Ciencias Físicas sobre Procesos de Auto-Organización en Sistemas Biológicos, menciona a Turing como pionero en este campo de investigación.

> Turing demostró la plausibilidad de la formación espontanea de un patrón espacial estacionario de morfógenos en un medio inicialmente uniforme. Para ello supuso solamente la existencia de reacciones químicas y de un proceso de transporte difusivo de los morfógenos (Strier, 2002:13)

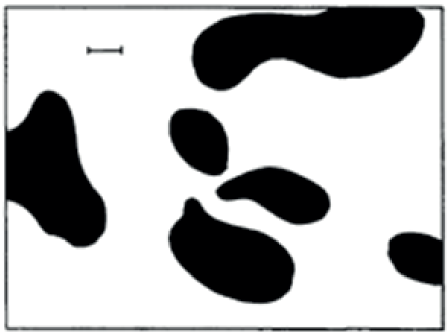
El modelo matemático propuesto por Turing trata de la morfogénesis de unas manchas, resulto suficiente para establecer un camino de investigación.

> … es claro que además de la información codificada a nivel genético intervienen otro tipo de factores los cuales determinan el patrón de encendido selectivo de genes en las distintas células. Este encendido… puede originarse… presencia de gradientes químicos… Esta posibilidad fue analizada por Alan Turing, encontrando que un sistema de reacción difusión con dos reactivos es, bajo ciertas condiciones, capaz de auto organizarse espontáneamente (Strier, 2002:61)

Turing en su publicación de 1952 formula un cálculo que resuelve ecuaciones que para la solución debió dedicar horas, obteniendo un gráfico que son unas manchas, similares a las que tiene un perro de raza dálmata.

> Este proceso es muy conveniente para la computación…La figura 2 muestra un patrón, el cual se obtiene en pocas horas por un cálculo manual. (Turing 1952:27)

Figura 4
Grafico obtenido por Turing basado en cálculos de ecuaciones diferénciales.



19 Filotaxis es la disposición que presentan las hojas en el tallo. La disposición se presenta en características en cada especie, se sostiene que cada hoja busca tener una mayor exposición al sol.

20 La serie de números de Fibonacci son aquéllos que los dos primeros números se mantienen fijos y los siguientes son la suma de los dos anteriores: 1, 2, 3, 5, 8,... n, m, n+m …

21 Damian Strier, argentino, doctor en ciencias físicas, investigador de los procesos auto organizados en los sistemas biológicos, el cual es una orientación que acerca la física a las reacciones químicas.

Así mismo el interés de Turing por la genética, queda manifiesto en una carta dirigida a C.H. Waddington[22], el 11 de setiembre de 1952, le indica su interés en las manchas que están presentes en diversos animales, éstas son las que tienen las cebras, tigres, leopardos u otro animal. Copeland menciona que Waddington comentó sobre la publicación de Turing como: "… que la aplicación más clara de la teoría de Turing parecía ser en los moretones que se producen en el deporte, manchas y rayas que se producen cosas de áreas uniforme aparentes." (Copeland, 2004: 509)

El documento de Turing sobre morfogénesis es una modelización matemática sobre reacciones químicas, utiliza para este fin ecuaciones diferenciales[23], que describen sistemas dinámicos. El trabajo de Turing contiene cierto rigor matemático, él así lo menciona aunque reconoce el propósito divulgativo del documento (Turing, 1952).

Turing desarrolla el modelo conocido como reacción y difusión, ponderando las reacciones químicas, supone que las velocidades de reacción obedecen a la "ley de acción de masas" y está en proporción a la concentración de sustancias que participan de la reacción, presenta un ejemplo de velocidad en la que se formará el cloruro de plata y precipitará del nitrato de plata y del cloruro de sodio.

> $Ag^{+} + Cl^{-} \rightarrow Ag\,Cl$. Será proporcional al producto de concentración de iones de plata Ag^{+} y los iones de CL^{-}. Esto es señalado en la ecuación $Ag\,NO_3 + Na\,Cl \rightarrow Ag\,Cl + Na\,NO_3$ (Turing, 1952:5)

Turing explica al final de su documento que lo tratado tiene consideraciones de límite y que la principal desventaja es que sólo obtiene resultados para casos particulares, explica que los temas considerados se ven pequeños ante la gran variedad de casos que se presentan en los complicados fenómenos biológicos. Finaliza indicando que es posible modelar matemáticamente muchos casos, y que su trabajo se orienta en la comprensión de las formas biológicas.

> Teniendo esto en combinación con las matemáticas utilizadas en este trabajo son relativamente elementales, difícilmente se podría esperar encontrar que muchos fenómenos biológicos observados estarían cubiertos. Se cree, sin embargo, que los imaginarios sistemas biológicos que han sido tratados, y los principios que se han discutido, deben ser de alguna ayuda en la interpretación real de las formas biológicas. (Turing, 1952:43)

El concepto de computabilidad utilizado por Turing está relacionado a los temas de la "vida" artificial, en referencia al algoritmo genético, que es el procedimiento que se emplea para simular la evolución natural para producir generaciones sucesivas de entidades definidas. Turing presenta la idea de un algoritmo genético[24] en su publicación *Intelligent Machinery,* de 1951, en la que denomina un procedimiento de búsqueda genética o evolutiva.

> Hay la búsqueda genética o evolutiva, por la cual una combinación de genes es buscado con un criterio de valor de supervivencia. El notable éxito de esta confirmación de búsqueda es en cierta medida la idea de que en la actividad intelectual consiste principalmente de diversos tipos de búsqueda (Turing, 1951:22)

La investigación de estos temas están en el mismo sentido al realizado por Turing, fue de interés de prestigiosos matemáticos, en cuanto a mecanismos del tipo auto reproductivos, como se muestra en los trabajos de von Newmann, referidos por Copeland, al describir una carta que von Neumann dirige a Norbert Wiener[25], el 29 de noviembre de 1946, sobre los mecanismos de auto reproducción: "Yo pensaba mucho sobre mecanismos auto-reproductivos. Puedo formular el problema rigurosamente alrededor del estilo en el que Turing lo hizo para sus mecanismos" (Copeland, 2004:515).

Turing en su ensayo *The Chess,* de 1953, trata sobre las computadoras digitales aplicadas a los juegos, establece criterios de naturaleza formal, argumenta la posibilidad de construir un programa donde la computadora actúe como un jugador en una partida de ajedrez. Empieza su ensayo con la pregunta ¿Podría uno hacer que una máquina juegue ajedrez?, respondiendo que es posible plantear varios significados en relación a la misma pregunta. Dice que una computadora electrónica programada adecuadamente se constituye en sí misma en una máquina, en referencia a la máquina universal de

22 Conrad Hal Waddington (1905- 1975) fue un biólogo, paleontólogo y genetista escocés, uno de los fundadores de la biología de sistemas.

23 Las Ecuaciones Diferenciales son ecuaciones que están presentes las derivadas de funciones de una o más variables, ejemplo: $a_n(x)y^{(n)} + a_{n-1}(x)y^{(n-1)} + \ldots + a_1(x)y' + a_0(x)y = g(x)$, si estas son de una variable se denominan Ecuaciones Diferenciables Ordinarias, caso contrario son Ecuaciones Diferenciables Parciales.

24 Algoritmo Genético es el término introducido en la Ciencia de la Computación por John Holland en 1975 en la Universidad de Michigan.

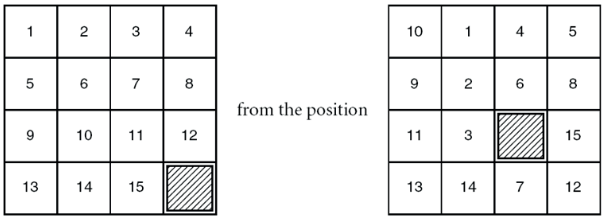
25 Norbert Wiener (26 de noviembre de 1894, Columbia, Missouri - 18 de marzo de 1964, Estocolmo, Suecia) matemático estadounidense, fundador de la Cibernética. Es uno de los precursores de la teoría de la comunicación.

Turing. De otro lado, como posibilidad hacia el futuro, propone una máquina que construida adecuadamente permitirá jugar ajedrez.

Antes de fallecer Turing, en 1954, presenta el documento *Solvable and Unsolvable Problem,* que resulta ser una explicación sencilla del *Entscheidungsproblem,* resuelto en su documento *On computable numbers, with application to the Entscheidungsproblem,* de 1936, con la diferencia metodológica, de que en el documento de 1936 desarrolla el concepto de la máquina abstracta, ahora en 1954, revisa el caso mediante un juego de rompecabezas, que trata de una cuadricula de 16 espacios, que contiene 15 fichas y una vacía, de forma que permite el desplazamiento de las fichas vecinas hacia el lugar vacío.

Figura 5

Puzzle de 16 casilleros, figura obtenida de Turing, 1954.



El artículo fue publicado en Science News[26] y trata de problemas que admiten solución algorítmica, Turing expone un ejemplo de un problema que no es resoluble por cualquier método sistemático, a la pregunta: ¿Un puzzle para todos los casos tiene solución? La respuesta es no, no hay un método que responda la pregunta de forma afirmativa o negativa. Prueba que es posible obtener la solución para los casos que se obtienen desde una posición inicial en el tablero del puzzle que dé luego de varias modificaciones queda “desordenada” (solución en relación a la posición inicial). Reflexiona sobre el ejemplo del tablero de 16 casilleros (4 x4) y 15 fichas, calculando las posibles posiciones de las fichas, resultando ser 2`092,278`988,000, y las posibilidades se reducen si se considera en el tablero un estado inicial y a partir de éste producimos los posibles estados finales del tablero.

En el documento Turing analiza el puzzle con infinitos casilleros, y concluye que no sería posible encontrar una solución, utilizando una búsqueda exhaustiva[27]; para los casos de tableros finitos, es posible un procedimiento de búsqueda sin descartar la utilización de la intuición para establecer estrategias que mejoren el mecanismo de búsqueda.

Turing relaciona el concepto de función computable con el de procedimiento efectivo, aclarando la similitud de ambos, en el sentido que establece un método que encuentre una explicación entre lo que se puede y no se puede decidir, remontándose a su documento de 1936.

> … se puede reducir a la definición de "función computable" o "procedimiento sistemático”. Una definición de cualquiera de estos puede definir todos los demás. Desde 1935 una serie de definiciones se han dado, explicando en detalle el significado de uno u otro de estos términos, y estos han demostrado ser equivalentes entre sí y por lo tanto equivalente a la declaración anterior. (Turing: 1954:8).

Define el concepto de procedimiento sistemático como aquel que se ejecuta para resolver un juego del tipo puzzle, precisando que tendrá una solución, si el resultado corresponde a la posición inicial de las fichas, solo así tiene sentido el procedimiento sistemático.

Consideramos que Turing sintetiza al conjunto de los temas tratados, en la búsqueda en aspectos relacionados a lo computable, esta se manifiesta explícitamente en la carta de 1951, dirigida al biólogo John Young[28], menciona sus intenciones sobre la investigación en morfogénesis, y su interés en las redes neuronales. La carta de dos páginas explica casi en la mitad de ella, temas relativos al cerebro y su capacidad de almacenar datos, indica que estos estudios son más difíciles de alcanzar. Precisa que está trabajando temas más sencillos que se orientan hacia la comprensión del funcionamiento del cerebro humano.

> Me temo que estoy muy lejos del escenario, donde me siento inclinado a empezar a hacer alguna pregunta anatómica [del cerebro]. De acuerdo con mis ideas de cómo establecer al respecto que no ocurrirá hasta una etapa tardía, cuando tengo una teoría bastante clara acerca de cómo se hacen las cosas. En la actualidad no estoy trabajando en el problema en absoluto, pero en mi teoría matemática de la embriología… dar explicaciones satisfactorias sobre i) gastrulación[29].

26 Science News, revista de divulgación científica de la época (1954).

27 Búsqueda exhaustiva, es el tipo de búsqueda que revisa o evalúa todos los posibles casos.

28 John Zachary Young (Bristol 18 de marzo de 1907 – Oxford 4 de julio de 1997) Zoólogo y neurofisiólogo ingles.

29 Gastrulación es el proceso en el que se forma las capas germinales del embrión que originan todos los tejidos el futuro bebé.

> ii) la poligonalidad de las flores y estructuras simétricas, por ejemplo, estrellas de mar. iii) la disposición de las hojas, en especial a las que están involucradas en la forma. La serie de los números de Fibonacci (0, 1, 1, 2, 3, 5, 8, 13 ,...). iv) los patrones de color de los animales, por ejemplo, rayas, manchas y motas. v) Los patrones en las estructuras casi esféricas examinándolas como en algunos radiolarios[30], pero esto es más difícil y dudoso. (Turing, 1951: 2)

Turing trabajó una teoría alrededor de la definición abstracta de máquina. De esta forma asoció el concepto de procedimiento efectivo con las que permitan operar aquello que es decidible, en orientación a lo propuesto en el *Entscheidungsproblem,* específicamente avanzó en el concepto de computabilidad como constante la utilización de algoritmos como procedimiento efectivo, en relación a programas de computadoras que simulan inteligencia mediante procedimientos mecánicos que operan dispositivos físicos que cumplen reglas de naturaleza lógica.

X. TESIS DE CHURCH-TURING

La tesis Church-Turing define la computabilidad de una función computable, resultado importante en la ciencia de la computación debido a que ayudo a formular una disciplina del conocimiento humano: La ciencia de la computación[31].

El éxito más evidente de la tesis de Church-Turing se manifiesta en la formulación de la teórica de los compiladores, que trata sobre los lenguajes para hacer programas que funcionan en las computadoras, en estrecha relación con los conceptos de algoritmo y de recursividad.

La tesis se enuncia de la siguiente forma: Una función es efectivamente computable si sólo si es Turing computable. Esta incluye dos conceptos, la función efectivamente computable que refiere a las funciones recursivas que son un cálculo efectivo, enunciado por Church en el cálculo lambda; el segundo el procedimiento efectivo, definición dada por Turing en el sentido de ejecutar exitosamente una secuencia de instrucciones en una máquina de Turing.

Tanto Church como Turing, trabajaron sus investigaciones en relación con los resultados del segundo teorema de Gödel (teorema de la incompletitud). Así ambos demuestran en forma diferente la naturaleza de realizar el cálculo, y la posibilidad e imposibilidad de calcular, tomando como base los números enteros, específicamente Church prueba la existencia de cálculos efectivos mientras que Turing prueba la indecibilidad de ciertos cálculos.

El matemático Kleene, alumno y amigo de Church, en el año 1952, publica en su libro *Introduction to Metamathematic*, la tesis de Church-Turing, el libro con un poco más de 564 páginas, es una defensa del concepto de función computable en relación a la función recursiva y que es posible de ser computada mediante una máquina de Turing: "La evidencia de que el análisis es completo, es decir, que para cualquier función que sea evidentemente calculable puede hallarse una máquina de Turing que la compute". (Kleene, 1952:322)

Asimismo, Kleene propone la tesis Church-Turing en el teorema XXX, indicando la equivalencia de la tesis de Church con lo enunciado por Turing.

> La tesis de Turing de que toda función que sea naturalmente considerada como computable, es computable en el sentido por él especificado, esto es, computable por una maquina de Turing, es equivalente a la tesis de Church por virtud del teorema XXX. (Kleene, 1952:340).

Kleene estudió permanentemente la computabilidad de las funciones matemáticas, incorporando el concepto abstracto de máquina, que es una definición que sería parte de la teoría de autómatas, que trata de los algoritmos que trabajan secuencias de caracteres expresadas en la forma normal de Kleene: Cadenas de caracteres que tienen representaciones de formas recursivas. Así mismo, en la definición de Turing se encuentra el concepto de algoritmo como secuencia de instrucciones, siendo interpretado como equivalente a las funciones recursivas, ganando aceptación en la comunidad científica en computación, interpretándose la tesis de Church-Turing en la definición adecuada que relaciona el concepto de algoritmo con el significado de computabilidad.

Hoy la definición de computabilidad tiene significados que no pueden desligarse de la computadora y su funcionamiento. El concepto de computable está incorporado en nuestro lenguaje cotidiano. Algunos científicos y filósofos de la ciencia de la computación cuestionan el término de computabilidad en su única equivalencia con las funciones recursivas,

30 Radiolario es un grupo de organismos microscópicos, unicelulares (protozoos) marinos.

31 El término *Computer Science* fue acuñado por George Elmer Forsythe, matemático especializado en análisis numérico que fundó uno de los primeros departamentos de Ciencia de la Computación.

como Robert Soare[32], quien sostiene: "Específicamente nosotros recomendamos: El término recursivo ya no lleva el significado adicional de computable o decidible" (Robert Soare, 1996)

Es importante considerar que el concepto de computabilidad, expresada en la tesis Church-Turing, se construye entre los años 1930 a 1960, y se considera que en este último año nace la disciplina de la ciencia de computación (en Estados Unidos). La computabilidad en un contexto histórico, se presentó en la búsqueda de la solución del *Entscheidungsproblem*, en medio de la crisis de los fundamentos de las matemáticas, expresadas en las tres corrientes filosóficas en las matemáticas, lográndose la fabricación de la computadora.

XI. COMPUTABILIDAD

Sostenemos que el concepto de computabilidad es más amplio a lo expresado mediante funciones recursivas, entendemos que la utilización de este tipo de función matemática en el inicio de la ciencia de la computación, inspiró y posibilitó una teoría de la computación. Se interpretó como equivalentes y coextensivas la función efectiva y el procedimiento efectivo, ambas referenciando a la función computable.

El procedimiento efectivo es la mejor expresión que elucida el algoritmo, como procedimiento automático, en cierto sentido relacionado a la axiomatización y deducción de teoremas de una teoría formalizada.

La tesis de Church-Turing, establece la equivalencia de los conceptos de función efectiva y procedimiento efectivo, resultando una definición básica y abriendo un campo de estudio en la teoría de la recursividad, tratado en las matemáticas, específicamente en las metamatemáticas.

Jack Copeland en su artículo publicado en el 2004, en el libro editado por Luciano Floridi, *Philosophy of Computing and Information*, expresa sobre la importancia de la máquina de Turing con respecto al concepto de programa, así nos dice: "El principio básico de la computadora moderna, es la idea de controlar las operaciones de la máquina por medio de un programa de instrucciones de códigos almacenados en la memoria de la computadora que fue pensado por Alan Turing en 1935" (Copeland, 2004:3)

32 Robert I. Soare, matemático, Universidad de Princeton en 1963, PhD. en Matemáticas en la Universidad de Cornell en 1967, es profesor de matemáticas de la Universidad de Chicago, tiene diversas publicaciones: Teoría de la recursividad y los cortes de Dedekind, La computabilidad y la recursividad, historia y el concepto de computabilidad.

Al hablar de la computabilidad referimos al de algoritmo como procedimiento que obtiene respuesta a un problema, mediante instrucciones expresadas en un texto finito, que para su ejecución solo basta seguir las instrucciones tal como se indican.

El concepto de procedimiento efectivo es matemático, pero resulta que los matemáticos cuando defienden un procedimiento, no necesariamente es riguroso, requiere de explicación, y si se habla de reglas estas no indican el orden preciso en que deben ser ejecutadas.

La computabilidad en relación a las funciones computables, refiere a números naturales, en contraste a los números reales, dado que este no es numerable, denominado inconmensurable[33], tal como lo indico Cantor[34].

Turing verso su investigación sobre los números computables con aplicación al *Entscheidungsproblem*, siendo equivalente al resultado de la investigación de Church, tal como lo indicó en una nota al final de su trabajo, pero Gödel pensó en forma distinta, expresando que el procedimiento mecánico definido por Turing precisa la definición de un sistema formal, en relación al concepto de algoritmo. Al respecto, Sieg[35] indica:

> Gödel subrayó la importancia del análisis de Turing, repetida y enfáticamente. Afirmó en 1964, que sólo el trabajo de Turing proporciona "una precisa, sin duda y adecuada definición del concepto general de sistema formal". Ya que un sistema formal para Gödel, es sólo un procedimiento mecánico para producción de teoremas, la adecuación de esta definición recae perfectamente sobre el correcto análisis de Turing de los procedimientos mecánicos (Sieg Wilfried, 2004:37)

El concepto de paso a paso, expresado por Church y Turing, fue considerada por una parte de la comunidad científica, como complementaria y por otros como coextensivas, razón por la cual se entiende la

33 Conjunto numerable o contable, cuando sus elementos pueden ponerse en correspondencia a los números naturales. Cantor fue el primero que uso esta definición, tal como lo muestra en su artículo en 1874 *Über reine Eignschaft des inbegriffes aller reellen algebraischen* Zahlen Journal de Crelle 77, p258-262.

34 Georg Cantor, nacido en San Petersburgo, 3 de marzo de 1845, fallece en Halle, 6 de enero de 1918. Matemático alemán, uno de los inventores de la teoría de conjuntos.

35 Wilfried Sieg, profesor de filosofía en Universidad Carnegie Mellon, Pittsburgh desde 1985, matemático, físico y lógico. PhD. en Stanford University, en Filosofía, Lógica Matemática, MS, matemáticas y lógica, en Westfälische Wilhelms-Universität, Münster, BS, Matemática y Física en Freie Universität, Berlín.

tesis Church-Turing, pero es en la máquina de Turing donde se expresa adecuadamente el concepto de paso a paso, en la medida de que se sigue en forma mecánica la ejecución de operaciones elementales, mientras que en las funciones recursivas, se requiere del entendimiento de la función. Sieg enuncia que es Turing quien logra el concepto del paso a paso:

> Al examinar el análisis y recursividad de Turing, encontraremos la clave para responder a la pregunta que he planteado en la diferencia entre las propuestas de Church y la de Turing. Muy brevemente: Turing profundizo el argumento de paso a paso de Church operaciones mecánicas subyacentes a los pasos elementales, mediante la adecuada formulación de las limitaciones que garantizan su recursividad (Sieg Wilfried, 2004:60-61)

Church consideraba que las funciones recursivas eran la mejor forma de obtener un cálculo efectivo de funciones, específicamente el cálculo lambda, y lo que hizo Turing fue el precisar los pasos elementales en el concepto paso a paso: “Church propuso en una reunión de la American Mathematical Society en abril de 1935, que la noción de una manera efectiva de calcular la función de enteros positivos debía ser identificado con la de una función recursiva.". (Sieg Wilfried, 1992:1).

El planteamiento de la similitud entre la máquina de Turing y la máquina de lápiz y de papel con presencia humana para ejecutar las instrucciones, resulta ser una relación entre el hombre que ejecuta un procedimiento y el concepto de algoritmo: “Turing…, subrayó en su documento de 1953. Precisando que el concepto (recursividad, Turing computabilidad) va a la captura de procesos mecánicos que pueden ser llevados a cabo por los seres humanos” (Sieg Wilfried, 1997:7).

Establecemos que Turing tenía claramente el concepto de computabilidad en correspondencia a la ejecución mediante lápiz y papel y que un hombre siga instrucciones. En palabras de Soare:

> En 1935 Turing y todos los demás utilizaron el término "computadora" para una idealización del cálculo de un humano con material extra como el lápiz y papel, calculadora o un escritorio, siendo un significado muy diferente a la utilización de la palabra al día de hoy (Soare Robert, 1996: 9)

La máquina de Turing opera signos, que no son números, carece de toda significación al momento que ejecuta instrucciones en un orden establecido, y los resultados no le dicen nada, lo único que determina es que ha terminado, mientras que en el cálculo de las funciones recursivas, se requiere de interpretación y jerarquización para la valuación de la función. Razón por la cual sostenemos que la teoría de la computabilidad es de diferente significado al de la teoría de la recursividad, Soare propone desligar ambos conceptos, así nos dice: “Algunos lógicos han señalado que es curioso que el tema de la teoría computabilidad (en adelante llamada "la materia") sea llamada "teoría de la función recursiva" o "teoría de la recursividad" en lugar de lo que sería más natural "teoría de la computabilidad"” (Soare Robert, 1996:2)

Church y Kleene consideraron que el cálculo lambda expresa adecuadamente las funciones recursivas generales, con relación al cálculo efectivo. Pero Gödel tenía dudas sobre la veracidad del concepto, llegando a calificar lo afirmado por Church como “totalmente insatisfactoria”, así lo menciona Soare sobre la conversación sostenida entre Church y Gödel (hay que considerar que ambos frecuentaban la misma facultad de matemáticas de la universidad de Princeton), así nos dice:

> En 1930 Church había estado estudiando una clase de funciones de cálculo efectivo llamadas funciones lambda-definible. El alumno de Church, Kleene demostró en 1933 que un gran número de clases de funciones teóricas eran lambda-definible. Con la fuerza de esta evidencia, Church propuso a Gödel alrededor de marzo de 1934 que la noción de "efectivamente calculable" se identifica con "lambda-definible" una sugerencia que Gödel, rechazó como "totalmente insatisfactoria" (Soare Robert, 1996:7)

Tanto Church como Kleene consideraron que la recursividad expresa el concepto de computabilidad, ampliándolo al manejo de símbolos como la Forma Normal[36] de Kleene, que es una formulación de cadenas de caracteres, que tienen una notación recursiva.

Según Robert Soare, Church utiliza el término recursivo como adverbio[37] en relación al término computable en la teoría de la función recursiva (definida por

36 Forma Normal de Kleene refiere a que una cadena de caracteres pueden ser expresadas con operaciones recursivas así tenemos a+a+a+… a =a*

37 Adverbio, es una parte invariable de la oración que califica o determina la significación del verbo, del adjetivo y veces de otro adverbio. Los adverbios sirven para indicar circunstancias del verbo.

Kleene). Ambos (Church y Kleene) estaban convencidos de que expresaban en mejor sentido el concepto de computabilidad, pero ocurrió, que en ningún momento Gödel ni Turing utilizaron el concepto de computable en alusión a las funciones recursivas.

> Church introdujo el uso de "recursivo" como un adverbio que significa "computable" por ejemplo, "Recursivamente enumerables" y más tarde Kleene introdujo el término teoría de la función "recursiva"… Gödel y Turing nunca utilizaron el término "recursivo" en el sentido computable y explícitamente rechazaron tales sugerencias (Soare Robert, 2007:2)

Gödel adiciona en 1963, una nota suplementaria al final de su trabajo sobre sentencia formalmente indecidibles que fuera escrita en 1931, en la que indica que el trabajo de Turing de 1936, expresa un sistema formal en forma precisa y adecuada en relación a sus teoremas VI y XI en la que hay sentencias aritméticas indecidibles y que no pueden probar su consistencia en el sistema. Para Gödel un sistema formal está relacionado al concepto de máquina de Turing.

> Como consecuencia de avances posteriores, en particular del hecho de que gracias a la obra de A.M. Turing ahora disponemos de una definición precisa e indudablemente adecuada de la noción general de sistema formal, ahora es posible dar una versión completamente general de los teoremas VI y XI. Es decir, se puede probar rigurosamente que en cada sistema formal consistente que contenga una cierta porción de teoría finitaria de números hay sentencia aritméticas indecidibles y que, además, la consistencia de cualquiera de esos sistemas no puede ser probada en el sistema mismo. (Gödel, en Mosterín 2006:87)

En 1964 Gödel proporciona a Martin Davis[38] diversas notas suplementarias para la reimpresión de algunos de sus trabajos. En opinión de Jesús Mosterín[39], había una importante posdata en la que Gödel propone identificar la noción de sistema formal con la de máquina de Turing: "… lo que equivale a identificar teoría formalizada con conjunto recursivamente enumerable- es decir, generable por una máquina de Turing" (Gödel en Mosterín, 2006:166)

38 Martin Davis, (nacido en 1928 en Nueva York) matemático estadounidense conocido por su trabajo relacionado con el décimo problema de Hilbert. Obtuvo su PhD en la Universidad de Princeton en 1950 y su tutor fue Alonzo Church.

39 Jesús Mosterín nació en Bilbao en 1941. Filósofo español, estudió en España, Alemania y Estados Unidos. Obtuvo la cátedra de Lógica y Filosofía de la Ciencia en la Universidad de Barcelona.

Gödel dice que la obra de Turing, permite una definición precisa, adecuada e incuestionable del concepto general de sistema formal, en relación al procedimiento mecánico.

> La obra de Turing proporciona un análisis del concepto de <<procesamiento mecánico>> (<<algoritmo>>, <<procedimiento computacional>> o <<procedimiento combinatorio finito>>). Se ha probado que este concepto es equivalente al de <<Máquina de Turing>>. Puede definirse un sistema formal simplemente como un procedimiento mecánico para producir filas de signos, llamadas formulas deducibles. (Gödel, en Mosterín 2006:197)

De otro lado, Gödel discrepa con Turing con respecto a ideas sobre la mente, específicamente en lo referente a la inteligencia en las máquinas, que son expresados por Turing con respecto al de procedimiento mecánico, así lo hace saber: "Téngase en cuenta que los resultados mencionados en esta posdata no establecen límites de la capacidad de la razón humana, sino más bien de la posibilidades del puro formalismo en matemáticas" (Gödel, en Mosterín 2006:197).

La diferencias entre Gödel y Turing, radica en la idea de Turing en equiparar en cierto sentido la mente humana con una máquina de Turing, Gödel refiere que se estaría limitando a la mente humana, así lo expresa Hao Wang[40] en su libro *A Logical Journey from Gödel to Philosophy*: "Con respecto a la cuestión central de superioridad de la mente sobre las computadoras, Gödel nota el error en la filosofía de Turing" (Hao Wang; 1974:202).

XII. RESUMIENDO

Hoy las computadoras son herramientas de apoyo a diversas actividades: calcular, almacenar, editar documentos, medio de comunicación entre otros. Están presentes en nuestra actividad cotidiana.

Para la comunicación entre las computadoras se utilizan algoritmos, basado en la estructura de capas, como exigen los protocolos de comunicación[41]. El concepto de protocolo no es un algorítmo, es un acuerdo que establece reglas, pero en cada nivel

40 Hao Wang, nacido el 20 de Mayo de 1921 en Jinan China y fallece el 13 de mayo de 1995 en EEUU, Chino americano, lógico, filósofo y matemático. Obtiene su maestría en matemáticas en la universidad de Tsinghua en 1945, se doctora en lógica en la universidad de Harvard en 1948.

41 Protocolo de comunicación, es un conjunto de reglas usadas para la comunicación de las computadoras mediante una red. El protocolo son las reglas que dominan la sintaxis, semántica y sincronización de la comunicación, permitiendo la comunicación, sincronización y transferencia de datos.

del mismo corresponde hablar de algoritmos, como ocurre en los temas académicos de reconocimiento de mensajes, detección de errores, recuperación de errores, entre otros.

La ciencia de la computación se inicia conociendo sus límites, en cuanto a lo computable y lo no computable. El *Entscheidungsproblem* propuesto por Hilbert y el teorema de la incompletitud de Gödel, están incorporadas a las conclusiones proporcionadas por Church y Turing, en referencia a la función computable, en este sentido, la ciencia de la computación a diferencia de otras ciencias, empezó con un límite en el horizonte, se avanzó en el campo de la lógica, en teorías de la demostración, teoría de la información, hasta plantearse los temas sobre la complejidad computacional. Así Sieg menciona:

> Entscheidungsproblem de Hilbert, el problema de la decisión en la lógica de primer orden, fue una cuestión que requiere una precisa caracterización de los "métodos efectivos", véase la tesis de la Church-Turing. Aunque parcial, se encontraron respuestas positivas durante la década de 1920, Church y Turing demostraron en 1936 que el problema general es indecidible. El resultado y las técnicas que intervienen en su prueba (por no hablar de conceptos muy matemáticos) inspirado en la investigación de la complejidad de la teoría de la recursión de conjuntos que llevó en primer lugar a la clasificación de la aritmética, híper-aritmética, jerarquías analíticas, y que más tarde de las clases de complejidad computacional. (Sieg Wilfried, 1997:4)

La ciencia de la computación tiene varios años de haberse constituido, podemos decir que son 80 años, desde que se definió la máquina de Turing y se formulara la teoría de la computabilidad como teoría de funciones recursivas, la tesis Church-Turing resulto ser el concepto base. En los últimos 20 años la computabilidad mejoro en diversos sentidos a la luz de las mayores capacidades de las computadoras, comunicaciones, y almacenamiento de datos; proporcionado nuevos horizontes a la ciencia de la computación.

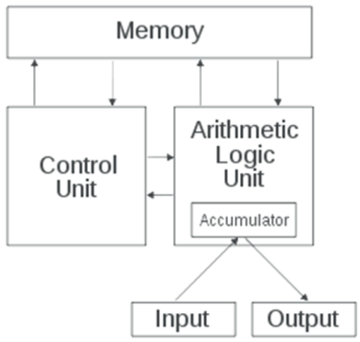
En palabras de Robert Soare: “la teoría de computabilidad se puede dividir en tres períodos. 1. Era de definibilidad Lambda 1931-1935. 2. Era de la teoría de la Recursión 1935-1995. 3. Era de la Computabilidad 1996-al presente” (Soare Robert, 2007:1)

Sostenemos que la máquina de Turing expresa adecuadamente el concepto de computabilidad, porque toma en cuenta: i) procedimiento efectivo, definición que trata sobre la ejecución secuencial de instrucciones, paso a paso. ii) máquina de lápiz y papel como procedimiento que realiza un hombre, iii) recursividad y recurrencia concepto matemático de funciones y relaciones computables expresada en procesos de forma recurrente. iv) interacción y comunicación como elementos necesarios en una máquina y entre máquinas. Martin Davis acota al respecto:

> Alan Turing descubre lo universal (o de uso general) de la computadora digital como una abstracción matemática... este trabajo abstracto ayudó a Turing y John von Neumann hacia la concepción moderna de la computadora electrónica (Martin Davis, 2006:125)

Las computadoras electrónicas tienen como arquitectura la estructura definida por von Neumann: Unidad de control, memoria, unidad aritmética lógica y dispositivo de entrada y de salida de datos.

Figura 6
Arquitectura de John von Neumann



La arquitectura de la computadora de von Neumann fue posible sobre la base del trabajo de Turing, como lo que afirma Martin Davis.

> Prácticamente todos los ordenadores de hoy de más de US$ 10 millones de las súper computadoras hasta los chips de teléfonos celulares y los furbies, tienen una cosa en común: todos ellos son "máquinas de Von Neumann", variaciones sobre la arquitectura básica de John von Neumann, construidas sobre la base del trabajo de Alan Turing, establecido en 1940 (Martin Davis, 2006:126)

Cuando uno revisa la arquitectura de las computadoras según el modelo de von Neumann, tomamos en cuenta en primer lugar, que refiere a componentes físicos y en segundo lugar a la definición los datos y programas como elementos externos. Esta separación es significativa dado que conceptúa a las computadoras en dos áreas tecnológicas (la ingeniería de las máquinas y la ingeniería de los programas).

Los datos en las computadoras se organizan de diferentes formas y se definen mediante estructuras que se relacionan a reglas lógicas. Destacamos de la máquina de Turing, que los datos se encuentran en símbolos que se almacenan en la memoria, mientras que el programa son parte de la definición de la máquina.

Consideramos que los datos son la parte variable, no es físico y los programas son en cierto sentido una parte fija, pero en la teoría de Turing se considera que los programas pueden ser variables como aquellos que "aprenden" (en el campo de la Inteligencia Artificial), es decir, programas que cambian su estructura lógica de la ejecución. En la máquina de Turing están presentes los componentes mencionados: físico, datos y programa, así Martin Davis precisa:

> ...las tres categorías, la máquina era un objeto físico… hardware. El programa fue el plan para hacer una computación... Los datos eran el ingreso numérico. La máquina universal de Turing demostró que la distinción de estas tres categorías es una ilusión... Por último, la máquina universal en sus acciones paso-a-paso ve que... el código de máquina como datos para ser trabajado. Esta fluidez... es fundamental para la práctica contemporánea del computador (Martin Davis, 2006:128)

En el propósito de calcular, está abierto el debate sobre la potencia para determinar si una función es computable, dado que en la máquina inicial de Turing, el problema se plantea con el proceso que no se detiene por ser no es computable, pero en la máquina-O, tiene la capacidad para determinar si una función computable. Es alrededor del dispositivo Oráculo como computadora en la que se manifiesta los temas como hipercomputación, así tenemos en palabras de Martin Davis: "sería tonto afirmar que no habrá ningún futuro dispositivo que será capaz de calcular lo no computable. De hecho, la necesidad actual del movimiento de la "hipercomputación" llama nuestra atención" (Martin Davis, 2006:130)

Para la definición de la arquitectura de una computadora, von Neumann profundiza sobre el código de la máquina en relación a la ejecución de procesos, esta refiere a la imitación de otra máquina (máquina-U), en el sentido que al ejecutar una secuencia de código, todas obtendrían el mismo resultado. El código en las computadoras, es la característica significativa en el concepto de la máquina: "En contraste con los códigos completos, existe otra categoría de códigos mejor designados como códigos cortos. Estos están basados en la siguiente idea." (von Neumann, 1958:71)

La recursividad no refleja completamente el sentido de un código máquina, es siguiendo esta reflexión donde encontramos adecuada la opinión de von Neumann resaltando lo propuesto por Turing, en el sentido que el concepto de máquina contribuye a la imitación de otras máquinas, así tenemos:

> El resultado importante de Turing es esta manera en la que la primera máquina puede imitar el comportamiento de cualquier otra máquina. La estructura de orden es lo que causa a seguir que puede hacer completamente diferente a una característica de la primera máquina que está involucrada. Así, la estructura de orden a la que se refiere, puede hacer frente a las órdenes en un carácter mucho más complejo que son característicos de la primera máquina: cada una de estas órdenes de la máquina secundaria puede implicar la realización de varias operaciones por la máquina mencionada en primer lugar… códigos cortos se han desarrollado como una ayuda a la codificación, es decir, son el resultado de la voluntad de ser capaz de código más breve para un equipo que posee su sistema de orden natural que permitiría, tratándola como si se tratara de una máquina diferente con un sistema más conveniente (von Neumann, 1958:73)

Consideramos que hemos profundizado en la diferenciación de conceptos que contribuyen al esclarecimiento del significado de la computabilidad, el ignorarlos reducen la interpretación en diversos temas que se dan en la ciencia de la computación.

Creemos que la diferenciación entre una función efectivamente computable que refiere a las funciones recursivas como un cálculo efectivo, enunciado por Church en el cálculo lambda, y el procedimiento efectivo, definición dada por Turing en el sentido de ejecutar exitosamente una secuencia de instrucciones en una máquina de Turing, contribuye a la discusión filosófica sobre la ciencia de la computación y a la rica tradición de la filosofía.

XIII. BIBLIOGRAFIA

[1] Coello Coello, C. (2004). *Breve historia de la computación y sus pioneros.* México DF, México: Fondo de Cultura Económica.

[2] Copeland Jack. (2004). *The Essential Turing Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus the Secrets of Enigma.* Oxford New York, USA: Clarendon Press-Oxford.

[3] __________ (2004). *The Church-Turing Thesis*. NeuroQuantology, Issue 2, PP 101-115.

[4] Church Alonzo. (1936). *An Unsolvable Problem of Elementary Number Theory*, www.jstor.org/stable/2371045. (2008, 25 de agosto).

[5] Davis Martin. (2006). *What is Turing Reducibility?* Notices of the AMS, Volume 53, Number 10, November, PP 1218-1219.

[6] Ferrater Mora. (2004). *Diccionario de Filosofía.* Barcelona, España: Editorial Ariel S.A.

[7] Hilbert David. (1900). *Mathematical Problems Lecture delivered before the International Congress of mathematical at Paris 1900*. aleph0.clarchku.edu/˜djoyce/hilbert/problems.html. (2008, 18 de marzo).

[8] Kleene (1952). *Introducción a la Metamatemáticas.* (1a ed.) (Garrido Manuel trad.) Madrid España: Tecnos S.A.

[9] Mosterín Jesús. (2006). *Kurt Gödel Obras Completas* (1a ed.). Madrid, España: Alianza Editorial S.A.

[10] Perazzo, Roberto. (1994). *De Cerebros Mentes y Máquinas*. Buenos Aires, Argentina: Fondo de Cultura Económica.

[11] Shannon Claude. (1956). *A Universal Turing Machine with Two Internal States.* Automa Studies, NJ Princeton: University Press pp:157-165.

[12] Sieg Wilfried. (1992). *Church Thesis, Consistancy, Formalization, Proof Theory: Dictonary Enties*.

[13] ________ (1997). *Step by Recursive Step: Church's analysis of effective calculability.* The Bulletin of Symbolic Logic, Volume 3, Number 2, June PP 154-180.

[14] ________ (1997). *Formal System, Church Turing Thesis & Gödel Theorems: Three contributions to the MIT.* Encyclopedias of Cognitive Science.

[15] ________ (2004). *Computability Theory,* The Material was Organized for four Seminars. November, University of Bologna.

[16] Soare Robert. (1996). *Computability and Recursion*. Bulletin of Symbolic Logic, Volume 2, Number 3, PP 284-321.

[17] ________ (2007). *Computability and Incomputability.* www.people.cs.uchicago.edu/~soare/History/siena.pdf. (2009, 15 de febrero)

[18] Strier Damian. (2002). *Procesos de Auto organización en Sistemas Biológicos.* Tesis de Grado de Doctor. Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales. Departamento de Fisca.

[19] Torretti Roberto. (1998). *El Paraíso de Cantor: La Tradición Conjuntista en la Filosofía de las Matemáticas*. Santiago, Chile: Editorial Universitaria, Universidad Nacional Andrés Bello.

[20] Turing, A. (1936). *On computable numbers, with application to the Entscheidungsproblem.* www.thocp.net/biographies/papers/turing_oncomputablenumbers_1936.pdf. (2008, 14 de junio).

[21] __________ (1938). *Systems of logic based on ordinals* www.turingarchive.org/browse.php/B/15. (2008, 14 de junio).

[22] __________(1948). *Intelligent Machinery*: www.alanturing.net/turing_archive/archive/1/132/L32-001.html. (2008, 14 de junio).

[23] __________ (1950). *Computing machinery and intelligence* www.loebner.net/Prizef/TuringArticle.html. (2008, 14 de junio).

[24] __________ (1951). *Can Digital Computer Think* www.turingarchive.org/browse.php/B/5. (2008, 14 de junio).

[25] __________ (1952). *The Chemical Basis of Morphogenesis* www.dna.caltech.edu/courses/cs191/paperscs191/turing.pdf. (2008, 14 de junio).

[26] __________ (1953). *The Chess.* http://www.turingarchive.org/browse.php/B/7 (2009, 20 de julio).

[27] __________ (1954). *Solvable and Unsolvable Problems.* http://www.turingarchive.org/browse.php/C/24. (2010, 29 de julio)

[28] Von Newmann John. (1958). *The Computer and the Brain.* (1a ed.) Yale USA: Yale University Press.

[29] Wang Hao. (1974). *A Logical Journey from Gödel to Philosophy*. Massachusetts, USA: Massachusetts Institute of Technology.

[30] Wittgenstein Ludwig. (1978). *Observaciones sobre los fundamentos de la matemática.* (Reguera Isidoro Trad.). Madrid, España: Alianza Editorial.

Fuentes de financiamiento:

Propia.

Conflictos de interés:

El autor declara no tener conflictos de interés.