

**Universidad Autónoma del Estado de México**

**Unidad Académica Profesional Tianguistenco**

**Ingeniería en software**

**Unidad de aprendizaje:**

Programación paralela

**Profesor:**

Gustavo Gómez Vergara

**Ensayo de:**

“Aportaciones de Alan Turing y Jhon Von Neumann a la computación

**Alumno:**

Julio Cesar Piña Estrada

**Fecha de entrega:** 29/octubre/2020

|  |
| --- |
| **Introducción** |

A continuación, en este documento se presentaran los aportes mas significativos a la computación y a la informática, por parte de los matemáticos Alan Turing y John von Neumann, además nombrar cada uno de sus aportes mas importantes a la computación, se dará una breve descripción de cada una de ellas, cabe mencionar que algunas necesitan de una descripción más amplia que otros, debido a que contienen mas procesos, o por otro lado necesitan de conceptos externos para su entendimiento del aporte.

Antes de iniciar con el tema principal, hay que conocer un poco sobre estos dos grandes personajes.

Alan Turing fue un brillante matemático, criptoanalista e informático teórico nacido el veintitrés de junio de 1912 en Maida Vale un distrito residencial al oeste de Londres. Turing, además de ser un brillante científico era homosexual, lo cual le costó la vida el siete de junio de 1954.

* Turing es mundialmente conocido por cuatro hechos:
* Formalizó los conceptos de algoritmo y computación con su máquina de Turing
* Es considerado el padre de la inteligencia artificial
* Su participación en el equipo de criptoanálisis de la máquina de criptografía alemana Enigma fue clave

Jhon Von Neumann fue un matemático húngaro nacionalizado estadounidense. Nacido en el seno de una familia de banqueros judíos, dio muestras desde niño de unas extraordinarias dotes para las matemáticas

Entre 1944 y 1946 colaboró en la elaboración de un informe para el ejército sobre las posibilidades que ofrecía el desarrollo de las primeras computadoras electrónicas; de su contribución a dicho desarrollo destaca la concepción de una memoria que actuase secuencialmente y no sólo registrara los datos numéricos de un problema, sino que además almacenase un programa con las instrucciones para la resolución del mismo.

Se interesó también por la robótica, y en 1952 propuso dos modelos de máquinas autorreproductoras

|  |
| --- |
| **Desarrollo** |

Alan Turing

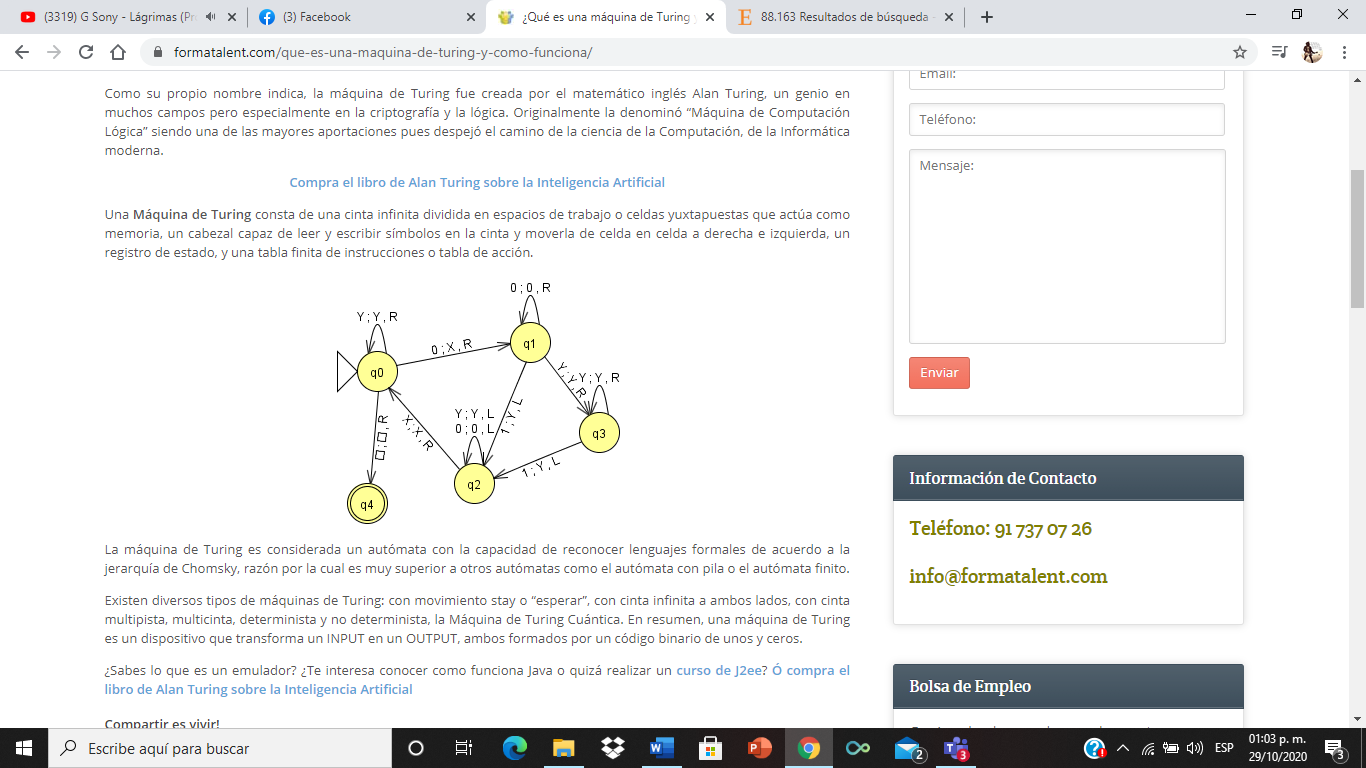
Máquina de Turing

Fue creada en 1936, en realidad un modelo matemático consistente en un autómata que es capaz de “implementar cualquier problema matemático expresado a través de un algoritmo.

La máquina de Turing destaca por su simplicidad pues manipula símbolos sobre una tira de cinta siguiendo una serie de reglas. A pesar de esta simplicidad, una máquina de Turing puede adaptarse para que simule la lógica de cualquier algoritmo de computador, de ahí su enorme potencial y valor.

Originalmente la denominó “Máquina de Computación Lógica” siendo una de las mayores aportaciones pues despejó el camino de la ciencia de la Computación, de la Informática moderna.

Una Máquina de Turing consta de una cinta infinita dividida en espacios de trabajo o celdas yuxtapuestas que actúa como memoria, un cabezal capaz de leer y escribir símbolos en la cinta y moverla de celda en celda a derecha e izquierda, un registro de estado, y una tabla finita de instrucciones o tabla de acción [1].



La máquina de Turing es considerada un autómata con la capacidad de reconocer lenguajes formales de acuerdo a la jerarquía de Chomsky, razón por la cual es muy superior a otros autómatas como el autómata con pila o el autómata finito [2].

Bombe

Alan Turing fue el principal responsable de descifrar Enigma, el código secreto utilizado por el Ejército alemán en la Segunda Guerra Mundial, contribuyendo con ello a acortar la guerra, cuando trabajo en descifrar los códigos nazis. Para descifrar estos códigos y prevenir ataques y movimientos militares nazis, Turing diseño la Bombe, una máquina electromecánica que eliminaba claves enigma candidatas. Esta máquina por los criptógrafos para leer las transmisiones Enigma. Gracias al descubrimiento de esos códigos, la guerra pudo ser terminada pronto [3].

¿Qué es enigma?

Después de la primera guerra mundial, el inventor alemán Arthur Scherbius y su amigo Richard Ritter fundaron una empresa de ingeniería y crearon la máquina Enigma con la finalidad de venderla no solo al ejército sino también a muchas empresas del país.

Enigma es como una máquina de escribir con un teclado luminoso. Cada vez que se pulsa una tecla, se enciende una letra, explica. Así, por ejemplo, si queremos codificar la palabra HELLO, la hache se convierte en una I.

Bombe fue una herramienta esencial en el proceso de romper las señales de Enigma; no es una computadora, no realiza cálculos numéricos. Su función era llevar a cabo una búsqueda sistemática basada en ciertas consideraciones lógicas, y así encontrar partes de las claves Enigma que se habían utilizado para cifrar los mensajes interceptados.

El diseño inicial del bombe fue producido en 1939 en el Government Code and Cypher School en Bletchley Park por Alan Turing. El diseño de ingeniería y la construcción fue trabajo de Harold Keen (British Tabulating Machine Company). Era un desarrollo mejorado de un dispositivo que había sido diseñado en 1938 por el criptologista polaco Marian Rejewski del Biuro Szyfrów, y conocido como la «bomba criptológica» (“bomba kryptologiczna”), que servía para eliminar una gran cantidad de claves enigmas candidatas [4].

El Test de Turing

En 1950, Turing abordó el problema de la inteligencia artificial y propuso el experimento que hoy se conoce como el test de Turing cuya intención era intentar definir un estándar por el cual se pudiera conocer si una máquina podía ser llamada “inteligente“ [5].

El test de Turing (o prueba de Turing) es una prueba de la habilidad de una máquina para exhibir un comportamiento inteligente similar al de un ser humano de tal manera que, interactuando con ella en una conversación, una persona pueda determinar si su interlocutor es una máquina o una persona.

Una buena forma de apartarse de la espinuda pregunta sobre si los computadores piensan es a través de rechazar el análisis del uso común de términos como 'piensa' y 'máquina', ya que dicho examen podría llevar a una encuesta sobre definiciones y usos. Turing manifiesta explícitamente la inconveniencia de debatir acerca del uso común de 'máquina' y 'piensa' de la siguiente manera:

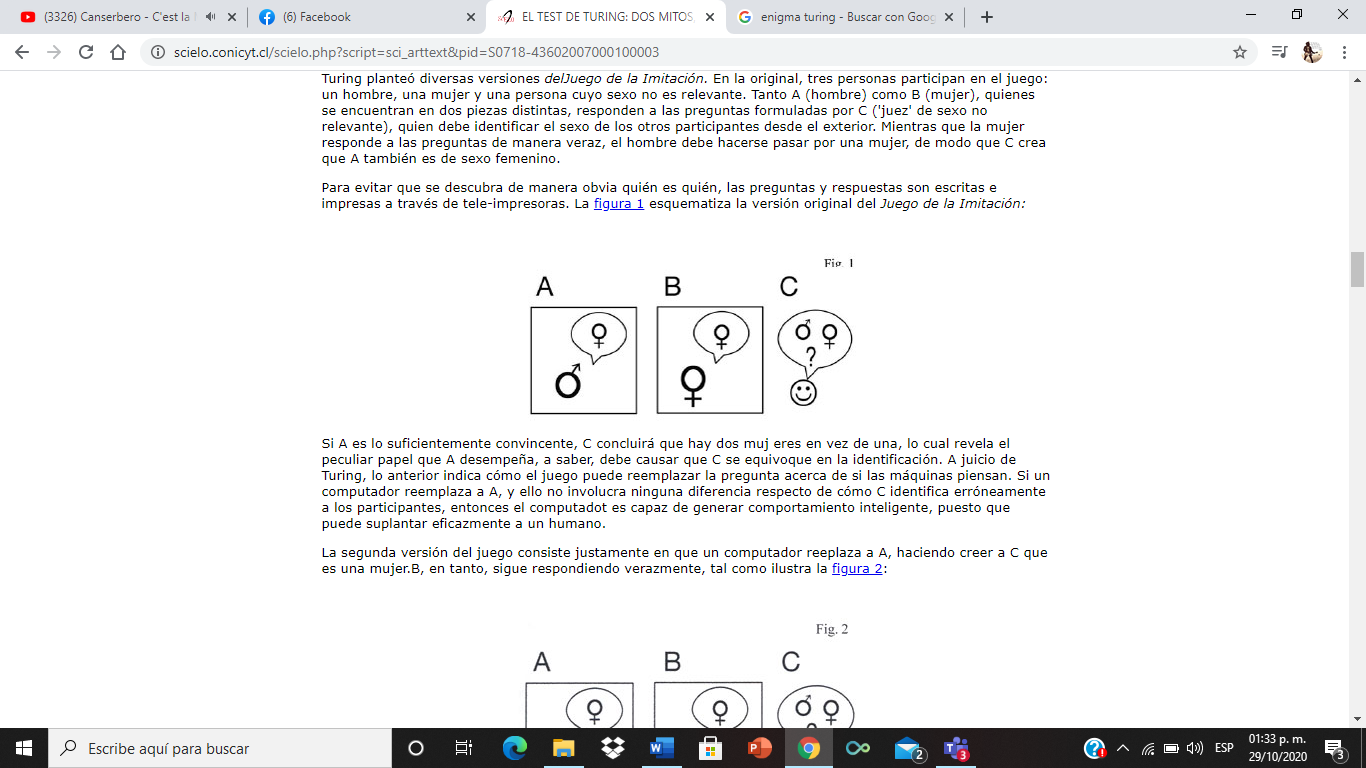
|  |
| --- |
| Si la exploración del significado de términos como 'máquina' y 'piensa' se debe efectuar a partir del análisis de cómo estos se usan regularmente, es difícil evitar la conclusión de que el significado y la respuesta a la pregunta, ¿pueden pensar las máquinas?, debe encontrarse a través de una investigación estadística similar a una encuesta Gallup [6] |

La idea consistía en que se puede llegar a decir que una computadora “piensa“ si un interrogador humano no pudiera distinguir a través de una conversación, si estuviera conversando con un ser humano o con una máquina. Turing propuso que en lugar de simular una mente artificial adulta, era mejor simular una mente infantil para someterla más tarde a un proceso de aprendizaje.

La desazón ante tal posibilidad motiva a Turing a plantear un test, entendido precisamente como un reemplazo de la mencionada pregunta. Turing, entonces, formula un método, el denominado *Juego de la Imitación [Imitation Game]*, que consiste en que ciertas personas juegan a descubrir la identidad de uno de ellos mediante un breve interrogatorio. Es importante destacar que las bases de taljuego fueron ideadas antes del Test de Turing, cuando el matemático-filósofo propone un experimento en el cual hay tres jugadores de ajedrez y uno de ellos debe adivinar con quién está jugando.

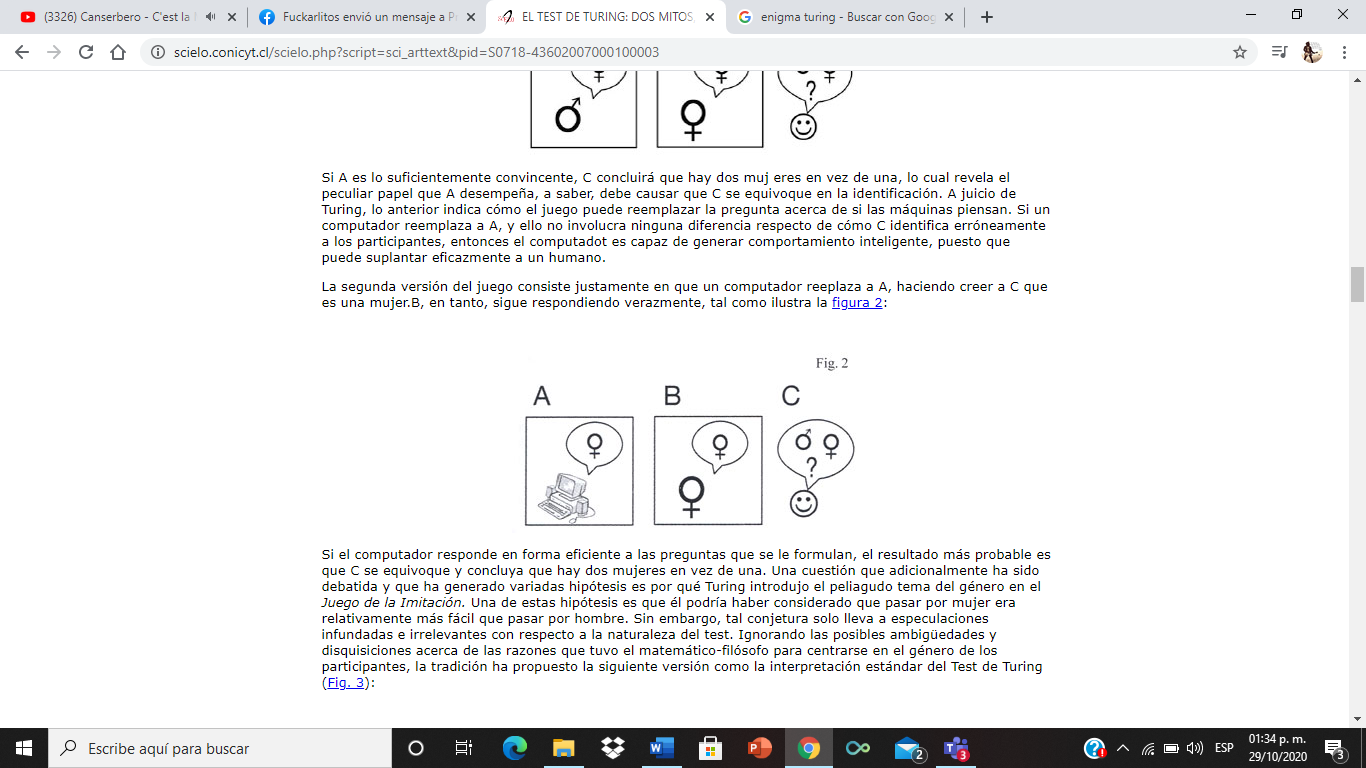
Turing planteó diversas versiones del Juego de la Imitación. En la original, tres personas participan en el juego: un hombre, una mujer y una persona cuyo sexo no es relevante. Tanto A (hombre) como B (mujer), quienes se encuentran en dos piezas distintas, responden a las preguntas formuladas por C ('juez' de sexo no relevante), quien debe identificar el sexo de los otros participantes desde el exterior. Mientras que la mujer responde a las preguntas de manera veraz, el hombre debe hacerse pasar por una mujer, de modo que C crea que A también es de sexo femenino [7]

Para evitar que se descubra de manera obvia quién es quién, las preguntas y respuestas son escritas e impresas a través de tele-impresoras.



Si A es lo suficientemente convincente, C concluirá que hay dos muj eres en vez de una, lo cual revela el peculiar papel que A desempeña, a saber, debe causar que C se equivoque en la identificación. A juicio de Turing, lo anterior indica cómo el juego puede reemplazar la pregunta acerca de si las máquinas piensan. Si un computador reemplaza a A, y ello no involucra ninguna diferencia respecto de cómo C identifica erróneamente a los participantes, entonces el computadot es capaz de generar comportamiento inteligente, puesto que puede suplantar eficazmente a un humano [7].

La segunda versión del juego consiste justamente en que un computador reeplaza a A, haciendo creer a C que es una mujer.B, en tanto, sigue respondiendo verazmente



Si el computador responde en forma eficiente a las preguntas que se le formulan, el resultado más probable es que C se equivoque y concluya que hay dos mujeres en vez de una. Una cuestión que adicionalmente ha sido debatida y que ha generado variadas hipótesis es por qué Turing introdujo el peliagudo tema del género en el Juego de la Imitación. Una de estas hipótesis es que él podría haber considerado que pasar por mujer era relativamente más fácil que pasar por hombre. Sin embargo, tal conjetura solo lleva a especulaciones infundadas e irrelevantes con respecto a la naturaleza del test. Ignorando las posibles ambigüedades y disquisiciones acerca de las razones que tuvo el matemático-filósofo para centrarse en el género de los participantes, la tradición ha propuesto la siguiente versión como la interpretación estándar del Test de Turing.

Esta versión revela de manera indiscutible qué está detrás del Juego de la Imitación. El hecho de que la finalidad del juego sea inducir a C a efectuar identificaciones erróneas sugiere que el Test de Turing intenta verificar la existencia de inteligencia en base a la imitación y la simulación eficaz.

La prueba se ha convertido en la vara con la que científicos y desarrolladores miden los avances que, en materia de Inteligencia Artificial, hemos venido desde hace unos 30 años pero que se han disparado en los últimos 3 años [8].

Ajedrez por computadora

Siendo un experto en el campo de la tecnología, en 1950 intentó crear un programa de ajedrez que se pudiera desarrollar entre un ordenador y una persona. Si bien los primeros intentos fracasaron, fue un gran adelanto para la época, y serviría de base para crear lo que ahora es tan común en los videojuegos: la creación de otro personaje con el que peleas pero en realidad es un personaje “manejado” por la computadora. Mismo concepto que se relaciona con las aplicaciones del Test de Turing [9].

Desarrolló el primer programa de ordenador de la historia para jugar al  ajedrez. Fue a  finales de los años 40, y está descrito en su artículo Digital Computers Applied to Games (que se publicó en el libro Faster than Thought, editado por B. V. Bowden, Pitman, Londres 1953). Allí Turing sienta lo que serán las bases de los programas posteriores de ajedrez por ordenador: la simulación de secuencias de movimientos, la evaluación de los estados finales de esas secuencias y la propagación de esa evaluación a los estados directamente sucesores de la configuración actual de juego. Turing elegía el siguiente movimiento como el que conducía al estado de mejor evaluación entre todos los estados posibles.  
  
Aunque este programa no fue implementado en su tiempo (los ordenadores eran muy escasos y con capacidad muy limitada), en 1952 Turing jugó una partida con Alick Glennie en donde Turing simulaba (con papel y lápiz) la ejecución de su programa. Turing perdió en 30 movimientos, pero viendo la partida uno puede comprobar que el programa de Turing no hacía movimientos estúpidos [10].

Jhon Von Neumann

Arquitectura Neumann

Este describe una arquitectura de diseño para un computador digital electrónico con partes que constan de una unidad de procesamiento que contiene una unidad aritmético lógica y registros del procesador, una unidad de control que contiene un registro de instrucciones y un contador de programa, una memoria para almacenar tanto datos como instrucciones, almacenamiento masivo externo, y mecanismos de entrada y salida. El significado ha evolucionado hasta ser cualquier computador de programa almacenado en el cual no pueden ocurrir una extracción de instrucción y una operación de datos al mismo tiempo, ya que comparten un bus en común. Esto se conoce como el cuello de botella Von Neumann y muchas veces limita el rendimiento del sistema [11].

El diseño de una arquitectura Von Neumann es más simple que la arquitectura Harvard más moderna, que también es un sistema de programa almacenado, pero tiene un conjunto dedicado de direcciones y buses de datos para leer datos desde memoria y escribir datos en la misma, y otro conjunto de direcciones y buses de datos para ir a buscar instrucciones.

Un computador digital de programa almacenado es uno que mantiene sus instrucciones de programa, así como sus datos, en memoria de acceso aleatorio (RAM) de lectura-escritura. Las computadoras de programa almacenado representaron un avance sobre los ordenadores controlados por programas de la década de 1940, como la Colossus y la ENIAC, que fueron programadas ajustando interruptores e insertando parches, conduciendo datos de la ruta y para controlar las señales entre las distintas unidades funcionales. En la gran mayoría de las computadoras modernas, se utiliza la misma memoria tanto para datos como para instrucciones de programa, y la distinción entre Von Neumann vs. Harvard se aplica a la arquitectura de memoria caché, pero no a la memoria principal.

Una máquina de Von Neumann tenía 5 partes básicas: La memoria, la unidad Aritmética lógica, la unidad de control del programa y los equipos de entrada y salida. La memoria constaba de 4096 palabras, cada una con 40 bits (0 o 1). Cada palabra podía contener 2 instrucciones de 20 bits o un número entero de 39 bits y su signo. Las instrucciones tenían 8 bits dedicados a señalar el tiempo de la misma y 12 bits para especificar alguna de las 4096 palabras de la memoria.

Dentro de la unidad aritmética/lógica, el antecedente directo actual CPU, había un registro interno especial de 40 bits llamado acumulador. Una instrucción típica era sumar una palabra de la memoria al acumulador o almacenar este en la memoria.

Las computadoras son máquinas de arquitectura Von Neumann cuando:

* Tanto los programas como los datos se almacenan en una memoria en común. Esto hace posible la ejecución de comandos de la misma forma que los datos.
* Cada celda de memoria de la máquina se identifica con un número único, llamado dirección.
* Las diferentes partes de la información (los comandos y los datos) tienen diferentes modos de uso, pero la estructura no se representa en memoria de manera codificada.
* Cada programa se ejecuta de forma secuencial que, en el caso de que no haya instrucciones especiales, comienza con la primera instrucción. Para cambiar esta secuencia se utiliza el comando de control de transferencia.

Estructura clásica de las máquinas Von Neumann

Una máquina Von Neumann, al igual que prácticamente todos los computadores modernos de uso general, consta de cuatro componentes principales [12]:

1. Dispositivo de operación (DO), que ejecuta instrucciones de un conjunto especificado, llamado sistema (conjunto) de instrucciones, sobre porciones de información almacenada, separada de la memoria del dispositivo operativo (aunque en la arquitectura moderna el dispositivo operativo consume más memoria “generalmente del banco de registros”), en la que los operandos son almacenados directamente en el proceso de cálculo, en un tiempo relativamente corto.
2. Unidad de control (UC), que organiza la implementación consistente de algoritmos de decodificación de instrucciones que provienen de la memoria del dispositivo, responde a situaciones de emergencia y realiza funciones de dirección general de todos los nodos de computación. Por lo general, el DO y la UC conforman una estructura llamada CPU. Cabe señalar que el requisito es consistente, el orden de la memoria (el orden del cambio de dirección en el contador de programa) es fundamental a la hora de la ejecución de la instrucción. Por lo general, la arquitectura que no se adhiere a este principio no se considera Von Neumann.
3. Memoria del dispositivo: un conjunto de celdas con identificadores únicos (direcciones), que contienen instrucciones y datos.
4. Dispositivo de E/S (DES): permite la comunicación con el mundo exterior de los computadores, son otros dispositivos que reciben los resultados y que le transmiten la información al computador para su procesamiento [12].

Autómatas celulares

Surgen en la década de 1940 con John Von Neumann, que intentaba modelar una máquina que fuera capaz de autorreplicarse, llegando así a un modelo matemático de dicha maquina con reglas complicadas sobre una red rectangular. Inicialmente fueron interpretados como conjunto de células que crecían, se reproducían y morían a medida que pasaba el tiempo. Su nombre se debe a esta similitud con el crecimiento de las células [13].

Un autómata celular es un modelo matemático para un sistema dinámico compuesto por un conjunto de celdas o células que adquieren distintos estados o valores. Estos estados son alterados de un instante a otro en unidades de tiempo discreto, es decir, que se puede cuantificar con valores enteros a intervalos regulares. De esta manera este conjunto de células logra una evolución según una determinada expresión matemática, que es sensible a los estados de las células vecinas, y que se conoce como regla de transición local.

El aspecto que más caracteriza a los AC es su capacidad de lograr una serie de propiedades que surgen de la propia dinámica local a través del paso del tiempo y no desde un inicio, aplicándose a todo el sistema en general. Por lo tanto, no es fácil analizar las propiedades globales de un AC desde su comienzo, complejo por naturaleza, si no es por medio de una simulación, partiendo de un estado o configuración inicial de células y cambiando en cada instante los estados de todas ellas de forma síncrona.

Elementos de un Autómata Celular

* La definición de un AC requiere mencionar sus elementos básicos:
* Un espacio regular. Ya sea una línea, un plano de 2 dimensiones o un espacio n-dimensional. Cada división homogénea del espacio es llamada célula.
* Conjunto de Estados. Es finito y cada elemento o célula del espacio toma un valor de este conjunto de estados. También se denomina alfabeto. Puede ser expresado en valores o colores.
* Configuración Inicial. Es la asignación inicial de un estado a cada una de las células del espacio.
* Vecindades. Define el conjunto de células que se consideran adyacentes a una dada, así como la posición relativa respecto a ella. Cuando el espacio es uniforme, la vecindad de cada célula es isomorfa (es decir, que tiene el mismo aspecto)[13]

Máquina de de Von Neumann

Es un concepto teórico formulado por John Von Neumann. Es básicamente un autómata autorreplicante: se envía en una nave espacial y tiene capacidad de encontrar un lugar para asentarse, extraer materiales y energía de los recursos naturales y fabricar una copia de sí mismo. La primera máquina hará una copia, luego serán dos las máquinas trabajando, luego cuatro, ocho… Finalmente llegados al momento adecuado pueden construir más naves y saltar a explorar las siguientes estrellas y planetas. La expansión es exponencial y al final toda la galaxia queda a su alcance [14].

Algoritmo merge sort

Inventor, en 1945, del algoritmo de ordenación por mezcla, en la que la primera y la segunda mitades de una matriz se ordenan de forma recursiva cada uno y luego se fusionan [15].

Funcionamiento 1. Si la longitud de la lista es 0 ó 1, entonces ya está ordenada.

En otro caso:

* Dividir la lista desordenada en dos sublistas de aproximadamente la mitad del tamaño.
* Ordenar cada sublista recursivamente aplicando el ordenamiento por mezcla.
* Mezclar las dos sublistas en una sola lista ordenada.

El ordenamiento por mezcla incorpora dos ideas principales para mejorar su tiempo de ejecución:

* Una lista pequeña necesitará menos pasos para ordenarse que una lista grande.
* Se necesitan menos pasos para construir una lista ordenada a partir de dos listas también ordenadas, que a partir de dos listas desordenadas [15].

|  |
| --- |
| **Conclusiones** |

Las aportaciones que hicieron estos dos informáticos brindaron las bases para la programación, la seguridad informática y la inteligencia artificial, ya que muchos algoritmos, proyectos o computadores llevan parte del trabajo de estos dos personajes.

Los aportes de Turing sedimentaron la informática moderna, debido a todos sus trabajos realizados, además que gracias a su invento denominado “la bombe” logro reducir años de conflicto en el segundo mundial.

El tema más interesante es el test de Turing, varios autores estaban en contra de lo que decía acerca de que un computador “pensaba” o que era “inteligente”, es un gran debate, pero sin duda alguna en la época moderna se empezando a consolidar este test, debido al avance tecnológico en software y hardware.

El señor Jhon Von Neumann consolido las bases del cómputo, gracias a su arquitectura logro formar lo que es hoy en día los quipos de cómputo, detallando las partes físicas y lógicas de un equipo de cómputo.

Los autómatas celulares, son un tema bastante extenso por lo que requiere de un estudio mas a fondo, por todos sus componentes que lo conforman, sus características, es muy interesante, pero se deben de conocer distintos conceptos de autómatas y lenguajes formales, para comprender más su composición y su propósito.

El algoritmo de ordenamiento por mezcla a pesar de ser el primero y uno de los más reconocidos en el mundo de la informática, actualmente hay algoritmos que tienen un mayor rendimiento y sobre todo mayor eficacia.

Es importante conocer a estos personajes del mundo de las matemáticas y de la informática, porque han hecho aportes que usamos de manera inconsciente al momento de programar o realizar alguna otra actividad laboral o escolar.

Al momento de investigar hay muchos tecnicismos que son propios del mundo de la informática, por lo que ayuda a expandir aun mas el conocimiento y entendimiento del tema.

|  |
| --- |
| **Fuentes Consultadas** |

[1] McCandliss, B. D., Cohen, L., & Dehaene, S. (2003). The visual word form area: Expertise for reading in the fusiform gyrus. Trends in Cognitive Sciences, 7(7), 293–299. https://doi.org/10.1016/S1364-6613 (03)00134-7.

[2] Menabrea, L. F. (1989). Sketch of the analytical engine invented by Charles Babbage: With notes upon the memoir by the translator, Ada Augusta, Countess of Lovelace. In P. Morrison & E. Morrison (Eds.), Charles Babbage: On the principles and development of the calculator and other seminal writings (pp. 225–295). Mineola: Dover Publications. Menary, R. (2007). Cognitive integration: Mind and cognition unbounded. Basingstoke, New York: Palgrave Macmillan.

[3] Menary, R. (2010). Dimensions of mind. Phenomenology and the Cognitive Sciences, 9(4), 561–578. https://doi.org/10.1007/s11097-010-9186-7.

[4] Menary, R. (2012). Cognitive practices and cognitive character. Philosophical Explorations, 15(2), 147–164. https://doi.org/10.1080/ 13869795.2012.677851.

[6] Menary, R. (2013). The enculturated hand. In Z. Radman (Ed.), The hand, an organ of the mind: What the manual tells the mental (pp. 349–367).

[7] Cambridge, Mass.: MIT Press. Menary, R. (2014). Neural plasticity, neuronal recycling and niche construction. Mind & Language, 29(3), 286–303. https://doi.org/ 10.1111/mila.12051.

[9] Menary, R. (2015a). Mathematical cognition: A case of enculturation. In T. Metzinger & J. M. Windt (Eds.), Open MIND (pp. 1–20). Frankfurt am Main: MIND Group. DOI: 10.15502/9783958570818. Menary, R. (2015b). What? now: Predictive coding and enculturation: A reply to Regina E. Fabry. In T. Metzinger & J. M. Windt (Eds.), Open MIND (pp. 1–8). Frankfurt am Main: MIND Group. https://doi.org/ 10.15502/9783958571198. Merkley, R., & Ansari, D. (2016).

[10] Price, C. J., & Devlin, J. T. (2011). The Interactive Account of ventral occipitotemporal contributions to reading. Trends in Cognitive Sciences, 15(6), 246–253

[11] R. Niederberger, V. Alessandrini, The DEISA Project: Motivations, Strategies, Technologies, Proceedings of the International Supercomputer Conference ISC 2004 (2004), CDROM, O.Z.

[12] G.S. Bali, Th. D¨ussel, T. Lippert, H. Neff, Z. Prkacin, K. Schilling, String breaking with dynamical Wilson fermions, Nucl. Phys. B Proc. Suppl., 140 (2005), 609-611

[13] S. Birmanns, M. Boltes, H. Zilken, W. Wriggers, Adaptive Visuo-Haptic Rendering for Hybrid Modeling of Macromolecular Assemblies, Proceedings of the Mechatronics & Robotics Conference 2004, 13.-15.9.2004, Aachen, Germany, 1351-1356.

[14] D. Wolf, G. M¨unster, M. Kremer (eds.), NIC Symposium 2004, John von Neumann Institute for Computing, J¨ulich, Germany, NIC Series Vol. 20 (2004), ISBN 3-00-012372-5.

[15]G. Sutmann, B. Steffen, A particle-particle particle-multigrid algorithm for long range interactions in molecular systems, Comp. Phys. Comm., 169 (2005), 343-346.