



Compiladores e Intérpretes

Apuntes de clase

Introducción a la Historia de la Computación

Profesor Francisco Torres

Cesar Borge Mora

08/02/17

## Introducción a la Historia de la Computación:

La mejor manera de empezar a estudiar la historia de un área tan importante como la computación, es conocer a aquellos personajes que sentaron las bases para la tecnología con la que contamos hoy en día, marcando así, un antes y un después en la historia de la humanidad.

Es pertinente entonces, empezar por **John William Mauchly**. Quien fue un físico estadounidense que nació el 30 de agosto de 1907 y falleció el 8 de enero de 1980. Al terminar sus estudios se convirtió en profesor de Ingeniería Eléctrica en la Universidad de Pensilvania. Durante la Segunda Guerra Mundial, conoció a **John Adam Presper Eckert**. Este último fue un Ingeniero Eléctrico estadounidense. Nació el 9 de Abril de 1919 y falleció el 3 de Junio de 1995. Juntos son conocidos como pioneros de la computación, pues son los creadores de **ENIAC** (Electronic Numerical Integrator and Computer), la primera computadora de proposito general.

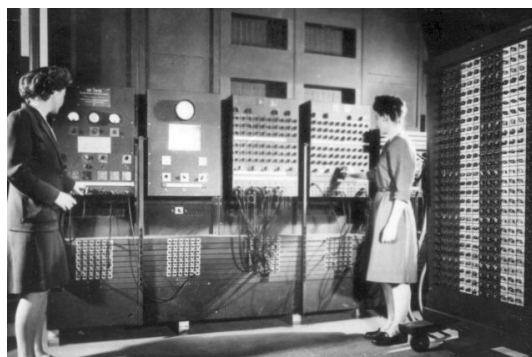


John William Mauchly



John Presper Eckert

ENIAC contaba con 17 468 tubos al vacío. Era decimal y ocupaba un espacio de 167m<sup>2</sup> con un peso de 27 toneladas. Tenía capacidad para resolver 5000 sumas y 300 multiplicaciones por segundo. Gracias a esto fue utilizada para calcular tablas de tiro de artillería para el Laboratorio de Investigación Balística del Ejército de los Estados Unidos.



ENIAC era programada por un equipo de 6 mujeres:

- Fran Bilas
- Betty Jennings
- Ruth Lichterman
- Kay McNulty
- Betty Snyder
- Marlyn Wescoff

A pesar de que estas mujeres jugaron un papel fundamental en el funcionamiento de ENIAC, no son muy reconocidas.

Dato Curioso:

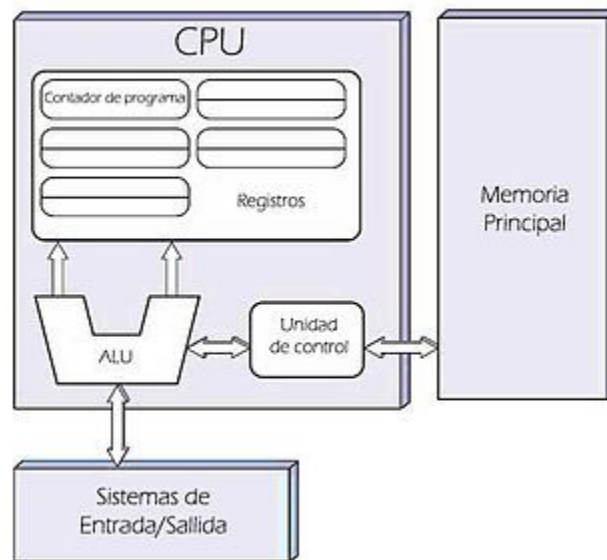
ENIAC fue financiada por el Ejército de los Estados Unidos en 1946.  
En aquel momento se invirtieron \$400,000.  
En el 2016, esa suma corresponde a \$4997053

Eckert y Mauchly sabían que ENIAC contaba con deficiencias, así que antes de que esta misma se empezará a usar, se pensó en un segundo ordenador llamado **EDVAC** (Electronic Discrete Variable Automatic Computer). A diferencia de ENIAC, EDVAC era una computadora binaria. Además gracias a su arquitectura, permitía almacenar programas en memoria.

Para el diseño de esta arquitectura se unió **John von Neumann**, un matemático húngaro - estadounidense que nació el 28 de Diciembre de 1903 y murió el 8 de febrero de 1957, considerado una de las personas más inteligentes y asombrosas de la historia, por sus múltiples aportes en diferentes áreas. Sin embargo, a parte de su inteligencia, también se le conoce por la mala jugada que le hizo a Eckert y Mauchly, ya que se dice que después de participar en reuniones en las que estos últimos explicaban sus ideas sobre esta nueva arquitectura, redactó un documento llamado "First Draft of a Report about the EDVAC" sin darle ningún crédito a Eckert y Mauchly.



¿Pero en qué consiste la famosa Arquitectura Von Neumann?



Como se aprecia en la imagen, la arquitectura contaba con una **memoria** que le permite guardar datos, así como el programa encargados de procesar dichos datos. Hoy en día esto es lo que conocemos como memoria RAM.

Además, la **Unidad de Control**, se encarga de mover los datos dentro y fuera de la memoria, mientras también se encarga de cada una de las instrucciones del programa.

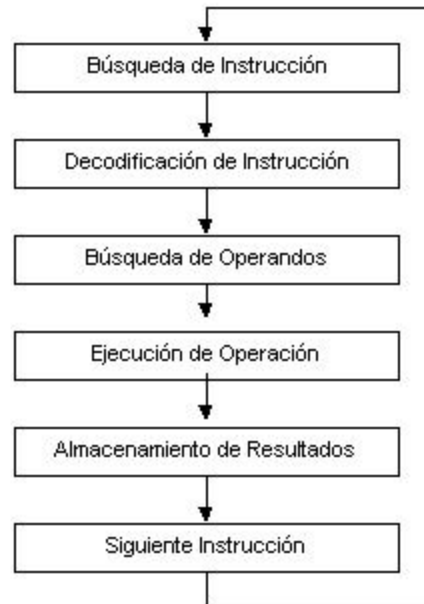
Los **registros** sirven para guardar valores intermedios de las instrucciones, o bien valores importantes, como el que se guarda en el **Contador de Programa**, cuyo valor representa la dirección de memoria de la siguiente instrucción por ejecutar.

La **ALU, o Unidad Aritmética Lógica**, es la encargada de realizar todos los cálculos necesarios con los datos, así como comparaciones de los mismos.

La idea es que los sistemas de **Entrada/Salida** sirvan como una interfaz entre las personas y la computadora, para el ingreso de parámetros y la salida de datos.

### Ciclo de Fetch

En resumen, este ciclo corresponde al proceso realizado por la Unidad de Control para traer una instrucción de memoria y ejecutarla. A continuación se muestra cada etapa de este ciclo:



Además, hay que tener en cuenta que en cada iteración se debe aumentar el Contador de Programa y revisar si hay interrupciones.

Material Extra para entender el ciclo de fetch:

<https://www.youtube.com/watch?v=QRyxgXAm5fc>

## Diseño de la Unidad de Control

Primero hay que tener clara la diferencia entre **Arquitectura** y **Organización**:

Arquitectura	Organización
Se refiere a: <ul style="list-style-type: none"> <li>• El lenguaje máquina que utiliza la UC.</li> <li>• Al conjunto de instrucciones disponibles.</li> </ul>	Se refiere a: <ul style="list-style-type: none"> <li>• Partes de la UC.</li> <li>• And, or, flip-flops</li> </ul>

Hay dos tipos de diseño de Unidad de Control:

El primero es la UC alambrada, que cuenta con un circuito físico correspondiente para cada instrucción. Gracias a esto es muy rápida y eficiente, pero también esto hace que no sea flexible a cambios, pues habría que rediseñar todo el circuito frente a un simple cambio.

Sin embargo, **Sir Maurice Vincent Wilkes**, quien fue un investigador de computación inglés, nacido en el 26 de junio de 1913 y que murió el 29 de noviembre de 2010, creó el concepto de UC Microprogramada. Es decir, que posee una arquitectura sobre la organización en la que cada instrucción de lenguaje máquina de la misma depende de una serie de microinstrucciones que se encuentran en una micromemoria.



Sir Maurice, intentando parecerse al Profesor Jorge Vargas

Cada microinstrucción consiste en una hilera binaria que representa señales eléctricas de la UC, estas señales a su vez controlan el comportamiento de la misma. A diferencia de las UC alambradas, su diseño era mucho más flexible pero no tan eficiente.

Aquí empezamos a observar cómo con cada avance o nivel hacia lo que conocemos como lenguajes de alto nivel, la sencillez a la hora de programar o diseñar va creciendo, aumentando también la productividad pero afectando la eficiencia.

La productividad se refiere a la cantidad de trabajo realizado o líneas de código programadas en una unidad de tiempo. Mientras que la eficiencia se refiere al aprovechamiento de los recursos por parte del hardware para trabajar de manera más rápida.

Nota:

Recuerden estudiar para el quiz del Miércoles y empezar desde ya el proyecto.