



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **Arqueología informática: el computador UNIVAC en los inicios de la computación moderna**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática

*Autor:* Diego García Garijo

*Tutor:* Xavier Molero Prieto

Curso 2015-2016



# Resumen

El UNIVAC I fue diseñado en la década de los años 50 del siglo XX por la empresa Remington Rand y es considerado el primer computador comercial desarrollado en Estados Unidos. Concebido por los pioneros informáticos John W. Mauchly y John P. Eckert, este computador estaba orientado a grandes empresas, tanto privadas como públicas, debido a su elevado coste y sus requerimientos de instalación. Con este computador, entre otras cosas, se elaboró el censo en los Estados Unidos del año 1951 y se predijo correctamente el ganador de las elecciones presidenciales en 1952. En ambas tareas el UNIVAC I, por aquel entonces una estrella mediática, demostró ser una máquina rápida y precisa.

En este trabajo se desarrolla un análisis del contexto histórico y social en el que se desarrolla el diseño e implementación de este computador. Así mismo, se estudia su arquitectura y programación con el fin de entender cómo se desarrollaba el proceso de programación en aquella fase inicial de la informática, al tiempo que se establecen comparaciones con la programación de un procesador moderno.

Dado el gran valor histórico del UNIVAC I, este trabajo incluye el diseño de una página web de carácter didáctico destinada al Museo de Informática de la Escuela Técnica Superior de Ingeniería Informática de la Universidad Politécnica de Valencia a fin de contribuir con su proyecto de difusión del patrimonio informático.

**Palabras clave:** arqueología informática, UNIVAC, programación, patrimonio digital, museo de informática

---

# Resum

L'UNIVAC I va ser dissenyat en la dècada dels anys 50 del segle XX per l'empresa Remington Rand i és considerat el primer computador comercial desenvolupat als Estats Units. Concebut pels pioners informàtics John W. Mauchly i John P. Eckert, aquest computador estava orientat a grans empreses, tant privades com públiques, a causa del seu elevat cost i els seus requeriments d'instal·lació. Amb aquest computador, entre altres coses, es va elaborar el cens als Estats Units de l'any 1951 i es va predir correctament el guanyador de les eleccions presidencials en 1952. En ambdós tasques l'UNIVAC I, en aquell moment una estrella mediàtica, va demostrar ser una màquina ràpida i precisa.

En aquest treball es du a terme una anàlisi del context històric i social en el què es desenvolupa el disseny i la implementació d'aquest computador. Així mateix, s'estudia la seua arquitectura i programació a fi d'entendre com era el procés de programació en aquella fase inicial de la informàtica, alhora que s'estableixen comparacions amb la programació d'un processador modern.

Donat el gran valor històric de l'UNIVAC I, aquest treball inclou el disseny d'una pàgina web de caràcter didàctic destinada al Museu d'Informàtica de l'Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València a fi de contribuir al seu projecte de difusió del patrimoni informàtic.

**Paraules clau:** arqueologia informàtica, UNIVAC, programació, patrimoni digital, museu d'informàtica

---

# Abstract

The UNIVAC I was designed in the late 50s of the twentieth century by the company Remington Rand and is considered the first commercial computer developed in the United States. Conceived by pioneers John W. Mauchly and John P. Eckert, this computer was aimed at large companies, both private and public, because of its high cost and installation requirements. With this computer, among other things, the census was developed in the United States in 1951 and correctly predicted the winner of presidential elections in 1952. In both tasks the UNIVAC I, at that time a media star, proved to be a machine quickly and accurately.

This paper presents an analysis of the historical and social context in which the design and implementation of this computer develops. Likewise, its architecture and programming is studied in order to understand how the programming process developed in this initial phase of computing, while comparisons are made with the programming of a modern processor.

Given the great historical value of the UNIVAC I, this work includes designing a website didactic designed the Museum of Informatics School of Engineering at the Polytechnic University of Valencia in order to contribute to the project diffusing computer heritage.

**Key words:** computer archeology, UNIVAC, programming, digital heritage, computer museum

---



# Índice general

---

<b>Índice general</b>	<b>V</b>
<b>Índice de figuras</b>	<b>VII</b>
<hr/>	
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Estructura de la memoria . . . . .	4
1.4 Notas sobre la bibliografía . . . . .	5
1.5 Agradecimientos . . . . .	8
<b>2 Contexto histórico</b>	<b>9</b>
2.1 Primera generación de computadores . . . . .	9
2.2 John Presper Eckert . . . . .	13
2.3 John William Mauchly . . . . .	16
2.4 EMCC y empresas sucesoras . . . . .	20
<b>3 Arquitectura interna</b>	<b>25</b>
3.1 Tipo de electrónica y memoria . . . . .	26
3.2 Cintas magnéticas . . . . .	29
3.3 Estética . . . . .	31
<b>4 Programación</b>	<b>37</b>
4.1 La memoria y la unidad aritmética . . . . .	37
4.2 La unidad de control . . . . .	39
4.3 Instrucciones . . . . .	40
<b>5 Programas</b>	<b>43</b>
5.1 Sumador simple . . . . .	43
5.2 Factorial . . . . .	47
5.3 Contador . . . . .	49
5.4 Programas en lenguaje de alto nivel . . . . .	52
5.5 Problemas encontrados programando . . . . .	57
<b>6 El UNIVAC y su repercusión social</b>	<b>59</b>
6.1 El miedo a las computadoras . . . . .	59
6.2 La publicidad del UNIVAC . . . . .	61
<b>7 Material didáctico</b>	<b>65</b>
7.1 Estructura de las páginas web . . . . .	67
<b>8 Conclusiones</b>	<b>69</b>
8.1 Trabajo futuro . . . . .	70
<b>Bibliografía</b>	<b>71</b>



# Índice de figuras

---

1.1	El UNIVAC, objeto de estudio del actual trabajo. . . . .	2
1.2	Símbolo de la UNESCO. . . . .	3
1.3	Libro: «Breve historia de la computación y sus pioneros». . . . .	6
1.4	Artículo: «Want to Buy a Brain?» . . . . .	7
1.5	Manual of operations . . . . .	7
1.6	Basic Programming: UNIVAC I . . . . .	8
2.1	El ENIAC en el sótano de la Universidad de Pensilvania . . . . .	10
2.2	Primeras imágenes del EDVAC en funcionamiento . . . . .	11
2.3	Cadena de montaje de la IMB 701 . . . . .	11
2.4	Z22 en la Universidad de Ciencias Aplicadas, Karlsruhe . . . . .	12
2.5	John Presper Eckert . . . . .	13
2.6	Eckert y Mauchly trabajando en el ENIAC . . . . .	14
2.7	Eckert y Mauchly trabajando en el UNIVAC . . . . .	15
2.8	John William Mauchly . . . . .	16
2.9	Mauchly y Eckert mejorando el ENIAC . . . . .	17
2.10	Grace Murray Hopper . . . . .	18
2.11	Atanasoff-Berry Computer (ABC) . . . . .	19
2.12	Eckert-Mauchly Computer Corporation (EMCC). . . . .	20
2.13	Departamento de software de la EMCC . . . . .	21
2.14	La BINAC recién instalada tras un largo viaje en barco . . . . .	22
2.15	Actual oficina de UNISYS . . . . .	23
3.1	Arquitectura de John von Neumann . . . . .	25
3.2	Tubos de vacío de primera generación . . . . .	26
3.3	El UNIVAC con los tubos de vacío que se emplearon . . . . .	27
3.4	Memoria del UNIVAC en el museo de Pensilvania . . . . .	28
3.5	Funcionamiento de la memoria línea de retardo . . . . .	28
3.6	Cinta magnética empleada en la época del UNIVAC . . . . .	29
3.7	Tipos de grabación de cintas magnéticas . . . . .	30
3.8	Convertor UNIVAC . . . . .	31
3.9	Esquema completo del UNIVAC . . . . .	32
3.10	Perspectiva delantera del UNIVAC . . . . .	32
3.11	Perspectiva lateral del UNIVAC . . . . .	33
3.12	Panel de mandos principal del UNIVAC . . . . .	33
3.13	Teclado del UNIVAC . . . . .	34
3.14	Impresora UNIVAC . . . . .	34
3.15	Herramientas de programación del UNIVAC . . . . .	35
4.1	63 caracteres válidos para la programación del UNIVAC . . . . .	37
4.2	Posiciones de memoria . . . . .	38

---

4.3	Registros de la unidad de control y su conexión . . . . .	39
4.4	Teclado con los botones de los registros . . . . .	41
4.5	Esquema de los registros en UNIVAC . . . . .	42
5.1	Código del sumador sencillo . . . . .	43
5.2	Resultado del sumador sencillo . . . . .	44
5.3	Debug del sumador sencillo . . . . .	45
5.4	Memoria tras la ejecución del sumador simple. . . . .	45
5.5	Código de la operación: $2A - B + 3(C + D)$ . . . . .	46
5.6	Resultado de la operación: $2A - B + 3(C + D)$ . . . . .	46
5.7	Código del programa factorial . . . . .	47
5.8	Resultado del programa factorial . . . . .	48
5.9	Debug del programa factorial . . . . .	49
5.10	Código del programa contador . . . . .	50
5.11	Resultado del programa contador . . . . .	51
5.12	Debug del programa contador . . . . .	51
5.13	Sumador simple en ensamblador . . . . .	52
5.14	Sumador complejo en ensamblador . . . . .	53
5.15	Factorial en ensamblador . . . . .	54
5.16	Contador en ensamblador . . . . .	55
5.17	Sumador hasta $n$ en ensamblador . . . . .	56
6.1	Elecciones a presidente de los Estados Unidos . . . . .	60
6.2	Harvard Mark I . . . . .	62
6.3	Ejemplo práctico sobre el UNIVAC y sus tubos de vacío . . . . .	63
7.1	Placa del museo y logotipo de la ICOM . . . . .	65
7.2	Página web en castellano . . . . .	66
7.3	Página web en valenciano . . . . .	67

---

---

# CAPÍTULO 1

## Introducción

---

Este trabajo fin de grado está enmarcado dentro del ámbito de la «arqueología informática» y el objeto de estudio de éste es el UNIVAC I. Esta máquina está considerada como la primera computadora comercial de la historia de los Estados Unidos de América. Para comenzar, en este capítulo se desarrolla una descripción tanto de la motivación de este trabajo fin de grado, como una enumeración de los objetivos que se pretenden conseguir con él.

### 1.1 Motivación

---

La arqueología informática [9] es el estudio del hardware, software, tecnología de comunicaciones y redes de equipos de cómputo y todo lo relacionado con estos tres elementos (creadores, época en que se desarrolló o construyó el equipo o software, razones de su surgimiento, ventajas y desventajas que ofrecían, entre otras cosas); con el propósito de investigar, analizar y entender los diseños y funcionamiento, disponibles o no hoy en día, de equipos de cómputo, software y redes de computadoras, con el fin de proveer la mayor cantidad de información al respecto, para reconstruir el conocimiento que encerraban los diseños, con dos propósitos fundamentales.

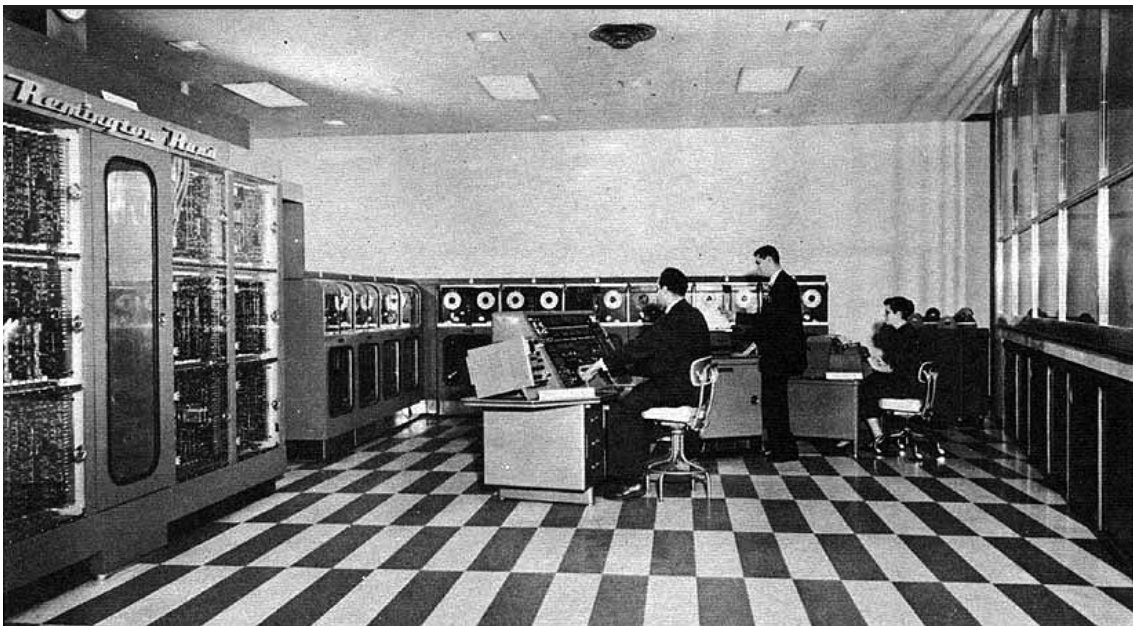
La arqueología informática, también llamada arqueología computacional, retroinformática o informática clásica. Esta naciente disciplina, que surgió inicialmente de actividades de coleccionismo y divulgación de ciertos grupos de usuarios aficionados, y puede dividirse en dos categorías según el ámbito en que se aplica. Por un lado está el ámbito más tecnológico, donde se desarrollan más a fondo las actividades de simulación y creación de software y hardware, mientras que por otro lado está el ámbito socio-cultural, donde podemos desarrollar actividades de divulgación, preservación, exposición, documentación, coleccionismo, congresos, asociacionismo y comercio.

Existe cierto criterio a la hora de llevar a cabo un estudio a cerca de un tema relacionado con la arqueología informática, los cuales podemos dividir en 4 importantes pasos:

1. Definir el objeto de estudio lo máximo posible antes de comenzar con su explicación teórica. Este puede ser un equipo antiguo y un software específico.

2. Buscar, recopilar, analizar y estructurar toda la información obtenida sobre el objeto estudiado: su historia, sus creadores, su diseño, su funcionamiento, su hardware y software, sus requisitos y sus características. En general, toda la información necesaria para su estudio, que puede provenir de diversas fuentes, como páginas webs, artículos antiguos o de actualidad, personajes relacionados o cualquier dispositivo o soporte que tenga alguna semejanza con el objeto de estudio.
3. Definir la interrelación y secuencia de los datos obtenidos de la información recabada sobre el diseño, construcción, estructura y/o funcionamiento del objeto de estudio en cuestión.
4. Redactar el documento en cuestión para su posterior uso como fuente de información de consulta, el cual podría servir para desarrollar en un futuro algún tipo de tecnología.

Los conocimientos que obtenemos de este tipo de narraciones pueden considerarse «patrimonio cultural», el cual en nuestro ámbito informático es llamado «patrimonio digital». Recordemos que el objeto de estudio del actual trabajo es el UNIVAC I, el cual podemos observar en la Figura 1.1.



**Figura 1.1:** El UNIVAC, objeto de estudio del actual trabajo.

Cuando hablamos de patrimonio, bien sea cultural y de cualquier otra índole, no podemos olvidar hablar de la UNESCO [8] (*Organización de las Naciones Unidas para la educación, la Ciencia y la Cultura*), la cual vela por la conservación de la educación, la cultura y la ciencia desde 1945. Podemos ver el símbolo de la UNESCO en la Figura 1.2.



Figura 1.2: Símbolo de la UNESCO.

La definición que nos da la UNESCO [8] a cerca del «patrimonio digital» es la siguiente: *El patrimonio digital consiste en recursos únicos que son fruto del saber o la expresión de los seres humanos. Comprende recursos de carácter cultural, educativo, científico o administrativo e información técnica, jurídica, médica y de otras clases, que se generan directamente en formato digital o se convierten a éste a partir de material analógico ya existente. Los productos “de origen digital” no existen en otro formato que el electrónico [16].*

Por todo lo comentado previamente tiene tanta importancia la arqueología informática. Es necesario conocer las dificultades, errores y trabas del pasado para poder avanzar en la dirección correcta en el futuro. Además, gracias a todo este tipo de información, toman más valor los desarrolladores de épocas antiguas debido a que fueron los precursores de lo que hoy en día conocemos, seguramente sin sus descubrimientos o avances (los cuales a día de hoy nos pueden parecer obvios) no tendríamos las tecnologías actuales.

## 1.2 Objetivos

---

Partiendo de un objeto de estudio, el cual, como hemos comentado anteriormente, es el UNIVAC I, en este trabajo se pretenden alcanzar los objetivos que se enumeran a continuación:

1. Se pretende situar al lector en el marco histórico en el que se creó el UNIVAC I, dándole a conocer la situación de aquella época, quiénes fueron sus creadores, cuales fueron sus problemas durante el desarrollo, dónde se desarrolló dicha máquina, cómo estaba el panorama empresarial, así como el desarrollo económico de la empresa que creó la máquina.
2. Posteriormente, se explica el tipo de arquitectura interna que empleaba el UNIVAC I, haciendo especial mención al tipo de electrónica que usaba (tubos de vacío) y su memoria interna, así como a las cintas magnéticas, las cuales fueron algo extraordinario para la época.

3. Se pretende hacer conocer al lector del tipo de programación que se empleaba en la época, así como cuales son las herramientas fundamentales que emplea el UNIVAC I internamente para interactuar con todos sus elementos entre sí y cuales eran las instrucciones más necesarias.
4. De la intención de que el lector sienta como se programaba en el marco temporal anteriormente nombrado, nace la necesidad de ejemplificar algunos programas, de modo que sirva para comprender las dificultades vividas en la época o incluso las que tendríamos en la actualidad.
5. Situaremos al lector en un marco más social, de modo que pueda percibir la situación que se vivió en aquella época de expansión informática y vea como repercutió el desarrollo de este tipo de tecnología en la sociedad.
6. Finalmente, y como parte de este trabajo y en colaboración con el Museo Informático, se ha desarrollado una página web con el fin de ayudar a la difusión cultural y situar esta gran máquina en el lugar de la historia que le corresponde.

### 1.3 Estructura de la memoria

---

Para la elaboración de este trabajo, se ha dividido el contenido de éste en ocho capítulos, los cuales abarcan todos los objetivos anteriormente mencionados. A continuación se explica a modo de resumen el contenido de cada uno de los ocho capítulos que hemos llevado a cabo:

1. Se realiza la presente introducción, la cual abarca la motivación por lo que hemos comenzado a llevar a cabo el actual trabajo, así como los objetivos que pretendemos alcanzar al finalizarlo, la presente estructura del contenido en su totalidad, unas aclaraciones sobre la bibliografía y los agradecimientos pertinentes a todas las personas que han colaborado para su posible finalización.
2. Situamos al lector en el contexto histórico en el que se desarrolló el UNIVAC I, haciéndolo conocer de sus predecesores en la primera generación de computadores, y se presentan a los dos creadores principales de dicha máquina: John Presper Eckert y John William Mauchly. Además, se hace especial mención a la empresa donde nació el UNIVAC I, la EMCC y las empresas tras las que pasó posteriormente.
3. Pasaremos a explicar como era la arquitectura interna del UNIVAC I, cuales fueron los elementos principales que se emplearon a nivel de electrónica, que tipo de memoria empleaba y como de potente era, así como sus dispositivos periféricos, haciendo especial mención al empleo de las cintas magnéticas y su conversor. Además echaremos un vistazo a como fue el diseño de dicho computador, el cual ocupaba una habitación entera.
4. Pasaremos a explicar como se programaba en aquella época, poniendo especial interés en las instrucciones más vitales para desarrollar los programas,



como pueden ser la suma, resta, multiplicación, división, carga en registros de datos en memoria, comparadores o saltos condicionales e incondicionales. Además se explicará como interactúan dichos registros entre sí para poder comprender el funcionamiento interno del UNIVAC I.

5. Tras explicar los aspectos previos de la programación, mostraremos unos ejemplos de modo que podamos facilitar la comprensión de toda la información narrada anteriormente. En nuestro caso, hemos elaborado código que lleva a cabo una serie de operaciones, tales como un sumador, calcular el factorial de un número o un contador. Además, haremos una comparativa con respecto a los lenguajes actuales de alto nivel, en nuestro caso en lenguaje ensamblador como es el procesador MIPS R2000, así como narraremos los problemas principales que hemos encontrado a la hora de programar dichos programas.
6. Posteriormente, conoceremos como fue la repercusión que tuvo este tipo de tecnología en aquella época y como se vendió el UNIVAC I. Veremos cuales fueron los miedos que nacieron en la sociedad y como se las ingeniaba la empresa creadora de la máquina para poder vender semejante computadora en una época difícil.
7. Se explica el material web didáctico que se ha creado para la difusión y conservación del patrimonio digital, así como se muestra la estructura de dicha página. En esta sección, se detalla su estructura y el tipo de información que aporta (en formato texto, imagen y audiovisual) para darle toda la difusión posible.
8. Exponemos las conclusiones que se alcanzan con este trabajo y que han surgido a partir de los objetivos descritos en el primer capítulo. También se hacen unas sugerencias de posibles futuros proyectos que han ido apareciendo en el transcurso de esta memoria.

## 1.4 Notas sobre la bibliografía

---

Tal y como podemos observar en la bibliografía del actual trabajo y en las notas a pie de página a lo largo del documento, se han empleado diversas fuentes de información de las cuales hemos extraído todo lo necesario para la elaboración de dicho trabajo. La mayoría de ellas se tratan de páginas web (como puede ser wikipedia o blogs privados), artículos antiguos y actuales, libros, manuales de los fabricantes de la máquina, etc...

Para la selección de la información válida y la que no, primero ha sido necesario contrastar y corroborar que la información en cuestión era correcta, de modo que no plasmáramos nada erróneo en el trabajo actual y podamos confiar en que todo lo mencionado aquí es cierto.

Existen infinidad de libros que hacen referencia al objeto de estudio, entre los cuales hemos empleado para la elaboración del actual trabajo algunos como son *Breve historia de la computación y sus pioneros* de Carlos A. Coello [2] o artículos como *Want to Buy a Brain?* [5]. Pero la mayor fuente de la cual hemos obtenido

multitud de información ha sido internet, y a la cual hacemos referencia en todo momento empleando las notas a pie de página en cada sección. Cabe destacar que el simulador empleado para la elaboración de programas fue encontrado por internet y sido esencial su uso para la demostración de numerosas pruebas. En la Figura 1.3 y 1.4 se muestran la portada del libro de Carlos Coello y el artículo anteriormente nombrado, respectivamente.



Figura 1.3: Libro: «Breve historia de la computación y sus pioneros».

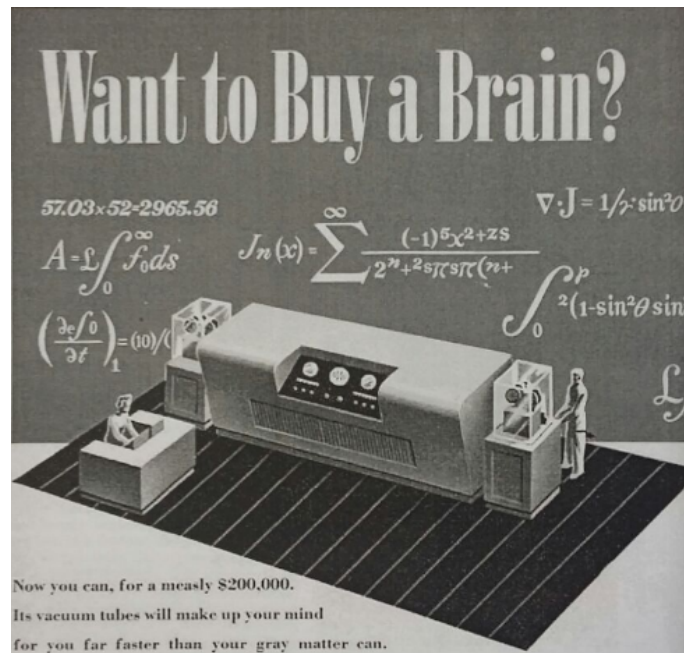


Figura 1.4: Artículo: «Want to Buy a Brain?»

Con respecto a los manuales empleados para la elaboración del código de los programas propuestos, ha sido imprescindible la ayuda de manuales encontrados a través de internet, tales como: *Manual of operations: The central computer of the UNIVAC System* [12] y *Basic Programming: UNIVAC I* [13]. En la Figura 1.5 y 1.6 se muestran la portada dichos manuales.

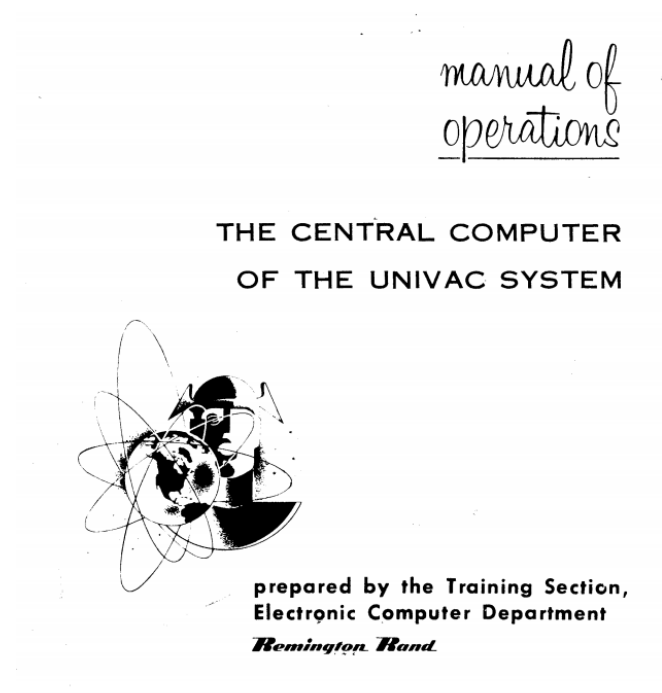


Figura 1.5: Manual of operations: The central computer of the UNIVAC System

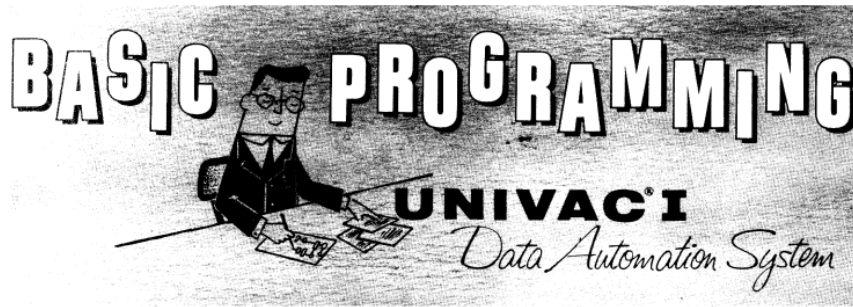


Figura 1.6: Basic Programming: UNIVAC I

Finalmente, ha sido de gran ayuda e interés el material audiovisual encontrado, como el vídeo de archivo *Remington-Rand Presents the Univac*<sup>1</sup>.

## 1.5 Agradecimientos

---

Agradecer en primer lugar a mi familia todo el apoyo mostrado durante la elaboración de este trabajo, ya que sin su incondicional apoyo no podría haber llegado hasta aquí, especialmente a mi madre y a mi padre por su paciencia y esfuerzo en mi educación. Mamá, papá, este proyecto es vuestro, ya que es el fruto de vuestro sacrificio.

No puedo olvidarme de mis dos amigos que me han acompañado durante estos largos años de carrera, Manu y Ana, a vuestro lado todo parece más fácil, gracias por lo que hemos compartido, y lo que queda por compartir.

Finalmente, el agradecimiento más sincero que soy capaz de mostrar va dirigido al director de este trabajo fin de grado, Xavier Molero Prieto, profesor del DISCA (Departamento de Informática de Sistemas y Computadores) y director del Museo de Informática, el cual ha sabido despertar en mí la devoción que él mismo siente por el ámbito de la arqueología informática, me ha guiado, tutelado, corregido y orientado en todo momento a través de este mundo.

---

<sup>1</sup><https://www.youtube.com/watch?v=j2fURxbdIZs>



---

# CAPÍTULO 2

## Contexto histórico

---

En este capítulo se detalla el momento en la historia de la informática en el que se creó el UNIVAC I, una detallada descripción de sus creadores más directos, como fueron John Presper Eckert y John William Mauchly, así como la repercusión que tuvo en la sociedad de antaño.

Además, se hace especial hincapié en la historia de la compañía y su posterior evolución a lo largo de los tiempos hasta el día de hoy.

Cabe destacar el mérito que tuvieron los creadores del UNIVAC I por llevar a cabo semejante proyecto, ya que hasta la fecha no había nada similar a lo que ellos comenzaron a desarrollar, por lo que no tenían ninguna referencia ni ejemplo en los que basarse para el diseño de estas máquinas, las cuales marcarían un antes y un después en la historia de la informática.

### 2.1 Primera generación de computadores

---

Para poder conocer la historia en su plenitud sobre el UNIVAC I, primero debemos ponernos en situación y conocer la realidad que se vivía en aquel entonces cuando fue creada [2]. El UNIVAC I pertenece a la primera generación de computadoras, que abarca desde el año 1951 hasta el año 1958, época en que la tecnología electrónica era a base de tubos de vacío, y la comunicación era en términos de nivel más bajo que puede existir, que se conoce como lenguaje de máquina.

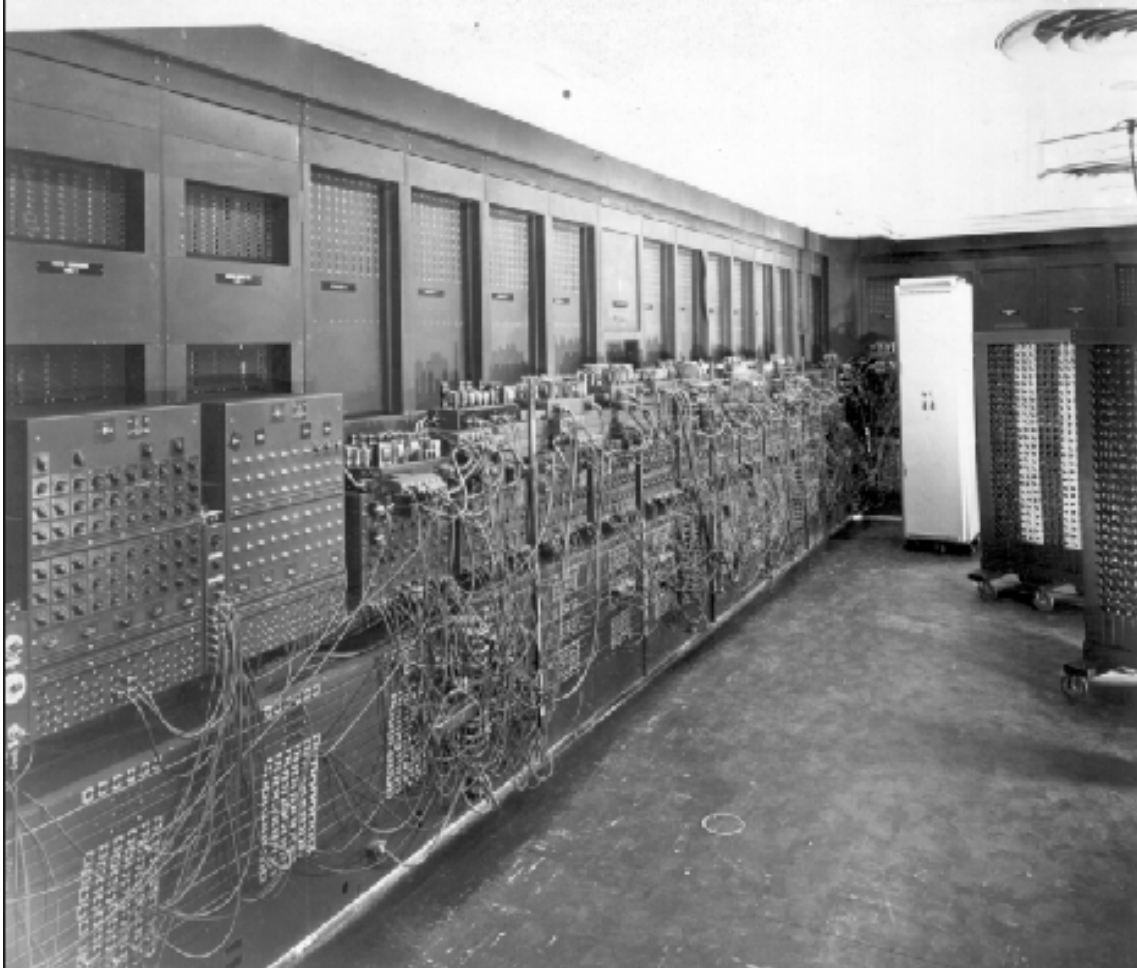
Dentro de esta primera generación, estaban los siguientes computadores:

**ENIAC (*Electronic Numerical Integrator And Computer*):** Fue la primera computadora digital de la historia <sup>1</sup>, pero no fue un modelo de producción, sino una máquina experimental. Tal y como observamos en la Figura 2.1, constaba de una máquina de proporciones gigantescas situada en un sótano de la Universidad de Pensilvania. Estaba construida básicamente con tubos de vacío, poseía cerca de 18 000 unidades. Además consumía mucha potencia eléctrica y pesaba algunas toneladas. Su creación fue encabezada por un grupo de científicos e ingenieros, todos ellos bajo el mando de J. Presper Eckert y John W. Mauchly (los cuales tam-

---

<sup>1</sup><https://es.wikipedia.org/wiki/ENIAC>

bién fueron los creados del UNIVAC) y era capaz de efectuar 5 000 sumas por segundo.



**Figura 2.1:** El ENIAC en el sótano de la Universidad de Pensilvania

**EDVAC (*Electronic Discrete Variable Automatic Computer*):** A diferencia del ENIAC, esta computadora era binaria, y tuvo el primer programa diseñado para ser almacenado, lo cual fue un referente para diseños posteriores, por ello el EDVAC es considerada un éxito en la historia de la informática <sup>2</sup>. Físicamente poseía 6 000 válvulas termoiónicas y 12 000 diodos. Cubría 45,5 metros cuadrados de superficie, tal y como vemos en la Figura 2.2, y pesaba cerca de 8 toneladas. Sus creadores, al igual que en el ENIAC, fueron J. Presper Eckert y John W. Machly, a los cuales se unió John von Neumann, y para hacer funcionar la máquina se requería la presencia de 30 personas cada turno, los cuales eran de 8 horas.

---

<sup>2</sup><https://es.wikipedia.org/wiki/EDVAC>

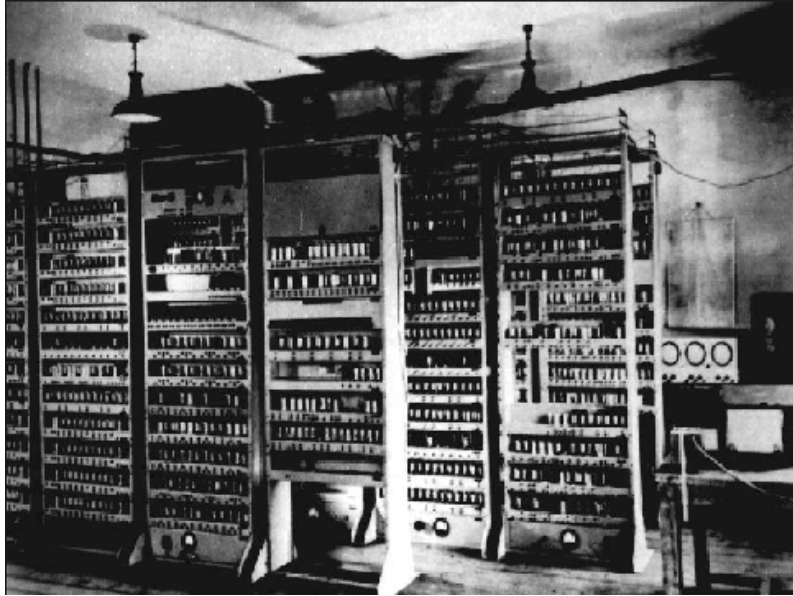


Figura 2.2: Primeras imágenes del EDVAC en funcionamiento

**IBM 701:** Fue conocida como “la calculadora de Defensa” mientras era desarrollada y durante cuatro años de producción, se vendieron veinte unidades. Se emplearon tubos Williams para la memoria, la cual tenía 72 con un diámetro de 76 mm con una capacidad de 1 024 bits cada uno, dando una memoria total de 2048 palabras de 36 bits cada uno. El tiempo de acceso a la memoria era de 12 microsegundos, y era necesario “refrescar” periódicamente la memoria de los tubos. En total se instalaron diecinueve sistemas IBM 701, y puede reclamar ser el primer computador que demostró el potencial de la inteligencia artificial con el juego de Damas de Arthur Samuel [3]. El IBM 701 fue la primera de una larga serie de computadoras de esta compañía. En la Figura 2.3 encontramos a los trabajadores de esta máquina junto a ella.



Figura 2.3: Cadena de montaje de la IBM 701

**Z22:** Fue el séptimo modelo de computador que desarrolló Konrad Zuse (los antecesores a este son el Z1, Z2, Z3, Z4, Z5 y Z11), tal y como observamos en la Figura 2.4. La diferencia más notable entre el Z11 y el Z22 fue el empleo de los tubos de vacío, a diferencia de los modelos anteriores que empleaban dispositivos electromecánicos. Fue la primera computadora con memoria de núcleos magnéticos, a pesar de que ya se trabajaba con ellos en los modelos militares. Físicamente posee 600 tubos trabajando en flip-flops, memoria basada en tarjetas perforadas y poseía un enfriamiento eléctrico, con una conexión a la red de agua corriente. Fue diseñada principalmente para la resolución de problemas matemáticos.



**Figura 2.4:** Z22 en la Universidad de Ciencias Aplicadas, Karlsruhe



## 2.2 John Presper Eckert

---

John Presper Eckert nació el 9 de Abril del año 1919 en Filadelfia, Estados Unidos, y está considerado uno de los pioneros en el ámbito de la informática y la electrónica. En la Figura 2.5 encontramos una fotografía de Eckert.



**Figura 2.5:** John Presper Eckert

John era el único hijo de una adinerada familia estadounidense y, desde su niñez, presentaba aptitudes excepcionales para las matemáticas, siendo capaz de resolver problemas con una facilidad innata para su edad. En 1937 se graduó en el colegio privado William Penn Charter School e ingresó en la Universidad de Pensilvania, en la cual se graduó como ingeniero eléctrico en 1941. Incluso antes de graduarse, John diseñó y construyó un dispositivo capaz de medir la concentración de vapor de naftaleno utilizando radiación ultravioleta <sup>3</sup>, además de trabajar en el desarrollo de instrumentos que median los límites de fatiga en los metales.

Tras graduarse, continuó trabajando en la universidad como profesor de cursos de electrónica y realizando estudios de postgrado. Fue entonces cuando conoció a uno de sus alumnos y mayores amigos hasta el fin de sus días, John William Mauchly, con el cual llevaría a cabo unos de los mayores hitos en la historia de la informática. Eckert se interesó rápidamente por las ideas de Mauchly sobre la construcción de un computador, y en 1943 comenzaron a trabajar en el

---

<sup>3</sup><http://www.buscabiografias.com/biografia/verDetalle/8675/John%20Presper%20Eckert%20Jr>.

ENIAC. En la Figura 2.6 encontramos a Eckert y Mauchly estudiando el diseño de el ENIAC.



**Figura 2.6:** Eckert y Mauchly trabajando en el ENIAC

La tarea principal de Eckert era el diseño de los circuitos electrónicos, y, tras terminar la computadora en 1946, ésta poseía 500 000 conexiones soldadas y utilizaba tarjetas perforadas para el almacenamiento de datos. Llegó a pesar 30 toneladas y, a pesar de que un chip actual de silicio posee mayor potencia de cálculo, en su época fue mil veces más rápida que las calculadoras de hoy en día. El propósito de la construcción de esta máquina era el empleo de la misma para el cálculo armamentístico en la Segunda Guerra Mundial, pero ésta ya había terminado una vez construida la máquina, por lo que se le dio un nuevo uso: el desarrollo de armas nucleares. Esto tuvo que llevarse en secreto, debido a que era ilegal llevar a cabo este tipo de estudios.

Tras acabar el ENIAC, ambos ingenieros dejaron la universidad y fundaron su propia empresa: Eckert-Mauchly Computer Corporation. En dicha empresa comenzaron a trabajar en la elaboración de un computador cuya función sería desarrollar un misil secreto de largo alcance llamado Snark, encargado por la empresa Northrop Aircraft Company. Dicho computador sería entregado en 1949 con el nombre de BINAC (*Binary Automatic Computer*), y una de sus características más innovadoras fue que estaba formado por dos procesadores que podrían realizar 3 500 sumas o restas por segundo y 1 000 multiplicaciones o divisiones por segundo. Además fue la primera computadora en utilizar cintas magnéticas como memoria secundaria, por lo que tuvieron que desarrollar un dispositivo que convirtiera las lecturas/escrituras en las cintas, el cual se empleó posteriormente en el UNIVAC, tal y como se puede apreciar en la Figura 2.7.



**Figura 2.7:** Eckert y Mauchly trabajando en el UNIVAC

Debido a diversos problemas que posteriormente comentaremos, la empresa tuvo que ser absorbida, y Eckert trabajó como ejecutivo de la misma hasta el fin de sus días.

Durante su vida, Eckert no solo trabajó en lo anteriormente nombrado, tuvo miles de proyectos los cuales le dieron gran fama y reconocimiento en su ámbito. Algunos de sus mayores logros son los siguientes <sup>4</sup>:

- John Presper Eckert patentó más de 85 inventos, mayormente electrónicos.

<sup>4</sup>[http://www.ecured.cu/John\\_Presper\\_Eckert](http://www.ecured.cu/John_Presper_Eckert)

- Recibió premios por ser pionero en el mundo de las computadoras.
- Recibió la Medalla Nacional de Ciencias, uno de los más prestigiosos de Estados Unidos.
- Fue elegido miembro de la Academia Nacional de Ingeniería en 1967.

John Presper Eckert murió el 3 de Junio de 1995 en Bryn Mawr, Pensilvania, Estados Unidos, a causa de una enfermedad contra la cual llevaba mucho tiempo luchando, la leucemia.

## 2.3 John William Mauchly

---

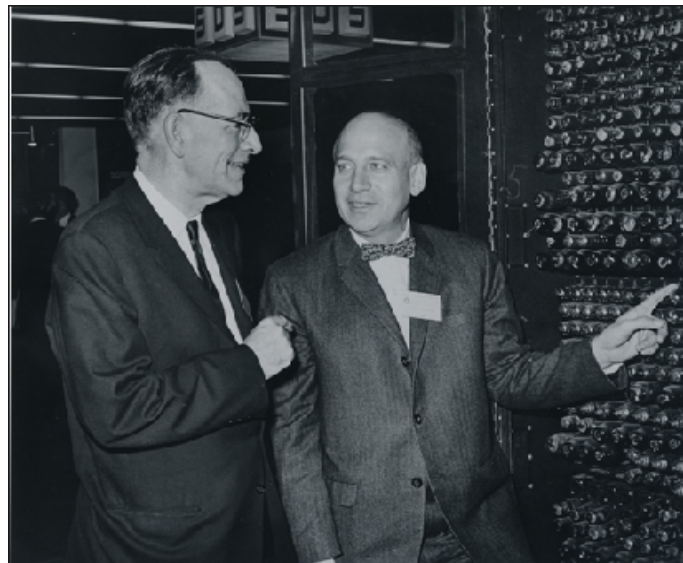
John William Mauchly nació el 30 de Agosto de 1907 en Cincinnati, Ohio, a pesar de que creció en Chevy Chase, Maryland, lugar donde su padre trabajaba como físico en el Carnegie Institute of Washington DC. Allí ganó la beca de Ingeniería del Estado de Maryland, lo que le dio acceso a la Johns Hopkins University en el otoño de 1925. En 1927 entró a un programa de doctorado y fue trasladado al programa de graduado en física de la universidad, donde lo completó en 1932 y se convirtió en profesor de física en el Ursinus College, en Filadelfia, en el cual ejercería su profesión del 1933 al 1941. En la Figura 2.8 encontramos una fotografía de Mauchly.



Figura 2.8: John William Mauchly

Tras esto, Mauchly hizo un curso de electrónica militar en la Moore School of Engineering <sup>5</sup>, en la Universidad de Pensilvania, donde conoció a John Presper Eckert, un amigo que mantendría el resto de su vida. Allí aceptó una plaza como profesor, y comenzó a interesarse por las ideas de Eckert, el cual mantenía que con la ingeniería apropiada los tubos de vacío podían servir de una manera muy fiable. Los estudios de Mauchly mientras estuvo en esta Universidad se basaron en el cálculo de tablas de tiro para el gran número de armas nuevas que la armada de los Estados Unidos estaba desarrollando para sus objetivos militares. Por ello, a principios de los 40, Mauchly escribió un libro titulado: «Utilización de Tubos de Vacío de Alta Velocidad para realizar Cálculos», gracias a la ayuda que le había proporcionado John Presper Eckert.

Como hemos comentado en la sección anterior, tanto Mauchly como Eckert comenzaron a trabajar juntos en el diseño del ENIAC <sup>6</sup>, como podemos observar en la Figura 2.9, pero tuvo que ser congelado en 1944 para comenzar su construcción. Ambos eran conscientes de las limitaciones de la máquina, por lo que comenzaron a diseñar un segundo ordenador: el EDVAC <sup>7</sup>. En enero de 1945 consiguieron un contrato para construir el ordenador, en el cual se podían almacenar programas gracias a la propuesta de Eckert de emplear una memoria de mercurio para guardar datos. Más tarde, el matemático John von Neumann se unió en algunas discusiones del proyecto, redactando un documento interno que describía el EDVAC titulado *First Draft of a Report about the EDVAC* en 1945.



**Figura 2.9:** Mauchly y Eckert estudiando las mejoras del ENIAC y su integración en el EDVAC

En marzo de 1946, nada más anunciarse el ENIAC, la Moore School decidió cambiar su política de patentes, y de ese modo poder ganar derechos comerciales por cualquier desarrollo futuro de ordenadores. Obviamente, a Eckert y Mauchly

<sup>5</sup><http://documentacion.nexun.org/mediawiki/index.php/UNIVAC>

<sup>6</sup>[www.dacostabalboa.com/es/quienes-son-john-william-mauchly-y-john-presper-eckert/](http://www.dacostabalboa.com/es/quienes-son-john-william-mauchly-y-john-presper-eckert/)  
4347

<sup>7</sup>[https://es.wikipedia.org/wiki/John\\_William\\_Mauchly](https://es.wikipedia.org/wiki/John_William_Mauchly)



les pareció inaceptable esta nueva premisa y dimitieron, pero no antes sin dar una serie de charlas en Moore School sobre diseño de ordenadores.

Tras abandonar la Moore School, Mauchly comienza a darle real importancia a las aplicaciones en ordenadores, así como a su arquitectura y organización. La experiencia que había tomado programando el ENIAC y sus sucesoras lo llevaron a crear el SHORT CODE, primer lenguaje de programación y usado actualmente en un ordenador. Era un intérprete de pseudo-código para problemas matemáticos propuesto en 1949 que funcionó sobre el UNIVAC I y el II. Debido a la importancia de los lenguajes, se vio obligado a contratar a Grace Murray Hopper, a la cual podemos encontrar en la Figura 2.10 trabajando para desarrollar un compilador para el UNIVAC.



**Figura 2.10:** Grace Murray Hopper creando el compilador del UNIVAC y a su equipo de trabajo

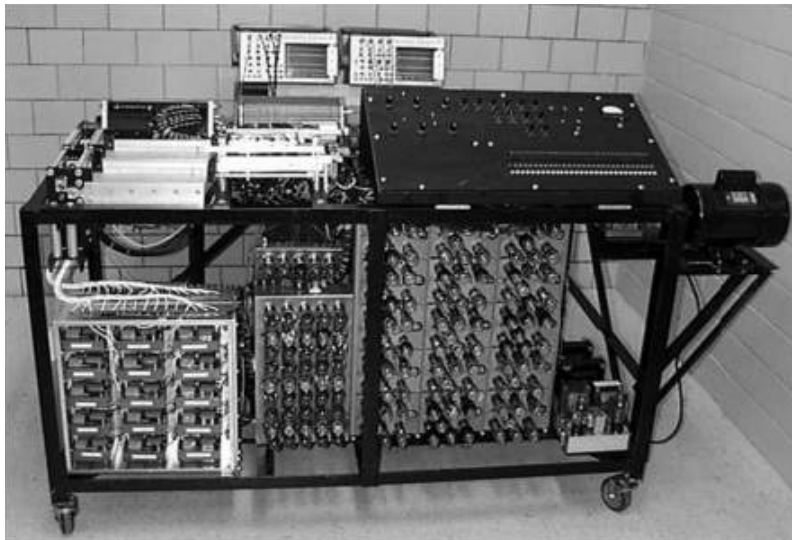
Mauchly continuó involucrado en ordenadores durante el resto de su vida. Fundó y fue presidente de la *Association for Computing Machinery* (ACM) y también ayudó a fundar la *Society of Industrial and Applied Mathematics* (SIAM). Más tarde, la Eckert-Mauchly Computer Corporation fue absorbida por Remington Rand en el 1950 y durante los diez años posteriores, el doctor Mauchly quedó como director de desarrollo de aplicaciones de el UNIVAC. En 1959 abandonó su puesto para formar la consultoría Mauchly-Associated hasta el 1967, que fundó Dynatrend, una organización consultora de ordenadores. Finalmente en 1973 se convirtió en asesor de Sperry Univac.

Al igual que su compañero y amigo, Mauchly no solo trabajó en lo anteriormente mencionado, sino que tuvo multitud de proyectos, los cuales acarrearón merecidos premios y menciones. Algunos de ellos son los siguientes <sup>8</sup>:

<sup>8</sup>[http://www.ecured.cu/John\\_William\\_Mauchly](http://www.ecured.cu/John_William_Mauchly)

- Era miembro vitalicio del Franklin Institute, la National Academy of Engineering y la Society for Advancement of Management.
- Fue escogido Socio de la IRE.
- Fue socio del American Statistical Association.
- Fue el primero en emplear el verbo programar en su artículo de 1942 sobre ordenadores electrónicos, a pesar de que fue en el contexto del ENIAC y no la programación como la conocemos hoy en día.
- A pesar que sus contribuciones a la industria de las computadoras eran relativamente desconocidas, dibujó el mapa para el desarrollo de la industria de la computación como la conocemos hoy en día.

Hubo cierto grado de polémica en referencia a las patentes del ENIAC y la *Atanasoff-Berry Computer (ABC)*, la cual se muestra en la Figura 2.11.



**Figura 2.11:** Atanasoff-Berry Computer (ABC)

Los defensores de Mauchly afirman que, mientras la corte dijo que el ABC fue el primer ordenador electrónico digital, no definió el término ordenador, ya que originalmente fue referido a la persona que computa, pero adaptado para ser aplicado a una máquina. Los opositores a la corte también dicen que no hay nada en común a las dos máquinas a nivel de componente.

- El ABC era binaria, mientras que el ENIAC era decimal.
- El ABC usaba memoria regenerativa cilíndrica, mientras que el ENIAC usaba contadores electrónicos.
- El ABC usaba sus tubos para implementar un sumador serial binario, mientras que el ENIAC usaba los tubos para implementar un kit completo de operaciones decimales.

Otro aspecto criticado fuertemente para el ABC es que nunca fue completamente funcional. Su mecanismo de entrada y salida no era lo suficientemente fiable para acabar un problema sin error, además de que la máquina usaba un reloj de 60 Hz, limitándose a treinta operaciones por segundo. Pero todos estos datos carecían de importancia a la hora de patentar el ABC o el ENIAC.

Mauchly falleció el 8 de enero de 1980, en Abington, Pensilvania, durante una cirugía para corregir una dolencia cardíaca.

## 2.4 EMCC y empresas sucesoras

La EMCC (*Eckert-Mauchly Computer Corporation*) fue fundada por John Presper Eckert y John William Mauchly el 22 de Diciembre de 1947<sup>9</sup>, ya que sus creadores pretendían seguir investigando y fabricando nuevos diseños de computadoras para uso comercial y militar tras haber creado el ENIAC. En un principio dicha empresa se llamó Electronic Control Company, pero una vez se constituyó como empresa, el nombre fue cambiado. En la Figura 2.12 observamos la formalización de la empresa.

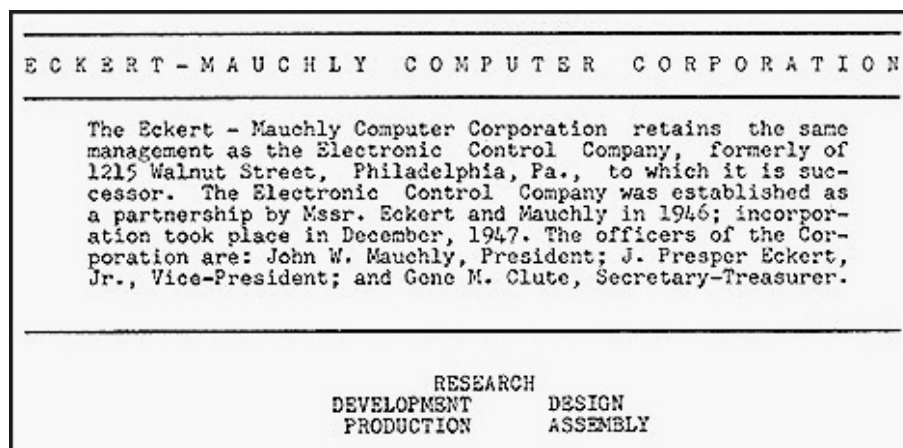


Figura 2.12: Eckert-Mauchly Computer Corporation (EMCC).

La fundación de esta empresa nace de la necesidad de Eckert y Mauchly, ya que ambos estaban trabajando en el EDVAC cuando las nuevas políticas les obligaban a ceder los derechos de sus trabajos, cosa que disgustó enormemente a ambos creadores. Antes de fundar su nueva empresa, estuvieron planteándose incorporarse a IBM y al equipo de John von Neumann del Institute for Advanced Study en Princeton, pero tras descartar esa opción, fundaron la EMCC.

Uno de los primeros proyectos que llevaron a cabo en su nueva empresa fue el EDVAC II, la cual posteriormente acabaría llamándose UNIVAC, que sería encargada por los Estados Unidos para realizar el censo. Eckert tuvo que contratar personal, incluyendo ingenieros de confianza de la Moore School y la empresa lanzó un ambicioso programa para diseñar y fabricar computadoras a gran escala. Entre los mayores logros que consiguió la empresa se encuentra el uso de la

<sup>9</sup><http://www24.helpep.eu/01196898/EckertMauchlyComputerCorporation>



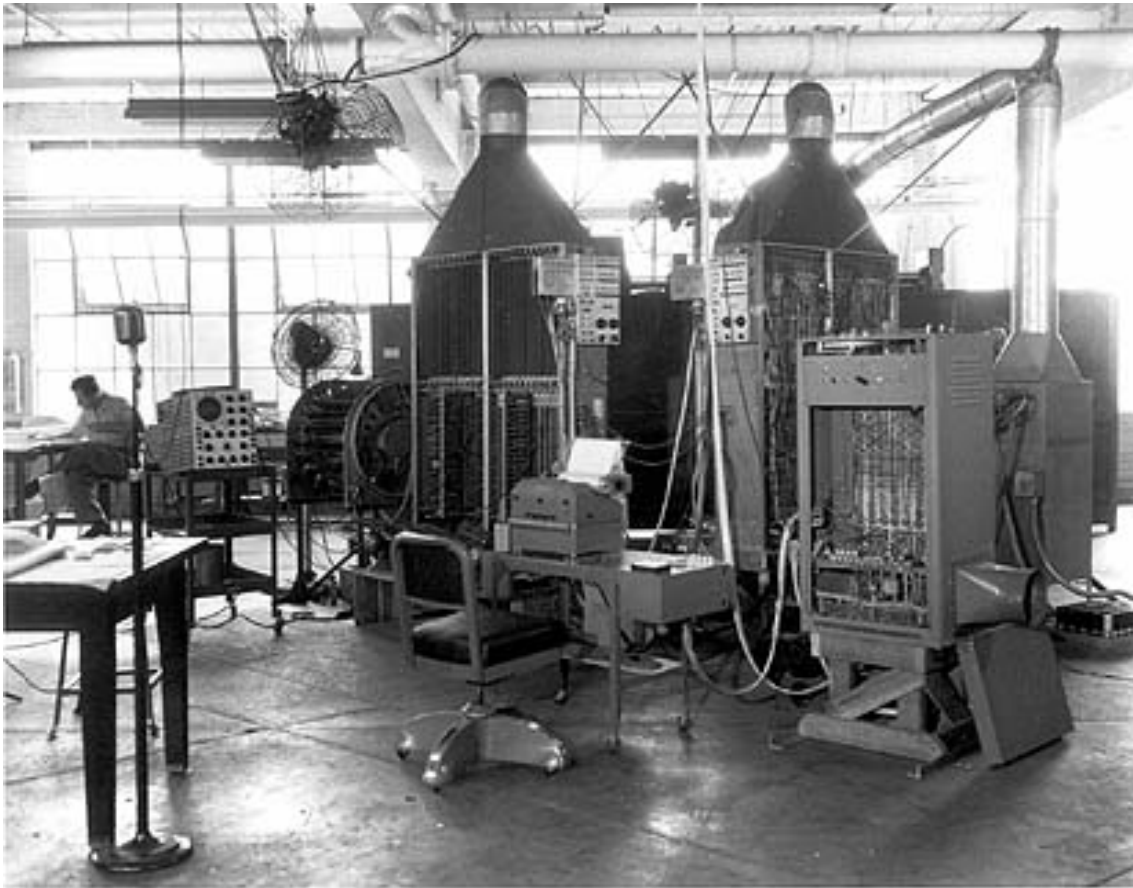
cinta magnética para el almacenamiento a gran velocidad y el primer compilador de lenguaje Shortcode (creado por Mauchly) del mundo. Tal era su éxito que Mauchly se vio obligado a contratar más trabajadores y formar así un departamento de software, tal y como podemos observar en la Figura 2.13.



Figura 2.13: Departamento de software de la EMCC

El UNIVAC tuvo más retrasos de los previstos y la economía de la empresa no pasaba por su mejor momento, por lo que tuvieron que llevar a cabo un proyecto que pudieran realizar rápidamente para poder mantener la economía. Por ello comenzaron a trabajar en el BINAC (*Binary Automatic Computer*) encargada por la compañía Northrop. Los presupuestos estimados para su construcción se quedaron muy lejos de la realidad, por lo que la EMCC quedó sin dinero. La empresa pudo continuar gracias a la inversión de Harry L. Straus, vicepresidente de la American Totalisator Company, el cual vio potencial en la empresa, y pensó que podría haber alguna aplicación en el negocio de las pistas de carreras e invirtió 500 000 dólares en la organización. Debido a su generosa inversión, Straus se convirtió en presidente de la junta directiva de la EMCC y la American Totalisator recibía un porcentaje de las existencias. Lamentablemente, Straus falleció en un accidente de avión en 1 949 y la directiva de la American Totalisator retiró el apoyo a la EMCC.

El BINAC fue entregado en 1 949, pero Northrop se quejó de que nunca llegó a funcionar correctamente. Los ingenieros de la EMCC alegaban que, tras el transporte en barco de la máquina, no les dejaron instalarla correctamente, y que habían tardado demasiado en hacerlo ellos mismos, habiendo estado parada mucho tiempo en un almacén. En la Figura 2.14 podemos apreciar el BINAC instalado tras su entrega.



**Figura 2.14:** La BINAC recién instalada tras un largo viaje en barco

Por si la empresa no llevara suficiente retraso de por sí, las acusaciones comunistas durante el gobierno de McCarthy no favorecieron a la empresa. El UNIVAC no iba a estar a tiempo, y además a Mauchly se le impidió trabajar en la empresa debido a que se le acusó de contratar a ingenieros comunistas para la misma. Mauchly intentó recurrir estas acusaciones para poder seguir trabajando en el UNIVAC lo antes posible, pero esto no serviría de nada hasta pasado dos años, por lo que John Mauchly tuvo que terminar por su cuenta el UNIVAC en su propia casa gracias a la ayuda de Max Woodbury, un estadístico de la Universidad de Pensilvania.

Los problemas continuaban apareciendo, y los retrasos impedían a la empresa poder cobrar por sus diseños, por lo que se vieron obligados a poner la EMCC en venta en 1950. Entre sus primeros compradores aparecieron la NCR y Remington Rand. Éste último hizo la primera oferta a EMCC y se convirtió en un departamento UNIVAC de la Remington Rand. No obstante, era demasiado tarde para poder efectuar el censo tal y como les había pedido, por lo que se le asignó la tarea de usar el UNIVAC como conversor de tarjetas perforadas a cinta.

Mauchly renunció en 1952, pero debido a su contrato se mantuvo hasta el año 1960, prohibiéndole trabajar en sus proyectos informáticos durante ese periodo de tiempo. En 1955 Remington Rand se fusionó con Sperry Corporation, dando un nuevo nombre a la compañía: Sperry UNIVAC. La empresa descendiente de ella que opera actualmente es UNISYS, la cual ofrece servicios de outsourcing, inte-

gración de sistemas, servicios de consultoría, servicios de infraestructura, servicios de mantenimiento y tecnología de servidor de alto nivel <sup>10</sup>. En la Figura 2.15 podemos apreciar la oficina actual de UNISYS.



**Figura 2.15:** Actual oficina de UNISYS

---

<sup>10</sup><https://es.wikipedia.org/wiki/Unisys>



---

## CAPÍTULO 3

# Arquitectura interna

---

En este capítulo se describen los aspectos relacionados con la arquitectura interna del UNIVAC I. Entraremos en profundidad en el tipo de electrónica empleada en su construcción, así como el tipo de memoria utilizada, las cintas magnéticas que se usaron y la estética final que presentaba esta computadora pionera en la historia de la informática.

Antes de comenzar con el capítulo, cabe destacar que esta máquina empleaba la arquitectura marcada por John von Neumann, la cual podemos apreciar en la Figura 3.1, en el modelo que lleva su nombre y que fue publicado el 30 de junio de 1945 por la Universidad de Pensilvania y la armada de los Estados Unidos en un borrador llamado «*First Draft of a report to the EDVAC*»

El modelo describe una arquitectura compuesta por una unidad de control conectada a una memoria, y esta memoria conectada a su vez con un dispositivo de entrada, otro de salida y una unidad aritmético-lógica, la cual se encarga de llevar a cabo las operaciones que van siendo almacenadas en la memoria recibidas por el dispositivo de entrada. Tanto el dispositivo de entrada como el de salida estaba constituido por cintas magnéticas.

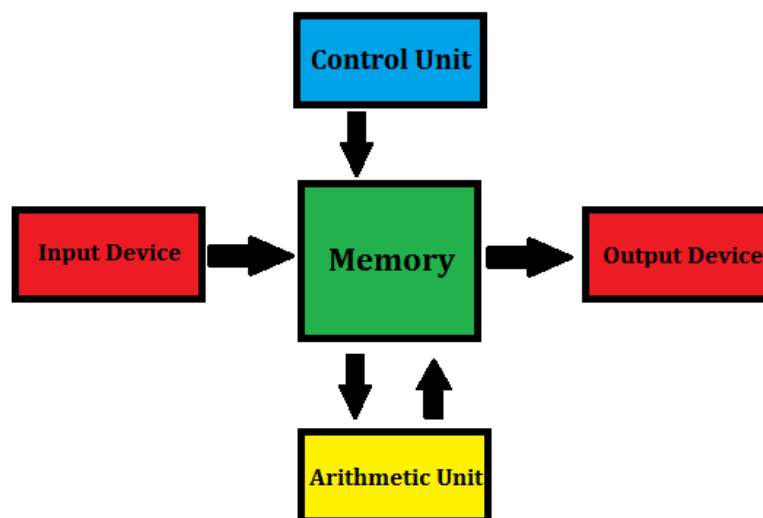


Figura 3.1: Arquitectura de John von Neumann

### 3.1 Tipo de electrónica y memoria

El elemento principal del UNIVAC, en cuanto a electrónica se refiere, fueron los tubos de vacío, también llamados válvula electrónica, válvula termoiónica, válvula de vacío o bulbo. La utilidad principal de este elemento es su capacidad para amplificar, conmutar o modificar una señal eléctrica mediante el control del movimiento de los electrones en un espacio, relativamente vacío, a muy baja presión o en presencia de gases específicamente seleccionados <sup>1</sup>. En la Figura 3.2 podemos apreciar este tipo de electrónica.

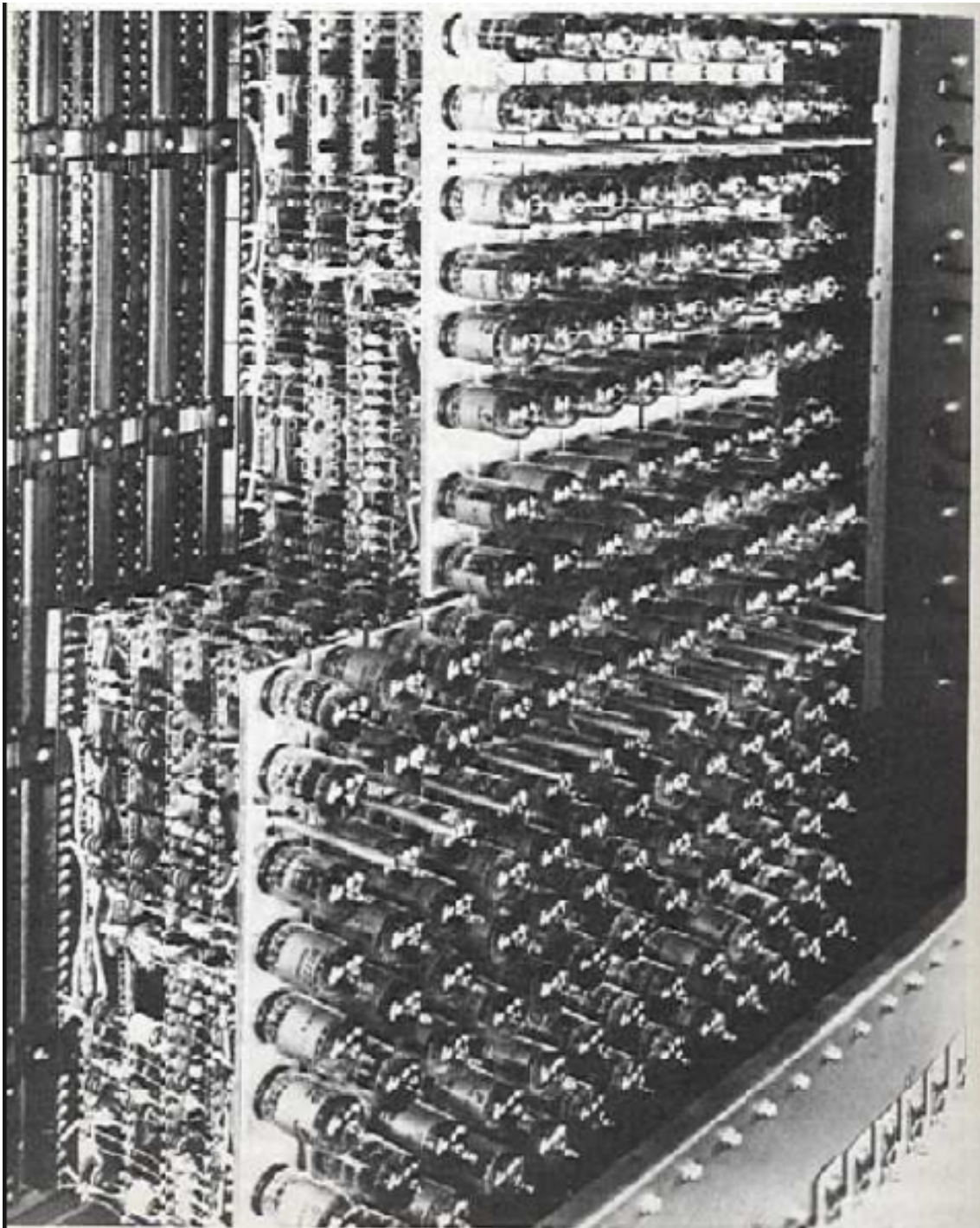


Figura 3.2: Tubos de vacío de primera generación

Concretamente el UNIVAC empleaba 5 mil tubos de vacío, cosa bastante novedosa ya que contrastaba a los casi 18 mil que usaba el legendario ENIAC, lo que le permitía que su tamaño fuera más reducido que la anteriormente nombrada. Sus dimensiones eran 3,00 x 4,20 x 2,70 metros, lo que la hacía relativamente pequeña para la época comparada con sus competidores en aquellos días. En la Figura 3.3 podemos observar los tubos de vacío que se empleaban en el UNIVAC

<sup>1</sup>[https://es.wikipedia.org/wiki/V%C3%A1lvula\\_termoi%C3%B3nica](https://es.wikipedia.org/wiki/V%C3%A1lvula_termoi%C3%B3nica)





**Figura 3.3:** El UNIVAC con los tubos de vacío que se emplearon

A día de hoy, este tipo de tecnología se sigue empleando en el ámbito musical, como amplificadores de micrófonos, guitarras y bajos, así como en equipos de sonido de alta fidelidad. También es la base de del funcionamiento del horno microondas, así como de ciertas televisiones y radares.

La memoria empleada en el UNIVAC fue de tipo «línea de retardo», la cual era capaz de almacenar datos aprovechando el tiempo que necesita una señal para propagarse por un medio físico. Su composición estaba básicamente constituida por un tubo relleno de mercurio con un transductor piezoeléctrico en cada

extremo. En la Figura 3.4 podemos apreciar la auténtica memoria que se empleó en el UNIVAC en su día.

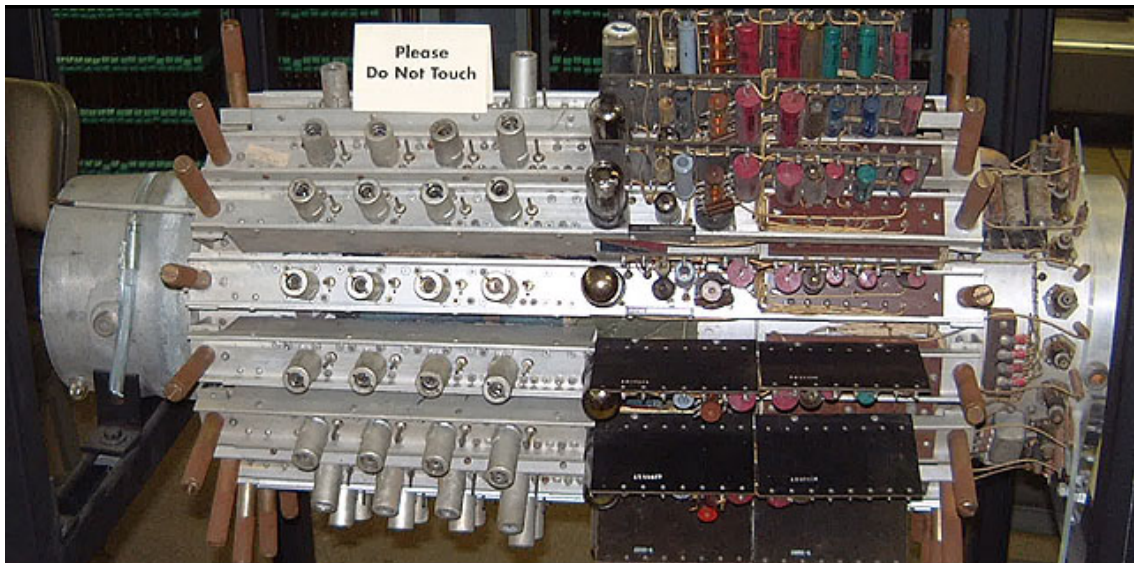


Figura 3.4: Memoria del UNIVAC en el museo de Pensilvania

Como se representa en la Figura 3.5, el funcionamiento de este tipo de memoria se basa fundamentalmente en el retardo que le supone a una onda acústica en transmitirse a lo largo de un medio, de modo que los datos se introducen en dicho medio a través de un transductor y, tras un retardo, esta señal se encuentra al final del alambre recogida por el otro transductor y retroalimentada formando un bucle cerrado y así mantiene, refresca o memoriza la información introducida <sup>2</sup>.

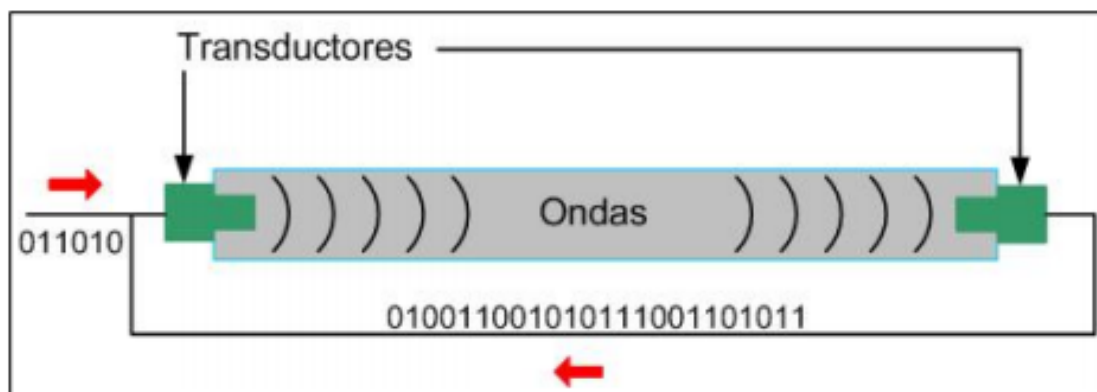


Figura 3.5: Funcionamiento de la memoria línea de retardo

Cabe destacar que este tipo de memorias fueron mejoradas por Eckert en la década de los cuarenta, pero a pesar de ello, este dispositivo era especialmente caro debido a la cantidad de mercurio que se necesitaba, por no hablar de su grado

<sup>2</sup><http://elmecanismoantecitera.blogspot.com.es/2011/07/lineas-de-retardo.html>



de toxicidad, el cual excedía los límites permitidos. Además, para que su funcionamiento fuera el adecuado, el mercurio tenía que mantenerse a una temperatura aproximada de 40° centígrados, por lo que era necesario un gasto considerable de energía para mantener dicha temperatura.

## 3.2 Cintas magnéticas

La cinta magnética es un tipo de medio o soporte de almacenamiento de datos que se graba en pistas sobre una banda plástica con un material magnetizado, generalmente óxido de hierro o algún cromato. El tipo de información que se puede almacenar en las cintas magnéticas es variado, como vídeo, audio y datos. En la Figura 3.6 podemos observar una cinta magnética de las que se emplearon en el UNIVAC

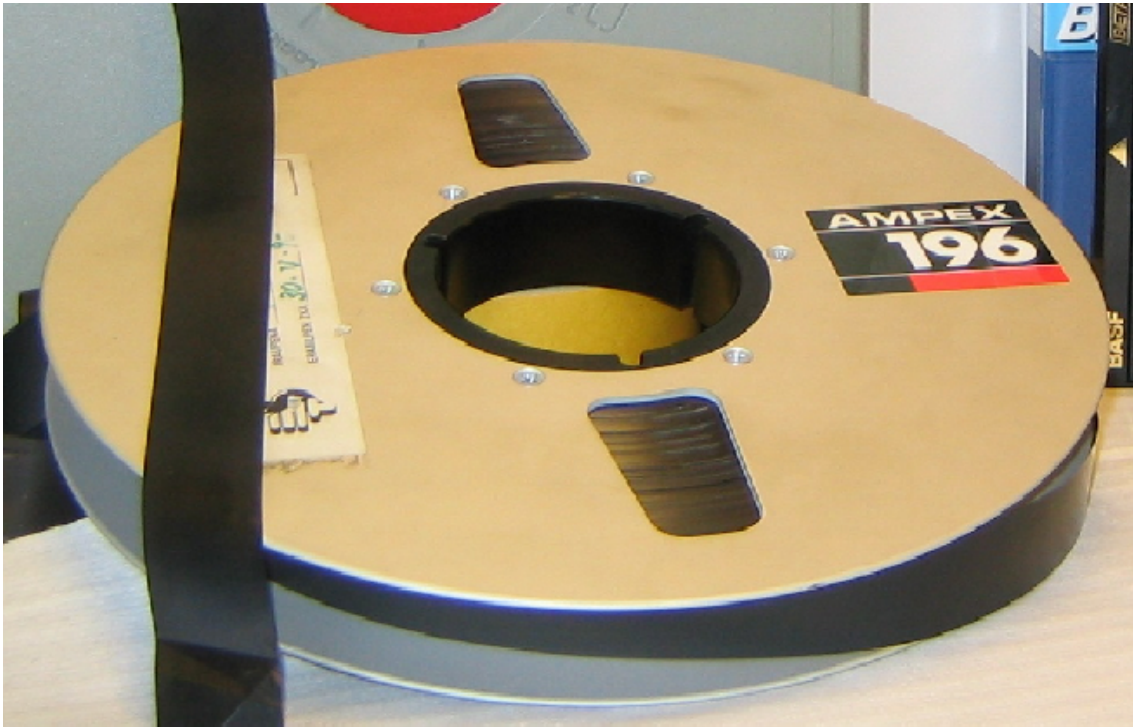


Figura 3.6: Cinta magnética empleada en la época del UNIVAC

Hay diferentes tipos de cintas, tanto en sus medidas físicas como en su constitución química, así como diferentes formatos de grabación, especializados en el tipo de información que se quiere grabar.

Hay diferentes tipos de grabación en cintas magnéticas, los cuales han sido perfeccionado con el paso de los años<sup>3</sup>:

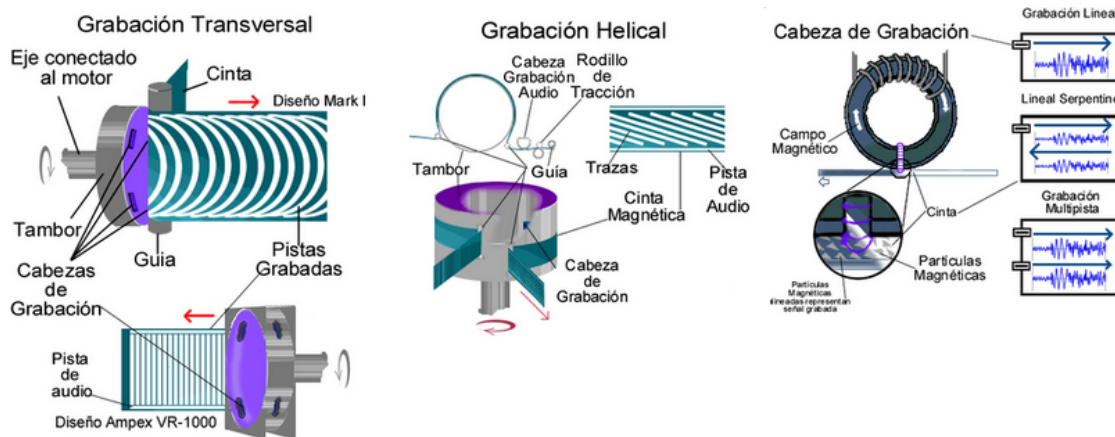
- **Grabación Lineal:** Con la grabación de señales análogas y digitales que se registran en la cinta magnética, como audio o datos codificados, se puede

<sup>3</sup>[https://es.wikipedia.org/wiki/Cinta\\_magn%C3%A9tica](https://es.wikipedia.org/wiki/Cinta_magn%C3%A9tica)

observar en las primeras cintas la traza de la señal en forma de línea horizontal, de acuerdo a las necesidades de ingenieros en audio vídeo o datos

- **Grabación Transversal:** La grabación lineal fue mejorada empleando este tipo de grabación, el cual aprovechaba la superficie grabable de la cinta magnética gracias al diseño de un tambor giratorio con cuatro cabezas de grabación para almacenar vídeo, lo que permitía una mayor cantidad de información almacenada a lo ancho de la cinta.
- **Grabación Helical:** La última mejora que se incorporó al tipo de grabación consistía en reducir el número de cabezales a dos, los cuales rotaban en diferentes ángulos hacia la dirección en la que se transportaba la cinta, de modo que la secuencia de grabación aprovechaba una mayor superficie de la cinta.

En la Figura 3.7 podemos observar los tres tipos de grabación anteriormente comentados.

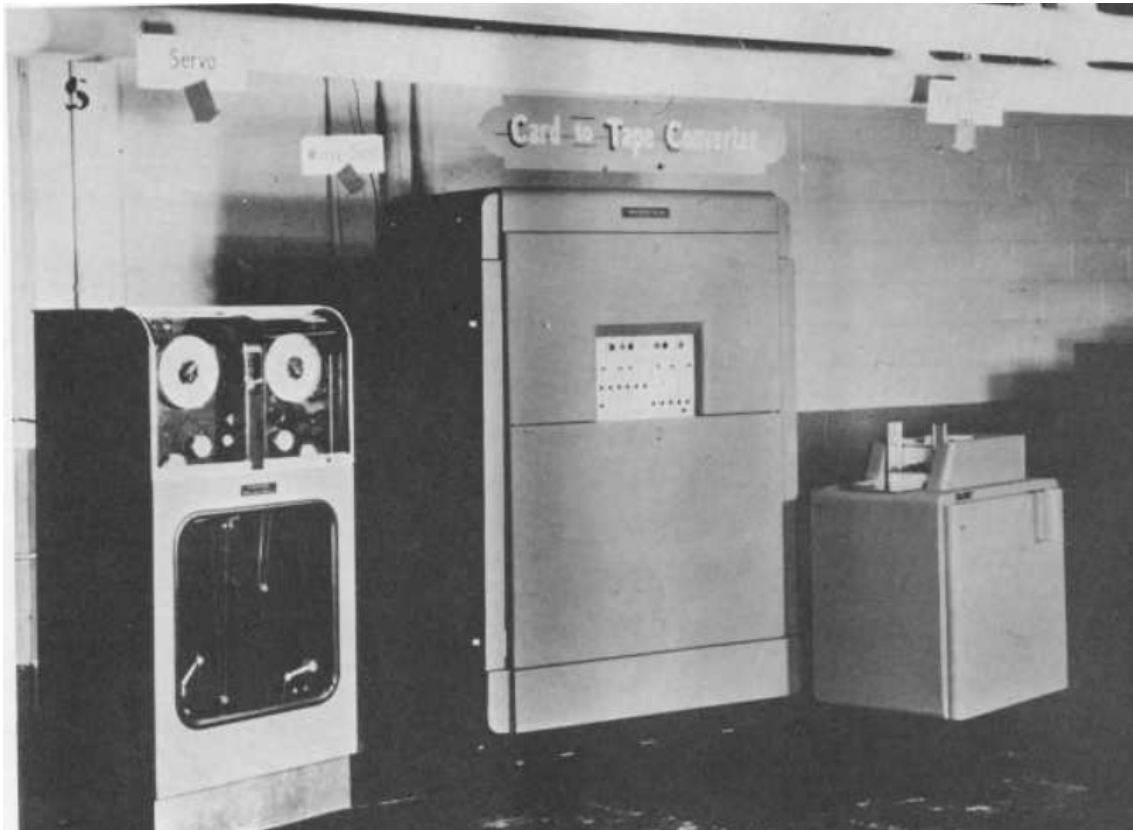


**Figura 3.7:** Los diferentes tipos de grabación de las cintas magnéticas

En el caso del UNIVAC, la cinta tenía 1,27 centímetros de ancho y entre 0,00254 y 0,00762 centímetros de espesor y cada carrete alojaba 360 metros de cinta, permitiendo almacenar en un área relativamente reducida más de un millón de caracteres.

La entrada y la salida de datos se efectuaba por medio de cintas magnéticas, usando un invento revolucionario de John Presper Eckert que condenó a la obsolescencia a las tarjetas perforadas de IBM, tan populares en esa época. Se usaba un equipo auxiliar externo (*out-line*) para convertir el medio de almacenamiento de datos, de tarjeta perforada a cinta magnética y de cinta magnética a tarjetas perforadas. Leía, revisaba y convertía hasta 120 tarjetas por minuto y grababa en la cinta magnética conocida comercialmente como Uniservo para el UNIVAC modelo 1103A<sup>4</sup>. En la Figura 3.8 podemos observar el denominado *Tape-to-Card converter*, empleado en la conversión Tarjeta Perforada-Cinta Magnética del UNIVAC.

<sup>4</sup><https://sites.google.com/site/hardwareconceptosbasicos/Inicio/ci>



**Figura 3.8:** Conversor de tarjeta perforada a cinta magnética empleado en el UNIVAC

### 3.3 Estética

---

En la época en la que nos encontramos, la estética no era precisamente uno de los puntos fuertes de este tipo de máquinas, ya que carecía de importancia, y estaban diseñadas para que funcionaran lo mejor posible sin tener en cuenta que tuvieran una estética agradable. De hecho, incluso a simple vista es difícil discernir si lo que estamos viendo es una máquina u otra, ya que eran muy parecidas todas ellas y para alguien que no hubiera trabajado con ellas podría resultar confuso.

Como hemos comentado anteriormente, las dimensiones del UNIVAC eran de 3,00 x 4,20 x 2,70 metros y pesaba cerca de las 13 toneladas.

A continuación se muestran una serie de figuras donde podemos apreciar al UNIVAC y algunos de sus componentes más importantes. La Figura 3.9 muestra un esquema de lo que sería el UNIVAC una vez estuviera completo y en pleno funcionamiento. Las Figuras 3.10 y 3.11 muestran una vista global de esta gran máquina, así como las Figuras 3.12 y 3.13 muestran una vista de todos los elementos que tenía el panel principal de esta máquina y desde donde se ejecutaban todos los comandos.

Por último, como complementos principales del UNIVAC podemos encontrar en la Figura 3.14 la impresora junto a las cintas magnéticas. Finalmente, en

la Figura 3.15 apreciamos las herramientas que se empleaban en la época para programar con el UNIVAC.



Figura 3.9: Esquema completo del UNIVAC

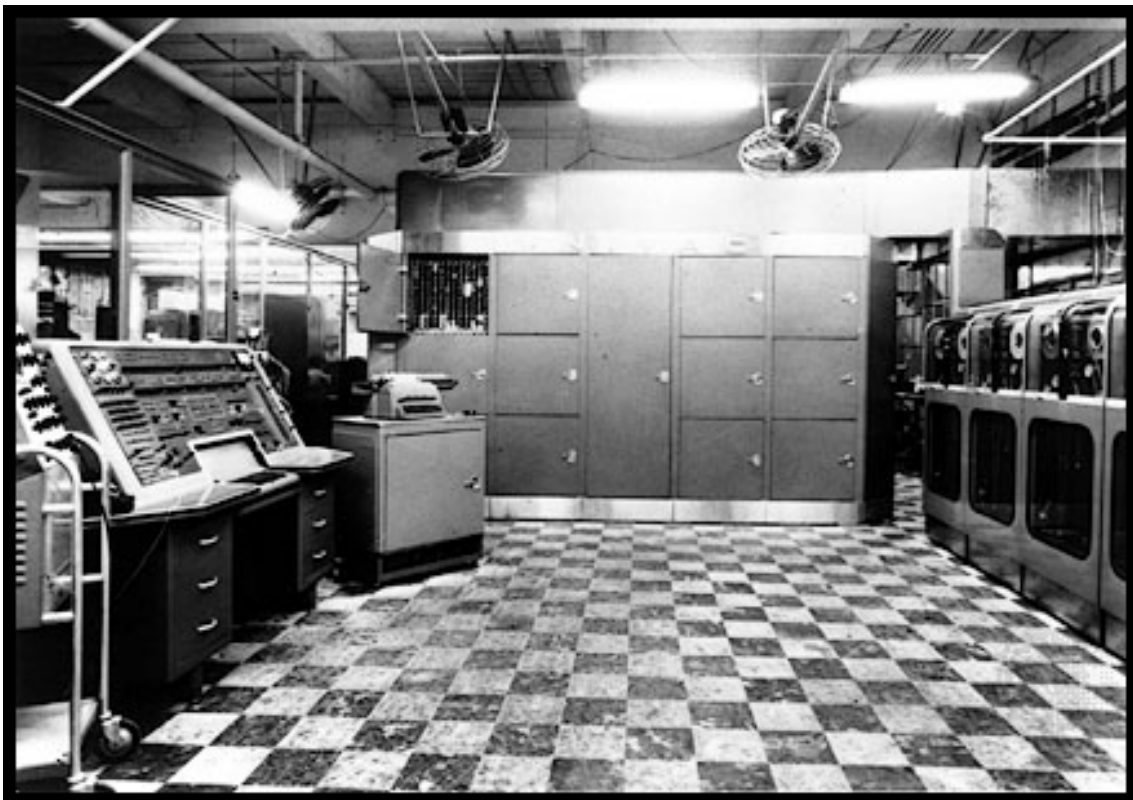


Figura 3.10: Perspectiva delantera del UNIVAC



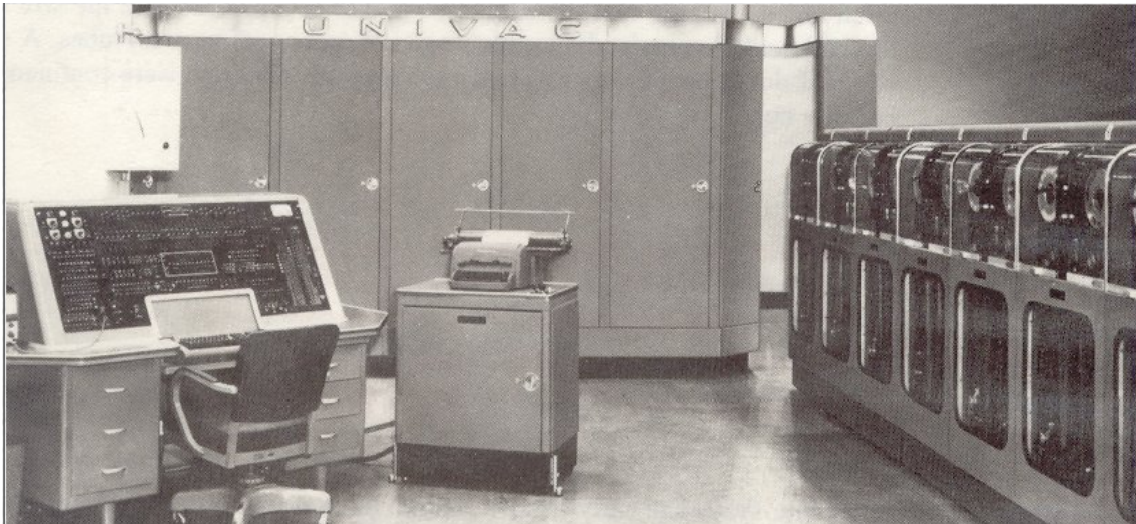


Figura 3.11: Perspectiva lateral del UNIVAC



Figura 3.12: Panel de mandos principal del UNIVAC





Figura 3.13: Teclado del UNIVAC



Figura 3.14: Impresora del UNIVAC junto al lector de cintas magnéticas

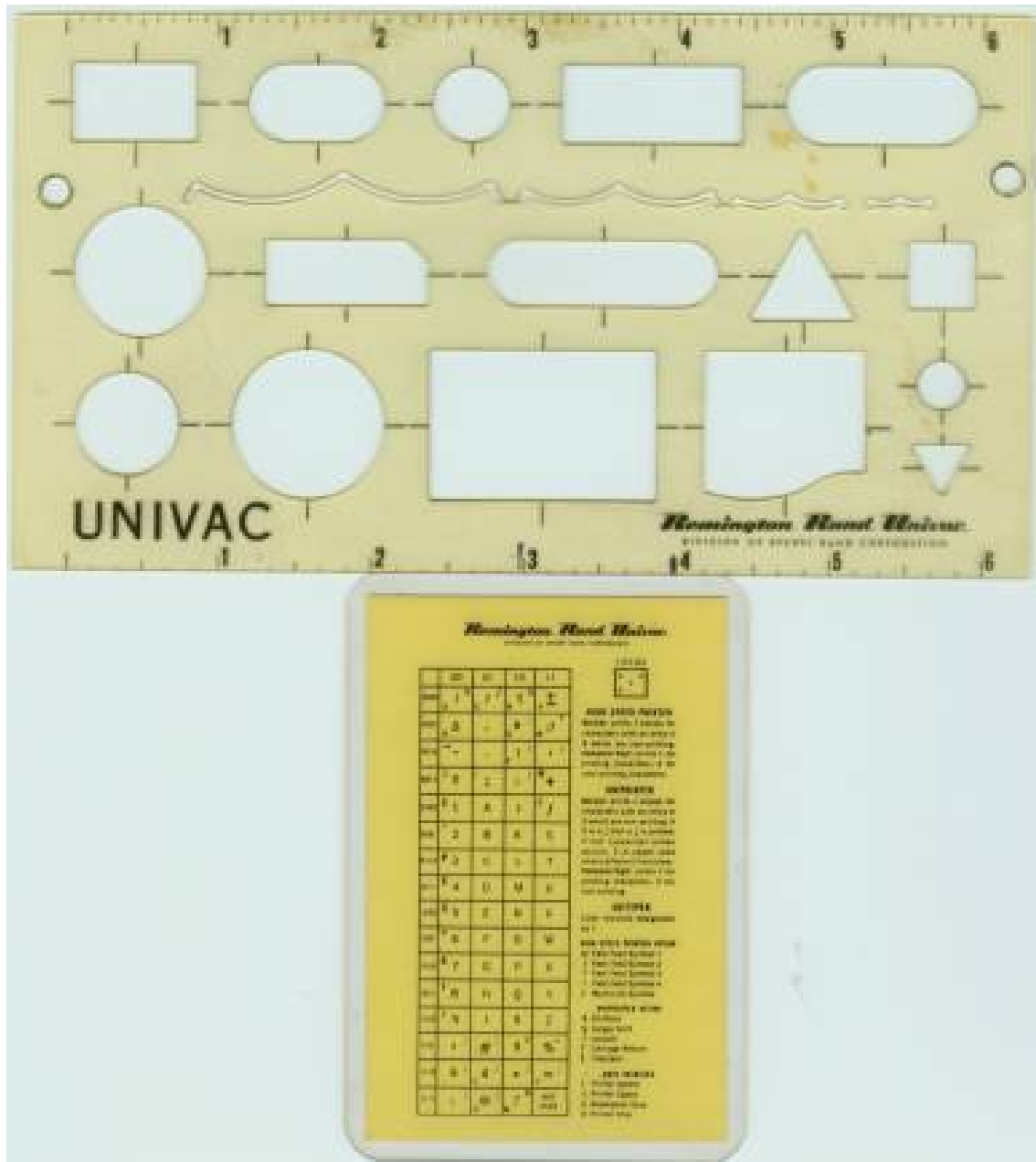


Figura 3.15: Herramientas empleadas en la programación del UNIVAC



# CAPÍTULO 4

## Programación

En este capítulo se describen los aspectos relacionados con la programación del UNIVAC. Para ello, será necesario comprender como funciona su memoria interna, qué registros interactúan entre si a la hora de ejecutar las instrucciones en la unidad de control, así como gran parte de las instrucciones que ésta emplea.

### 4.1 La memoria y la unidad aritmética

Para poder comprender como funciona el UNIVAC y los aspectos relacionados con su programación, comenzaremos hablando sobre la unidad principal: la memoria.

La unidad básica de la memoria del UNIVAC es la *palabra*. Una palabra siempre contiene 12 caracteres o dígitos. Hay 63 posibles caracteres disponibles, y cualquier combinación de 12 de estos 63 caracteres constituye una palabra. La Figura 4.1 muestra los 63 caracteres disponibles para la programación del UNIVAC, de los cuales nos centraremos básicamente, y para una comprensión más sencilla de la programación, en los números (0, 1, . . . , 9), las letras del alfabeto (A, B... Z) y el símbolo del guión (–), el cual emplearemos mayormente para separar palabras o indicar si un número es negativo o positivo <sup>1</sup>.

X33 ZONE \	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1	Δ	–	0	1	2	3	4	5	6	7	8	9	'	&	(
01	r	,	.	;	A	B	C	D	E	F	G	H	I	#	←	@
10	t	"		)	J	K	L	M	N	O	P	Q	R	\$	*	?
11	Σ	β	:	+	/	S	T	U	V	W	X	Y	Z	%	=	

= NEW PULSE COMBINATION     
A  
B = A - NEW SYMBOL REPRESENTATION    B - OLD SYMBOL REPRESENTATION

**Figura 4.1:** 63 caracteres válidos para la programación del UNIVAC

<sup>1</sup>[http://www.bitsavers.org/pdf/univac/univac1/UNIVAC1\\_OperMan.pdf](http://www.bitsavers.org/pdf/univac/univac1/UNIVAC1_OperMan.pdf)

La memoria del UNIVAC tiene capacidad para albergar o recordar 1000 palabras, las cuales podríamos visualizarlas en cajas para una mejor comprensión, y cada caja contiene una única palabra con sus correspondientes 12 caracteres. Para hacer referencia a cada una de las cajas, utilizamos una dirección de caja, las cuales están numeradas desde la 000 hasta la 999. Las palabras se leen de izquierda a derecha, y el guión también consume una de las doce posiciones que puede poseer una palabra. En la Figura 4.2 podemos observar que la palabra *JOHNNY-JONES* ocupa la caja número 001 y su correspondiente guión toma la posición número 7.

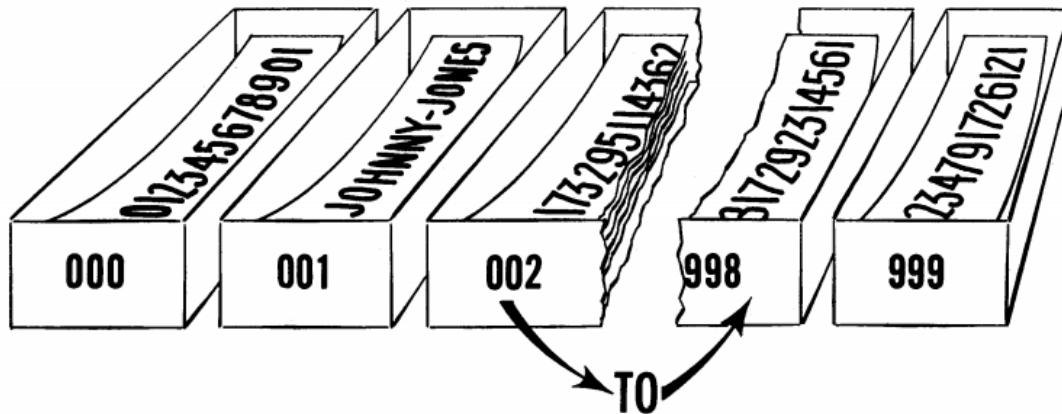


Figura 4.2: Ejemplo de las posiciones de memoria y su representación en cajas

A la hora de representar números, el guión toma un papel diferente y muy importante. Si el primer carácter de los 12 es un guión, el número que estamos representando es negativo, mientras que si es un 0, el número es positivo. Además, el computador asume que todos los números son menores de cierta magnitud, por lo que la coma decimal se sitúa entre los dos primeros dígitos del número. A pesar de esto, también es posible representar números de mayor longitud mediante el uso de otras técnicas.

Las instrucciones del UNIVAC se guardan del mismo modo en la memoria que los datos, por lo que comparten esas 1000 cajas de las que hablamos anteriormente. Una instrucción está constituida por los seis primeros dígitos de la palabra (de izquierda a derecha), de los cuales el cuarto, quinto y sexto dígito hacen referencia a la dirección en memoria de alguna de las cajas que vamos a emplear para la ejecución de dicha instrucción, y el primer y segundo dígito hacen referencia a que operación vamos a llevar a cabo sobre la dirección anteriormente nombrada.

Por último, cabe destacar que una instrucción emplea solamente 6 dígitos, y debido a que una palabra en memoria consta de 12, podemos tener 2 instrucciones por palabra. Los seis primeros dígitos de una palabra son los denominados «*Left Hand Instruction*» (posiciones digitales 1-6), mientras que a los seis últimos dígitos se les denomina «*Right Hand Instruction*» (posiciones digitales 7-12).

Otro de los elementos principales del UNIVAC es la **unidad aritmética (AU)**, la cual está compuesta por una serie de elementos: un sumador algebraico, un multiplicador-divisor, un comparador y numerosas cajas de memoria especiales las cuales denominaremos registros. De todos ellos, el registro más importante



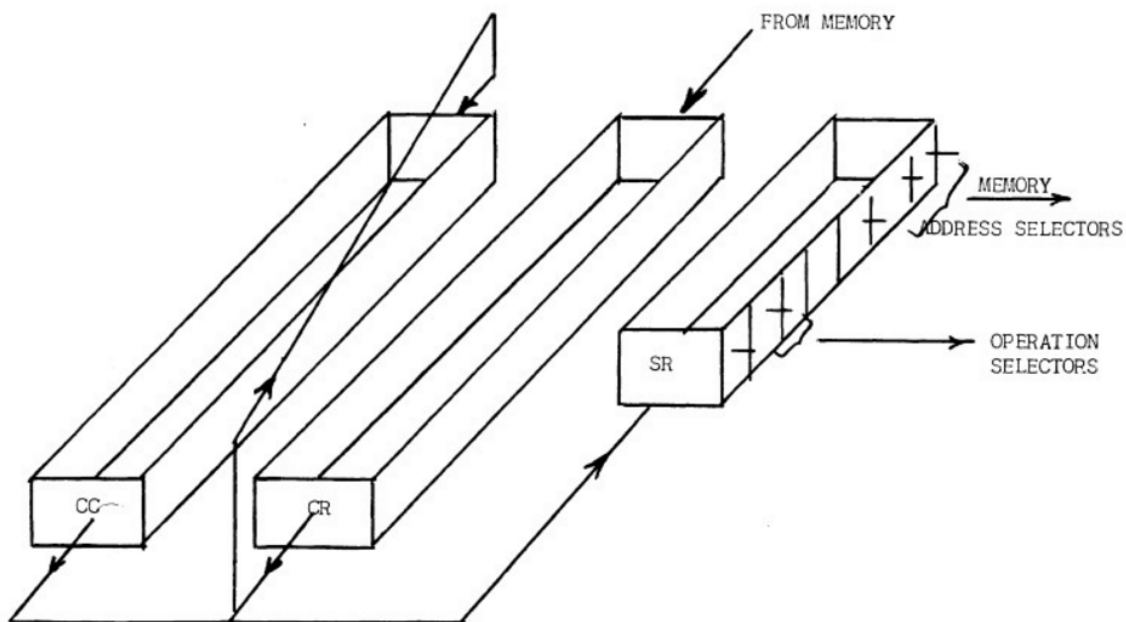
es el denominado **registro A (rA)**, el cual puede albergar una palabra, al igual que el resto de registros, y es el más importante debido a que en él será donde se guarden los resultados de cualquier cálculo aritmético.

## 4.2 La unidad de control

La unidad de control esta constituida por tres registros, los cuales necesitan estar totalmente sincronizados para poder efectuar las instrucciones en el orden correcto.

- El **contador de control**, CC, el cual guarda una palabra.
- El **registro de control**, CR, el cual guarda una palabra.
- El **registro estático**, SR, el cual guarda media palabra.

En la Figura 4.3 podemos observar como interactúan los diferentes registros a la hora de ejecutar las instrucciones dentro de la unidad de control.



**Figura 4.3:** Diagrama de la interconexión entre los tres registros de la unidad de control

El contador de control (CC) tiene la siguiente apariencia: 000000000XXX, donde XXX es un número entre 000 y 999, que corresponde con la dirección en memoria del siguiente par de instrucciones a ejecutar. El orden que lleva un ordenador que guarda sus instrucciones en memoria es, primero localizar dicha instrucción, extraerla y posteriormente ejecutarla.

El registro de control (CR) es usado para guardar una instrucción mientras la otra está siendo ejecutada. Esto es necesario debido a lo que comentábamos anteriormente sobre la paridad de las instrucciones en el UNIVAC, por lo que mientras una es ejecutada, el CR se encarga de guardar la parte restante de la palabra

que será ejecutada nada más acabe la primera. La extracción y ejecución de las instrucciones en el UNIVAC están divididas en 4 pasos, los cuales se identifican por una letra del alfabeto griego:

- **Alfa:** Los seis dígitos del *Right Hand* almacenados en el CC se duplican al SR. De este modo, el SR contiene la dirección de la memoria de la siguiente instrucción a ejecutar tras terminar la que está pendiente.
- **Beta:** Los circuitos del SR cargan el contenido de la celda de memoria especificada por la sección de dirección del SR para ser duplicada en el CR. Además, se añade un 1 en la posición número 12 del CC.
- **Gamma:** Los seis dígitos del *Left Hand* que están en CR son duplicados al SR, y estando en el SR se ejecuta la instrucción.
- **Delta:** La parte del *Right Hand* del CR es duplicada en SR y ejecutada.

El UNIVAC va efectuando estos pasos de manera cíclica, de modo que cuando el paso delta finaliza, comienza automáticamente por el paso alfa otra vez. Es importante remarcar que si el CC es 000000000000 inicialmente, el UNIVAC ejecutará la *Left Hand Instruction* encontrada en la celda de memoria número 000 y luego seguirá con la *Right Hand Instruction* en dicha celda. Después seguirá con la *Left Hand Instruction* de la celda 001, *Right Hand Instruction* de la celda 001, *Left Hand Instruction* de la celda 002, *Right Hand Instruction* de la celda 002, etc. Como hemos observado las instrucciones solo se ejecutan cuando están en el SR durante los pasos gamma y delta.

### 4.3 Instrucciones

---

En este punto introduciremos algunos aspectos básicos sobre las instrucciones que pueden ser ejecutadas en el UNIVAC. Existe 43 instrucciones las cuales pueden ser clasificadas en tres grandes grupos:

- Las instrucciones que **transfieren** información desde cierta dirección de memoria a otra o de un medio a otro.
- Las instrucciones que **realizan operaciones aritméticas**, tales como sumar, restar, multiplicar o dividir.
- Las instrucciones que permiten **escoger** entre la primera o la segunda secuencia de instrucciones.

La mayor parte de las instrucciones del UNIVAC son redundantes, debido a que pueden ser construidas con la combinación de otras. Se seleccionaron particularmente ciertas instrucciones para el UNIVAC después de estudiar muchas aplicaciones para determinar cuales eran las que mejoraban la velocidad de funcionamiento. Dichas instrucciones están organizadas en grupos, de los cuales nos centraremos en el estudio de tres de ellos (el grupo A, el grupo B y el grupo C),

debido a que, tal y como hemos comentado, serán suficientes para la elaboración del resto de instrucciones.

En el **Grupo A**, las seis instrucciones básicas son: B  $m$ , L  $m$ , C  $m$ , H  $m$ , 500  $m$  y 900.

**Instrucción B  $m$ :** Esta instrucción hace que el UNIVAC extraiga la palabra en la celda de memoria  $m$  y la sitúe en  $rA$  y  $rX$ , limpiando previamente el contenido de dichos registros. La palabra seguirá existiendo en la celda de memoria de la cual la hemos extraído, y ocurre lo mismo con todas las instrucciones de transferencia de datos fuera de la memoria principal. Podemos definir esta instrucción como una copia de la celda de memoria  $m$  en los registros  $rA$  y  $rX$ . La apariencia de los 6 dígitos que hacen referencia a esta instrucción son: B00XXX, donde XXX es una dirección de memoria entre el 000 y el 999.

**Instrucción L  $m$ :** Esta instrucción es muy similar a B  $m$ , debido a que transfiere la palabra en la celda de memoria  $m$  a  $rL$  y  $rX$ , eliminando el contenido previo de ambos registros.

**Instrucción C  $m$ :** Cuando el UNIVAC ejecuta esta instrucción elimina el contenido de la celda de memoria  $m$  y lo sitúa en  $rA$ . Después de la transferencia, el contenido de  $rA$  es reemplazado con ceros decimales.

**Instrucción H  $m$ :** Esta instrucción, cuando el UNIVAC se está ejecutando, guarda el contenido del registro  $rA$  en la celda de memoria  $m$ . El contenido de  $rA$  permanece inalterado, pero el de la dirección de memoria se borra.

**Instrucción 500  $m$ :** Esta instrucción hace que el UNIVAC imprima por pantalla el resultado de la celda de memoria  $m$ .

**Instrucción 900:** Esta instrucción hace que el UNIVAC pare de ejecutar instrucciones.

En la Figura 4.4 podemos observar más de cerca el teclado del UNIVAC y apreciar los botones de los registros que comentábamos anteriormente. ,



**Figura 4.4:** Teclado con los botones de registros  $rA$ ,  $rX$  y  $rL$  (entre otros).

En lo que al **Grupo B** respecta, únicamente hablaremos sobre la instrucción A m:

Instrucción A m: Esta instrucción se ejecuta en dos pasos. El primero coge la palabra de la posición de memoria  $m$  y la transfiere a rX, cuyo previo contenido es eliminado. Después, el contenido de rA y rX se manda al sumador algebraico, se suma, y el resultado es almacenado en rA.

Para finalizar, del **Grupo C** hablaremos sobre las cuatro instrucciones más importantes: 00 m, 9 m, U m y Q m.

Instrucción 00 m: Esta instrucción requiere que su segundo dígito sea un cero, debido al orden de salto. Esto le dice al UNIVAC que pase directamente a la siguiente instrucción sin alterar ningún registro o celda de memoria. La dirección  $m$  es ignorada.

Instrucción 9 m: Esta instrucción ordena una parada, lo que hace que el UNIVAC deje de ejecutar instrucciones, dejando los registros y la memoria sin modificaciones.

Instrucción U m: Las instrucciones previamente comentadas podrían funcionar tanto como *Right Hand Instruction* como *Left Hand Instruction*. Esto es cierto para todas las instrucciones del UNIVAC excepto para las instrucciones U m, Q m y T m, las cuales solo deben usarse como *Right Hand Instruction* para que funcionen adecuadamente. Concretamente, en este caso la instrucción U m es usada como una transferencia de control incondicional, la cual elimina los tres últimos dígitos del *Right Hand Instruction* del CC y los sustituye por la celda de memoria  $m$ . Por ejemplo, si la instrucción ejecutada es U00325 y el CC contiene 00000000114, el 114 es sustituido por 325.

Instrucción Q m: Esta instrucción ejecuta una transferencia de control condicional. Si el contenido de rA y el de rL es igual, la instrucción es interpretada como una instrucción U m. De no ser igual el contenido de dichos registros, la instrucción Q m se interpreta como un salto.

Para una mejor comprensión, y a modo de resumen, en la Figura 4.5 podemos observar un esquema de los registros que son empleados por las distintas partes del UNIVAC.

Arithmetic Registers	{ Register A, rA, which holds one word Register x, rX, " " " " Register L, rL, " " " " Register F, rF, " " " "
Control Registers	{ Control Counter, CC, a one-word register Control Register, CR, " " " " Static Register, SR, " half-word register
Transfer Registers	{ Register V, rV, a two-word register Register Y, rY, a ten-word register Register I, rI, a 60-word register Register O, rO, " " " "

Figura 4.5: Esquema de los registros que se emplean en las instrucciones del UNIVAC.

---

# CAPÍTULO 5

## Programas

---

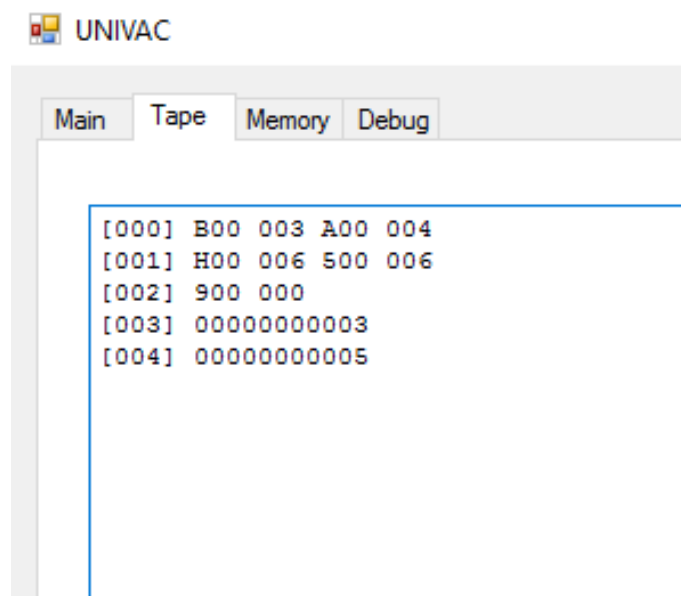
En este capítulo se muestra el funcionamiento de la programación explicada en el capítulo anterior basándonos en ejemplos. Para ello ha sido necesario programar en lenguaje UNIVAC los ejercicios que más adelante mostraremos. Además comentaremos cuales han sido los principales problemas que hemos encontrado al ponernos en la piel de los programadores de la época y las dificultades que podrían encontrar debido a los limitados recursos que disponían en aquel entonces.

Por último, haremos una comparativa del mismo programa programado en UNIVAC y en un lenguaje de alto nivel, en nuestro caso Java, y veremos las principales diferencias que encontramos.

### 5.1 Sumador simple

---

El primer ejemplo que vamos a mostrar se trata de un sumador simple, el cual podemos observar en la Figura 5.1.



```
UNIVAC
Main Tape Memory Debug
[000] B00 003 A00 004
[001] H00 006 500 006
[002] 900 000
[003] 00000000003
[004] 00000000005
```

Figura 5.1: Código de un sumador sencillo programado en UNIVAC



En primer lugar comentaremos las líneas del programa. Como podemos ver, el programa es sencillo y consta de 5 líneas, de las cuales la [003] y la [004] contienen los números que procederemos a sumar, en este caso son un 3 y un 5 respectivamente.

La línea [000] contiene dos instrucciones, B00 003 lee lo que hay en la celda de memoria [003] y lo lleva (sin modificar dicha celda) al registro rA, mientras que A00 004 suma el contenido del registro rA con el número de la celda [004].

La línea [001] contiene dos instrucciones más, H00 006, la cual guarda la suma anteriormente realizada en la celda de memoria [006] sin modificar el resultado en rA, por lo que dicho registro seguirá teniendo el resultado de la suma, y 500 006, la cual imprime por pantalla la cantidad almacenada en el registro [006], es decir, el resultado de la suma.

Por último, la línea [002] contiene una única instrucción, 900 000, la cual indica al UNIVAC que debe detener su ejecución.

Tras poner en funcionamiento el simulador, podemos observar en la siguiente Figura 5.2 el resultado obtenido.

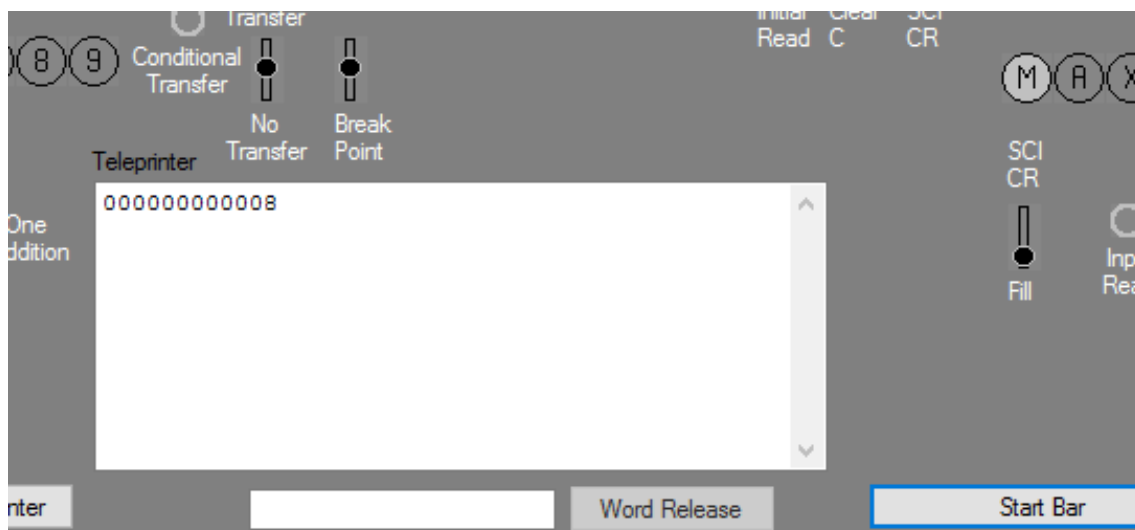


Figura 5.2: Restulado del ejemplo del sumador sencillo.

Además, el simulador del que disponemos posee una pestaña de *Debug*, la cual nos permite ver con detalle las operaciones que se van efectuando y el orden en que estas se llevan a cabo. En la Figura 5.3 podemos observar la de este ejemplo en concreto. Por último, podemos observar en la pestaña de *Memory* el estado en el que han quedado las celdas en la memoria, tal y como se muestra en la Figura 5.4.

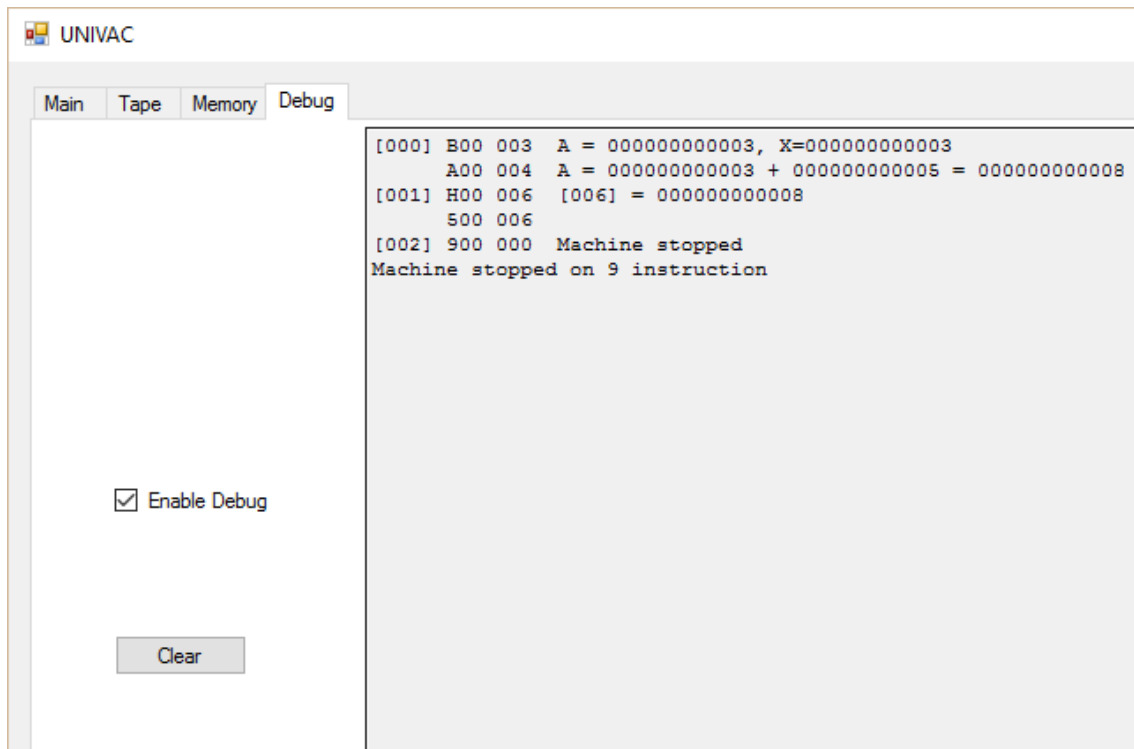


Figura 5.3: Pasos que lleva a cabo el UNIVAC a la hora de efectuar las instrucciones.

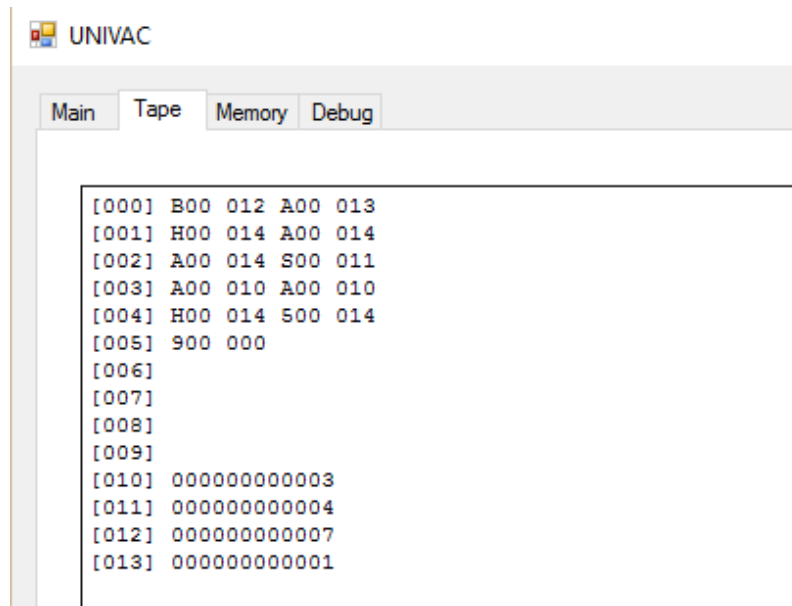
The screenshot shows the UNIVAC simulator interface with the 'Memory' tab selected. It displays a table of memory contents:

Address	Value	Bitmap
0	B00 003 A00 004	010101 000011 000011 000011 000011 000110
1	H00 006 500 006	011011 000011 000011 000011 000011 001001
2	900 000 000 000	001100 000011 000011 000011 000011 000011
3	000000000003	000011 000011 000011 000011 000011 000011
4	000000000005	000011 000011 000011 000011 000011 000011

Figura 5.4: Memoria tras la ejecución del sumador simple.

Otro ejemplo sobre el mismo ámbito pero con una complejidad un poco mayor sería el problema de resolver la siguiente operación:  $2A - B + 3(C + D)$ , teniendo en cuenta que  $A = 3$  y está en el registro [010],  $B = 4$  y está en el registro [011],  $C = 7$  y está en el registro [012] y  $D = 1$  y está en el registro [013].

Para resolver dicha operación hemos programado el siguiente código, el cual podemos apreciar en la Figura 5.5.



```

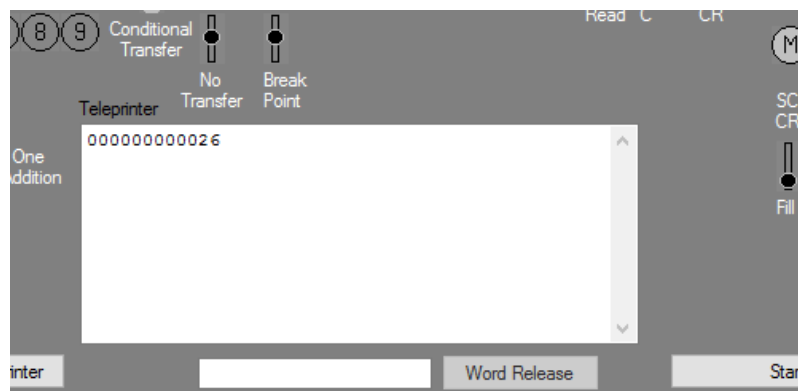
[000] B00 012 A00 013
[001] H00 014 A00 014
[002] A00 014 S00 011
[003] A00 010 A00 010
[004] H00 014 500 014
[005] 900 000
[006]
[007]
[008]
[009]
[010] 0000000000003
[011] 0000000000004
[012] 0000000000007
[013] 0000000000001

```

**Figura 5.5:** Código en lenguaje UNIVAC de la operación:  $2A - B + 3(C + D)$

Para la resolución de la operación propuesta, hemos tenido que programar el código respetando las propiedades matemáticas, tales como la propiedad asociativa, conmutativa y distributiva. Por ello, lo primero que vemos en el código es la suma de  $C + D$ , debido a que están dentro de los paréntesis, por lo que la primera línea del código suma el registro [012] con el [013] y la siguiente línea lo guarda en el registro [014]. Debido a que tenemos que multiplicarlo por 3, basta con sumar dos veces más el contenido del registro [014] a sí mismo. Posteriormente restamos  $B$  mediante la instrucción `S00 011`, la cual se encarga de hacer una suma normal, pero cambiando el signo del contenido de la celda de memoria en cuestión, por lo que convierte al número en negativo. Para acabar, basta con sumar dos veces el registro [010] al resultado, guardarlo en [014], imprimirlo por pantalla y parar la ejecución del simulador con las instrucciones que hemos comentado en el sumador simple.

Como podemos apreciar en la Figura 5.6, el resultado obtenido es el correcto, por lo que la programación del código es la idónea.



**Figura 5.6:** Resultado de la operación:  $2A - B + 3(C + D) = 26$

## 5.2 Factorial

El siguiente ejercicio que vamos a proceder a comentar se trata del **factorial (n!)**. La fórmula para calcular el factorial de un número es la siguiente:  $n! = n \times (n - 1) \times \dots \times 2 \times 1$ . En la Figura 5.7 podemos apreciar el código en lenguaje UNIVAC de dicho programa.

```

UNIVAC

Main  Tape  Memory  Debug

[000] L00 011 000 000
[001] M00 012 H00 013
[002] L00 012 B00 010
[003] 000 000 Q00 009
[004] B00 012 A00 011
[005] H00 012 L00 013
[006] 000 000 U00 001
[007]
[008]
[009] 500 013 900 000
[010] 04000000000000
[011] 01000000000000
[012] 02000000000000
[013]

```

**Figura 5.7:** Código del programa factorial en lenguaje UNIVAC

Como podemos apreciar, el código consta de 6 líneas principalmente, además de las líneas de la [009] a la [012] que se emplean para almacenar en memoria las constantes necesarias para la correcta ejecución del mismo.

En primer lugar, la línea [000] contiene la instrucción necesaria para cargar el número 1 y comenzar con la multiplicación gracias a la instrucción *L00 011* (recordemos que para multiplicar, es necesario cargar la palabra en memoria con la instrucción *L m* y no con *B m*, debido a que la primera carga el registro rL y la segunda el registro rA, y la multiplicación trabaja con el registro rL). La segunda

parte de la instrucción de esa celda es un simple salto, necesario totalmente para el salto incondicional que comentaremos más adelante.

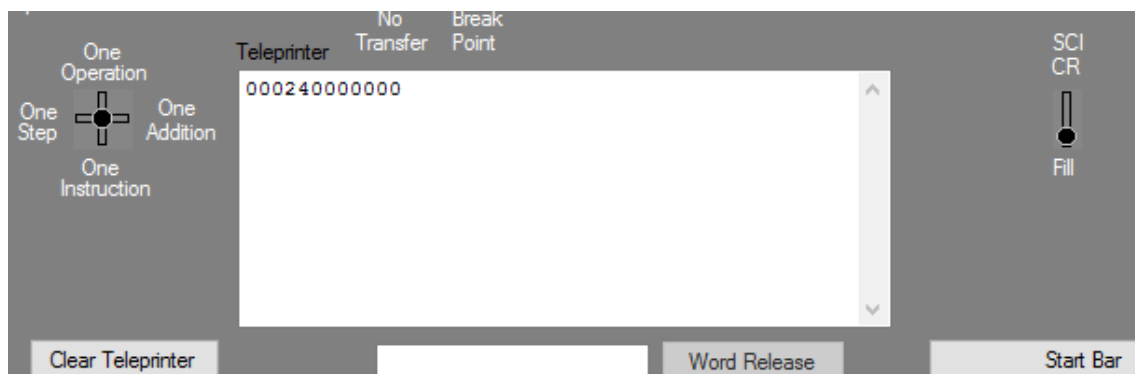
La siguiente línea, es decir, la [001], contiene la instrucción de multiplicación por el número almacenado en la celda [012] (el cual usaremos de acumulador para multiplicar por +1 cada vez) y guarda el resultado obtenido en la celda [013], que es la celda donde iremos acumulando el resultado del factorial.

Una vez multiplicado, cargamos en rL el contenido de la celda [012] y en rA el contenido de la celda [010] (donde tenemos el número del cual estamos calculando el factorial) mediante las instrucciones *L m* y *B m* respectivamente para, en la siguiente línea, comparar dichos registros mediante la instrucción *Q00 009*, la cual, si ambos registros son iguales, saltará a la línea [009] que es la encargada de imprimir el resultado por pantalla y parar la ejecución del programa.

En caso contrario, el programa continua con normalidad en la línea [005], la cual carga el contenido de la celda [012] una vez más para sumarle 1 y poder así multiplicar por el número siguiente una vez se complete el bucle. Además cargamos el contenido de la celda [013] para mantener el registro rL actualizado.

Por último, la línea [006] consta de una instrucción vacía y un salto incondicional, que redirige el código a la línea [001], haciendo que el código se ejecute recursivamente hasta que se complete la comparación de que el registro [012] y [010] sean iguales.

En nuestro ejemplo, hemos calculado el factorial del número 4, del cual podemos ver el resultado en la Figura 5.8. Cabe destacar que el resultado está reflejado con la coma decimal entre la posición de los dígitos 5 y 6, por lo que el resultado de  $4! = 24$ .



**Figura 5.8:** Restulado del programa factorial en lenguaje UNIVAC

Además, gracias al debug, podemos observar los pasos que va haciendo el UNIVAC en su ejecución. Estos pasos están reflejados en la Figura 5.9.



```

UNIVAC
Main  Tape  Memory  Debug
[000] L00 011 L=010000000000, X=010000000000
      000 000 Skip
[001] M00 012 A=010000000000 * 020000000000 = 002000000000
      H00 013 [013] = 002000000000
[002] L00 012 L=020000000000, X=020000000000
      B00 010 A = 040000000000, X=040000000000
[003] 000 000 Skip
      Q00 009 A<>L, Jump not taken
[004] B00 012 A = 020000000000, X=020000000000
      A00 011 A = 020000000000 + 010000000000 = 030000000000
[005] H00 012 [012] = 030000000000
      L00 013 L=002000000000, X=002000000000
[006] 000 000 Skip
      U00 001 Unconditional Jump to 001
[001] M00 012 A=002000000000 * 030000000000 = 000600000000
      H00 013 [013] = 000600000000
[002] L00 012 L=030000000000, X=030000000000
      B00 010 A = 040000000000, X=040000000000
[003] 000 000 Skip
      Q00 009 A<>L, Jump not taken
[004] B00 012 A = 030000000000, X=030000000000
      A00 011 A = 030000000000 + 010000000000 = 040000000000
[005] H00 012 [012] = 040000000000
      L00 013 L=000600000000, X=000600000000
[006] 000 000 Skip
      U00 001 Unconditional Jump to 001
[001] M00 012 A=000600000000 * 040000000000 = 000240000000
      H00 013 [013] = 000240000000
[002] L00 012 L=040000000000, X=040000000000
      B00 010 A = 040000000000, X=040000000000
[003] 000 000 Skip
      Q00 009 A=L, Jump to 009
[009] 500 013
      900 000 Machine stopped
Machine stopped on 9 instruction
  
```

Enable Debug

Clear

Figura 5.9: Debug del programa factorial en lenguaje UNIVAC

## 5.3 Contador

El siguiente ejemplo que procederemos a comentar se trata de un contador, el cual, en nuestro caso, cuenta del 0 hasta el 99 (debido a que hemos acotado el rango a menor estricto que 100). En la Figura 5.10 podemos apreciar el código en lenguaje UNIVAC de dicho programa.

```

UNIVAC
Main  Tape  Memory  Debug
[000] B00 009 C00 010
[001] 500 010 000 000
[002] B00 010 A00 008
[003] H00 010 L00 007
[004] 000 000 Q00 006
[005] 000 000 U00 001
[006] 900 000
[007] 00000000100\r
[008] 000000000010
[009] 000000000000\r
[010] 000000000000\r
[011] 000000000000\r

```

Figura 5.10: Código del programa contador en lenguaje UNIVAC.

El código consta de 12 líneas, de las cuales, de la [007] a la [011] conforman el contenido de la memoria que vamos a manipular durante la ejecución. La celda de memoria [007] contiene el número límite al que nuestro contador llegará, la celda [008] contiene la constante 1 y la [009] la constante 0, y la celda [010] contiene la cuenta actual, la cual iremos mostrando por pantalla en cada iteración. Cabe destacar que el comando contrabarra + r es un salto de línea, necesario para que la cuenta no salga toda seguida, sino una en cada línea.

Las 7 primeras líneas del código son las que conforman la parte importante, ya que contienen las instrucciones que hacen que la cuenta sea efectiva. En primer lugar, la línea [000] establece el contador a 0, y la línea [001] imprime por pantalla la cuenta que se lleva hasta el momento. La línea [002] suma 1 a la cuenta actual y la siguiente, es decir, la línea [003] guarda el resultado en la celda de memoria [010], chafando así el número anterior, y mediante la instrucción *L00 007* cargamos en el registro de memoria rL el número 100 para posteriormente usarlo en el comparador.

Las siguientes líneas del código son las más importantes debido a que son las que contienen los saltos, lo que harán que los bucles sean efectivos y continuemos con la cuenta de manera recursiva. De ese modo, la línea [004] contiene en primera instancia una instrucción vacía, debido a que la siguiente, que es una instrucción de comparación, debe estar siempre en el *Right Hand Instruction*. Por lo tanto, la segunda parte de esta instrucción es *Q00 006*, el cual compara el contenido de los registros rA y rL. En caso de que dichos registros sean iguales, el UNIVAC saltará a la línea [006], con la cual finalizaría la ejecución del programa. En caso contrario, seguiría el orden habitual y saltaría a la línea número [005], cuya línea de código contiene una instrucción vacía debido a que, igual que an-

teriormente, la siguiente instrucción condicional debe estar en el *Right Hand Instruction*. Dicha instrucción es *U00 001*, la cual hace un salto sin condición alguna a la línea [001]. De este modo conseguimos que el programa se ejecute hasta que la comparación de la línea [004] se cumpla y termine saltando a la línea [006].

Como podemos observar en la Figura 5.11, el resultado que obtenemos por pantalla son todos los números comprendidos entre el 0 y el 99.

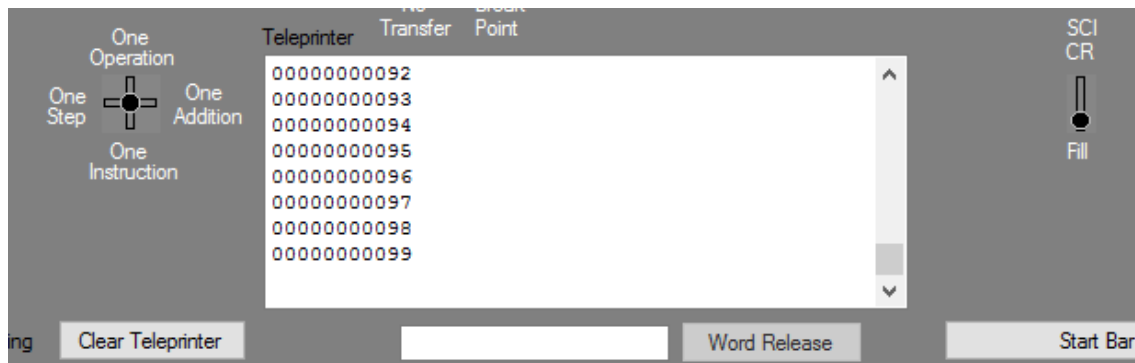


Figura 5.11: Resultado del programa contador.

Además, en la Figura 5.12 podemos ver los pasos que va ejecutando el UNIVAC gracias a la pestaña de *Debug*.

```
[001] 500 010
      000 000  Skip
[002] B00 010  A = 00000000098r, X=00000000098r
      A00 008  A = 00000000098r + 000000000010 = 00000000099r
[003] H00 010  [010] = 00000000099r
      L00 007  L=00000000100r, X=00000000100r
[004] 000 000  Skip
      Q00 006  A<>L, Jump not taken
[005] 000 000  Skip
      U00 001  Unconditional Jump to 001
[001] 500 010
      000 000  Skip
[002] B00 010  A = 00000000099r, X=00000000099r
      A00 008  A = 00000000099r + 000000000010 = 00000000100r
[003] H00 010  [010] = 00000000100r
      L00 007  L=00000000100r, X=00000000100r
[004] 000 000  Skip
      Q00 006  A=L, Jump to 006
[006] 900 000  Machine stopped
Machine stopped on 9 instruction
```

Figura 5.12: Debug del programa contador.

La traza muestra los pasos finales del programa, lo cual no es demasiado importante ya que son bucles y se repiten desde el 0 hasta el 99, por lo que todos

son iguales. Como podemos observar en la traza, y hemos comentado anteriormente, el UNIVAC coge el contenido de la celda de memoria [010], que es donde tenemos la cuenta actual, y le suma 1. Comparamos con el contenido de la celda [007] y no salta a la [006] hasta que no es igual a 100.

## 5.4 Programas en lenguaje de alto nivel

En esta sección haremos una comparativa de los programas vistos anteriormente programados en lenguaje UNIVAC, y éstos mismos programados en un lenguaje de alto nivel, en nuestro caso, **MIPS R2000**. Para ello hemos desarrollado el código equivalente al de los programas expuestos anteriormente en lenguaje ensamblador. Como podemos observar en la Figura 5.13, el código del programa sumador simple quedaría de la siguiente manera:

```

                                .data 0x10000000
A:                                .word 3
B:                                .word 5
C:                                .space 4

                                .text 0x00400000
                                .globl __start
__start:
                                la $t0, A
                                lw $t1, 0($t0)
                                lw $t2, 4($t0)
                                add $t3, $t1, $t2
                                sw $t3, 8($t0)
                                addi $v0, $zero, 10
                                syscall
                                .end

```

Figura 5.13: Sumador simple en ensamblador.

Como podemos apreciar, el programa hace exactamente lo mismo que el sumador simple, pero escrito en un lenguaje diferente. Lo primero que hemos hecho ha sido declarar en memoria el contenido de A y B y hemos declarado la longitud de la palabra C. Al comienzo del programa cargamos la palabra A y la guardamos en  $t1$  y hacemos lo propio con B pero en el registro  $t2$ . Posteriormente sumamos con el comando *add* y guardamos el contenido en el registro  $t3$  para, seguidamente, sacarlo por pantalla y hacer que el programa finalice.

Del mismo modo, hemos elaborado el código correspondiente para el programa que calculaba la fórmula:  $2A - B + 3(C + D)$  en lenguaje ensamblador. En la Figura 5.14 podemos observar el código nombrado.

```
                                .data 0x10000000
A:                                .word 3
B:                                .word 4
C:                                .word 7
D:                                .word 1
E:                                .space 4

                                .text 0x00400000
                                .globl __start
__start:
                                la $t0, A
                                lw $t1, 0($t0)
                                addi $t2, $zero, 2
                                mult $t1, $t2
                                mflo $t1
                                lw $t2, 4($t0)
                                sub $t1, $t1, $t2
                                lw $t2, 8($t0)
                                lw $t3, 12($t0)
                                add $t2, $t3, $t2
                                addi $t3, $zero, 3
                                mult $t3, $t2
                                mflo $t2
                                add $t1, $t1, $t2
                                sw $t1, 16($t0)
                                addi $v0, $zero, 10
                                syscall
                                .end
```

**Figura 5.14:** Código en ensamblador para la fórmula:  $2A - B + 3(C + D)$

Como podemos apreciar, y del mismo modo que en el programa anterior, el contenido de las palabras A, B, C y D está en memoria y se irán cargando posteriormente tras la ejecución del programa. Además, se ha declarado la longitud de la palabra E, la cual contendrá el resultado de la operación. En primer lugar cargamos la palabra A en el registro  $t1$ , del mismo modo que hemos hecho anteriormente, así como cargamos un 2 pero en el registro  $t2$ . Tras esto, multiplicamos el registro  $t1$  por 2, que es lo que contiene el registro  $t2$  y movemos el resultado a  $t1$  mediante el comando *mflo*. Posteriormente cargamos la palabra B en el registro  $t2$  y restamos el contenido al resultado que acabamos de guardar en  $t1$ . Después cargamos tanto el contenido de C como el de D en los registros  $t2$  y  $t3$  respectivamente. Sumamos dichos registros y guardamos el contenido en el registro  $t2$ , tras lo cual introduciremos un 3 en el registro  $t3$  para la posterior



multiplicación del resultado almacenado en  $\$t2$ . Dicha multiplicación la guardaremos en  $\$t2$  y finalmente sumaremos el resultado con el registro  $\$t1$  para tener el resultado final de la operación y, de este modo, imprimirlo por pantalla y poner fin a la ejecución del programa.

Siguiendo en la misma línea de trabajo, hemos elaborado el código del programa factorial en lenguaje ensamblador, tal y como podemos observar en la Figura 5.15.

```

                                .data 0x10000000
n:                               .word 4
f:                               .space 4

                                .text 0x00400000
                                .globl __start
__start:
                                la $t0, n
                                lw $t1, 0($t0)
                                addi $t2, $zero, 1
bucle:                          beq $t1, $zero, fin
                                mult $t2, $t1
                                mflo $t2
                                addi $t1, $t1, -1
                                j bucle
fin:                             sw $t2, 4($t0)
                                addi $v0, $zero, 10
                                syscall
                                .end

```

**Figura 5.15:** Código en ensamblador para la el programa factorial

Tal y como se muestra en la figura, lo primero que hemos hecho ha sido declarar la palabra  $n$  (la cual será el número del cual calcularemos el factorial), que en nuestro caso es 4, y la longitud de la palabra  $f$ , donde obtendremos el resultado. Posteriormente, la ejecución del programa comienza con la carga de  $n$  en  $\$t0$ , así como en  $\$t1$  para ir restando en el bucle hasta llegar a cero y hacer así que se detenga. Justo antes de comenzar el bucle, mediante la instrucción *addi*, introducimos en el registro  $\$t2$  un 1. Una vez comienza el bucle primero hacemos una comparación, en cuyo caso si el registro  $\$t1$  es cero, terminaría el bucle y saltaríamos a la etiqueta *fin*, la cual contiene las instrucciones necesarias para guardar el resultado e imprimirlo por pantalla. De lo contrario, el código se eje-

cutaría con normalidad y multiplicaríamos los registros  $t1$  y  $t2$ , cuyo resultado almacenaríamos en  $t2$ . Seguidamente, restaríamos 1 al registro  $t1$ , decrementando el número de la multiplicación en 1, y haríamos un salto incondicional a la etiqueta bucle, haciendo que el código sea recursivo y se sigan ejecutando las instrucciones anteriormente nombradas.

El último caso que programamos en UNIVAC, también ha sido transcrito a lenguaje ensamblador, por lo que la Figura 5.16 muestra nuestro código del programa contador en MIPS R2000.

```
                                .data 0x10000000
n:                                .word 101

                                .text 0x00400000
                                .globl __start
__start:
                                la $t0, n
                                lw $t1, 0($t0)
                                addi $t2, $zero, 1

bucle:                            beq $t1, $t2, fin
                                add $a0, $zero, $t2
                                addi $v0, $zero, 1
                                syscall
                                addi $a0, $zero, 13
                                addi $v0, $zero, 11
                                syscall
                                addi $t2, $t2, 1
                                j bucle
fin:                              addi $v0, $zero, 10
                                syscall
                                .end
```

Figura 5.16: Código en ensamblador para la el programa contador

Lo primero que debemos hacer, al igual que en casos anteriores, es declarar las palabras en memoria, en nuestro caso declararemos  $n$  como 101. Seguidamente, cargamos  $n$  en  $t0$  y hacemos lo mismo en el registro  $t1$ , así como incluimos un 1 en el registro  $t2$  mediante la instrucción de la suma inmediata. Llegados al bucle, lo primero que hacemos es comprobar si hemos llegado al final (a  $n$  en nuestro

caso), por lo que comparamos los registros  $\$t1$  y  $\$t2$  y en caso de ser iguales, saltaríamos a la etiqueta `fin`, que se encarga de imprimir el último número. En caso de que no sean iguales los registros anteriormente nombrados, el bucle se ejecuta con normalidad e iría imprimiendo todos los números hasta que la comparación del bucle se cumpliera, lo que haría que el programa se detuviera tras ejecutar el contenido de la etiqueta `fin`.

Además de los casos expuestos, también hemos elaborado un ejemplo de un programa similar al contador, pero en lugar de ir imprimiendo todos los números hasta  $n$ , suma todos los números hasta  $n$ . En la Figura 5.17 podemos observar el caso que comentamos.

```

                                .data 0x10000000
n:                                .word 100
S:                                .space 4

                                .text 0x00400000
                                .globl __start
__start:
                                la $t0, n
                                lw $t1, 0($t0)
                                add $t2, $zero, $zero
bucle:                          beq $t1, $zero, fin
                                add $t2, $t2, $t1
                                addi $t1, $t1, -1
                                j bucle
fin:                             sw $t2, 4($t0)
                                addi $v0, $zero, 10
                                syscall
                                .end

```

**Figura 5.17:** Código en ensamblador para el sumador hasta  $n$ .

Para comenzar hemos declarado tanto el número en cuestión  $n$  (en nuestro caso 100), como  $S$ , que será donde almacenemos el resultado de la suma hasta el número  $n$ . Al comenzar la ejecución del programa, cargamos  $n$  en  $\$t0$ , hacemos lo mismo con el registro  $\$t1$  y añadimos un 0 en  $\$t2$ . Tras esto, comienza la ejecución del bucle, cuya primera comparación es ver si  $\$t1$  es igual a cero, en cuyo caso saltará a la etiqueta `fin`, guardará el resultado de la suma e imprimirá el número. En caso contrario, sumará los registros  $\$t1$  y  $\$t2$  y guardará el resultado en  $\$t2$ . Una vez hecho esto, restará uno al registro  $\$t1$  y hará un salto incondicional a la

etiqueta bucle, completando así el ciclo y haciendo el código recursivo hasta que el registro  $\$t1$  llegue a 0.

## 5.5 Problemas encontrados programando

---

Esta sección de la memoria la dedicaremos a narrar la problemática que hemos ido encontrando a la hora de programar en lenguaje UNIVAC, las partes que más nos ha costado hacer, de modo que nos metamos en la piel de los programadores de aquella época y sintamos que necesidades tenían y cuales fueron las soluciones que fueron dando a según que tipo de problema se encontraban.

En un principio, el primero de los problemas que encontramos fue el orden que tenían que tener las instrucciones, así como su estructuración para que fuera más sencilla a simple vista. El problema vino dado cuando la una de los comandos, más concretamente las operaciones de comparación como  $Q m$ , no funcionaban si estaban situadas en el *Left Hand Instruction*. Al descubrir esta problemática, bastó con introducir una instrucción vacía  $000\ 000$ , de modo que las instrucciones de comparación quedaran siempre en el *Right Hand Instruction*.

Para continuar, destacaremos la problemática encontrada a la hora de emplear la operación de la multiplicación. El problema principal encontrado en esta parte es debido a la coma decimal, la cual se colocaba en la posición  $m + n$  de los números a multiplicar. Cuando comenzamos a programar, desconocíamos este dato, y observamos que las operaciones no resultaban como debían, ya que poníamos ejemplos como:  $00000000002 \times 00000000005 = 00000000000$  y el resultado obtenido no nos cuadraba, mientras que una operación de suma con los mismos datos sí era el correcto. Tras mucho investigar, nos dimos cuenta que para multiplicar, debíamos colocar los números en el segundo dígito, debido que el primero estaba reservado para el signo, de ese modo las operaciones resultaban de manera correcta teniendo en cuenta que la coma decimal se colocaba, como hemos comentado anteriormente, en la posición  $m + n$ , es decir, si los número a multiplicar eran 21 y 362, la coma decimal se colocaría en la posición  $2 + 3$  desde el segundo dígito.

Además, para cargar los datos a la hora de usar la instrucción de la multiplicación es diferente a la de cargar cualquier otro tipo de dato, y no debe hacerse con el comando  $B m$ , sino con el  $L m$ , debido a que la multiplicación selecciona el contenido del registro  $rL$  y no el de  $rA$ . Esto nos retrasó enormemente a la hora de realizar pruebas debido a que el resultado obtenido siempre era 0 y no entendíamos el motivo, hasta que nos dimos cuenta que era necesario realizar la carga de los datos con el comando adecuado.

Otro de los problemas que encontramos vino dado a la hora de utilizar el comparador  $T m$ , el cual debía comparar si un número dado era mayor estricto que otro, y de ser así, saltar hacia la dirección de memoria indicada en  $m$ . Desde un primer momento, el comando funcionaba correctamente, hasta que llegaba a un línea en la cual en vez de comparar de manera correcta, miraba si el número era menor estricto que el dado. Tras seguir investigando, observamos que si el signo del número cambiaba, la operación actuaba de manera diferente como hemos relatado, por lo que su actuación era normal. Para corregir este error optamos por

emplear la operación  $Q m$ , la cual compara si dos número son iguales, debido a que nos pareció más estable y predecible y se adaptaba de manera correcta a nuestro programa.



---

## CAPÍTULO 6

# El UNIVAC y su repercusión social

---

En este capítulo se pretende mostrar la repercusión social que causó el mundo de la informática en sus inicios, y más concretamente el UNIVAC en cuanto a ello. Este miedo se vería incrementado en un futuro por la literatura y la industria del cine, pero todo comenzó en la prensa, donde la gente comenzó a ver que las predicciones de un ordenador parecían de otro planeta.

### 6.1 El miedo a las computadoras

---

Todo comenzó en 1952, cuando la empresa *Columbia Broadcasting System (CBS)* solicitó la ayuda de la Remington Rand pidiendo algunas sumadoras e impresoras para realizar sus estimaciones con respecto a quién ganaría las elecciones a presidente de los Estados Unidos de América. ¿Que ganaría la Remington Rand con todo esto? La CBS prometió mostrar su logotipo de empresa ante las cámaras para hacer publicidad [1].

Alguien del departamento de relaciones públicas de la Remington sugirió que sería conveniente uso del UNIVAC I para realizar la estimación, aprovechando de ese modo el publicitar su estrenada nueva computadora. Fue entonces cuando se le ofreció a la CBS que se montara un pequeño espectáculo donde el UNIVAC I haría su estelar aparición y mostraría todo su potencial frente a las cámaras y miles de espectadores. Finalmente, la CBS accedió, pero no del todo conforme debido a que algunos altos ejecutivos de la cadena desconfiaban de dicha máquina y pensaban que había que tratar el tema con especial cautela.

Sería la primera vez en la historia que se haría algo similar, los ingenieros de la Remington emplearon cientos de miles de datos estadísticos de las pasadas elecciones (las cuales fueron en 1928) para que el UNIVAC I pudiera emplearlos, además se usaron politólogos que les indicaban qué datos eran más importantes y cuales podrían ser descartados, haciendo así una buena criba de la información [17]. Siendo conscientes de que miles de espectadores observarían el espectáculo, la Remington Rand tomó toda clase de precauciones para que todo saliera a la perfección, por lo que usaron tres máquinas para la ardua tarea: una para procesar los datos cuando estuvieran al aire, otra para verificar los resultados y una tercera a modo de respaldo por si una de las dos anteriores fallaban. Toda la información sería transmitida entre Nueva York y Filadelfia [15].

La transmisión del programa comenzó a las seis de la tarde del 6 de noviembre de 1952, y los resultados de la elección se enviaron por triplicado mediante teletipo y se pasaron a tres cintas magnéticas. Si el resultados de las tres máquinas coincidía, entonces dicha información producida se guardaba en una cuarta cinta. El programa que se elaboró para la misión detectaba posibles inconsistencias de la información y procesaba los datos según las tendencias históricas que el UNIVAC I tenía almacenado.

A las 8 y media de la noche y con tan solo el 7% de los votantes, el UNIVAC I predijo una victoria aplastante del candidato republicano *Dwight D. Eisenhower* frente a su contrincante demócrata *Adlai E. Stevenson*, lo que causó un gran revuelo debido a que lo que la máquina estaba prediciendo era todo lo contrario a lo que los grandes expertos en política afirmaban. Una curiosa anécdota referente a esto fue que el reportero responsable de dar la noticia a los telespectadores, *Charles Collingwood*, no se atrevía a dar dicha información debido al escándalo que causaron los datos expuestos por la máquina, e hizo que se comprobaran hasta nueve veces antes de salir e informar a la audiencia. En la Figura 6.1 podemos apreciar al presentador sujetando los datos extraídos del UNIVAC I junto con Eckert y un operario de la máquina.



**Figura 6.1:** Charles Collingwood, Eckert y un operario del UNIVAC I la noche de las elecciones.

El director de investigación de la Remington Rand del momento, *Arthur Draper*, tuvo que ordenarle a los ingenieros que modificaran el programa del UNI-

VAC I para que coincidiera con lo que los expertos en política afirmaban y que el escándalo se calmara. Tras realizar una serie de modificaciones, el resultado seguía siendo una victoria para los republicanos, aunque esta vez con una ventaja mínima frente a los demócratas [17].

Pensando en el escándalo público, a las 10 de la noche la CBS transmitió los resultados alterados para que coincidieran con los expertos, debido a que no querían exponerse a equivocarse frente a esa multitudinaria audiencia. Solo una hora más tarde, se constató que el UNIVAC I había acertado de lleno en su predicción, y a pesar de todas las alteraciones sufridas en su programa, las posibilidades eran de 100 sobre 1 de que ganaría *Eisenhower*. Tras este bochornoso espectáculo, la CBS tuvo que salir de nuevo al aire y admitir todo lo que había sucedido detrás de las cámaras, quedando totalmente en ridículo junto con el presidente de la Remington Rand, el cual admitió todas las modificaciones que se le habían causado al programa para que la predicción coincidiera con los expertos políticos.

Cuando todo acabó, quedó constante que el UNIVAC I había predicho con tan solo el 7% de los votantes que el candidato republicano obtendría 438 distritos electorales y el conteo oficial arrojó la cifra de 442, por lo que el UNIVAC I solo falló por un 1% [17].

Desde ese momento, se cree que la gente le tiene miedo a las computadoras, debido a que, frente a todo pronóstico y frente a toda modificación, las predicciones ganaron frente a la experiencia de los más reputados expertos en política.

## 6.2 La publicidad del UNIVAC

---

Durante la elaboración del UNIVAC I hubo un tema que dio mucho de qué hablar dentro de Remington Rand: ¿cómo venderían semejante máquina y la harían viable comercialmente hablando? Se comentaba que la máquina era demasiado grande como para poder hacerla comercial, demasiado cara tras su finalización, debido a que se había presupuestado algo muy lejano a la realidad, y que las probabilidades de que se vendiera lo suficientemente bien como para que la empresa tuviera beneficios era más bien remota.

Debido a esto, Remington Rand comenzó a elaborar un estudiado plan de marketing para poder publicitar como era debido aquella magnífica máquina. Bajo el lema *Want to buy a brain?* [5], ofrecían una serie de servicios que jamás antes se habían ofertado. Alardeaban de que la máquina podía hacer todo lo que tu cerebro fuera capaz de pensar o imaginar, y como ejemplo decían cosas como: Preguntale como de grueso debe ser el cable de un puente nuevo, y te lo dirá. Pregunta como de rápido puede ir un coche nuevo antes de que su cigüeñal comience a vibrar, y te dará la velocidad exacta, textualmente «*Ask one of these machines how thick the cables for a new bridge should be, and it will tell you. Ask how fast a new automobile can go before its crankshaft starts to vibrate to pieces, and it will give the exact speed*».

A la hora de vender el producto, brillaron por su perspicacia, ya que comparaban continuamente con los ordenadores del mercado e informaban al cliente sobre todas sus mejoras. Su principal competidor de aquellos tiempos era el ENIAC, cuyo precio rondaba los 400,000 dólares, y el **Harvard Mark I**, cuyo pre-

cio rondaba los 250,000 dólares y el cual podemos apreciar en la Figura 6.2. Ellos mejoraban cualquier oferta y lo pusieron a la venta por 200,000 dólares. Además, presumían de que el UNIVAC era mucho más rápido y preciso que sus competidores, y que sería muy útil sobretodo en proyectos de ingenierías (entre otros). A modo de ejemplo, aseguraban que mientras que un modelo de bombardeo podría costar cerca de los 30,000,000 de dólares, la máquina podría simularlo todo incluso antes de que el avión fuera construido, y eliminando la necesidad de cualquier modelo experimental de ello.



Figura 6.2: Harvard Mark I

Por último, para las personas más entendidas sobre el tema, en su publicidad explicaban ligeramente el modo de funcionamiento interno de la máquina, de ese modo demostraban a los entendidos en la materia el gran potencial que ésta poseía. Hacían especial hincapie en el uso de los tubos de vacío, ya que eran la base de operaciones del UNIVAC. También estaban muy orgullosos de la memoria que habían elaborado, por lo que aseguraban que era la más rápida para la época y capaz de albergar infinidad de datos (en parte gracias a las cintas magnéticas). En la Figura 6.3 podemos observar un ejemplo que los creadores del UNIVAC mostraban al público.

### How Univac Adds by Two's on "Gate" Vacuum Tubes

To Univac, numbers are a binary code of electrical pulses and blank spaces (no pulses). Typical code, simplified slightly from Univac's, is shown below (X means pulse, and 0 means blank space).

0=0000	5=0X0X
1=000X	6=0XX0
2=00X0	7=0XXX
3=00XX	8=X000
4=0X00	9=X00X

This code is added by ordinary rules:  $0 + 0 = 0$ ;  $X + 0 = X$ ;  $X + X = 0$  and carry X (like  $5 + 5 = 0$  and carry 1). Try adding 3 and 6 (below). First digit pair (at right) gives X. Second digit pair gives 0 and carry X. Third digit pair, plus carry from second, gives 0 and carry X. Fourth digit pair plus carry gives X.

$$\begin{array}{r} 3 \quad 00XX \\ + 6 \quad 0XX0 \\ \hline 9 \quad X00X \end{array}$$

Univac adds much this way with circuits similar to the one below. "Gate" tubes—which transmit a pulse only when both their grids are positive—are arranged in an "and" circuit and an "or" circuit that are used together. The "or" circuit has

two tubes, each having one grid kept positive; the other grid must be made positive by an incoming pulse. Each tube receives one digit of a pair being added. The "or" circuit thus transmits only a blank if two blanks are received. If one blank and one pulse are received, it transmits a pulse, since one tube has its negative grid made positive. If two pulses are received, however, it must transmit a blank and carry a pulse to the next addition. This it does with the "and" circuit.

The "and" circuit has one "gate" tube, with both grids normally negative. Thus it sends a pulse only when it receives two pulses, one for each grid. It receives the same pairs of signals that the "or" circuit gets, but operates only when both signals are pulses. When this happens, the "and" pulse affects the "or" tubes, making one grid in each "or" tube negative, and the "or" circuit transmits a blank. The "or" answer becomes the total for that column, while the "and" answer is fed to a delay circuit. This "and" carry pulse is delayed until it can be added to the sum resulting from the following pair of digits. The second addition is done by another set of "gate" tubes much like the first.

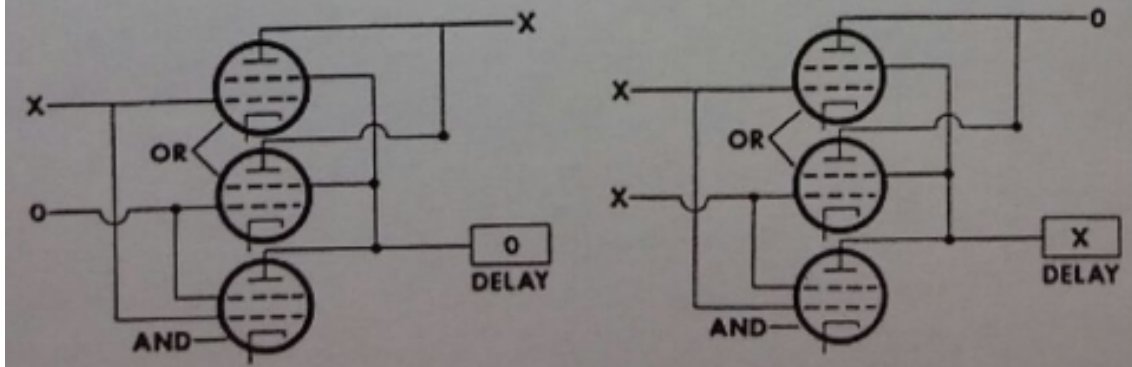


Figura 6.3: Ejemplo práctico sobre el UNIVAC y sus tubos de vacío





---

## CAPÍTULO 7

# Material didáctico

---

El Museo de Informática de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València [7] fue inaugurado el 11 de diciembre de 2001, y se encuentra incluido desde 2015 en la base de datos mundial de socios del Consejo Internacional de Museos, ICOM. El recorrido del museo nos refleja el vertiginoso progreso de la tecnología durante las últimas décadas y nos ofrece la posibilidad de visitar su exposición a través de las diferentes vitrinas, que muestran de una manera didáctica y divulgativa la evolución de los primeros computadores personales, dispositivos de almacenamiento, estaciones de trabajo, sistemas operativos, videojuegos entre otros.

El 13 de mayo de 2013 fue reconocido oficialmente por la Conselleria d'Educació, Cultura i Esport como museo oficial de la Comunidad Valenciana, y finalmente el Consejo Ejecutivo del ICOM-España ha aprobado que el Museo de Informática de la UPV forme parte de esta organización internacional.

En la Figura 7.1 podemos apreciar la placa acreditativa como museo oficial y el logotipo del ICOM.



Figura 7.1: Placa del museo y logotipo de la ICOM

La finalidad de este museo es difundir y promover en la sociedad el conocimiento del patrimonio digital que se ha generado a través de los tiempos, y hacer recapacitar sobre su función en la sociedad a lo largo de su historia, presente y del que puede tener en el futuro.

Este trabajo final de grado pretende ayudar al Museo de Informática a que cumpla con su misión mediante la incorporación de una página web dentro de la sección «Artículos: El museo te cuenta» del sitio web de dicho museo. Para ello se incluyen textos explicativos, imágenes y vídeos relacionados con el UNIVAC I.

La inclusión de esta información en la web se ha realizado en colaboración con el personal del museo encargado a tal efecto respetando los formatos y estilos ya existentes en dicha web, y quedando una página como la que se muestra en la Figura 7.2.



[ES](#) [CA](#) [EN](#) [FR](#)

[museo@inf.upv.es](mailto:museo@inf.upv.es) [96 387 7200](tel:963877200)



Search...

[INICI](#) [EL MUSEU](#) [ACTIVITATS](#) [HORARIS I VISITES](#) [DIDÀCTICA](#) [SABER MÉS](#) [DONACIONS](#) [ENLLAÇOS](#) [CONTACTE](#)

Home »

## El UNIVAC, el primer computador comercial de Estados Unidos

Diego García Garijo

UNIVAC es el acrónimo de *Universal Automatic Computer*, fue diseñado en la década de los años 50 del siglo XX por la empresa **Remington Rand** y es considerado el primer computador comercial desarrollado en Estados Unidos. Concebido por los pioneros informáticos **John W. Mauchly** y **John P. Eckert**, este computador estaba orientado a grandes empresas, tanto privadas como públicas, debido a su elevado coste y sus requerimientos de instalación. Con este computador, entre otras cosas, **se elaboró el censo en los Estados Unidos del año 1951** y **se predijo correctamente el ganador de las elecciones presidenciales en 1952**. En ambas tareas el UNIVAC I, por aquel entonces una estrella mediática, demostró ser una máquina rápida y precisa.



“

Vista general de la instalación del UNIVAC

A la izquierda tenemos a Mauchly a los mandos del UNIVAC junto a dos operarios de la máquina.

Figura 7.2: Página web creada en castellano para el Museo de Informática sobre el UNIVAC I

El Museo de Informática pertenece a los museos oficiales de la Comunidad Valenciana, por lo tanto, se ha procedido a crear una segunda página web idéntica a la mencionada pero en este caso traducida al valenciano, lengua cooficial de esta comunidad, dando como resultado una página como la que aparece en la Figura 7.3.



**museu informàtica** **COM**  
FEDERATIONAL COUNCIL OF MUSEUMS

Search...

INICI EL MUSEU ACTIVITATS HORARIS I VISITES DIDÀCTICA SABER MÉS DONACIONS ENLLAÇOS CONTACTE

Home >

## L'UNIVAC, el primer computador comercial dels Estats Units

Diego García Garijo

UNIVAC és l'acrònim d'*Universal Automatic Computer*, va ser dissenyat en la dècada dels anys 50 del segle XX per l'empresa **Remington Rand** i és considerat el primer computador comercial desenrotllat als Estats Units. Concebut pels pioners informàtics **John W. Mauchly** i **John P. Eckert**, este computador estava orientat a grans empreses, tant privades com públiques, a causa del seu elevat cost i els seus requeriments d'instal·lació. Amb este computador, entre altres coses, **es va elaborar el cens als Estats Units de l'any 1951 i es va predir correctament el guanyador de les eleccions presidencials en 1952**. En ambdós tasques l'UNIVAC I, en aquell moment una estrella mediàtica, va demostrar ser una màquina ràpida i precisa.

Vista general de la instal·lació de l'UNIVAC.

A l'esquerra tenim Mauchly als comandaments de l'UNIVAC junt amb dos operaris de la màquina.

**Figura 7.3:** Pàgina web creada en valenciano para el Museo de Informática sobre el UNIVAC I

## 7.1 Estructura de las páginas web

Las dos páginas web desarrolladas a partir de este documento tienen el mismo contenido pero en idiomas diferentes —castellano y valenciano— y su organización sigue la estructura que se explica en este apartado.

En primer lugar se hace una presentación de la máquina, explicando aspectos generales de modo que al lector le resulte atractivo el tema y se interese por él. Seguidamente se hace referencia a los aspectos más físicos de la máquina, dando a ver por qué estaba compuesto, cuales eran sus medidas y quiénes fueron sus creadores.

Tras estos, explicamos aspectos más técnicos referentes a la memoria, las cintas magnéticas donde se almacenaban los datos, el tipo de programación, así como un ejemplo explicativo de uno de los programas que hemos hecho durante el presente trabajo de fin de grado.

Por último, se habla de su aspecto más social, relatando como fueron las elecciones del año 1952 y cuán polémico fue dicho evento, así como de la publicidad que se le dio a la máquina. Finalmente podemos ver un vídeo donde anunciaron por primera vez en televisión el UNIVAC I.

---

---

# CAPÍTULO 8

## Conclusiones

---

Este trabajo ha consistido en un estudio dedicado al UNIVAC I, abarcando sus aspectos históricos y técnicos, dando una visión de quién, dónde y cómo fue creada, los retos y logros alcanzados en ese proceso, así como detalles de su arquitectura interna y funcionamiento. En él se ha pretendido alcanzar los objetivos que se describieron en el primer capítulo de la siguiente forma:

1. Se ha enmarcado históricamente la aparición del UNIVAC I y todo lo relacionado con sus creadores, primera empresa y sucesoras, los problemas que acontecieron en aquella época durante su creación, así como el panorama empresarial del momento y el ámbito social de la época.
2. Se ha abordado el tema de la arquitectura del UNIVAC I y hemos sido conocedores del tipo de electrónica que empleaba, así como el tipo de memoria y las características esenciales de las cintas magnéticas.
3. Se ha explicado como se programaba en aquella época con semejante máquina, haciendo especial mención a todo tipo de registros y componentes software de la computadora. Además se han visto numerosas instrucciones, las cuales pueden servir al lector para crear sus propios programas.
4. Se han ejemplificado varios programas, de modo que el lector pueda experimentar con el simulador y compare sus códigos con los que nosotros ofrecemos. Hemos experimentado también los problemas que sufrían los programadores de la época y documentado, de modo que pueda servir para ayudar a futuros estudiantes de la materia a que no cometan los mismos errores.
5. Se ha introducido al lector en el ambiente que había en la década de los 50, cuando fue creado el UNIVAC I, para que pudiera sentir lo que la sociedad de aquél entonces sentía de cara a las nuevas tecnologías.
6. Se ha creado una página web para ayudar a la difusión de todo este material didáctico y donado al Museo Informático de la Universidad Politécnica de Valencia, cuyo director es el tutor del presente trabajo de fin de grado.

Para la realización del presente trabajo ha sido necesario, en primer lugar, aprender los aspectos referentes a la herramienta LaTeX, mediante la cual se ha



redactado todo el proyecto. Ha sido necesario consultar y estudiar los manuales de su útil interfaz, la cual ha sido de gran ayuda a la hora de trabajar, ya que una vez aprendido su funcionamiento, resulta realmente rápida y fácil de gestionar.

## 8.1 Trabajo futuro

---

A partir de este trabajo fin de grado se pueden plantear diversos futuros proyectos, que como éste, pueden ayudar al Museo de Informática en su labor de difusión y preservación del patrimonio digital. Algunos de estos futuros proyectos podrían ser máquinas que hemos nombrado en nuestro trabajo, como el *Harvard Mark I*, el *Z22* o el *IBM 701*.

También podría seguirse por la misma línea de investigación el estudio del UNIVAC II, sucesor del UNIVAC I, y comparar las características de ambos, viendo en que mejoraron al crear la segunda versión de esta gran máquina.

# Bibliografía

---

- [1] A. Coello Coello, Carlos; *El origen del miedo a las computadoras*; Cinvestav; abril-junio 2007; Disponible en [http://www.cinvestav.mx/Portals/0/SiteDocs/Sec\\_Difusion/RevistaCinvestav/abril-junio2007/miedo.pdf](http://www.cinvestav.mx/Portals/0/SiteDocs/Sec_Difusion/RevistaCinvestav/abril-junio2007/miedo.pdf)
- [2] A. Coello Coello, Carlos; *Breve historia de la computación y sus pioneros*; Fondo de cultura económica; primera edición en castellano; 2003.
- [3] Hally, Mike; *Electronic brains; Stories from the dawn of the computer age*; Granta books; Londres; 2006.
- [4] J. Presper Eckert; Jr. James R. Weiner; H. Frazer Welsh; Herbert F. Mitchell; *The UNIVAC System; Organization of the UNIVAC System, Technology and Culture*; Marzo; 1999.
- [5] Mann, Martin; *Want to Buy a Brain?*; *Popular Science*; Mayo; 1949.
- [6] McCartney, Scott; *ENIAC; The triumphs and tragedies of the world's first computer*; Walker Publishing; Estados Unidos de América; 1999.
- [7] Museo de Informática de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València; Disponible en: <http://museo.inf.upv.es>
- [8] Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura; Disponible en: <http://www.unesco.org/new/es>
- [9] Pérez Pérez, Isaías; *Introducción a la Arqueología Informática*; Disponible en [http://www.uaeh.edu.mx/nuestro\\_alumnado/icbi/articulos/introduccion\\_arq\\_inf.pdf](http://www.uaeh.edu.mx/nuestro_alumnado/icbi/articulos/introduccion_arq_inf.pdf).
- [10] R. Williams, Michael; *History of computing technology. The institute of electrical and electronics engineers; Inc*; Second Edition; IEEE Computer Society Press; Los Alamitos; California; 1997
- [11] Remington Rand; *Introduction to Computers*. Electronic Computer Department; Estados Unidos de América; 1998.
- [12] Remington Rand; *The central computer of the UNIVAC System*; Disponible en [www.bitsavers.org/pdf/univac/univac1/UNIVAC1\\_OperMan.pdf](http://www.bitsavers.org/pdf/univac/univac1/UNIVAC1_OperMan.pdf)

- 
- [13] Remington Rand; *Basic Programming: UNIVAC I*; Disponible en: [http://bitsavers.trailing-edge.com/pdf/univac/univac1/UNIVAC1\\_Programming\\_1959.pdf](http://bitsavers.trailing-edge.com/pdf/univac/univac1/UNIVAC1_Programming_1959.pdf)
- [14] Shurkin, Joel; *Engines of the mind. The Evolution of the Computer from Mainframes to Microprocessors*; Maple-Vail Book Manufacturing Group; Estados Unidos de América; 1996.
- [15] Shurkin, Joel; *The Evolution of the Computer from Mainframes to Microprocessors*. W. W. Norton & Company; New York; 1996.
- [16] UNESCO; *Carta sobre la preservación del patrimonio digital*; Disponible en [http://portal.unesco.org/es/ev.php-URL\\_ID=17721&URL\\_DO=DO\\_TOPIC&URL\\_SECTION=201.html](http://portal.unesco.org/es/ev.php-URL_ID=17721&URL_DO=DO_TOPIC&URL_SECTION=201.html)
- [17] Wulforst, Harry; *Breakthrough to the Computer Age*. Scribners; New York; 1982.