

# Historia de las computadoras, UNIX y una comparación entre las arquitecturas de Von Neumann y Harvard

## Tema: UNIX

Estudiante	Escuela	Asignatura
Rodrigo Infanzón Acosta rinfanzona@ulasalle.edu.pe	Carrera Profesional de Ingeniería de Software	Sistemas Operativos

Informe	Tema	Duración
02	UNIX	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - B	23/08/24	29/08/24

## Índice

1. Actividades a realizar	1
2. Línea del tiempo: evolución del computador	2
3. Análisis: ¿Por qué UNIX es el sistema operativo más importante de la historia?	2
4. Modelos arquitectónicos: Von Neumann vs Harvard	3
4.1. Modelo arquitectónico Von Neumann . . . . .	3
4.1.1. Características . . . . .	3
4.1.2. Ventajas . . . . .	3
4.1.3. Desventajas . . . . .	4
4.2. Modelo arquitectónico Harvard . . . . .	4
4.2.1. Características . . . . .	4
4.2.2. Ventajas . . . . .	4
4.2.3. Desventajas . . . . .	4

## 1. Actividades a realizar

- Elaborar una línea de tiempo detallada sobre la evolución histórica de las computadoras.
- Realizar un análisis crítico del video titulado ¿Por qué UNIX es el sistema operativo más importante de la historia?. Enlace: <https://www.youtube.com/watch?v=KuGCq7L6WaM>
- Elabore una explicación de los modelos arquitectónicos de Von Neumann y Harvard, desarrollando sus características, ventajas y desventajas.

## 2. Línea del tiempo: evolución del computador

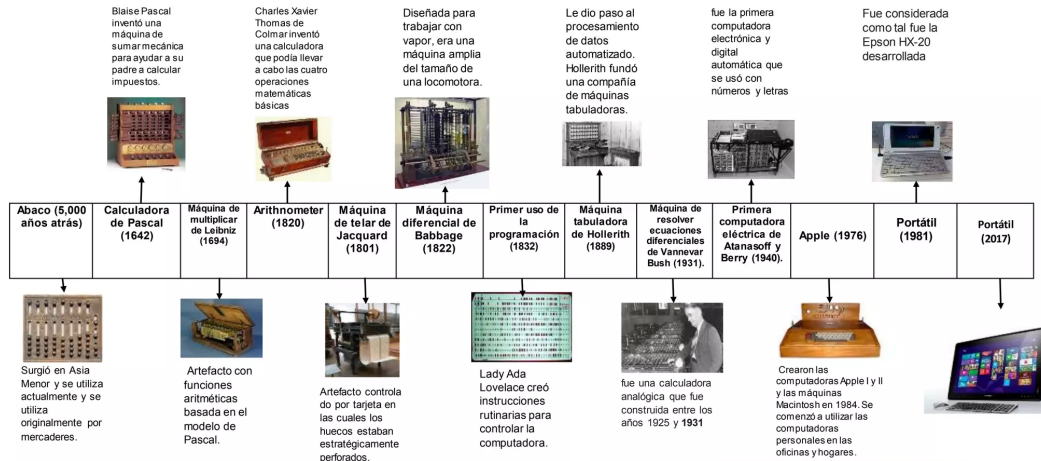


Imagen recuperada de: slideshare.net

La línea de tiempo proporciona un recorrido por los hitos clave en la evolución de los dispositivos de cálculo. Aquí una descripción de los más importantes:

- **Ábaco (5,000 años atrás)**: herramienta manual antigua para cálculos básicos, aún usada en algunos contextos.
- **Calculadora de Pascal (1642)**: dispositivo creado por Pascal para simplificar cálculos de impuestos.
- **Máquina de Leibniz (1694)**: mejoró la calculadora de Pascal, añadiendo la multiplicación.
- **Arithmometer (1820)**: primera calculadora mecánica capaz de realizar las cuatro operaciones básicas.
- **Máquina diferencial de Babbage (1822)**: intento de crear una máquina para cálculos complejos.
- **Programación (1832)**: Ada Lovelace creó las primeras instrucciones de programación.
- **Apple I (1976)**: introdujo las computadoras personales accesibles para hogares.
- **Portátil (1981)**: la Epson HX-20, primera computadora portátil, marcó el inicio de la computación móvil.
- **Portátil (2017)**: dispositivo moderno que combina potencia y portabilidad en la computación.

## 3. Análisis: ¿Por qué UNIX es el sistema operativo más importante de la historia?

UNIX es considerado el sistema operativo más influyente de la historia, según el video, por los siguientes motivos:

- **Portabilidad y flexibilidad**: UNIX se destaca por su portabilidad, permitiendo que funcione en diferentes tipos de hardware con pocos cambios. Esta flexibilidad lo convirtió en la base de muchos sistemas operativos, incluyendo Linux y macOS, y permitió su adopción en diversos entornos.

- **Filosofía de diseño:** la filosofía de UNIX, basada en la creación de herramientas pequeñas y especializadas, permitió la combinación de estas para realizar tareas complejas. Introdujo el sistema de archivos jerárquico, influyendo en otros sistemas operativos.
- **Estándares y comunidad:** UNIX fue crucial para la estandarización de los sistemas operativos, promoviendo la colaboración y el desarrollo de estándares abiertos como POSIX, lo que ha asegurado la interoperabilidad entre sistemas y fomentado el crecimiento de la informática en red.
- **Impacto duradero:** UNIX ha dejado un impacto profundo en la informática moderna, siendo la base de sistemas como Linux y macOS. Su estructura de comandos y filosofía de diseño siguen siendo fundamentales en el desarrollo y operación de software hoy en día.

## 4. Modelos arquitectónicos: Von Neumann vs Harvard

Las arquitecturas de Von Neumann y Harvard definen cómo se organizan y manejan la memoria y los datos en un sistema informático. He aquí sus definiciones:

### 4.1. Modelo arquitectónico Von Neumann

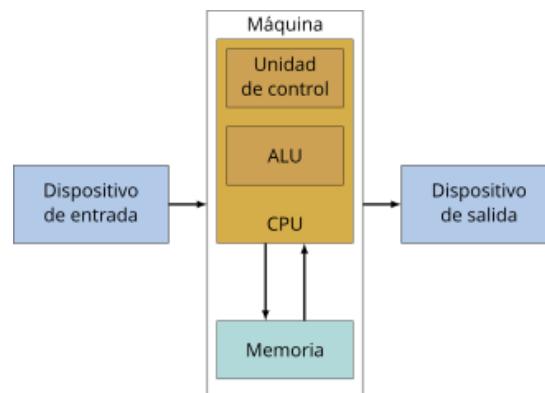


Imagen recuperada de: Wikipedia

En la arquitectura de Von Neumann, los programas y los datos se almacenan en la misma memoria física (una sola memoria única). La CPU accede a la memoria para leer y escribir tanto instrucciones como datos, utilizando el mismo bus de datos.

#### 4.1.1. Características

- **Memoria unificada:** instrucciones y datos en un solo bloque de memoria.
- **Bus de datos único:** un solo bus conecta la CPU con la memoria y dispositivos.
- **Ciclo de ejecución alternado:** la CPU alterna entre obtener instrucciones y ejecutar datos, lo que puede causar un cuello de botella.

#### 4.1.2. Ventajas

- **Simplicidad:** diseño más sencillo y económico con un solo tipo de memoria y bus.
- **Flexibilidad:** permite manipular y modificar programas y datos en la misma memoria.
- **Coste reducido:** menor costo de hardware al evitar la necesidad de memorias separadas.

#### 4.1.3. Desventajas

- **Cuello de botella:** el rendimiento se limita por el acceso secuencial a memoria para instrucciones y datos, impidiendo el acceso simultáneo.
- **Riesgo de corrupción:** compartir la misma memoria para datos e instrucciones puede causar sobrescritura accidental de instrucciones importantes.

#### 4.2. Modelo arquitectónico Harvard



Imagen recuperada de: Wikipedia

En la arquitectura de Harvard, la memoria de instrucciones y la memoria de datos están separadas. La CPU tiene dos buses diferentes, uno para acceder a las instrucciones del programa y otro para acceder a los datos.

##### 4.2.1. Características

- **Memoria separada:** instrucciones y datos se almacenan en bloques de memoria distintos.
- **Buses de datos separados:** hay un bus para instrucciones y otro para datos, permitiendo acceso simultáneo.
- **Eficiencia:** mejora el rendimiento al permitir la lectura concurrente de instrucciones y datos.

##### 4.2.2. Ventajas

- **Mayor rendimiento:** la separación de buses y memorias permite el acceso simultáneo a instrucciones y datos, reduciendo cuellos de botella.
- **Seguridad:** disminuye el riesgo de sobrescribir instrucciones al manipular datos, gracias a la separación de memorias.

##### 4.2.3. Desventajas

- **Complejidad:** la implementación es más costosa y complicada debido a la gestión de dos memorias y buses separados.
- **Menor flexibilidad:** las instrucciones y los datos tienen espacios de memoria separados, limitando su manipulación comparado con la arquitectura de Von Neumann.