PUCE Virtual

CLASE

2

Arquitectura de computadoras

Arquitectura Von Neumann y su impacto en la seguridad

QS WORLD UNIVERSITY RANKINGS

Educación de calidad

INTRODUCCIÓN DE LA CLASE

GRADO / POSGRADO / TECNOLOGÍAS

Estudia a tu tiempo
sin interrupciones

EDUCACIÓN EN LÍNEA DE CALIDAD

Cuando hablamos del funcionamiento interno de un computador, no solo nos referimos a circuitos, procesadores o memorias, sino a decisiones de diseño arquitectónico que influyen directamente en el rendimiento, la seguridad y la forma en que se ejecuta el software. En esta clase, vamos a dialogar sobre cómo la arquitectura Von Neumann, uno de los modelos más influyentes en la historia de la computación, sigue presente en los sistemas actuales y cómo sus características impactan en la ejecución de código malicioso y la protección de la información.

Además, abordaremos el funcionamiento básico del procesador o CPU, entendiendo su arquitectura interna, la forma en que ejecuta instrucciones y cómo los niveles de privilegio separan las acciones del usuario y del sistema operativo. Al finalizar esta clase, el estudiante será capaz de comprender la relación entre arquitectura, seguridad y control del hardware, fortaleciendo su base conceptual para analizar riesgos y mecanismos de protección en los sistemas computacionales modernos.

3. Arquitectura Von Neumann y su impacto en la seguridad

La arquitectura Von Neumann constituye uno de los pilares históricos de la computación moderna y sigue siendo, con variaciones, la base conceptual de la mayoría de los sistemas actuales. Este modelo surge como una propuesta para organizar de manera eficiente los componentes de un computador, permitiendo que un mismo sistema físico pueda ejecutar distintos programas sin necesidad de modificar su estructura interna. De acuerdo con los textos base de arquitectura de computadoras, este enfoque marcó el inicio de la computación de propósito general y posibilitó la rápida evolución del software (Jaramillo Villegas et al., 2022).

El principio esencial de la arquitectura Von Neumann es la utilización de una única memoria principal para almacenar tanto las instrucciones del programa como los datos que dichas instrucciones procesan. Esta característica introdujo una gran flexibilidad operativa, ya que permitió que los programas se cargaran, modificaran y ejecutaran dinámicamente. Sin embargo, desde una perspectiva contemporánea, este mismo principio representa una de las principales fuentes de riesgo en términos de seguridad informática.

La arquitectura Von Neumann permite que instrucciones y datos compartan el mismo espacio de memoria, lo que facilita la programación, pero incrementa la superficie de ataque del sistema.

Estructura funcional de la arquitectura Von Neumann

La arquitectura Von Neumann se organiza a partir de una estructura lógica compuesta por varios elementos claramente definidos. Entre ellos se encuentran la Unidad Central de Procesamiento (CPU), la memoria principal, los dispositivos de entrada y salida, y el sistema de buses que interconecta todos los componentes. Esta organización ha sido descrita y analizada en los textos fundamentales de arquitectura de computadoras utilizados en la formación universitaria (Jaramillo Villegas et al., 2022).

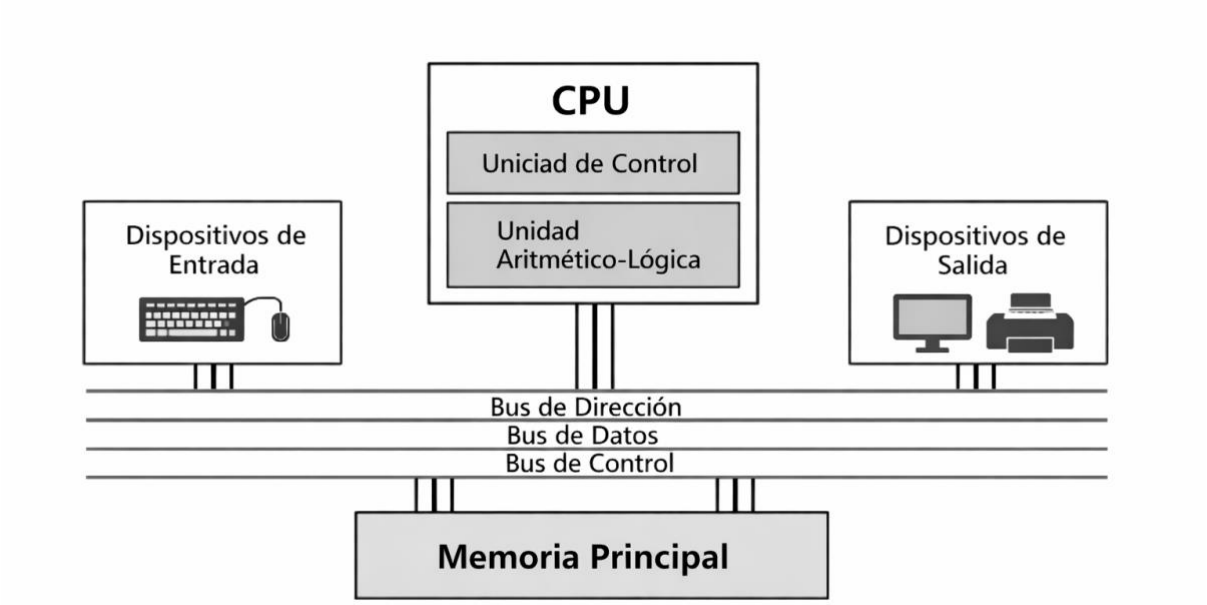
La CPU es el componente encargado de interpretar y ejecutar las instrucciones almacenadas en memoria. La memoria principal, por su parte, actúa como un repositorio único donde coexisten código y datos. El sistema de buses permite el transporte de información entre la CPU y la memoria, así como entre los dispositivos periféricos. Esta estructura simplificada fue clave para el desarrollo de sistemas eficientes y económicamente viables.

No obstante, el uso de un único espacio de memoria implica que la CPU no distingue, a nivel físico, entre lo que es una instrucción y lo que es un dato. Esta falta de diferenciación, aunque transparente para el funcionamiento normal del sistema, tiene implicaciones profundas desde el punto de vista de la seguridad.

En la Figura 1, se muestra el modelo básico de la arquitectura de Von Neumann.

Figura 1

Modelo general de la arquitectura Von Neumann



Modelo de Arquitectura Von Neumann

Autor: Milton Román, adaptado de (Stallings, 2019)

Arquitectura Von Neumann y ejecución del software

En un sistema basado en arquitectura Von Neumann, el software se ejecuta siguiendo un flujo secuencial de instrucciones que la CPU obtiene desde la memoria principal. Cada instrucción es tratada como un conjunto de bits que debe ser interpretado y ejecutado. Este enfoque permite que el programa pueda modificarse a sí mismo o cargar nuevas instrucciones durante su ejecución, lo cual fue considerado una ventaja significativa en los primeros sistemas computacionales.

El tratamiento homogéneo de código y datos es una característica clave del modelo Von Neumann y una de las causas estructurales de múltiples vulnerabilidades.

Desde el punto de vista de la seguridad, esta característica implica que cualquier error que permita escribir en una región de memoria puede ser explotado para alterar el comportamiento del programa. Los textos de evolución de las arquitecturas de computadoras destacan que muchos de los ataques clásicos a sistemas informáticos se apoyan precisamente en esta característica estructural (Ibarra Mayorga, 2009).

En el siguiente enlace se dispone de un video que indica cómo se organiza un sistema computacional y cómo estos conceptos están relacionados con la ejecución de código y la seguridad

- **Título**: “Introducción a la Arquitectura de Computadoras. Modelo Von Neumann y Componentes Básicos”
- **Descripción**: Este video educativo ofrece una explicación clara de los conceptos esenciales de la arquitectura de computadores, centrándose en el modelo de Von Neumann, los componentes estructurales de un sistema (CPU, memoria, buses) y la forma en que estos interactúan para ejecutar programas.
- **Enlace**: https://www.youtube.com/watch?v=Z2eW0nI06n8

Implicaciones de seguridad del modelo Von Neumann

En los primeros años de la computación, la seguridad no era una preocupación prioritaria. Los sistemas eran utilizados por un número reducido de usuarios en entornos controlados. Sin embargo, con la masificación del uso de computadoras y la interconexión mediante redes, las debilidades del modelo Von Neumann se hicieron evidentes.

Uno de los principales problemas es que la arquitectura no impide, por sí misma, que datos introducidos desde el exterior sean interpretados como instrucciones ejecutables. Esta situación se presenta cuando existen errores de programación, especialmente en la gestión de memoria.

La arquitectura Von Neumann depende en gran medida de la correcta gestión de la memoria por parte del software para garantizar la seguridad.

Según los textos base de la asignatura, esta dependencia del software convierte a la arquitectura en un factor de riesgo cuando se combina con prácticas de programación deficientes o con aplicaciones desarrolladas sin considerar principios de seguridad (Martínez Amador, 2012).

Ataques asociados a la arquitectura Von Neumann

Entre los ataques más comunes asociados a esta arquitectura se encuentran los desbordamientos de búfer, la inyección de código y la alteración del flujo de ejecución. Estos ataques explotan la posibilidad de sobrescribir regiones de memoria y redirigir la ejecución del programa hacia instrucciones no autorizadas.

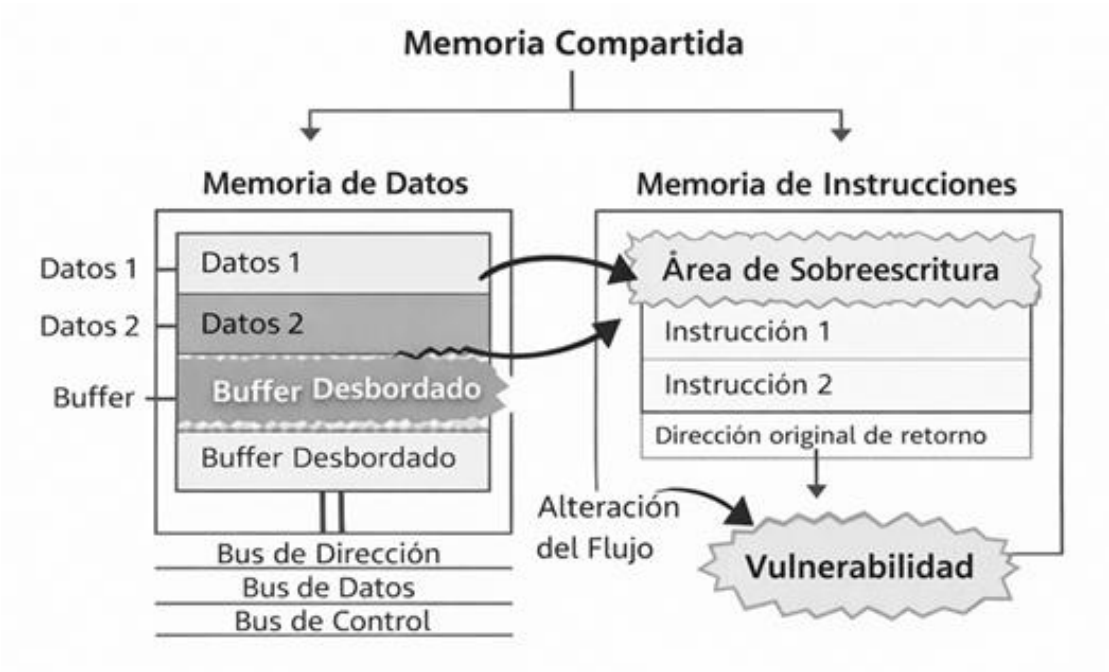
El desbordamiento de búfer ocurre cuando un programa escribe más datos de los que una región de memoria puede contener. Como consecuencia, se sobrescriben áreas adyacentes que pueden incluir direcciones de retorno o instrucciones críticas. Este tipo de ataque ha sido ampliamente documentado en estudios comparativos sobre contenidos de arquitectura y seguridad informática (Fabra Gallo et al., 2020).

Los ataques basados en corrupción de memoria son una consecuencia directa de la organización propuesta por la arquitectura Von Neumann.

Para comprender de manera visual cómo la memoria compartida puede convertirse en un punto crítico de vulnerabilidad en la arquitectura Von Neumann, en la Figura 2 se presenta un esquema que ilustra el impacto de una escritura indebida en memoria de datos sobre las zonas destinadas a instrucciones.

Figura 2

Relación entre memoria compartida y vulnerabilidades en la arquitectura Von Neumann



Autor: Milton Román

3.1 Comparación de modelos arquitectónicos y su influencia en la ejecución de código malicioso y la protección de datos

La arquitectura Von Neumann no es el único modelo existente para organizar un sistema computacional. A lo largo de la evolución de la informática han surgido otros enfoques arquitectónicos cuyo objetivo ha sido mejorar el rendimiento, la eficiencia y, especialmente, la seguridad. Comparar estos modelos permite comprender cómo las decisiones de diseño influyen en la ejecución de código malicioso y en la protección de los datos.

Arquitectura Harvard

La arquitectura Harvard se presenta como una alternativa directa al modelo Von Neumann. Su característica principal es la separación física entre la memoria de instrucciones y la memoria de datos. Esto significa que el procesador accede a dos espacios de memoria distintos, cada uno con su propio bus.

Desde el punto de vista de la seguridad, esta separación introduce una barrera significativa frente a ciertos tipos de ataques. Al no compartir el mismo espacio de memoria, resulta mucho más difícil que datos maliciosos sean ejecutados como instrucciones.

La separación física entre código y datos en la arquitectura Harvard reduce significativamente la posibilidad de ejecución de código malicioso.

Los textos de arquitectura de computadoras con enfoque en competencias destacan que este modelo es especialmente adecuado para sistemas embebidos y aplicaciones críticas, donde la seguridad y la predictibilidad son prioritarias (Martínez Amador, 2012).

La comparación entre las arquitecturas Von Neumann y Harvard permite identificar cómo la organización de la memoria y los buses incide en la flexibilidad del sistema y en su nivel de seguridad frente a la ejecución de código no autorizado, esta comparación se observa en la Tabla1.

Tabla 1

Comparación entre arquitectura Von Neumann y Harvard

	Arquitectura de Von Neumann	Arquitectura Harvard
Memoria	Memoria compartida para datos e instrucciones	Memoria separada para datos e instrucciones
Buses	Un único bus compartido para datos e instrucciones	Buses independientes para datos e instrucciones
Flexibilidad	Alta, permite modificación dinámica del código	Limitada, código almacenado en memoria fija
Protección frente a ejecución de código no autorizado	Baja, datos pueden ejecutarse como código malicioso	Alta, separación entre datos e instrucciones

Nota: Tabla comparativa que resume diferencias clave en memoria, buses, flexibilidad y nivel de protección frente a ejecución de código no autorizado.

Autor: Milton Román, adaptado de (Hennessy et al., 2019)

Arquitecturas híbridas o modificadas

En la práctica, muchos sistemas modernos utilizan arquitecturas híbridas que combinan elementos de los modelos Von Neumann y Harvard. Un ejemplo común es la utilización de memorias caché separadas para instrucciones y datos, mientras que la memoria principal sigue siendo compartida.

Este enfoque busca equilibrar flexibilidad, rendimiento y seguridad. Aunque no elimina completamente los riesgos asociados al modelo Von Neumann, sí dificulta ciertos ataques y mejora la eficiencia del procesamiento, tal como se describe en los textos sobre evolución de arquitecturas (Ibarra Mayorga, 2009).

Las arquitecturas híbridas representan un compromiso entre simplicidad, rendimiento y seguridad.

Influencia del modelo arquitectónico en la protección de datos

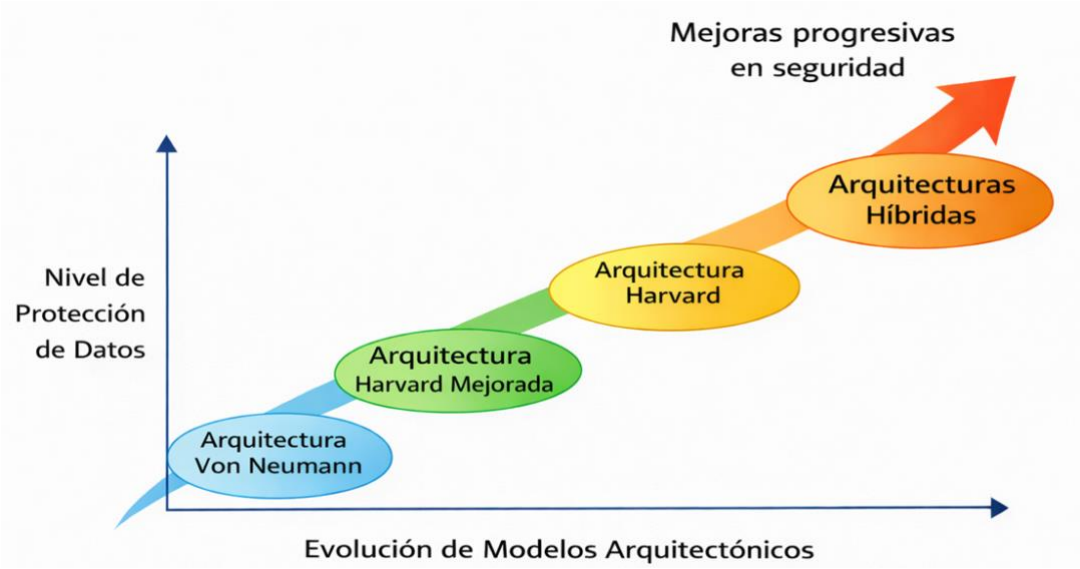
La protección de los datos no depende únicamente del software o del sistema operativo, sino también de la arquitectura subyacente. En modelos con memoria compartida, la protección se logra mediante mecanismos adicionales como paginación, segmentación y permisos de acceso. En modelos con separación física, parte de la protección está integrada en el propio hardware.

Este análisis comparativo permite al estudiante comprender que la seguridad informática no es un problema aislado, sino el resultado de múltiples decisiones de diseño que se interrelacionan.

La Figura 3, muestra una línea conceptual de la evolución desde Von Neumann hasta arquitecturas híbridas, destacando mejoras progresivas en protección de datos.

Figura 3

Evolución de modelos arquitectónicos y su relación con la seguridad.



Autor: Milton Román

4. Funcionamiento del procesador (CPU)

El procesador o Unidad Central de Procesamiento (CPU) es el componente encargado de ejecutar las instrucciones de los programas y coordinar el funcionamiento general del sistema. Comprender su funcionamiento básico es esencial para analizar cómo se ejecuta el software y cómo se implementan mecanismos de control y seguridad.

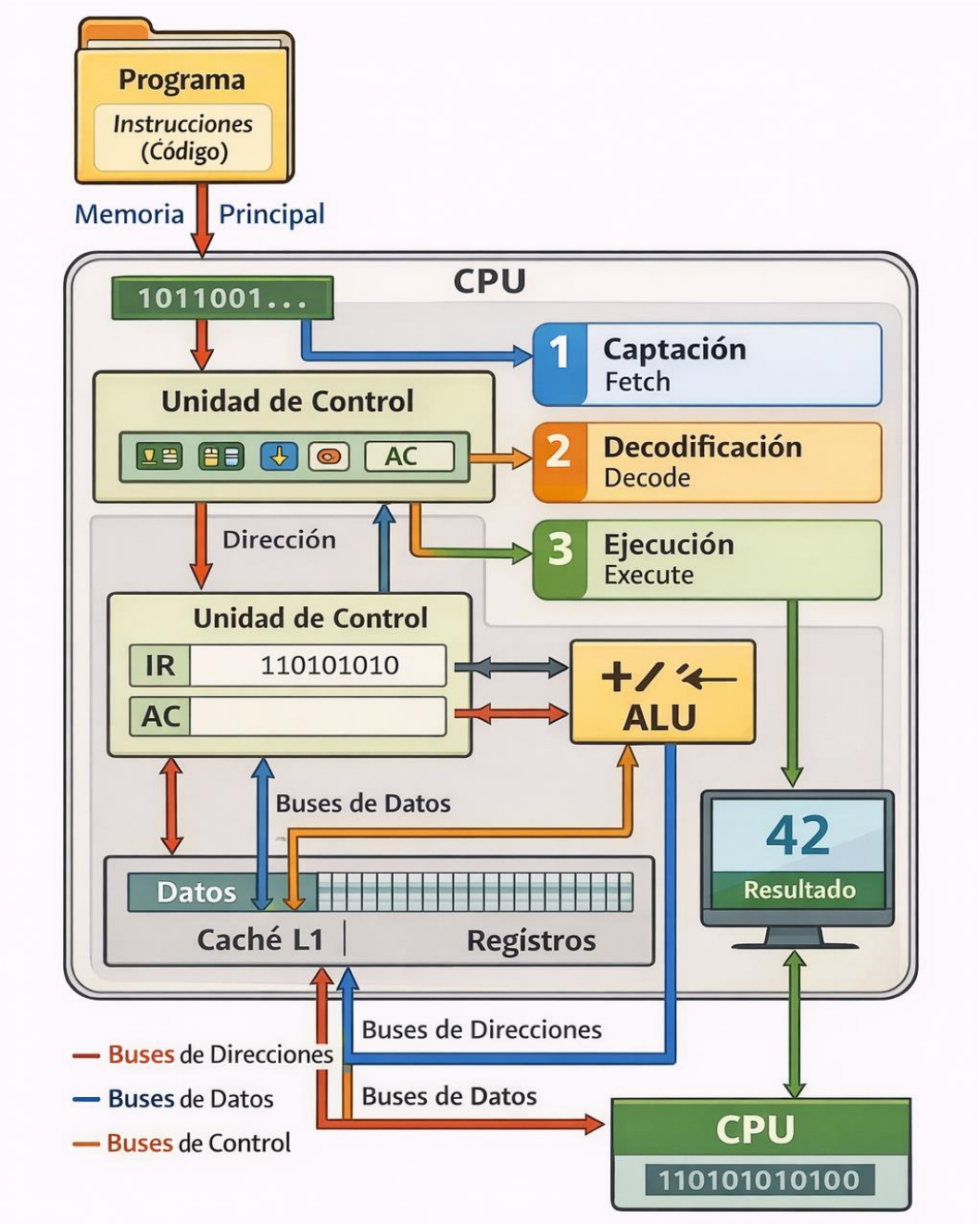
Desde una perspectiva introductoria, la CPU puede entenderse como un sistema secuencial que procesa instrucciones una tras otra, transformando datos de entrada en resultados de salida. Este comportamiento ha sido ampliamente descrito en los textos fundamentales de arquitectura de computadoras utilizados en el sílabo (Jaramillo Villegas et al., 2022).

La CPU es el núcleo operativo del sistema computacional y el punto donde convergen arquitectura, rendimiento y seguridad.

La Figura 4 permite visualizar cómo fluye la información desde el programa hasta la obtención del resultado, destacando el papel central del CPU en la ejecución de instrucciones.

Figura 4

Funcionamiento del computador



Autor: Milton Román

4.1 Arquitectura interna del CPU

La arquitectura interna del CPU está compuesta por varios bloques funcionales que trabajan de manera coordinada. Entre los más importantes se encuentran la Unidad de Control, la Unidad Aritmético-Lógica, los registros y los mecanismos de almacenamiento interno.

La Unidad de Control es responsable de interpretar las instrucciones y generar las señales necesarias para que los demás componentes actúen de forma sincronizada. La Unidad Aritmético-Lógica realiza las operaciones matemáticas y lógicas fundamentales. Los registros almacenan temporalmente datos e instrucciones, permitiendo un acceso extremadamente rápido.

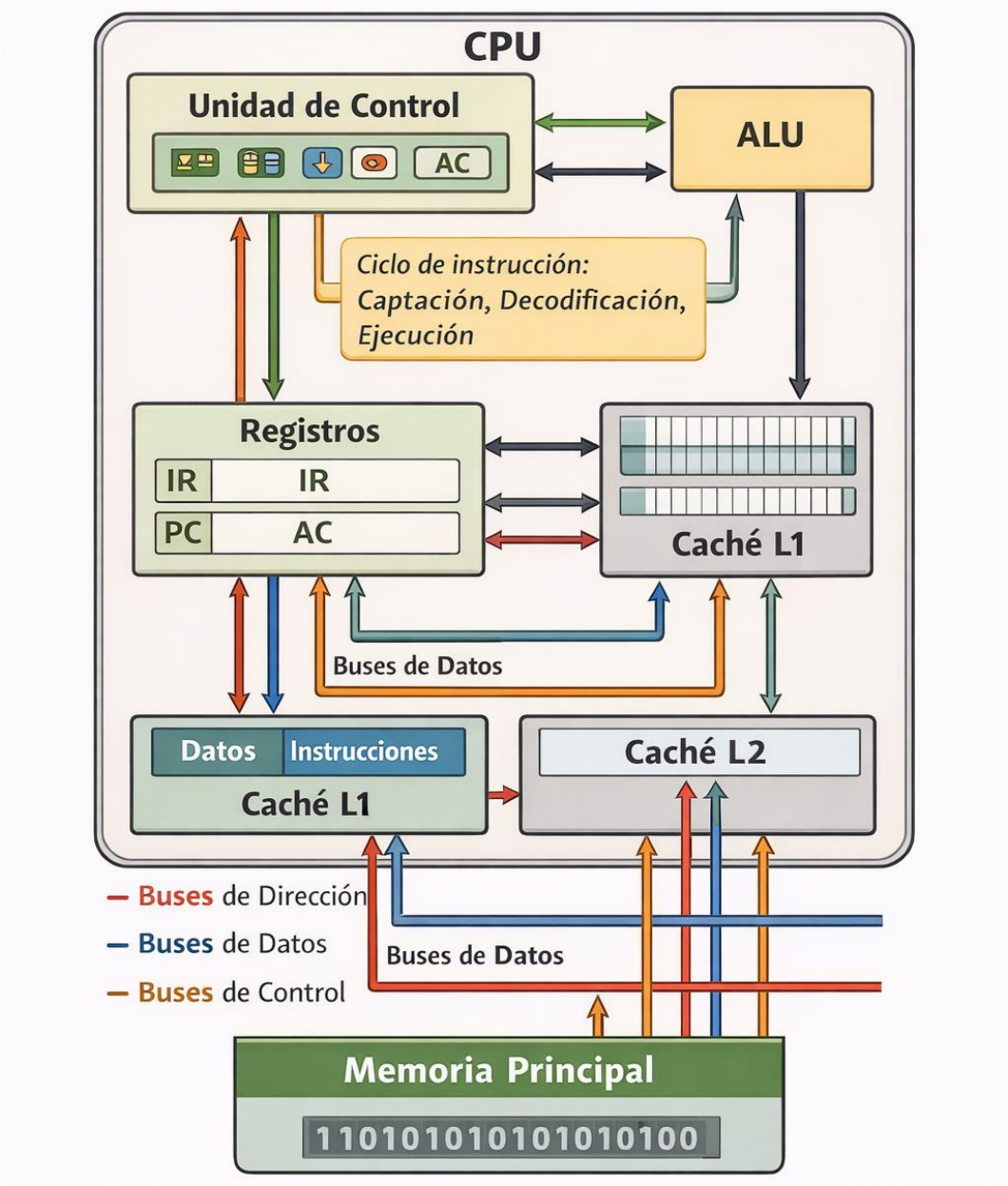
Los registros del CPU son esenciales para el rendimiento y la correcta ejecución de las instrucciones.

Los textos base destacan que la organización interna del procesador influye directamente en la forma en que se implementan mecanismos de control y protección, ya que ciertas instrucciones solo pueden ejecutarse bajo condiciones específicas (Jaramillo Villegas et al., 2022).

La Figura 5 muestra los principales componentes internos del procesador: unidad de control, ALU, registros y memorias caché, y su interconexión durante la ejecución de instrucciones.

Figura 5

Arquitectura interna del computador



Autor: Milton Román (adaptado de Stallings 2019)

4.2 Ejecución de instrucciones y niveles de privilegio (usuario y kernel)

La ejecución de instrucciones en la CPU sigue un proceso cíclico conocido como ciclo de instrucción, que incluye la captación, decodificación, ejecución y escritura de resultados. Este ciclo se repite millones de veces por segundo y constituye la base del funcionamiento del sistema.

Desde el punto de vista de la seguridad, uno de los conceptos más importantes asociados a la ejecución de instrucciones es el de niveles de privilegio. Estos niveles determinan qué instrucciones puede ejecutar un programa y a qué recursos del sistema puede acceder.

Los niveles de privilegio permiten separar las acciones del usuario de las funciones críticas del sistema operativo.

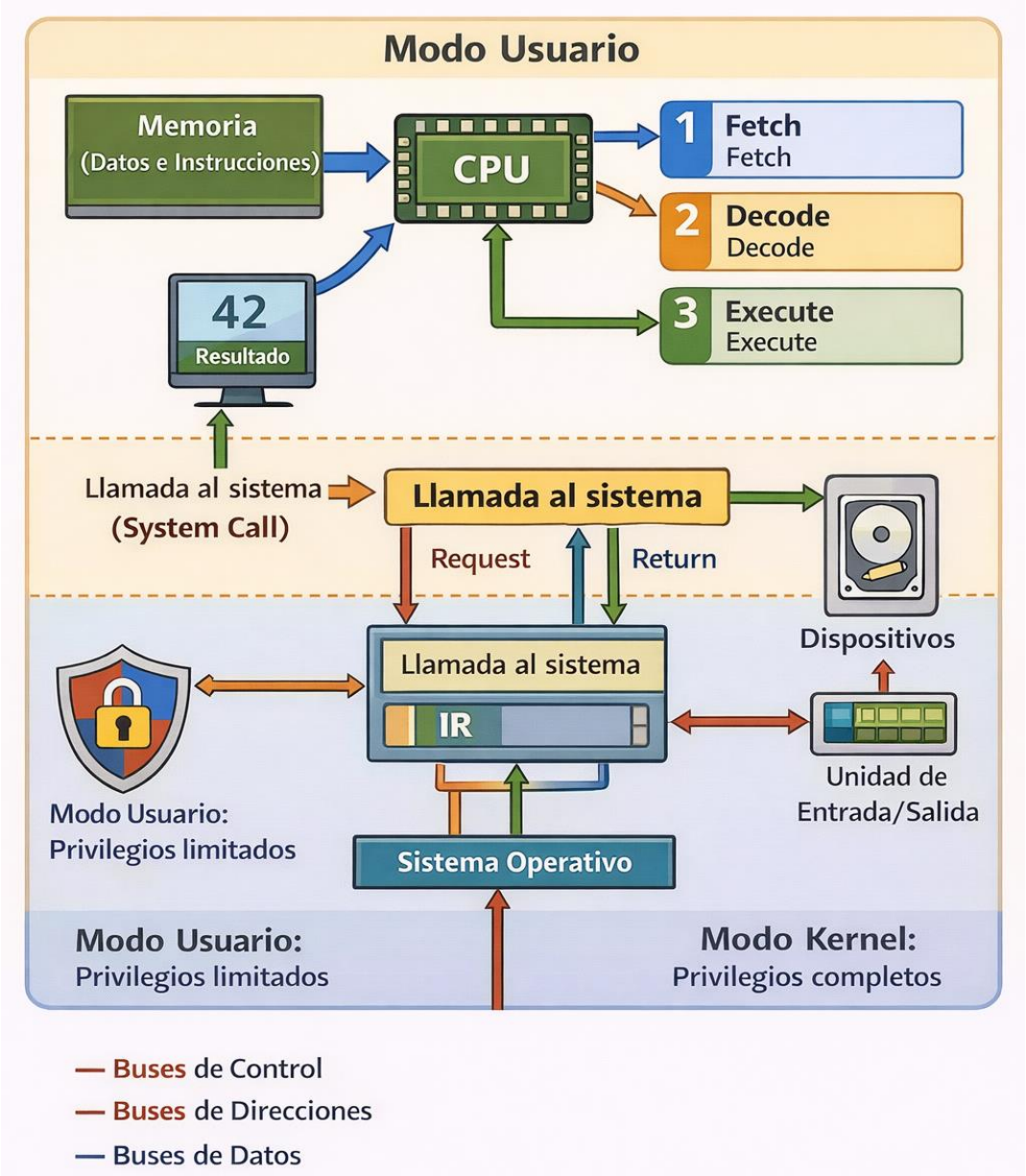
En términos generales, se distinguen dos niveles básicos: el modo usuario y el modo kernel. El modo usuario está destinado a la ejecución de aplicaciones comunes y tiene acceso restringido al hardware. El modo kernel, en cambio, está reservado para el sistema operativo y permite un control total de los recursos.

Esta separación es fundamental para evitar que programas maliciosos comprometan el sistema completo, tal como se analiza en los textos de arquitectura y sistemas operativos incluidos en la bibliografía del sílabo (Fabra Gallo et al., 2020).

La Figura 6 representa el ciclo de ejecución de instrucciones y la separación entre modo usuario y modo kernel, destacando las transiciones controladas mediante llamadas al sistema.

Figura 6

Ejecución de instrucciones y niveles de privilegio



Autor: Milton Román (adaptado de Hennessy y Patterson (2019))

Para reforzar el conocimiento sobre los conceptos que se vinculan con la seguridad del sistema, como la separación de tareas y la importancia de entender los privilegios de ejecución, te invito a mirar el siguiente video:

- **Título:** Ciclo de Instrucción del Procesador | Arquitectura y Funcionamiento Interno de la CPU.
- **Descripción:** Este video educativo explica cómo la CPU ejecuta instrucciones paso a paso, describiendo el ciclo de instrucción (fetch-decode-execute) y la interacción entre los distintos componentes internos del procesador.
- **Enlace:** Ejecución de instrucciones | Arquitectura Von Neumann | CPU: El procesador

Glosario:

Fabra Gallo, J. M., Martínez Márquez, Y., & Valcárcel Izquierdo, N. (2020). Estudio comparado de los contenidos de la asignatura Arquitectura. *13*. https://doi.org/ISSN-e 2306-2495

Hennessy, Hennessy, J. L., & Patterson, D. A. (2019). *ARQUITECTURA DE COMPUTADORES: Un enfoque cuantitativo.* McGrawHill. https://doi.org/1-55860-069-8

Ibarra Mayorga, M. F. (2009). *Evolución de las arquitecturas de las computadoras.* eLibro. Obtenido de https://elibro.puce.elogim.com/es/lc/puce/titulos/28671

Jaramillo Villegas, J. A., Zuluaga Bucheli, H. M., & Sepúlveda Caviedes, C. (2022). *Arquitectura de Computadoras con RISC-V.* Universidad Tecnológica de Pereira. https://doi.org/9789587227956

Martínez Amador, H. (2012). *Arquitectura de computadoras: basado en competencias para nivel superior.* Obtenido de https://elibro.puce.elogim.com/es/lc/puce/titulos/130397

Stallings, W. (2019). *Computer organization and architecture : designing for performance* (11ava ed.). Pearson Education. Obtenido de https://os.ecci.ucr.ac.cr/ci0114/material/Stallings/Computer-Organization-Architecture-11th.pdf

Definiciones de términos destacados:

- **Arquitectura Von Neumann**: Modelo computacional que utiliza una memoria única para almacenar datos e instrucciones, permitiendo la ejecución flexible de programas.
- **Niveles de privilegio**: Mecanismo de control que restringe la ejecución de instrucciones y el acceso a recursos según el contexto del programa.

Recursos de profundización:

Recurso 1:

- **Título:** Banco de preguntas comentadas sobre arquitectura de computadoras, CPU y fundamentos de seguridad

- **Descripción**: Este recurso está orientado a reforzar la comprensión de los conceptos fundamentales relacionados con la arquitectura Von Neumann, la comparación de modelos arquitectónicos, el funcionamiento interno del CPU y los niveles de privilegio (usuario y kernel). Cada pregunta incluye su respuesta correcta acompañada de una justificación técnica, lo que permite al estudiante reflexionar sobre la relación entre el diseño arquitectónico del sistema y los principios básicos de seguridad informática.
- **Documento**: Recurso de profundización 1 Banco de preguntas.docx

Recurso 2:

- **Título:** Cómo la arquitectura del computador ayuda a proteger el sistema
- **Descripción:** Ejercicio que permite comprender, de manera sencilla, cómo la arquitectura del computador, el funcionamiento básico del CPU y los niveles de privilegio ayudan a proteger un sistema frente a errores de los programas. No se requiere conocimiento previo de programación.
- **Documento:** Recurso de profundización 2 Ejercicio guiado.docx

Actividad de evaluación:

PREGUNTAS DE AUTOEVALUACIÓN

INSTRUCCIONES	Lea atentamente cada pregunta y seleccione la alternativa correcta. Al finalizar cada respuesta, revise la retroalimentación proporcionada para reforzar su comprensión sobre la arquitectura de computadoras, el funcionamiento del procesador y su impacto en la seguridad del sistema.					
Titulo	Arquitectura de Computadores y Seguridad: Evaluación conceptual					
Dificultad		Baja	**X**	Media		Alta
RDAs Evaluados:	**X**	RDA 1				
		RDA 2				
		RDA 3				
Secciones evaluadas		3. Arquitectura Von Neumann y su impacto en la seguridad 3.1. Comparación de modelos arquitectónicos y cómo influyen en la ejecución de código malicioso y en la protección de datos. 4. Funcionamiento del procesador (CPU) 4.1. Arquitectura interna del CPU 4.2. Ejecución de instrucciones y niveles de privilegio (usuario y kernel) desde una perspectiva básica.				
PREGUNTA 1	¿Cuál es la principal implicación de seguridad de la arquitectura Von Neumann?					
CORRECTA	La memoria compartida entre datos e instrucciones puede facilitar la ejecución de código malicioso.					
Incorrecta 1	La arquitectura Von Neumann impide que los datos sean modificados durante la ejecución.					

Incorrecta 2	La separación física de memorias elimina completamente las vulnerabilidades.
Incorrecta 3	El uso de buses independientes garantiza mayor flexibilidad y seguridad.
Retroalimentación de las respuestas	
Correcta	En la arquitectura Von Neumann, datos e instrucciones comparten el mismo espacio de memoria, lo que puede permitir que datos manipulados sean ejecutados como código, facilitando ataques como los desbordamientos de búfer (Stallings, 2016).
Incorrecta 1	Los datos sí pueden modificarse durante la ejecución, lo que precisamente genera riesgos de seguridad.
Incorrecta 2	La separación física corresponde a la arquitectura Harvard, no a Von Neumann.
Incorrecta 3	El uso de buses independientes no es una característica del modelo Von Neumann.
PREGUNTA 2	¿Cuál es una ventaja de seguridad de la arquitectura Harvard frente a la Von Neumann?
CORRECTA	La separación entre memoria de datos y memoria de instrucciones dificulta la ejecución de código no autorizado.
Incorrecta 1	Permite mayor flexibilidad para ejecutar cualquier tipo de programa.
Incorrecta 2	Reduce el uso de registros internos del CPU.
Incorrecta 3	Elimina la necesidad de mecanismos de control de acceso.
Retroalimentación de las respuestas	
Correcta	La arquitectura Harvard separa físicamente datos e instrucciones, reduciendo la posibilidad de que datos maliciosos sean ejecutados como código (Hennessy & Patterson, 2019).
Incorrecta 1	La flexibilidad es mayor en Von Neumann, no en Harvard.
Incorrecta 2	El uso de registros no depende del modelo arquitectónico.
Incorrecta 3	Los mecanismos de control siguen siendo necesarios en cualquier arquitectura.
PREGUNTA 3	¿Cuál de los siguientes componentes del CPU coordina la ejecución de las instrucciones?
CORRECTA	La unidad de control.
Incorrecta 1	La memoria caché.
Incorrecta 2	La unidad aritmético-lógica (ALU).
Incorrecta 3	Los dispositivos de entrada/salida.

Retroalimentación de las respuestas	
Correcta	La unidad de control interpreta las instrucciones y genera las señales necesarias para coordinar el funcionamiento del CPU (Stallings, 2016).
Incorrecta 1	La memoria caché acelera el acceso a datos, pero no coordina la ejecución.
Incorrecta 2	La ALU ejecuta operaciones, pero no controla el flujo de instrucciones.
Incorrecta 3	Los dispositivos de entrada/salida no forman parte del núcleo del CPU.
PREGUNTA 4	¿Cuál es la función principal de los niveles de privilegio en un sistema computacional?
CORRECTA	Restringir el acceso a recursos críticos según el tipo de código que se ejecuta.
Incorrecta	Aumentar la velocidad de ejecución de las aplicaciones.
Incorrecta	Permitir que todos los programas accedan directamente al hardware.
Incorrecta	Eliminar la necesidad del sistema operativo.
Retroalimentación de las respuestas	
Correcta	Los niveles de privilegio separan la ejecución en modo usuario y modo kernel, evitando que aplicaciones comunes accedan directamente a recursos críticos del sistema (Hennessy & Patterson, 2019).
Incorrecta 1	La velocidad no es el objetivo principal de los privilegios.
Incorrecta 2	Permitir acceso directo al hardware incrementaría los riesgos de seguridad.
Incorrecta 3	El sistema operativo es fundamental para gestionar los privilegios.

PUCE Virtual

síguenos en:

www.pucevirtual.puce.edu.ec