PUCE Virtual

CLASE
6

Arquitectura de computadoras

Arquitectura de entrada / salida (I / O)

QS WORLD UNIVERSITY RANKINGS

Educación de calidad

INTRODUCCIÓN DE LA CLASE

GRADO / POSGRADO / TECNOLOGÍAS

Estudia a tu tiempo
son interrupciones

EDUCACIÓN EN LÍNEA DE CALIDAD

El contenido a estudiar en esta clase presenta una propuesta formativa orientada al análisis de la seguridad en la arquitectura de computadores, con énfasis en los subsistemas de entrada/salida (I/O) y el Acceso Directo a Memoria (DMA). A través de ejercicios prácticos, estudios de caso y actividades de evaluación, se busca que el estudiante comprenda cómo los componentes de hardware y los mecanismos de comunicación con dispositivos externos pueden convertirse en vectores de ataque si no se gestionan adecuadamente. Este enfoque permite trascender la visión tradicional centrada únicamente en el rendimiento, integrando principios de ciberseguridad desde la base arquitectónica del sistema.

Además, el documento promueve el desarrollo de habilidades analíticas mediante la resolución de escenarios reales, como ataques tipo BadUSB, accesos indebidos a memoria mediante DMA y compromisos en firmware. En este contexto, el estudiante no solo identifica vulnerabilidades, sino que también diseña estrategias de mitigación y fortalece la comprensión de la cadena de confianza del sistema. De esta manera, el material contribuye a la formación de profesionales capaces de evaluar riesgos, tomar decisiones informadas y diseñar soluciones seguras y eficientes en entornos tecnológicos actuales.

RDA 3: Diseñar soluciones eficientes en términos de consumo de recursos para diferentes tipos de aplicaciones.

Reto # 4.

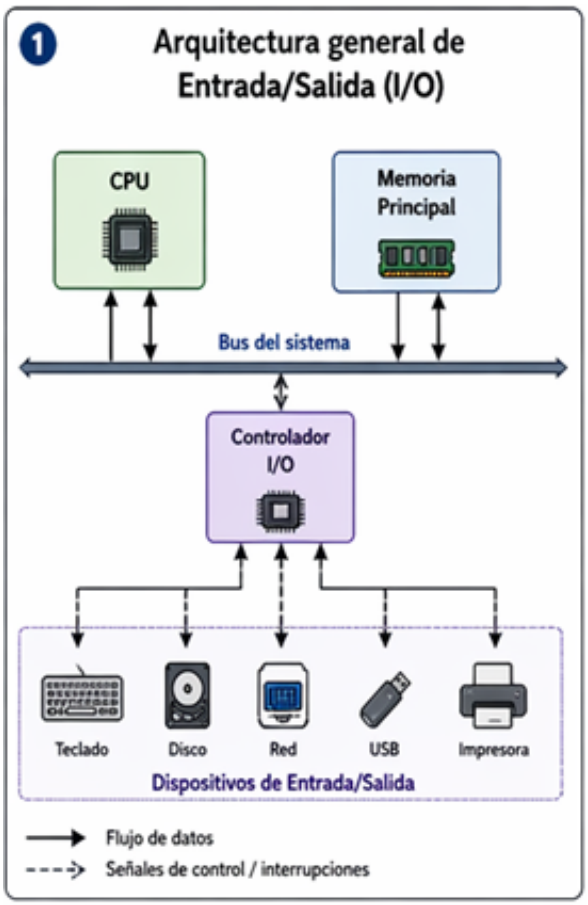
9. ARQUITECTURA DE ENTRADA/SALIDA (I/O)

La arquitectura de entrada y salida (I/O, Input/Output) es un componente esencial en los sistemas computacionales, ya que permite la comunicación entre la CPU, la memoria y los dispositivos externos. Este subsistema no solo facilita la transferencia de datos, sino que también influye directamente en el rendimiento y la seguridad del sistema. Según Stallings (2019), el diseño de los mecanismos de I/O es determinante para lograr un equilibrio entre eficiencia y control en la ejecución de procesos.

La figura 1 representa la estructura básica del sistema. Ilustra el flujo de datos y señales de control entre los componentes.

Figura 1

Arquitectura general de Entrada/Salida (I/O)



Autor: Milton Román

Desde una perspectiva funcional, la arquitectura I/O se encarga de coordinar la interacción con periféricos como teclados, discos, redes y dispositivos USB. Esta interacción se realiza a través de controladores y buses que gestionan tanto el flujo de datos como la sincronización de operaciones. Hennessy y Patterson (2019) destacan que un diseño adecuado de I/O permite reducir cuellos de botella, optimizando el uso de los recursos del sistema, lo cual se relaciona directamente con el RDA 3.

9.1 Comunicación del sistema con dispositivos externos

La comunicación con dispositivos externos se implementa mediante distintos mecanismos que impactan tanto en el rendimiento como en la seguridad:

a) **Entrada/Salida programada:** En este enfoque, la CPU controla directamente el estado del dispositivo mediante consultas constantes. Aunque es sencillo, resulta ineficiente porque consume ciclos de procesamiento innecesarios, afectando el rendimiento global (Stallings, 2019).
b) **Entrada/Salida por interrupciones:** Permite que el dispositivo notifique a la CPU cuando está listo para transferir datos. Esto mejora la eficiencia, ya que la CPU puede

ejecutar otras tareas mientras espera, optimizando el uso del procesador (Hennessy et al., 2019).

c) **Acceso Directo a Memoria (Direct Memory Access - DMA):** Este mecanismo permite que los dispositivos transfieran datos directamente a la memoria sin intervención constante de la CPU. Su uso es clave en sistemas modernos porque reduce la carga de procesamiento y mejora el rendimiento, alineándose con el diseño eficiente exigido por el RDA 3.

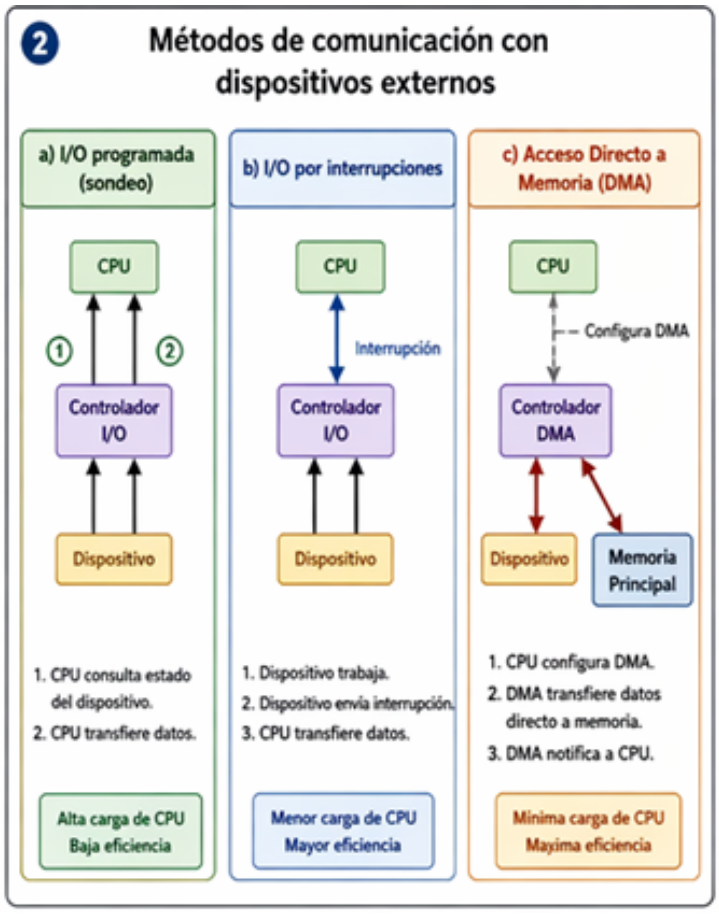
Estos mecanismos evidencian cómo la arquitectura I/O no solo tiene implicaciones técnicas, sino también estratégicas en el diseño de sistemas eficientes y seguros.

La figura 2 compara tres mecanismos:

1. I/O programada (alta carga de CPU)
2. I/O por interrupciones (eficiencia media)
3. DMA (alta eficiencia). Permite visualizar cómo cambia el rol de la CPU en cada método.

Figura 2

Métodos de comunicación con dispositivos externos



Autor: Milton Román

Riesgos de seguridad asociados a periféricos

En el ámbito de la ciberseguridad, los dispositivos de entrada y salida representan una superficie de ataque relevante. Su conexión directa con el sistema los convierte en posibles vectores de amenazas que pueden comprometer la integridad del sistema.

Entre los principales riesgos se destacan:

a) **Dispositivos USB maliciosos**: pueden contener malware que se ejecuta automáticamente al conectarse, aprovechando la confianza del sistema.

b) **Manipulación de firmware**: los periféricos incluyen firmware que puede ser alterado para ejecutar código malicioso a bajo nivel, dificultando su detección (Martínez Amador, 2012).

c) **Ataques DMA**: permiten acceder directamente a la memoria del sistema, evitando controles del sistema operativo.

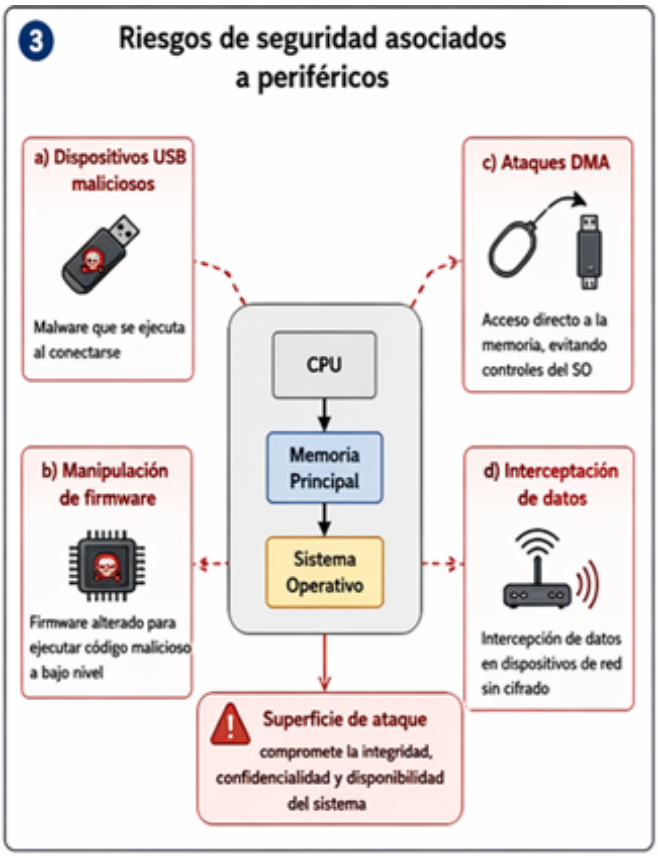
d) **Interceptación de datos**: especialmente en dispositivos de red, donde la falta de cifrado puede exponer información sensible.

Estos riesgos evidencian la necesidad de integrar la seguridad desde el diseño de la arquitectura, especialmente en entornos donde los dispositivos externos son ampliamente utilizados.

La figura 3 muestra los principales vectores de ataque y destaca cómo los periféricos pueden comprometer el sistema.

Figura 3

Riesgos de seguridad asociados a periféricos



Autor: Milton Román

Relación con BIOS/UEFI (Interfaz de Firmware Extensible Unificada) y el proceso de arranque

El proceso de arranque del sistema constituye la base sobre la cual se establece la confianza en la ejecución del software. Tradicionalmente, los sistemas utilizaban BIOS, pero este modelo presenta limitaciones en términos de seguridad. La evolución hacia UEFI ha permitido incorporar mecanismos más robustos, mejorando la protección del sistema desde sus etapas iniciales (Ibarra Mayorga, 2009).

En este contexto, destacan dos mecanismos clave:

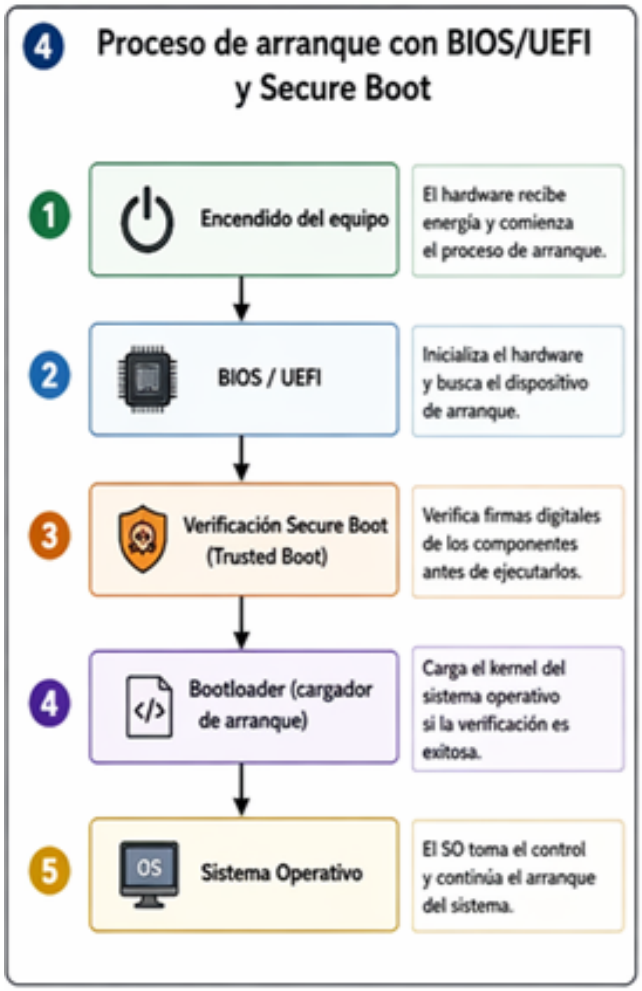
- **Secure Boot**: garantiza que solo se ejecuten componentes firmados digitalmente durante el arranque.
- **Trusted Boot**: verifica la integridad de cada componente cargado en el sistema.

Estos mecanismos son fundamentales para prevenir la ejecución de código malicioso, especialmente en entornos donde los dispositivos I/O pueden ser utilizados como vectores de ataque.

La figura 4 describe el flujo de arranque e incluye controles de verificación de integridad.

Figura 4

Proceso de arranque con BIOS/UEFI y Secure Boot



Autor: Milton Román

Cadena de confianza

La cadena de confianza se basa en la idea de que cada componente del sistema debe verificar la integridad del siguiente antes de ejecutarlo. Este proceso inicia en el firmware (BIOS/UEFI) y continúa hasta el sistema operativo. La cadena de confianza protege el sistema desde el arranque hasta las aplicaciones, integrando seguridad en todos los niveles:

1. **Firmware (UEFI):** uso de Secure Boot para evitar la ejecución de software no autorizado.
2. **Hardware:** incorporación de TPM para garantizar integridad y gestión segura de claves.
3. **Sistema operativo:** control de acceso a dispositivos y monitoreo de actividades.
4. **Aplicaciones:** validación de datos provenientes de periféricos.

Además, se aplican medidas de protección clave en la arquitectura I/O, como:

- Control y restricción de dispositivos
- Actualización segura de firmware
- Uso de cifrado en comunicaciones
- Monitoreo y auditoría de eventos
- Segmentación de hardware

En conjunto, estas acciones permiten reducir riesgos de seguridad sin afectar el rendimiento, cumpliendo con el enfoque de ciberseguridad y el RDA 3.

El enfoque de ciberseguridad en la arquitectura de entrada y salida (I/O) considera a los dispositivos externos como posibles puntos de vulnerabilidad, por lo que integra mecanismos de protección desde el diseño del sistema. Este enfoque no solo busca prevenir ataques, sino también optimizar el uso de recursos. El uso de técnicas como el DMA y el manejo de interrupciones permite mejorar la eficiencia del sistema, reduciendo la carga de la CPU y aumentando el rendimiento (Hennessy et al., 2019). A su vez, la incorporación de mecanismos de seguridad desde el hardware disminuye la dependencia de soluciones complejas en software, lo que contribuye a un menor consumo de recursos.

Según Fabra Gallo et al. (2020), la enseñanza de la arquitectura de computadoras debe integrar seguridad y eficiencia para responder a los retos tecnológicos actuales. En este contexto, la arquitectura I/O resulta clave tanto para el rendimiento como para la protección del sistema, ya que la interacción con dispositivos externos introduce riesgos que deben ser gestionados adecuadamente. El uso de tecnologías como Secure Boot y Trusted Boot permite establecer una cadena de confianza desde el arranque, garantizando la integridad del sistema. En conjunto, este enfoque favorece el desarrollo de sistemas seguros, eficientes y resilientes.

Para reforzar el conocimiento sobre los dispositivos den entrada y salida y su funcionamiento, ingresa al enlace del video con las siguientes características:

- **Título**: Entradas/Salidas: Arquitectura de computadoras 2
- **Descripción**: Explica cómo la CPU se comunica con los dispositivos externos. Introduce la interfaz de entrada/salida y su relación con el rendimiento, permite inferir riesgos al mostrar cómo interactúan los periféricos con el sistema.
- Enlace: Entradas y salidas (E/S) | Arquitectura de Computadores 2

10. ACCESO DIRECTO A MEMORIA

El DMA es un mecanismo fundamental dentro de la arquitectura de computadoras que permite a los dispositivos periféricos transferir datos directamente hacia o desde la memoria principal sin requerir la intervención constante de la CPU. Este enfoque representa un avance significativo en términos de eficiencia, ya que libera al procesador de tareas repetitivas relacionadas con la transferencia de datos, permitiéndole concentrarse en procesos de mayor complejidad.

Desde una perspectiva arquitectónica, el DMA se integra como un controlador especializado que gestiona las transferencias entre dispositivos y memoria. Según Stallings (2019), este mecanismo es esencial en sistemas modernos debido a la creciente demanda de procesamiento de datos provenientes de dispositivos externos, como discos duros, tarjetas de red y dispositivos multimedia. Sin DMA, la CPU se convertiría en un cuello de botella, afectando el rendimiento global del sistema.

El uso del DMA está estrechamente relacionado con el diseño eficiente de sistemas, lo cual se vincula directamente con el RDA 3, que plantea la necesidad de optimizar el consumo de recursos. Hennessy y Patterson (2019) destacan que la eficiencia en la transferencia de datos es un factor clave en el rendimiento de aplicaciones intensivas, como sistemas de bases de datos, procesamiento multimedia y entornos de virtualización.

10.1 Funcionamiento del Acceso Directo a Memoria

El funcionamiento del DMA se basa en la delegación de la transferencia de datos a un controlador especializado, lo que permite reducir la carga de trabajo de la CPU. Este proceso puede entenderse a través de una secuencia estructurada:

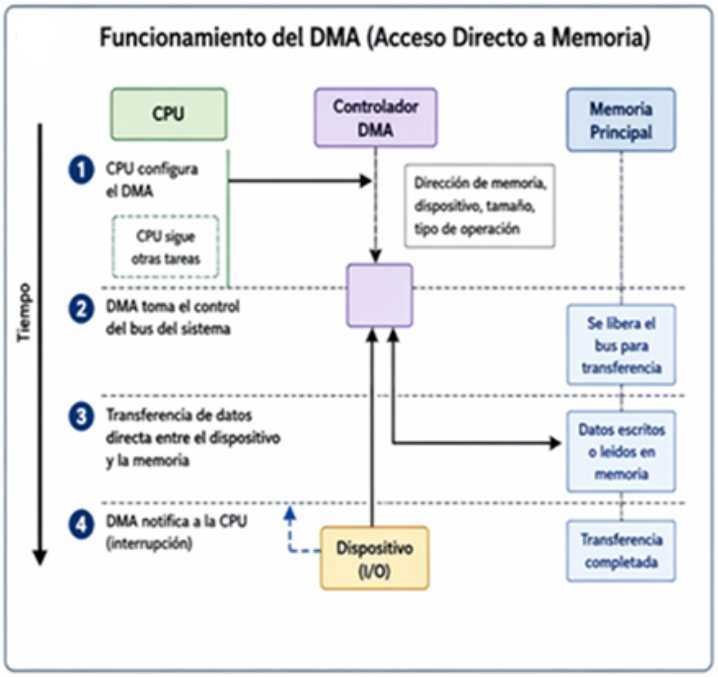
1. **Inicialización por parte de la CPU**
   La CPU configura el controlador DMA indicando:
   - La dirección de memoria de origen o destino.
   - El dispositivo involucrado.
   - La cantidad de datos a transferir.
   - El tipo de operación (lectura o escritura).
2. **Transferencia de datos**
   Una vez configurado, el controlador DMA toma el control del bus del sistema y realiza la transferencia directamente entre el dispositivo y la memoria. Durante este proceso, la CPU puede continuar ejecutando otras tareas, lo que mejora la eficiencia del sistema.
3. **Notificación de finalización**
   Al concluir la transferencia, el controlador DMA envía una interrupción a la CPU para indicar que la operación ha finalizado.

La figura 5 explica el proceso y resalta la eficiencia del mecanismo:

1. CPU configura DMA
2. DMA controla el bus
3. Transferencia directa memoria-dispositivo
4. Interrupción a CPU

Figura 5

Funcionamiento del DMA



Autor: Milton Román

Este modelo permite una gestión más eficiente de los recursos, ya que evita el uso innecesario del procesador en tareas de bajo nivel. Según Stallings (2019), existen diferentes modos de operación del DMA, como el modo ráfaga y el modo ciclo robado, cada uno con implicaciones específicas en el rendimiento del sistema.

Relación del DMA con la ciberseguridad

Aunque el DMA ofrece ventajas significativas en términos de eficiencia, también introduce riesgos importantes desde el punto de vista de la ciberseguridad. Su capacidad para acceder directamente a la memoria del sistema lo convierte en un vector potencial de ataques.

Entre los principales riesgos asociados al DMA se pueden identificar:

- **Acceso no autorizado a memoria:** Un atacante podría utilizar un dispositivo con capacidades DMA para leer información sensible directamente desde la memoria, sin pasar por los controles del sistema operativo.
- **Manipulación de datos:** La escritura directa en memoria permite modificar información crítica, lo que podría comprometer la integridad del sistema.
- **Ataques persistentes:** El uso de dispositivos comprometidos puede permitir la ejecución de código malicioso de forma persistente, incluso después de reinicios.

Martínez Amador (2012) señala que los ataques a bajo nivel son especialmente peligrosos porque operan por debajo de los mecanismos tradicionales de seguridad, lo que dificulta su detección.

Relación con BIOS/UEFI y el proceso de arranque

El análisis del DMA no puede separarse del proceso de arranque del sistema, especialmente en el contexto del Reto 3. Durante el arranque, el firmware (BIOS o UEFI) inicializa los dispositivos de hardware, incluyendo aquellos con capacidades DMA. Si estos dispositivos no son verificados adecuadamente, pueden convertirse en una amenaza desde las primeras etapas del sistema.

La evolución hacia UEFI ha permitido incorporar mecanismos de seguridad más avanzados, como Secure Boot, que garantiza la ejecución únicamente de componentes firmados digitalmente. Ibarra Mayorga (2009) destaca que este tipo de mecanismos es fundamental para establecer una base segura en el sistema.

En este contexto, el concepto de cadena de confianza cobra especial relevancia, ya que permite asegurar que cada componente del sistema, incluyendo los controladores DMA, sea validado antes de su ejecución.

Para mitigar los riesgos asociados al DMA y fortalecer la seguridad del sistema, se propone una cadena de confianza que integre los siguientes niveles:

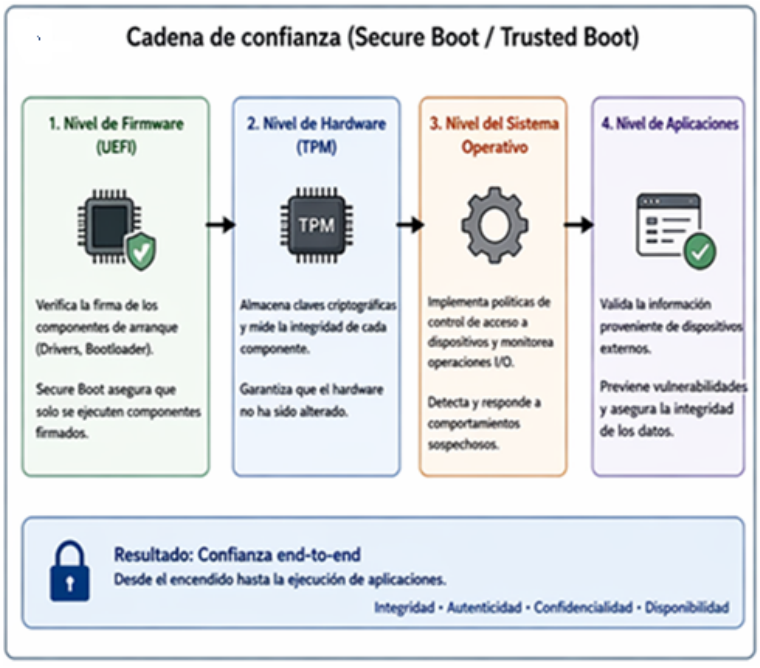
1. **Nivel de firmware (UEFI)**
    - Activación de Secure Boot.
    - Validación de controladores de dispositivos con capacidades DMA.
    - Restricción de dispositivos no autorizados.
2. **Nivel de hardware**
    - Implementación de IOMMU para controlar el acceso DMA a la memoria.
    - Uso de TPM para garantizar la integridad del sistema.
3. **Nivel del sistema operativo**
    - Gestión de permisos para dispositivos DMA.
    - Monitoreo de accesos a memoria.
4. **Nivel de aplicaciones**
    - Validación de datos provenientes de dispositivos externos.
    - Uso de APIs seguras.

Jaramillo et al. (2022) indican que las arquitecturas modernas están incorporando mecanismos de seguridad directamente en el hardware, lo que permite mitigar riesgos asociados al acceso directo a memoria.

La figura 6 ilustra los niveles de seguridad y muestra cómo cada nivel valida al siguiente.

Figura 6

Cadena de confianza (Secure Boot / Trusted Boot)



Autor: Milton Román

Para garantizar un uso seguro del DMA, es necesario implementar medidas que reduzcan los riesgos sin afectar la eficiencia del sistema:

- **Configuración de IOMMU:** Permite controlar y restringir el acceso de dispositivos a la memoria.
- **Deshabilitación de dispositivos no utilizados:** Reduce la superficie de ataque al limitar el acceso físico.
- **Actualización de firmware:** Corrige vulnerabilidades en controladores y dispositivos.
- **Monitoreo de actividad:** Permite detectar comportamientos anómalos relacionados con accesos DMA.

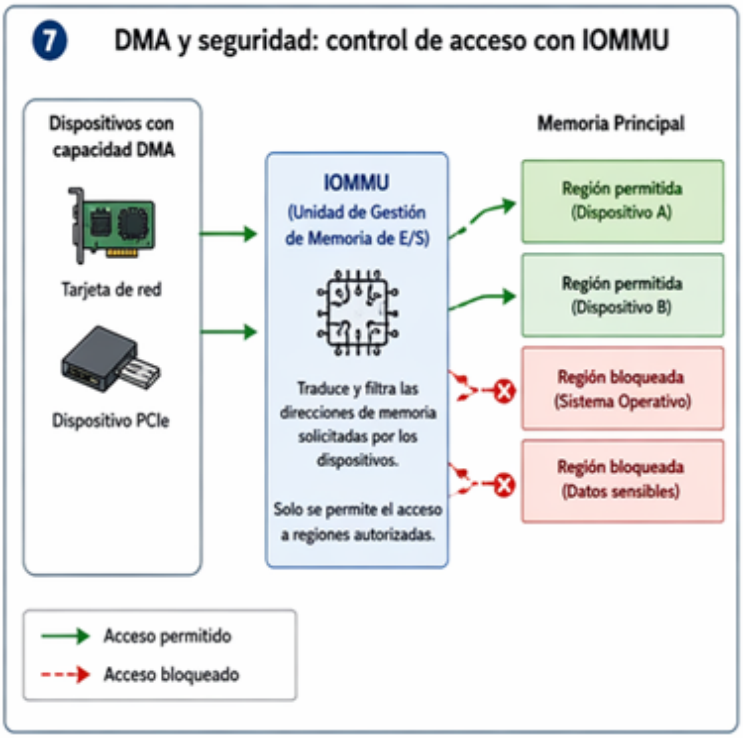
- **Segmentación de memoria:** Limita el acceso de dispositivos a regiones específicas de memoria.

Estas medidas permiten equilibrar la seguridad y el rendimiento, alineándose con los objetivos del RDA 3.

La figura 7 muestra cómo la IOMMU permite accesos válidos a memoria, bloquea accesos no autorizados y visualiza claramente la protección contra ataques DMA.

Figura 7

DMA y seguridad con IOMMU



Autor: Milton Román

El DMA ejemplifica cómo un mecanismo diseñado para mejorar la eficiencia puede convertirse en un riesgo si no se gestiona adecuadamente. Desde la ciberseguridad, es esencial analizar no solo su funcionamiento técnico, sino también las vulnerabilidades que puede introducir, especialmente por su capacidad de acceder directamente a la memoria del sistema. El diseño de sistemas modernos exige equilibrar rendimiento y seguridad, ya que la optimización de recursos es clave en entornos con grandes volúmenes de datos (Hennessy et al., 2019).

Además, la formación en arquitectura de computadoras debe integrar ambos enfoques, permitiendo desarrollar soluciones que respondan a las demandas actuales (Fabra Gallo et al., 2020). Aunque el DMA mejora significativamente la transferencia de datos y reduce la carga de la CPU, también puede ser explotado si no se implementan mecanismos de protección adecuados.

En este sentido, el uso de tecnologías como Secure Boot permite establecer una cadena de confianza desde el arranque del sistema, mientras que herramientas como IOMMU y TPM refuerzan la seguridad en el acceso a memoria. En conjunto, integrar DMA de forma segura y eficiente no solo mejora el rendimiento, sino que también responde al RDA 3, formando profesionales preparados para enfrentar desafíos en ciberseguridad.

Ingresa al siguiente enlace, revisa el contenido del video y refuerza tu conocimiento sobre el tema de acceso directo a memoria:

- **Título:** Acceso Directo a Memoria (DMA): Qué es, Cómo Funciona y Para Qué Sirve
- **Descripción:** El video tiene un enfoque didáctico y técnico, ideal para comprender cómo este mecanismo DMA optimiza el uso de recursos en los sistemas modernos.
- **Enlace**: Acceso Directo a Memoria (DMA): Qué es, Cómo Funciona y Para Qué Sirve

Educación de calidad

REFERENCIAS

- **Fabra Gallo, J. M., Martínez Márquez, Y., & Valcárcel Izquierdo, N. (2020).** Estudio comparado de los contenidos de la asignatura Arquitectura de Computadoras: oportunidades de mejoras. Dialnet. Obtenido de https://dialnet.unirioja.es/servlet/articulo?codigo=8590371
- **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
- **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

PUCE Virtual

síguenos en:

www.pucevirtual.puce.edu.ec