

# Descubriendo el Corazón del Computador: Hardware y Software en Acción

Ingeniería | Ingeniería de sistemas | Aprendizaje Basado en Proyectos

## Descripción

Este plan de clase está diseñado para que estudiantes universitarios de Ingeniería de Sistemas comprendan profundamente los conceptos fundamentales de hardware y software en los computadores. A través de un enfoque activo basado en proyectos, los estudiantes explorarán cómo estas dos áreas interactúan para hacer funcionar un computador, conectando la teoría con aplicaciones reales y actuales. Comprender esta dualidad es esencial para cualquier ingeniero de sistemas, pues sienta las bases para diseñar, analizar y optimizar sistemas computacionales. Además, este conocimiento permite a los estudiantes vincular la tecnología con su vida cotidiana y profesional, facilitando una visión crítica y creativa sobre la evolución y futuro de la informática. Durante la sesión, los estudiantes trabajarán colaborativamente en un proyecto que les permitirá experimentar el funcionamiento interno de un computador, identificar componentes y analizar la relación entre hardware y software, fomentando el aprendizaje autónomo y el desarrollo de competencias técnicas y transversales.

## Objetivos de Aprendizaje

- Analizar la función y características principales de los componentes de hardware y software en un computador.
- Identificar y describir la interacción entre hardware y software mediante la elaboración de un proyecto práctico.
- Crear un modelo conceptual que represente la arquitectura básica de un computador y su funcionamiento integrado.
- Evaluar casos reales donde la integración hardware-software impacta en el desempeño y utilidad del sistema computacional.

## Recursos Necesarios

- Computadores de escritorio o portátiles (1 por grupo de 3-4 estudiantes).
- Software de simulación de hardware, por ejemplo, simuladores de circuitos básicos o de arquitectura de computadores (como Logisim o similares).
- Proyector y computador para presentación docente.
- Material impreso: esquemas básicos de arquitectura de computadores, glosario de términos técnicos.
- Hojas blancas, marcadores, post-its para elaboración de mapas conceptuales.
- Acceso a internet para consulta rápida durante la sesión.

## Requisitos Previos

- Conocimientos básicos de informática general y componentes físicos de un computador.
- Habilidad para trabajar en equipo y comunicarse efectivamente.
- Familiaridad con conceptos elementales de sistemas operativos y software básico.
- Experiencia previa en lectura de textos técnicos y realización de presentaciones breves.

## Actividades

### Fase de Inicio

**Tiempo estimado: 20 minutos**

#### Propósito de la sesión:

**Docente:** “Hoy exploraremos cómo el hardware y software trabajan juntos para que un computador funcione. Entenderemos su relación y cómo esto es la base de todo sistema computacional, lo que es esencial para su formación como ingenieros.”

#### Activación de conocimientos previos:

- **Docente:** “Para comenzar, reflexionemos: ¿Podrían dar un ejemplo de hardware y otro de software que usen en su vida diaria? ¿Qué pasaría si uno de ellos faltara?”
- **Estudiantes:** Responden brevemente en plenaria, compartiendo ejemplos concretos y planteando ideas sobre la importancia de ambos.

#### Motivación y enganche:

- **Docente:** Presenta un dato curioso: “¿Sabían que el primer computador personal pesaba más de 27 kg y tenía un software muy limitado? Hoy, en sus computadores, hardware y software han evolucionado para multiplicar su capacidad y versatilidad.”
- Breve demostración: muestra un video de 2 minutos con la evolución histórica del hardware y software.

#### Contextualización:

- **Docente:** “Como futuros ingenieros, comprenderán cómo estas piezas fundamentales trabajan juntas para desarrollar aplicaciones que usamos todos los días, desde redes sociales hasta sistemas críticos.”
- **Estudiantes:** Reflexionan y conectan el tema con sus experiencias personales y profesionales.

### Fase de Desarrollo

**Tiempo estimado: 80 minutos**

#### Presentación del contenido:

**Docente:** Introduce el proyecto: “Vamos a crear un modelo funcional simple que muestre la interacción entre hardware y software. Esto nos permitirá visualizar y comprender su complementariedad.”

### **Actividad 1: “Mapa conceptual colaborativo: Hardware vs Software”**

- **Objetivo:** Analizar y describir los componentes y funciones de hardware y software.
- **Instrucciones:**
  - En grupos de 3-4, usen las hojas y marcadores para construir un mapa conceptual que incluya componentes, funciones y ejemplos de hardware y software.
  - Incluyan conexiones que expliquen cómo interactúan entre sí.
- **Organización:** Grupos de 3-4 estudiantes.
- **Producto:** Mapa conceptual en formato físico.
- **Tiempo:** 25 minutos.
- **Rol docente:** Circular por grupos, hacer preguntas como: “¿Cómo creen que este componente de hardware se comunica con el software?” o “¿Qué software es esencial para que este hardware funcione?”

### **Actividad 2: “Simulación práctica de arquitectura básica”**

- **Objetivo:** Crear un modelo conceptual que represente la arquitectura básica del computador y entender su funcionamiento integrado.
- **Instrucciones:**
  - Con apoyo del software simulador (Logisim o similar), cada grupo debe armar un circuito básico que represente la unidad central de procesamiento (CPU), memoria y dispositivos de entrada/salida.
  - Simulen la ejecución de una instrucción simple para observar la interacción hardware-software.
- **Organización:** Grupos de 3-4 estudiantes frente a un computador.
- **Producto:** Simulación funcional y breve explicación del proceso.
- **Tiempo:** 35 minutos.
- **Rol docente:** Apoyar con dudas técnicas, guiar preguntas: “¿Qué parte del hardware procesa la información del software?” “¿Cómo se transfiere la información entre componentes?”

### **Actividad 3: “Análisis de caso real: Impacto hardware-software”**

- **Objetivo:** Evaluar casos reales donde la integración hardware-software afecta el desempeño del sistema.
- **Instrucciones:**
  - Proporcionar un caso breve (por ejemplo, problemas de compatibilidad o actualizaciones de software que requieren mejoras en hardware).
  - En grupos, analicen el caso y propongan soluciones o mejoras.
  - Presenten sus conclusiones en una breve exposición de 5 minutos por grupo.
- **Organización:** Grupos de 3-4 estudiantes.

- **Producto:** Presentación oral y conclusiones escritas.
- **Tiempo:** 20 minutos.
- **Rol docente:** Facilita recursos, escucha exposiciones, formula preguntas para profundizar: “¿Qué implicaciones tendría para el usuario final?” “¿Cómo podría evolucionar esta situación con nuevas tecnologías?”

### Diferenciación:

- **Estudiantes adelantados:** Invitados a explorar funciones avanzadas del simulador o investigar ejemplos adicionales de hardware/software emergente para compartir con el grupo.
- **Estudiantes con dificultades:** Reciben apoyo adicional en el mapa conceptual con ejemplos guiados y asesoría paso a paso en la simulación.

### Transiciones:

- Al finalizar la actividad 1, el docente conecta: “Ahora que identificamos y organizamos los conceptos, vamos a ver cómo funcionan en conjunto mediante una simulación práctica.”
- Luego de la simulación, el docente introduce el análisis de casos: “Finalmente, observaremos cómo estas interacciones afectan sistemas reales, para que puedan aplicar su conocimiento a situaciones cotidianas y profesionales.”

### Fase de Cierre

#### Tiempo estimado: 20 minutos

### Síntesis:

- **Actividad:** “Ticket de salida digital” (puede ser mediante una plataforma o en papel).
- **Instrucciones:** Cada estudiante responde en 3 frases:
  - Una idea clave que aprendió sobre hardware y software.
  - Cómo cree que esto impactará su formación profesional.
  - Una pregunta que le gustaría profundizar en el futuro.

### Reflexión metacognitiva:

- ¿Cómo describirías la relación entre hardware y software a alguien sin conocimientos técnicos?
- ¿Qué partes del proyecto te ayudaron más a entender estos conceptos y por qué?
- ¿Qué aspectos te resultaron más difíciles y cómo los superaste o qué necesitarías para hacerlo?

### Retroalimentación:

- **Docente:** Lee y comenta algunos tickets de salida en voz alta, resalta logros y aclara dudas frecuentes.
- Brinda comentarios específicos a cada grupo sobre sus mapas, simulaciones y presentaciones, destacando puntos fuertes y áreas de mejora.

### Transferencia:

- **Docente:** “En próximas sesiones profundizaremos en software específico y diseño de sistemas, usando esta base para entender mejor cómo optimizar y crear soluciones tecnológicas.”
- Invita a los estudiantes a observar los dispositivos y sistemas que usan diariamente desde la perspectiva aprendida hoy.

### Tarea o reto:

- Investigar un caso real reciente donde una actualización de hardware o software haya generado un cambio significativo en el desempeño de una aplicación o sistema, y preparar un breve informe para compartir en la siguiente clase.

## Evaluación

**Tipo de evaluación:** La evaluación es formativa y sumativa. Se realiza durante la fase de desarrollo (evaluación formativa a través de observación directa y retroalimentación continua) y en la fase de cierre (evaluación sumativa con el ticket de salida y presentación grupal).

### Criterios de evaluación:

- Capacidad para identificar y explicar correctamente componentes de hardware y software (Objetivo 1).
- Habilidad para representar la interacción hardware-software en un proyecto colaborativo (Objetivo 2 y 3).
- Calidad del análisis crítico y propuestas en el caso real presentado (Objetivo 4).

### Instrumentos sugeridos:

- Lista de cotejo para evaluación de mapas conceptuales y presentaciones.
- Rúbrica para valorar la simulación práctica y la integración conceptual.
- Observación directa y registro anecdótico durante actividades grupales.
- Autoevaluación y coevaluación mediante reflexiones al final de la sesión.

### Evidencias de aprendizaje:

- Mapa conceptual colaborativo que refleja comprensión de hardware y software.
- Simulación funcional que demuestra el modelo básico de computador y su interacción interna.
- Presentación oral y análisis escrito del caso real.
- Respuestas en el ticket de salida que evidencian reflexión y metacognición.

## Enriquecimientos

### Recomendaciones - Tic\_ia

### Fase de Inicio

- **Herramienta:** [Kahoot!](#) (Plataforma de cuestionarios interactivos)

Implementación: Utilizar Kahoot! para realizar una breve encuesta o cuestionario en vivo sobre ejemplos de hardware y software que los estudiantes usan cotidianamente. Esto sustituye la ronda de preguntas tradicional en plenaria con una actividad digital interactiva que motiva la participación.

Contribución a objetivos: Activa conocimientos previos y genera interés, fomentando la reflexión sobre hardware y software en un formato dinámico y accesible para universitarios.

Nivel SAMR: Sustitución

- **Herramienta:** Video histórico interactivo con [Edpuzzle](#)

Implementación: Usar Edpuzzle para presentar el video sobre la evolución del hardware y software, insertando preguntas interactivas que los estudiantes responden mientras ven el video. Esto aumenta la comprensión y mantiene la atención.

Contribución a objetivos: Facilita la motivación y contextualización con apoyo multimedia interactivo, reforzando conceptos clave y promoviendo la reflexión crítica.

Nivel SAMR: Aumento

## Fase de Desarrollo

- **Herramienta:** [Miro](#) (Pizarra colaborativa digital)

Implementación: Los grupos crean mapas conceptuales colaborativos en Miro, integrando texto, imágenes y conexiones para describir hardware y software. Permite colaboración simultánea en línea o en aula con laptops.

Contribución a objetivos: Rediseña la actividad tradicional con hojas y marcadores, facilitando la co-creación en tiempo real, mejor organización visual y posibilidad de enriquecer con recursos digitales.

Nivel SAMR: Modificación

- **Herramienta:** [ChatGPT](#) (IA para generación de contenido y apoyo conceptual)

Implementación: Los estudiantes consultan a ChatGPT para obtener explicaciones, ejemplos adicionales o aclarar dudas sobre hardware y software durante la construcción del mapa conceptual. El docente supervisa el uso para garantizar precisión y fomentar pensamiento crítico.

Contribución a objetivos: Permite a los estudiantes ampliar su comprensión con apoyo inmediato, fomentando un aprendizaje autodirigido y mejor fundamentado en conceptos técnicos.

Nivel SAMR: Modificación

## Fase de Cierre

- **Herramienta:** [Padlet](#) (Muro colaborativo digital)

Implementación: Cada grupo sube su mapa conceptual finalizado a un Padlet común. Todos pueden revisar, comentar y votar por los mapas más claros o creativos, promoviendo retroalimentación entre pares.

Contribución a objetivos: Facilita la presentación y discusión de trabajos en un espacio digital accesible, enriqueciendo la evaluación formativa y fomentando la reflexión crítica colectiva.

Nivel SAMR: Aumento

- **Herramienta:** Simulador de arquitectura de computador como [Nand2Tetris Hardware Simulator](#)

Implementación: Como actividad final, los estudiantes exploran un simulador que permite construir y probar componentes básicos de hardware y su interacción con software simple, creando una experiencia práctica y profunda.

Contribución a objetivos: Permite crear una tarea que antes no era posible en aula tradicional, profundizando la comprensión mediante experimentación virtual directa con conceptos teóricos.

Nivel SAMR: Redefinición

## Desarrollo - Ejemplos

### Ejemplos Prácticos y Casos de Estudio para "Descubriendo el Corazón del Computador: Hardware y Software en Acción"

Estos ejemplos y casos de estudio están diseñados para que los estudiantes universitarios de Ingeniería de Sistemas trabajen en equipo, promoviendo el Aprendizaje Basado en Proyectos. Cada actividad conecta con los objetivos de aprendizaje sobre el entendimiento profundo de hardware y software, su interacción y aplicación práctica en contextos reales.

#### Ejemplo Práctico 1: Diagnóstico y Mejora de un Computador Personal

- **Descripción:** En grupos, los estudiantes reciben un computador de escritorio con problemas específicos (lentitud, fallos en la instalación de software, errores de hardware simulados).
- **Objetivo:** Identificar componentes de hardware y software involucrados en el problema, diagnosticar la causa, y proponer soluciones integrales.
- **Actividad:**
  - Analizar el hardware (CPU, RAM, disco duro, periféricos) y detectar posibles fallos.
  - Revisar el software instalado y su compatibilidad con el hardware.
  - Elaborar un reporte técnico que explique el diagnóstico y las acciones correctivas.
- **Duración sugerida:** 50 minutos

#### Ejemplo Práctico 2: Comparación de Sistemas Operativos para Entornos Específicos

- **Descripción:** Cada grupo selecciona un entorno de aplicación (servidores, computadoras personales, dispositivos móviles) y analiza dos sistemas operativos diferentes (por ejemplo, Windows vs Linux, Android vs iOS).
- **Objetivo:** Entender cómo el software (sistemas operativos) interactúa con el hardware para optimizar el desempeño en contextos específicos.

- **Actividad:**

- Investigar características técnicas de cada sistema operativo.
- Evaluar compatibilidad con hardware típico del entorno seleccionado.
- Presentar un informe comparativo y recomendar el sistema operativo más adecuado según criterios técnicos y prácticos.

- **Duración sugerida:** 40 minutos

## **Caso de Estudio: Implementación de un Sistema Embebido en un Proyecto de Automatización**

- **Descripción:** Se presenta un proyecto donde se debe diseñar un sistema embebido para controlar sensores y actuadores en un ambiente industrial.
- **Objetivo:** Analizar la integración entre hardware específico (microcontroladores, sensores) y software (firmware, sistemas operativos en tiempo real) para lograr funcionalidad y eficiencia.

- **Actividad:**

- Identificar los componentes hardware necesarios para el proyecto.
- Seleccionar y justificar el tipo de software embebido que será utilizado.
- Desarrollar un esquema de funcionamiento y presentar un plan de implementación.

- **Duración sugerida:** 30 minutos

## **Integración de las Actividades**

Al finalizar, los grupos comparten sus hallazgos y propuestas en una discusión guiada para consolidar el aprendizaje, analizar diferentes enfoques y relacionar teoría con práctica.