



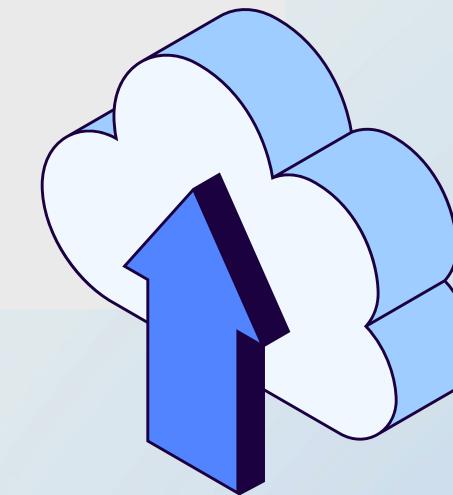
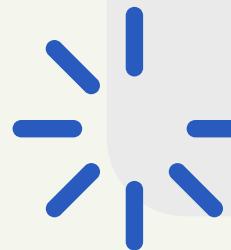
2.1 Registros visibles para el usuario

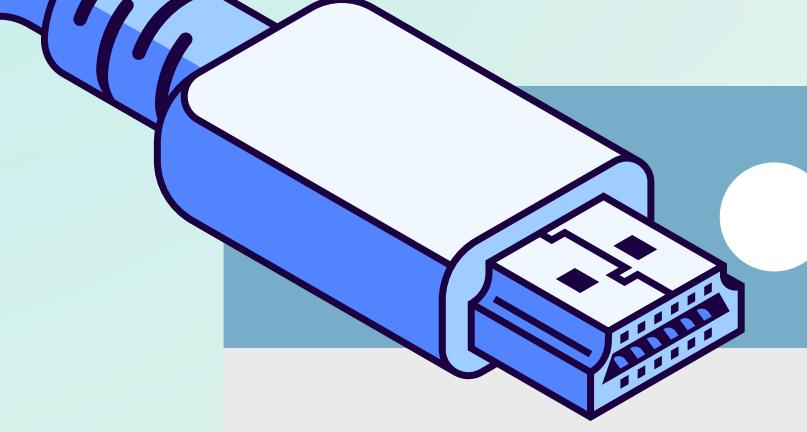




¿Que son?

Los registros visibles para el usuario son aquellos que pueden ser accedidos directamente por los programadores para manipular datos y direcciones.





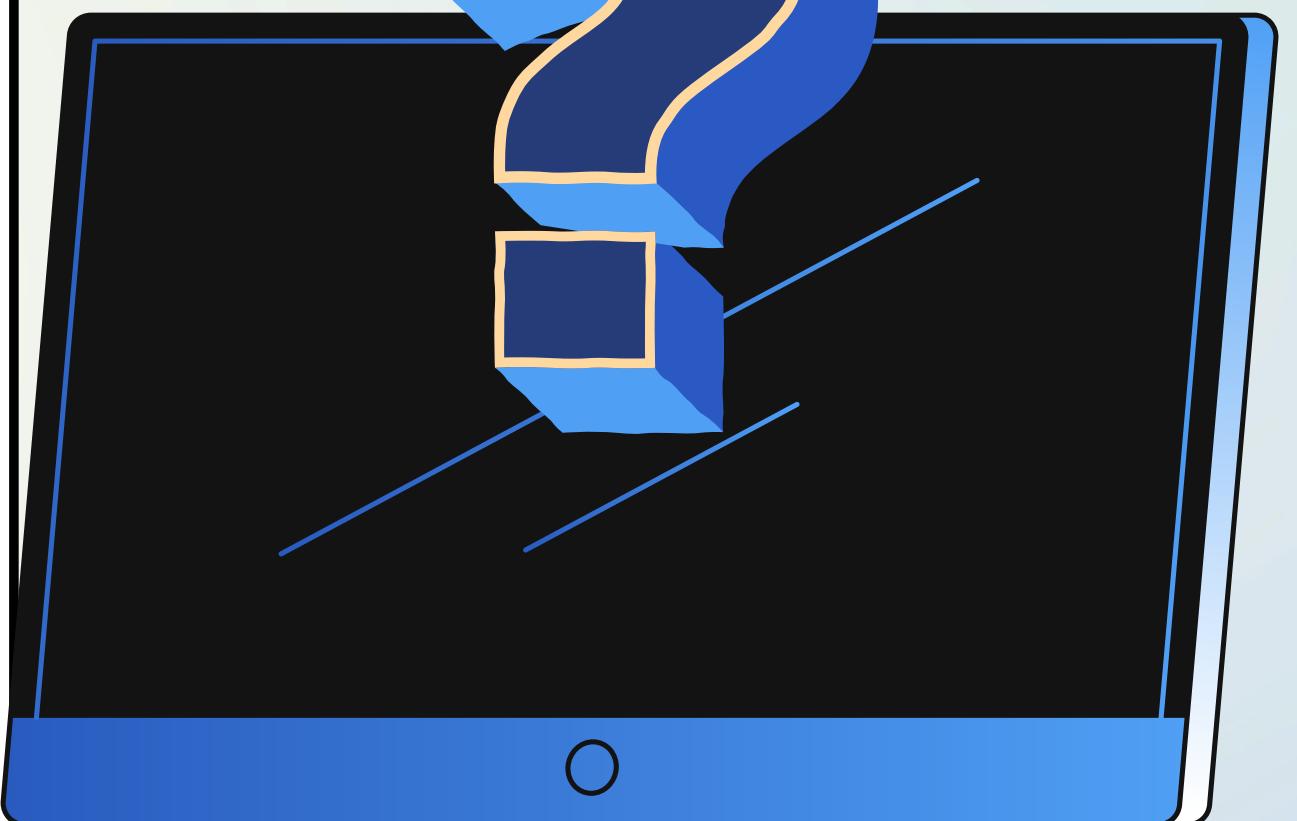
Tipos de registros visibles para el usuario

- **Registros de datos:** Almacenan valores numéricos para operaciones.
- **Registros de direcciones:** Contienen direcciones de memoria.
- **Registros de propósito general:** Usados para diferentes tareas según la necesidad del programa.
- **Registros de condición:** Guardan información sobre el resultado de una operación.



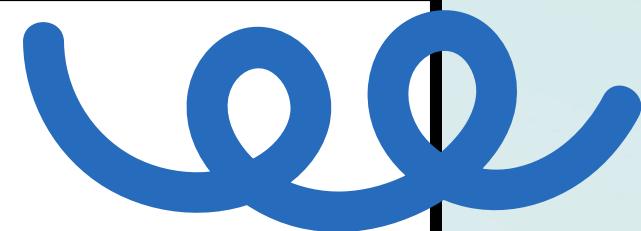
Importancia

Estos registros permiten que los programas accedan y manipulen datos de manera rápida, optimizando la ejecución de instrucciones. Los registros visibles para el usuario facilitan la programación y mejoran el rendimiento del procesador.

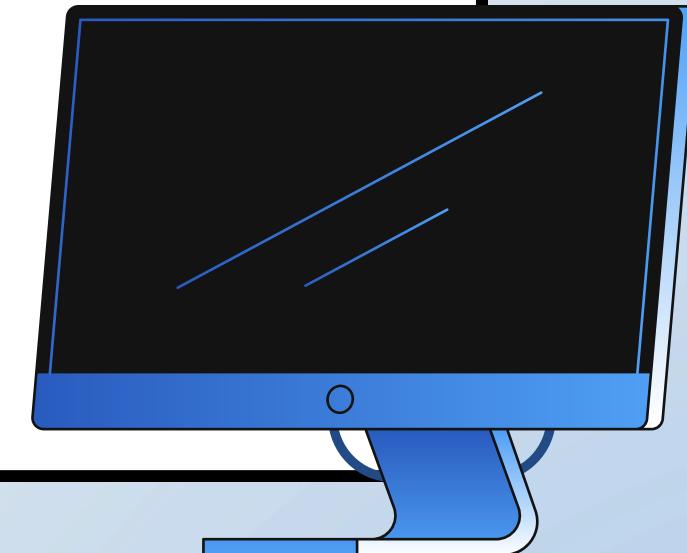


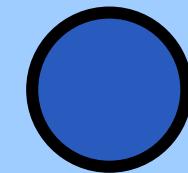
2.2.2 Registros de control y de estados



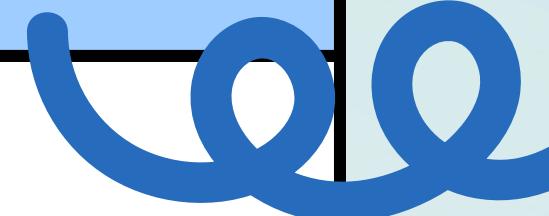


Estos registros son utilizados por la CPU para gestionar su operación y supervisar el estado del sistema. Son esenciales para el control interno del procesador, ya que almacenan información sobre la ejecución de instrucciones, posibles errores y condiciones específicas del hardware. Gracias a estos registros, la CPU puede tomar decisiones sobre su funcionamiento y garantizar que los procesos se realicen de manera estable y segura.

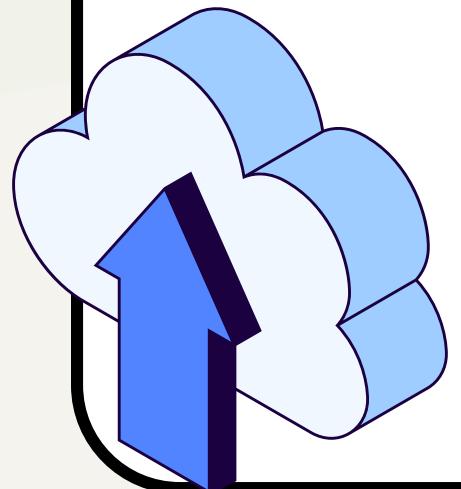


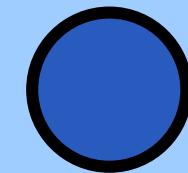


Registros de control



Determinan el modo de operación de la CPU. Estos registros permiten gestionar cómo se ejecutan las instrucciones y configuran parámetros esenciales para el funcionamiento del procesador. Entre ellos se encuentran registros que controlan el acceso a memoria, la gestión de interrupciones y el manejo del contexto de los procesos.

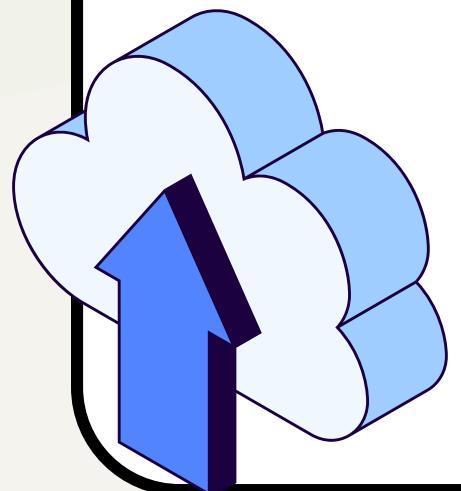




Registros de estado



Contienen información sobre el estado actual del procesador, como indicadores de errores o banderas de condición. Estos registros almacenan datos sobre los resultados de las operaciones realizadas por la CPU y permiten tomar decisiones sobre el flujo de ejecución del programa. Incluyen banderas de estado como el bit de desbordamiento (**overflow**), el bit de acarreo (**carry**) y el bit de cero, que ayudan a determinar el comportamiento de las instrucciones siguientes.



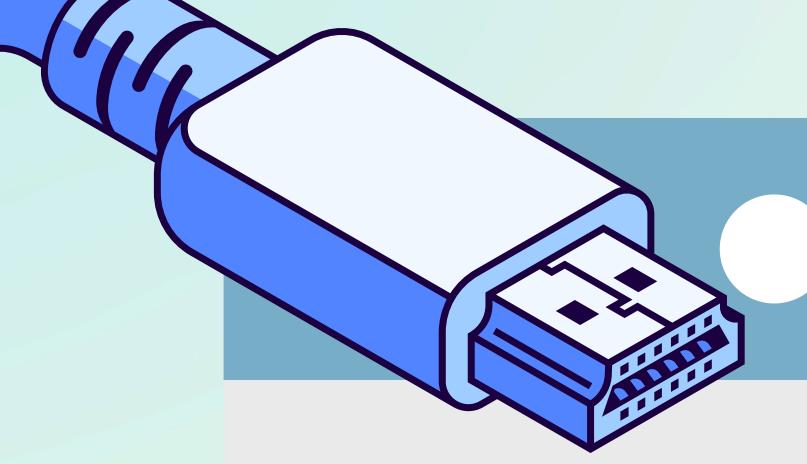
Importancia

Estos registros permiten que el procesador funcione correctamente, asegurando que las instrucciones se ejecuten de manera eficiente y sin errores. Los registros de control y de estado son esenciales para la estabilidad y el control del procesador, permitiendo un funcionamiento eficiente.



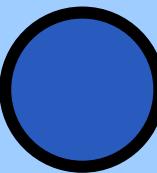
2.2.3 Ejemplos de organización de registros de CPU reales.



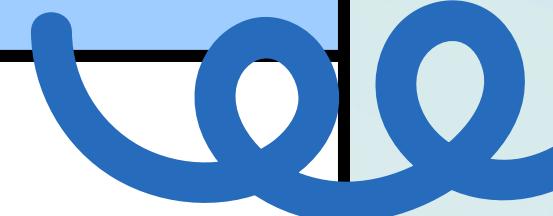


Tipos de registros de la CPU

- **Registros de propósito general:** Usados para almacenar datos temporales y variables en operaciones. Ejemplo: AX, BX en x86.
 - **Registros de propósito específico:** Incluyen el Contador de Programa (PC), que guarda la dirección de la siguiente instrucción, y el Registro de Instrucción (IR), que almacena la instrucción en ejecución.
 - **Registros de estado y control:** Como el Registro de Banderas (FLAGS en x86), que almacena información sobre el estado del procesador.
 - **Registros de punto flotante:** Usados para cálculos matemáticos avanzados (FPU en x86).
- 



Ejemplos de CPU reales

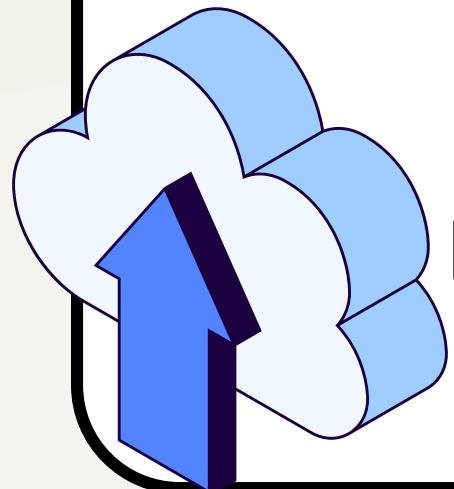


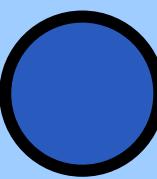
Arquitectura RISC-V

- Utiliza 32 registros de propósito general (x0-x31), donde x0 siempre es cero.
- Tiene registros especiales como el PC (Program Counter) y el CSR (Control and Status Registers).
- Es ampliamente usada en sistemas embebidos y computación de alto rendimiento.

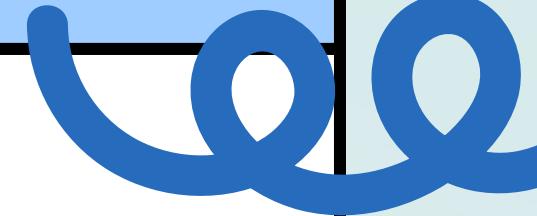
Arquitectura PowerPC (IBM)

- Usa 32 registros de propósito general (GPR0-GPR31) para operaciones enteras.
- Tiene 32 registros de punto flotante (FPR0-FPR31).
- Incluye registros especiales como el LR (Link Register) para llamadas a subrutinas y el CTR (Count Register) para bucles.





Ejemplos de CPU reales

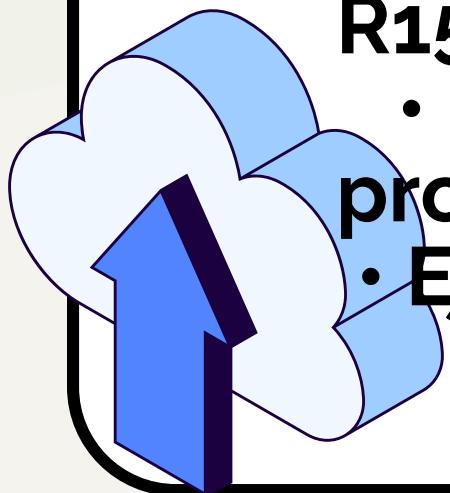


Arquitectura x86 (Intel/AMD)

- Registros de propósito general: EAX, EBX, ECX, EDX (para operaciones y almacenamiento de datos).
 - Registros especiales: ESP (Stack Pointer), EBP (Base Pointer), EIP (Instruction Pointer).
 - Registros de estado: EFLAGS (banderas como Zero Flag y Carry Flag).
 - Ejemplo real: Intel Core i7, AMD Ryzen usan esta organización.

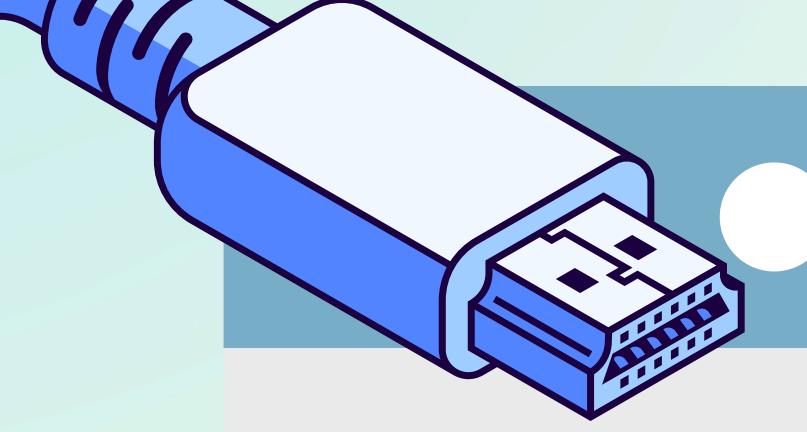
Arquitectura ARM

- Registros de propósito general: R0-R12 (para datos y parámetros de funciones).
 - Registros especiales: R13 (SP - Stack Pointer), R14 (LR - Link Register), R15 (PC - Program Counter).
 - Registros de estado: CPSR (almacena banderas y control del procesador).
 - Ejemplo real: Usado en procesadores de celulares y Raspberry Pi.



2.3 El ciclo de instrucción





Etapas del ciclo de instrucción



El ciclo de instrucción es el proceso mediante el cual la CPU ejecuta instrucciones, dividido en tres etapas principales:

- 1. Fetch (Búsqueda):** Se obtiene la instrucción desde la memoria y se almacena en el IR.
 - 2. Decode (Decodificación):** Se interpreta la instrucción y se determinan los registros y operaciones necesarias.
 - 3. Execute (Ejecución):** Se realizan las operaciones y se almacenan los resultados.
- 

2.3.1 Ciclo Fetch- Decode-Execute.



Importancia

El Ciclo de Instrucción (Fetch-Decode-Execute) es el proceso fundamental mediante el cual una CPU ejecuta instrucciones de un programa.



Fetch

Fetch (Búsqueda de instrucción)

- El procesador obtiene la instrucción desde la memoria RAM.
- La dirección de la instrucción se encuentra en el contador de programa (PC, Program Counter).
- La instrucción se almacena en el Registro de Instrucción (IR, Instruction Register).



Decode

Decode (Decodificación de instrucción)

- La CPU interpreta la instrucción para determinar qué operación realizar y qué operandos necesita.
- Se activa la Unidad de Control para coordinar el proceso.



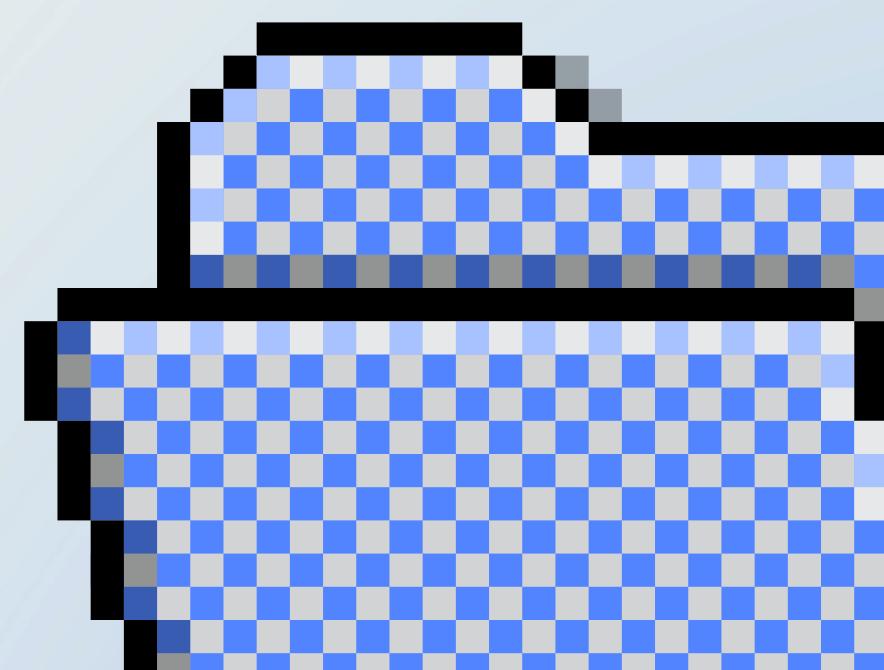
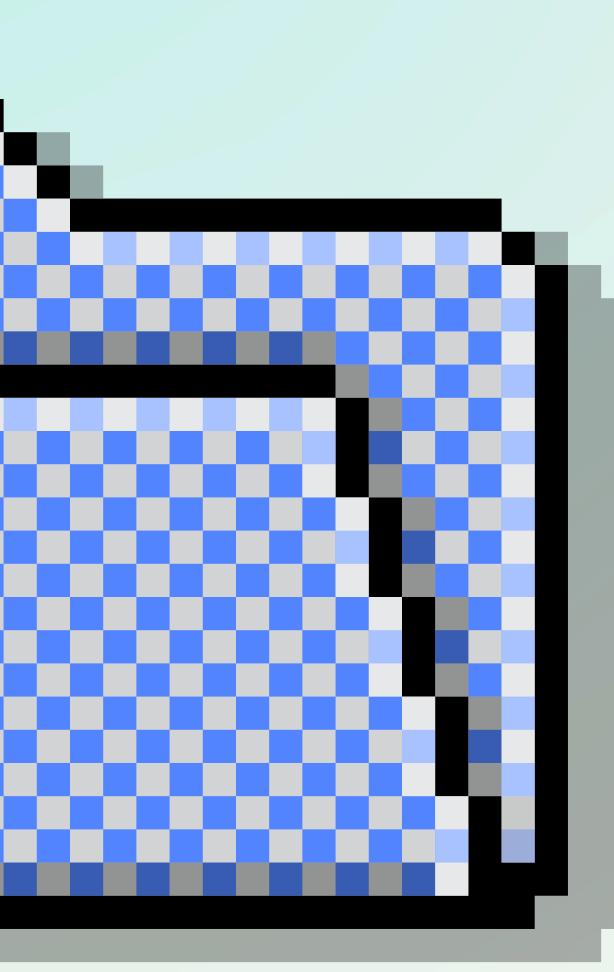
Execute



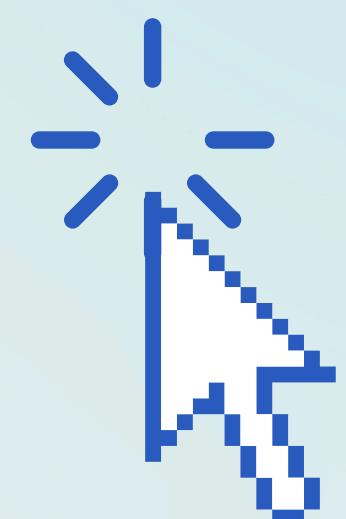
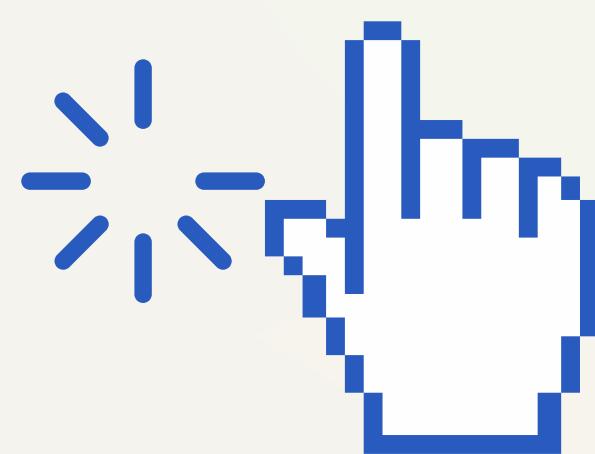
Execute (Ejecución de instrucción)

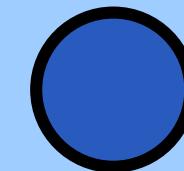
- Se realizan las operaciones necesarias:
- Si es una operación aritmética/lógica, se usa la ALU (Unidad Aritmética y Lógica).
- Si implica memoria, se accede a los datos en RAM.
- Si es una instrucción de control, el flujo del programa cambia (saltos, llamadas a funciones).
- Los resultados pueden almacenarse en registros o en la memoria.



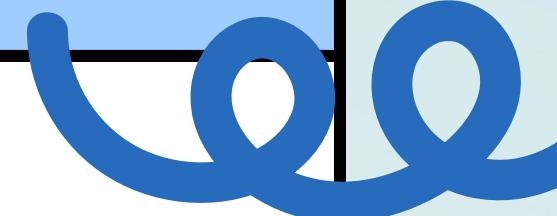


2.3.2 Segmentacion de instrucciones.

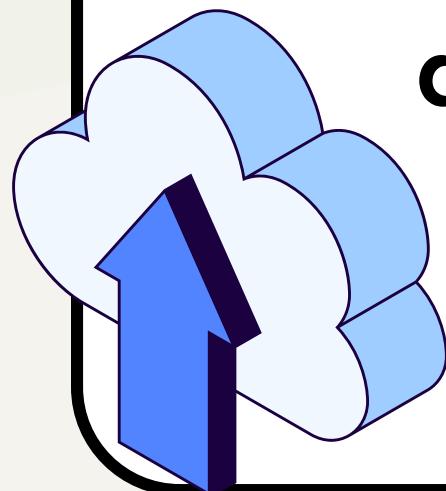




¿Que es ?



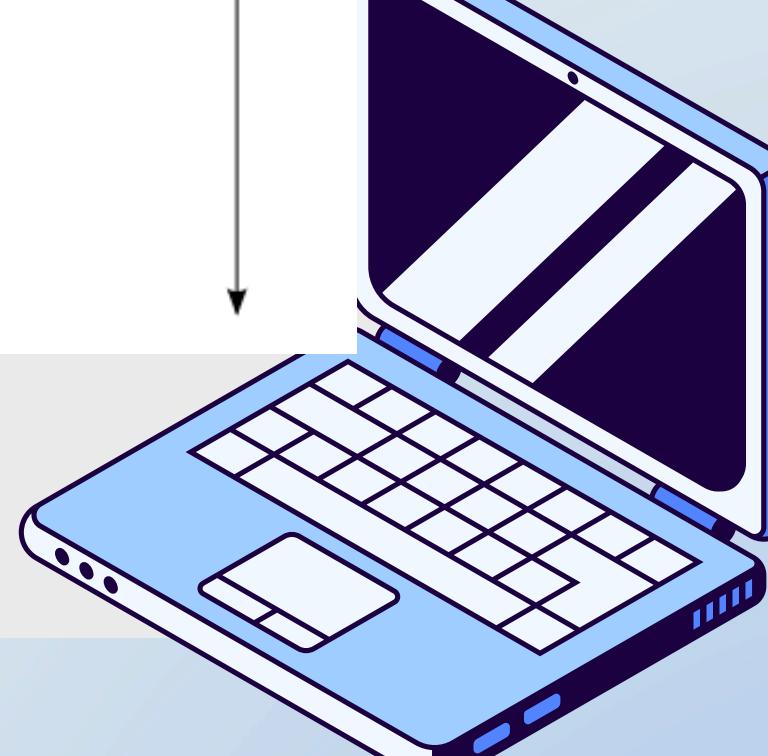
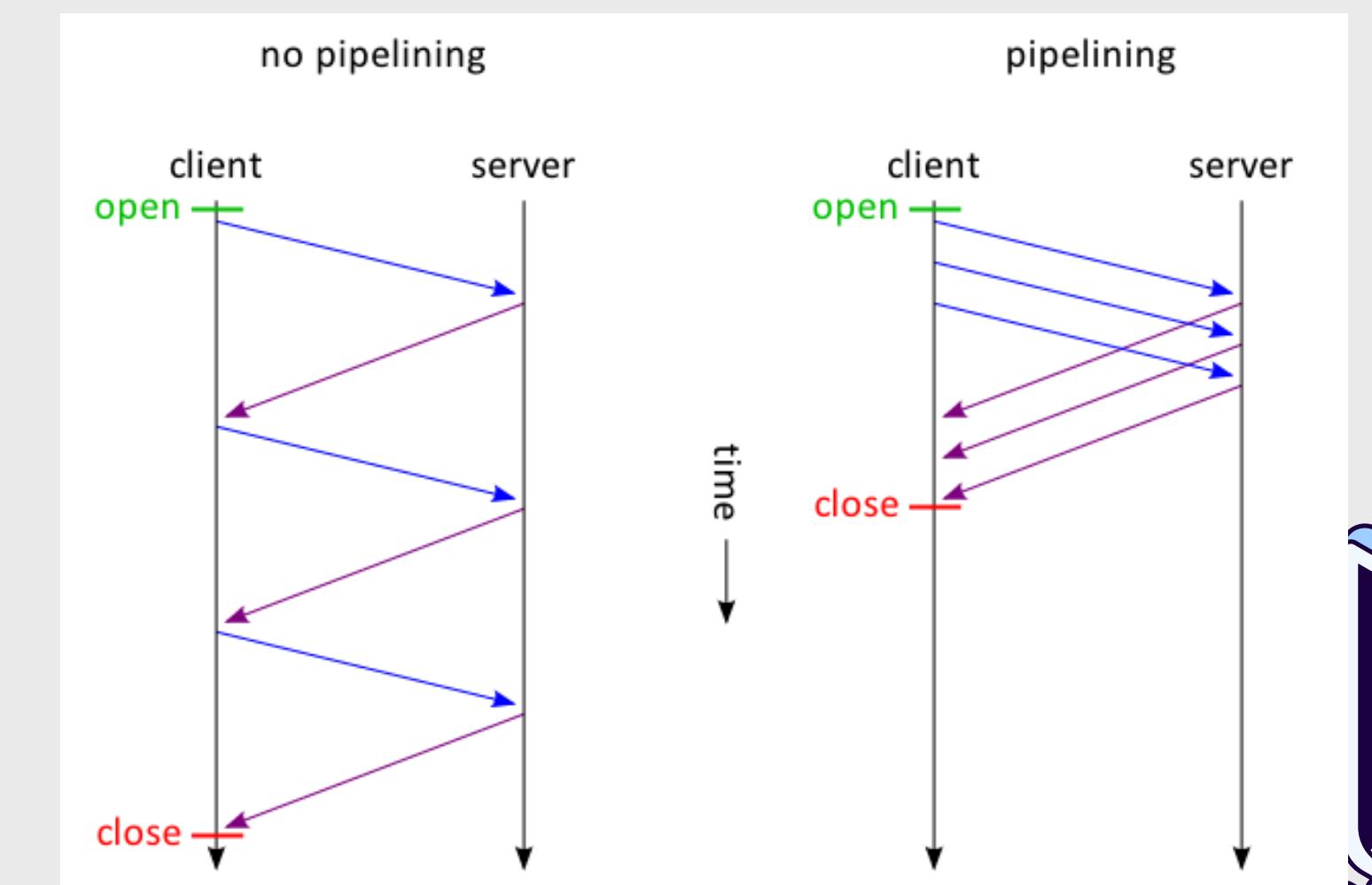
La segmentación de instrucciones es una técnica que permite implementar el paralelismo a nivel de instrucción en un único procesador. La segmentación intenta tener ocupadas con instrucciones todas las partes del procesador dividiendo las instrucciones en una serie de pasos secuenciales que efectuarán distintas unidades de la CPU, tratando en paralelo diferentes partes de las instrucciones.



Tecnica pipeline

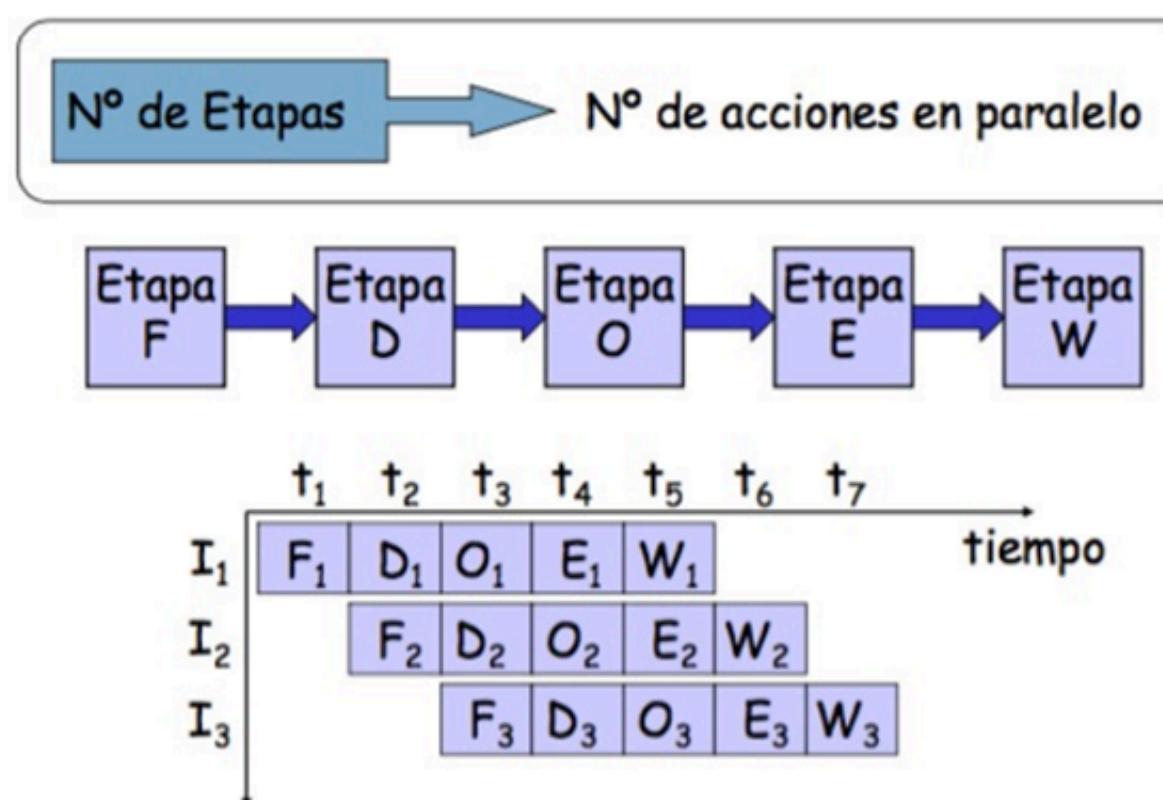
Es un conjunto de elementos de procesadores de datos conectados en serie, en donde la salida de un elemento es la entrada del siguiente.

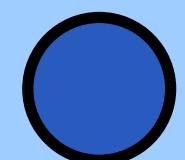
Los elementos del pipeline son generalmente ejecutados en paralelo, en esos casos, debe haber un almacenamiento tipo buffer insertado entre elementos.



Procesador de Pipeline

Esta compuesto por una lista de segmentos lineales y secuenciales en donde cada segmento lleva acabo una tarea o un grupo de tareas computacionales.



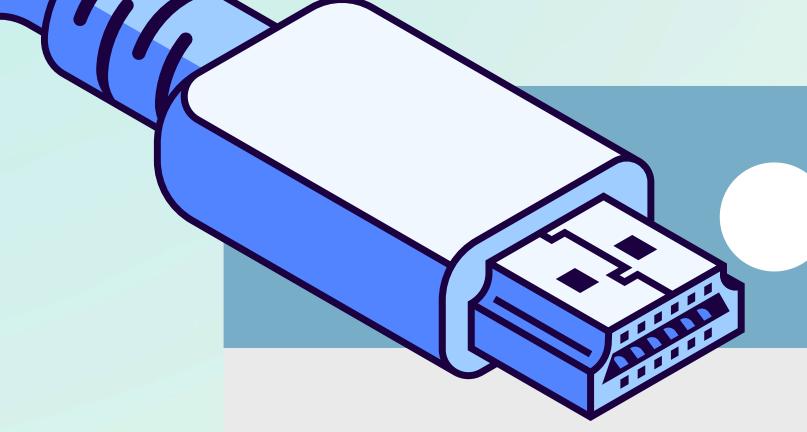


Procesador de Pipeline



- **Multiprogramación y Multiprocesamiento:** Estas acciones se toman a nivel de Programa o Trabajo.
- **Tarea o Procedimientos:** Acciones que se toman dentro de un mismo programa ejecutándose procesos independientes en forma simultánea.
- **Interinstrucciones:** Acciones a nivel de instrucción o sea dentro de mismo proceso o tarea se pueden ejecutar instrucciones independientes en forma simultánea.





Aplicaciones de pipeline

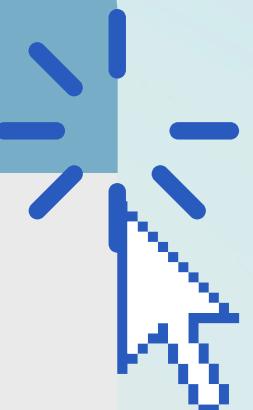
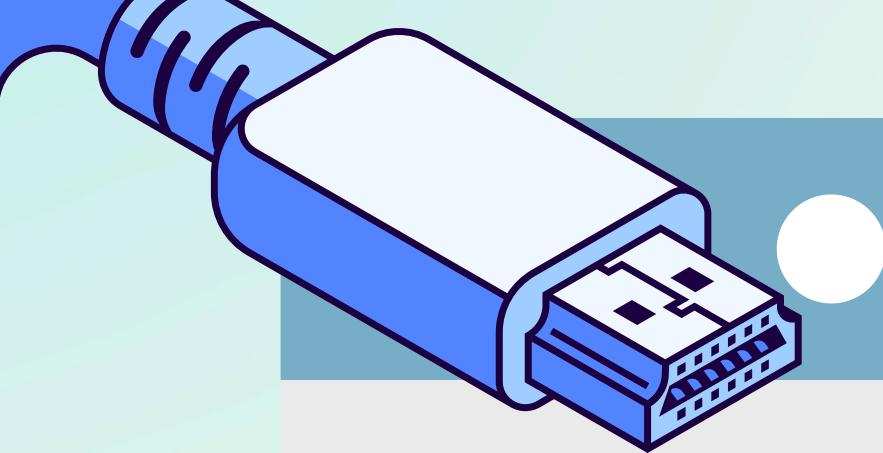


- Pipelines gráficos, se encuentran en la mayoría de las tarjetas gráficas, consiste en múltiples unidades aritméticas o CPUs completas, que implementan variados escenarios de operaciones típicas de renderizado, por ejemplo, cálculos de luz y colores, renderizado, proyección de perspectiva, etc.
 - Pipelines de software o tuberías, consiste en múltiples procesos ordenados de tal forma que el flujo de salida de un proceso alimenta la entrada del siguiente proceso. Por ejemplo, los pipelines de Unix.
- 

Etapas de Ejecucion

- Fetch: Alimentación de la instrucción (fetch).
- Decode: Decodificación de la instrucción.
- Operation: Extracción y cálculo de los operandos.
- Execute: Ejecución (en la ALU)
- Write-back: Escritura del resultado (write)





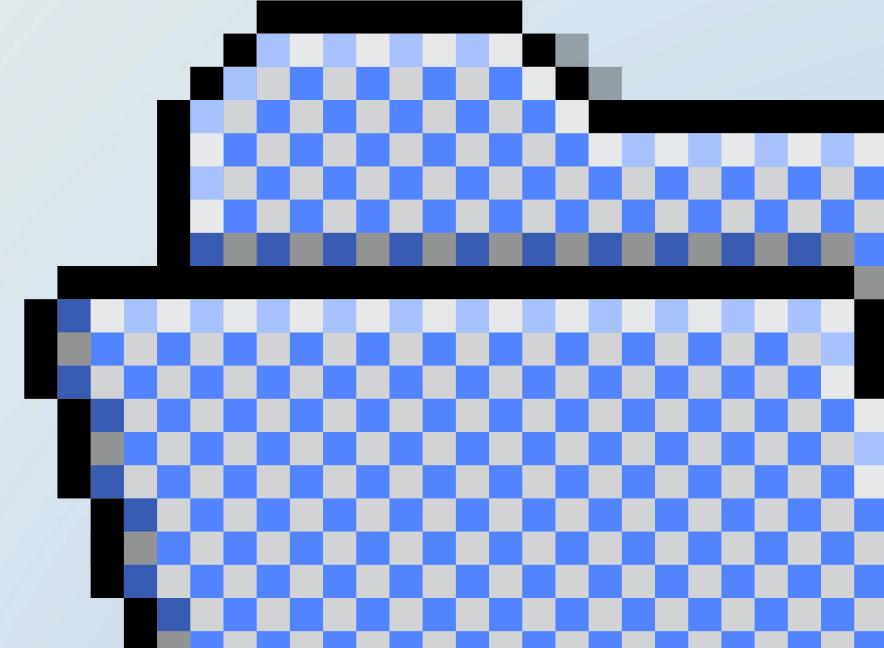
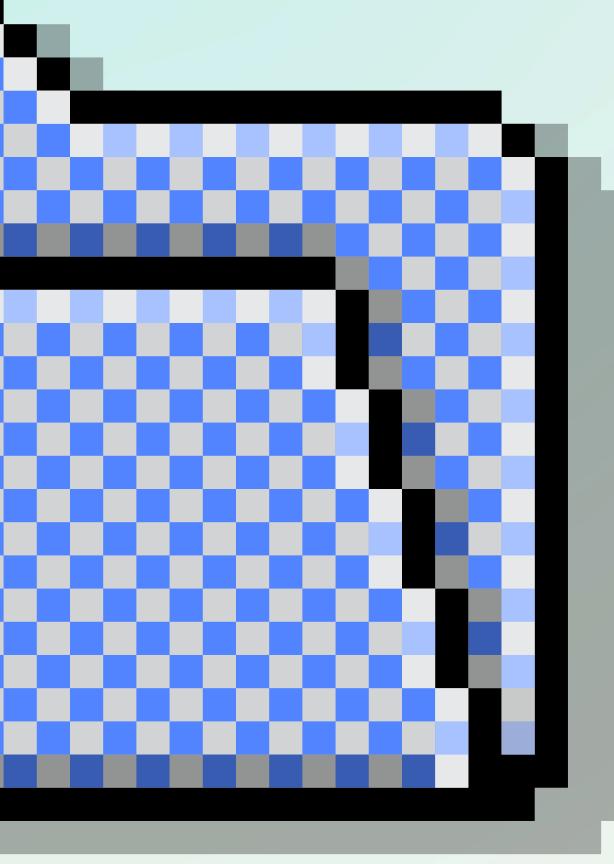
Ventajas

- El tiempo de ciclos del procesador es reducido
- Aumenta el rendimiento del sistema
- Hace que el sistema sea confiable

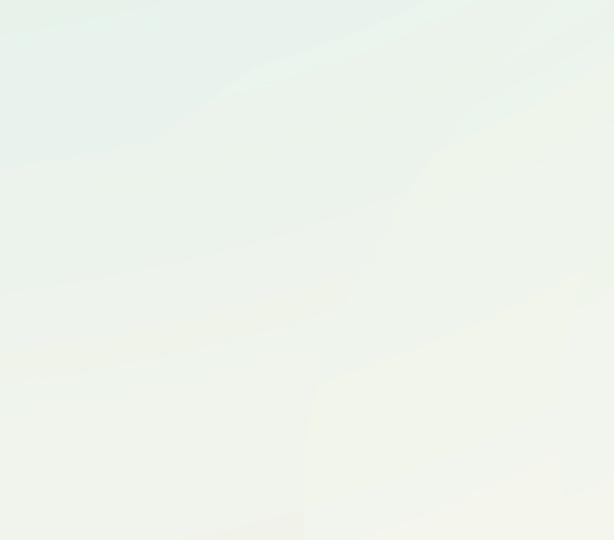
Desventajas

- El diseño de un procesador pipeline es complejo por lo tanto es más costoso de fabricar.
- La latencia de instrucciones es más.





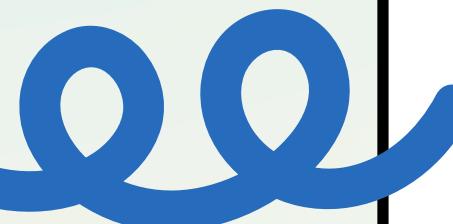
2.3.3 Conjunto de instrucciones: Características y funciones.

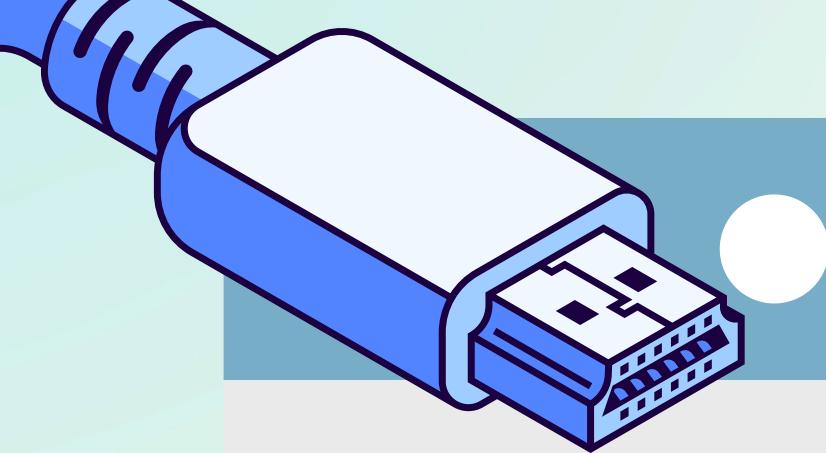


¿Que es?



- El conjunto de instrucciones es el conjunto de comandos o instrucciones que un procesador puede ejecutar. Define cómo el software interactúa con el hardware, especificando las operaciones que el procesador puede realizar, los tipos de datos soportados y el formato de las instrucciones

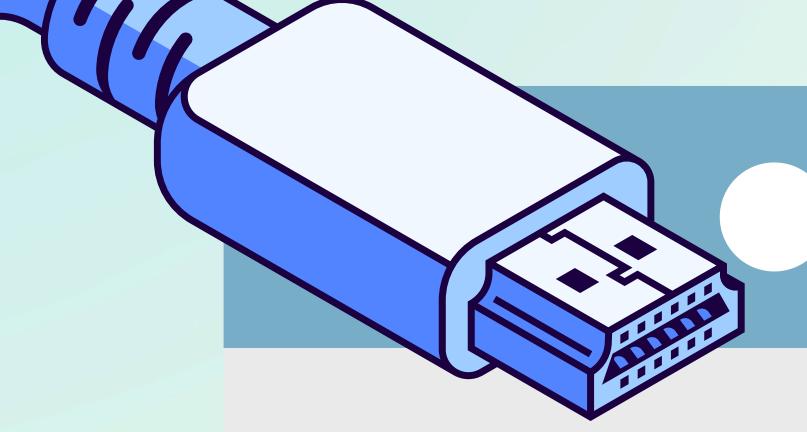




Características de un conjunto de instrucciones

- **Tipos de instrucciones:**
 - Aritméticas: Suma, resta, multiplicación, división.
 - Lógicas: AND, OR, XOR, NOT.
 - Transferencia de datos: MOV (muestra datos entre registros y memoria).
- **Cantidad de registros:** Algunos procesadores tienen más registros para mejorar la eficiencia.
- **Formato de instrucciones:** Puede ser de longitud fija o variable.





RISC vs CISC



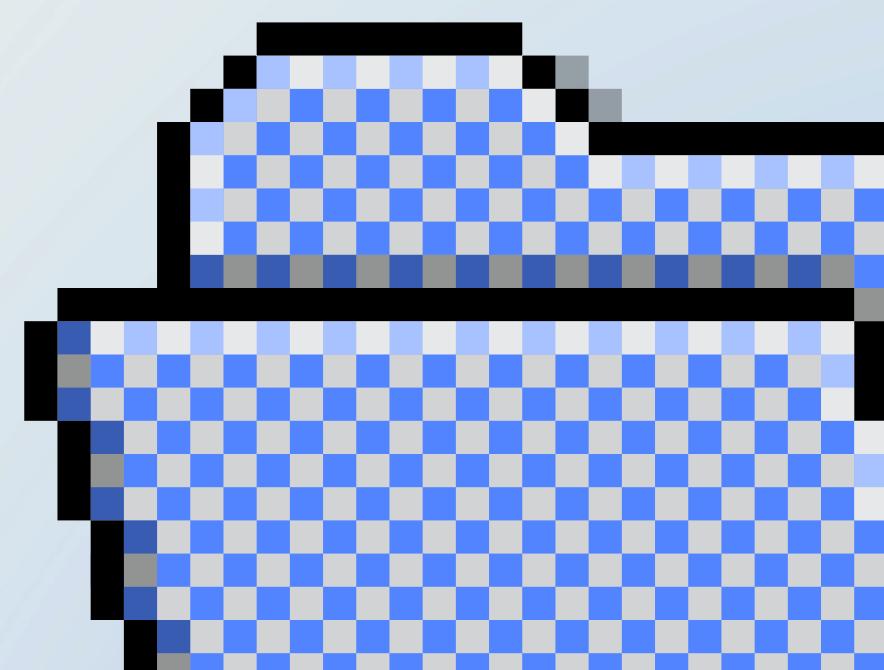
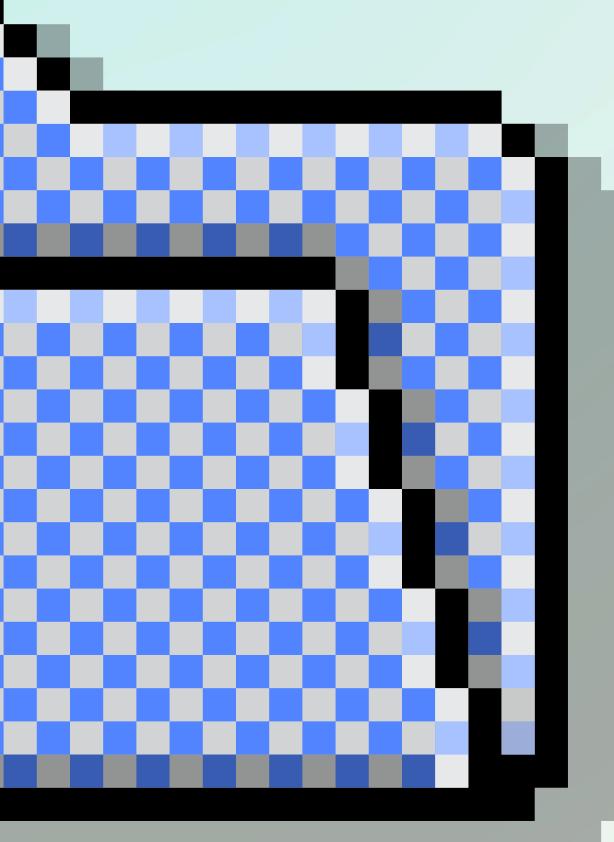
RISC (Reduced Instruction Set Computing):

- Instrucciones simples y rápidas.
- Menos modos de direccionamiento.
- Arquitecturas ARM, MIPS.

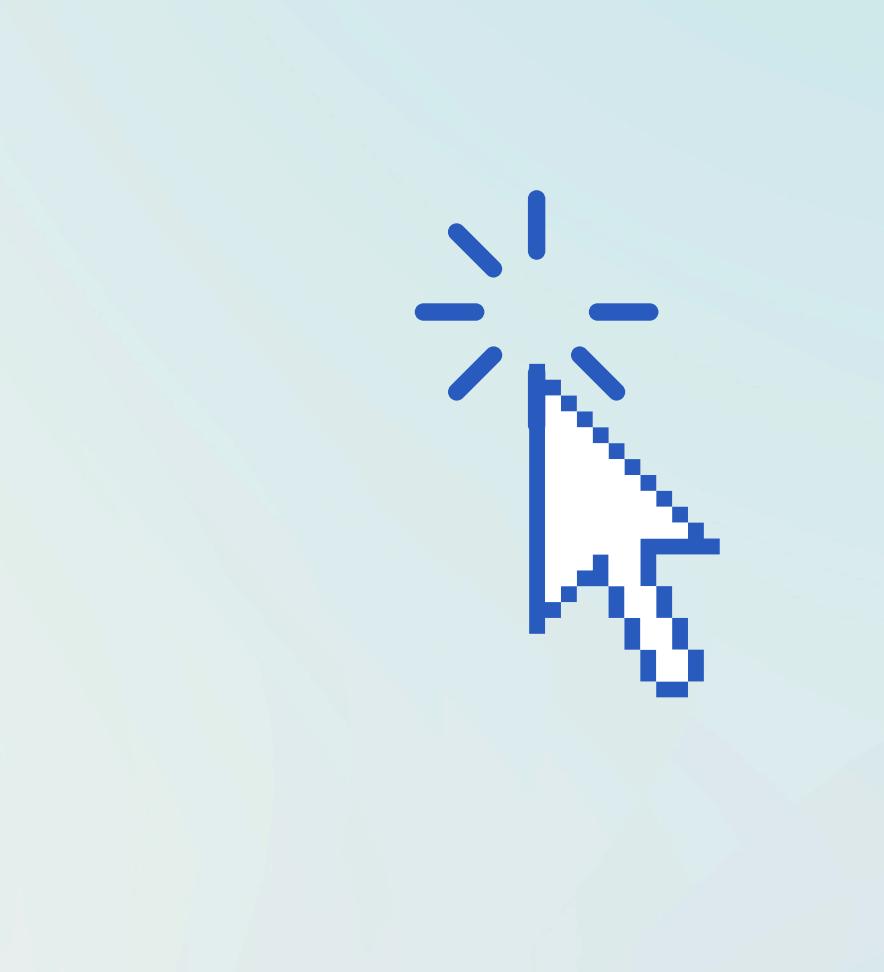
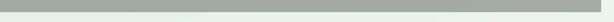
CISC (Complex Instruction Set Computing):

- Instrucciones más complejas y variadas.
- Mayor uso de memoria y modos de direccionamiento.
- Arquitectura x86 (Intel, AMD).





2.3.4 Modos de direcccionamiento y formatos.



¿Que es?



- Los modos de direccionamiento son las diferentes maneras en las que un procesador puede acceder a los datos en memoria o registros. Determinan cómo se calcula la dirección del dato que se utilizará en una instrucción.
- El formato de instrucción define la estructura de una instrucción en un procesador. Especifica cómo están organizados los diferentes campos dentro de una instrucción.

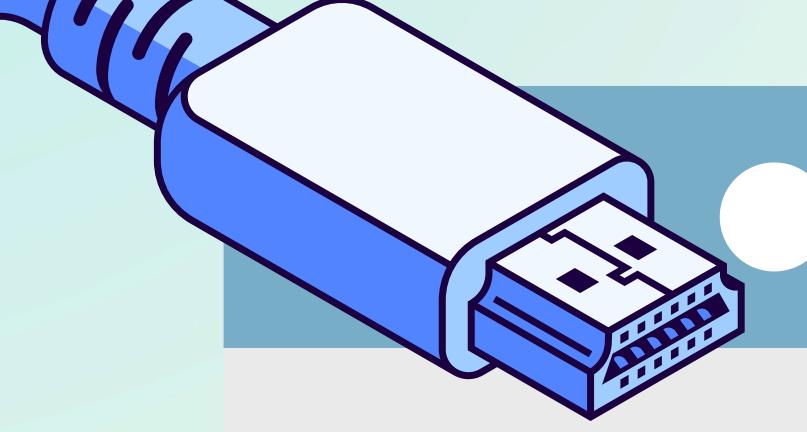




Modos de Direcccionamiento



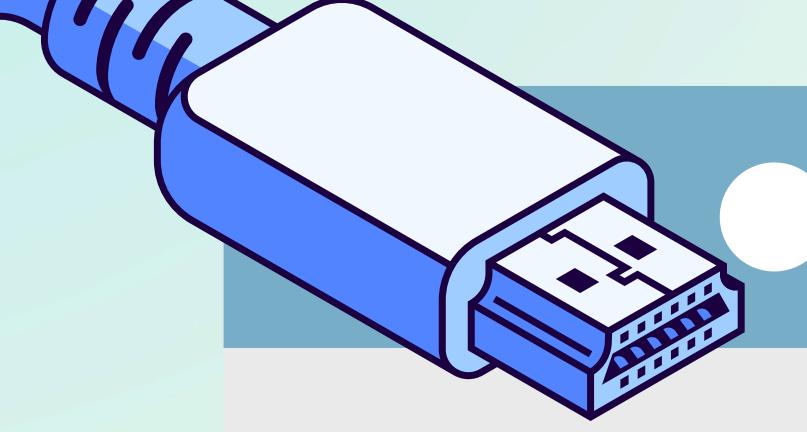
- **Direccionamiento inmediato:** El operando está incluido en la propia instrucción.
 - **Direccionamiento directo o absoluto:** La dirección de memoria donde se encuentra el operando está especificada en la instrucción.
 - **Direccionamiento indirecto:** La dirección del operando se encuentra almacenada en un registro o en otra ubicación.
- 



Modos de Direcccionamiento



- **Direccionamiento por registro:** El operando se encuentra en un registro del procesador.
 - **Direccionamiento relativo:** Se usa para instrucciones de salto y se basa en un desplazamiento respecto a la posición actual.
 - **Direccionamiento indexado:** Se usa un registro índice más un desplazamiento para calcular la dirección del operando.
- 



Tipos de formatos



- **Formato de longitud fija:** Todas las instrucciones tienen el mismo tamaño.
 - **Formato de longitud variable:** Las instrucciones pueden tener diferentes tamaños según su complejidad.
 - **Formato extendido:** Permite incluir más operandos o modos de direccionamiento adicionales.
- 

2.4 Casos de estudio de CPU reales





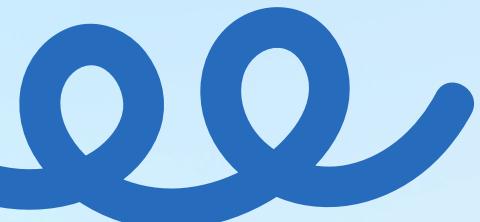
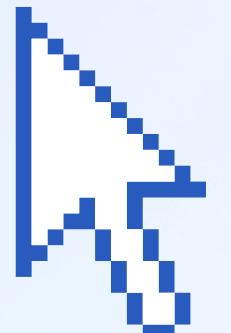
1. Intel 8086 (1978) – El inicio de la arquitectura x86

Características clave:

- Arquitectura de 16 bits
- Frecuencia de reloj de 5-10 MHz
- 29,000 transistores
- Conjunto de instrucciones CISC

Importancia:

- Sentó las bases de la familia x86, que sigue dominando el mercado.
- Introdujo el concepto de *segmentación de memoria*, permitiendo gestionar más de 64 KB de RAM.





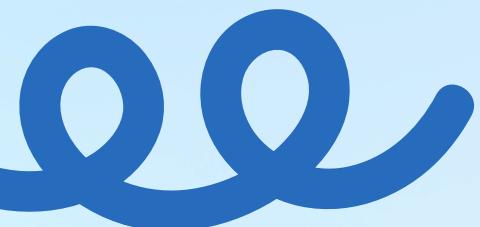
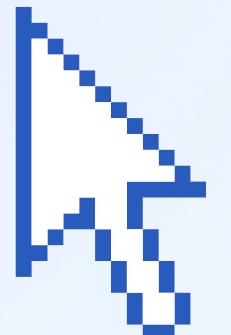
2. Intel Pentium (1993) – Avance en rendimiento y ejecución superscalar

Características clave:

- Arquitectura de 32 bits
- Frecuencia de hasta 66 MHz
- Dos pipelines para ejecución *superscalar*
- Memoria caché L1 integrada

Importancia:

- Introdujo la ejecución fuera de orden y técnicas de predicción de saltos para mejorar el rendimiento.
- Popularizó la computación personal de alto rendimiento.





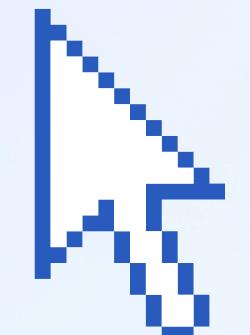
3. AMD Athlon 64 (2003) – Primer CPU de 64 bits para consumidores

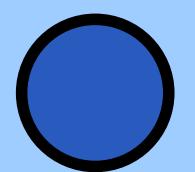
Características clave:

- Arquitectura x86-64 (compatible con 32 y 64 bits)
- HyperTransport para comunicación rápida con el chipset
- Controlador de memoria integrado en el procesador
- Frecuencia de hasta 2.4 GHz

Importancia:

- Permitió la transición a sistemas operativos y software de 64 bits.
- Eliminó el cuello de botella en la comunicación con la RAM gracias al controlador integrado.





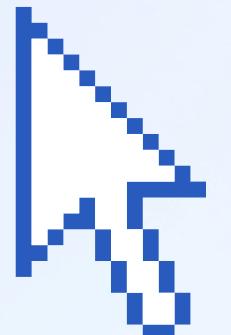
4. Intel Core i7-8700K (2017) – Más núcleos y mayor eficiencia

Características clave:

- 6 núcleos y 12 hilos (Hyper-Threading)
- Frecuencia base de 3.7 GHz, turbo hasta 4.7 GHz
- Proceso de fabricación de 14 nm
- Cache L3 de 12 MB

Importancia:

- Responde a la creciente demanda de *procesamiento paralelo* en aplicaciones modernas.
- Mejora en eficiencia energética con tecnologías como Speed Shift y Turbo Boost.



5. Apple M1 (2020) – Cambio a arquitectura ARM en computadoras personales



Características clave:

- Basado en arquitectura ARM en lugar de x86
- 8 núcleos de CPU (4 de alto rendimiento, 4 de eficiencia)
- Integración de CPU, GPU y memoria en un *SoC (System on Chip)*
- Alto rendimiento con bajo consumo de energía

Importancia:

- Supone una revolución en la industria al mostrar que ARM puede competir con x86 en rendimiento.
- Gran eficiencia energética, lo que permite mayor duración de batería en dispositivos portátiles.

