



Software para sistemas embebidos

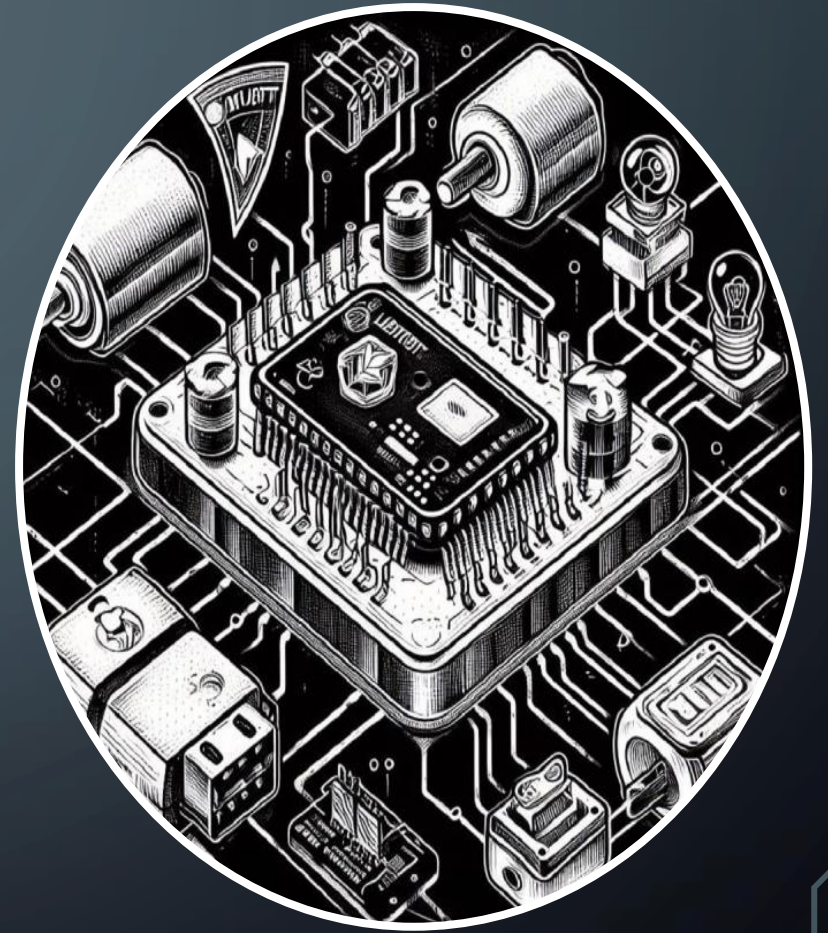
IEO. Vera Z. Pérez Ariza Ph.D.

vera.perez@upb.edu.co

IEO. Hernán D. Patarroyo S.

hernan.patarroyo@upb.edu.co

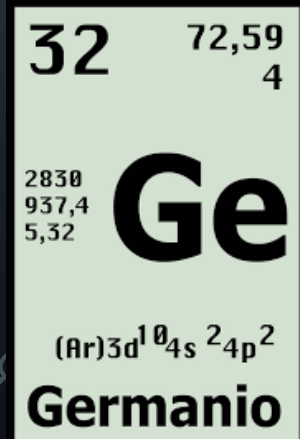
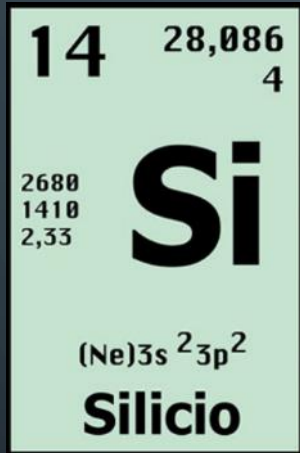
I. ¿QUÉ ES UN MICROCONTROLADOR?



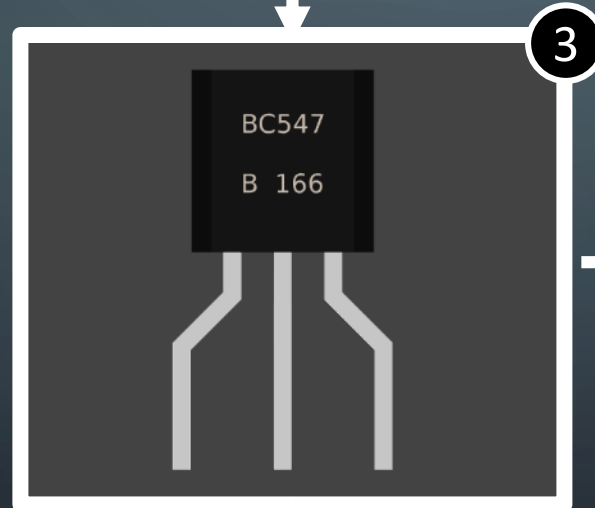
EL ORIGEN DE LA ELECTRÓNICA



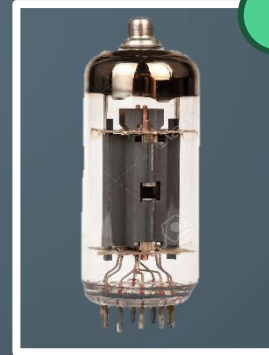
Semiconductores



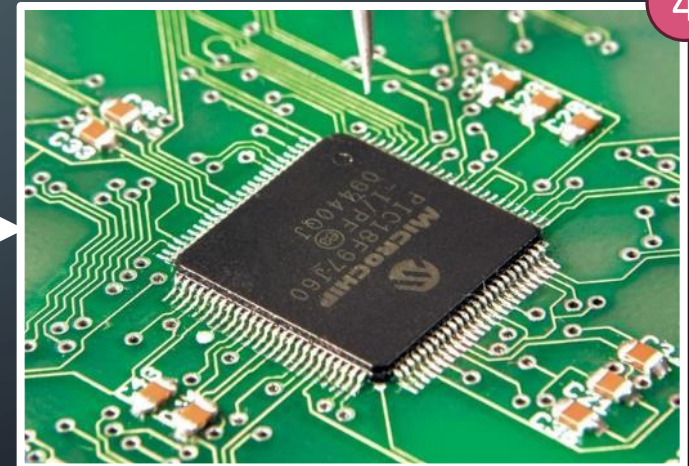
TRANSISTOR



Tubo de vacío



Tubo de rayos
catódicos - TRC



Circuito Integrado (IC) - Chip



AYUDA

1

2

3

4

Un circuito integrado (CI), también conocido como chip o microchip, es una estructura de pequeñas dimensiones de material semiconductor, normalmente silicio, de algunos milímetros cuadrados de superficie, sobre el que se fabrican circuitos electrónicos generalmente mediante fotolitografía y que está protegida dentro de un encapsulado plástico o cerámico. Además, el encapsulado posee conductores metálicos apropiados para hacer conexión entre el circuito integrado y un circuito impreso. Los CI tienen dos principales ventajas sobre los circuitos discretos: costo y rendimiento.

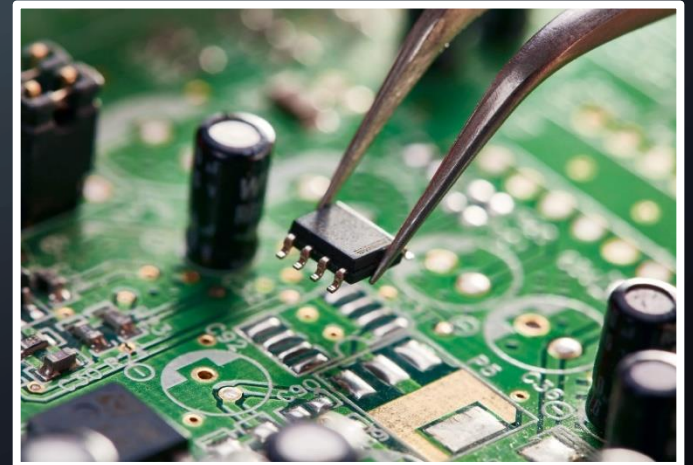
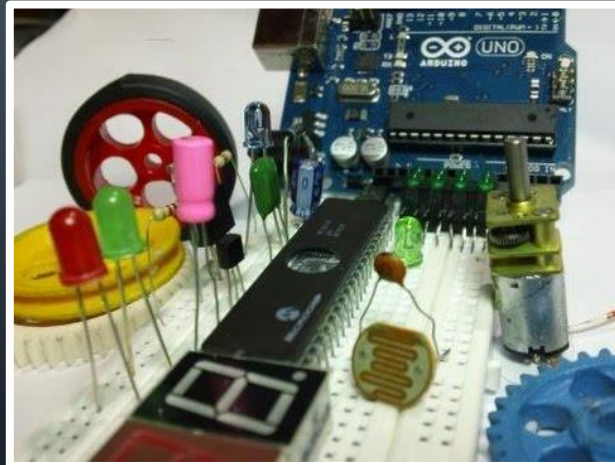
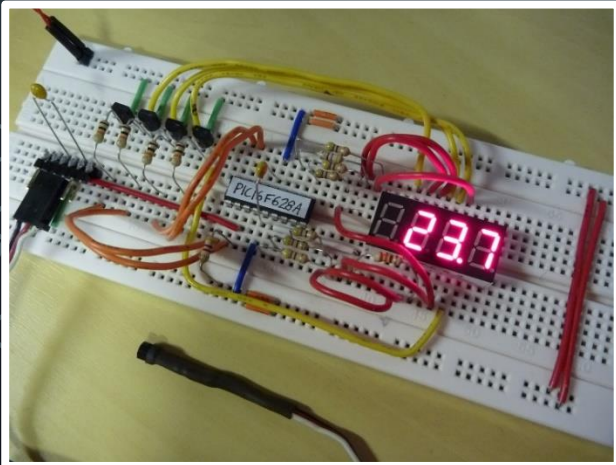
Desde 2012, el intervalo de área de chips típicos es desde unos pocos milímetros cuadrados hasta unos 450 mm^2 , con incluso nueve millones de transistores por mm^2 .



¿QUÉ ES LA ELECTRÓNICA?



“La electrónica es una rama de la física aplicada que comprende la física, la ingeniería, la tecnología y las aplicaciones que tratan con la emisión, el flujo y el control de los electrones u otras partículas atómicas cargadas eléctricamente en el vacío y/o la materia. La identificación del electrón en 1897, junto con la invención del tubo de vacío inauguraron el campo de la electrónica y la era del electrón”.

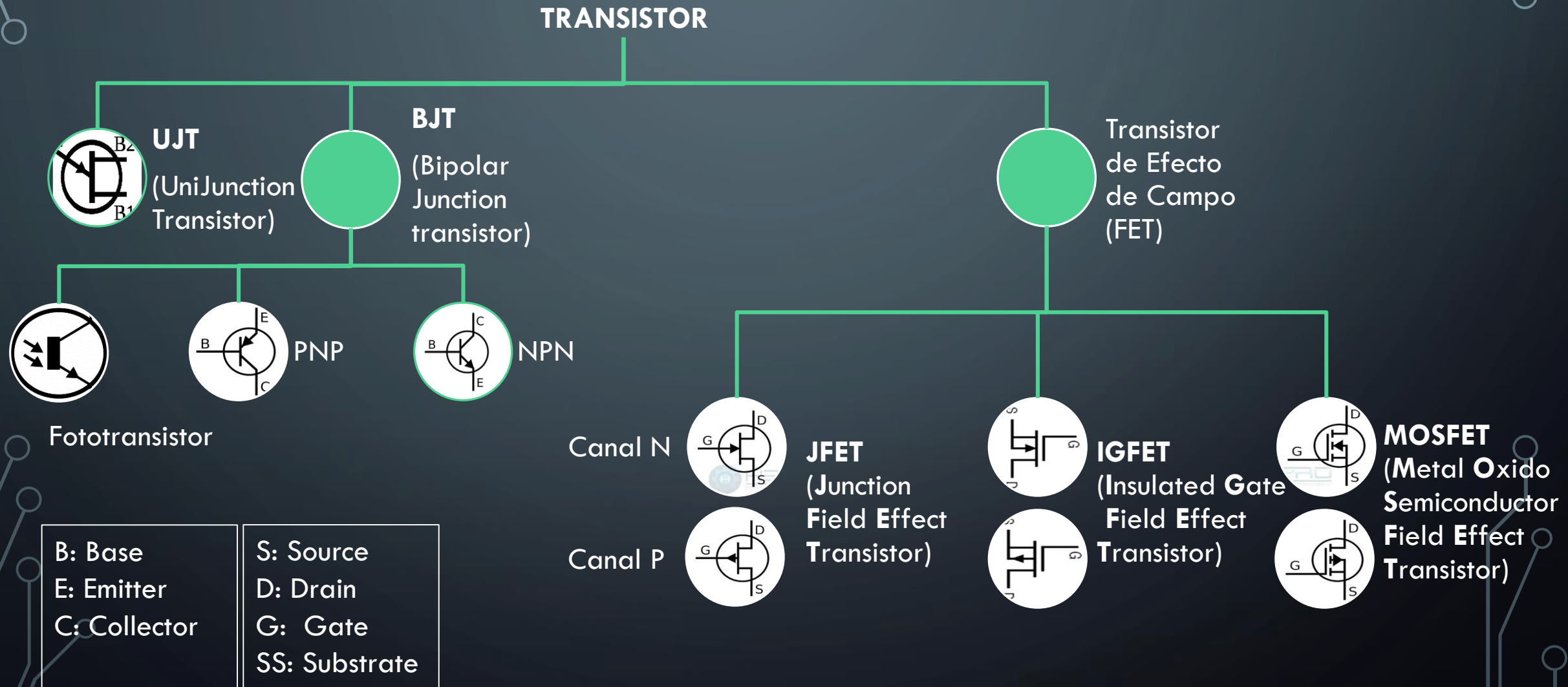


¿QUÉ ES LA ELECTRÓNICA?

- El comportamiento no lineal de los componentes activos y su capacidad para controlar el flujo de electrones hace posible la amplificación de señales débiles.
- La capacidad de los dispositivos electrónicos para actuar como interruptores hace posible el procesamiento digital de la información.
- Las tecnologías de interconexión, como los circuitos impresos, más la tecnología de encapsulado electrónico y otras formas variadas de infraestructuras de comunicación, completan la funcionalidad del circuito electrónico y transforman los componentes electrónicos en un sistema de trabajo, llamado sistema electrónico; son ejemplos las computadoras, microcontroladores, los sistemas de control, etc.
- Un **sistema electrónico** puede ser un dispositivo independiente o un componente de otro sistema diseñado.
- La *electrónica* es ampliamente utilizada en: el procesamiento de datos, las telecomunicaciones, el procesamiento de señales, entre otras aplicaciones.

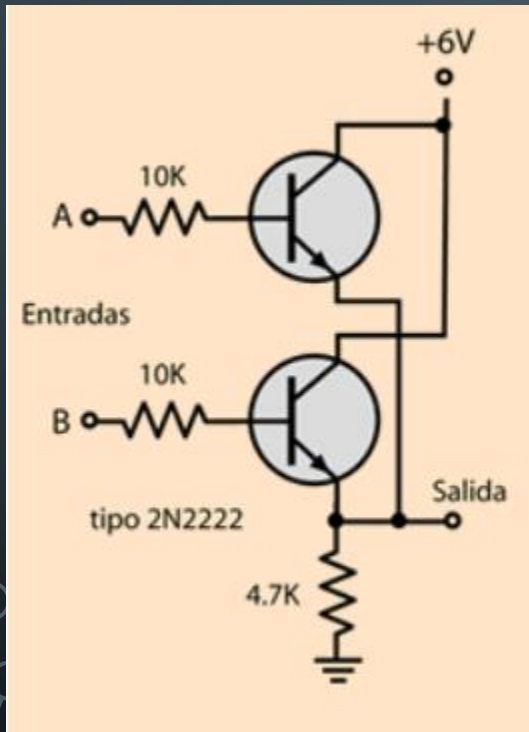


CLASIFICACIÓN DE LOS TRANSISTORES

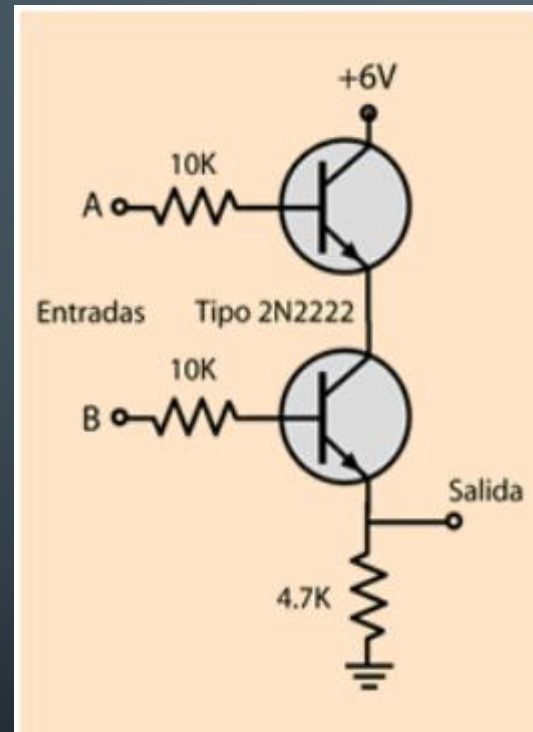


COMPUERTAS LÓGICAS

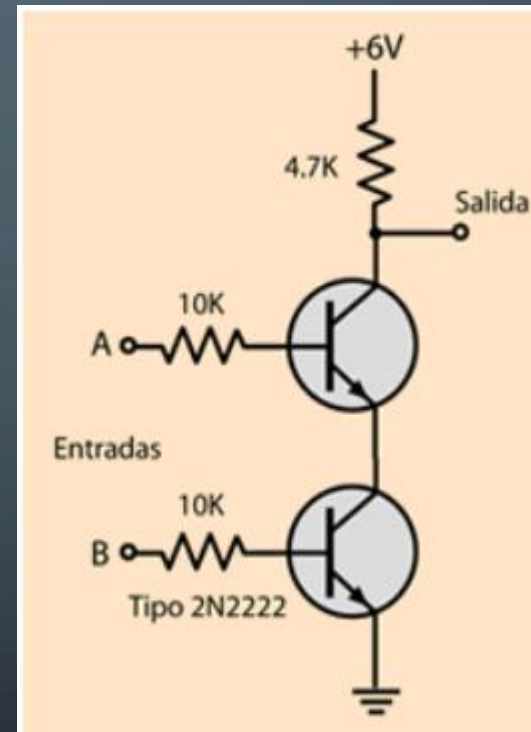
TRANSISTOR



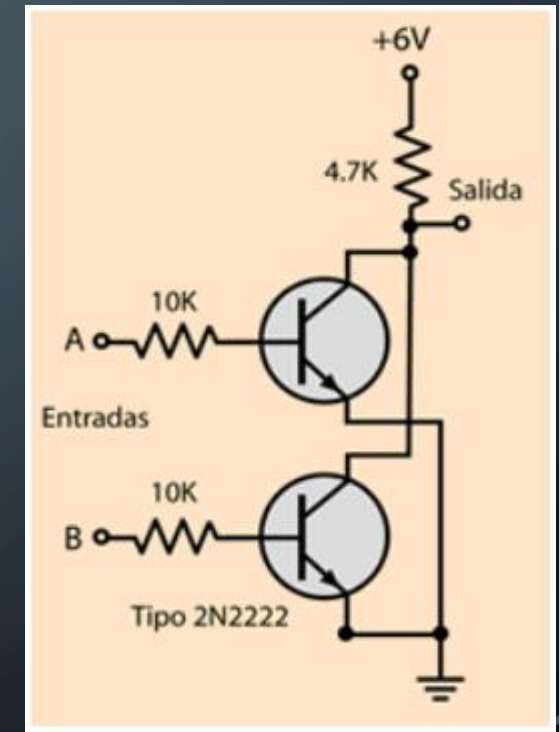
**Compuerta
OR**



**Compuerta
AND**



**Compuerta
NAND**



**Compuerta
NOR**

COMPUERTAS LÓGICAS

Compuerta
AND



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$Y = A.B$$

Compuerta
OR



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$$Y = A+B$$

COMPUERTAS LÓGICAS

Compuerta
NOT



A	Y
0	1
1	0

$$Y = \bar{A}$$

Compuerta
XOR



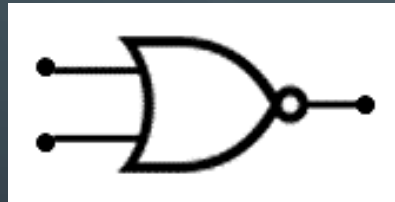
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$Y = A \oplus B$$

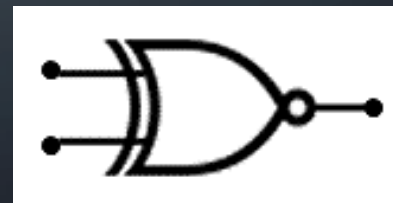
COMPUERTAS LÓGICAS



A	B	Y
0	0	
0	1	
1	0	
1	1	



A	B	Y
0	0	
0	1	
1	0	
1	1	



¿Cuál sería el valor de la salida para c/u de las compuertas



A	B	Y
0	0	
0	1	
1	0	
1	1	

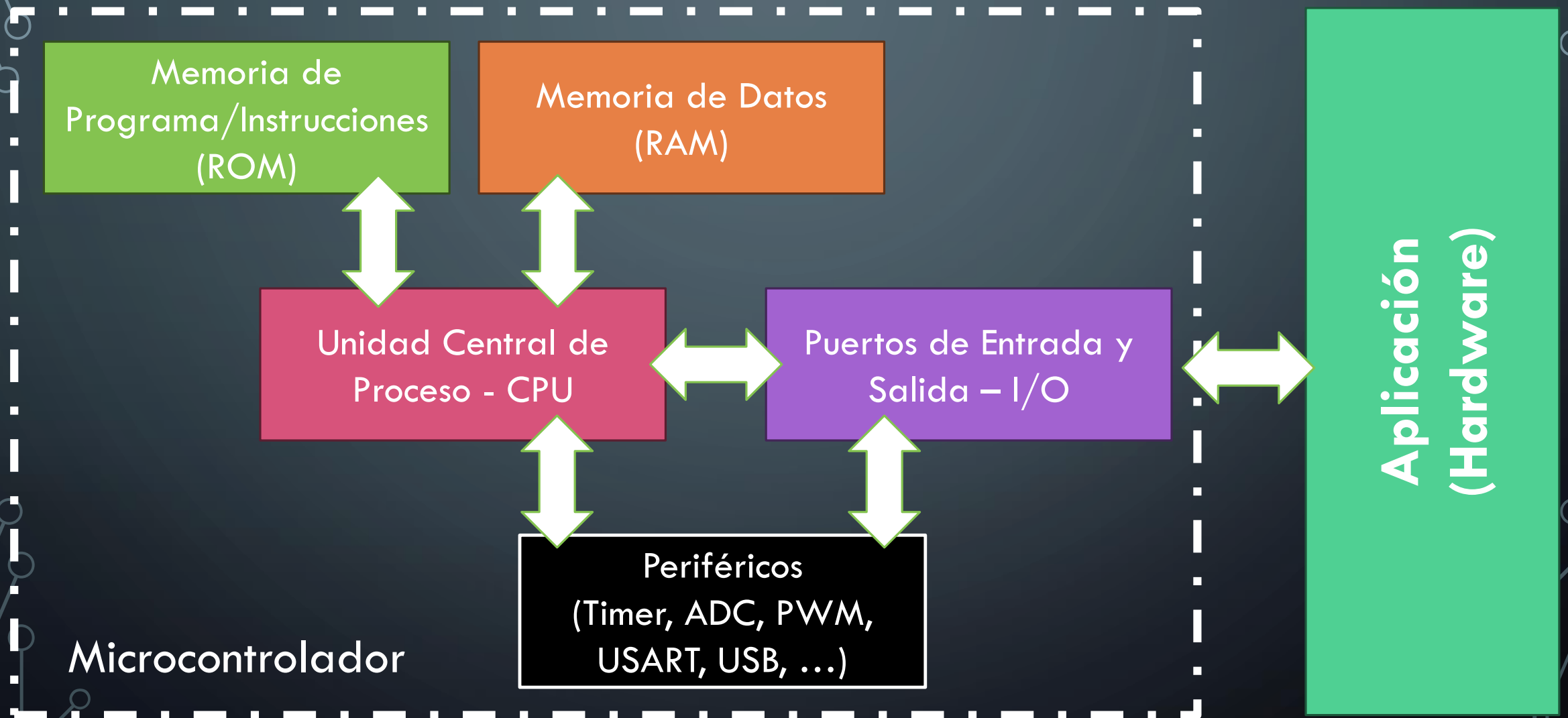
¿QUÉ ES UN MICROCONTROLADOR?



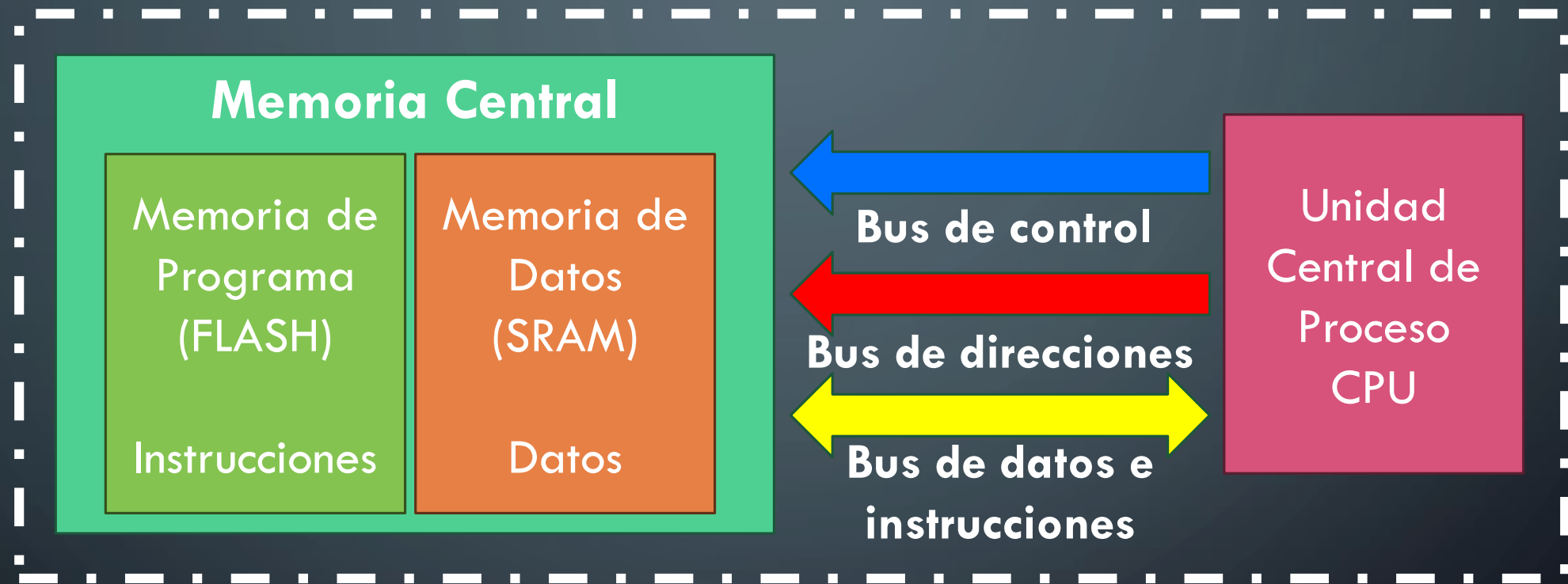
Un **microcontrolador** (abreviado μC o MCU) es un dispositivo electrónico, más específicamente es un circuito integrado, capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados por el usuario utilizando algún lenguaje de programación (assembler o C/C++) y son traducidos y grabados en su memoria a través de un programador.

Un **microcontrolador** está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Particularmente, incluye en su interior las tres principales unidades funcionales de una computadora: i) unidad central de procesamiento, ii) memoria de datos e instrucciones y iii) periféricos de entrada/salida.

ESTRUCTURA GENÉRICA DE UN μ C



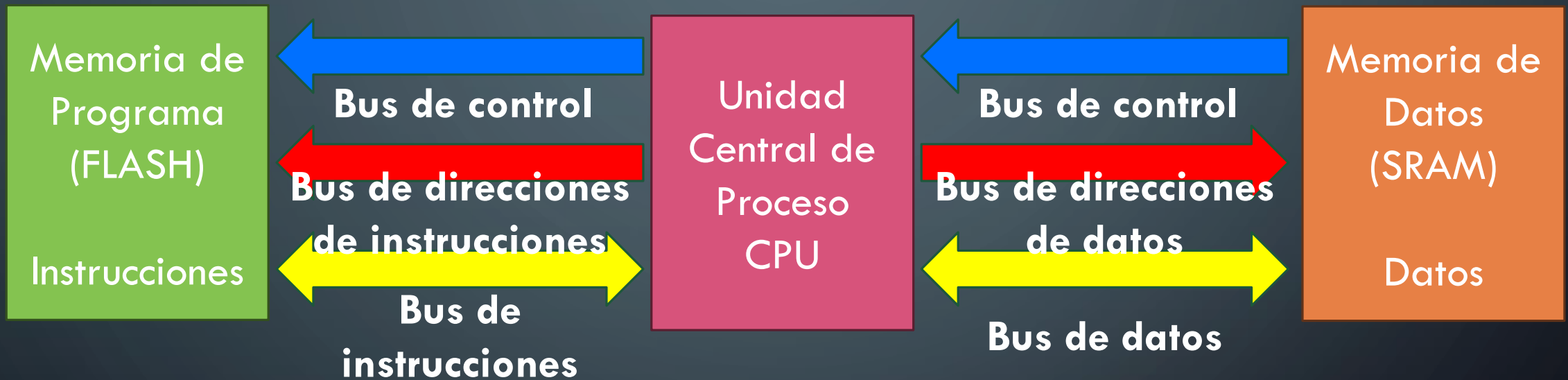
ARQUITECTURAS: VON NEUMANN



La arquitectura de Von Neumann se caracteriza por disponer de una única memoria principal donde se almacenan tanto datos e instrucciones de forma “indistinta”. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control)

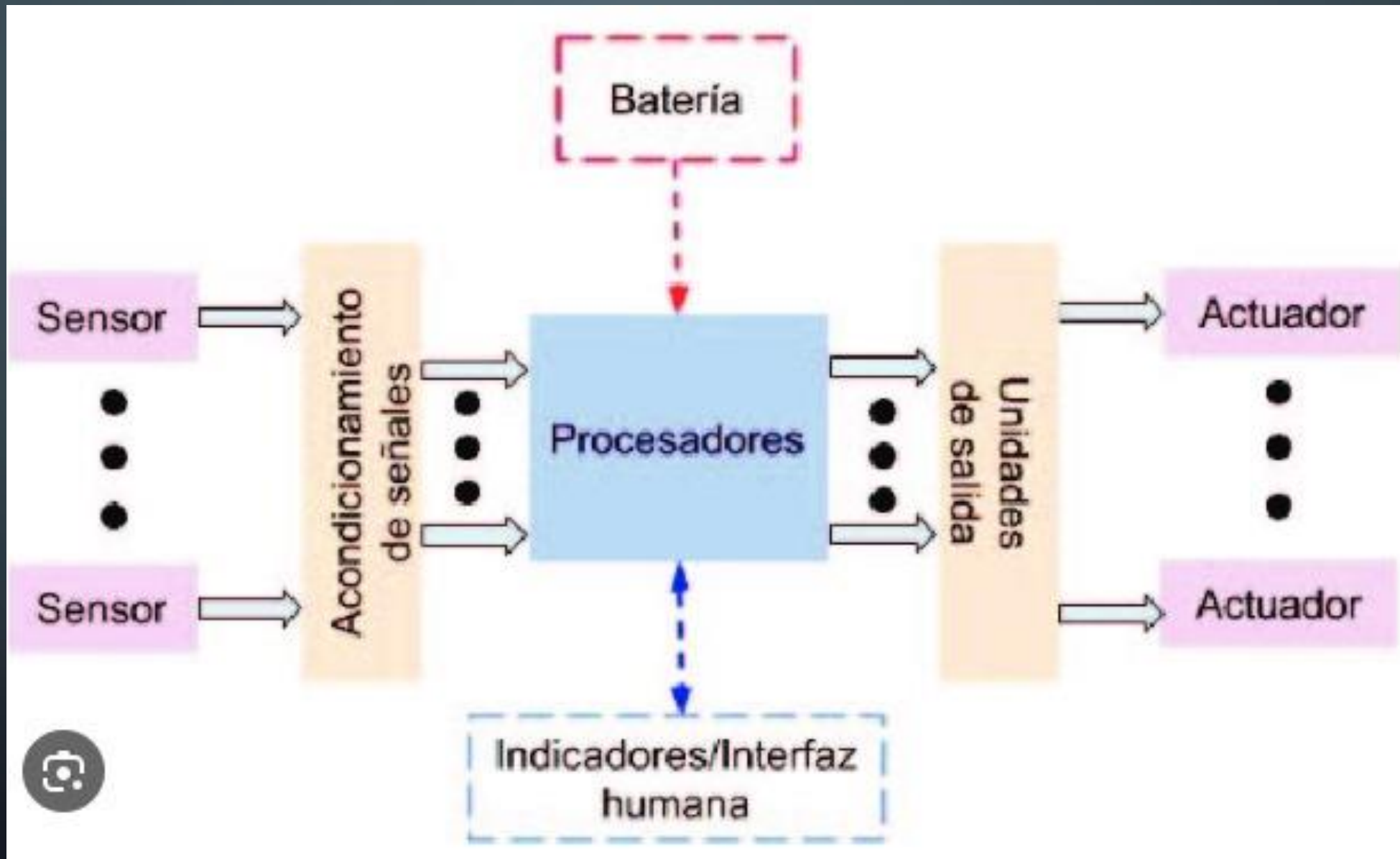


ARQUITECTURAS: HARVARD



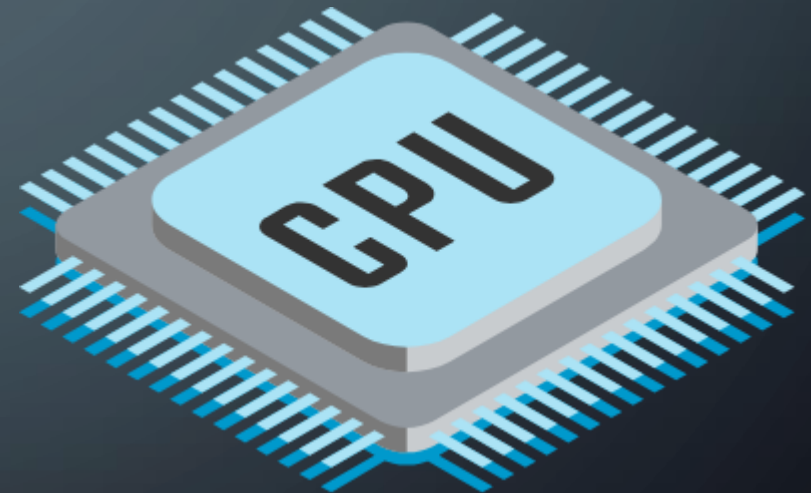
La arquitectura Harvard dispone de dos memorias independientes: una que contiene sólo instrucciones y otra que contiene sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.





EL PROCESADOR - CPU

- ❑ Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como a nivel de software.
- ❑ Se encarga de direccionar la memoria de instrucciones, recibir el código de operación (OP Code) de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

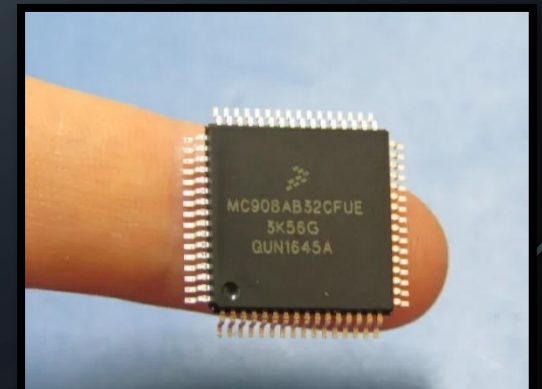
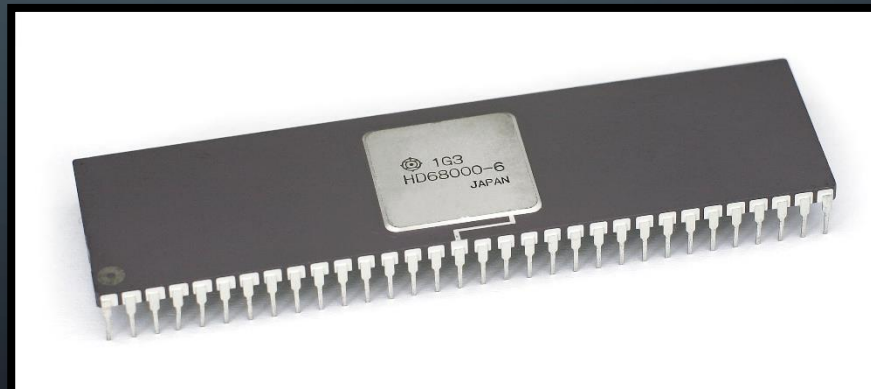


EL PROCESADOR - CPU

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales:

❑ CISC - Computadores de Conjunto de Instrucciones Complejas.

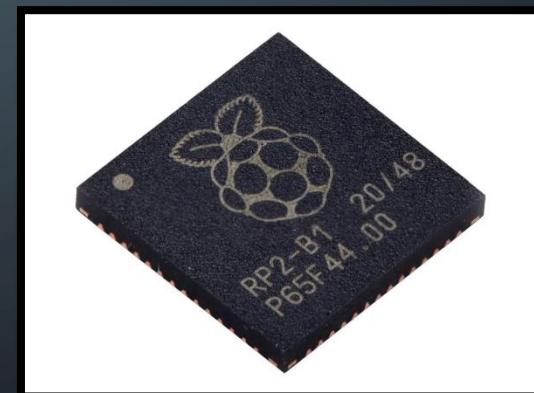
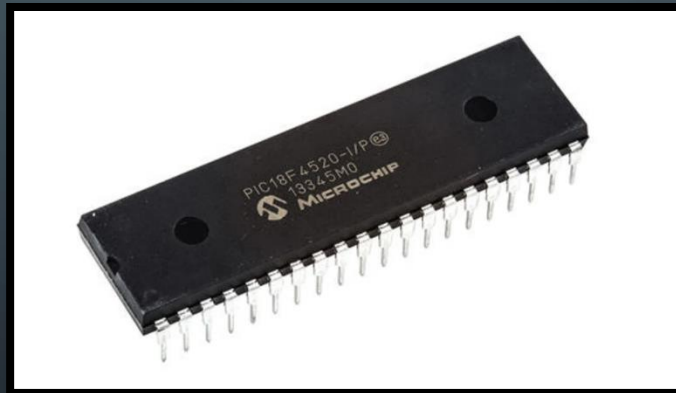
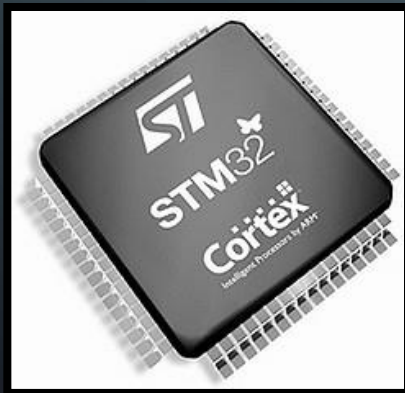
Disponen de un repertorio de más de 80 instrucciones máquina, algunas de las cuales son muy sofisticadas y potentes, requiriendo varios ciclos para su ejecución. Ejemplos: Intel 8051, Intel x86, Motorola 68000, Motorola/Freescale HC08, entre otros.



EL PROCESADOR - CPU

❑ RISC - Computadores de Conjunto de Instrucciones Reducido.

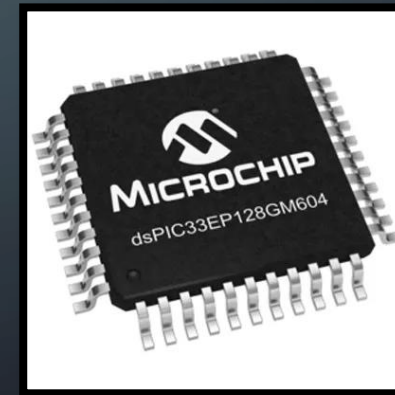
En estos procesadores, el repertorio de instrucciones máquina es limitado y las instrucciones son simples, generalmente ejecutándose en un ciclo. Ejemplos: Serie ARM Cortex-M (como el utilizado en el Raspberry Pi Pico), Microchip PIC, Serie Renesas RX, ESP32 de Espressif Systems, entre otros.



EL PROCESADOR - CPU

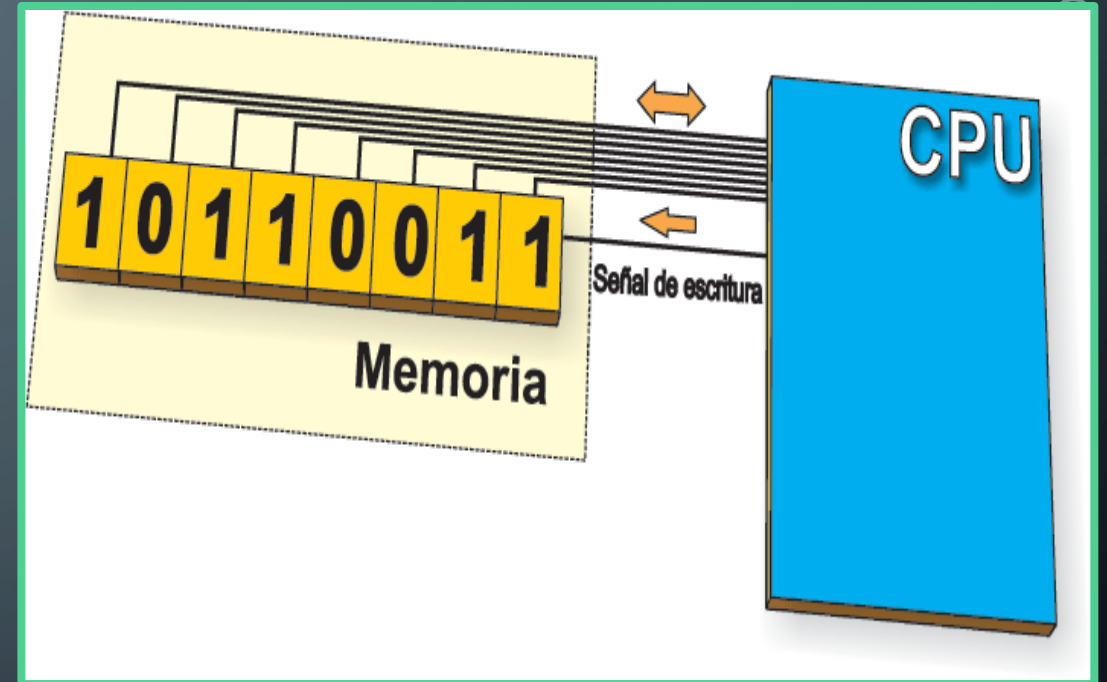
❑ **SISC - Computadores de Conjunto de Instrucciones Específico.**

En los microcontroladores destinados a aplicaciones muy concretas, el conjunto de instrucciones, además de ser reducido, es "específico", adaptándose a las necesidades de la aplicación. Ejemplos: DSP de la serie TMS320 de Texas Instruments o de la serie ADSP de Analog Devices o de la serie dsPIC de Microchip.



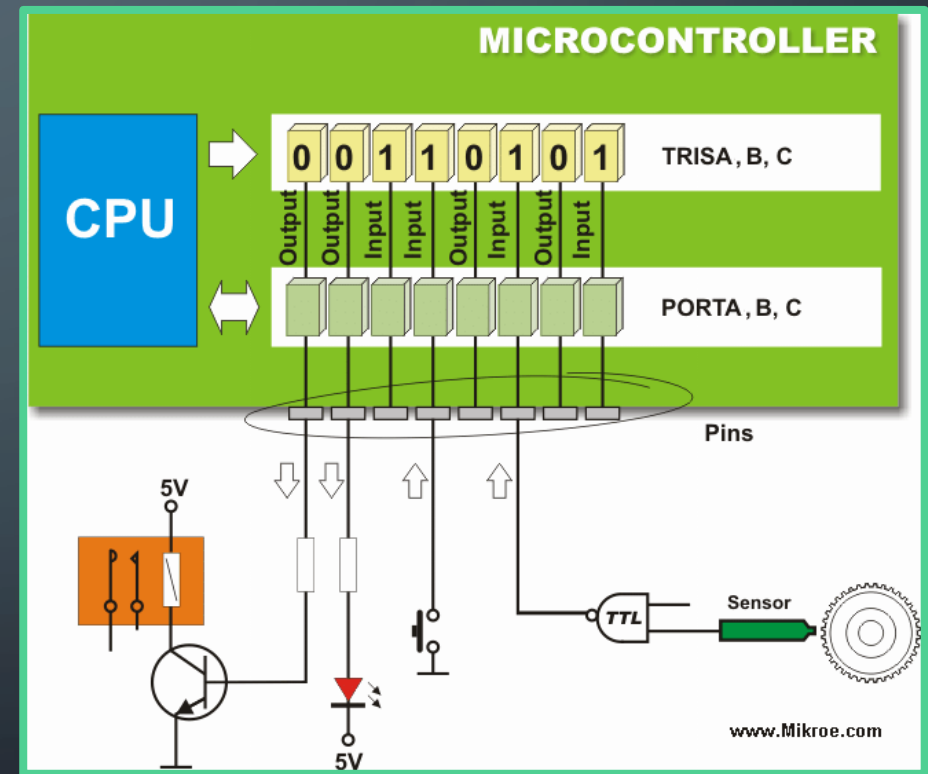
MEMORIA

- ❑ En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.
- ❑ La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa.
- ❑ Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.
- ❑ Las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado son: ROM con máscara, OTP, EPROM, EEPROM y FLASH.



PUERTOS DE ENTRADA Y SALIDA

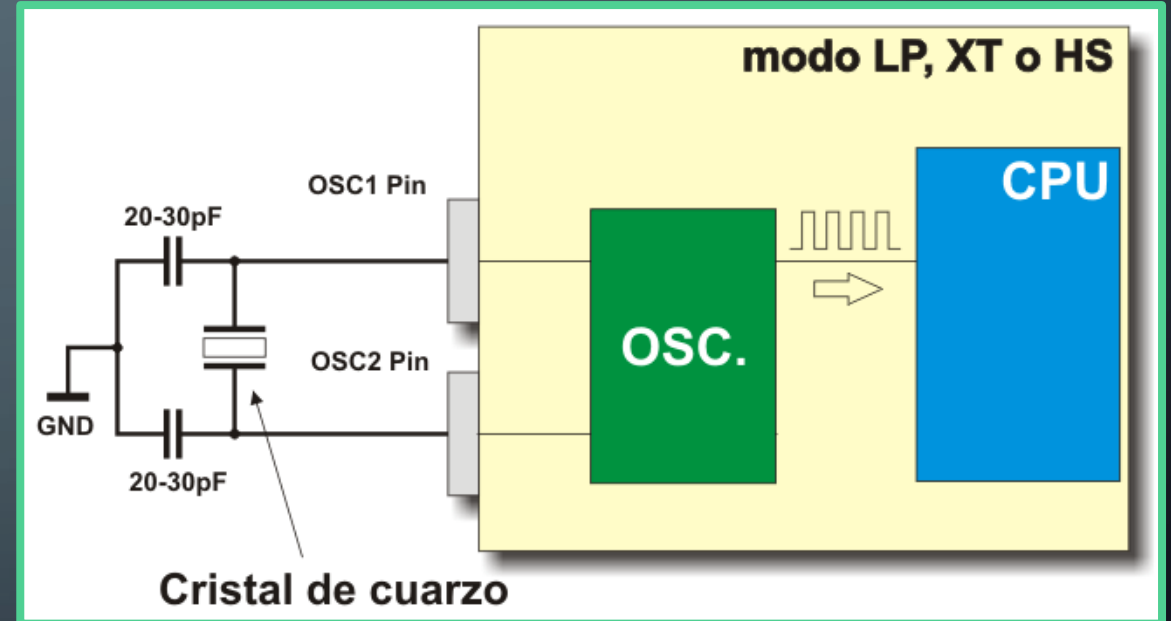
La principal utilidad de los pines que posee el encapsulado que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno (CPU) con los periféricos exteriores.



RELOJ PRINCIPAL

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo (cristal de cuarzo o un resonador cerámico o una red RC).



RECURSOS ESPECIALES

Wath dog, timmers, ...

En edición...

PROGRAMACIÓN DE UN MICROCONTROLADOR

Compilación

Programación

```
int resultado = 5 + 3;
```

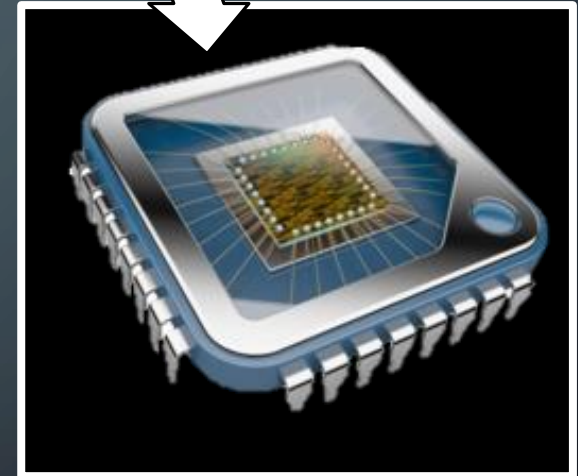
**CÓDIGO
FUENTE**



lenguaje de alto nivel
(C/C++)

0010	0000	1010
1010	1011	1000
0110	1101	0001
1110	1101	0010
0110	0100	1011
1010	0001	0110
1011	1110	1001
0110	0101	1001


**CÓDIGO
BINARIO**




Aplicación

PROGRAMACIÓN DE UN MICROCONTROLADOR


Compilación: El código escrito en el lenguaje de alto nivel (C/C++) se compila utilizando un compilador específico para el uC objetivo. El compilador traduce el código de alto nivel a código de máquina, que está compuesto por instrucciones específicas para el uC en cuestión. Además, el compilador verifica la sintaxis del código fuente y detecta cualquier error sintáctico o de otro tipo que pueda existir en el código fuente. Si se encuentran errores de sintaxis, el compilador genera mensajes de error que indican la naturaleza y la ubicación de los errores en el código fuente.




Generación del archivo binario: El compilador produce un archivo binario, que contiene las instrucciones en forma de unos y ceros. Estas instrucciones están codificadas de acuerdo con el conjunto de instrucciones del microcontrolador específico.



PROGRAMACIÓN DE UN MICROCONTROLADOR



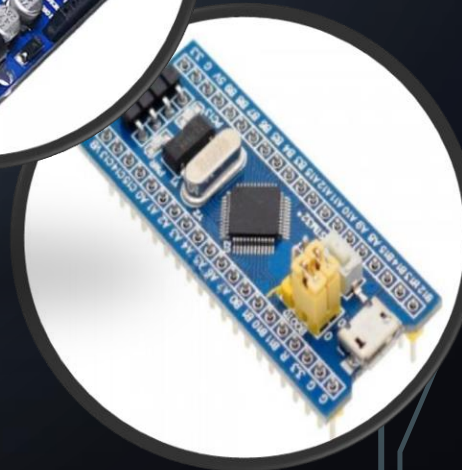
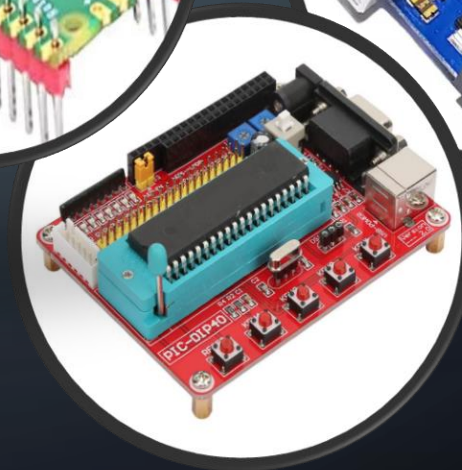
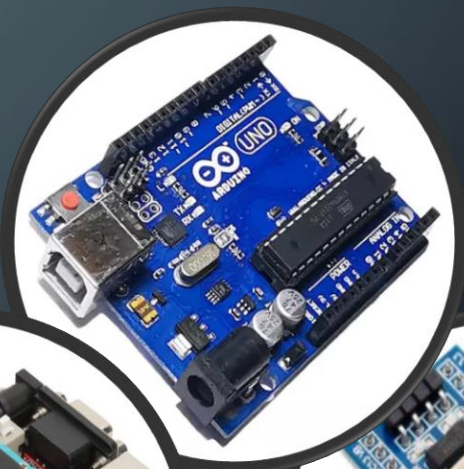
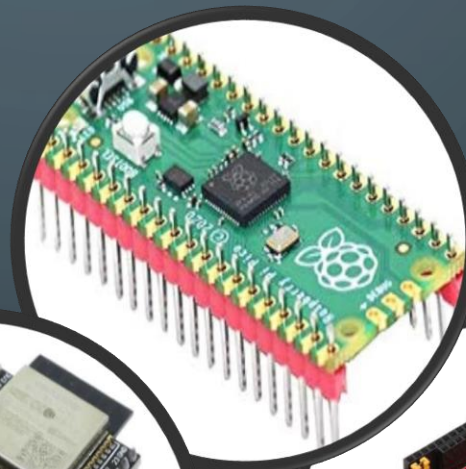
Programación del microcontrolador: El programador utiliza un dispositivo de programación para cargar el archivo binario en la memoria de programa del microcontrolador. Esta memoria es generalmente de tipo no volátil, como la memoria FLASH o la memoria EEPROM, la cual retiene los datos incluso cuando se apaga la energía.



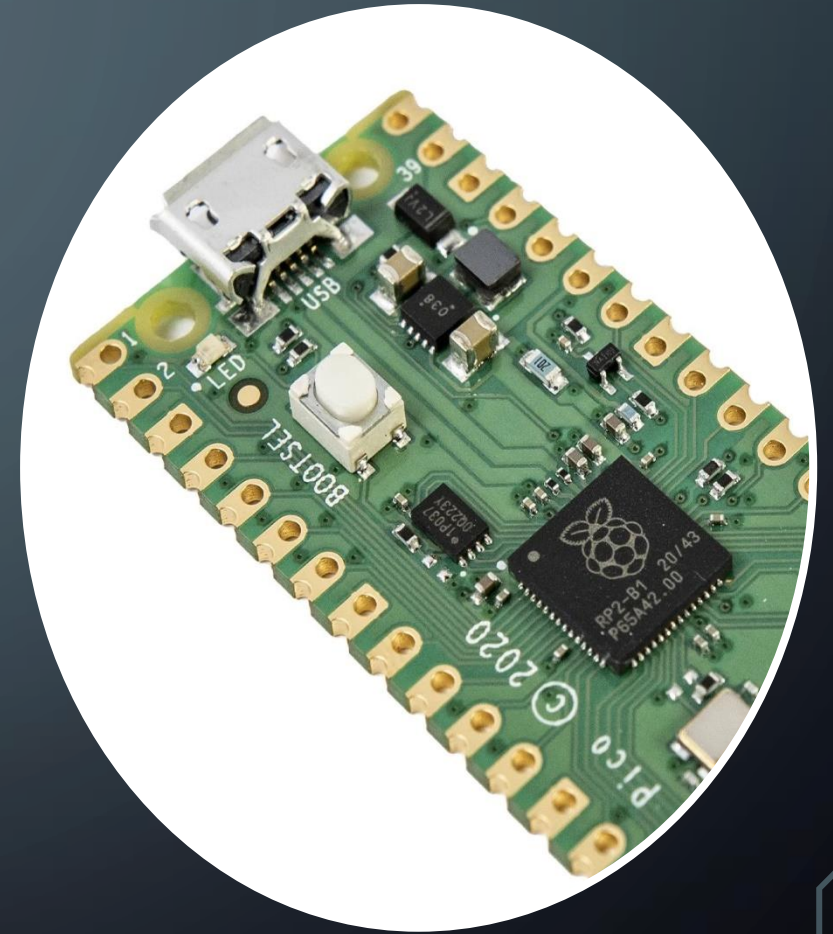
Ejecución de las instrucciones: Una vez que el programa se ha cargado en la memoria de programa del uC, la CPU lee las instrucciones una por una y las ejecuta secuencialmente. Por ejemplo, para sumar 5 y 3, la CPU ejecutaría una serie de instrucciones, que podrían incluir una instrucción de carga de los valores 5 y 3 en registros internos, seguida de una instrucción de suma y finalmente una instrucción de almacenamiento del resultado en la memoria o en otro registro (*acumulador*).

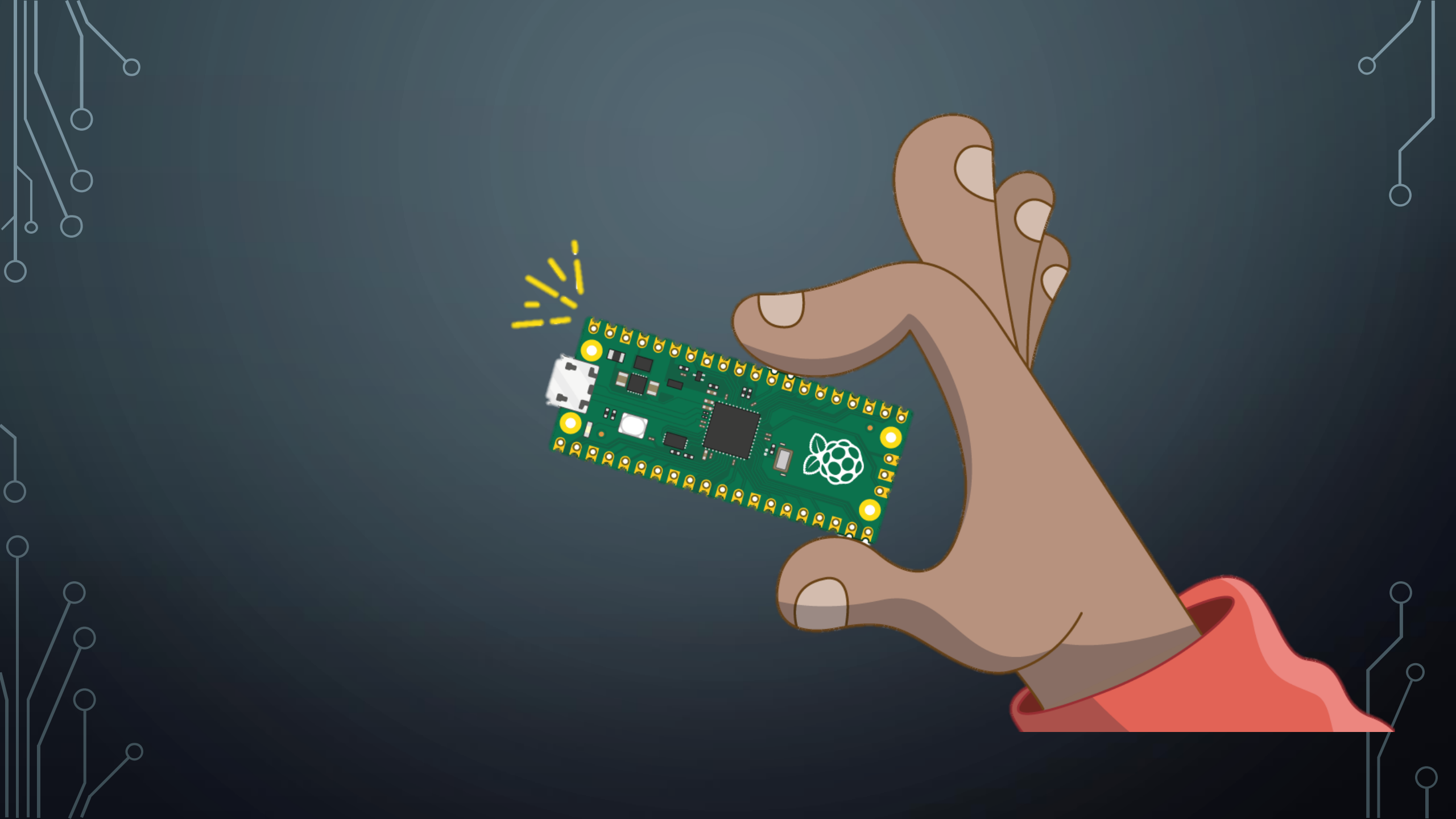
TARJETAS DE ENTRENAMIENTO

Es un dispositivo diseñado para facilitar el aprendizaje y la experimentación con microcontroladores y sistemas embebidos. Por lo general, incluye un microcontrolador o microprocesador, así como una variedad de periféricos, interfaces de entrada y salida, y puertos de conexión para facilitar la programación y el desarrollo de proyectos.

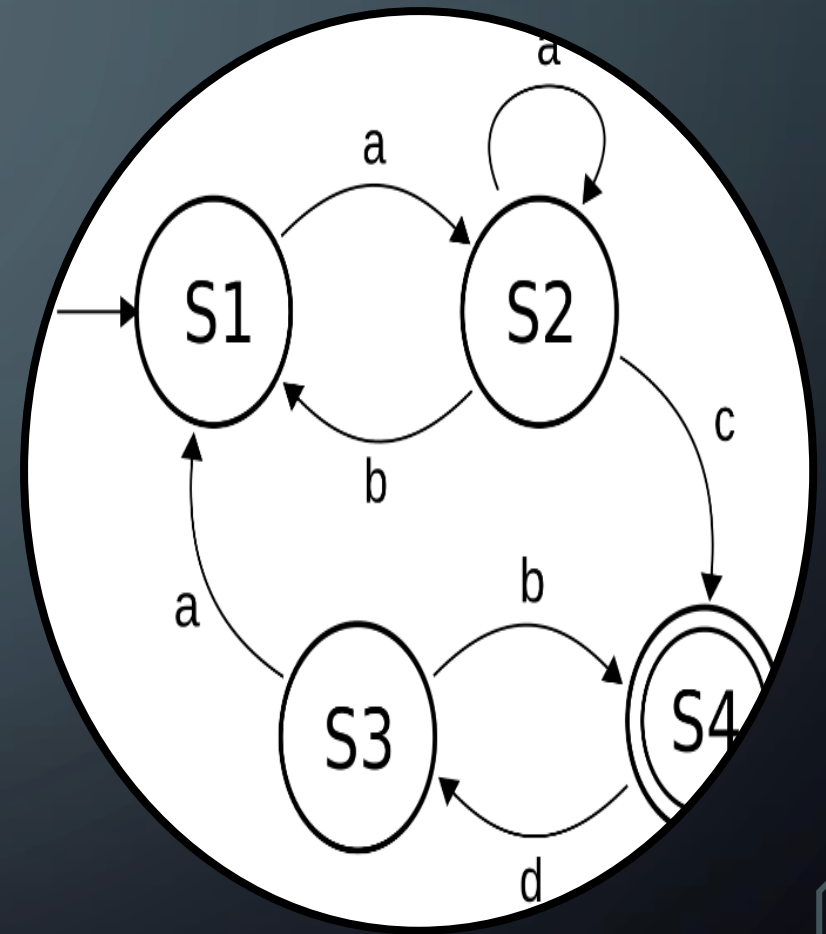


II. EL MICROCONTROLADOR RASPBERRY PI PICO






III. MÁQUINAS DE ESTADO FINITAS - FMS



QUÉ ES UNA MÁQUINA DE ESTADO FINITAS

“Una **Máquina de Estado Finitas** (*FSM – Finite State Machine*) es un modelo matemático (abstracción) que describe el comportamiento de un sistema que puede tener un número finito de estados diferentes en cualquier momento dado (modelo determinístico). Estos **estados** pueden cambiar en respuesta a estímulos externos o internos, conocidos como **eventos**, y están definidos por una serie de **transiciones** entre ellos. Cada transición está asociada con una condición o conjunto de condiciones que deben cumplirse para que ocurra el cambio de estado”.



Es la forma más
sencilla de procesar
la información

QUÉ ES UNA MÁQUINA DE ESTADO FINITA

Las **máquinas de estado finitas** son utilizadas en una amplia gama de aplicaciones, desde sistemas embebidos hasta aplicaciones de software de usuario.

En el desarrollo de software, las FSM se utilizan a menudo para controlar el flujo de un programa, especialmente en aplicaciones donde el comportamiento del sistema es complejo y se pueden identificar claramente los diferentes estados y transiciones entre ellos.

En el desarrollo de videojuegos, las FSM se aplican para controlar el flujo de la lógica del juego y gestionar las interacciones entre los personajes, objetos y eventos del juego. Por ejemplo, se pueden utilizar para modelar el comportamiento de los enemigos, las animaciones de los personajes, el sistema de colisiones, entre otros aspectos clave de la jugabilidad. De esta manera, las FSM permiten crear experiencias de juego dinámicas y envolventes al definir claramente los diferentes estados del juego y las transiciones entre ellos.



EJEMPLO FSM

Requisitos del personaje

Comportamiento del personaje

Transiciones de estado

Transiciones de estado:

El personaje puede estar en uno de varios estados: QUIETO, CORRIENDO, SALTANDO o AGACHADO.

Los eventos ocurren
Los eventos ocurren
Los eventos ocurren

Veamos un ejemplo
para el control de un
personaje en un juego
utilizando FSM



DIAGRAMA DE ESTADOS (ejemplo)



Estado
inicial



QUIETO

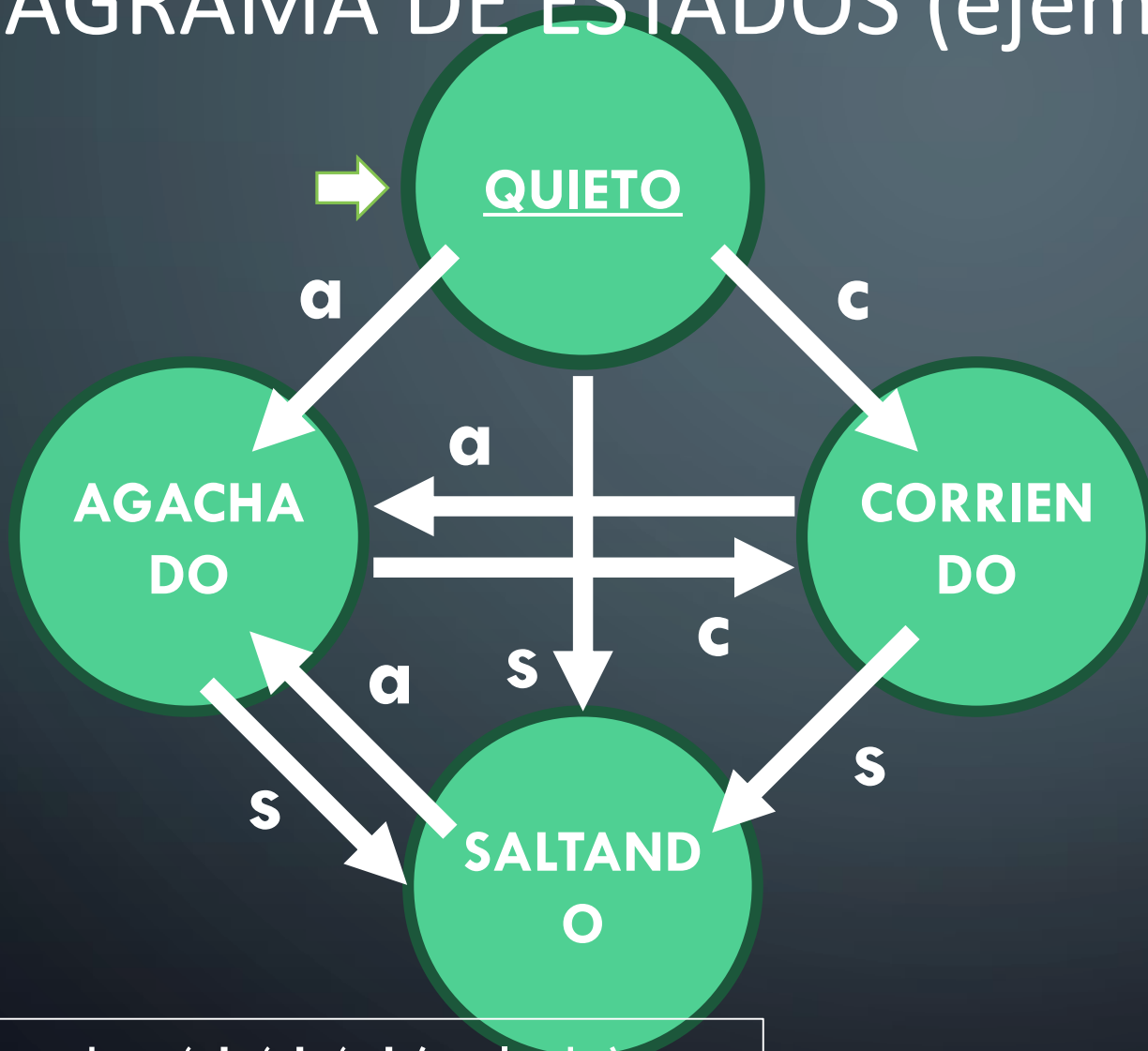
AGACHA
DO

CORRIEN
DO

SALTAND
O

Inicialmente podríamos pensar que una máquina de estados no es necesaria, porque con un 'if', 'else if' 'else' o un 'while' es suficiente. Sin embargo, con una máquina de estados finitos podemos conocer dónde nos encontramos en cada momento, simplificando, optimizando y haciendo más eficiente el código. Nótese que el número de estados es finito (limitado) y cada estado ocurre en un momento determinado.

DIAGRAMA DE ESTADOS (ejemplo)



Entradas: 'c', 's', 'a' (teclado)
Salidas: No hay

Desde el estado QUIETO, el personaje puede pasar a los estados CORRIENDO, SALTANDO o

Desde el estado CORRIENDO, el personaje puede pasar a los estados SALTANDO o AGACHADO.

Desde el estado SALTANDO, el personaje puede pasar al estado AGACHADO.

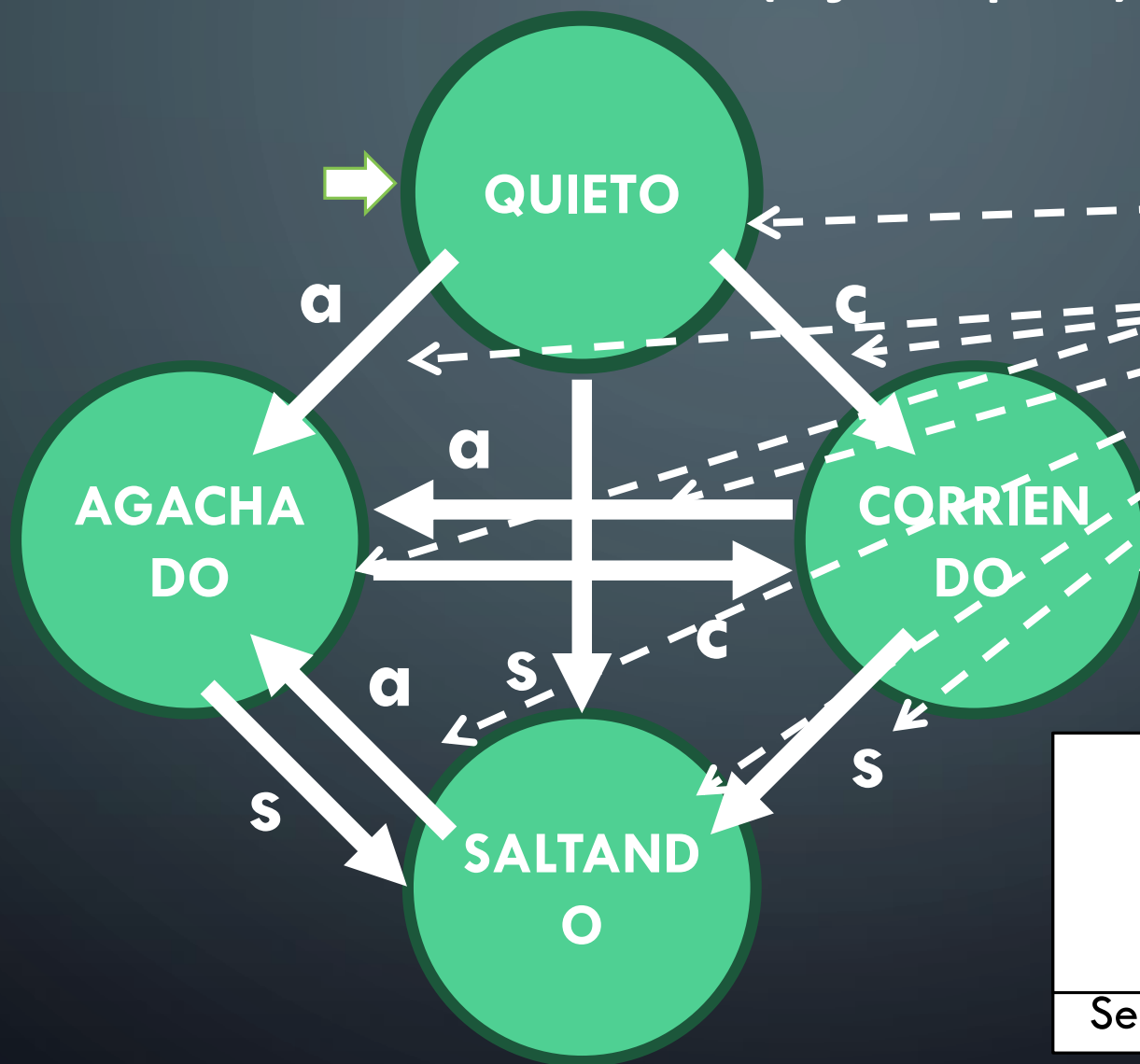
Desde el estado AGACHADO, el personaje puede pasar a los estados CORRIENDO o SALTANDO.

DIAGRAMA DE ESTADOS (ejemplo)



Veamos un
resumen

Estados
Transiciones



Se representan por FLECHAS
Van de un estado inicial a un estado
final
Entrada/Salida

Se debe indicar cual es el estado inicial

DIAGRAMA DE ESTADOS (ejemplo)

Estados:

Son los distintos valores que puede tomar la máquina de estados finitos.

Nomenclatura: Conj. de Estados (Q) = {QUIETO, CORRIENDO, SALTANDO, AGACHADO}

Estado inicial ($Q_0 \in Q$) = {QUIETO}

Conj. de estados finales o de aceptación ($F \subseteq Q$) = $\emptyset = \{ \}$

Eventos:

Son aquellas condiciones (entradas & estado $_{k-1}$) que producen el cambio de un estado a otro.

Nomenclatura: Entrada (Σ) = {a, s, c}

Finalmente



DIAGRAMA DE ESTADOS (ejemplo)

Transiciones:

Se definen como los eventos que cambian los estados. Cada transición consta de tres elementos: un evento, un estado origen y un estado destino. Indica que cuando se produzca el evento si el estado actual es el estado origen este cambiará por el estado destino.

Nomenclatura: Función de transición (δ)

$$\delta: Q \times \Sigma \rightarrow Q$$

Ver Tabla de Estado Siguiente

Finalmente



TABLA DE ESTADO SIGUIENTE

Una **TABLA DEL ESTADO SIGUIENTE** lista en forma tabular cada estado actual de la máquina con su respectivo estado siguiente. Recordemos que el estado siguiente es el estado al que la MSF pasará, desde su estado actual, al aplicar una entrada específica o ante la combinación de las entradas.

Por lo general, la **TABLA DEL ESTADO SIGUIENTE** se obtiene a partir del diagrama de estados (viceversa).

Estado ACTUAL	Entradas			Estado SIGUIENTE
	'a'	's'	'c'	
QUIETO	0	0	1	CORRIENDO
QUIETO	0	1	0	SALTANDO
QUIETO	1	0	0	AGACHADO
CORRIENDO	0	1	0	SALTANDO
CORRIENDO	1	0	0	AGACHADO
SALTANDO	1	0	0	AGACHADO
AGACHADO	0	0	1	CORRIENDO
AGACHADO	0	1	0	SALTANDO

MÁQUINA DE MEALY Y DE MOORE

Máquina de Mealy:

- ❑ La salida está asociada tanto a los estados como a las transiciones de entrada; es decir, la salida de la máquina de estado **depende del estado actual y de la entrada actual**.
- ❑ La función de transición define los cambios de estado y las salidas asociadas con cada transición de entrada.

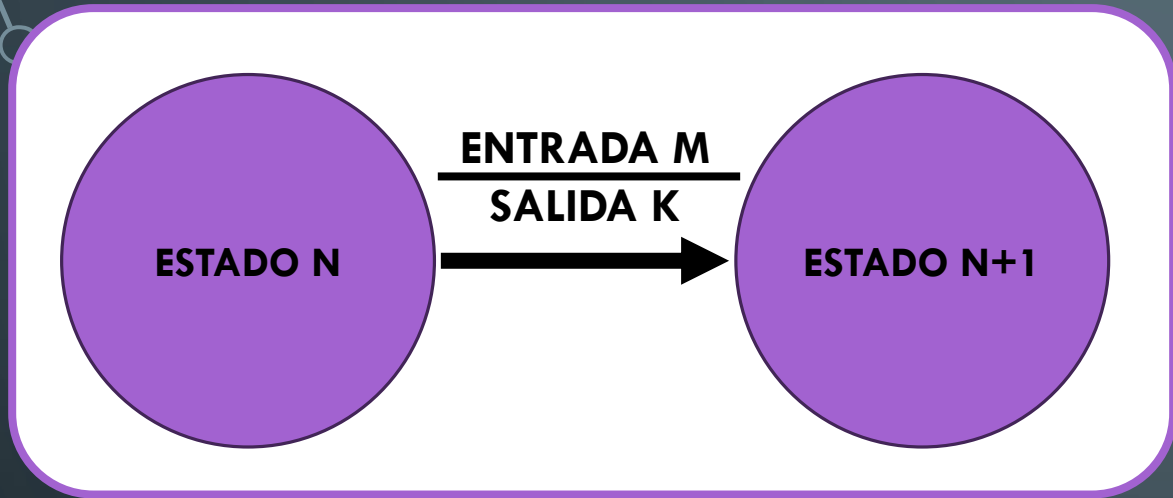
Máquina de Moore:

- ❑ La salida está asociada únicamente a los estados; es decir, **la salida de la máquina de estado depende únicamente del estado actual**.
- ❑ La función de transición define los cambios de estado, y la salida asociada con cada estado se define por separado.

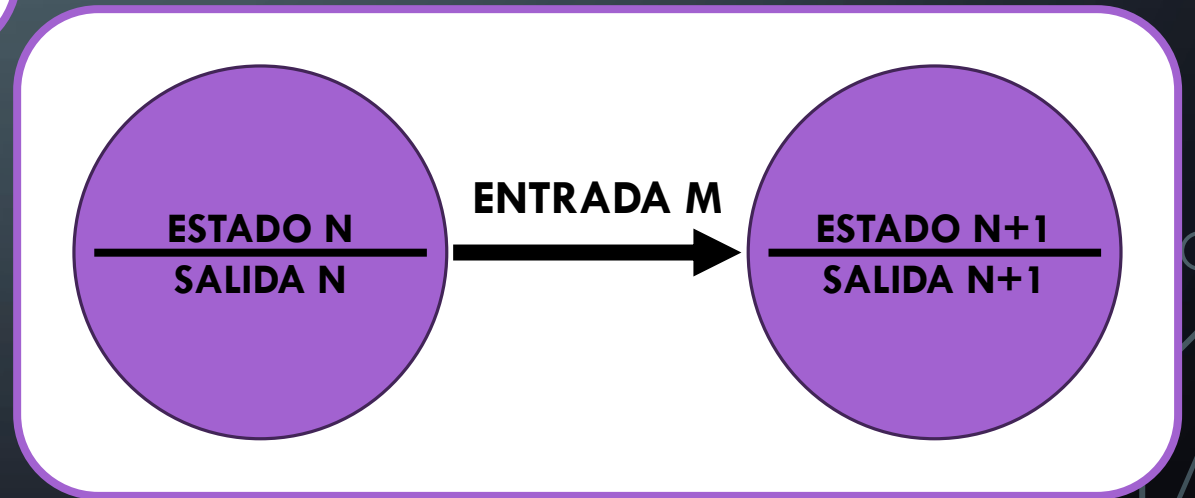
Estos dos modelos difieren en cómo manejan la salida en función de los estados y las entradas



MÁQUINA DE MEALY Y DE MOORE



MÁQUINA DE MEALY



MÁQUINA DE MOORE

MÁQUINA DE MEALY Y DE MOORE

En resumen, en la máquina o el modelo Mealy, la salida se produce inmediatamente después de la transición de estado, mientras que en la máquina o el modelo Moore, la salida se asocia directamente con el estado en el que se encuentra la máquina.

La elección entre estos modelos depende de los requisitos específicos del sistema que se está modelando y del comportamiento deseado de la máquina de estado.

