



Institución
Universitaria
Reacreditada en Alta Calidad

MICROCONTROLADORES

Clase 1 – Introducción a los microcontroladores y lenguajes de programación

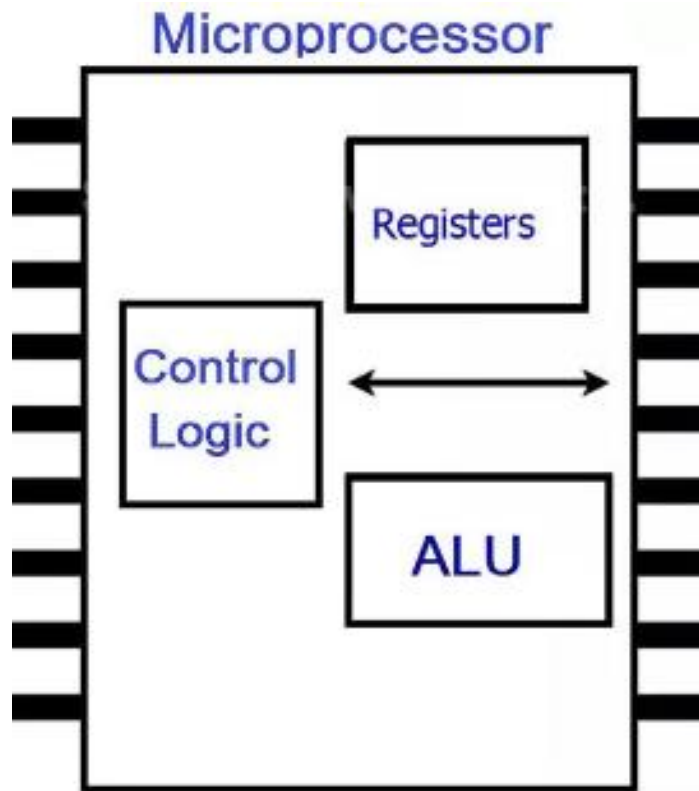
Yomin Jaramillo M.

Somos Innovación Tecnológica con *Sentido Humano*



Alcaldía de Medellín

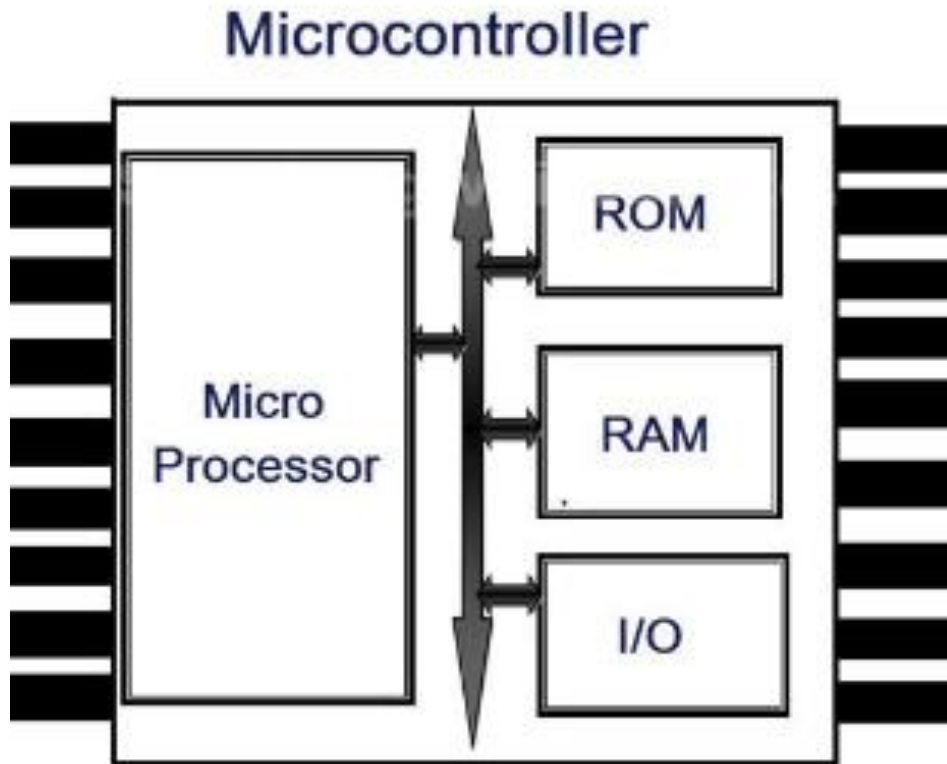
¿Qué es un microprocesador?



Un microcontrolador siempre contiene un microprocesador en su interior. Un microprocesador es la unidad central de procesamiento de un sistema informático.

Para construir o diseñar un sistema (computadora), un microprocesador debe estar conectado externamente a algunos otros componentes como la memoria (RAM y ROM) y los puertos de entrada / salida.

¿Qué es un microcontrolador?

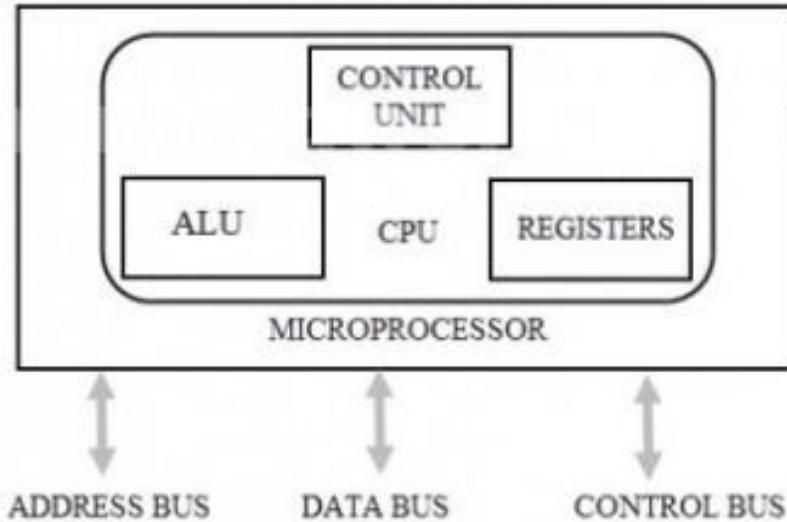


Un microcontrolador (a veces llamado MCU o unidad de microcontrolador) es un solo integrado que se utiliza normalmente para una aplicación específica y está diseñado para implementar determinadas tareas.

Básicamente, un microcontrolador recopila entradas, procesa esta información y genera una determinada acción en función de la información recopilada.

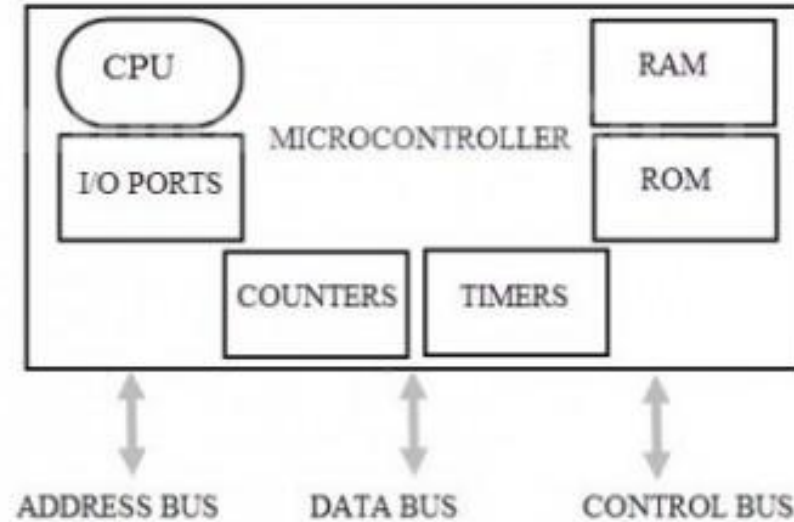
COMPARACIÓN

Microprocesador



El microprocesador asimila la función de una unidad central de procesamiento (CPU) en un solo circuito integrado (IC).

Microcontroladores

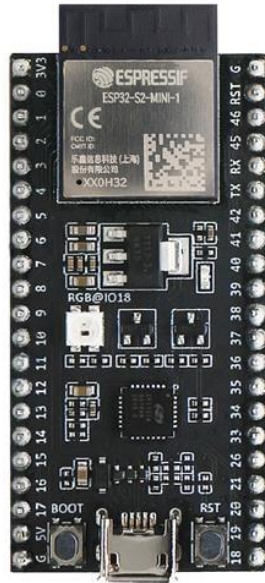


Un microcontrolador se puede considerar como una pequeña computadora que tiene un procesador y algunos otros componentes para convertirla en una computadora.

Definiciones importantes en microcontroladores:

1. Los Microcontroladores generalmente operan a velocidades más bajas, alrededor del rango de 1MHz a 200 MHz, y deben diseñarse para consumir menos energía porque están integrados dentro de otros dispositivos que pueden tener un mayor consumo de energía en otras áreas.
2. Clock o reloj: una señal periódica que se utiliza para sincronizar las operaciones internas del microcontrolador. El reloj se usa para regular la velocidad a la que se ejecutan las instrucciones y las operaciones en el microcontrolador, así como para sincronizar la comunicación con otros dispositivos externos.
3. Registros: son pequeñas unidades de almacenamiento de alta velocidad ubicadas dentro del microprocesador. Se utilizan para almacenar datos temporales, instrucciones y direcciones durante la ejecución de programas.

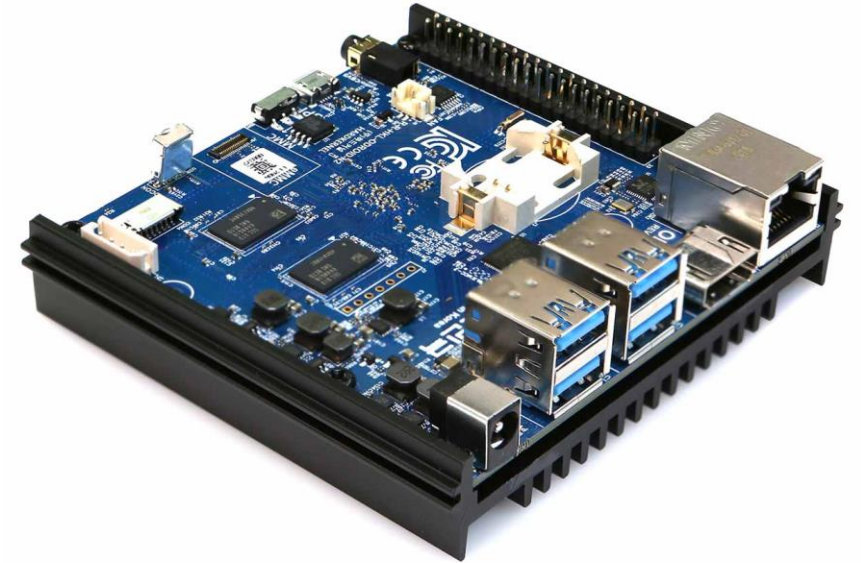
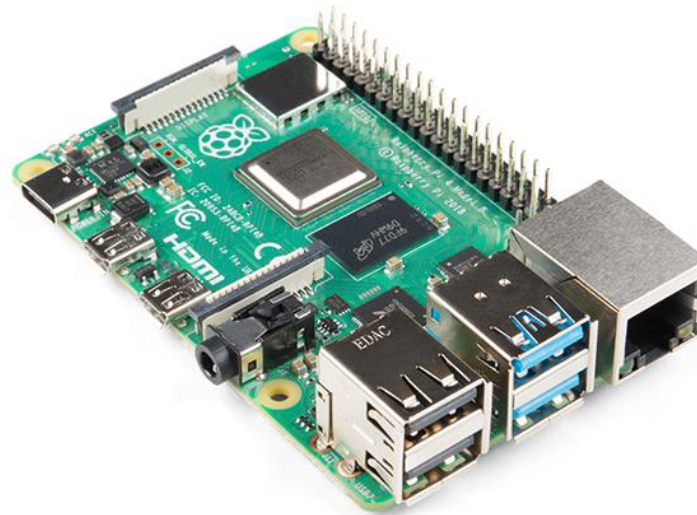
Sistemas embebidos



Un sistema embebido es un sistema electrónico completo que incluye hardware y software diseñados para realizar tareas específicas. Puede contener uno o varios componentes, como microcontroladores, microprocesadores, memoria, periféricos y otros dispositivos, y se utiliza en una amplia gama de aplicaciones, desde electrodomésticos hasta automóviles y dispositivos médicos.

ALGUNAS MARCAS

- Arduino
- ESP32 –ESP8266
- Raspberry Pi
- Odroid
- BeagleBone
- Nvidia
- Intel Edison



DESCRIPCIÓN ESP 32

El ESP32 es un microcontrolador con capacidades avanzadas de conectividad, lo cual lo convierte en un sistema embebido.

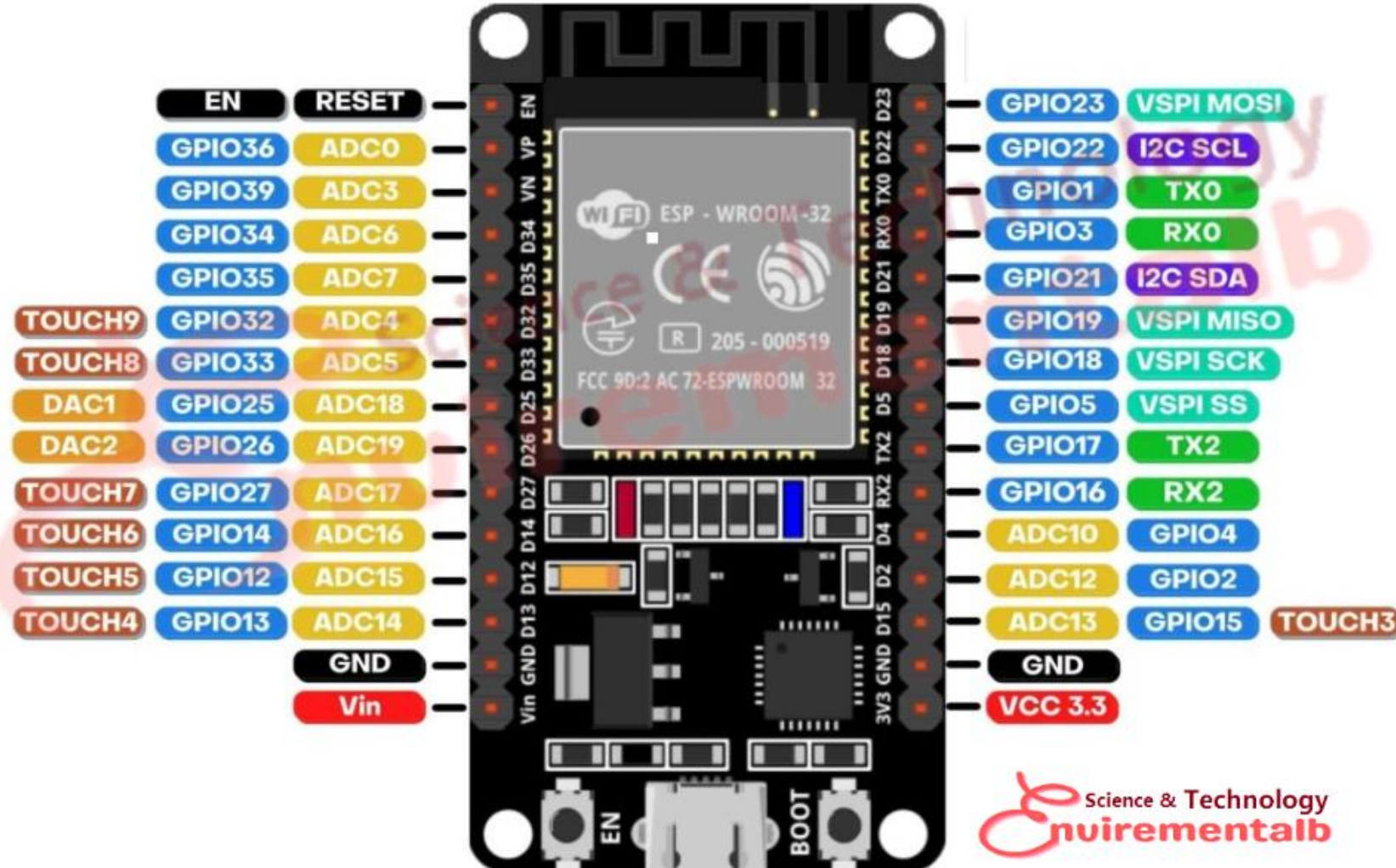
Es un chip de bajo costo y baja potencia diseñado específicamente para IoT (Internet de las cosas) y aplicaciones de comunicación inalámbrica.

Tiene integrada una unidad de procesamiento de aplicaciones Xtensa de doble núcleo y un coprocesador de Wi-Fi y Bluetooth de bajo consumo.

Por lo tanto, puede funcionar como un microcontrolador independiente o se puede usar como parte de un sistema embebido más grande.

DESCRIPCIÓN ESP 32

ESP32 DEV-KIT V1 - PINOUT DIAGRAM

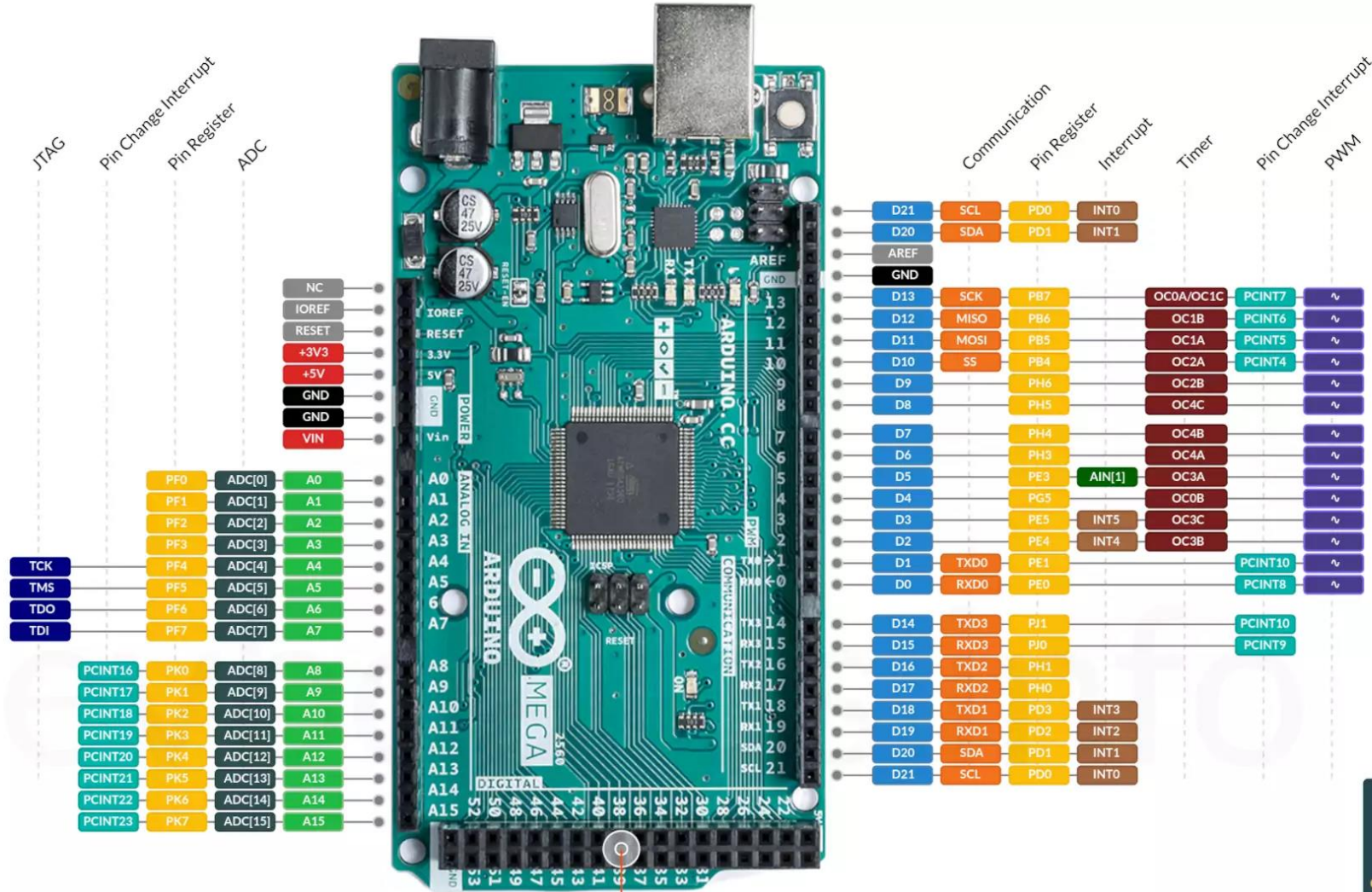


DESCRIPCIÓN ARDUINO

Arduino es una **plataforma de hardware libre** basada en microcontroladores que permite crear prototipos electrónicos de manera sencilla. Incluye una placa (como Arduino UNO o MEGA) y un entorno de desarrollo (IDE) que facilita la programación con un lenguaje similar a C/C++.

- La programación en Arduino es más simple y directa. IDE muy amigable, ideal para principiantes.
- Hay abundante documentación, ejemplos y librerías listas para usar.
- No necesitas configurar tantas cosas como en ESP32.

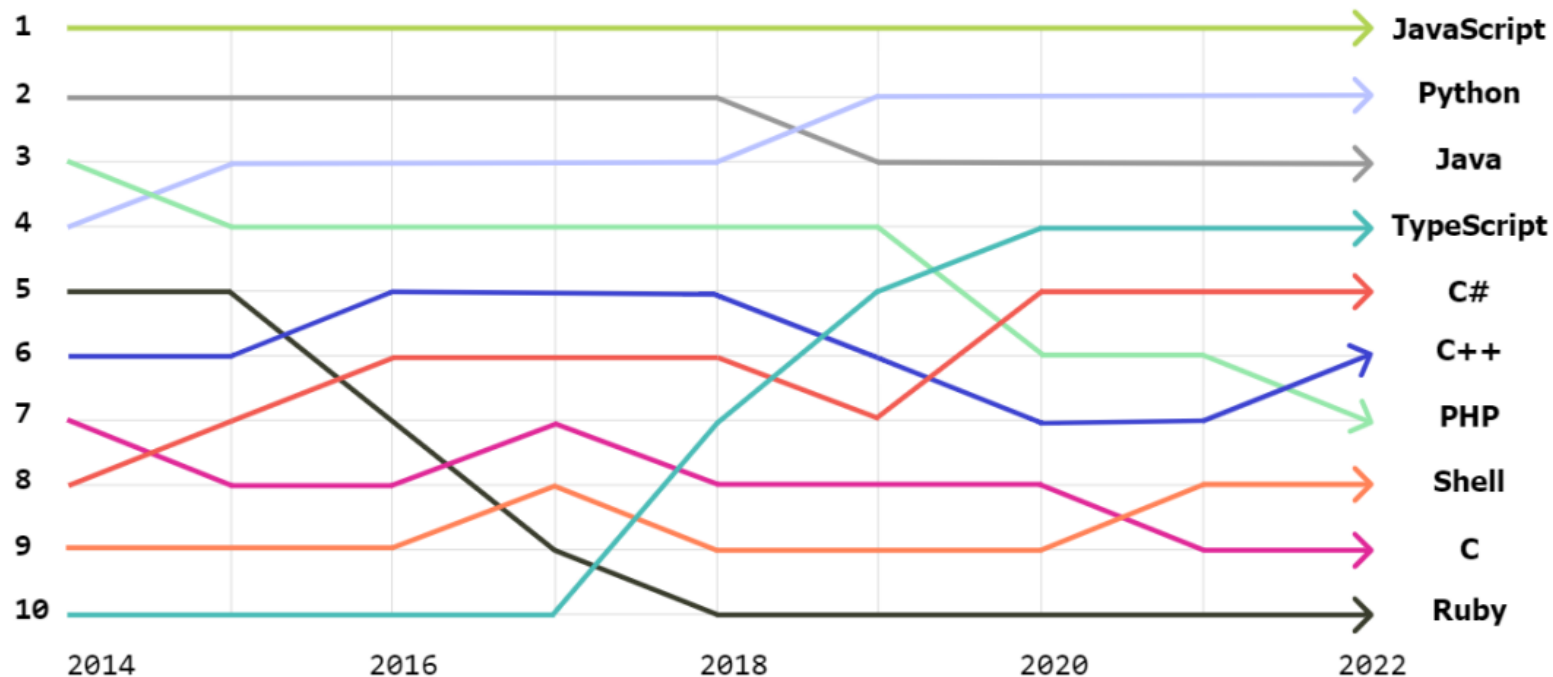
ARDUINO MEGA PINOUT



Alcance de los sistemas embebidos

1. Electrónica de consumo: Los sistemas embebidos se utilizan en dispositivos electrónicos de uso diario como teléfonos móviles, cámaras digitales, relojes inteligentes, televisores, electrodomésticos, etc.
2. Automoción: Los sistemas embebidos se encuentran en vehículos para controlar y gestionar diferentes aspectos, como la gestión del motor, la seguridad (como los sistemas de frenado ABS), la navegación y el entretenimiento.
3. Industria: Los sistemas embebidos se utilizan en procesos industriales para el control y monitoreo de maquinaria, como en sistemas de automatización y control de fábricas y plantas.
4. Medicina y salud: Los sistemas embebidos se aplican en dispositivos médicos como monitores de signos vitales, marcapasos, bombas de infusión y dispositivos de diagnóstico.

Lenguajes de programación



Son instrucciones específicas, que permiten crear acciones consecutivas.

<https://octoverse.github.com/2022/top-programming-languages>

Elementos de los lenguajes de programación

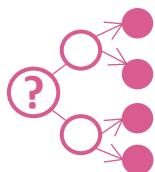
- Estructura del código (Comandos, variables...)
- Palabras reservadas (cambian de acuerdo a los lenguajes)
- Sintaxis (Como se hace)
- Semántica (Que hace)

¡Importante!

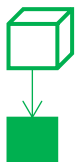
Existen errores de programación, estos pueden darse por sintaxis, semántica.

Tener presente que es lo que esperamos de nuestro código es importante, no todos los errores se nos muestran como tal

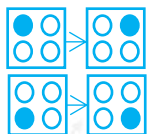
Solución a problemas de programación



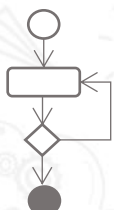
Descomposición de un problema en partes pequeñas para que sea más fácil solucionarlo.



Generalización y Abstracción, centrándose en la información importante y dejando de lado detalles irrelevantes.



Reconocimiento de patrones repetitivos para encontrar características comunes.



Lógica y Algoritmia para desarrollar instrucciones paso a paso para resolver los problemas.



Evaluación de la solución para garantizar que se resuelve el problema.

Algoritmia

Un algoritmo es una lista finita de instrucciones **claras, ordenadas y precisas** que describen un procedimiento para resolver un problema o realizar un cálculo.

En el contexto de la computación **se espera que las instrucciones puedan ser implementadas en un computador**, usando un lenguaje de programación.

Un algoritmo se puede representar a través de un diagrama de flujo o un pseudocódigo.



Problema 1

Diseñe un algoritmo que pida la edad de cada uno de los miembros de su equipo y le retorne la edad promedio de su equipo.

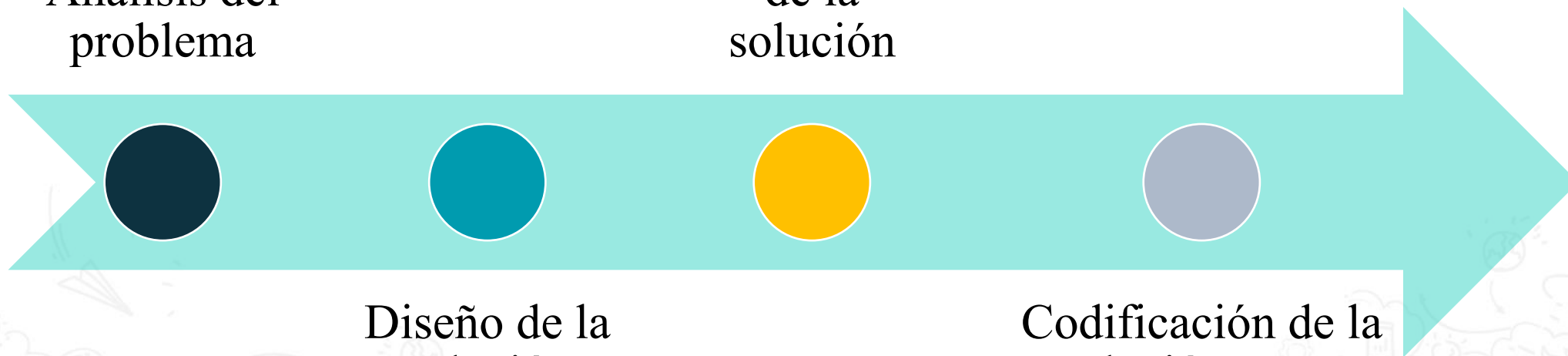
Metodología de trabajo con algoritmos

Análisis del
problema

Validación
de la
solución

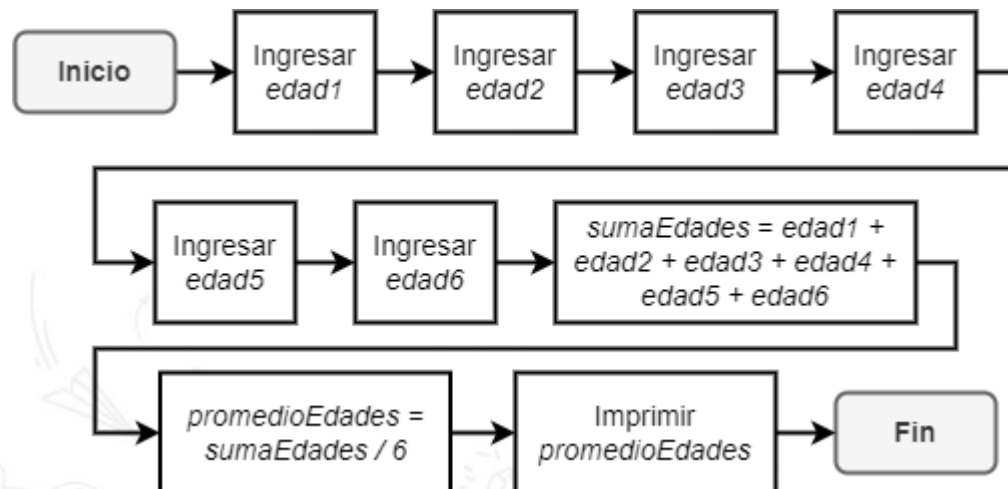
Diseño de la
solución

Codificación de la
solución en un
lenguaje de
programación



ALGORITMOS

Ejemplo de diseño con diagrama de flujo



Ejemplo de diseño con pseudocódigo

Inicio

Ingresar *edad1*

Ingresar *edad2*

Ingresar *edad3*

Ingresar *edad4*

Ingresar *edad5*

Ingresar *edad6*

$sumaEdades = edad1 + edad2 + edad3 + edad4 + edad5 + edad6$

$promedioEdades = sumaEdades / 6$

Imprimir *promedioEdades*



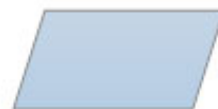


Fin

Un pseudocódigo es una
representación textual de un
algoritmo.

DIAGRAMAS DE FLUJO

- Representaremos los algoritmos por medio de diagramas de flujo o pseudocódigos

Un diagrama de flujo describe un proceso, sistema o algoritmo informático

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

EJERCICIOS

Somos Innovación Tecnológica con *Sentido Humano*

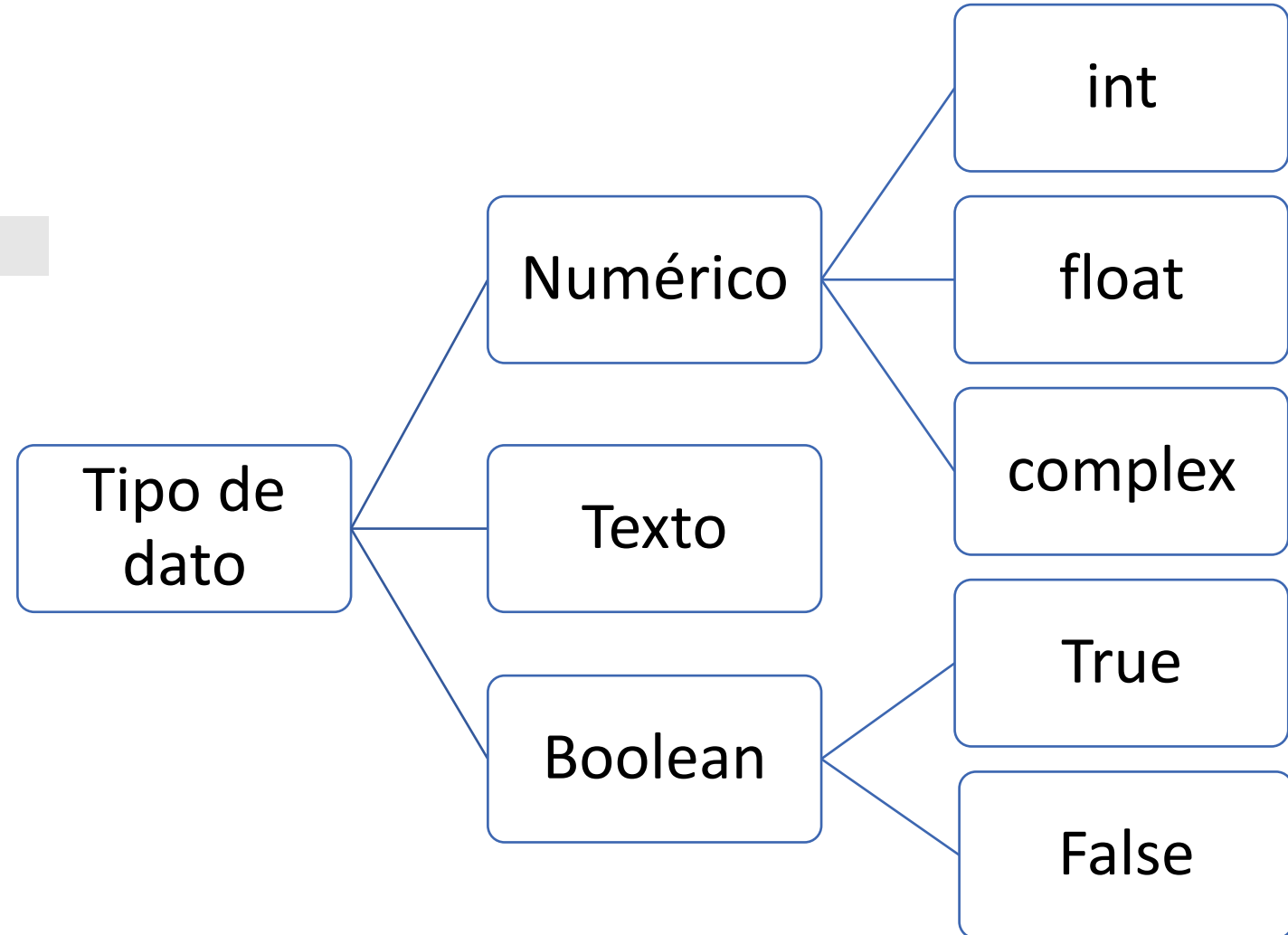


Alcaldía de Medellín

Introducción a Python

TIPOS DE DATOS:

Object type	Example literals/creation
Numbers	1234, 3.1415, 3+4j, Decimal, Fraction
Strings	'spam', "guido's", b'a\x01c'
Lists	[1, [2, 'three'], 4]
Dictionaries	{'food': 'spam', 'taste': 'yum'}
Tuples	(1, 'spam', 4, 'U')
Files	myfile = open('eggs', 'r')
Sets	set('abc'), {'a', 'b', 'c'}
Other core types	Booleans, types, None



Introducción a Python

LECTURA E IMPRESIÓN DE VARIABLES:

Lectura

```
a = input("Ingrese el valor de a: ")
```

Impresión

```
print("El valor de a es: ", a)
```



Institución
Universitaria
Reacreditada en Alta Calidad

¡Gracias!

Somos Innovación Tecnológica con *Sentido Humano*



Alcaldía de Medellín