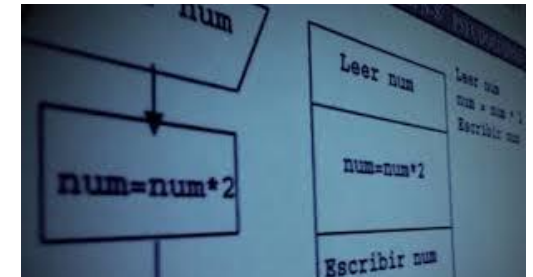


# Bloque1: Programaci n b sica con JavaScript (JS)

## Introducci n a la programaci n



# ¿Conceptos básicos (I)?

- Un **programa (software)** es un conjunto de instrucciones que permiten que el ordenador ejecute una serie de acciones.
- La **programación** es el proceso de escritura de un programa.
- Los **Lenguajes de programación** permiten la comunicación entre el usuario (que realiza los programas) y el ordenador (que los ejecuta). Permiten crear programas y software.
- Existen **lenguajes de bajo nivel** y **lenguajes de alto nivel**.
- Un **algoritmo** es un método previo a la programación que describe la solución de un problema computacional, mediante una serie de pasos precisos, definidos y finitos.

# ¿Conceptos básicos (II)?

- Los **algoritmos** suelen tener **tres partes básicas**:



**Entrada:** Datos necesarios para poder ejecutarse

**Proceso:** Acciones o cálculos a realizar

**Salida:** Resultado esperado

- Las **fases principales de creación de un algoritmo** son:

- 1. Análisis:** Se determina cuál es el problema a resolver
- 2. Diseño:** Elaboramos el algoritmo
- 3. Pruebas:** Comprobamos el resultado

# ¿Conceptos básicos (III)?

- **Ejemplo de Algoritmo** para calcular la media aritmética de dos número (con una calculadora no científica):



- 1) Introducir el primer número
- 2) Pulsar la tecla "+"
- 3) Introducir el segundo número
- 4) Pulsar la tecla "/"
- 5) Teclear el número "2"
- 6) Pulsar la tecla "="

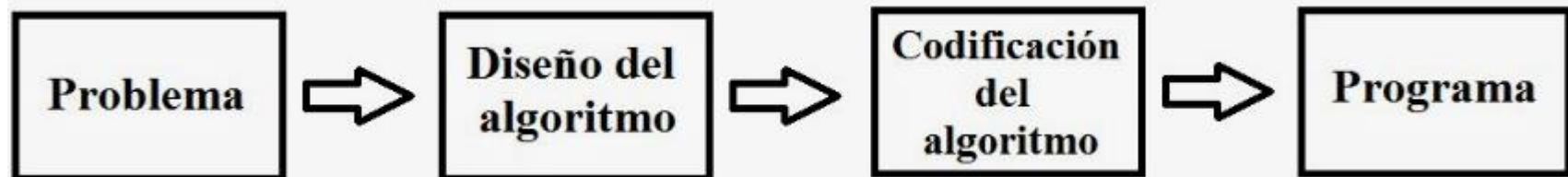
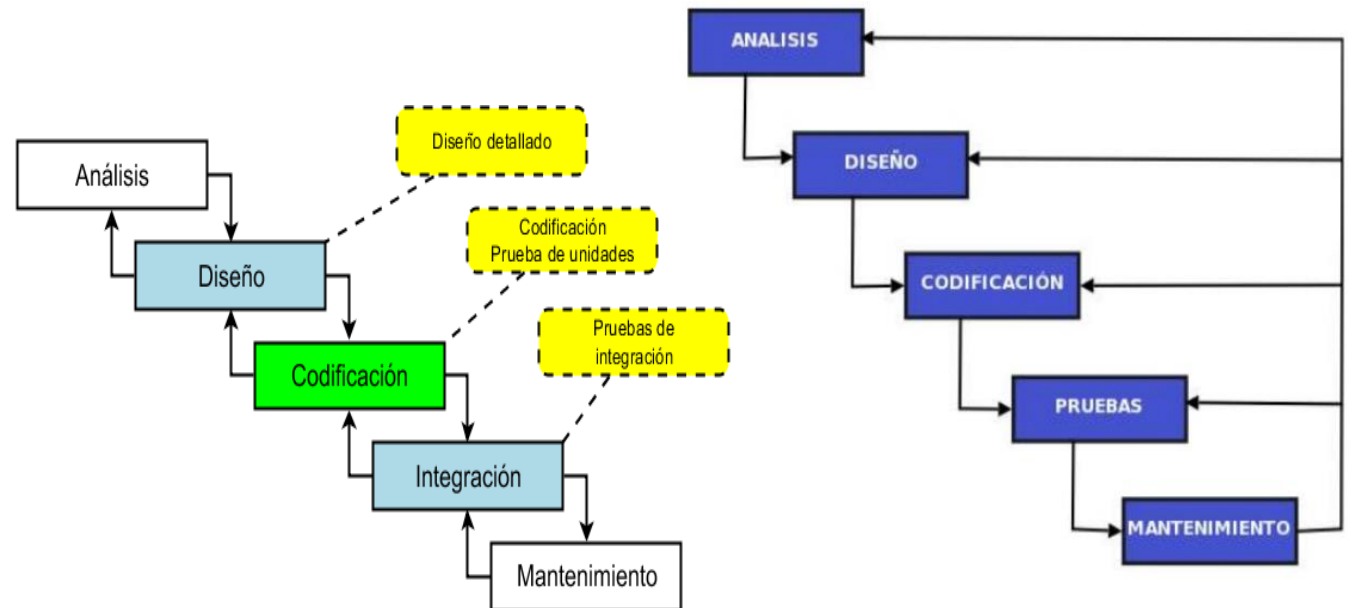
- El **Algoritmo** es un paso previo a la **programación !!!**
- Los algoritmos se expresan normalmente en **pseudocódigo**.

***Programar implica “pensar de forma algorítmica”***

# ¿Conceptos básicos (IV)?

## – Fases de la programación tradicional.

- Definición del problema.
- Análisis del problema.
- Diseño de la solución.
- Codificación.
- Compilación y ejecución.
- Verificación y pruebas.
- Depuración.
- Documentación.



# ¿Conceptos básicos (V)?

## Fases – Etapas en el desarrollo de Software

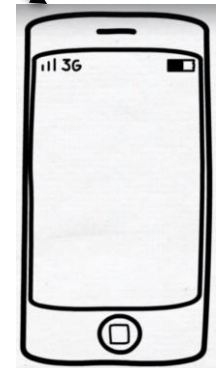
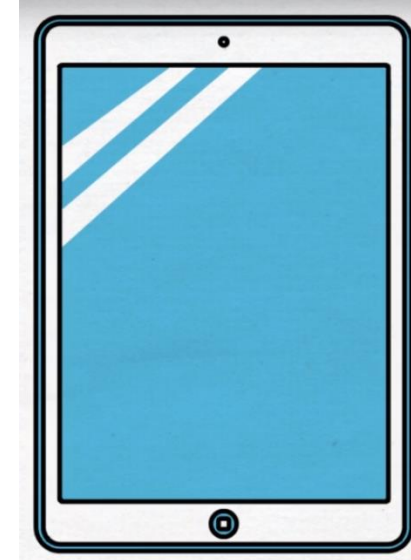


# Dispositivos Programables

–  Programar?...  Para qu  dispositivos?



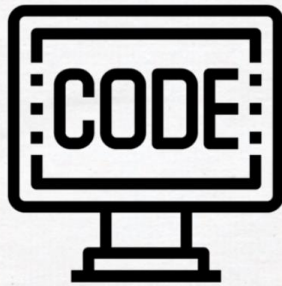
 Solo estos?





## SOFTWARE

ES EL QUE SE ENCARGA DE LAS TAREAS  
LÓGICAS DE LA COMPUTADORA



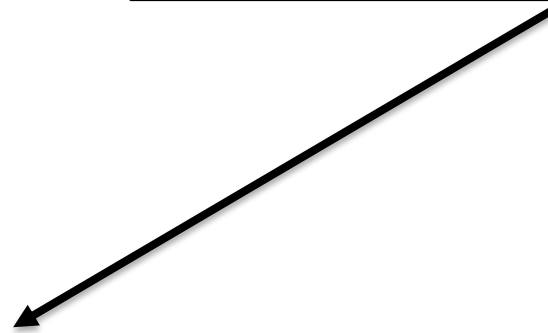
¿Qué hacer?

¿Cómo resolver problemas?

## LENGUAJE DE PROGRAMACIÓN

ES LA FORMA DE DECIRLE A LA COMPUTADORA

¿QUÉ HACER? Y ¿CÓMO HACERLO?



UN IDIOMA INTERMEDIO  
QUE NOS AYUDA A TRANSMITIRLE ÓRDENES A LA COMPUTADORA

JUNTO CON UN EQUIPO FÍSICO  
QUE REALICE ESE TRABAJO





## – Tareas de programación.

- Representación gráfica de Mario Bros y movimientos.
- Conteo de las monedas.
- Conteo de Puntos.
- Tiempo del juego.
- Etc...

## TERMINOLOGÍA EXTRAÑA

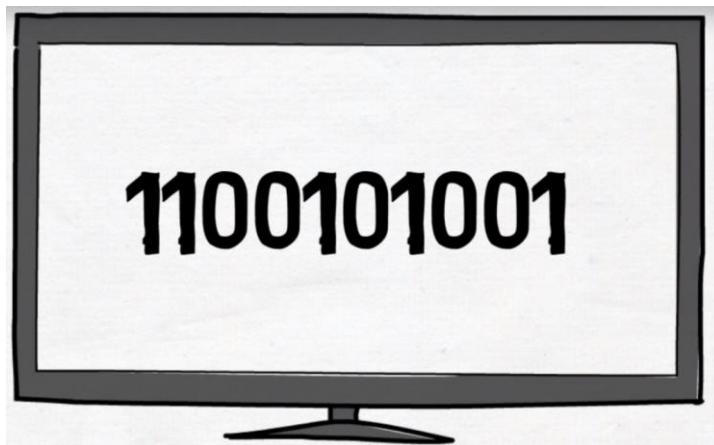
Acumuladores  
Contadores  
Clases  
Variables  
Constantes

## LENGUAJE DE BAJO NIVEL

Es un lenguaje más directo para las computadoras y no se parece a como hablan los seres humanos

**BINARIO**

**LENGUAJE DE MÁQUINA**  
**ENSAMBLADOR**



1° VOLTAGE ALTO

0° VOLTAGE BAJO O NADA DE VOLTAGE

```
1 ; HOLA.ASM
2 ; Programa clasico de ejemplo. Despliega una leyenda en pantalla.
3 STACK     SEGMENT STACK           ; Segmento de pila
4           DW      64 DUP (?)      ; Define espacio en la pila
5 STACK     ENDS
6
7 DATA     SEGMENT                 ; Segmento de datos
8 SALUDO    DB      "Hola mundo!!",13,10,"$" ; Cadena
9 DATA     ENDS
10
11 CODE     SEGMENT                 ; Segmento de Codigo
12           ASSUME CS:CODE, DS:DATA, SS:STACK
13
14 INICIO:                                     ; Punto de entrada al programa
15           MOV  AX,DATA              ; Pone direccion en AX
16           MOV  DS,AX               ; Pone la direccion en los registros
17           MOV  DX,OFFSET SALUDO    ; Obtiene direccion del mensaje
18           MOV  AH,09H              ; Funcion: Visualizar cadena
19           INT  21H                 ; Servicio: Funciones alto nivel DOS
20           MOV  AH,4CH              ; Funcion: Terminar
21           INT  21H
22 CODE     ENDS
23           END  INICIO              ; Marca fin y define INICIO
24
```

## LENGUAJE DE ALTO NIVEL

Es un lenguaje parecido al español o al inglés



## TRABAJAREMOS CON LENGUAJES DE ALTO NIVEL

```
1 var helado = 'chocolate';
2 if (helado === 'chocolate') {
3     alert('¡Si, amo el helado de chocolate!');
4 } else {
5     alert('Awww, pero mi favorito es el de chocolate...');
6 }
```



NO ES LO MISMO



Fragmentos de código

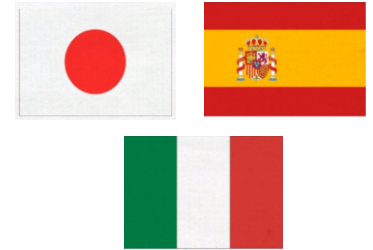
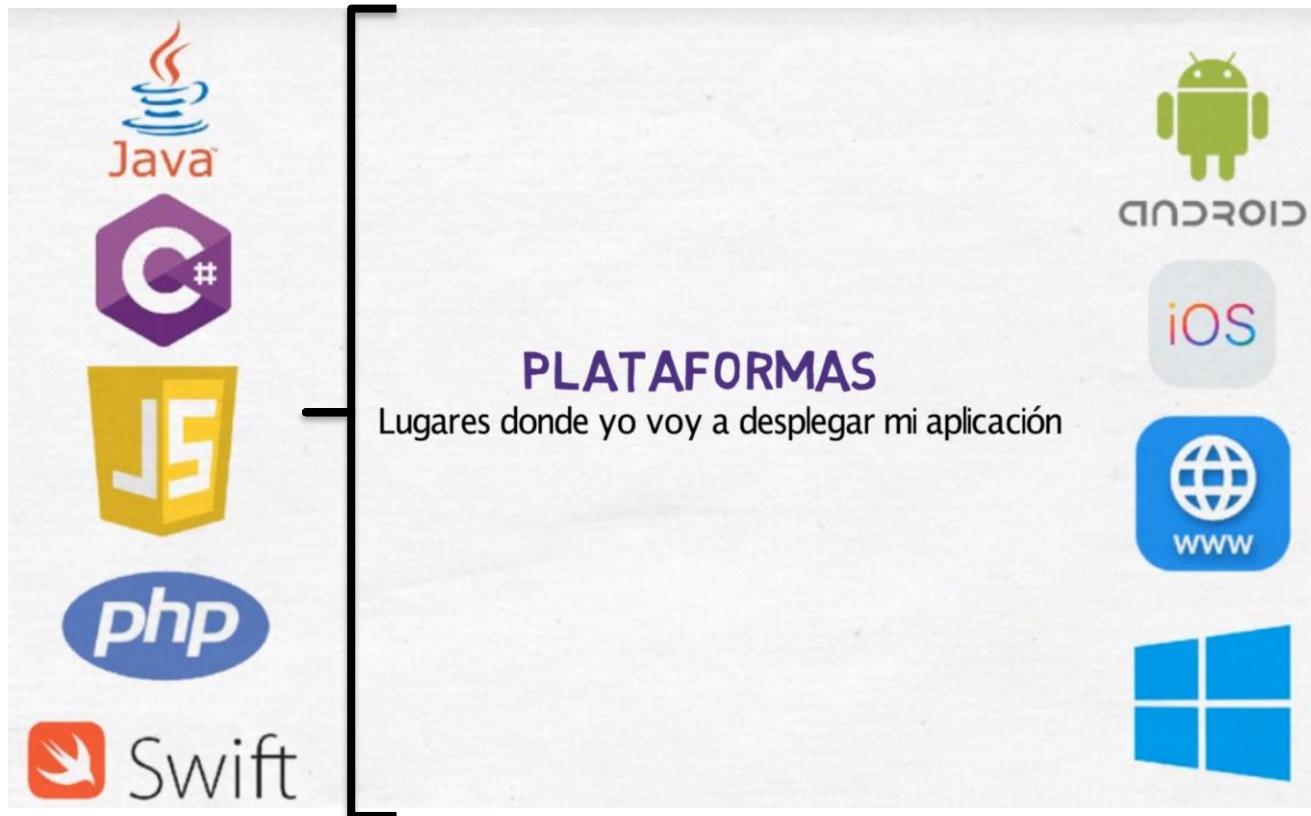
```
int n = 0;
while (n < 5)
{
    Console.WriteLine(n);
    n++;
}
```



# Lenguajes de alto nivel

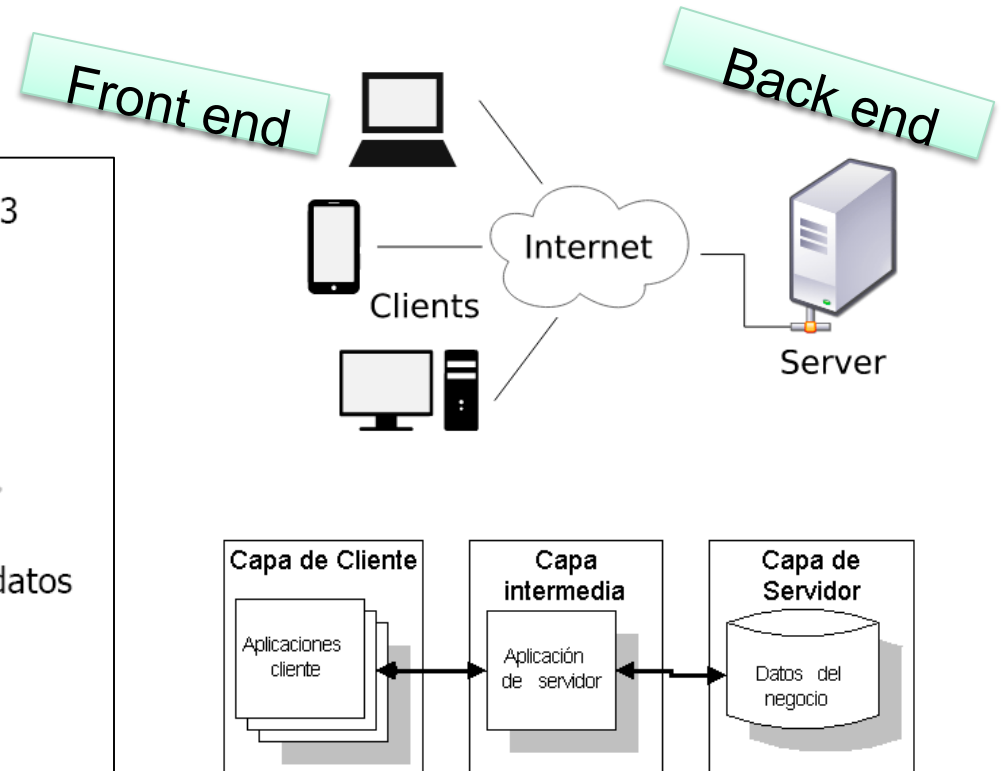
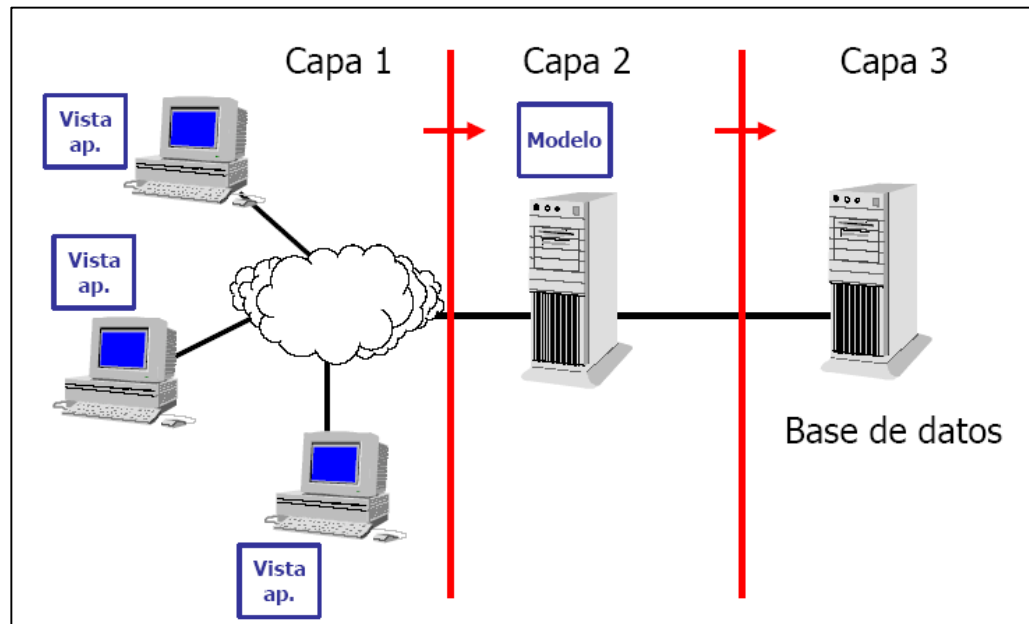
¿CÚANTOS LENGUAJES DE PROGRAMACIÓN EXISTEN?

¿POR QUÉ HAY TANTOS?



# Modelo Cliente - Servidor

La arquitectura **cliente-servidor modelo** de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**. Un **cliente** realiza peticiones a otro programa, el **servidor**, quien le da respuesta.





## ¡ Bienbenidos al nuevo curso acadEmiko!

Existe una serie de reglas y normas que dicen como se debe de escribir en español

Eso es un framework

**Framework.** En el desarrollo de **Software**, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.



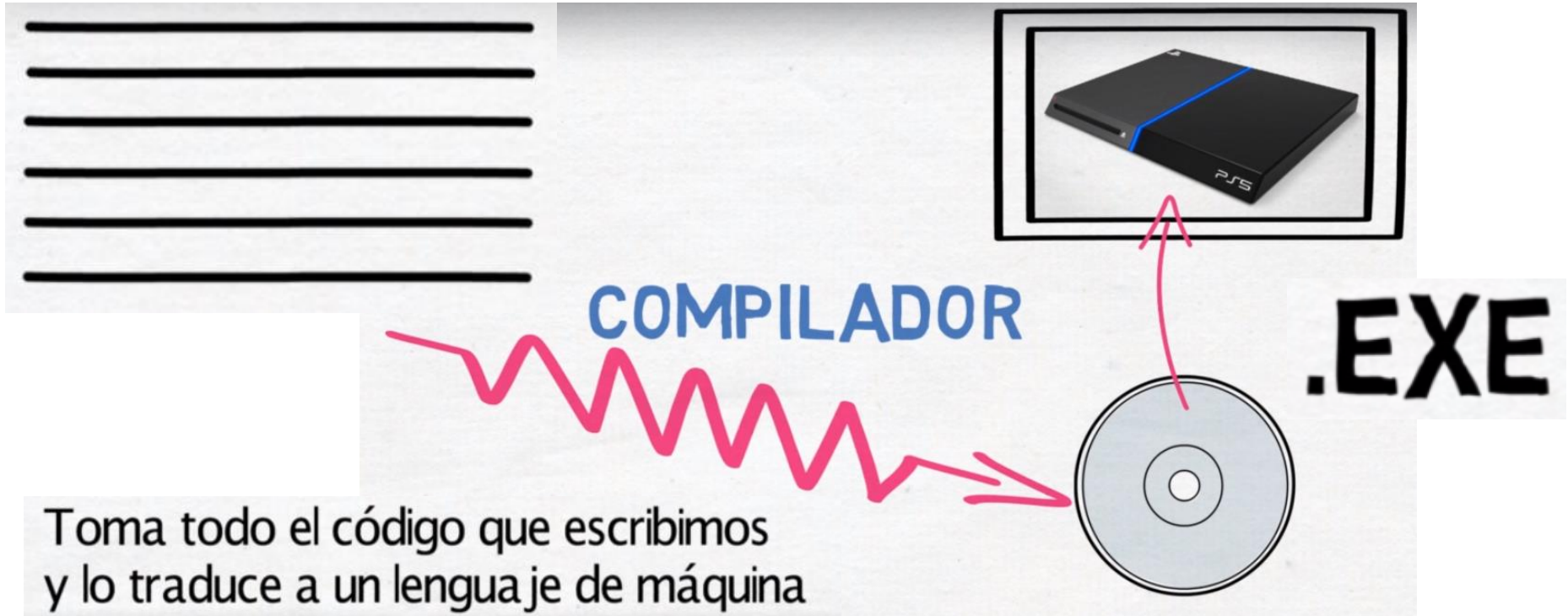
Los mejores JavaScript Frameworks

[Link al artículo](#)





# Compilador Vs Intérprete



## Lenguajes compilados

C, C++, C#, Java, Visual Basic

```
package main
import "fmt"

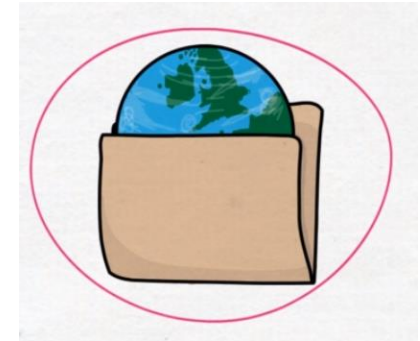
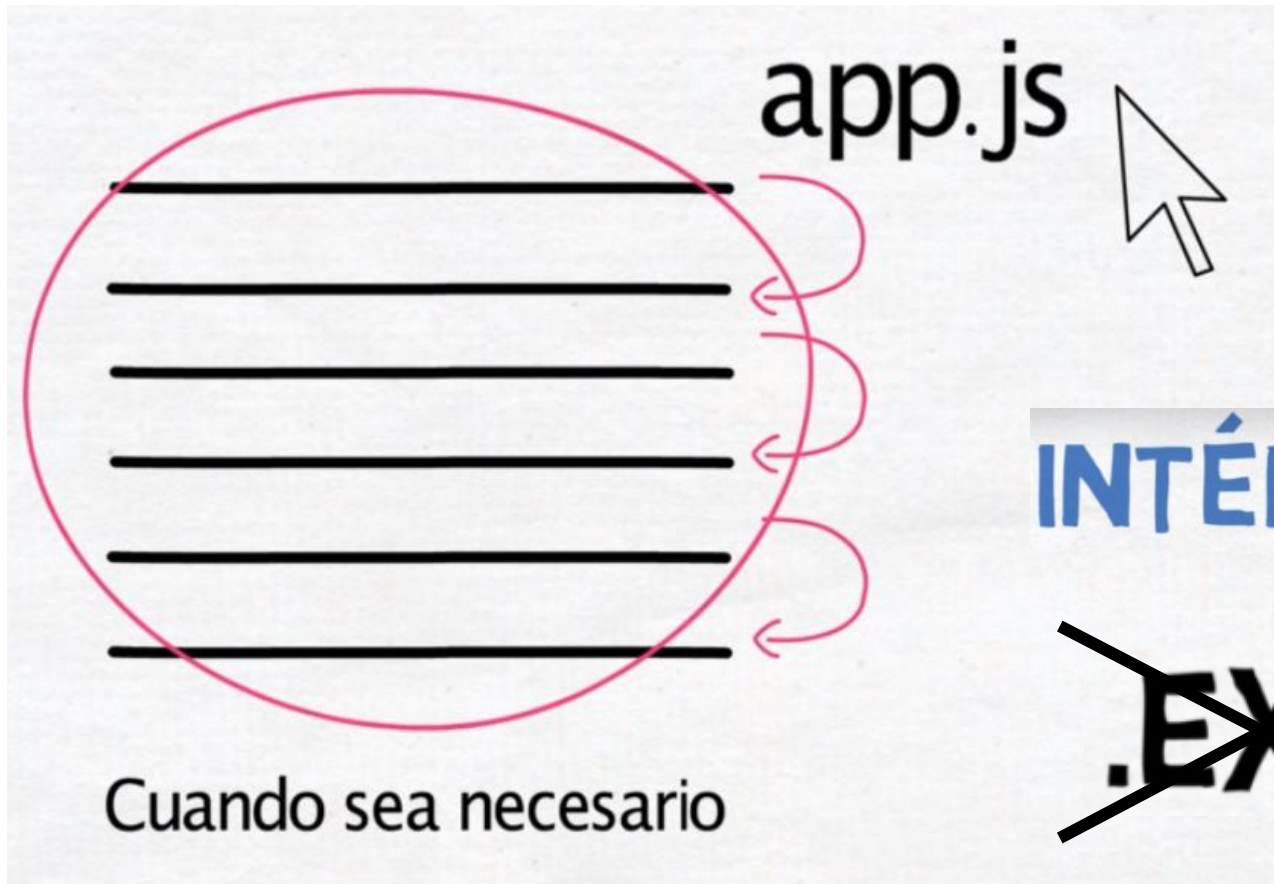
func main() {
    fmt.Printf("hello, world")
}
```

Lenguaje de alto nivel que entiende el programador

```
0101010111101110001101
0100010100010101001010
0101010010101010000101
0011010001010100011110
011001010010101010001
1110001101010010010001
```

Lenguaje de máquina que entiende el procesador

# Compilador Vs Intérprete



## Lenguajes interpretados

JavaScript, PHP, Python, Ruby...

- Lógica de programación para resolver problemas.

$$10 \rightarrow 20$$
$$10 + 30 = 40$$
$$40 \div 2 = 20$$

**ALGORITMO**  
ES UN PROCESO PARA RESOLVER UN PROBLEMA.  
POR LO GENERAL ES ORDENADO Y SECUENCIAL

$$10 + 10 = 20$$

- Hay diferentes formas de resolver un mismo problema.
- Hay algoritmos más eficientes que otros (resolviendo el mismo problema).

## – Programando en JavaScript el algoritmo

10 → 20

10 + 10 = 20

### VARIABLE

Es el espacio en memoria donde podemos almacenar datos, y estos datos pueden cambiar en un futuro

Es una palabra que no podemos utilizar fuera de ese contexto

```
let a = 10;  
a = a + 10; // El valor de a, es 20  
console.log( a );
```