

DESARROLLO WEB FULL STACK

Introducción a la programación

DWFS - COR



Programa Informático



Programa informático

Un programa informático o programa de computadora es una **secuencia de instrucciones**, escritas para realizar una **tarea específica** en una computadora.

El programa tiene un formato **ejecutable** que la computadora puede utilizar directamente para ejecutar las instrucciones.

Lo escribimos en un **lenguaje formal**.

Una colección de programas de computadora y datos relacionados se conoce como software.

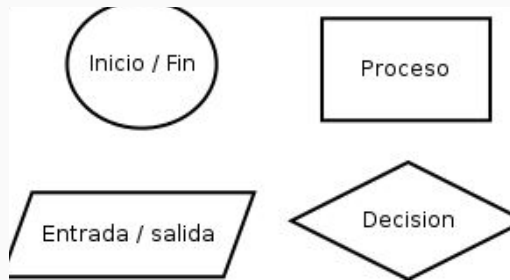


Programa informático

Los programas informáticos no son más que procesos de toma de decisiones. Podemos visualizar un programa mediante un diagrama de flujo, que nos muestran de forma gráfica como es la lógica que aplica un programa informático, a esto en el ambiente se lo llama “lógica de negocio” o “reglas de negocio”.



Diagramas de flujo



Elementos de un diagrama

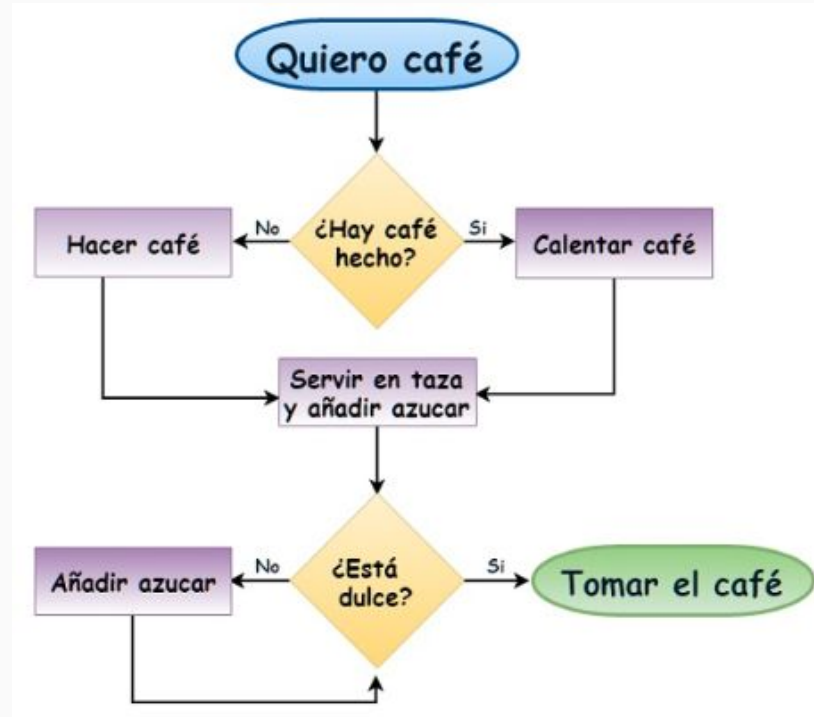
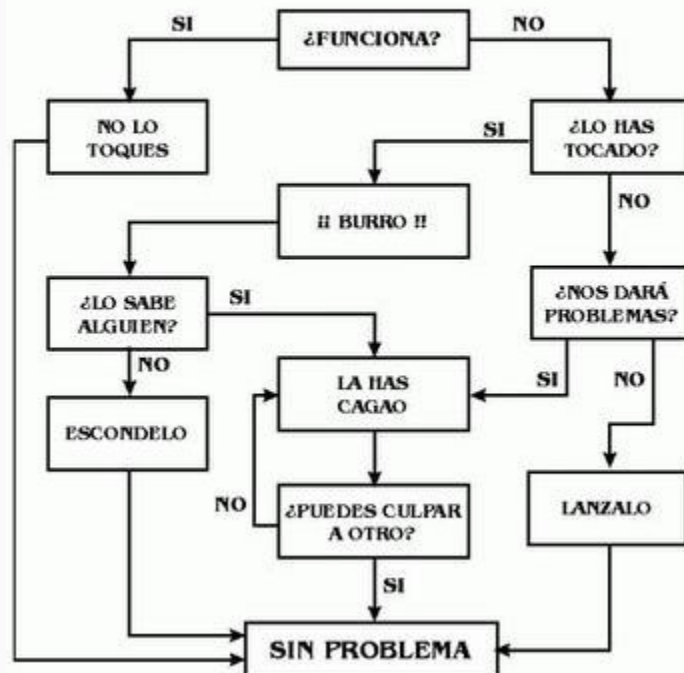


DIAGRAMA DE FLUJO PARA SOLUCIONAR PROBLEMAS



Lenguaje formal

En matemáticas, lógica y ciencias de la computación, un lenguaje formal es un lenguaje cuyos símbolos primitivos y reglas para unir esos símbolos están **formalmente especificados**. Al conjunto de los símbolos primitivos se le llama el **alfabeto (o vocabulario)** del lenguaje, y al conjunto de las reglas se lo llama la **gramática formal (o sintaxis)**.



Niveles de lenguajes

- Bajo Nivel
- Alto Nivel



Lenguajes de bajo nivel

Un lenguaje de programación de características de bajo nivel, es aquel en el que sus instrucciones **ejercen un control directo sobre el hardware** y están condicionados por la estructura física de las computadoras que lo soportan.

Ejemplo: Assembly

```
DATA SEGMENT
| | MESSAGE DB "HOLA MUNDO"
ENDS

CODE SEGMENT
| | ASSUME DS:DATA CS:CODE
START:
| | MOV AX, DATA
| | MOV DS, AX
| | LEA DX, MESSAGE
| | MOV AH, 9
| | INT 21H
| | MOV AH, 4CH
| | INT 21H
ENDS
END START
```



Lenguajes de alto nivel

Se caracterizan por expresar los algoritmos de una manera **adecuada a la capacidad cognitiva humana**, en lugar de la capacidad con que los ejecutan las máquinas.

Estos lenguajes permiten una máxima flexibilidad al programador a la hora de abstraerse o de ser literal.

Permiten una expresión casi oral entre la escritura del programa y su posterior compilación.

Ejemplo: C++

```
#include <iostream>

int main() {
    std::cout << "Hola mundo!";
    return 0;
}
```

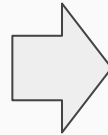


Compilación

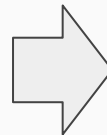
```
DATA SEGMENT
    MESSAGE DB "HOLA MUNDO"
ENDS

CODE SEGMENT
    ASSUME DS:DATA CS:CODE
START:
    MOV AX, DATA
    MOV DS, AX
    LEA DX, MESSAGE
    MOV AH, 9
    INT 21H
    MOV AH, 4CH
    INT 21H
ENDS
END START
```

Lenguaje de bajo
nivel



E
N
S
A
M
B
L
A
D
O
R



```
01010101110101110101010
01101010110110001101111
01011000100100000110120
10010110011000010011020
01010100010000000100011
```

Lenguaje máquina

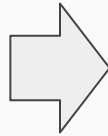


Compilación

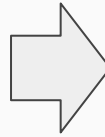
```
#include <iostream>

int main() {
    std::cout << "Hola mundo!";
    return 0;
}
```

Lenguaje de alto
nivel



C
O
M
P
I
L
A
D
O
R



```
01010101110101110101010
01101010110110001101111
01011000100100000110120
10010110011000010011020
01010100010000000100011
```

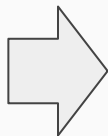
Lenguaje máquina



Interpretación (JavaScript)

```
const saludo = "Hola mundo";  
console.log(saludo);
```

Lenguaje de alto
nivel interpretado



I
N
T
E
R
P
R
E
T
E



```
01010101110101110101010  
01101010110110001101111  
01011000100100000110120  
10010110011000010011020  
01010100010000000100011
```

Lenguaje máquina

V8



Introducción a JavaScript



JavaScript

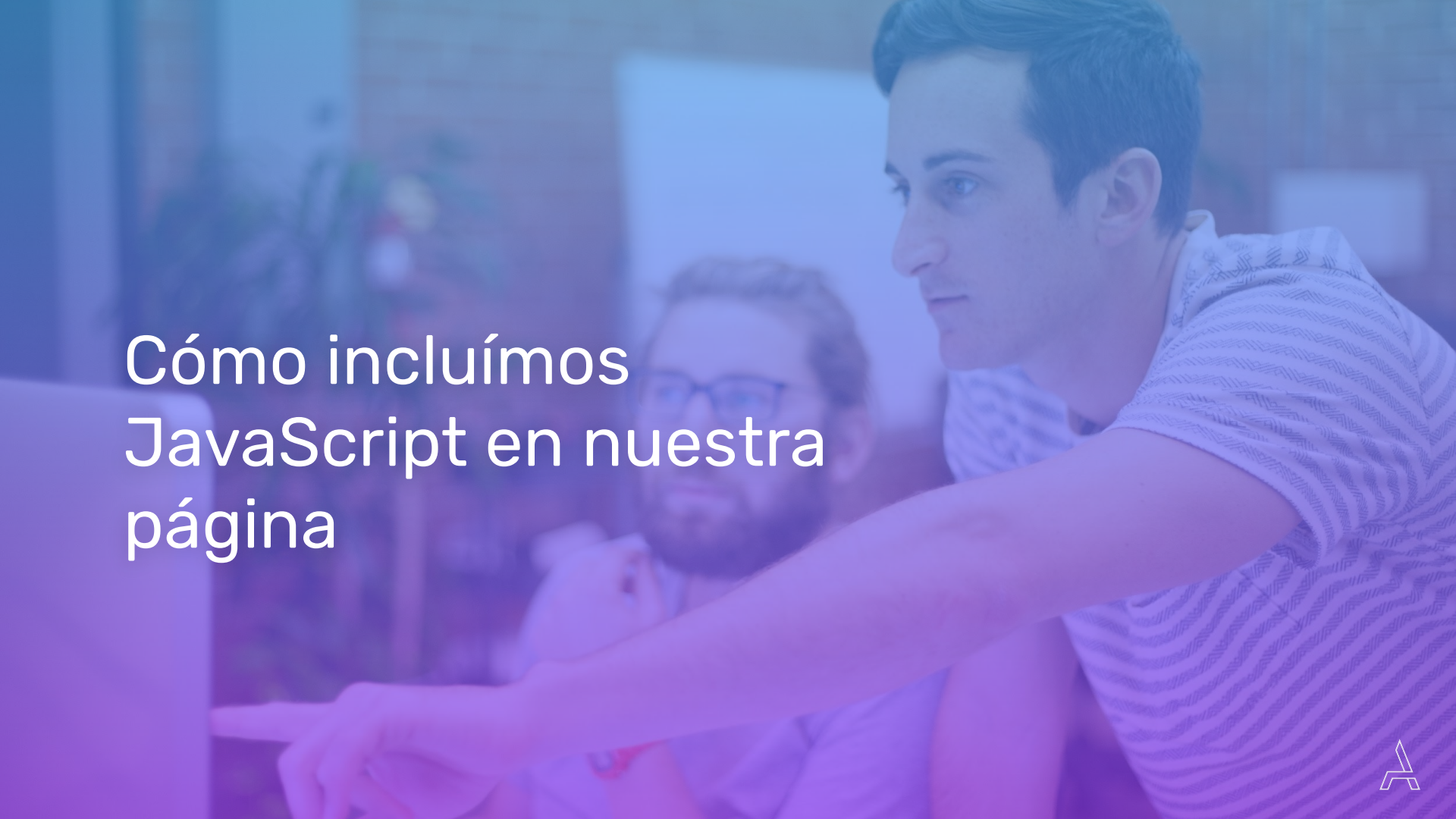
Es un lenguaje **interpretado, tipado debil** que corre en el navegador y servidores y nos permite generar interacciones e interfaces complejas en nuestros proyectos web.



Sintaxis JavaScript

- **No se tienen en cuenta los espacios en blanco y las nuevas líneas:** el intérprete de JavaScript ignora cualquier espacio en blanco sobrante.
- **Se distinguen las mayúsculas y minúsculas**
- **No se define el tipo de las variables:** al crear una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución.
- **No es necesario terminar cada sentencia con el carácter de punto y coma (;):** en la mayoría de lenguajes de programación, es obligatorio terminar cada sentencia con el carácter ;. Aunque JavaScript no obliga a hacerlo, es conveniente hacerlo.
- **Se pueden incluir comentarios:** los comentarios se utilizan para añadir información en el código fuente del programa, los mismos no se ejecutan.



A photograph of two men in a professional setting, likely a tech office. The man in the foreground, wearing a striped shirt, is pointing at a computer screen. The man in the background, wearing glasses and a beard, is looking at the screen. The image has a blue and purple color overlay.

Cómo incluimos JavaScript en nuestra página



Embebido en nuestra página

Usando la etiqueta
<script>

```
<script>  
    alert("Hola mundo!");  
</script>
```



Desde un archivo externo

Usando la etiqueta
<script> en el head del
documento

```
<script src="programa.js"></script>
```



Mi primer código JS

Mostrar un mensaje en el navegador

```
// A continuación se muestra un mensaje  
alert("Hola mundo!");
```

```
/* A continuación se muestra un mensaje  
multilinea, son muy útiles para mejorar  
la legibilidad de comentarios largos */  
alert("Hola mundo!");
```



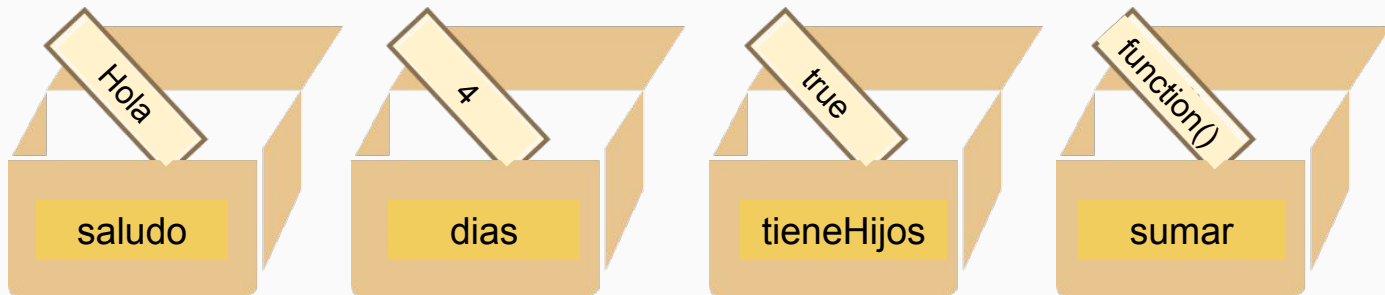
Variables



Definición

Las variables son contenedores para el almacenamiento de valores de datos.





Variables

Donde guardamos datos internos de nuestro programa.

“Viven” durante el tiempo que nuestro programa corre.

```
var saludo = 'Hola mundo';  
alert(saludo);
```



Variables

Podemos sobreescribirlas.

```
var saludo = 'Hola mundo!';  
saludo = 'Hola pepe!';
```



Variables

Podemos guardar números, texto y cosas más complejas.

```
var saludo = '¡Hola mundo!';  
var mes = 4;  
var tieneHijos = true;  
var otro = function(){};
```



Variables

Su nombre no puede tener ñ, tildes ni empezar con números.

```
var 1saludo = "Hola mundo!";  
var día = 2019;  
var año = 2019;
```



Variables convención de nombrado (JS)

- Siempre se comienza con letra minúscula.
- Si el nombre está compuesto por dos o más palabras se usa el formato camelCase.

```
var saludoLargo = "Hola a todas las personas  
del mundo!";  
var saludoMasLargo = "Hola a todas las personas  
del mundo que me conocen!";
```



Operaciones aritméticas

Todas las operaciones aritméticas se pueden realizar en javascript

```
var suma = 1 + 2;  
var resta = 2 - 1;  
var multiplicacion = 2 * 3;  
var division = 4 / 2;  
var modulo = 4 % 2;
```



Uso de variables

Las variables transportan sus valores.

```
var x = 3;  
var y = 2;  
var resultado1 = x + y;  
var resultado2 = y - 3;
```



A photograph of two men in an office setting, looking at a computer screen. The man in the foreground is pointing at the screen. The image has a blue overlay.

Nuevos tipos de variables (ECMAScript 6)



Nuevos tipos de variables

let, const


```
// Alcance de bloque  
let numero = 2;
```

```
// Alcance de bloque pero además constante  
const aceleracionGravitatoria = 9.8;
```



Ejercicio: Hacer un programa que almacene en una variable una operación aritmética y muestre el resultado por un alert en el navegador.



A photograph of two men in a professional setting, looking at a computer screen. The man in the foreground is pointing at the screen. The image has a blue overlay. The text 'Explorando la consola de desarrolladores del navegador' is written in white on the left side.

Explorando la consola de desarrolladores del navegador

