

# PROGRAMACIÓN WEB EN EL ENTORNO CLIENTE

MF0491\_3



*Realizado por Manuel Macías*

### Qué es la lógica de programación

La lógica de programación consiste en la organización y planificación coherente de las instrucciones necesarias para ejecutar con éxito un programa.

Dentro de ello hay una serie de patrones que se repiten en todos los lenguajes, como el uso de variables, métodos o funciones, condicionales y bucles.

### Qué es un ordinograma

Se trata de un estilo específico de diagramas de flujo, que muestra la secuencia lógica y detallada de las operaciones que necesitamos para la realización de un programa, paso a paso y con sus distintos caminos, tomas de decisiones, repeticiones, etc.

### Símbolos utilizados



**Descripción:** Para inicio/fin o para una parada indeterminada



**Descripción:** Símbolo de entrada/salida genérico

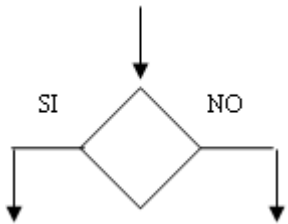


**Descripción:** Representa una operación o proceso general con datos de memoria.

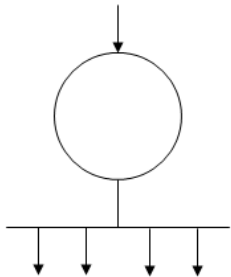
### Símbolos utilizados



**Descripción:** Símbolo de subprograma o subrutina. Se utiliza para realizar una llamada a un modulo del programa.



**Descripción:** Símbolo de decisión para realizar una pregunta con dos posibles respuestas. Es lo que llamamos símbolo de selección simple.



**Descripción:** Símbolo de selección múltiple

### Símbolos utilizados



**Descripción:** Símbolo de bucle definido.



**Descripción:** Conector. Se utiliza para agrupar varias líneas de flujo que salen del mismo origen.

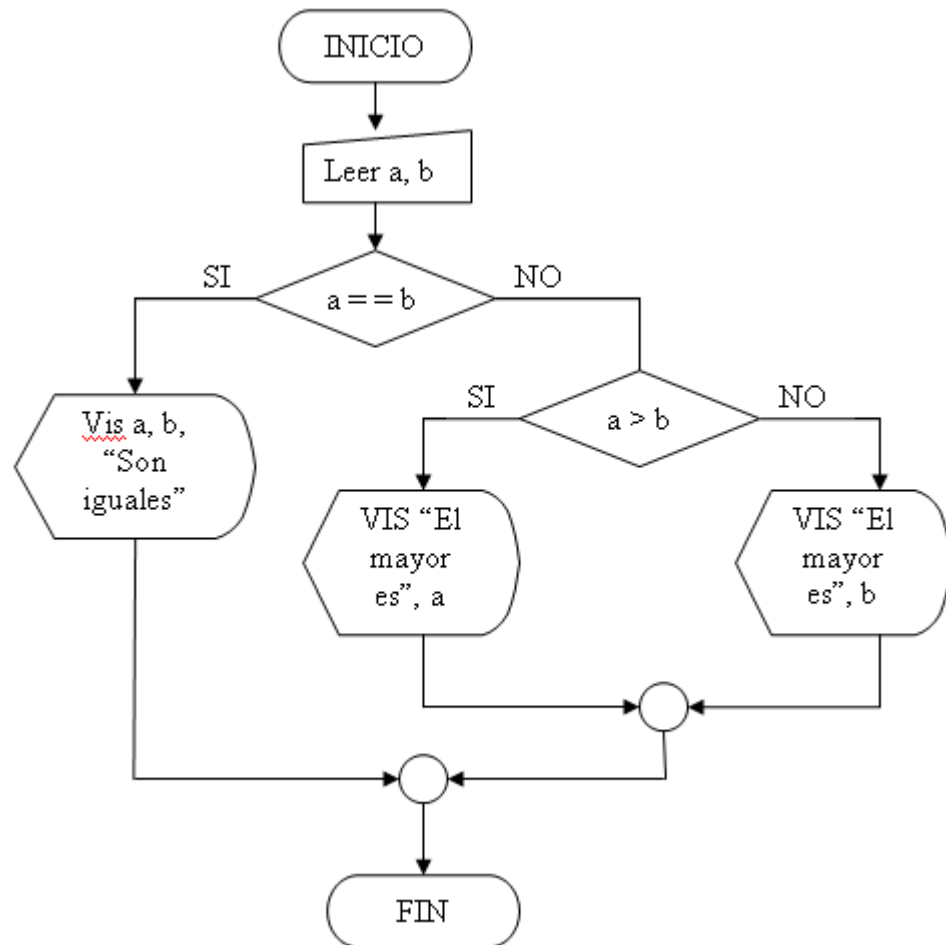
Esto sería un  
comentario

**Descripción:** Símbolo para poner comentarios

### Reglas a la hora de hacer ordinogramas

- Todos los símbolos utilizados deben estar unidos por líneas de flujo.
- No se pueden cruzar las líneas de flujo
- A un símbolo de proceso pueden llegarle varias líneas de flujo pero solo puede salir una de él.
- Al símbolo de inicio no puede llegarle ninguna línea de flujo
- De un símbolo de fin no puede salir ninguna línea de flujo pero si le pueden llegar varias.

### Ejemplo de ordinograma





### ¿Qué Es el Pseudocódigo?

El pseudocódigo es una forma de representar código, como algoritmos, funciones y otros procesos, utilizando una combinación de lenguaje natural y elementos similares al lenguaje de programación.

Se llama «pseudocódigo» porque no es realmente ejecutable. En cambio, es una forma de que los humanos comprendan y planifiquen la lógica de la programación — describir los pasos de un programa de forma que sea fácil de entender para los humanos, sin dejar de ser lo suficientemente detallado como para convertirse rápidamente en un lenguaje de programación específico.

### Ejemplo de Pseudocódigo.

Dadas dos variables numéricas A y B, que el usuario debe teclear, se pide realizar un algoritmo que intercambie los valores de ambas variables y muestre cuanto valen al final las dos variables:

InicioProceso

Escribir "Introduce el valor de A"

Leer A

Escribir "Introduce el valor de B"

Leer B

C<-A

A<-B

B<-C

Escribir "A vale " A " y B vale " B

FinProceso

### Programa:

En programación se suele llamar «programa» al conjunto total de código que se desarrolla.

En Javascript, quizás el término más utilizado es aplicación web (cuando es un desarrollo con mucha cantidad de Javascript). También se suelen generalizar utilizando términos como «script» o «código Javascript».

### Algoritmo:

Un algoritmo es un conjunto de pasos conocidos, en un determinado orden, para conseguir realizar una tarea satisfactoriamente y lograr un objetivo.

### Comentarios:

Los comentarios en nuestro código son fragmentos de texto o anotaciones que el navegador ignora y no repercuten en el programa. Sirven para dejar por escrito detalles importantes para el programador. De esta forma cuando volvamos al código, nos será más rápido comprenderlo.

Es una buena costumbre comentar en la medida de lo posible nuestro código.

### Indentación:

Se llama indentar a la acción de colocar espacios o tabuladores antes del código, para indicar si nos encontramos dentro de un if, de un bucle, etc...

### Variables:

Es el nombre genérico que se le da a pequeños espacios de memoria donde guardas una información determinada, de forma muy similar a las incógnitas en matemáticas. Un programa puede tener muchas variables, y cada una de ellas tendrá un nombre, un valor y un tipo de dato.

### Constantes:

Es el mismo concepto de una variable, salvo que en este caso, la información que contiene es siempre la misma (no puede variar).



### Funciones:

Cuando comenzamos a programar, nuestro código se va haciendo cada vez más y más grande, por lo que hay que buscar formas de organizarlo y mantenerlo lo más simple posible. Las funciones son agrupaciones de código que, entre otras cosas, evitan que tengamos que escribir varias veces lo mismo en nuestro código. Una función contendrá una o mas acciones a realizar y cada vez que ejecutemos una función, se realizarán todas ellas.

### Parámetros:

Es el nombre que reciben las variables que se le pasan a las funciones. Muchas veces también se les denomina argumentos.

### Bucles:

Cuando estamos programando, muchas veces necesitaremos realizar tareas repetitivas. Una de las ventajas de la programación es que permite automatizar acciones y no es necesario hacerlas varias veces. Los bucles permiten indicar el número de veces que se repetirá una acción. De esta forma, sólo la escribimos una vez en nuestro código, y simplemente indicamos el número de veces que queremos que se repita.

### Iteración:

Cuando el programa está en un bucle repitiendo varias veces la misma tarea, cada una de esas repeticiones se denomina iteración.

### Librería:

Muchas veces, desarrollamos código que resuelve tareas o problemas que, posteriormente, querremos reutilizar en otros programas. Cuando eso ocurre, en Javascript se suele empaquetar el código en lo que se llaman librerías, que no es más que código listo para que otros programadores puedan utilizarlo fácilmente en sus programas y beneficiarse de las tareas que resuelven de forma muy sencilla.

## Elementos del lenguaje de guion:

- Variables (Numéricas, Caracteres, Cadena de Caracteres, Booleanos)
  - Variables Globales
  - Variables Locales

## Elementos del lenguaje de guion:

- Operaciones
  - Aritméticas
  - Lógicas
  - Binarias
  - Otras

### Elementos del lenguaje de guion:

- Comparaciones
  - Igual '==').
  - Distinto '!=').
  - Mayor '>').
  - Menor '<').
  - Mayor o igual '>=').
  - Menor o igual '<=').



### Elementos del lenguaje de guion:

- Asignaciones
  - `nombre_variable = valor;`

**No confundir Comparación (==) y asignación (=).**

### Objetos del lenguaje de guion:

- Métodos (función asociada a un objeto)
- Eventos (acción del usuario en nuestra web a la que tenemos que dar respuesta.
- Atributos (definen datos particulares sobre el estado de un objeto)
- Funciones (Es un conjunto de instrucciones que agrupamos bajo un determinado nombre)
  - Definición o nombre de la función
  - Parámetros que recibe la función
  - Cuerpo principal de la función

### Objetos del lenguaje de guion:

```
Nombre función sumar (a,b) {  
    Return a+b;  
}
```

```
function nombre(param,param, ...param) {  
    instrucciones  
}
```

### Ejecución de un script.

Los scripts pueden ejecutarse o cuando la página se está cargando o cuando se produce un determinado evento.

### Declaración de variables

Para poder trabajar con variables , previamente tenemos que declararlas y darles un identificador para poder referenciarlas en nuestro programa.

Las operaciones con variables las podemos usar para:

- Mostrar datos.
- Pedir datos.
- Enviar datos.
- Recibir datos.
- En el uso de expresiones.
- En funciones.
- En llamadas a las funciones.
- En objetos.

### Declarando una variable en JavaScript

```
var variable = 2;
```

### Usando el contenido de una variable

```
var variable = 2;  
console.log(variable);
```

### Asignando un nuevo valor a la variable

```
var variable = 2;  
variable = 3;  
console.log(variable);
```

### Asignando el valor de una variable a otra variable

```
var variable = 2;  
var variable2 = variable;  
console.log(variable);
```



### Realizando una operación con el valor almacenado en una variable

```
var variable = 2;  
variable = variable + 1; // 3  
console.log(variable);
```

### Realizando operaciones complejas

```
var variable = 2;  
variable = variable * 25 + 3; // 53  
console.log(variable);
```

### Controlando la precedencia en la operación

```
var variable = 2;  
variable = variable * (25 + 3); // 56  
console.log(variable);
```

### Incrementando el valor de una variable

```
var variable = 0;  
variable ++;  
console.log(variable); // 1
```

### Incrementando el valor de una variable en un valor específico

```
var variable = 0;  
variable += 5;  
console.log(variable); // 5
```

### Calculando el resto de una división

```
var variable = 11 % 3; // 2
```

```
console.log(variable);
```

### Determinando la paridad de un número

```
var variable = 11;
```

```
console.log(variable % 2); // 1
```

### Asignando una cadena de caracteres a una variable

```
var variable = "Hola Mundo!";  
console.log(variable);
```

### Concatenando texto

```
var variable = "Mi nombre es ";  
variable = variable + "Juan";  
console.log(variable);
```

### Agregando texto al comienzo del valor

```
var variable = "Juan";  
variable = "Mi nombre es " + variable;  
console.log(variable);
```

### Concatenando texto con números

```
var variable = "El número es " + 3;  
console.log(variable);
```

### Concatenando números

```
var variable = "20" + 3;  
console.log(variable); // "203"
```

### Escapando caracteres

```
var variable = 'El \'libro\' es interesante';  
console.log(variable); // "El 'libro' es interesante"
```

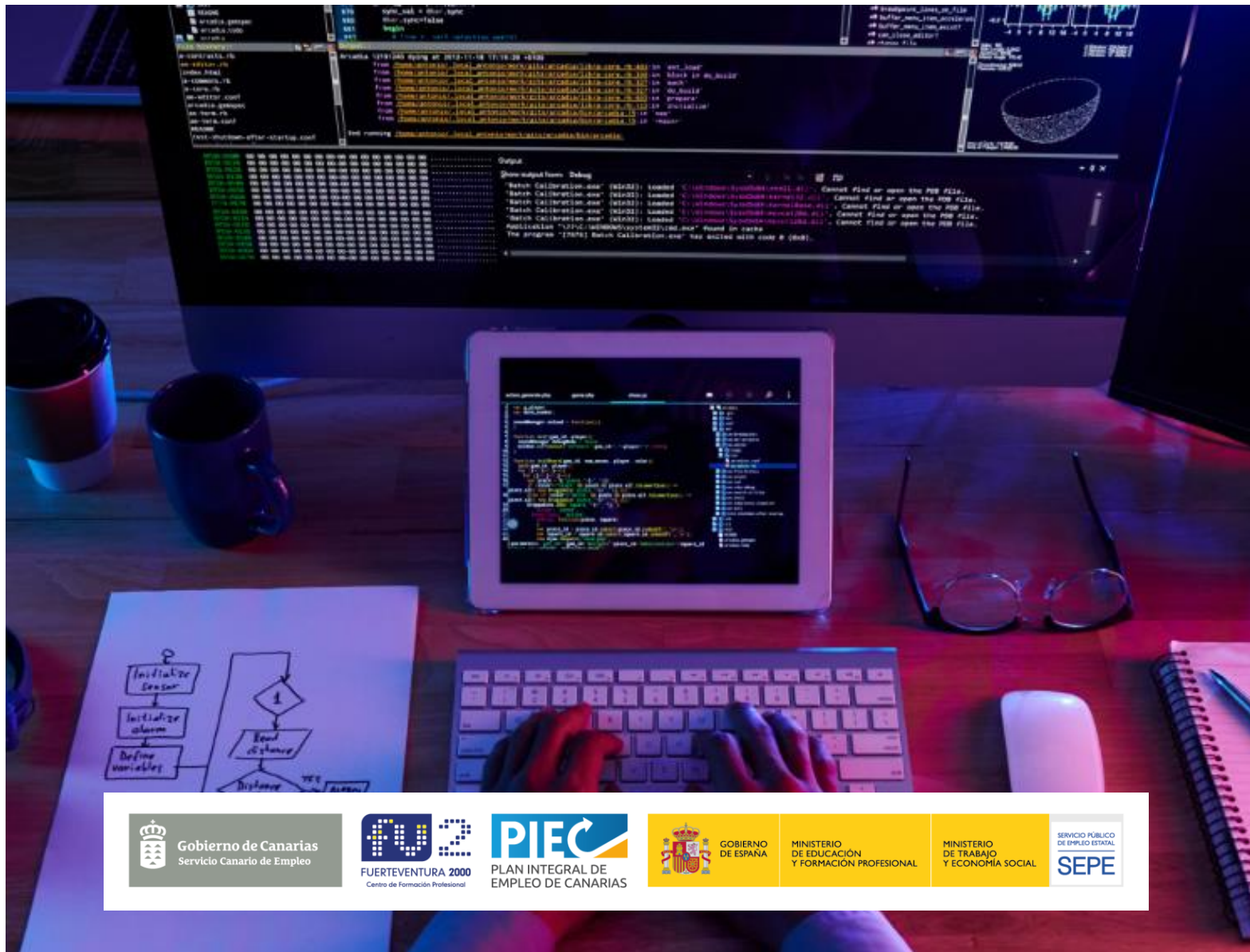
### Generando nuevas líneas de texto

```
var variable = "Felicidad no es hacer lo que uno quiere\r\n";  
variable = variable + "sino querer lo que uno hace."  
console.log(variable);
```



### Declarando una variable Booleana

```
var variable = true;  
console.log(variable);
```



Programación web en el entorno cliente

# GRACIAS

MANUEL MACÍAS

[TUTORIAS@MANUELMACIAS.ES](mailto:TUTORIAS@MANUELMACIAS.ES)

## PROGRAMACIÓN JAVASCRIPT

