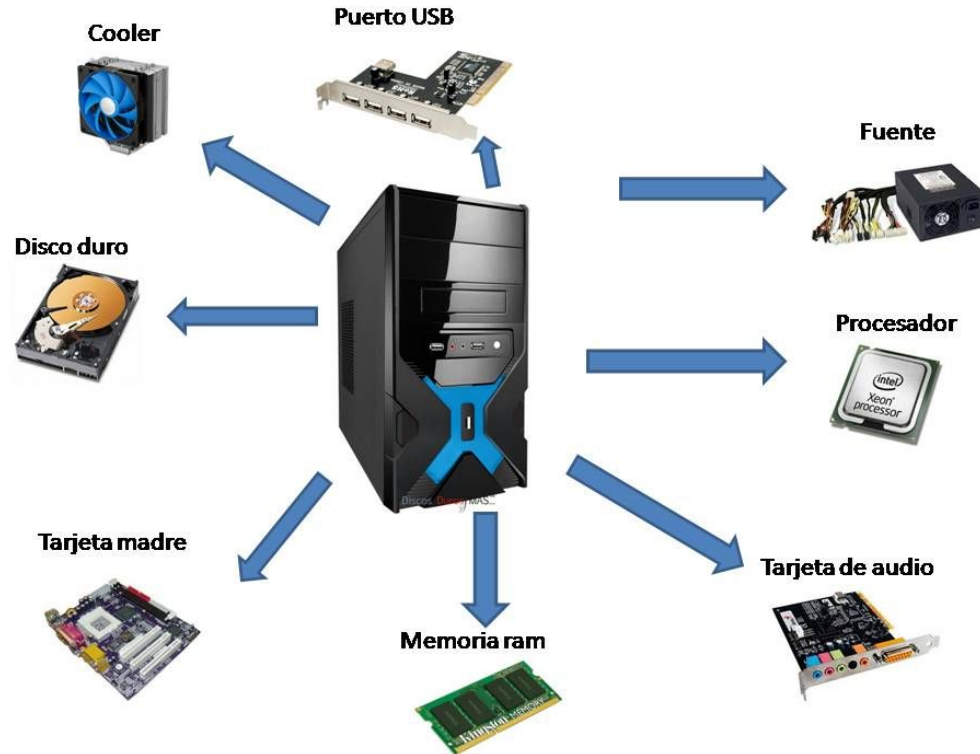


Programación de aplicaciones Web

Parte 1



Componentes de una CPU



Componentes principales de una computadora

HARDWARE AND SOFTWARE



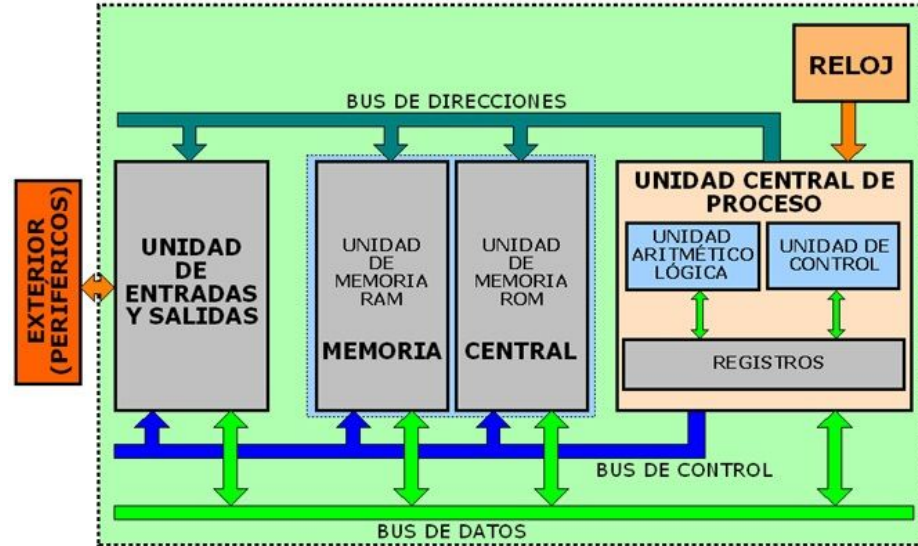
HARDWARE

SOFTWARE

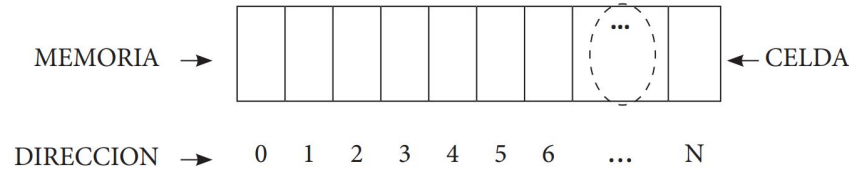
Funcionamiento de interno

Funciona usando cuatro simples pasos: **buscar, decodificar, ejecutar, almacenar**, llamado el “Ciclo de la máquina”.

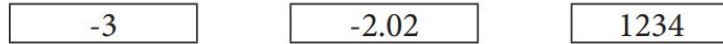
Las instrucciones son obtenidas por la CPU desde la memoria. La CPU luego decodifica y ejecuta estas instrucciones. El resultado es almacenado de nuevo en la memoria luego que se complete el ciclo de ejecución de las instrucciones.



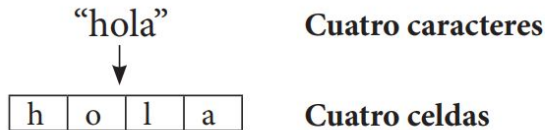
Tipos de datos simples y compuestos



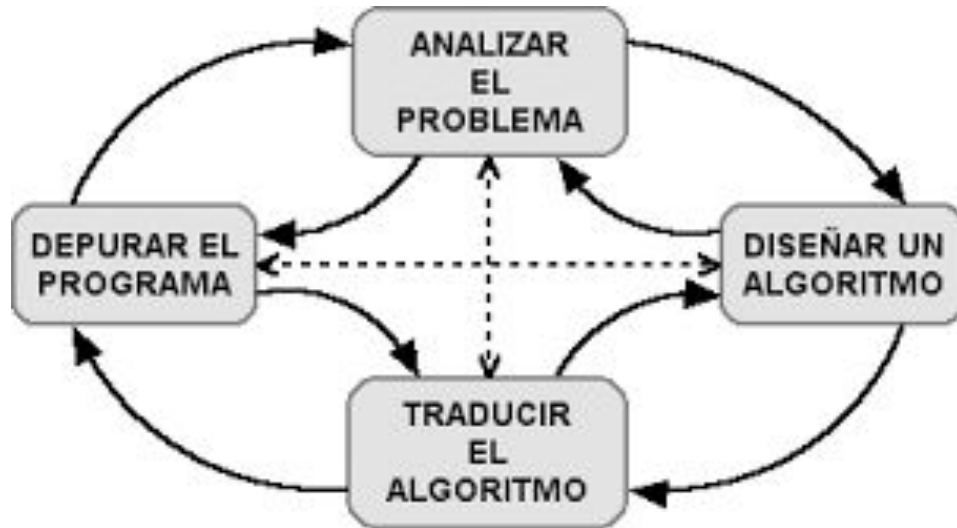
- Simples: usan una celda de memoria



- Compuestas: usan varias celdas de memoria



Resolver problemas



¿QUÉ ES UN ALGORITMO?



Es la **secuencia de pasos** que resuelve un problema y es la base de la programación.

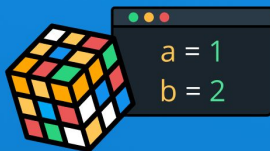
Prof. Alexys Lozada



PARTES DE UN ALGORITMO

ENTRADA

Son los datos que se le dan al algoritmo.



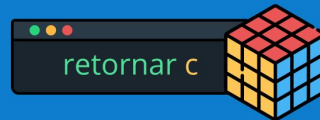
PROCESO

Operaciones que se hacen con los datos.



SALIDA

Resultado final que se obtiene de las operaciones, en este caso será 3.



CARACTERÍSTICAS



PRECISO

Tiene que resolver el problema sin errores.



DEFINIDO

Si ejecutas el algoritmo varias veces, los datos de salida serán iguales en cada repetición.



FINITO

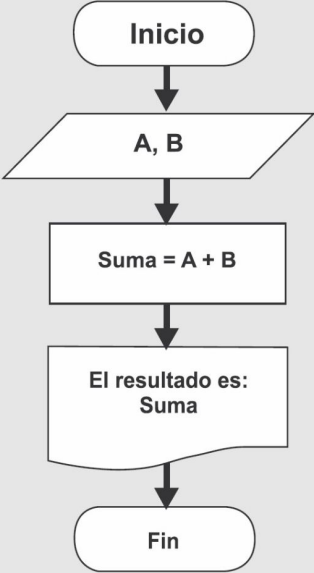
Debe tener un inicio y un final.

LEGIBLE



Cualquier persona que vea el algoritmo debe ser capaz de comprenderlo.

Definición del problema: Elaborar un algoritmo para calcular la suma de dos números y representar el algoritmo gráficamente.











Análisis del problema	Algoritmo	Diagrama de flujo
<p>Entrada:</p> <p><i>A y B representan los dos números.</i></p> <p>Proceso:</p> <p><i>Suma = A + B</i></p> <p>Salida:</p> <p><i>Resultado es Suma</i></p>	<p>1.- INICIO</p> <p>2.- LEER A y A</p> <p>3.- SUMA = A + B</p> <p>4.- IMPRIMIR ("EL RESULTADO ES: SUMA")</p> <p>5.- FIN</p>	 <pre>graph TD; Inicio([Inicio]) --> Input[/A, B/]; Input --> Process[Suma = A + B]; Process --> Output[/El resultado es: Suma/]; Output --> Fin([Fin]);</pre>

Instrucción en Inglés

Begin
End
Read / Input
Write / Print
If ____ then
Else
For
While
Repeat
Until ____

Pseudocódigo en español

Inicio
Fin
Leer / Entrada de Datos
Escribir / Salida de Datos
Si ____ entonces
Sino / Caso Contrario
Desde
Mientras
Repetir
Hasta ____

Símbolo	Significado
	Inicio / Fin del Algoritmo
	Entrada de Datos
	Proceso
	Salida
	Selección Simple/Doble
	Conector en la misma hoja
	Conector en otra hoja
	Ciclo "para"
	Subrutina
	Selección múltiple

Componentes de una página web

HTML



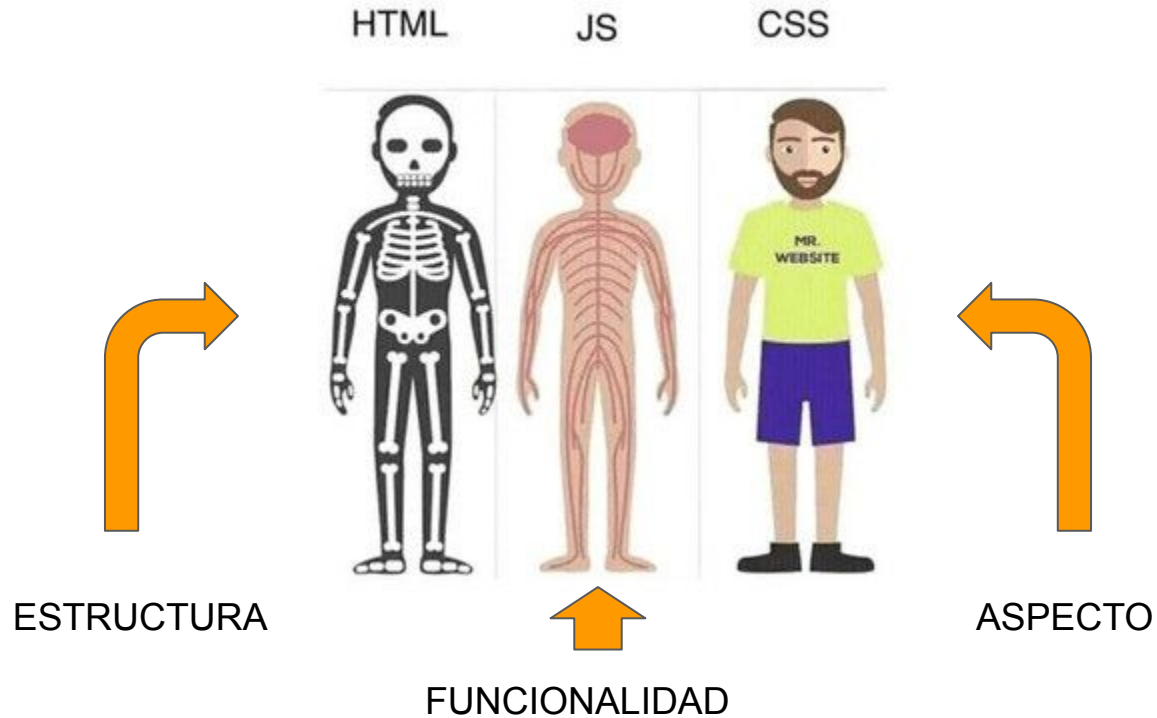
CSS



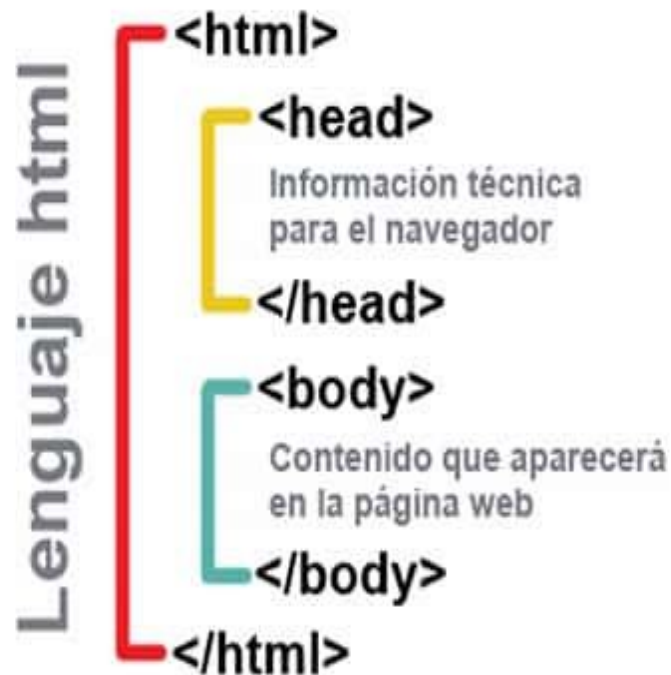
JS



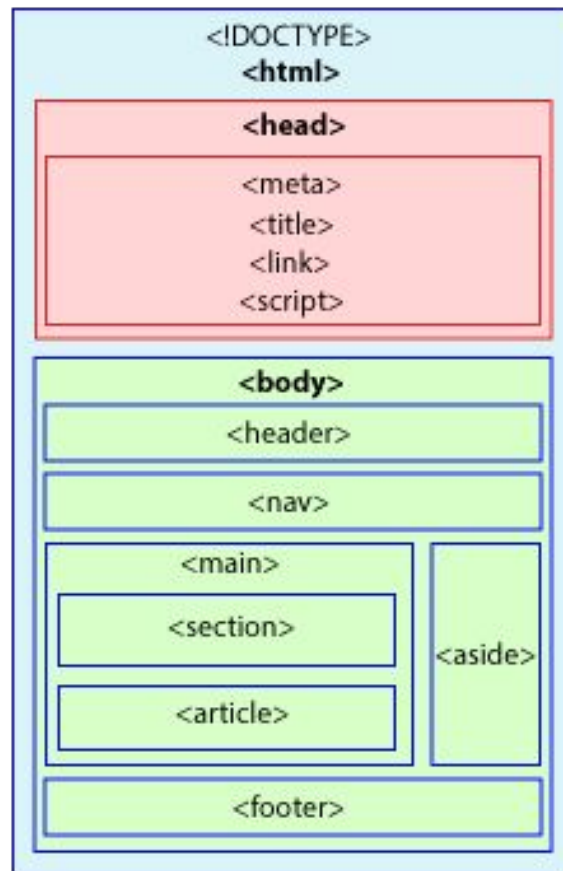
Analogías



Estructura básica



Estructura página web



REGLA CSS

#id

.clase

: estado

[] atributo

* todos



VARIABLES

Una variable puede considerarse como un contenedor donde se pueden almacenar datos, una vez guardada podemos acudir a dicha variable a través del nombre que se le haya asignado.

`{ 1010
0001
1100 }`



DECLARACIÓN Y ASIGNACIÓN

- Palabra reservada de una variable, dependiendo el caso puede ser: (var, let, const)

- Nombre de la variable

```
var mi_variable = "Declarar variables";
```

- Operador de asignación

- Dato asignado a la variable

TIPOS DE DATOS



STRING: `var name = "Juan";`



NUMBER: `var x = 15;`



BOOLEAN: `var pass = true;`



ARRAY: `var animals = ["cat", "dog"];`



OBJECT: `var person = {name: "jon"};`



EMPTY: `var foo = null;`

BUENAS PRÁCTICAS

- 1 El nombre de la variable debe ser un nombre descriptivo que se identifique con el código
- 2 Las variables no se pueden iniciar con números
- 3 No se pueden usar caracteres especiales, ni es recomendable empezar una variable con mayúscula
- 4 Los tipos de escritura de una variable son: lowerCamelCase, UpperCamelCase, snake_case.
- 5 Tener en cuenta que hay nombres que no se pueden usar ya que se encuentran reservadas para el lenguaje.

¿CONOCES LA SINTAXIS DE JAVASCRIPT?

```
1 let language = 'JavaScript']
2 let company = {
3   name: 'EDteam',
4   slogan: 'Nunca te detengas',
5   founded: 2015
6 }
7 console.log(company.name)]
8 // 'EDteam'
9 const getMajorNumber = (a,b) => {
10   if (a > b) { return a }
11   else { return b }
12 }
13 getMajorNumber(4,6)
14 // 6
```

Las variables se declaran con **let** (no hay que indicar el tipo de dato)

Los objetos encierran entre llaves parejas con el formato **propiedad: valor** separadas por comas.

console.log() imprime en consola la expresión entre los paréntesis.

Para obtener el valor de una propiedad de un objeto se usa **objeto.propiedad**

Condicional (**if** / **else**)

Comentarios (**líneas 8 y 14**)

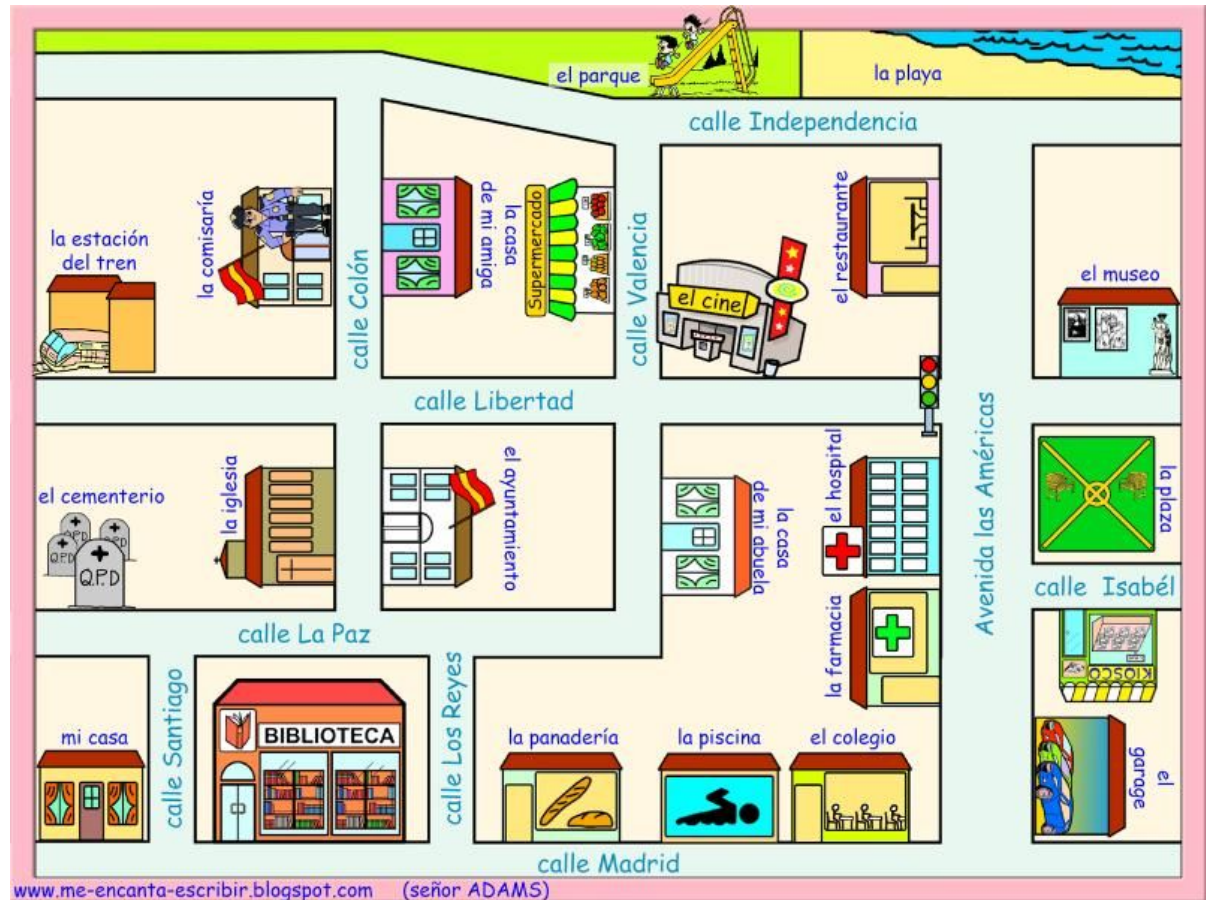
Ejecución de la **función**

Definición de función (se recomienda usar constantes con **const**)



JS

Algoritmos
para indicarle
como llegar al
supermercado



El corralón “Las Piedritas” no pidió una aplicación que les permita calcular la cantidad de ladrillos cocidos necesario para un muro según el largo y el alto del mismo.

$$CL = \frac{1}{(L + J_h) \times (H + J_v)}$$

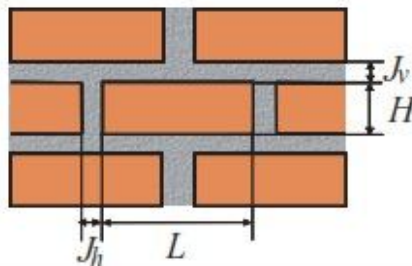
CL = cantidad de ladrillos por m^2

L = longitud de ladrillo (m)

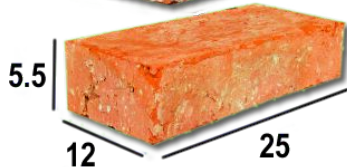
J_h = espesor junta horizontal (m)

H = altura del ladrillo (m)

J_v = espesor junta vertical (m)



con un espesor
de junta de 2cm



LAS PIEDRITAS



Cálculo de la cantidad de
ladrillos Cocido para un
muro

Largo:

Alto:

CALCULAR