



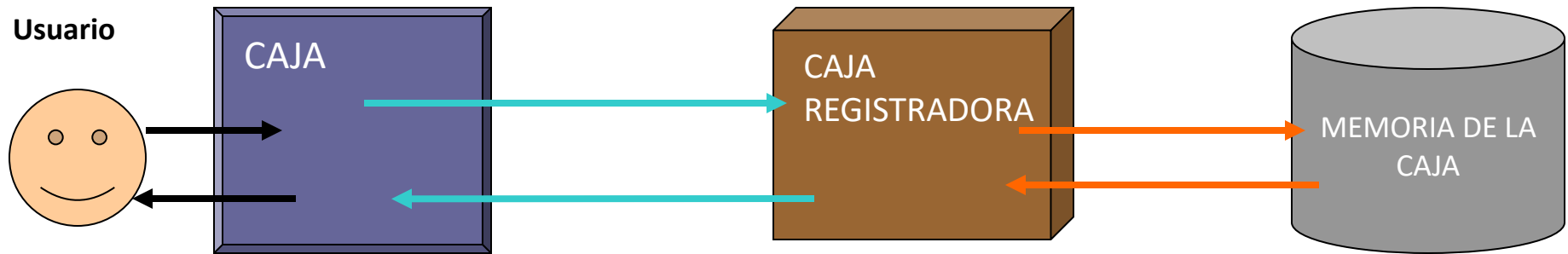
Linkia FP

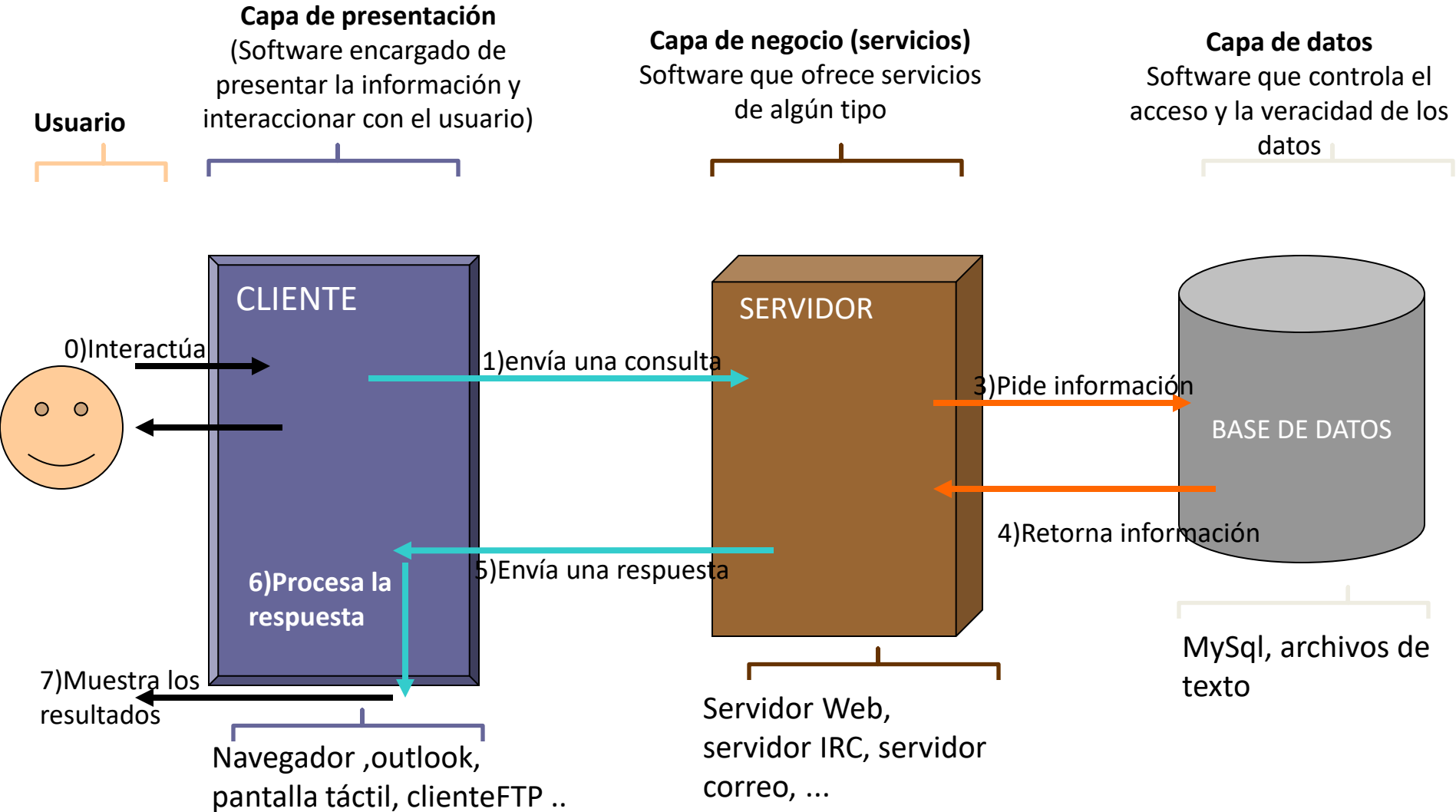
DAW	M06	T01	Material	Arquitecturas web e introducción a JavaScript
-----	-----	-----	----------	---

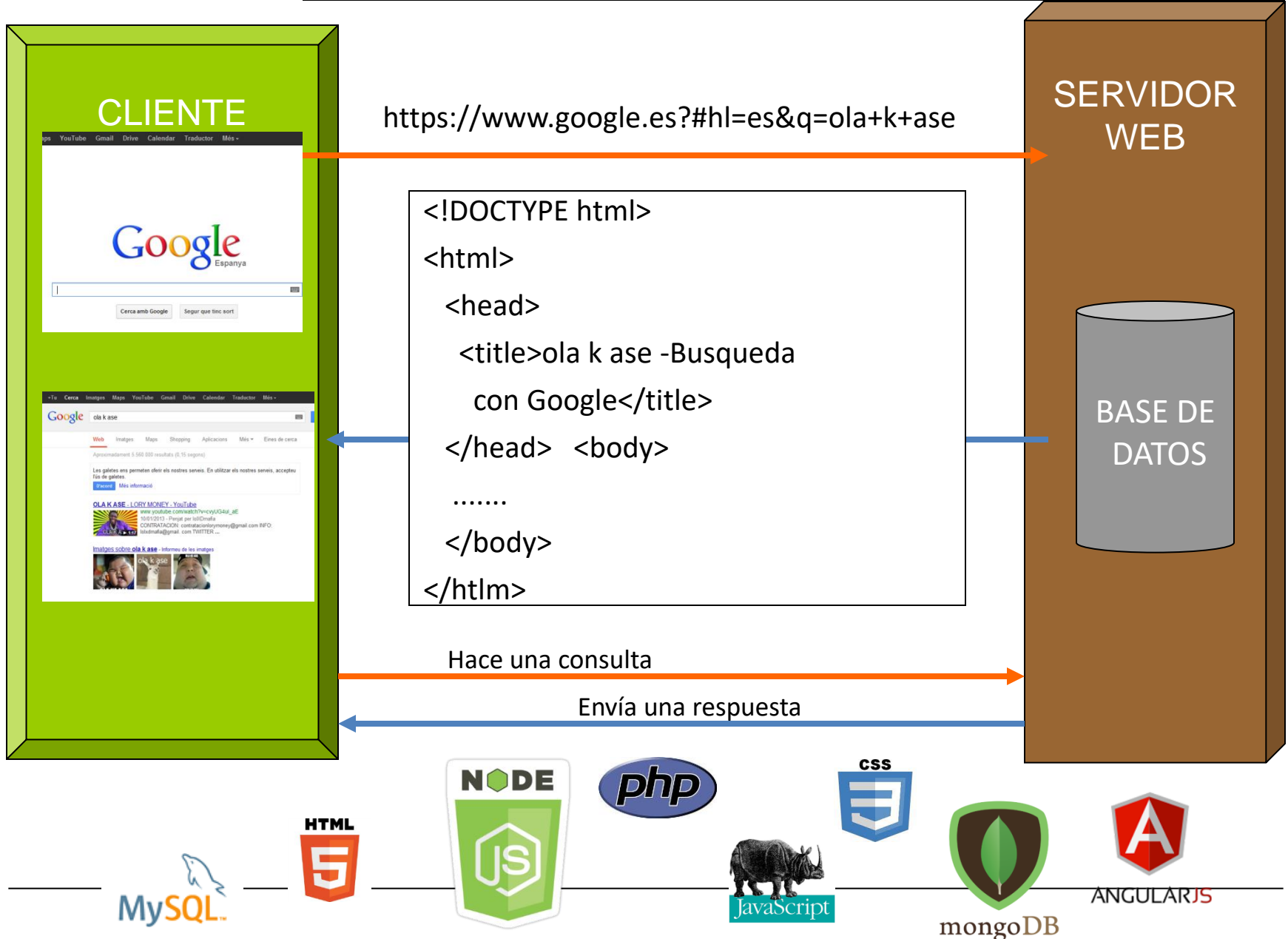
ARQUITECTURA EN CAPAS

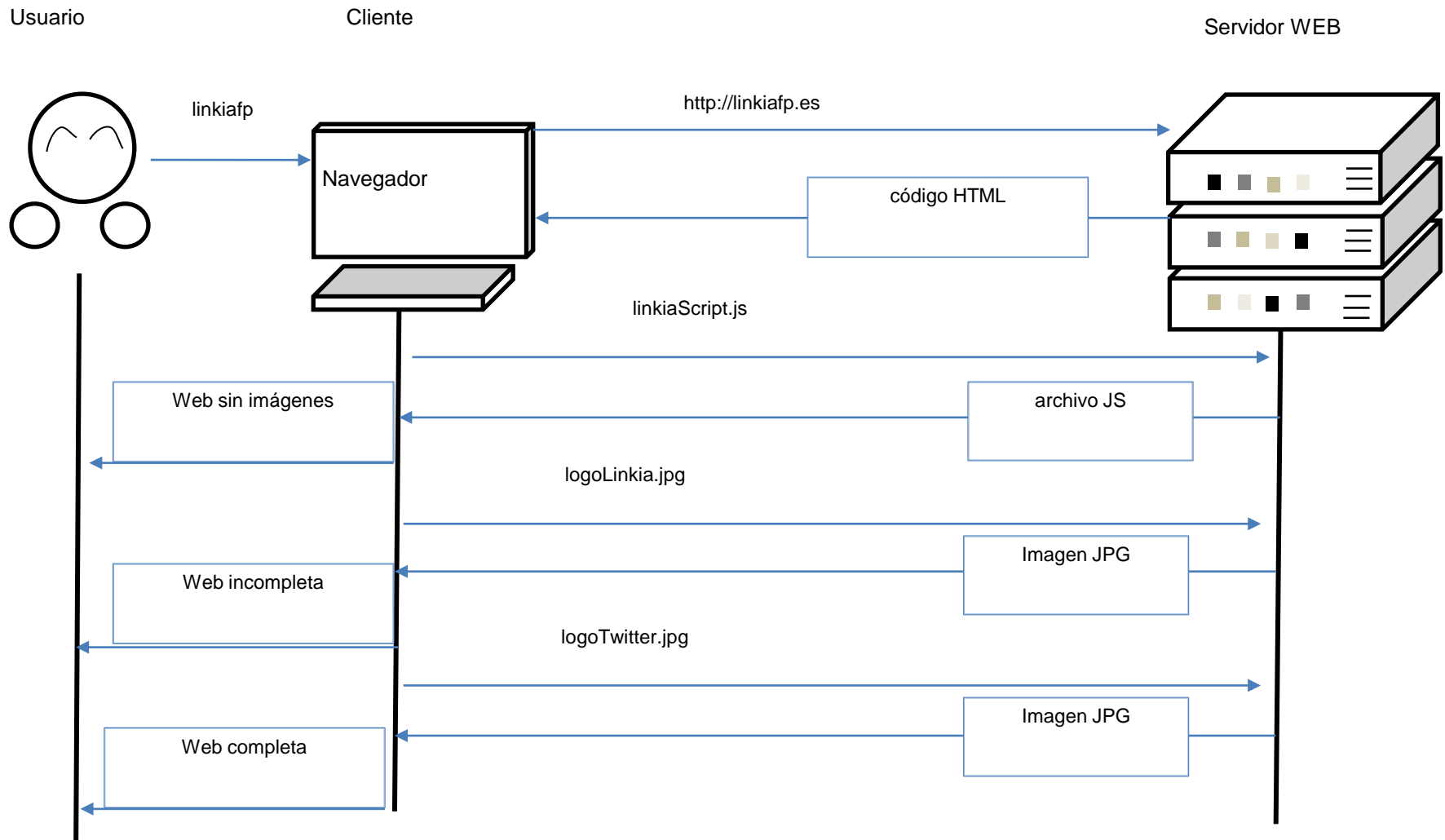
- Se basa en la separación de los distintos roles que intervienen en la resolución de un problema i en la planificación de sus protocolos de comunicación.
- Cada capa como mínimo ha de tener una función i solo ha de realizar operaciones relacionadas con su función.
- Cada capa solo puede interactuar con capas anexas.

Ejemplo de una caja en un supermercado









Librerías



- Conjunto de clases o funciones pensadas para resolver problemas puntuales.
- Las usaremos desde nuestro código cuando nosotros queramos.
- No altera el paradigma de programación de un lenguaje, por ello se pueden incorporar fácilmente en proyectos ya creados.

Framework



- Conjunto de software que define un nuevo paradigma de desarrollo con el objetivo de facilitar la implementación de ciertas funcionalidades genéricas y complejas.
 - Los proyectos ya creados deberán de ser diseñados y programados de nuevo si se quieren implementar un framework.
 - Nosotros configuramos el Framework para que cuando se ejecute haga aquello que nos interesa.
-

Principales

- 1. JavaScript**
- 2. Typescript**
- 3. VisualBasicScript (ActiveX)**
- 4. ActionScript (Flash)**

Objetivo

- 1. Facilitar la vista y la interacción de la web con el usuario.**
- 2. Crear aplicaciones completas.**

Capacidades y limitaciones de ejecución

- Son ejecutados por un intérprete del mismo navegador.
 - El código se comprueba cuando se ejecuta
 - Solo pueden acceder a los recursos delimitados por el navegador
 - El cliente puede no tener el intérprete instalado o habilitado
 - Pueden acceder i modificar los datos de la web y páginas creadas
 - Pueden comunicarse con el servidor
 - El cliente tiene acceso al código
-

Añadir código dentro del HTML

Index.html

```
<script type="text/javascript">  
    alert("Mensaje de alerta con JS");  
</script>
```

Importar un código de un documento .js

Index.html

```
<script type="text/javascript" src="archivo.js">  
</script>
```

archivo.js

```
alert("Alerta!!");
```

Se declaran con la palabra **var** , **let** o **const** seguido del nombre de la variable:

Podemos cambiar el tipo de variable *al vuelo*.

let : variables disponibles en el bloque actual.

var: variables globales.

const: valores inalterables.

```
const vidas_gatos= 7;  
let nombregato = "Isidoro";  
var vidas_restantes = 3.5;  
vidas_restantes="Joselito";  
nombregato = 33;
```

Consideraciones

- "use strict"; al inicio de un .js obliga a declarar las variables.
- No es necesario declarar una variable con **var**, pero es una buena practica.
- No es necesario especificar el tipo de variable.
- Redefinir una variable no implica que pierda su valor.
- No podemos utilizar como nombre: **break,continue, delete,else,false,for,function, if,in,new,null,return, this,true,typeof,var,void,while,with**
- Podemos especificar el tipo de variable:

```
var nombre_gato   = new String;  
var vidas_restantes = new Number;  
var tiene_pedigri  = new Boolean;  
var hijos_gato     = new Array;  
var gato           = new Object
```

```
var vidas_restantes =3;
```

Entrada de datos

```
var numero= window.prompt("incremento de vidas");
```

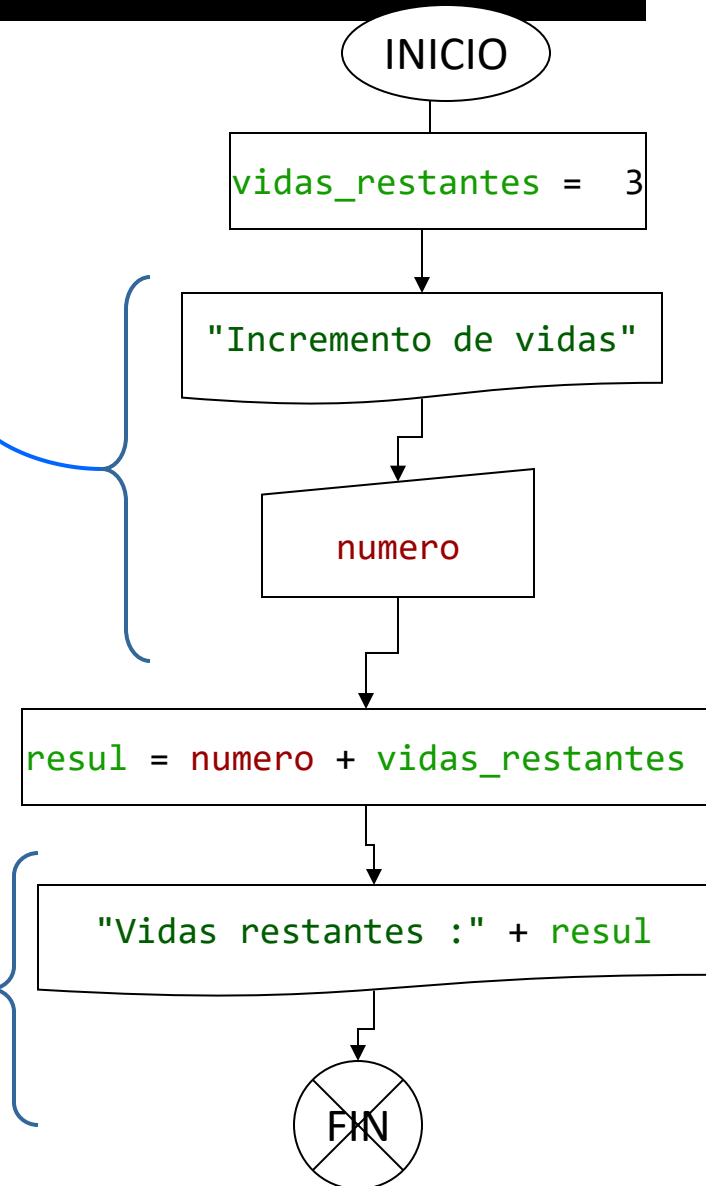
```
var resul = numero+vidas_restantes;
```

Salida de datos

```
window.alert("Vidas restantes"+numero);
```

```
console.log("Vidas restantes"+numero);
```

```
console.info({nombre:"Juan",edad:23});
```



If/ELSE

```
var primerNumero =10;
var segonNumero=5;

var resul = primerNumero+segonNumero;

if( resul>= 20 ){
  alert("resultado mayor o igual que 20");
} else{
  alert("El resultado es menor que 20");
}
```

```
variable = (condicion)? valorTrue:valorfalse;
```

```
var x = (time < 12)? "mdrugada" : "tarde";
```

For

```
for(var cont=0; cont <=10; cont++){
  console.log("repetició número:"+cont);
}
```

While

```
var cont =0;

while(cont <=10){
  console.log("repeticion numero:"+cont);
  cont++;
}

console.log("Valor final:"+cont);
```

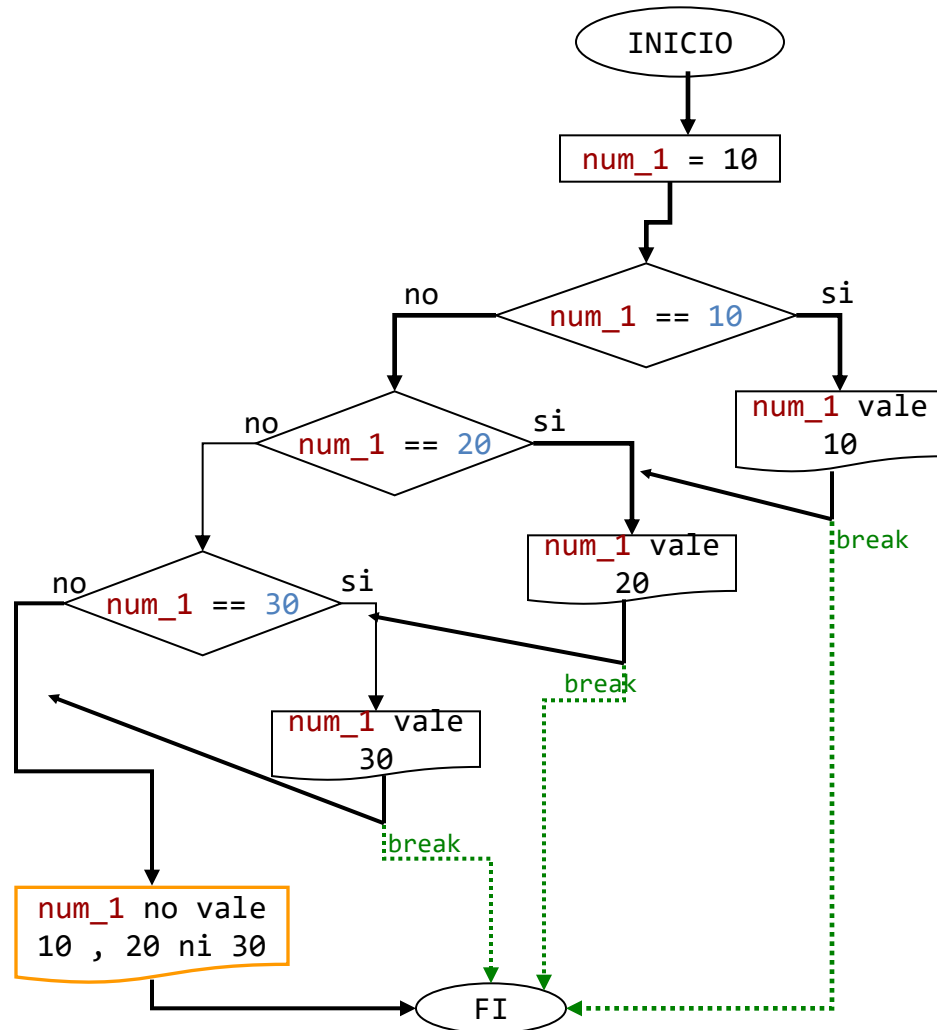
Do While

```
var cont=0;

do{
  console.log("repeticion numero:"+cont);
  cont++;
}while(cont <=10);

console.log("Valor final:"+cont);
```

```
num_1=10;
switch( num_1 ){ //valor a comparar
case "10":
    alert(num_1+" vale 10");
    break;
case 20:
    alert(num_1+" vale 20");
    break;
case 30:
    alert(num_1+" vale 30");
    break;
default:
    alert(num_1+" no vale 10,20 ni 30");
}
```



• Sin el **break**, se ejecutan las acciones del siguiente **case**.

Si estamos dentro de un bucle podemos utilizar **continue** o **break** para alterar el flujo de ejecución:

- **continue**: inicia otra iteración en el bucle validando de nuevo la condición e incremento
- **break** : sale directamente del bucle.
- **return**: sale de la función en ejecución.

```
var i=0;
while (i<12){

    document.write("<br /> Iteración número:"+i);
    i=++i;
    if(i * i > 6){
        if(i==3){
            continue;
        }
        document.write("<br />" + i + "Iteración finalizada");
        break;
    }
}
```

Aritméticos

- suma (+)
- multiplicación (*)
- resta (-)
- división (/)
- resto de la división (%)

x es el resultado de sumar 5 mas el valor de x

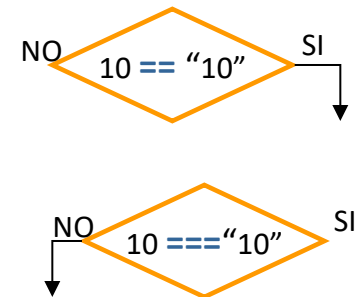
x +=5

x resultado de sumar el valor de x mas 1

x++

De Comparación

- $x == y$ Retorna si x te el mismo valor que y . ($5 == "5"$ && $5 == 5$)
- $x === y$ Retorna si x és igual a y . ($5 === 5$)
- $x != y$ Retorna si x te un valor diferente a y
- $x !== y$ Retorna si x és completamente diferente a y
- $>, <, >=, <=$ Más grande, más petit, más gran o igual, más petit o igual.



Lógicos

- AND (&&) Retorna SI , si las dos comparaciones son ciertas.
- OR (||) Retorna SI , si alguna de les dos comparaciones es cierta.
- NEGACION (!) Si el resultado es SI, retorna NO i viceversa.

LEER Y MODIFICAR CONTENIDO HTML DE UN ELEMENTO

```
•var mySpan= document.getElementById('span1');  
  
var contenidoTag=mySpan.innerHTML;  
  
mySpan.innerHTML = 'Nuevo contenido de span1';
```

EVENTOS Y FUNCIONES EN JS

```
function suma1(a, b=9){  
    var resul = a+b;  
    alert(resul);  
}
```

```
<body>  
    <span id="span1" style="border:0">  
        My Text  
    </span>  
    Text Body  
    <div onclick="suma1(10,90);">  
        Suma 10+9</div>  
</body>
```

```
<div onclick="suma1(10,90);">  
    Suma 10+9</div>  
</body>
```

CONTROLAR O LANZAR EXCEPCIONES

```
var x=4, y=0;
try {
  eval("a er");
  if(y===0){
    throw "el denominador no puede ser 0";
  }
  console.log(x+"/"+y+"=" + (x/y) );
}
catch (err) {
  console.log("Input " + err);
  if( err instanceof SyntaxError){
    console.log("Error SyntaxError");
  }
} finally {
  console.log("si entra o no ") ;
}
```




Linkia FP

Formación Profesional
Oficial a Distancia