

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Creaci3n de arrays

```
1. // Declaraci3n de un array vacio
2. let miArray = [];
3.
4. // Declaraci3n de un array con elementos
5. let numeros = [1, 2, 3, 4, 5];
6. let colores = ["rojo", "verde", "azul"];
7.
8. // Declaraci3n de un array con diferentes tipos de datos
9. let mezclado = [1, "dos", true];
10.
11. // Declaraci3n de una matriz (array bidimensional)
12. let matriz = [
13.     [1, 2, 3],
14.     [4, 5, 6],
15.     [7, 8, 9]
16. ];
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Creaci3n de arrays

```
1. let numeros = [10, 20, 30];  
2. console.log(numeros[0]); // Salida: 10  
3. console.log(numeros[2]); // Salida: 30
```



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

Ejemplo de acceso a una matriz en JavaScript: `Com matriz[1][2]` accede al elemento en la segunda fila (índice 1) y la tercera columna (índice 2), que es 6.

```
1. // Declaración de una matriz (array bidimensional)
2. let matriz = [
3.   [1, 2, 3],
4.   [4, 5, 6],
5.   [7, 8, 9]
6. ];
7.
8. // Acceso al elemento en la segunda fila y la tercera columna
9. let elemento = matriz[1][2];
10. console.log(elemento); // Output: 6
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Modificaci3n de elementos

```
1. let colores = ["rojo", "verde", "azul"];  
2. colores[1] = "amarillo";  
3. console.log(colores); // Salida: ["rojo", "amarillo", "azul"]
```

# *Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión*

## *4. Lenguajes de guión general.*

### *4.6 Listas arrays.*

#### *Métodos de arrays*

*Los arrays en JavaScript vienen con una variedad de métodos integrados que facilitan la manipulación de sus elementos. Veamos algunos métodos comunes:*



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### M3todos de arrays

#### push()

El m3todo push() agrega uno o m3s elementos al final de un array y devuelve la nueva longitud del array.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### M3todos de arrays

#### push()

```
1. let numeros = [1, 2, 3];  
2. numeros.push(4);  
3. console.log(numeros); // Salida: [1, 2, 3, 4]
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### M3todos de arrays

#### pop()

El m3todo pop() elimina el 3ltimo elemento de un array y lo devuelve.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### M3todos de arrays

#### pop()

```
1. let numeros = [1, 2, 3];  
2. let ultimo = numeros.pop();  
3. console.log(ultimo); // Salida: 3  
4. console.log(numeros); // Salida: [1, 2]
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

#### Métodos de arrays

#### `splice()`

El método `splice()` cambia el contenido de un array eliminando elementos existentes y/o agregando nuevos elementos.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### M3todos de arrays

#### splice()

```
1. let frutas = ["manzana", "banana", "cereza", "d3tiles"];
2. frutas.splice(2, 1); // Elimina 1 elemento desde la posici3n 2
3. console.log(frutas); // Salida: ["manzana", "banana", "d3tiles"]
```



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

### Métodos en JavaScript

Los métodos en JavaScript son funciones integradas que se aplican a objetos para realizar operaciones específicas.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

### M3todos en JavaScript

Uso	M3todo	Descripci3n
Conversi3n de Datos	<code>parseInt()</code>	Convierte una cadena en un n3mero entero.
	<code>parseFloat()</code>	Convierte una cadena en un n3mero decimal.
	<code>isNaN()</code>	Determina si un valor es NaN (Not a Number).
Formato de N3meros	<code>toFixed()</code>	Formatea un n3mero utilizando notaci3n de punto fijo con un n3mero espec3fico de d3gitos a la derecha del punto decimal.
Interacci3n con el Usuario	<code>prompt()</code>	Muestra un cuadro de di3logo con un mensaje para que el usuario ingrese datos.
	<code>alert()</code>	Muestra un cuadro de di3logo con un mensaje de advertencia.
Registro y Depuraci3n	<code>console.log()</code>	Muestra un mensaje en la consola del navegador o en la consola del sistema.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

### Métodos en JavaScript

Manipulación de Arrays	<code>concat()</code>	Combina dos o más arrays y devuelve un nuevo array.
	<code>join()</code>	Une todos los elementos de un array en una cadena y devuelve esa cadena.
	<code>pop()</code>	Elimina el último elemento de un array y lo devuelve.
	<code>push()</code>	Agrega uno o más elementos al final de un array y devuelve la nueva longitud del array.
	<code>splice()</code>	Cambia el contenido de un array eliminando elementos existentes y/o agregando nuevos elementos.
	<code>reverse()</code>	Invierte el orden de los elementos de un array.
	<code>sort()</code>	Ordena los elementos de un array localmente y devuelve el array ordenado.
	<code>forEach()</code>	Ejecuta una función por cada elemento presente en el array.
	<code>toString()</code>	Convierte un array a una cadena de texto y devuelve esta cadena.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

### Métodos en JavaScript

Manipulación de Arrays	<code>concat()</code>	Combina dos o más arrays y devuelve un nuevo array.
	<code>join()</code>	Une todos los elementos de un array en una cadena y devuelve esa cadena.
	<code>pop()</code>	Elimina el último elemento de un array y lo devuelve.
	<code>push()</code>	Agrega uno o más elementos al final de un array y devuelve la nueva longitud del array.
	<code>splice()</code>	Cambia el contenido de un array eliminando elementos existentes y/o agregando nuevos elementos.
	<code>reverse()</code>	Invierte el orden de los elementos de un array.
	<code>sort()</code>	Ordena los elementos de un array localmente y devuelve el array ordenado.
	<code>forEach()</code>	Ejecuta una función por cada elemento presente en el array.
	<code>toString()</code>	Convierte un array a una cadena de texto y devuelve esta cadena.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

### M3todos en JavaScript

Manipulaci3n de Cadenas de Texto	<code>toUpperCase()</code>	Devuelve la cadena de texto en may3sculas.
	<code>toLowerCase()</code>	Devuelve la cadena de texto en min3sculas.
	<code>indexOf()</code>	Devuelve la posici3n de la primera ocurrencia de un valor especificado en una cadena.
	<code>lastIndexOf()</code>	Devuelve la posici3n de la 3ltima ocurrencia de un valor especificado en una cadena.
	<code>slice()</code>	Extrae una secci3n de una cadena y devuelve una nueva cadena.
	<code>replace()</code>	Reemplaza una parte de la cadena con otra.
	<code>split()</code>	Divide una cadena en un array de subcadenas.
	<code>trim()</code>	Elimina los espacios en blanco del principio y del final de una cadena.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

### M3todos en JavaScript

Manipulaci3n de Cadenas de Texto	<code>toUpperCase()</code>	Devuelve la cadena de texto en may3sculas.
	<code>toLowerCase()</code>	Devuelve la cadena de texto en min3sculas.
	<code>indexOf()</code>	Devuelve la posici3n de la primera ocurrencia de un valor especificado en una cadena.
	<code>lastIndexOf()</code>	Devuelve la posici3n de la 3ltima ocurrencia de un valor especificado en una cadena.
	<code>slice()</code>	Extrae una secci3n de una cadena y devuelve una nueva cadena.
	<code>replace()</code>	Reemplaza una parte de la cadena con otra.
	<code>split()</code>	Divide una cadena en un array de subcadenas.
	<code>trim()</code>	Elimina los espacios en blanco del principio y del final de una cadena.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

Manipulaci3n de Cadenas de Texto	<code>toUpperCase()</code>	Devuelve la cadena de texto en may3sculas.
	<code>toLowerCase()</code>	Devuelve la cadena de texto en min3sculas.
	<code>indexOf()</code>	Devuelve la posici3n de la primera ocurrencia de un valor especificado en una cadena.
	<code>lastIndexOf()</code>	Devuelve la posici3n de la 3ltima ocurrencia de un valor especificado en una cadena.
	<code>slice()</code>	Extrae una secci3n de una cadena y devuelve una nueva cadena.
	<code>replace()</code>	Reemplaza una parte de la cadena con otra.
	<code>split()</code>	Divide una cadena en un array de subcadenas.
	<code>trim()</code>	Elimina los espacios en blanco del principio y del final de una cadena.

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Declaraci3n de un array bidimensional

Veamos un ejemplo sencillo de un array de 2 X 2:

```
1. var nuevoArray = new Array(2);  
2. // En cada posici3n de nuevoArray guardamos un nuevo array  
3. nuevoArray[0] = new Array(2);  
4. nuevoArray[1] = new Array(2);
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Declaraci3n de un array bidimensional

#### Acceder a los datos de arrays bidimensionales

```
1. // Declaramos el array bidimensional
2. var nuevoArray = new Array(2);
3. nuevoArray[0] = new Array(2);
4. nuevoArray[1] = new Array(2);
5. // Metemos un dato en cada posici3n
6. nuevoArray[0][0] = 25;
7. nuevoArray[0][1] = 12;
8. nuevoArray[1][0] = 34;
9. nuevoArray[1][1] = 6;
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4.Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Declaraci3n de un array bidimensional

El resultado ser3a algo como:

TABLA	COLUMNA 0	COLUMNA 1
FILA 0	posici3n 0,0 = 25	posici3n 0,1 = 12
FILA 1	posici3n 1,0 = 34	posici3n 1,1 = 6

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

#### Declaración de un array bidimensional

Si queremos acceder al array anterior lo haríamos accediendo a cada posición del array de la siguiente forma

```
1. // Declaración del array de 10 posiciones
2. var nuevoArray = new Array(10);
3. // Bucle para meter en cada posición otros array de 10
4. for(var i=0; i<10; i++) {
5.     nuevoArray[i] = new Array(10);
6. }
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Recorrer arrays bidimensionales

Si para recorrer un array de una posici3n utiliz3bamos un bucle for, haremos lo mismo pero usando dos bucles for, uno dentro de otro.

```
1. // Bucle que recorre el primer array
2. for(var i=0; i<10; i++) {
3.     // Bucle que recorre el array que est3 en la posici3n i
4.     for(var j=0; j<10; j++) {
5.         document.write(nuevoArray[i][j]);
6.     }
7. }
```



## Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

### 4. Lenguajes de guión general.

#### 4.6 Listas arrays.

##### Recorrer arrays bidimensionales

En el ejemplo anterior sabemos que se trata de un array bidimensional de  $10 \times 10$ , pero puede que no conozcamos el tamaño del array. Para estos casos vamos a ver un ejemplo utilizando el método `length`

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

#### Recorrer arrays bidimensionales

```
1. // Bucle que recorre el primer array
2. for(var i=0; i<nuevoArray.length; i++) {
3.     // Bucle que recorre el array que est1 en la posici3n i
4.     for(var j=0; j<nuevoArray[i].length; j++) {
5.         document.write(nuevoArray[i][j]);
6.     }
7. }
```

# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.





# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.





# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.





# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.





# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.





# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

## EJERCICIO IMPORTANTE



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

### SOLUCIÓN





# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

## EJERCICIO IMPORTANTE



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

## EJERCICIO IMPORTANTE



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

## EJERCICIO IMPORTANTE



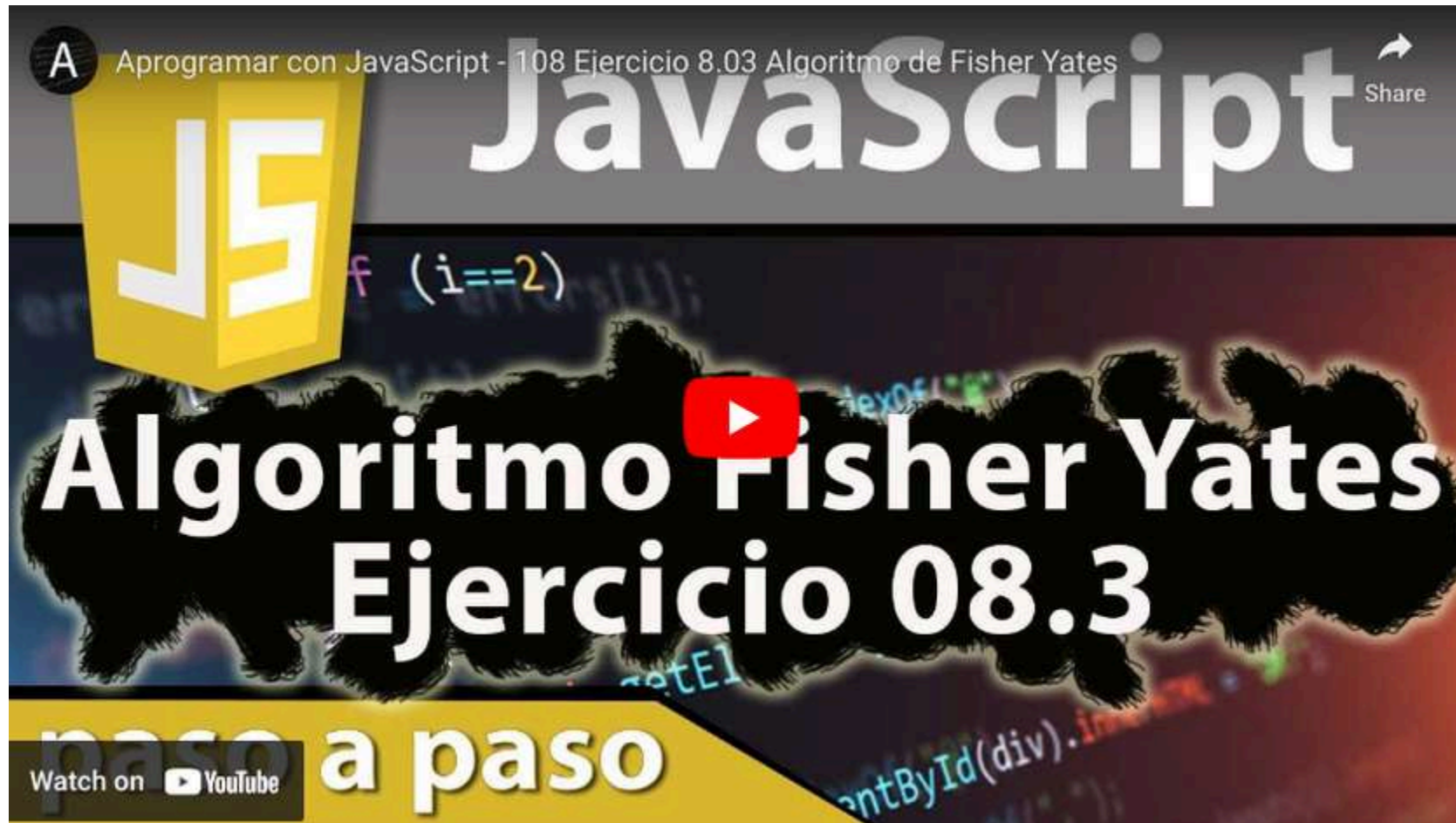


# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

## EJERCICIO IMPORTANTE

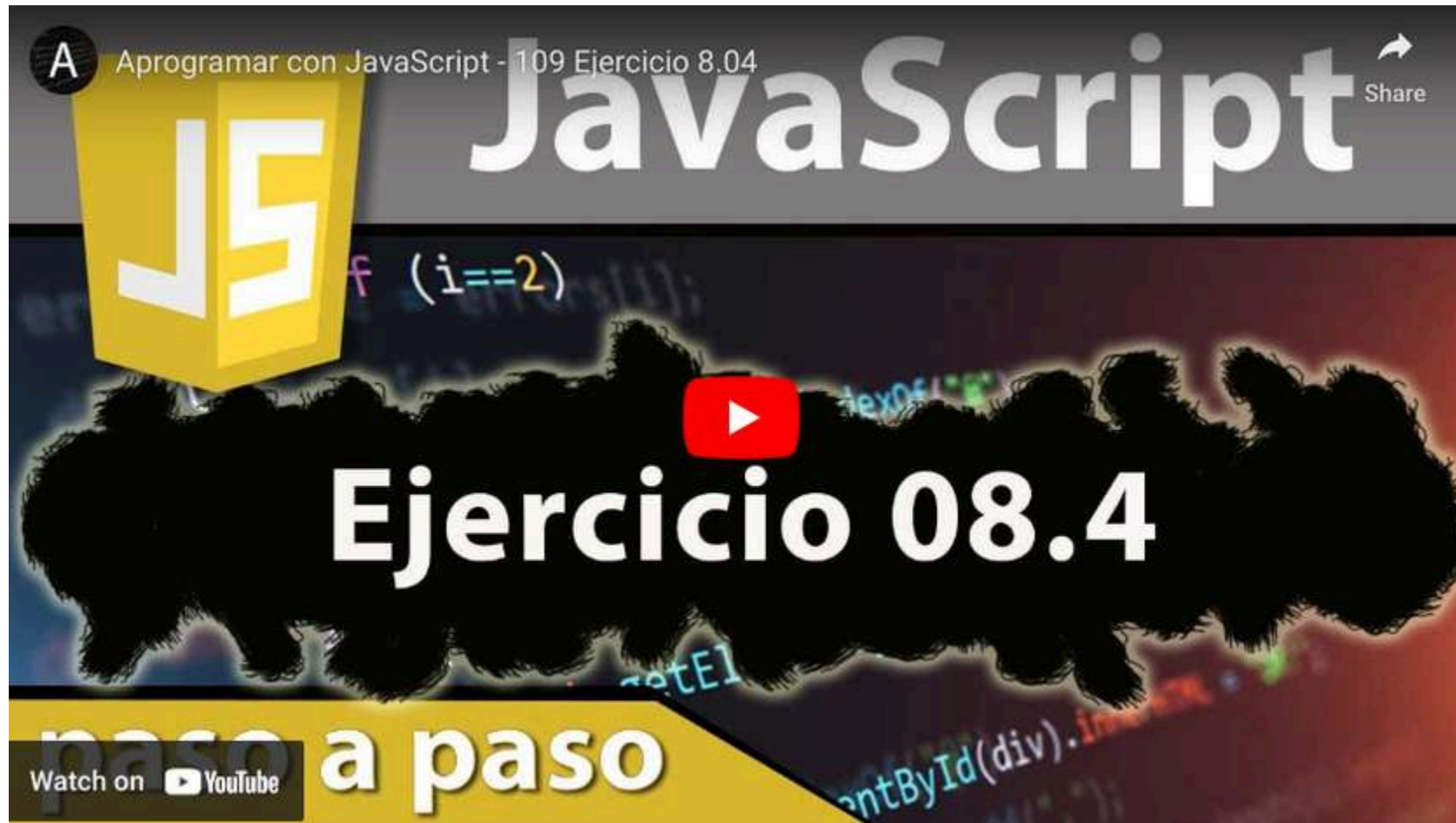


# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

## EJERCICIO IMPORTANTE

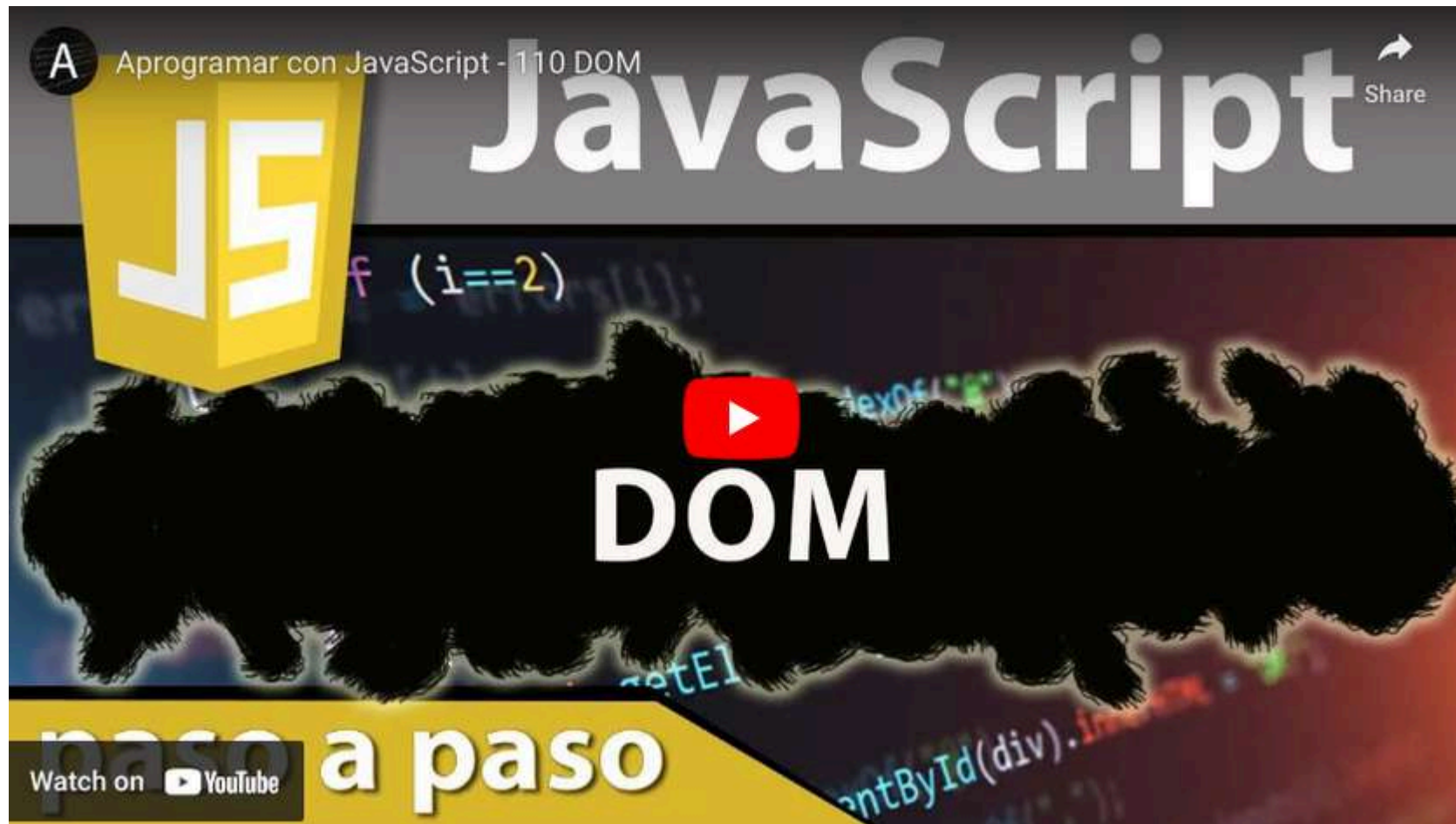


# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

DOM





# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

#### DOCUMENTO



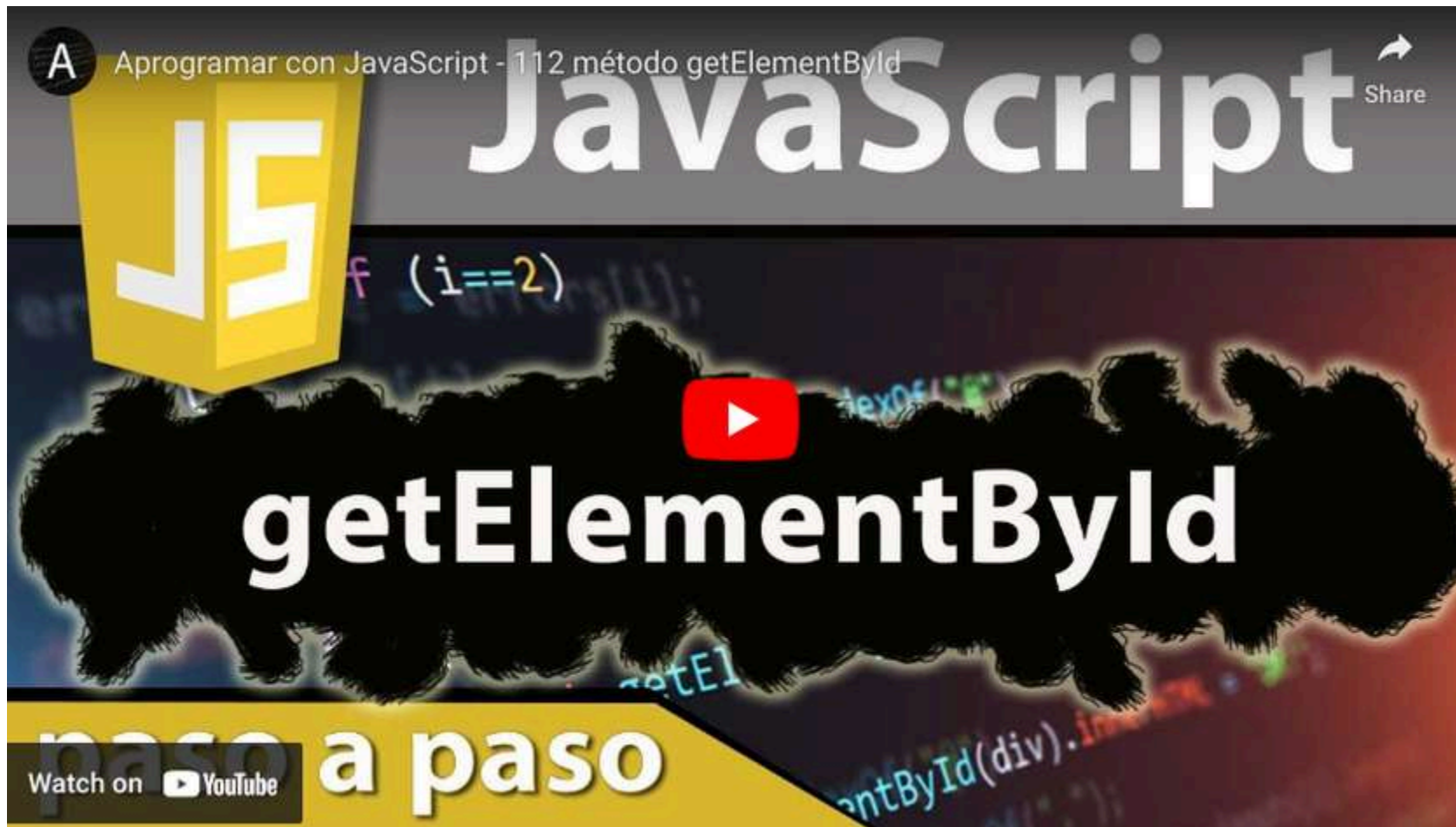


# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

método getElementById



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

m3todo getElementById



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de gui3n

## 4. Lenguajes de gui3n general.

### 4.6 Listas arrays.

m3todo getElementById



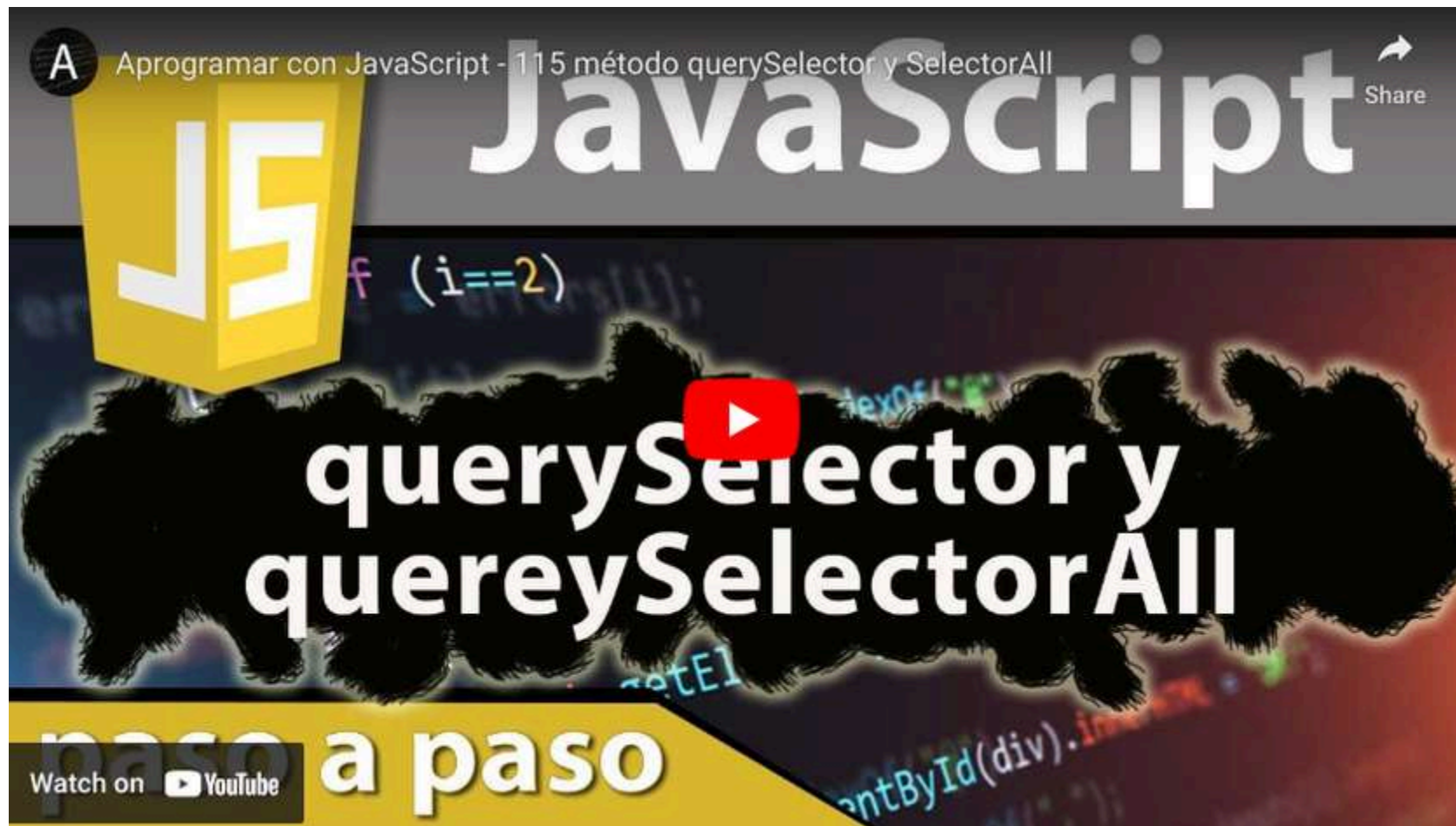


# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

método `querySelector` y `SelectorAll`



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

propiedad `innerText` vs `textContent`



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

#### La herencia (teoría)



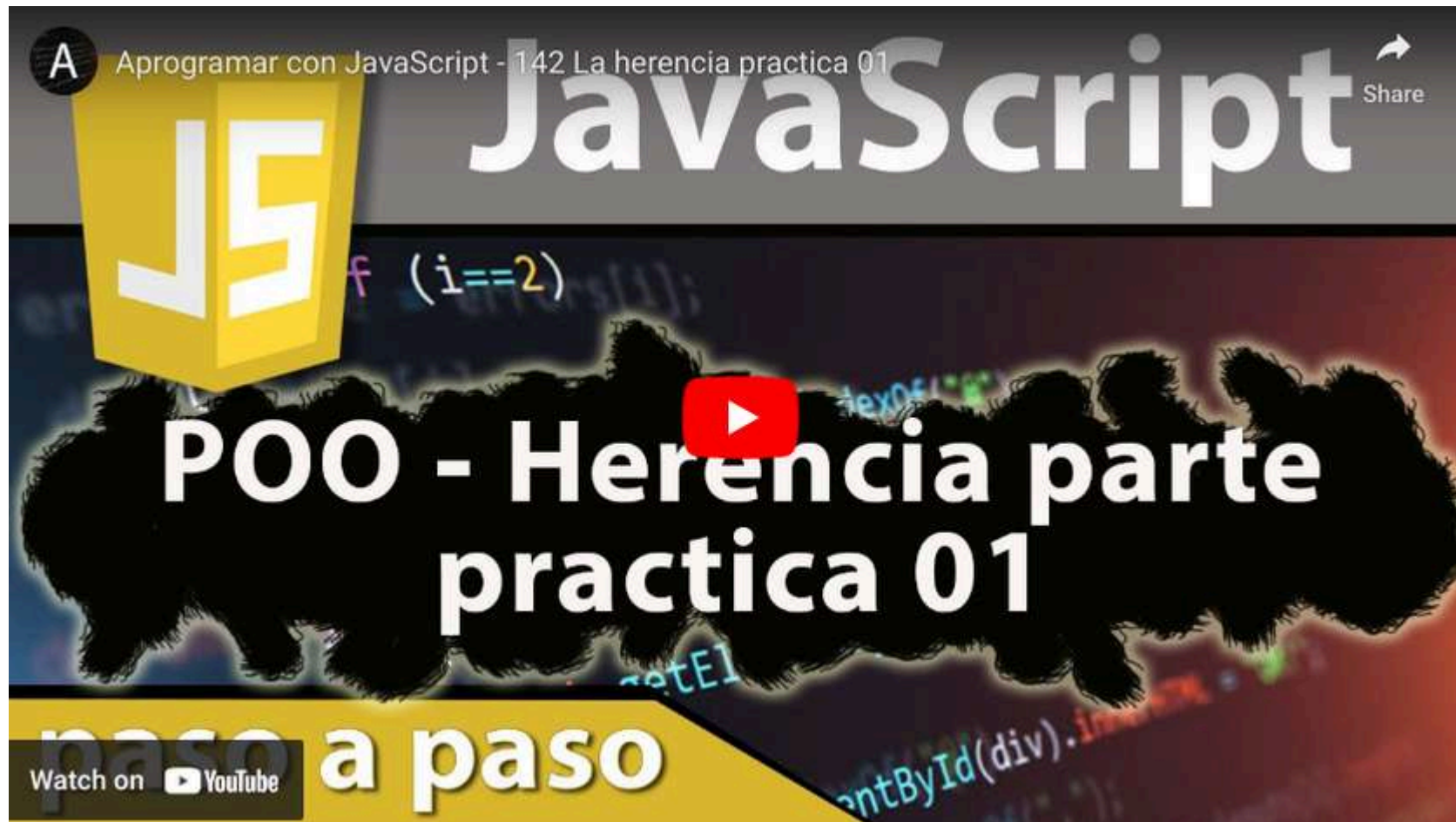


# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

#### La herencia practica 01



# Unidad 2: Desarrollo y reutilización de componentes software y multimedia mediante lenguajes de guión

## 4. Lenguajes de guión general.

### 4.6 Listas arrays.

#### La herencia practica 02

