

# Simulacro

DESARROLLO WEB EN ENTORNO CLIENTE

**Técnico de Grado Superior Desarrollo de Aplicaciones Web**

**2024-25**



J. Mario Rodríguez  
jrodper183e@g.educaand.es

# Contenido

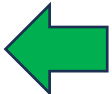
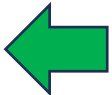
- El examen consta de tres partes:
  - test
  - preguntas de respuesta corta
  - ejercicios de desarrollo
- Lea atentamente cada pregunta antes de responder
- En las preguntas tipo test, seleccione ÚNICAMENTE la respuesta correcta
- En las preguntas de respuesta corta, razone y explique de forma *concisa*
- En los ejercicios de desarrollo, implemente el código solicitado
  - Utiliza comentarios donde sea oportuno para aclarar decisiones de desarrollo
    - No comentar el efecto trivial / evidente del código.



**NETFLIX**

**CLIP  
OFFICIAL**

# TEST

- ¿Qué método del DOM se utiliza para crear un nuevo elemento?
  - ☐ newElement()
  - ☐ createElement() 
  - ☐ buildElement()
  - ☐ createNode()
- ¿Qué operador se utiliza para la comparación estricta en JavaScript?
  - ☐ ==
  - ☐ === 
  - ☐ !=
  - ☐ !==

# TEST

- ¿Cuál es la forma correcta de declarar una variable con ámbito de bloque en ES6?

- ☐ var
- ☐ const
- ☐ let
- ☐ global

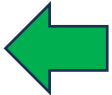



- ¿Qué método se utiliza para añadir un nuevo nodo como último hijo de un elemento?

- ☐ innerChild
- ☐ addLast
- ☐ insertLast
- ☐ appendChild



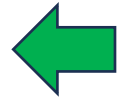
# TEST

- ¿Cómo se previene el envío de un formulario en un manejador de eventos?
  - ☐ return false;
  - ☐ stopPropagation()
  - ☐ preventDefault() 
  - ☐ cancelSubmit()
- ¿Qué método de los objetos *Array* permite crear una nueva matriz con todos los elementos que pasan la prueba implementada por la función proporcionada?
  - ☐ select()
  - ☐ choose()
  - ☐ match()
  - ☐ filter() 

# TEST

- ¿Cómo se crea un nuevo elemento de texto?

- ☐ `createTextNode();`



- ☐ `newTextElement()`

- ☐ `addText()`

- ☐ `innerText()`

- ¿Cómo se obtiene la longitud de un array?

- ☐ `count`

- ☐ `size()`

- ☐ `length`

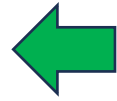
- ☐ `getLength()`



# TEST

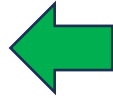
- ¿Qué evento se dispara cuando se presiona una tecla?

- ☐ onkeydown
- ☐ onkeypress
- ☐ onkeyup
- ☐ ontyping



- ¿Cuál es el resultado de: "5" + 3?

- ☐ 53
- ☐ undefined
- ☐ 8
- ☐ NaN





# PREGUNTAS PRÁCTICAS

- Explica las limitaciones de la ejecución de código JavaScript en el navegador cliente.



Respuesta orientativa!

Acceso limitado al sistema de archivos, ya que JavaScript no puede leer o escribir archivos directamente en el sistema del usuario sin la intervención del usuario.

Dependencia de capacidades del navegador, ya que cada navegador e incluso cada versión del navegador puede implementar unas u otras funcionalidades de cada tecnología web. A veces la implementación de JavaScript varía ligeramente, pero por ejemplo las características más modernas de JavaScript no estarán disponibles en navegadores antiguos.

Posible desactivación de la ejecución de JavaScript por usuario, que no es muy habitual, pero podría darse.

Rendimiento limitado para tareas complejas, ya que JavaScript no accede directamente al hardware.

# PREGUNTAS PRÁCTICAS

- Explica la diferencia entre el operador `==` y `===` en JavaScript, y proporciona un ejemplo que ilustre su comportamiento. ¿Cuándo usarías uno u otro?

`==` realiza conversión de tipo antes de comparar  
`===` compara valor y tipo sin conversión

Ejemplo:

`5 == "5"` devuelve `true`

`5 === "5"` devuelve `false`

Se recomienda usar `===` para comparaciones más estrictas y predecibles

Respuesta orientativa!

# PREGUNTAS PRÁCTICAS

- Un formulario de registro solicita una contraseña. Describe qué validaciones implementarías y qué utilizarías.

Respuesta orientativa!

Validaciones:

Campo requerido

Longitud mínima (por ejemplo 8 caracteres)

Combinación de mayúsculas, minúsculas, números y símbolos

...

Utilizaría validación en cliente con JavaScript, campos y atributos adecuados en html (input type password, atributo required...), expresiones regulares para comprobar la complejidad, implementación del listener del evento onsubmit para hacer la validación y información visual de la validez o no al usuario mediante html y CSS.

# PREGUNTAS PRÁCTICAS

- Explique el concepto de delegación de eventos y su utilidad en la gestión de eventos en JavaScript.

Respuesta orientativa!

Consiste en añadir un único manejador de eventos a un elemento padre, en lugar de a cada uno de sus hijos.

Gestiona eventos para múltiples elementos hijos, optimiza el uso de memoria, ya que los listeners son relativamente pesados en cliente. Además facilita añadir dinámicamente elementos.

# PREGUNTAS PRÁCTICAS

- ¿Cuál es la salida de este código?

Hello, undefined

```
function greet(name) {  
    console.log("Hello, " + name + "!");  
}  
greet();
```

- ¿Cuál es la salida de este código?

20

10

```
let x = 10;  
if (x === 10) {  
    let x = 20;  
    console.log(x);  
}  
console.log(x);
```

# PREGUNTAS PRÁCTICAS

- ¿Cuál es la salida de este código?

[2, 4, 6, 8, 10]

```
let numbers = [1, 2, 3, 4, 5];  
let result = numbers.map(x => x * 2);  
console.log(result);
```

- ¿Habrá algún error en este código?

Sí, el objeto person no tiene la propiedad address

```
let person = {  
  name: "John Doe",  
  age: 30  
};  
console.log(person.address);
```

# PREGUNTAS PRÁCTICAS

- Completa el código

```
items.forEach(item => {  
  const li = document.createElement("li");  
  li.textContent = item;  
  list.appendChild(li);  
});
```

```
<ul id="myList"></ul>  
<script>  
  const items = ["apple", "banana", "cherry"];  
  const list = document.getElementById("myList");  
  // Completa el código para agregar los items a la lista  
</script>
```

- ¿Habrá algún error en este código?

No, no habrá error. Aunque person se declaró como constante, el contenido del objeto sí puede modificarse.

```
const person = { name: "Alice" };  
person.name = "Bob";  
console.log(person.name);
```

# DESARROLLO

- Crea una página web que muestre una imagen y un botón. Implementa comportamiento en el clic del botón de modo que al pulsarlo se cambie la fuente de la imagen.

```
document.getElementById("myImage").src = "new_image.jpg";
```



# DESARROLLO

- Completa el código para alternar la clase "hidden" del div cuando se hace clic en el botón.

```
<button onclick="toggleClass()">Toggle Class</button>
<div id="myDiv"></div>
<style>
    .hidden {
        display: none;
    }
</style>
<script>
    function toggleClass() {
        // Completa el código para alternar la clase "hidden"
    }
</script>
```

```
const div = document.getElementById("myDiv");
div.classList.toggle("hidden");
```

# DESARROLLO

- Implementa un contador de visitas que almacene el número de veces que el usuario de ese navegador ha cargado una página y lo muestre en pantalla. Utiliza cookies para persistir el valor entre visitas.

# DESARROLLO

- Algún problema de comunicación asíncrona

