



## Aprendizaje Esperado N°3



Actividad:

### Explorando Arreglos y Ciclos en JavaScript

**Objetivo:** Comprender arreglos, ciclos iterativos y buenas prácticas en JavaScript.

**Descripción:** Los estudiantes realizarán ejercicios prácticos para familiarizarse con la creación y manipulación de arreglos, así como con la implementación de ciclos iterativos (for, while, do/while). También aplicarán buenas prácticas de codificación y explorarán operaciones con arreglos.



#### INSTRUCCIONES:

- Revisa las instrucciones específicas.
- Desarrolla todo lo solicitado.
- Sigue las indicaciones de entrega.

#### INDICACIONES



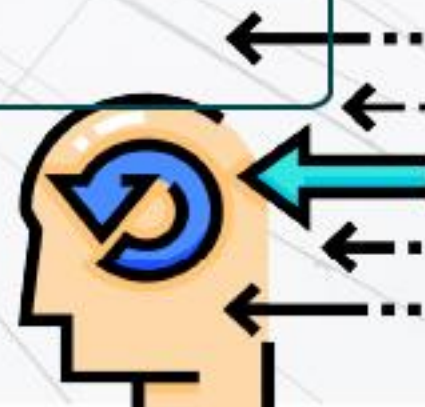
Tipo de Entrega: .rar, .zip, Repositorio GitHub.



Tiempo: 60 minutos.



Número de participantes. Individual o en grupos de 3 personas.



## INSTRUCCIONES ESPECIFICAS

### 1. Explorando Arreglos en JavaScript

- Define un arreglo con 5 elementos de tu elección.
- Accede al tercer elemento del arreglo y muéstralo en la consola.
- Usa `push()` para agregar un nuevo elemento y `pop()` para eliminar el último elemento, mostrando los resultados en la consola (captura pantalla).

### 2. Iterando sobre un Arreglo

- Crea un ciclo `for` que recorra todos los elementos de un arreglo y los imprima en la consola.
- Usa un `forEach()` para lograr lo mismo y compara ambos métodos.
- Captura la salida de la consola.

### 3. Operaciones con Arreglos

- Crea dos arreglos de números y encuentra su unión y diferencia.
- Concatenar dos arreglos en uno solo y mostrarlo en la consola.
- Implementa un filtro para mostrar solo los números mayores a 10 dentro de un arreglo.

### 4. Aplicando Ciclos Iterativos

- Implementa un ciclo `while` que sume los números del 1 al 10.
- Usa un ciclo `do/while` que imprima los números del 5 al 15.
- Implementa un `for` anidado que genere una tabla de multiplicar del 1 al 5.
- Captura los resultados en la consola.

### 5. Buenas Prácticas y Código Limpio

- Organiza tu código usando indentación adecuada.
- Usa nombres de variables descriptivos.
- Explica por qué seguir buenas prácticas ayuda a mantener código entendible y escalable.