

# Reporte de Evaluación - Fork de GitHub

## Información General

Estudiante: Julio Eduardo Correa  
Repositorio: Jcorrea24/act\_web1\_s9  
Fecha de evaluación: 5/10/2025, 13:04:54  
Evaluado por: Sistema de Evaluación Masiva

## Resumen de Calificaciones

Calificación general: 2.3/5.0  
Actividades completadas: 10/10  
Porcentaje de completitud: 100.0%

## Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Información Básica del Documento - Crea ...	ejercicios/ejercicio_01.js	Sí	1.0
2	Seleccionar Elementos por ID - Crea una ...	ejercicios/ejercicio_02.js	Sí	1.0
3	Seleccionar Elementos por Clase - Crea e...	ejercicios/ejercicio_03.js	Sí	1.0
4	Cambiar Contenido de Elementos - Crea el...	ejercicios/ejercicio_04.js	Sí	5.0
5	Modificar Atributos - Crea elementos con...	ejercicios/ejercicio_05.js	Sí	5.0
6	Agregar y Quitar Clases CSS - Crea eleme...	ejercicios/ejercicio_06.js	Sí	1.0
7	Crear y Agregar Elementos - Crea nuevos ...	ejercicios/ejercicio_07.js	Sí	2.0
8	Eventos Básicos - Crea elementos interac...	ejercicios/ejercicio_08.js	Sí	5.0
9	Formularios y Validación Simple - Crea u...	ejercicios/ejercicio_09.js	Sí	1.0
10	Navegación entre Elementos - Crea una es...	ejercicios/ejercicio_10.js	Sí	1.0

## Retroalimentación Detallada

**Actividad 1: Información Básica del Documento - Crea una página HTML simple que muestre: Título de la página usando document.title, URL actual usando document.URL, Mostrar esta información en un div en la página**

Archivo esperado: ejercicios/ejercicio\_01.js  
Estado: Archivo encontrado  
Calificación: 1.0/5.0  
Retroalimentación:  
El código presentado no corresponde a la descripción del ejercicio. Se implementó un inventario de productos en lugar de mostrar el título y la URL del documento HTML.

**Actividad 2: Seleccionar Elementos por ID - Crea una página con varios elementos que tengan ID y: Usa getElementById() para seleccionar elementos, Cambia el texto de los elementos seleccionados, Cambia el color de fondo de un elemento**

Archivo esperado: ejercicios/ejercicio\_02.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código no cumple con la descripción de la actividad (seleccionar elementos por ID, cambiar texto y color de fondo). En su lugar, implementa filtros sobre un array de productos. La lógica de filtrado es correcta, pero no corresponde al problema planteado.

**Actividad 3: Seleccionar Elementos por Clase - Crea elementos con la misma clase y: Usa getElementsByClassName() para seleccionarlos, Cambia el estilo de todos los elementos de esa clase, Cuenta cuántos elementos tienen esa clase**

Archivo esperado: ejercicios/ejercicio\_03.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código entregado no corresponde a la descripción de la actividad (selección de elementos por clase y manipulación del DOM). En cambio, se enfoca en manipulación de arrays con map, reduce, etc. Necesitas revisar y completar la actividad solicitada.

**Actividad 4: Cambiar Contenido de Elementos - Crea elementos con texto y: Usa.textContent para cambiar el texto, Usa.innerHTML para agregar HTML, Crea un botón que cambie el contenido al hacer clic**

Archivo esperado: ejercicios/ejercicio\_04.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta, clara y eficiente. Utiliza `reduce()` de manera efectiva para calcular los resultados solicitados. El código está bien estructurado y es fácil de entender.

**Actividad 5: Modificar Atributos - Crea elementos con atributos y: Usa.getAttribute() para leer atributos, Usa.setAttribute() para cambiar atributos, Cambia el src de una imagen y el href de un enlace**

Archivo esperado: ejercicios/ejercicio\_05.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y resuelve correctamente todos los requerimientos utilizando las funciones de array de manera eficiente. Se cumplen todas las buenas prácticas.

**Actividad 6: Agregar y Quitar Clases CSS - Crea elementos con estilos CSS y: Usa.classList.add() para agregar clases, Usa.classList.remove() para quitar clases, Usa.classList.toggle() para alternar clases**

Archivo esperado: ejercicios/ejercicio\_06.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código presentado no corresponde a la descripción de la actividad (Agregar y Quitar Clases CSS). Se enfoca en manipulación de arrays en lugar de clases CSS. Es necesario revisar el enunciado y corregir la solución.

**Actividad 7: Crear y Agregar Elementos - Crea nuevos elementos dinámicamente: Usa createElement() para crear elementos, Usa appendChild() para agregarlos al DOM, Crea una lista de elementos con un botón**

Archivo esperado: ejercicios/ejercicio\_07.js

Estado: Archivo encontrado

Calificación: 2.0/5.0

Retroalimentación:

El código se enfoca en ordenamiento de arrays y no en la manipulación del DOM con createElement y appendChild como se solicitaba. El código proporcionado es correcto en cuanto a ordenamiento, pero no cumple con el objetivo principal de la actividad.

**Actividad 8: Eventos Básicos - Crea elementos interactivos: Usa addEventListener() para eventos de click, Maneja eventos de mouseover y mouseout, Cambia elementos cuando ocurran los eventos**

Archivo esperado: ejercicios/ejercicio\_08.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y cumple con todos los requisitos de la actividad. Demuestra un buen entendimiento de las técnicas de manipulación de arrays y desestructuración.

**Actividad 9: Formularios y Validación Simple - Crea un formulario simple y: Obtén valores de inputs con value, Valida que los campos no estén vacíos, Muestra mensajes de error o éxito**

Archivo esperado: ejercicios/ejercicio\_09.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código presentado no corresponde a la descripción de la actividad (Formularios y Validación). Se evalúa la solución proporcionada, no la intención.

**Actividad 10: Navegación entre Elementos - Crea una estructura HTML y demuestra: Usa parentElement para acceder al elemento padre, Usa children para acceder a elementos hijos, Usa nextElementSibling para el siguiente hermano, Muestra la información de navegación en la página**

Archivo esperado: ejercicios/ejercicio\_10.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código proporcionado no corresponde a la descripción de la actividad (Navegación entre Elementos del DOM). Se enfoca en manipulación de objetos, no en la navegación del DOM. Es necesario que la solución se centre en el manejo de `parentElement`, `children`, y `nextElementSibling` en un documento HTML.

## Resumen General

Necesita mejorar. Completó 10/10 actividades (100%) con una calificación promedio de 2.3/5. Se recomienda revisar los conceptos fundamentales.

## Recomendaciones

- Revisar y mejorar las actividades con calificación baja
- Enfocarse en mejorar la documentación y comentarios del código
- Aplicar mejores prácticas de programación