

# Reporte de Evaluación - Fork de GitHub

## Información General

Estudiante: Roque Aldana  
Repositorio: Julio28012020/act\_web1\_s9  
Fecha de evaluación: 5/10/2025, 13:14:24  
Evaluado por: Sistema de Evaluación Masiva

## Resumen de Calificaciones

Calificación general: 4.7/5.0  
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	5.0
2	Seleccionar Elementos por ID - Crea una ...	ejercicios/ejercicio_02.js	Sí	5.0
3	Seleccionar Elementos por Clase - Crea e...	ejercicios/ejercicio_03.js	Sí	5.0
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	5.0
7	Crear y Agregar Elementos - Crea nuevos ...	ejercicios/ejercicio_07.js	Sí	4.0
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	4.0
10	Navegación entre Elementos - Crea una es...	ejercicios/ejercicio_10.js	Sí	4.0

## 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: 5.0/5.0  
Retroalimentación:  
Excelente solución. El código es claro, funcional y sigue las buenas prácticas al usar DOMContentLoaded para asegurar que el script se ejecute después de que el HTML se haya cargado.

**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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y cumple con todos los requisitos de la actividad. Bien hecho.

**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: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con todos los requisitos. El código es limpio, bien estructurado y utiliza buenas prácticas. La lógica para destacar y contar productos es clara y funcional.

**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:

Excelente solución. El código es claro, funcional y demuestra un buen entendimiento del uso de `textContent` e `innerHTML` para manipular el DOM. Bien implementado el reseteo.

**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, funcional y cumple con todos los requisitos de la actividad. Bien hecho!

**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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y cumple con todos los requisitos de la actividad, utilizando correctamente los métodos de `classList`.

**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: 4.0/5.0

Retroalimentación:

La solución cumple con los requisitos de la actividad. Buen uso de `createElement` y `appendChild`. Considerar la posibilidad de refactorizar para evitar repetición de código (e.g., el `getElementById('contenedor')`).

**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, funcional y cumple con todos los requisitos de la actividad. Bien hecho usando ``classList.toggle`` para el botón y gestionando el estado visual y el mensaje correctamente.

**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: 4.0/5.0

Retroalimentación:

La solución cumple con los requisitos básicos. Se podría mejorar la validación del email con una expresión regular más robusta y la función ``limpiar`` no se está utilizando.

**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: 4.0/5.0

Retroalimentación:

La solución cumple con los requisitos y el código es funcional. Se podría mejorar la presentación visual de la información en la página, además de añadir manejo de errores en caso de no existir un elemento siguiente.

## Resumen General

Excelente trabajo. Completó 10/10 actividades (100%) con una calificación promedio de 4.7/5. Demuestra buen dominio de los conceptos.

## Recomendaciones

- Continuar con el excelente trabajo y mantener la calidad del código