

# Reporte de Evaluación - Fork de GitHub

## Información General

Estudiante: Julio Correa  
Repositorio: Jcorrea24/act\_web1\_s7  
Fecha de evaluación: 11/9/2025, 14:23:18  
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	Gestión de Inventario Básico - Crea un a...	ejercicios/ejercicio_01.js	Sí	5.0
2	Filtrado de Productos por Categoría - Us...	ejercicios/ejercicio_02.js	Sí	4.0
3	Transformación de Datos con map() - Crea...	ejercicios/ejercicio_03.js	Sí	5.0
4	Análisis de Ventas con reduce() - Dado u...	ejercicios/ejercicio_04.js	Sí	4.0
5	Búsqueda y Verificación - Crea un array ...	ejercicios/ejercicio_05.js	Sí	5.0
6	Manipulación de Arrays - Crea un array i...	ejercicios/ejercicio_06.js	Sí	5.0
7	Ordenamiento y Reversión - Crea arrays d...	ejercicios/ejercicio_07.js	Sí	5.0
8	Desestructuración de Arrays - Dado el ar...	ejercicios/ejercicio_08.js	Sí	4.0
9	Desestructuración de Objetos - Crea un o...	ejercicios/ejercicio_09.js	Sí	5.0
10	Métodos de Objeto - Crea un objeto y dem...	ejercicios/ejercicio_10.js	Sí	5.0

## Retroalimentación Detallada

**Actividad 1: Gestión de Inventario Básico - Crea un array de objetos que represente un inventario de productos. Cada producto debe tener: id, nombre, precio, categoria, stock. Declara al menos 5 productos y muestra todos los productos, total de productos en inventario y valor total del inventario.**

Archivo esperado: ejercicios/ejercicio\_01.js  
Estado: Archivo encontrado  
Calificación: 5.0/5.0  
Retroalimentación:

La solución es correcta y cumple con todos los requisitos. El código es limpio, fácil de entender y utiliza buenas prácticas como el uso de `reduce` para calcular el valor total del inventario.

**Actividad 2: Filtrado de Productos por Categoría - Usando el inventario del ejercicio anterior, utiliza el método filter() para mostrar solo productos de la categoría 'Electrónicos', productos con stock menor a 10 y productos con precio mayor a \$500.**

Archivo esperado: ejercicios/ejercicio\_02.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional, pero la descripción de la actividad solicitaba filtrar por 'Electrónicos', no 'Ropa'. Se podrían mejorar los nombres de las variables para reflejar mejor los filtros aplicados (precio en lugar de precioAlto).

**Actividad 3: Transformación de Datos con map() - Crea un array de estudiantes con nombre, edad, notas (array de números). Usa map() para crear un nuevo array con solo los nombres, crear un array con el promedio de cada estudiante y agregar una propiedad estado ('Aprobado' si promedio  $\geq 70$ , 'Reprobado' si  $< 70$ ).**

Archivo esperado: ejercicios/ejercicio\_03.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y completa. El código es legible y utiliza `map` y `reduce` de manera eficiente para lograr los resultados deseados.

**Actividad 4: Análisis de Ventas con reduce() - Dado un array de ventas con producto, cantidad, precio, fecha. Usa reduce() para calcular total de ingresos, producto más vendido (por cantidad) y promedio de venta por transacción.**

Archivo esperado: ejercicios/ejercicio\_04.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional, usando `reduce` de manera efectiva. Se podría mejorar la legibilidad del código en la parte de `productoMasVendido` extrayendo la lógica a funciones más pequeñas y usando nombres de variables más descriptivos.

**Actividad 5: Búsqueda y Verificación - Crea un array de usuarios con id, nombre, email, activo. Implementa búsquedas usando find() para buscar usuario por email, findIndex() para obtener posición de usuario por id, some() para verificar si hay usuarios inactivos y every() para verificar si todos tienen email válido (contiene @).**

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 de la actividad, usando adecuadamente los métodos `find`, `findIndex`, `some` y `every`.

**Actividad 6: Manipulación de Arrays - Crea un array inicial [1, 2, 3, 4, 5] y demuestra push() y pop() (agregar y quitar del final), shift() y unshift() (agregar y quitar del inicio), splice() (insertar elementos en posición específica) y slice() (extraer porción sin modificar original).**

Archivo esperado: ejercicios/ejercicio\_06.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y demuestra correctamente el uso de los métodos de array solicitados. Muy buena presentación con los `console.log` mostrando el estado del array en cada paso.

**Actividad 7: Ordenamiento y Reversión - Crea arrays de números desordenados (ordena ascendente y descendente), nombres de personas (ordena alfabéticamente), objetos con propiedad edad (ordena por edad) y usa reverse() para invertir el orden.**

Archivo esperado: ejercicios/ejercicio\_07.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. Se utilizan buenas prácticas como la copia de arrays para evitar mutaciones.

**Actividad 8: Desestructuración de Arrays - Dado el array ['JavaScript', 'Python', 'Java', 'C++', 'Go']: extrae los primeros 3 lenguajes, extrae el primero y el último, usa rest operator para separar el primero del resto e intercambia dos variables usando desestructuración.**

Archivo esperado: ejercicios/ejercicio\_08.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Podrías mejorar la extracción del primer y último elemento usando desestructuración para mayor concisión y consistencia con el resto del código.

**Actividad 9: Desestructuración de Objetos - Crea un objeto persona con propiedades anidadas (dirección, contacto). Demuestra desestructuración básica, renombrado de variables, valores por defecto, desestructuración anidada y rest operator en objetos.**

Archivo esperado: ejercicios/ejercicio\_09.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. Demuestra correctamente todos los conceptos de desestructuración solicitados de forma clara y concisa. El código es limpio y fácil de entender.

**Actividad 10: Métodos de Objeto - Crea un objeto y demuestra Object.keys() (obtener claves), Object.values() (obtener valores), Object.entries() (obtener pares clave-valor) e iterar sobre el objeto con forEach().**

Archivo esperado: ejercicios/ejercicio\_10.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y demuestra correctamente el uso de los métodos Object.keys(), Object.values(), Object.entries() y la iteración con forEach().

## 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