

# Reporte de Evaluación - Fork de GitHub

## Información General

Estudiante: Estudiante desconocido  
Repositorio: Lizeth64/act\_web1\_s7  
Fecha de evaluación: 5/10/2025, 11:57:30  
Evaluado por: Sistema de Evaluación de No Calificados

## Resumen de Calificaciones

Calificación general: 0.0/5.0  
Actividades completadas: 0/10  
Porcentaje de completitud: 0.0%

## Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Gestión de Inventario Básico - Crea un a...	src/ejercicio_01.js	Sí	0.0
2	Filtrado de Productos por Categoría - Us...	src/ejercicio_02.js	Sí	0.0
3	Transformación de Datos con map() - Crea...	src/ejercicio_03.js	Sí	0.0
4	Análisis de Ventas con reduce() - Dado u...	src/ejercicio_04.js	Sí	0.0
5	Búsqueda y Verificación - Crea un array ...	src/ejercicio_05.js	Sí	0.0
6	Manipulación de Arrays - Crea un array i...	src/ejercicio_06.js	Sí	0.0
7	Ordenamiento y Reversión - Crea arrays d...	src/ejercicio_07.js	Sí	0.0
8	Desestructuración de Arrays - Dado el ar...	src/ejercicio_08.js	Sí	0.0
9	Desestructuración de Objetos - Crea un o...	src/ejercicio_09.js	Sí	0.0
10	Métodos de Objeto - Crea un objeto y dem...	src/ejercicio_10.js	Sí	0.0

## 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: src/ejercicio\_01.js  
Estado: Archivo encontrado  
Calificación: 0.0/5.0  
Retroalimentación:  
    Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_02.js  
Estado: Archivo encontrado  
Calificación: 0.0/5.0  
Retroalimentación:  
    Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_03.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_04.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_05.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_06.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_07.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_08.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_09.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

**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: src/ejercicio\_10.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

## Resumen General

Necesita mejorar. Completó 0/10 actividades (0%) con una calificación promedio de 0.0/5. 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