

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Mario Alonso Munera Velez
Repositorio: MarioMunera1993/act_web1_s7
Fecha de evaluación: 11/9/2025, 14:25:18
Evaluado por: Sistema de Evaluación Masiva

Resumen de Calificaciones

Calificación general: 2.6/5.0
Actividades completadas: 6/10
Porcentaje de completitud: 60.0%

Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Gestión de Inventario Básico - Crea un a...	ejercicios/ejercicio_01.js	Sí	4.0
2	Filtrado de Productos por Categoría - Us...	ejercicios/ejercicio_02.js	Sí	5.0
3	Transformación de Datos con map() - Crea...	ejercicios/ejercicio_03.js	Sí	4.0
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	4.0
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	0.0
8	Desestructuración de Arrays - Dado el ar...	ejercicios/ejercicio_08.js	Sí	0.0
9	Desestructuración de Objetos - Crea un o...	ejercicios/ejercicio_09.js	Sí	0.0
10	Métodos de Objeto - Crea un objeto y dem...	ejercicios/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: ejercicios/ejercicio_01.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Se recomienda uniformizar la nomenclatura de las propiedades (ej: 'stock' en lugar de 'Stock') y agregar comentarios para mejorar la legibilidad.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con todos los requisitos. El código es legible y bien estructurado, usando `filter` de forma adecuada. ¡Excelente trabajo!

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 ≥ 70 , 'Reprobado' si < 70).

Archivo esperado: ejercicios/ejercicio_03.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Se podría mejorar la legibilidad usando nombres de variables más descriptivos y simplificar la función `revision` directamente dentro del `map`.

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. Se podría mejorar la legibilidad y eficiencia del cálculo del producto más vendido usando `reduce` de manera más directa para obtener el producto máximo sin necesidad de iterar nuevamente sobre las claves del objeto.

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: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Podrías mejorar la validación de email utilizando expresiones regulares para una mayor precisión.

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:

La actividad se resuelve correctamente, demostrando el uso adecuado de los métodos de array. El código es claro y bien estructurado, facilitando su comprensión.

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: 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: ejercicios/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: ejercicios/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: ejercicios/ejercicio_10.js

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Archivo vacío - sin contenido para evaluar

Resumen General

Necesita mejorar. Completó 6/10 actividades (60%) con una calificación promedio de 2.6/5. 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