Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Irwin Colmenarez
Repositorio: irwincol/act_web1_s7

Fecha de evaluación: 11/9/2025, 14:16:55 Evaluado por: Sistema de Evaluación Masiva

Resumen de Calificaciones

Calificación general: 4.2/5.0 Actividades completadas: 9/10 Porcentaje de completitud: 90.0%

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	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	0.0
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	4.0
8	Desestructuración de Arrays - Dado el ar	ejercicios/ejercicio_08.js	Sí	5.0
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:

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

`forEach` y `reduce` es apropiado y eficiente.

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 completa. El código es legible, bien estructurado y cumple con los requisitos de la

actividad. ¡Buen 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 cumple con los requisitos del ejercicio. Se podría mejorar la legibilidad extrayendo la lógica de cálculo del promedio a una función reutilizable y evitando la mutación innecesaria de objetos en el último `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: 0.0/5.0 Retroalimentación:

Error al procesar la evaluación

Actividad 5: Búsqueda y Verificación - Crea un array de usuarios con id, nombre, email, activo. Implementa búsquedas usando find() para buscar usuario por email, findlndex() 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 cumple con los requisitos, pero la búsqueda del usuario por correo tiene un error (busca un correo inexistente). Considera refactorizar para mejorar la legibilidad y evitar funciones anónimas redundantes.

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 todos los métodos de array solicitados. Bien hecho.

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: 4.0/5.0 Retroalimentación:

La solución es correcta y funcional, cubriendo los requisitos del ejercicio. Sin embargo, sería mejor encapsular la lógica de ordenamiento en funciones para mejorar la legibilidad y reutilización del código.

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: 5.0/5.0 Retroalimentación:

La solución es correcta y completa, abordando todos los requisitos de la actividad con claridad. El código es legible y utiliza las características de desestructuración de arrays de manera eficiente.

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 un entendimiento completo de la desestructuración de objetos en JavaScript, incluyendo todas las características solicitadas. El código es limpio, bien estructurado 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 conforEach().

Archivo esperado: ejercicios/ejercicio_10.js

Estado: Archivo encontrado

Calificación: 5.0/5.0 Retroalimentación:

La solución es correcta y completa, demostrando el uso de Object.keys, Object.values, Object.entries e iterando sobre el objeto con forEach. El código es legible y bien estructurado.

Resumen General

Excelente trabajo. Completó 9/10 actividades (90%) con una calificación promedio de 4.2/5. Demuestra buen dominio de los conceptos.

Recomendaciones

· Revisar y mejorar las actividades con calificación baja