

# Reporte de Evaluación - Fork de GitHub

## Información General

Estudiante: Esteban Garces Alzate  
Repositorio: EstebanGarcesA/act\_web1\_s4  
Fecha de evaluación: 24/8/2025, 23:51:02  
Evaluado por: Sistema de Evaluación

## Resumen de Calificaciones

Calificación general: 4.5/5.0  
Actividades completadas: 20/20  
Porcentaje de completitud: 100.0%

## Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Calcular el cuadrado de un número - Crea...	ejercicios/ejercicio_01.js	Sí	4.0
2	Saludar con nombre por defecto - Crea un...	ejercicios/ejercicio_02.js	Sí	4.0
3	Sumar elementos de un arreglo - Crea una...	ejercicios/ejercicio_03.js	Sí	4.0
4	Ejecutar un mensaje instantáneamente - C...	ejercicios/ejercicio_04.js	Sí	5.0
5	Contar vocales en una cadena - Crea una ...	ejercicios/ejercicio_05.js	Sí	5.0
6	Filtrar números mayores a 10 - Crea una ...	ejercicios/ejercicio_06.js	Sí	5.0
7	Convertir a mayúsculas - Crea una funció...	ejercicios/ejercicio_07.js	Sí	4.0
8	Temporizador con mensaje - Crea una func...	ejercicios/ejercicio_08.js	Sí	5.0
9	Crear un contador privado - Crea una fun...	ejercicios/ejercicio_09.js	Sí	5.0
10	Calcular factorial - Crea una función qu...	ejercicios/ejercicio_10.js	Sí	4.0
11	Verificar si un número es par - Crea una...	ejercicios/ejercicio_11.js	Sí	4.0
12	Multiplicar elementos de un arreglo - Cr...	ejercicios/ejercicio_12.js	Sí	5.0
13	Reemplazar espacios por guiones - Crea u...	ejercicios/ejercicio_13.js	Sí	4.0
14	Generar un ID único - Crea una función q...	ejercicios/ejercicio_14.js	Sí	5.0
15	Invertir una cadena - Crea una función q...	ejercicios/ejercicio_15.js	Sí	3.0
16	Sumar argumentos variables - Crea una fu...	ejercicios/ejercicio_16.js	Sí	5.0
17	Ejecutar operación personalizada - Crea ...	ejercicios/ejercicio_17.js	Sí	5.0
18	Validar correo electrónico - Crea una fu...	ejercicios/ejercicio_18.js	Sí	4.0
19	Retrasar ejecución de un mensaje - Crea ...	ejercicios/ejercicio_19.js	Sí	5.0
20	Calcular promedio de un arreglo - Crea u...	ejercicios/ejercicio_20.js	Sí	4.0

## Retroalimentación Detallada

### Actividad 1: Calcular el cuadrado de un número - Crea una función que reciba un número y devuelva su cuadrado. (Tipo de función: Declaración de función)

Archivo esperado: ejercicios/ejercicio\_01.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función calcula correctamente el cuadrado. Sin embargo, debería retornar el valor en lugar de imprimirlo directamente para mayor flexibilidad. Considera usar 'return producto;' y no imprimirlo en la función.

### Actividad 2: Saludar con nombre por defecto - Crea una función que salude a una persona por su nombre. Si no se proporciona un nombre, usa 'Invitado'. (Tipo de función: Expresión de función)

Archivo esperado: ejercicios/ejercicio\_02.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función cumple con el objetivo. Podrías simplificar la condición `nombre == null` a `!nombre`. Considera usar `return` para retornar el saludo en lugar de `console.log` para mayor flexibilidad.

### Actividad 3: Sumar elementos de un arreglo - Crea una función que sume todos los números de un arreglo. (Tipo de función: Función flecha)

Archivo esperado: ejercicios/ejercicio\_03.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Sería mejor retornar el resultado en lugar de imprimirlo directamente y se podría usar `reduce` para una solución más concisa y funcional.

### Actividad 4: Ejecutar un mensaje instantáneamente - Crea una función que imprima '¡Bienvenido!' en la consola al definirse. (Tipo de función: IIFE)

Archivo esperado: ejercicios/ejercicio\_04.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y utiliza una IIFE para imprimir el mensaje al definirse. El código es limpio y conciso.

### Actividad 5: Contar vocales en una cadena - Crea una función que cuente las vocales (a, e, i, o, u) en una cadena. (Tipo de función: Función recursiva)

Archivo esperado: ejercicios/ejercicio\_05.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. La función recursiva `contarVocales` es clara, concisa y resuelve el problema correctamente. El código es legible y sigue buenas prácticas.

### Actividad 6: Filtrar números mayores a 10 - Crea una función que reciba un arreglo y devuelva solo los números mayores a 10. (Tipo de función: Función de orden superior)

Archivo esperado: ejercicios/ejercicio\_06.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. Utiliza `filter` de forma efectiva y la sintaxis es clara. Buen trabajo.

**Actividad 7: Convertir a mayúsculas - Crea una función que convierta una cadena a mayúsculas. (Tipo de función: Función flecha)**

Archivo esperado: ejercicios/ejercicio\_07.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función resuelve correctamente el problema, pero la salida debería ser retornada en lugar de imprimirse directamente con `console.log`. Considera retornar el valor para mayor flexibilidad y reutilización.

**Actividad 8: Temporizador con mensaje - Crea una función que imprima un mensaje después de 3 segundos usando `setTimeout`. (Tipo de función: Función anónima)**

Archivo esperado: ejercicios/ejercicio\_08.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. Utiliza correctamente `setTimeout` con una función anónima para cumplir con los requisitos.

**Actividad 9: Crear un contador privado - Crea una función que devuelva otra función para contar incrementos, manteniendo el contador privado. (Tipo de función: IIFE con closure)**

Archivo esperado: ejercicios/ejercicio\_09.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Solución correcta y concisa. Implementa un closure de forma adecuada para mantener el estado privado del contador. Código limpio y fácil de entender.

**Actividad 10: Calcular factorial - Crea una función que calcule el factorial de un número. (Tipo de función: Función recursiva)**

Archivo esperado: ejercicios/ejercicio\_10.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función calcula correctamente el factorial de un número de forma recursiva. Se podría mejorar la validación de entrada para retornar un valor en caso de número negativo en lugar de solo imprimirlo en consola.

**Actividad 11: Verificar si un número es par - Crea una función que determine si un número es par. (Tipo de función: Declaración de función)**

Archivo esperado: ejercicios/ejercicio\_11.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función resuelve correctamente el problema. Sin embargo, la variable 'par' es redundante; se puede simplificar el código retornando directamente el resultado de la condición (`num % 2 == 0`).

**Actividad 12: Multiplicar elementos de un arreglo - Crea una función que multiplique todos los números de un arreglo. (Tipo de función: Función de orden superior)**

Archivo esperado: ejercicios/ejercicio\_12.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. Utiliza `reduce` de forma correcta y eficiente para multiplicar todos los elementos del arreglo. El código es conciso y fácil de entender.

**Actividad 13: Reemplazar espacios por guiones - Crea una función que reemplace los espacios en una cadena por guiones. (Tipo de función: Expresión de función)**

Archivo esperado: ejercicios/ejercicio\_13.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Se podría mejorar la legibilidad utilizando el método ``replace`` de strings para simplificar el código y evitar la iteración manual.

**Actividad 14: Generar un ID único - Crea una función que genere un ID único basado en un contador interno. (Tipo de función: IIFE con closure)**

Archivo esperado: ejercicios/ejercicio\_14.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código cumple con los requisitos utilizando una IIFE con closure para mantener el contador interno, generando IDs únicos correctamente. Bien hecho.

**Actividad 15: Invertir una cadena - Crea una función que invierta una cadena de texto. (Tipo de función: Función recursiva)**

Archivo esperado: ejercicios/ejercicio\_15.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

La solución invierte la cadena correctamente, pero no utiliza recursión como se solicitaba. Considera implementar la función de forma recursiva para cumplir con los requisitos.

**Actividad 16: Sumar argumentos variables - Crea una función que sume un número variable de argumentos. (Tipo de función: Función flecha con parámetro rest)**

Archivo esperado: ejercicios/ejercicio\_16.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. Utiliza correctamente el parámetro `rest` y `reduce` para sumar todos los argumentos. El código es conciso, legible y cumple con la especificación.

**Actividad 17: Ejecutar operación personalizada - Crea una función que reciba dos números y una función callback para realizar una operación. (Tipo de función: Función de orden superior)**

Archivo esperado: ejercicios/ejercicio\_17.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. El código es legible y cumple con la descripción de la actividad al utilizar una función de orden superior y un callback. Buen trabajo.

**Actividad 18: Validar correo electrónico - Crea una función que valide si una cadena es un correo electrónico básico (contiene @ y .com). (Tipo de función: Declaración de función)**

Archivo esperado: ejercicios/ejercicio\_18.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función cumple con los requisitos básicos, pero podría mejorarse la validación del dominio (no solo `.com`) y agregar una expresión regular para una validación más robusta. El código es legible y bien estructurado.

**Actividad 19: Retrasar ejecución de un mensaje - Crea una función que imprima un mensaje después de un tiempo definido por el usuario. (Tipo de función: Función anónima con setTimeout)**

Archivo esperado: ejercicios/ejercicio\_19.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. El código es limpio y cumple con el objetivo de la actividad utilizando setTimeout y una función anónima.

**Actividad 20: Calcular promedio de un arreglo - Crea una función que calcule el promedio de un arreglo de números. (Tipo de función: Función flecha)**

Archivo esperado: ejercicios/ejercicio\_20.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y utiliza una función flecha como se solicitó. Se recomienda retornar el promedio en lugar de imprimirlo directamente para mayor reutilización de la función. Considera agregar validación para arreglos vacíos.

## **Resumen General**

Excelente trabajo. Completó 20/20 actividades (100%) con una calificación promedio de 4.5/5. Demuestra buen dominio de los conceptos.

## **Recomendaciones**

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