

# Reporte de Evaluación - Fork de GitHub

## Información General

Estudiante: irwin Colmenaerez Castillo  
Repositorio: irwincol/act\_web1\_s4  
Fecha de evaluación: 21/8/2025, 16:40:58  
Evaluado por: Sistema de Evaluación

## Resumen de Calificaciones

Calificación general: 4.9/5.0  
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	5.0
2	Saludar con nombre por defecto - Crea un...	ejercicios/ejercicio_02.js	Sí	5.0
3	Sumar elementos de un arreglo - Crea una...	ejercicios/ejercicio_03.js	Sí	5.0
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	5.0
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	5.0
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	5.0
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	5.0
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	5.0

## 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: 5.0/5.0

Retroalimentación:

La solución es correcta, clara y concisa. La función calcula el cuadrado del número correctamente y se incluye un ejemplo de uso con `console.log`.

### 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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa, cumple con los requisitos de la actividad. Se utiliza la sintaxis de función flecha de forma adecuada y se maneja el valor por defecto correctamente.

### 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: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con los requisitos. El código es legible y la función flecha está implementada correctamente. Podrías considerar usar `reduce` para una solución más concisa.

### 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:

¡Excelente! La solución implementa correctamente una IIFE que imprime el mensaje deseado. 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 recursiva para contar vocales. El código es limpio, legible y cumple con los requisitos. Buen uso de la recursión y lógica condicional.

**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, utilizando `filter` de manera eficiente. El código es legible y cumple con el objetivo de la actividad.

**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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. El código es legible y utiliza la función flecha como se solicitó. ¡Buen trabajo!

**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 cumple con los requisitos de la actividad. El código es limpio y fácil de entender, utilizando correctamente setTimeout y la función anónima.

**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:

La solución es correcta y elegante, implementando una IIFE con closure para mantener el estado privado del contador. El código es limpio y funcional.

**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: 5.0/5.0

Retroalimentación:

La función factorial es correcta y eficiente. El código es limpio y fácil de entender, cumpliendo con la descripción de la actividad al usar recursión. Excelente trabajo.

**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 función debería retornar un booleano (true/false) en lugar de un string. Elimina los console.log de la función, estos deben estar fuera.

**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:

La solución es correcta y concisa, utilizando ``reduce`` de forma eficiente. El código es limpio y fácil de entender, cumpliendo con el objetivo de la actividad.

**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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa, utilizando una expresión regular para reemplazar todos los espacios. El código es limpio y funcional.

**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. Cumple con todos los requisitos de la actividad, utilizando una IIFE con closure de forma correcta para generar IDs únicos.

**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: 5.0/5.0

Retroalimentación:

Excelente solución recursiva. El código es conciso, claro y cumple con los requisitos de la actividad. La función ``invertir`` funciona correctamente.

**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. Cumple con todos los requisitos, usando correctamente la función flecha con parámetro rest y ``reduce`` para sumar los argumentos. El código es claro y conciso.

**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, clara y concisa. El código cumple con la descripción de la actividad y las buenas prácticas de programación. Bien hecho.

**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 la validación básica solicitada. Sin embargo, una validación de correo electrónico robusta debería considerar más patrones y caracteres permitidos para una mejor experiencia de usuario. Se recomienda usar expresiones regulares para validaciones más completas.

**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 cumple con todos los requisitos. El código es limpio y fácil de entender.

**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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es conciso, legible y utiliza correctamente la función flecha y `reduce` para calcular el promedio. La solución es completa y funcional.

## Resumen General

Excelente trabajo. Completó 20/20 actividades (100%) con una calificación promedio de 4.9/5. Demuestra buen dominio de los conceptos.

## Recomendaciones

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