

# Reporte de Evaluación - Fork de GitHub

## Información General

Estudiante: Víctor Alejandro Berrío Rivera  
Repositorio: vastrocode72/act\_web1\_s3  
Fecha de evaluación: 24/8/2025, 23:22:03  
Evaluado por: Sistema de Evaluación

## Resumen de Calificaciones

Calificación general: 4.0/5.0  
Actividades completadas: 20/20  
Porcentaje de completitud: 100.0%

## Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Crea un programa que use un ciclo while ...	ejercicios-while/ejercicio_while_01.js	Sí	1.0
2	Utiliza un ciclo while para calcular la ...	ejercicios-while/ejercicio_while_02.js	Sí	5.0
3	Implementa un sistema que use un ciclo w...	ejercicios-while/ejercicio_while_03.js	Sí	4.0
4	Crea un programa que use un ciclo while ...	ejercicios-while/ejercicio_while_04.js	Sí	3.0
5	Desarrolla una función que use un ciclo ...	ejercicios-while/ejercicio_while_05.js	Sí	4.0
6	Dado un array de nombres, usa un ciclo w...	ejercicios-while/ejercicio_while_06.js	Sí	4.0
7	Implementa un programa que use un ciclo ...	ejercicios-while/ejercicio_while_07.js	Sí	4.0
8	Utiliza un ciclo while para generar los ...	ejercicios-while/ejercicio_while_08.js	Sí	4.0
9	Crea un juego que use un ciclo while don...	ejercicios-while/ejercicio_while_09.js	Sí	4.0
10	Desarrolla un programa que use un ciclo ...	ejercicios-while/ejercicio_while_10.js	Sí	5.0
11	Usa un ciclo for anidado para crear las ...	ejercicios-for/ejercicio_for_01.js	Sí	5.0
12	Implementa un programa que use ciclos fo...	ejercicios-for/ejercicio_for_02.js	Sí	5.0
13	Dado un array de 20 números aleatorios, ...	ejercicios-for/ejercicio_for_03.js	Sí	5.0
14	Utiliza ciclos for anidados para encontr...	ejercicios-for/ejercicio_for_04.js	Sí	5.0
15	Crea un programa que use un ciclo for pa...	ejercicios-for/ejercicio_for_05.js	Sí	5.0
16	Implementa una función que use un ciclo ...	ejercicios-for/ejercicio_for_06.js	Sí	4.0
17	Usa ciclos for anidados para crear una m...	ejercicios-for/ejercicio_for_07.js	Sí	5.0
18	Desarrolla un programa que use un ciclo ...	ejercicios-for/ejercicio_for_08.js	Sí	2.0
19	Utiliza un ciclo for para generar la sig...	ejercicios-for/ejercicio_for_09.js	Sí	1.0
20	Implementa un programa que use un ciclo ...	ejercicios-for/ejercicio_for_10.js	Sí	5.0

## Retroalimentación Detallada

**Actividad 1: Crea un programa que use un ciclo while para mostrar una cuenta regresiva desde 20 hasta 1, mostrando cada número en la consola. Al final debe mostrar '¡Despegue!'.**

Archivo esperado: ejercicios-while/ejercicio\_while\_01.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código no resuelve el problema original (cuenta regresiva desde 20). Además, la variable 'numero' no está definida ni se incrementa correctamente, lo que provoca un bucle infinito. Revisa la lógica del ciclo while y asegúrate de cumplir con la descripción del problema.

**Actividad 2: Utiliza un ciclo while para calcular la suma de todos los números pares entre 1 y 50. Muestra el resultado final y cuántos números pares se sumaron.**

Archivo esperado: ejercicios-while/ejercicio\_while\_02.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y resuelve el problema correctamente. El uso de variables es adecuado y la lógica del ciclo while es impecable.

**Actividad 3: Implementa un sistema que use un ciclo while para solicitar una contraseña al usuario hasta que ingrese 'admin123'. Debe mostrar cuántos intentos fallidos hubo antes del acceso correcto.**

Archivo esperado: ejercicios-while/ejercicio\_while\_03.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

El código implementa la lógica de validación de contraseña con varios criterios. Sin embargo, la descripción inicial solicitaba validar hasta que se ingresara 'admin123', y en cambio se valida que cumpla varios criterios (longitud, mayúsculas, minúsculas, números). Considera refactorizar para alinearlo con la descripción original del ejercicio. El código es legible y bien estructurado.

**Actividad 4: Crea un programa que use un ciclo while para generar números aleatorios entre 1 y 100 hasta que salga un número mayor a 95. Muestra cada número generado y al final indica cuántos números se generaron.**

Archivo esperado: ejercicios-while/ejercicio\_while\_04.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

El código no cumple con la condición de detenerse al encontrar un número mayor a 95. El objetivo es encontrar un número mayor a 95, no el número 50. Considera corregir la condición del ciclo while.

**Actividad 5: Desarrolla una función que use un ciclo while para calcular el factorial de un número dado. Debe mostrar paso a paso cómo se va calculando el factorial.**

Archivo esperado: ejercicios-while/ejercicio\_while\_05.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Se podrían añadir mensajes paso a paso dentro del while para cumplir completamente con la descripción de mostrar el proceso de cálculo, aunque el código es limpio y bien estructurado.

**Actividad 6: Dado un array de nombres, usa un ciclo while para buscar un nombre específico. El programa debe mostrar en qué posición se encontró el nombre o indicar si no existe.**

Archivo esperado: ejercicios-while/ejercicio\_while\_06.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

El código resuelve correctamente el problema y es legible. Sería bueno evitar generar el número buscado aleatoriamente para poder probar el código de forma más predecible y asegurar que la búsqueda realmente funciona. También podrías considerar usar `indexOf` para simplificar la solución.

**Actividad 7: Implementa un programa que use un ciclo while para contar cuántos dígitos tiene un número entero positivo. Por ejemplo, 12345 tiene 5 dígitos.**

Archivo esperado: ejercicios-while/ejercicio\_while\_07.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funciona adecuadamente. El código es legible y tiene en cuenta el caso de entrada no válida y el número 0. Podrías mejorar la claridad inicial comentando el propósito del script antes de la solicitud del prompt.

**Actividad 8: Utiliza un ciclo while para generar los primeros 15 números de la secuencia de Fibonacci. Muestra cada número de la secuencia.**

Archivo esperado: ejercicios-while/ejercicio\_while\_08.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. El enunciado solicitaba 15 números de la secuencia, pero el código genera 20. Considera usar nombres de variables más descriptivos para mejorar la legibilidad.

**Actividad 9: Crea un juego que use un ciclo while donde el usuario debe adivinar un número entre 1 y 50. Proporciona pistas ('muy alto', 'muy bajo', 'cerca') y limita a 7 intentos máximo.**

Archivo esperado: ejercicios-while/ejercicio\_while\_09.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es funcional y bien estructurada. Podría mejorarse la validación del número secreto y la suposición para evitar el uso de `continue` en el ciclo while principal.

**Actividad 10: Desarrolla un programa que use un ciclo while para procesar calificaciones de estudiantes. Debe continuar pidiendo calificaciones hasta que se ingrese -1, luego calcular y mostrar el promedio, la calificación más alta y más baja.**

Archivo esperado: ejercicios-while/ejercicio\_while\_10.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Solución correcta y completa. El código es limpio, bien estructurado y aplica validaciones adecuadas. Buen manejo de los casos borde y el uso de buenas prácticas.

**Actividad 11: Usa un ciclo for anidado para crear las tablas de multiplicar del 1 al 10. Cada tabla debe mostrarse claramente separada y formateada.**

Archivo esperado: ejercicios-for/ejercicio\_for\_01.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio y fácil de entender, logrando el objetivo de la actividad de forma excelente.

**Actividad 12: Implementa un programa que use ciclos for anidados para crear un patrón de asteriscos en forma de diamante: \*** **\*\*** **\*\*\*** **\*\*\*\*** **\*\*\*\*\*** **\*\*\*\*\*** **\*\*\*\*** **\*\*\*** **\*\*** **\***

Archivo esperado: ejercicios-for/ejercicio\_for\_02.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución implementa correctamente el patrón de diamante usando ciclos for anidados. El código es limpio y fácil de entender. ¡Excelente trabajo!

**Actividad 13: Dado un array de 20 números aleatorios, usa un ciclo for para encontrar: el número mayor, el menor, la suma total, el promedio, y cuántos números son pares e impares.**

Archivo esperado: ejercicios-for/ejercicio\_for\_03.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio, legible y sigue buenas prácticas. ¡Excelente trabajo!

**Actividad 14: Utiliza ciclos for anidados para encontrar y mostrar todos los números primos entre 1 y 100. Debe indicar cuántos números primos se encontraron.**

Archivo esperado: ejercicios-for/ejercicio\_for\_04.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio, bien estructurado y fácil de entender. ¡Excelente trabajo!

**Actividad 15: Crea un programa que use un ciclo for para invertir una cadena de texto carácter por carácter. Muestra tanto la cadena original como la invertida.**

Archivo esperado: ejercicios-for/ejercicio\_for\_05.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio, fácil de entender y cumple con el objetivo de la actividad.

**Actividad 16: Implementa una función que use un ciclo for para calcular la potencia de un número sin usar Math.pow(). Debe mostrar el proceso paso a paso.**

Archivo esperado: ejercicios-for/ejercicio\_for\_06.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La función calcula correctamente la potencia, pero falta mostrar el proceso paso a paso como se solicitó. Considera agregar logs para visualizar cada iteración del ciclo for.

**Actividad 17: Usa ciclos for anidados para crear una matriz 5x5 donde cada elemento sea la suma de sus índices (fila + columna). Muestra la matriz formateada.**

Archivo esperado: ejercicios-for/ejercicio\_for\_07.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y cumple con los requisitos del problema. Buen uso de los ciclos for anidados y la lógica para generar la matriz.

**Actividad 18: Desarrolla un programa que use un ciclo for para contar cuántas vocales (a, e, i, o, u) hay en una frase dada. Debe mostrar el conteo de cada vocal por separado.**

Archivo esperado: ejercicios-for/ejercicio\_for\_08.js

Estado: Archivo encontrado

Calificación: 2.0/5.0

Retroalimentación:

El código intenta contar la frecuencia de cada carácter en lugar de solo las vocales como se pedía. Además, hay errores de sintaxis (como ``cadena.lenght`` en lugar de ``cadena.length`` y el uso de ``frecuenciaCaracteres`` antes de ser definido).

**Actividad 19: Utiliza un ciclo for para generar la siguiente secuencia: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. Debe mostrar cada término y la suma total de la secuencia.**

Archivo esperado: ejercicios-for/ejercicio\_for\_09.js

Estado: Archivo encontrado

Calificación: 1.0/5.0

Retroalimentación:

El código proporcionado no corresponde a la descripción del ejercicio. En lugar de generar la secuencia solicitada, simula el lanzamiento de dados. El código funciona, pero no resuelve el problema planteado.

**Actividad 20: Implementa un programa que use un ciclo for para verificar si una palabra o frase es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda). Ignora espacios y mayúsculas.**

Archivo esperado: ejercicios-for/ejercicio\_for\_10.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es limpio, eficiente y resuelve el problema correctamente. Se aplican buenas prácticas al limpiar la cadena y optimizar el ciclo for.

## Resumen General

Excelente trabajo. Completó 20/20 actividades (100%) con una calificación promedio de 4.0/5. Demuestra buen dominio de los conceptos.

## Recomendaciones

- Revisar y mejorar las actividades con calificación baja