

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Sergio Andres Montoya Monsalve
Repositorio: SerelbochaD3773/act_web1_s3
Fecha de evaluación: 21/8/2025, 15:20:18
Evaluado por: Sistema de Evaluación de No Calificados

Resumen de Calificaciones

Calificación general: 4.5/5.0
Actividades completadas: 19/20
Porcentaje de completitud: 95.0%

Detalle de Actividades

| # | Descripción | Archivo | Encontrado | Calificación |
|----|---|--|------------|--------------|
| 1 | Crea un programa que use un ciclo while ... | ejercicios-while/ejercicio_while_01.js | Sí | 5.0 |
| 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 | 5.0 |
| 5 | Desarrolla una función que use un ciclo ... | ejercicios-while/ejercicio_while_05.js | Sí | 5.0 |
| 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 | 5.0 |
| 8 | Utiliza un ciclo while para generar los ... | ejercicios-while/ejercicio_while_08.js | Sí | 5.0 |
| 9 | Crea un juego que use un ciclo while don... | ejercicios-while/ejercicio_while_09.js | Sí | 3.0 |
| 10 | Desarrolla un programa que use un ciclo ... | ejercicios-while/ejercicio_while_10.js | No | 0.0 |
| 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 | 4.0 |
| 19 | Utiliza un ciclo for para generar la sig... | ejercicios-for/ejercicio_for_09.js | Sí | 5.0 |
| 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: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio, fácil de entender y cumple con todos los requisitos de la actividad.

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:

La solución es correcta, eficiente y bien estructurada. El código es claro y fácil de entender, cumpliendo con todos los requisitos de la actividad.

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:

La solución es funcional, pero usa un array `intentos` predefinido en lugar de solicitar la contraseña al usuario en cada iteración del `while`. Para simular la interacción real, usa `prompt()` dentro del `while` y elimina el array `intentos`.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con todos los requisitos. El código es legible y utiliza buenas prácticas como la inicialización de variables y la claridad en la condición de parada del bucle.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y clara. El código está bien estructurado y cumple con los requisitos de la actividad, mostrando el cálculo paso a paso.

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:

La solución es correcta y funcional. Considera usar template literals para una mejor legibilidad al imprimir el resultado y evitar concatenaciones innecesarias. Podrías usar ``nombres.indexOf(nombreBuscado)`` para simplificar la búsqueda.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio y fácil de entender. Excelente uso del ciclo while y `Math.floor`.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio y fácil de entender, implementando la lógica de Fibonacci adecuadamente.

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: 3.0/5.0

Retroalimentación:

El código resuelve el problema, pero usa un array predefinido de respuestas en lugar de solicitar la entrada del usuario, lo cual limita la interactividad. Además, la estructura del código podría ser más modular.

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 no encontrado

Calificación: 0.0/5.0

Retroalimentación:

Error al evaluar: got status: 503 . {"error":{"code":503,"message":"The model is overloaded. Please try again later."},"status":"UNAVAILABLE"}}

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:

Excelente solución. El código es claro, funcional y cumple con todos los requisitos de la actividad. Bien hecho.

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:

Excelente solución. El código es claro, bien estructurado y produce el patrón de diamante correctamente. Se podría considerar una generalización a diferentes alturas.

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, clara y eficiente. El código está bien estructurado y cumple con todos los requisitos del problema.

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 para encontrar números primos. El código es limpio y fácil de entender, cumpliendo con las buenas prácticas y la completitud de la solución.

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:

Excelente solución. El código es claro, conciso y funcional. Cumple con los requisitos del problema de manera eficiente.

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 solución es funcional y muestra el proceso paso a paso, como se requería. Sin embargo, se espera una función para encapsular la lógica y hacer el código reutilizable, además de recibir los argumentos base y exponente por parámetros.

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:

La solución es correcta y eficiente. El código es claro y fácil de entender, aplicando buenas prácticas para la creación y visualización de 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: 4.0/5.0

Retroalimentación:

La solución es correcta y funciona como se espera. Sin embargo, la inicialización de `conteoVocales` es redundante y la actualización dentro del `switch` es innecesaria, ya que ya tienes los conteos individuales.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio y fácil de entender, cumpliendo con todos los requisitos del problema.

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:

La solución es correcta, clara y bien documentada. Utiliza buenas prácticas para limpiar la frase y la lógica del palíndromo es impecable. Excelente trabajo.

Resumen General

Excelente trabajo. Completó 19/20 actividades (95%) con una calificación promedio de 4.5/5. Demuestra buen dominio de los conceptos.

Recomendaciones

- Completar los archivos faltantes: ejercicios-while/ejercicio_while_10.js