

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Estudiante desconocido
Repositorio: Diegoarango20/act_web1_s3
Fecha de evaluación: 21/8/2025, 14:46:51
Evaluado por: Sistema de Evaluación de No Calificados

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	3.0
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	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	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	3.0
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	5.0
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	3.0
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:

La solución no cumple con la descripción del problema (cuenta regresiva de 20 a 1). El código actual cuenta del 1 al 50 e imprime los números divisibles por 3 y 5. Revisa cuidadosamente el enunciado del ejercicio y adáptalo.

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 cumple con todos los requisitos del ejercicio, incluyendo la correcta suma de los números pares y el conteo. Solo considera que la descripción solicitaba el rango de 1 a 50, pero el código lo hizo de 1 a 100, esto no afecta la calidad del código.

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: 3.0/5.0

Retroalimentación:

El código proporcionado no coincide con la descripción de la actividad. Valida una contraseña compleja, no la contraseña 'admin123'. No cuenta el número de intentos fallidos como se indica en la descripción.

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 funciona, pero no cumple con la descripción original de la actividad (número mayor a 95). El comentario inicial tampoco coincide con el código. Considera revisar y adaptar el código a la descripción dada.

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 función calcula correctamente el factorial. Sería bueno agregar validación para la entrada del usuario (asegurarse de que sea un número positivo) y mostrar el proceso paso a paso en la consola tal como se solicita en la descripción del problema.

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, utilizando un ciclo while para la búsqueda. Se puede mejorar la legibilidad del código separando la lógica de generación del array y la búsqueda en funciones separadas.

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, completa y bien estructurada. El código es legible y considera todos los casos, incluyendo números negativos y el cero.

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 genera la secuencia de Fibonacci solicitada. El código es limpio y fácil de entender, utilizando buenas prácticas de programación.

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 cumple con los requisitos. Podrías mejorar la experiencia del usuario incluyendo la pista 'cerca' y validando la entrada del usuario para asegurar que sea un número válido.

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:

La solución es correcta, completa y bien estructurada. El código es limpio y fácil de entender, y maneja correctamente las validaciones y casos borde.

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, utilizando ciclos for anidados para generar las tablas de multiplicar del 1 al 12. El código es legible y bien estructurado.

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: 3.0/5.0

Retroalimentación:

El código genera una pirámide, no un diamante como se solicitaba. La lógica para imprimir la parte inferior del diamante (decreciente) está ausente.

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:

Excelente solución. El código es limpio, bien estructurado y cumple con todos los requisitos de la actividad. Se aplican buenas prácticas y la lógica es correcta.

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 claro y utiliza `Math.sqrt()` para optimizar la búsqueda de divisores, una buena práctica.

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, clara y eficiente. El código invierte la cadena usando un bucle for de manera adecuada y presenta los resultados solicitados.

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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, maneja los casos especiales correctamente (exponente negativo, cero, base cero) y muestra el proceso paso a paso como se solicitó.

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 resuelve correctamente el problema planteado. Buena práctica al usar `join`` para formatear la salida.

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: 3.0/5.0

Retroalimentación:

El código no resuelve el problema planteado (conteo de vocales). En cambio, cuenta la frecuencia de cada carácter en la cadena. Debes enfocar la lógica en contar específicamente las vocales (a, e, i, o, u).

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 presentado no resuelve el problema planteado en la descripción de la actividad. El código actual simula el lanzamiento de dados en lugar de generar la secuencia numérica solicitada.

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, bien estructurado y resuelve correctamente el problema del palíndromo, incluyendo la eliminación de caracteres no alfanuméricos y conversión a minúsculas.

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