

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Estudiante desconocido
Repositorio: Diegoarango20/act_web1_s3
Fecha de evaluación: 21/8/2025, 14:13:45
Evaluado por: Sistema de Evaluación Masiva

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:

El código no cumple con la descripción de la actividad. El ciclo while cuenta del 1 al 50 y muestra los números divisibles por 3 y 5 en lugar de una cuenta regresiva de 20 a 1 y el mensaje '¡Despegue!'.

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 y eficiente, cumpliendo con los requisitos del problema. El código es limpio y fácil de entender, utilizando nombres de variables descriptivos.

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:

La solución implementa un validador de contraseñas con criterios de seguridad, en lugar de simplemente verificar 'admin123' como lo pedía la descripción original. El uso de expresiones regulares es correcto, pero la descripción no se cumple completamente.

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 condición solicitada (número > 95). Se debe modificar la condición del while para que coincida con la descripción del problema. Además, el mensaje final debe indicar que salió un número mayor a 95, no 50.

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. Podrías mejorar la presentación del proceso paso a paso dentro del ciclo while para cumplir completamente con la descripción de la actividad. Además, sería bueno agregar validación para el ingreso del número.

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 buscar un número en el array. Sería recomendable validar si el valor ingresado por el usuario es realmente un número antes de realizar 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 legible y cubre los casos borde (números negativos y 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 eficiente. El código es claro, legible y cumple con el objetivo de generar la secuencia de Fibonacci utilizando un ciclo while. Se podría mejorar la documentación del código (comentarios).

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:

El código funciona correctamente y cumple con los requisitos del problema. Se podría mejorar la experiencia de usuario indicando qué tan cerca está el número ingresado al número secreto (ej: 'cerca', 'muy lejos').

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 y completa, calcula el promedio, la nota más alta y la más baja correctamente. El código es limpio y fácil de entender, con validación de entrada.

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 resuelve el problema de las tablas de multiplicar del 1 al 12, cumpliendo con la descripción. 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: 3.0/5.0

Retroalimentación:

El código genera una pirámide, no un diamante como se solicitaba. Además, la descripción original pedía un diamante, pero el comentario en el código indica una pirámide de 10 niveles, generando confusión.

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, fácil de entender 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, eficiente y bien estructurada. El código es limpio y fácil de entender, utilizando la optimización de `Math.sqrt(num)` para mejorar el rendimiento.

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, eficiente y fácil de entender. El código es limpio 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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, completo y considera casos especiales como exponentes negativos y cero. La explicación paso a paso es muy útil.

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, bien estructurado y cumple con el objetivo planteado utilizando ciclos for anidados para generar la matriz y mostrarla formateada.

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 cumple con la descripción del problema (contar vocales). Realiza un conteo general de caracteres. Debes enfocarte en 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 proporcionado no resuelve la actividad descrita. Implementa un simulador de dados en lugar de generar la secuencia solicitada (2, 4, 8, ... , 1024).

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 y eficiente. El código está bien estructurado y utiliza buenas prácticas como la limpieza del texto antes de la validación.

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