

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Albany Anmabel Luciani Mujica
Repositorio: albanyluciani/act_web1_s3
Fecha de evaluación: 21/8/2025, 15:11:33
Evaluado por: Sistema de Evaluación de No Calificados

Resumen de Calificaciones

Calificación general: 3.6/5.0
Actividades completadas: 16/20
Porcentaje de completitud: 80.0%

Detalle de Actividades

| # | Descripción | Archivo | Encontrado | Calificación |
|----|---|--|------------|--------------|
| 1 | Crea un programa que use un ciclo while ... | ejercicios-while/ejercicio_while_01.js | No | 0.0 |
| 2 | Utiliza un ciclo while para calcular la ... | ejercicios-while/ejercicio_while_02.js | No | 0.0 |
| 3 | Implementa un sistema que use un ciclo w... | ejercicios-while/ejercicio_while_03.js | No | 0.0 |
| 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 | 4.0 |
| 6 | Dado un array de nombres, usa un ciclo w... | ejercicios-while/ejercicio_while_06.js | No | 0.0 |
| 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 | 2.0 |
| 10 | Desarrolla un programa que use un ciclo ... | ejercicios-while/ejercicio_while_10.js | Sí | 4.0 |
| 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 | 5.0 |
| 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 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 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 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 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 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 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 eficiente. El código es limpio, fácil de entender y cumple con los requisitos del problema. Excelente trabajo.

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. Sin embargo, la descripción pedía mostrar el cálculo paso a paso, lo cual no se implementó en el console.log. Se podría mejorar mostrando el valor del resultado y contador en cada iteración.

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 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 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 claro, bien estructurado y considera el caso especial donde el número es 0.

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, conciso y fácil de entender. Buen uso del ciclo while para generar la secuencia de Fibonacci.

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: 2.0/5.0

Retroalimentación:

El código tiene problemas importantes. No solicita input al usuario, y el uso de 'break' dentro del ciclo impide que el juego funcione correctamente, limitando la ejecución a un solo intento. Se necesita implementar la entrada del usuario y remover el 'break' innecesario para que el juego pueda progresar a través de los intentos permitidos.

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: 4.0/5.0

Retroalimentación:

La solución funciona correctamente y calcula el promedio, máximo y mínimo. Sin embargo, utiliza un arreglo hardcoded en lugar de solicitar las calificaciones interactivamente (usando `prompt`), lo cual limita su aplicabilidad real.

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 completa, utilizando ciclos for anidados para generar las tablas de multiplicar del 1 al 10. El código es limpio y legible, cumpliendo con los requisitos de la actividad.

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:

La lógica del diamante es correcta, pero la variable `simbolo` nunca se utiliza y en su lugar se imprime `línea` que no está definida. Corrige la variable a imprimir y considera usar template literals para construir la cadena de asteriscos.

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 legible y cumple con todos los requisitos de la actividad. Buen 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:

Excelente solución. El código es claro, conciso y eficiente para encontrar números primos. La lógica es correcta y se cumplen todos los requisitos de la actividad.

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 legible y cumple con el objetivo de invertir la cadena usando un ciclo for.

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, funcional y demuestra el proceso paso a paso como se solicitó. Bien hecho!

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:

Solución correcta y eficiente. El código es claro, legible y cumple con los requisitos del ejercicio. Buen uso de los ciclos for anidados.

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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es claro, conciso y aborda correctamente el conteo de vocales, incluyendo el manejo de acentos. Bien hecho.

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, clara y eficiente. Utiliza buenas prácticas para generar la secuencia y calcular la suma, cumpliendo con todos los requisitos de la actividad.

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 es limpio, legible y cumple con la descripción del problema. Se aplican buenas prácticas como la conversión a minúsculas y la eliminación de espacios.

Resumen General

Buen trabajo general. Completó 16/20 actividades (80%) con una calificación promedio de 3.6/5. Hay oportunidades de mejora en algunos aspectos.

Recomendaciones

- Completar los archivos faltantes: ejercicios-while/ejercicio_while_01.js, ejercicios-while/ejercicio_while_02.js, ejercicios-while/ejercicio_while_03.js, ejercicios-while/ejercicio_while_06.js
- Revisar y mejorar las actividades con calificación baja