

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Mario Alonso Munera Velez
Repositorio: MarioMunera1993/act_web1_s2
Fecha de evaluación: 21/8/2025, 17:59:46
Evaluado por: Sistema de Evaluación

Resumen de Calificaciones

Calificación general: 3.8/5.0
Actividades completadas: 10/10
Porcentaje de completitud: 100.0%

Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Par o Impar - Declara la variable numero...	src/ejercicio1.js	Sí	4.0
2	Mayor de dos números - Declara a = 14 y ...	src/ejercicio2.js	Sí	3.0
3	Clasificación de edad - Declara edad = 1...	src/ejercicio3.js	Sí	5.0
4	Calculadora básica con switch - Declara ...	src/ejercicio4.js	Sí	3.0
5	Día laborable o fin de semana - Declara ...	src/ejercicio5.js	Sí	4.0
6	Nota a letra - Declara nota = 83. Convie...	src/ejercicio6.js	Sí	4.0
7	Descuento por volumen - Declara precioUn...	src/ejercicio7.js	Sí	3.0
8	Transporte con tarifa reducida - Declara...	src/ejercicio8.js	Sí	4.0
9	Conversor de notas musicales (switch) - ...	src/ejercicio9.js	Sí	4.0
10	Autenticación simple - Declara usuario =...	src/ejercicio10.js	Sí	4.0

Retroalimentación Detallada

Actividad 1: Par o Impar - Declara la variable numero = 7. Imprime 'par' o 'impar' según corresponda.

Archivo esperado: src/ejercicio1.js
Estado: Archivo encontrado
Calificación: 4.0/5.0
Retroalimentación:
La solución funciona correctamente para el número asignado. Sin embargo, faltó la declaración de la variable 'numero' con 'let' o 'const'. Considera usar const si el valor no debe cambiar.

Actividad 2: Mayor de dos números - Declara a = 14 y b = 9. Imprime el mayor de ambos.

Archivo esperado: src/ejercicio2.js
Estado: Archivo encontrado
Calificación: 3.0/5.0
Retroalimentación:
El código funciona correctamente para determinar el mayor de dos números. Sin embargo, no usa los valores a=14 y b=9 especificados en la descripción del problema, y usa valores hardcoded. Se recomienda usar los valores indicados en el enunciado y usar variables con nombres más descriptivos.

Actividad 3: Clasificación de edad - Declara edad = 15. Imprime 'niño' si es 0-12, 'adolescente' si es 13-17, 'adulto' si es 18-64, 'anciano' si es 65 o más.

Archivo esperado: src/ejercicio3.js

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con los requisitos. El código es legible y bien estructurado. Podrías mejorar la reutilización del código utilizando una estructura switch o una función.

Actividad 4: Calculadora básica con switch - Declara a = 10, b = 3, operador = '*'. Imprime el resultado o 'operador no válido'. Si b === 0 y operador === '/' imprime 'indeterminado'.

Archivo esperado: src/ejercicio4.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

La lógica del switch funciona, pero los valores iniciales de las variables (numero1, numero2) no coinciden con los requerimientos de la actividad (a=10, b=3). Además, el 'operador no válido' no se imprime como se pide en la descripción.

Actividad 5: Día laborable o fin de semana - Declara día = 'sabado'. Imprime 'laborable' para lunes-viernes y 'fin de semana' para sábado o domingo.

Archivo esperado: src/ejercicio5.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional, utilizando un `switch` para determinar si es día laborable o fin de semana. Se podría mejorar declarando la variable `dia` con `let` o `const` para mayor claridad y evitar el ámbito global implícito.

Actividad 6: Nota a letra - Declara nota = 83. Convierte la nota numérica ('B', 70-79 ! 'C', 60-69 ! 'D', < 60 ! 'F'.

Archivo esperado: src/ejercicio6.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional, pero podría ser más concisa utilizando un `switch` o una función para evitar la repetición de `console.log`. Considera usar `else if` encadenados para mayor legibilidad.

Actividad 7: Descuento por volumen - Declara precioUnitario = 25, cantidad = 60. Aplica descuento: 0% si cantidad < 10, 10% si 10 <= cantidad < 50, 20% si cantidad >= 50. Imprime p decimales.

Archivo esperado: src/ejercicio7.js

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

La lógica del descuento es correcta, pero los valores iniciales de `precioUnitario` y `cantidad` no coinciden con los indicados en la descripción del ejercicio. Considera usar `toFixed(2)` para redondear el precio final a dos decimales.

Actividad 8: Transporte con tarifa reducida - Declara edad = 67, distanciaKm = 120, tarifaBase = distanciaKm * 0.21. Aplica 20% descuento si edad < 18, 40% descuento si edad > 65, redondeado a 2 decimales.

Archivo esperado: src/ejercicio8.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La lógica es correcta y funcional, pero falta redondear el precio final a dos decimales como se pide en la descripción. Además, el código podría mejorarse usando `else` para cubrir el caso en que no aplica ningún descuento.

Actividad 9: Conversor de notas musicales (switch) - Declara nota = 'LA'. RE!' 62, MI!' 64, FA!' 65, SOL!' 67, LA!' 69, SI!' 71. Cualquier otro valor!' 70.

Archivo esperado: src/ejercicio9.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es funcional y resuelve el problema planteado. Considerar usar 'let' o 'const' para declarar la variable 'nota' en lugar de una asignación global implícita y ser consistente con el uso de mayúsculas/minúsculas (ej: 'SI' vs 'Si').

Actividad 10: Autenticación simple - Declara usuario = 'admin', contraseña = '1234'. Imprime 'acceso concedido' si coincide exactamente, 'acceso denegado' en cualquier otro caso.

Archivo esperado: src/ejercicio10.js

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La lógica es correcta pero la contraseña definida en el código no coincide con la descripción del problema. Revisar la asignación de valores a las variables para que coincidan con la descripción.

Resumen General

Buen trabajo general. Completó 10/10 actividades (100%) con una calificación promedio de 3.8/5. Hay oportunidades de mejora en algunos aspectos.

Recomendaciones

- Continuar con el excelente trabajo y mantener la calidad del código