Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Juan Miguel Santamaría Múnera
Repositorio: PeperoniSword/act_ntp_s3
Fecha de evaluación: 22/8/2025, 18:15:53
Evaluado por: Sistema de Evaluación

Resumen de Calificaciones

Calificación general: 3.9/5.0
Actividades completadas: 19/20
Porcentaje de completitud: 95.0%

Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Usando un ciclo for, imprime los números...	src/ejercicio_01.py	Sí	3.0
2	Mediante un ciclo while, imprime los núm...	src/ejercicio_02.py	Sí	5.0
3	Con un ciclo for, calcula la suma de tod...	src/ejercicio_03.py	Sí	5.0
4	Utilizando un ciclo while, solicita al u...	src/ejercicio_04.py	Sí	3.0
5	Con un ciclo for, imprime la tabla de mu...	src/ejercicio_05.py	Sí	4.0
6	Mediante un ciclo while, genera y muestr...	src/ejercicio_06.py	Sí	5.0
7	Con un ciclo for, cuenta cuántas letras ...	src/ejercicio_07.py	Sí	5.0
8	Usando un ciclo while, calcula y muestra...	src/ejercicio_08.py	Sí	5.0
9	Con un ciclo for, imprime todos los núme...	src/ejercicio_09.py	Sí	4.0
10	Mediante un ciclo while, solicita al usu...	src/ejercicio_10.py	Sí	4.0
11	Con un ciclo for, imprime cada carácter ...	src/ejercicio_11.py	Sí	4.0
12	Utilizando un ciclo while, calcula el fa...	src/ejercicio_12.py	Sí	0.0
13	Con un ciclo for, imprime los números de...	src/ejercicio_13.py	Sí	5.0
14	Mediante un ciclo while, implementa un j...	src/ejercicio_14.py	Sí	4.0
15	Con un ciclo for, imprime un triángulo r...	src/ejercicio_15.py	Sí	4.0
16	Utilizando un ciclo while, simula un rel...	src/ejercicio_16.py	Sí	3.0
17	Con un ciclo for, solicita al usuario qu...	src/ejercicio_17.py	Sí	4.0
18	Mediante un ciclo while, genera y muestr...	src/ejercicio_18.py	Sí	3.0
19	Con un ciclo for, cuenta cuántas vocales...	src/ejercicio_19.py	Sí	4.0
20	Utilizando un ciclo while, solicita al u...	src/ejercicio_20.py	Sí	4.0

Retroalimentación Detallada

Actividad 1: Usando un ciclo for, imprime los números enteros del 0 al 9, cada uno en una línea.

Archivo esperado: src/ejercicio_01.py

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

La solución usa una comprensión de lista para imprimir, lo cual no es incorrecto, pero un ciclo `for` explícito sería más legible. Además, el rango empieza en 1, debería empezar en 0 para incluir el 0 en la salida, y el 10 es exclusivo, el 9 sí se imprime.

Actividad 2: Mediante un ciclo while, imprime los números enteros del 10 al 1 en orden descendente, cada número en una línea.

Archivo esperado: src/ejercicio_02.py

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es limpio y fácil de entender.

Actividad 3: Con un ciclo for, calcula la suma de todos los enteros del 1 al 100 (inclusive) y muestra el resultado.

Archivo esperado: src/ejercicio_03.py

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. El código es limpio y fácil de entender. ¡Excelente trabajo!

Actividad 4: Utilizando un ciclo while, solicita al usuario que ingrese números. El proceso termina cuando el usuario escriba 0. Al final, muestra la suma total de todos los números ingresados.

Archivo esperado: src/ejercicio_04.py

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

El programa funciona, pero no calcula la suma total de los números ingresados. Falta acumular los números ingresados en una variable para luego mostrar el resultado.

Actividad 5: Con un ciclo for, imprime la tabla de multiplicar del 7, es decir, 7 × 1, 7 × 2, …, 7 × 10, cada resultado en una línea.

Archivo esperado: src/ejercicio_05.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y concisa utilizando una list comprehension. Sin embargo, el uso de list comprehension para este caso específico (solo imprimir) se considera menos legible que un bucle for tradicional. Aunque funciona, la claridad podría mejorarse.

Actividad 6: Mediante un ciclo while, genera y muestra los primeros 15 múltiplos de 3, comenzando desde 3.

Archivo esperado: src/ejercicio_06.py

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con los requerimientos. El código es limpio y fácil de entender.

Actividad 7: Con un ciclo for, cuenta cuántas letras 'a' (minúscula) hay en la cadena texto = "manzana" y muestra el total.

Archivo esperado: src/ejercicio_07.py

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución. El código es limpio, funcional y sigue buenas prácticas al utilizar una función para generalizar el conteo de caracteres.

Actividad 8: Usando un ciclo while, calcula y muestra los cuadrados de los números del 1 al 20 (1^2, 2^2, …, 20^2), cada resultado en una línea.

Archivo esperado: src/ejercicio_08.py

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 el objetivo planteado.

Actividad 9: Con un ciclo for, imprime todos los números pares del 2 al 50 (ambos inclusive), cada número en una línea.

Archivo esperado: src/ejercicio_09.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y concisa utilizando una comprensión de lista. Sin embargo, para un problema tan simple, un bucle `for` explícito sería más legible. Además, se podría optimizar el `range` comenzando en 2 y usando un paso de 2.

Actividad 10: Mediante un ciclo while, solicita al usuario que escriba palabras. El proceso termina cuando el usuario escriba la palabra "fin". Al final, muestra cuántas palabras se leyeron (sin contar "fin").

Archivo esperado: src/ejercicio_10.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La lógica es correcta y cumple con el objetivo. Se podría mejorar la legibilidad usando un contador dentro del bucle while en lugar de un diccionario si solo se necesita la cantidad de palabras. Considera usar nombres de variables más descriptivos (por ejemplo, `palabra_ingresada` en lugar de `palabra`).

Actividad 11: Con un ciclo for, imprime cada carácter de la palabra "python" en una línea separada.

Archivo esperado: src/ejercicio_11.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y concisa utilizando una comprensión de lista. Sin embargo, un ciclo for tradicional sería más legible en este caso particular para principiantes. Considera la legibilidad sobre la brevedad.

Actividad 12: Utilizando un ciclo while, calcula el factorial de un número entero n introducido por el usuario y muestra el resultado.

Archivo esperado: src/ejercicio_12.py

Estado: Archivo encontrado

Calificación: 0.0/5.0

Retroalimentación:

Contenido insuficiente - solo comentarios o código trivial

Actividad 13: Con un ciclo for, imprime los números del 1 al 30 saltando de 3 en 3 (1, 4, 7, …, 28), cada número en una línea.

Archivo esperado: src/ejercicio_13.py

Estado: Archivo encontrado

Calificación: 5.0/5.0

Retroalimentación:

Excelente solución utilizando comprensión de listas para lograr el resultado de manera concisa. El código es funcional y cumple con los requisitos de la actividad.

Actividad 14: Mediante un ciclo while, implementa un juego de adivinanza: el programa genera un número aleatorio del 1 al 10 y solicita al usuario que lo adivine. El proceso se repite hasta que el usuario acierte. Muestra un mensaje de felicitación al final.

Archivo esperado: src/ejercicio_14.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional, pero el número aleatorio se genera en cada iteración del ciclo, lo que no es lo ideal. Debería generarse una sola vez al principio. Considera agregar mensajes de ayuda al usuario si falla.

Actividad 15: Con un ciclo for, imprime un triángulo rectángulo de 5 filas usando el carácter '*'.

Archivo esperado: src/ejercicio_15.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es concisa y correcta usando una comprensión de lista. Podría ser más legible con un ciclo for tradicional para principiantes. Considerar agregar comentarios para mejor claridad.

Actividad 16: Utilizando un ciclo while, simula un reloj digital que muestre cada segundo desde 00:00 hasta 00:59 en formato MM:SS, cada valor en una línea.

Archivo esperado: src/ejercicio_16.py

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

El código funciona parcialmente, pero no se reinicia el contador de segundos a 0 después de 59. Además, el formato MM:SS no siempre se cumple (falta el 0 inicial para segundos menores a 10). Considera usar f-strings para formatear la salida correctamente y la lógica del ciclo while necesita revisión.

Actividad 17: Con un ciclo for, solicita al usuario que ingrese un número entero positivo y calcula la suma de sus dígitos, mostrando el resultado final.

Archivo esperado: src/ejercicio_17.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución funciona correctamente y es concisa. Podrías agregar validación para asegurar que la entrada sea un número entero positivo y mejorar los nombres de las variables para mayor claridad.

Actividad 18: Mediante un ciclo while, genera y muestra la secuencia de Fibonacci empezando por 1, 1, 2, 3, 5, … y termina cuando se alcance el primer valor mayor que 1000.

Archivo esperado: src/ejercicio_18.py

Estado: Archivo encontrado

Calificación: 3.0/5.0

Retroalimentación:

El código genera la secuencia de Fibonacci, pero imprime valores incorrectamente y no se detiene justo después del primer valor mayor que 1000. Deberías calcular el siguiente valor antes de imprimirlo y verificar la condición de parada después del cálculo.

Actividad 19: Con un ciclo for, cuenta cuántas vocales (sin distinción de mayúsculas/minúsculas) hay en la frase frase = "programacion es divertida" y muestra el total.

Archivo esperado: src/ejercicio_19.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Podrías mejorar la legibilidad utilizando `lower()` para hacer la comparación insensible a mayúsculas y minúsculas, evitando así duplicar las vocales en la lista.

Actividad 20: Utilizando un ciclo while, solicita al usuario que ingrese edades una a una. El proceso termina cuando se introduzca -1. Al final, muestra la edad mayor que se haya ingresado.

Archivo esperado: src/ejercicio_20.py

Estado: Archivo encontrado

Calificación: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Podría mejorarse el manejo de excepciones para ser más específico (ValueError en lugar de Exception) y evitar capturar errores no relacionados con la conversión a entero. El código es legible y la lógica implementada es clara.

Resumen General

Buen trabajo general. Completó 19/20 actividades (95%) con una calificación promedio de 3.9/5. Hay oportunidades de mejora en algunos aspectos.

Recomendaciones

- Revisar y mejorar las actividades con calificación baja