

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Juan Manuel Rivera Restrepo

Repositorio: JuanRivera24/act_ntp_s3

Fecha de evaluación: 22/8/2025, 20:30:29

Evaluated por: Sistema de Evaluación

Resumen de Calificaciones

Calificación general: 4.1/5.0

Actividades completadas: 20/20

Porcentaje de completitud: 100.0%

Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Usando un ciclo for, imprime los números...	src/ejercicio_01.py	Sí	5.0
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	5.0
5	Con un ciclo for, imprime la tabla de mu...	src/ejercicio_05.py	Sí	5.0
6	Mediante un ciclo while, genera y muestr...	src/ejercicio_06.py	Sí	3.0
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	5.0
10	Mediante un ciclo while, solicita al usu...	src/ejercicio_10.py	Sí	3.0
11	Con un ciclo for, imprime cada carácter ...	src/ejercicio_11.py	Sí	5.0
12	Utilizando un ciclo while, calcula el fa...	src/ejercicio_12.py	Sí	5.0
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	3.0
15	Con un ciclo for, imprime un triángulo r...	src/ejercicio_15.py	Sí	2.0
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	1.0
20	Utilizando un ciclo while, solicita al u...	src/ejercicio_20.py	Sí	5.0

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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. Cumple con todos los requisitos de la actividad.

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, concisa y cumple con los requisitos de la actividad. 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 eficiente. El código es claro, conciso y cumple con los requisitos del problema.

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: 5.0/5.0

Retroalimentación:

Excelente solución. El código es conciso, legible y cumple con todos los requisitos de la actividad. Bien hecho.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y concisa. El código es limpio y cumple con el objetivo del ejercicio.

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: 3.0/5.0

Retroalimentación:

El código genera múltiplos de 3, pero el bucle termina cuando i es igual a 15 en lugar de generar los primeros 15 múltiplos. El código necesita iterar más veces o tener una condición de parada diferente.

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:

La solución es correcta y eficiente. El código es limpio, fácil de entender y cumple con el objetivo planteado.

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, concisa y eficiente. El código es limpio y fácil de entender, cumpliendo con todos los requisitos de la actividad.

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: 5.0/5.0

Retroalimentación:

La solución es correcta, eficiente y concisa. Utiliza el ciclo for con un paso adecuado para imprimir los números pares en el rango especificado.

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: 3.0/5.0

Retroalimentación:

El código funciona, pero no cuenta las palabras correctamente y también imprime la palabra 'fin'. Deberías usar un contador y evitar imprimir 'fin'.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y cumple con los requisitos. El código es conciso y legible.

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: 5.0/5.0

Retroalimentación:

La solución es correcta, eficiente y legible. Utiliza un ciclo `while` de forma apropiada para calcular el factorial.

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:

La solución es correcta y eficiente. El código es limpio y fácil de entender, cumpliendo con la descripción 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: 3.0/5.0

Retroalimentación:

La lógica del ciclo while no es la correcta. Se genera un nuevo número aleatorio en cada iteración, en lugar de compararlo con la entrada del usuario. Deberías generar el número aleatorio una sola vez fuera del ciclo y mantenerlo constante.

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: 2.0/5.0

Retroalimentación:

El código imprime un rectángulo, no un triángulo rectángulo. La cantidad de asteriscos en cada fila debería incrementarse con el número de fila.

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 muestra un reloj, pero no se limita al rango 00:00-00:59 como se solicitó en la descripción. Además, el ciclo `while True` hace que el programa se ejecute indefinidamente sin cumplir con el objetivo especificado.

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 es correcta y funcional, calcula la suma de los dígitos de un número entero positivo. Sin embargo, la actividad solicitaba el uso de un ciclo `for`, mientras que el código utiliza un ciclo `while`. Considera usar un ciclo `for` como se especificó en la descripción.

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 parece intentar generar la secuencia de Fibonacci, pero falta inicializar la variable 'anterior'. Además, la lógica para calcular el siguiente número de Fibonacci es incorrecta, y no se inicializa el primer valor de la secuencia (1). La condición `numero < 1001` es correcta para detener la secuencia.

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: 1.0/5.0

Retroalimentación:

El código no resuelve el problema planteado. Está contando letras mayúsculas en lugar de vocales. Debes revisar la lógica de la condición dentro del bucle for.

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: 5.0/5.0

Retroalimentación:

Excelente solución. El código resuelve correctamente el problema, es legible y utiliza buenas prácticas como la validación de lista vacía antes de buscar el máximo.

Resumen General

Excelente trabajo. Completó 20/20 actividades (100%) con una calificación promedio de 4.1/5. Demuestra buen dominio de los conceptos.

Recomendaciones

- Revisar y mejorar las actividades con calificación baja