

Reporte de Evaluación - Fork de GitHub

Información General

Estudiante: Yeferson Santamaria Holguin
Repositorio: yeferlol12/act_ntp_s3
Fecha de evaluación: 31/7/2025, 23:48:05
Evaluado por: Sistema de Evaluación Masiva

Resumen de Calificaciones

Calificación general: 2.5/5.0
Actividades completadas: 11/20
Porcentaje de completitud: 55.0%

Detalle de Actividades

#	Descripción	Archivo	Encontrado	Calificación
1	Usando un ciclo for, imprime los números...	src/ejercicio_01.py	Sí	4.0
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	4.0
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	4.0
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	3.0
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	5.0
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	0.0
13	Con un ciclo for, imprime los números de...	src/ejercicio_13.py	Sí	0.0
14	Mediante un ciclo while, implementa un j...	src/ejercicio_14.py	Sí	0.0
15	Con un ciclo for, imprime un triángulo r...	src/ejercicio_15.py	Sí	0.0
16	Utilizando un ciclo while, simula un rel...	src/ejercicio_16.py	Sí	0.0
17	Con un ciclo for, solicita al usuario qu...	src/ejercicio_17.py	Sí	0.0
18	Mediante un ciclo while, genera y muestr...	src/ejercicio_18.py	Sí	0.0
19	Con un ciclo for, cuenta cuántas vocales...	src/ejercicio_19.py	Sí	0.0
20	Utilizando un ciclo while, solicita al u...	src/ejercicio_20.py	Sí	0.0

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: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Se podría mejorar la legibilidad omitiendo el prefijo 'n =' en la salida, ya que solo se pedía imprimir el número.

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, clara y concisa. El código funciona como se espera y es 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, clara y concisa. El código es funcional y cumple con el objetivo de la actividad.

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: 4.0/5.0

Retroalimentación:

La solución es correcta y funcional. Se podría mejorar la validación de la entrada para evitar errores si el usuario ingresa texto en lugar de números, y la indentación.

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, clara y concisa. El código es funcional y aplica buenas prácticas al usar f-strings para formatear la salida.

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: 4.0/5.0

Retroalimentación:

La solución resuelve el problema correctamente, pero el límite del ciclo (contador=48) podría ser más explícito ($n \leq 45$) o calculado a partir del número de múltiplos deseados (15). Considera usar un comentario para explicar el propósito del valor 48.

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 para el problema planteado. El código es legible y fácil de entender. Podrías considerar usar nombres de variables más descriptivos para mejorar aún más la claridad.

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: 3.0/5.0

Retroalimentación:

El código itera correctamente del 1 al 20, pero calcula incorrectamente el cuadrado (multiplica por 2 en lugar de elevar al cuadrado). Deberías multiplicar el número por sí mismo para obtener el cuadrado.

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 y eficiente. El código es legible y cumple con el objetivo planteado de imprimir los números pares del 2 al 50.

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: 5.0/5.0

Retroalimentación:

La solución es correcta y eficiente. El código es legible y cumple con el objetivo planteado de manera clara y concisa.

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 concisa. El código es legible y cumple con los requisitos de la actividad.

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:

El código presentado no calcula el factorial de un número, ni utiliza un ciclo while. El código itera sobre un string e imprime cada carácter. Se debe implementar la lógica del factorial con un ciclo while.

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: 0.0/5.0

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: 0.0/5.0

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: 0.0/5.0

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: 0.0/5.0

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: 0.0/5.0

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: 0.0/5.0

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: 0.0/5.0

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: 0.0/5.0

Resumen General

Necesita mejorar. Completó 11/20 actividades (55%) con una calificación promedio de 2.5/5. Se recomienda revisar los conceptos fundamentales.

Recomendaciones

- Revisar y mejorar las actividades con calificación baja
- Enfocarse en mejorar la documentación y comentarios del código
- Aplicar mejores prácticas de programación