

| Tabla de comparación   |  |   |
|--|--|---|
| Positivo   | Negativo   | Interesante   |
| <p><b>Código ia:</b> El código es simple y directo, lo que facilita su comprensión.</p> <p><b>Código sin ia:</b> El código es modular y reutilizable gracias al uso de una función separada para calcular la suma.</p> | <p><b>Código ia:</b> No lleva indicación</p> <p><b>Código sin ia:</b> El código no identifica, los números que no son naturales</p>  | <p><b>Código ia:</b> El código proporciona un mensaje de error si el número ingresado no es un número natural.</p> <p><b>Código sin ia:</b> El código utiliza un bucle for para calcular la suma de los números naturales desde 1 hasta n.</p>    |
| <p><b>Código ia:</b> No se desperdician recursos en variables adicionales ni en condicionales innecesarios.</p> <p><b>Código sin ia:</b> Concisos y realiza la tarea de manera efectiva.</p>                           | <p><b>Código ia:</b> Comienza desde 99 y también va hacia abajo hasta 1. Aunque ambas formas son válidas, comenzar desde 99 y reducir de 2 en 2 es más eficiente en términos de rendimiento.</p> <p><b>Código sin ia:</b> Utiliza una función imprimir() para imprimir los números impares, lo que podría considerarse innecesario para una tarea tan simple</p>   | <p><b>Código ia:</b> El código salta de 2 en 2 comenzando desde 99.</p> <p><b>Código sin ia:</b> El primer código itera sobre todos los números entre 1 y 100, Esto muestra cómo hay múltiples formas de abordar un problema en programación.</p> |
| <p><b>Código ia y Código sin ia:</b> Ambos códigos utilizan un bucle for para iterar sobre los números del 1 al 100 y calcular el cuadrado de cada número.</p>   | <p><b>Código ia:</b> Comienza desde <math>i = 1</math>, lo que es más preciso en este sentido.</p> <p><b>Código sin ia:</b> Comienza desde <math>i = 0</math> en lugar de <math>i = 1</math>, lo que significa que también incluye el cuadrado de 0 en la suma. Esto no es incorrecto, pero podría considerarse innecesario ya que el problema pide la suma de los cuadrados de los números entre 1 y 100.</p> | <p><b>Código ia y Código sin ia:</b> Ambos códigos son efectivos y dan el mismo resultado, pero el código hecho por ia es más claro y sigue las convenciones estándar de codificación de C++.</p>   |
| <p><b>Código ia y Código sin ia:</b> Ambos códigos son claros y directos en su implementación. Utilizan un bucle for para solicitar las notas de los estudiantes y</p>   | <p><b>Código ia y Código sin ia:</b> La única diferencia entre los dos códigos es el nombre de la variable numestu en el primer código y num_estudiantes en el</p>   | <p><b>Código ia y Código sin ia:</b> Ambos códigos pueden ser mejorados incluyendo la validación de las notas ingresadas para asegurarse de que estén dentro de un</p>  |

|  |   |   |
|--|---|---|
| <p>calcular la suma total de estas notas.<br/>Las variables y los nombres de las variables son descriptivos y fáciles de entender en ambos códigos.</p>  | <p>segundo. Esto no afecta la funcionalidad del programa, pero puede causar confusión si alguien más lee o mantiene el código. Sería preferible mantener la consistencia en la nomenclatura de las variables.</p>   | <p>rango válido. Esto evitaría que el usuario ingrese notas inválidas y mejoraría la robustez del programa.</p>   |
| <p><b>Código ia:</b> Utiliza bucles para iterar sobre las notas de los estudiantes y calcular la cantidad de aprobados y reprobados.<br/><b>Código sin ia:</b> Utiliza una función para agregar las notas de los estudiantes y luego procesarlas. Esta división en funciones mejora la modularidad y facilita la comprensión del código.</p> | <p><b>Código ia:</b> No incluye la funcionalidad de agregar las notas por separado, lo que podría ser útil si el usuario desea ingresar las notas en momentos diferentes<br/><b>Código sin ia:</b> El manejo de índices en la función agregarNotas podría causar un problema si el usuario intenta agregar más notas después de que todas las notas hayan sido ingresadas. Podría ser preferible manejar este caso de manera más explícita.</p> | <p><b>Código ia:</b> Simplifica el proceso al calcular directamente la cantidad de aprobados y reprobados dentro del bucle de ingreso de notas, lo que elimina la necesidad de un bucle adicional.<br/><b>Código sin ia:</b> Una interfaz más interactiva al permitir al usuario seleccionar qué acción realizar a través de un menú de opciones.</p> |
| <p><b>Código ia y Código sin ia:</b> Ambos códigos son efectivos para realizar la tarea requerida y dan el resultado correcto. Utilizan un bucle for para iterar sobre los números entre 100 y 200 e identificar los números pares.</p>  | <p><b>Código ia y Código sin ia:</b> Mi código utiliza una función sumarPares() para realizar la suma de los números pares, lo que podría considerarse innecesario para una tarea tan simple. El código generado por la IA lo hace directamente en la función main(), lo que simplifica el programa.</p>  | <p><b>Código ia y Código sin ia:</b> Ambos códigos tienen una estructura similar y siguen un enfoque común para resolver el problema. No hay diferencias significativas en términos de eficiencia o rendimiento entre los dos códigos.</p>  |
| <p><b>Código ia:</b> Utiliza la biblioteca vector de C ++, lo que permite trabajar con vectores de cualquier longitud.<br/><b>Código sin ia :</b> Utiliza un menú para permitir al usuario elegir entre ingresar números a los vectores y sumar los vectores</p>   | <p><b>Código ia:</b> El usuario debe proporcionar la longitud de los vectores antes de ingresar los elementos.<br/><b>Código sin ia:</b> Los vectores tienen una longitud fija definida por la constante MAX. Esto limita la flexibilidad del programa.</p>   | <p><b>Código ia:</b> El código es simple y directo, lo que facilita su comprensión.<br/><b>Código sin ia:</b> El uso de funciones separadas para ingresar los números y sumar los vectores hace que el código sea modular y reutilizable.</p>   |
| <p><b>Código ia:</b> Utiliza la</p>  | <p><b>Código ia:</b> El usuario debe</p>  | <p><b>Código ia:</b> El código es</p>   |

|   |  |  |
|---|--|--|
| <p>biblioteca vector de C++, lo que permite trabajar con vectores de cualquier longitud.</p> <p><b>Código sin ia:</b><br/>Utiliza un menú para permitir al usuario elegir entre ingresar números a los vectores y calcular el producto punto.</p>                                       | <p>proporcionar la longitud de los vectores antes de ingresar los elementos.</p> <p><b>Código sin ia:</b> Los vectores tienen una longitud fija definida por la constante MAX. Esto limita la flexibilidad del programa.</p>   | <p>simple y directo, lo que facilita su comprensión.</p> <p><b>Código sin ia:</b><br/>El uso de funciones separadas para ingresar los números y calcular el producto punto hace que el código sea modular y reutilizable.</p>  |
| <p><b>Código ia:</b> Utiliza la biblioteca vector de C++, lo que permite trabajar con matrices de cualquier dimensión.</p> <p><b>Código sin ia:</b> Utiliza un menú para permitir al usuario elegir entre ingresar números a las matrices y calcular el producto de las matrices.</p>   | <p><b>Código ia:</b> El usuario debe proporcionar las dimensiones de las matrices antes de ingresar los elementos.</p> <p><b>Código sin ia:</b> Las matrices tienen dimensiones fijas definidas por la constante MAX. Esto limita la flexibilidad del programa</p>                 | <p><b>Código ia:</b> El código es simple y directo, lo que facilita su comprensión.</p> <p><b>Código sin ia:</b> El uso de funciones separadas para ingresar los números y calcular el producto de las matrices hace que el código sea modular y reutilizable.</p>   |
| <p><b>Código ia:</b> Utiliza la biblioteca vector de C++, lo que permite trabajar con matrices de cualquier dimensión.</p> <p><b>Código sin ia:</b> Utiliza un menú para permitir al usuario elegir entre ingresar números a las matrices y calcular la transposición de la matriz.</p> | <p><b>Código ia:</b> El usuario debe proporcionar las dimensiones de las matrices antes de ingresar los elementos.</p> <p><b>Código sin ia:</b> Las matrices tienen dimensiones fijas definidas por las constantes FILAS y COLUMNAS. Esto limita la flexibilidad del programa.</p> | <p><b>Código ia:</b> El código es simple y directo, lo que facilita su comprensión.</p> <p><b>Código sin ia:</b> El uso de funciones separadas para ingresar los números y calcular la transposición de la matriz hace que el código sea modular y reutilizable.</p> |