**Análisis de Complejidad**

Integrantes:

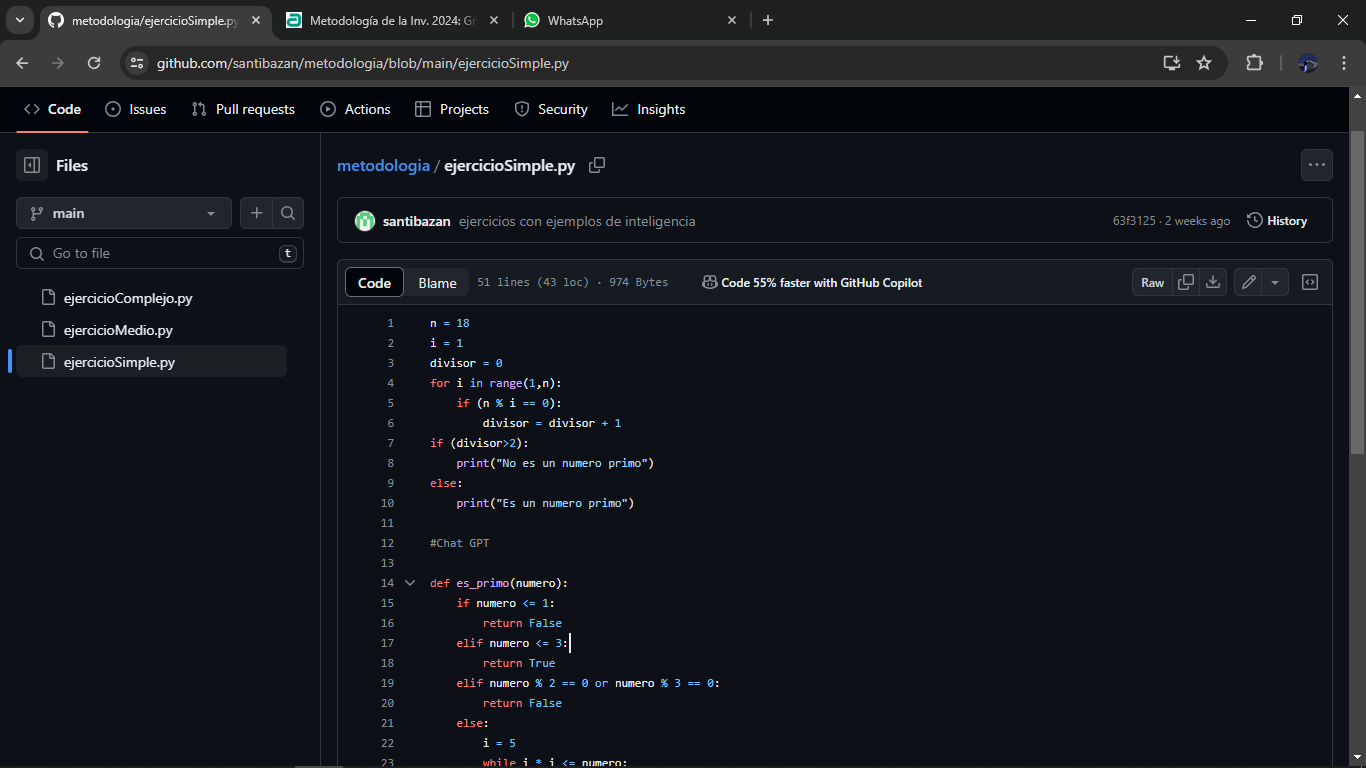
* Jorge García Ojeda
* Matías Games
* Santiago Bazán
* Matías Rodríguez

**Ejercicio 1: Determinar si un número es primo**

**Código 1 (Hecho por nosotros):**

* Complejidad: O(n)

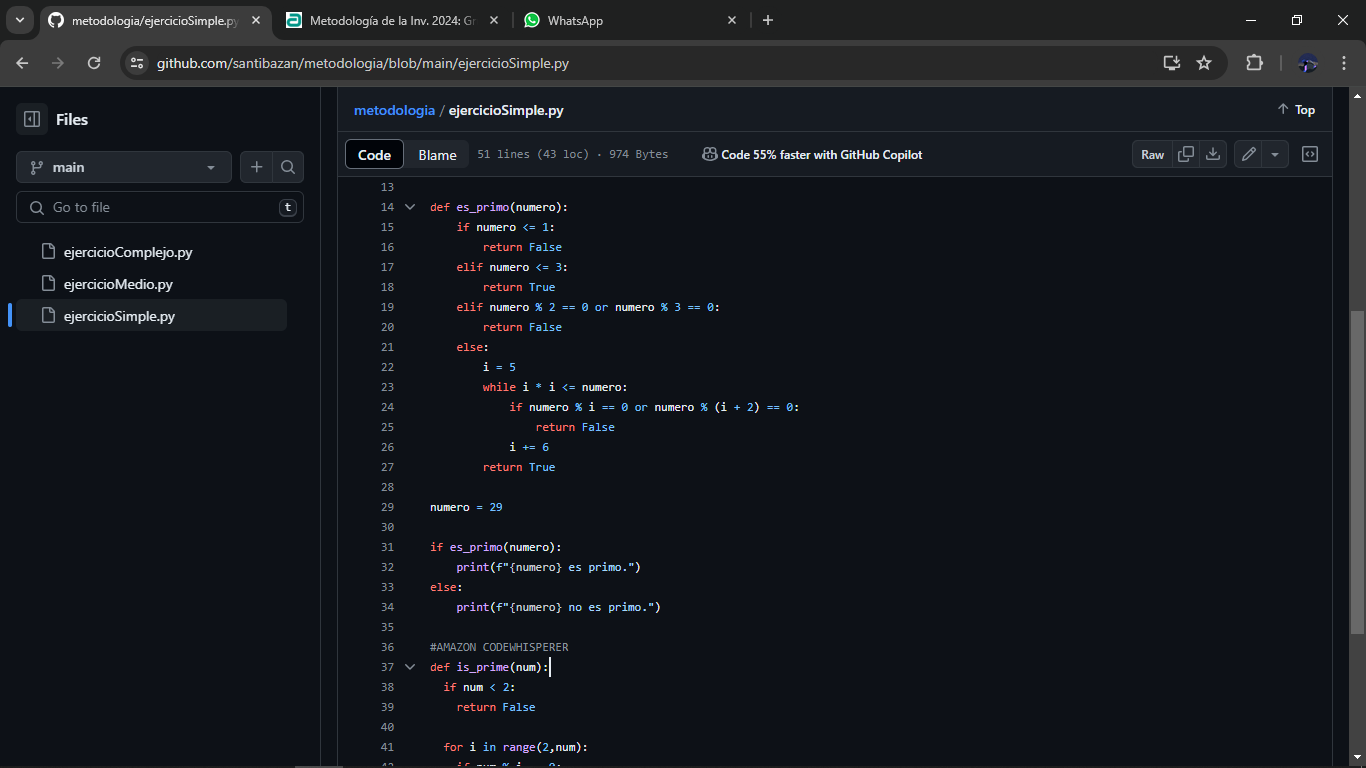
El bucle “for” recorre todos los números desde 1 hasta n-1, lo que da como resultado una complejidad lineal, ya que el tiempo de ejecución aumenta linealmente con el tamaño de la entrada n.



**Código 2 (Chat GPT):**

* Complejidad: O(sqrt(n))

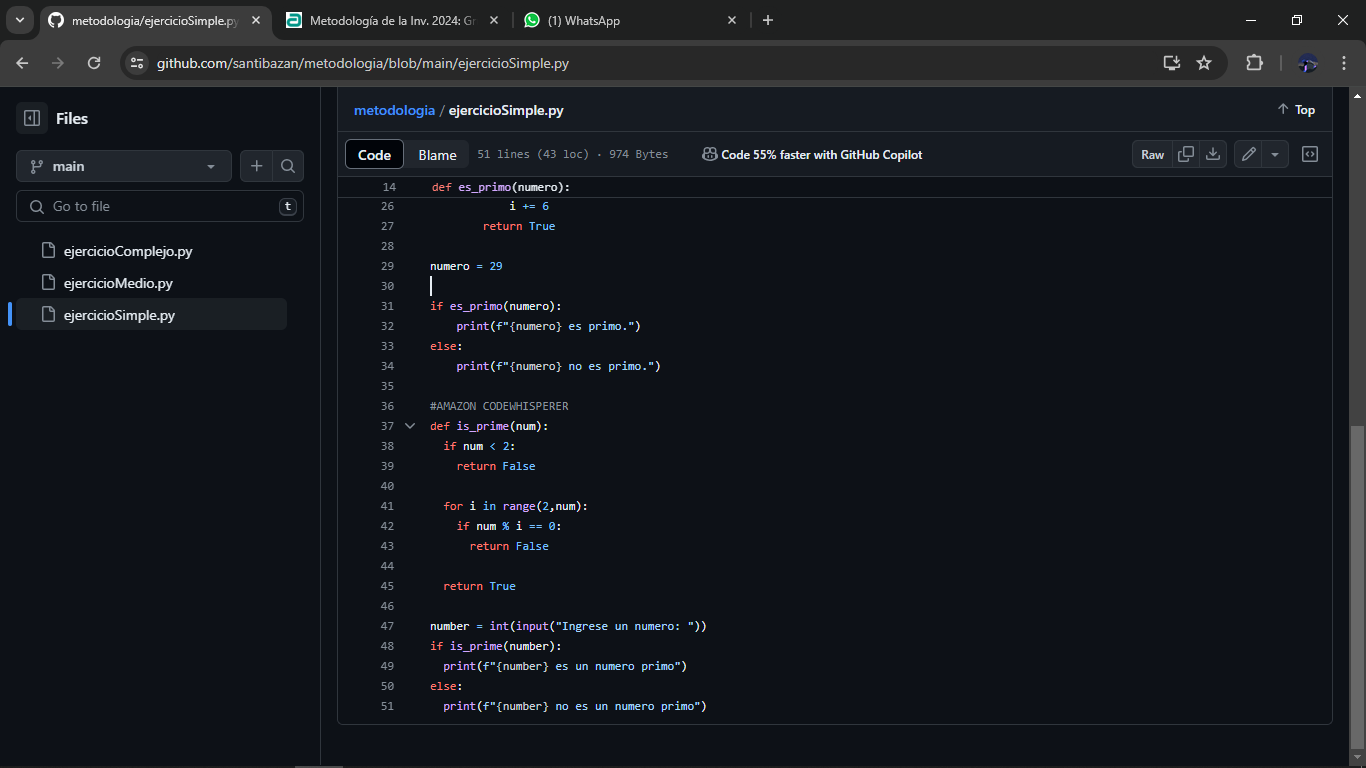
Este algoritmo utiliza el método de prueba de primalidad de la raíz cuadrada. Realiza la división hasta la raíz cuadrada de número. Dado que la raíz cuadrada de número es una constante en términos de la entrada número, la complejidad es O(sqrt(n)).



**Código 3 (Amazon CodeWhisperer):**

* Complejidad: O(n)

Este algoritmo también utiliza un bucle “for” que recorre todos los números desde 2 hasta num-1. Como resultado, la complejidad es lineal, O(n).



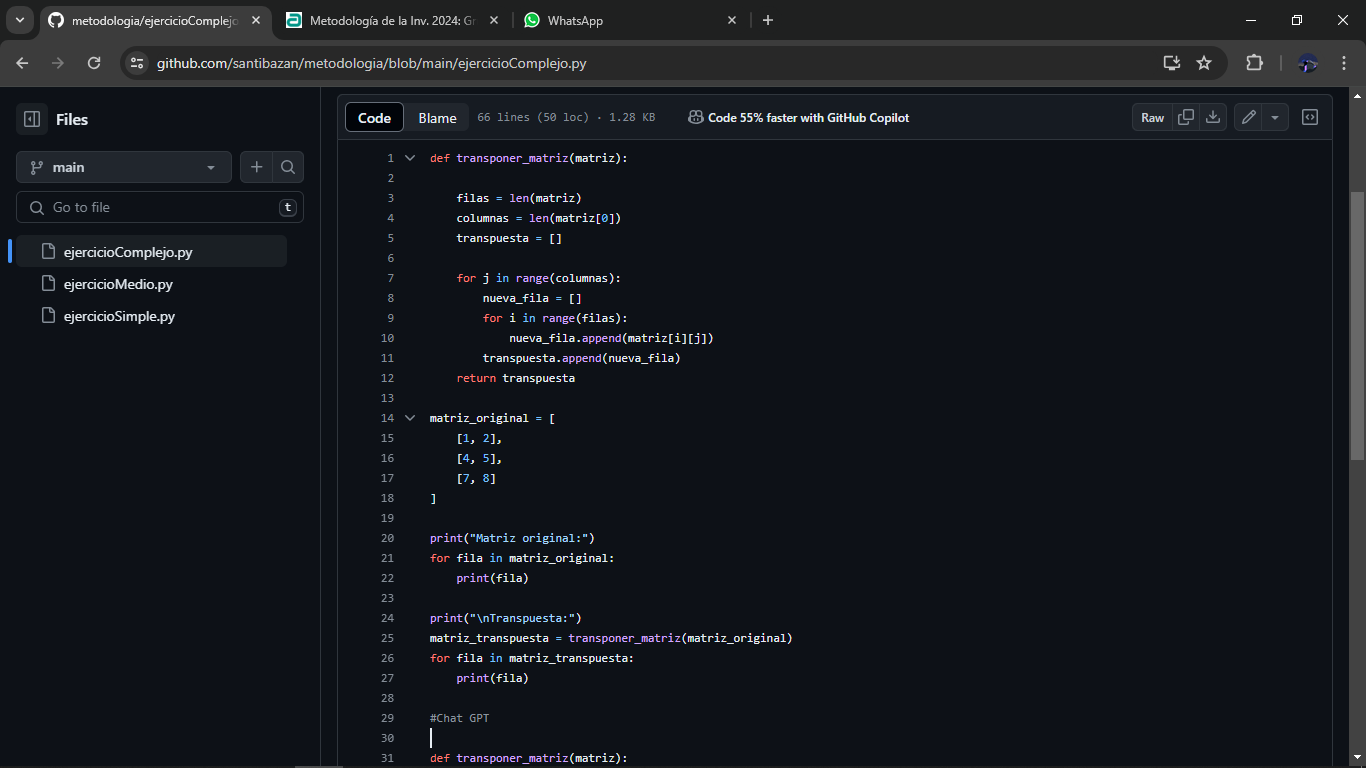
En resumen, el algoritmo más eficiente en términos de tiempo de ejecución es el código proporcionado por el modelo de chat GPT, ya que tiene una complejidad de O(sqrt(n)), mientras que los otros dos tienen una complejidad lineal O(n).

**Ejercicio 2: Devolver la transpuesta de una matriz**

**Código 1 (Hecho por nosotros):**

* Complejidad: O(n\*m)

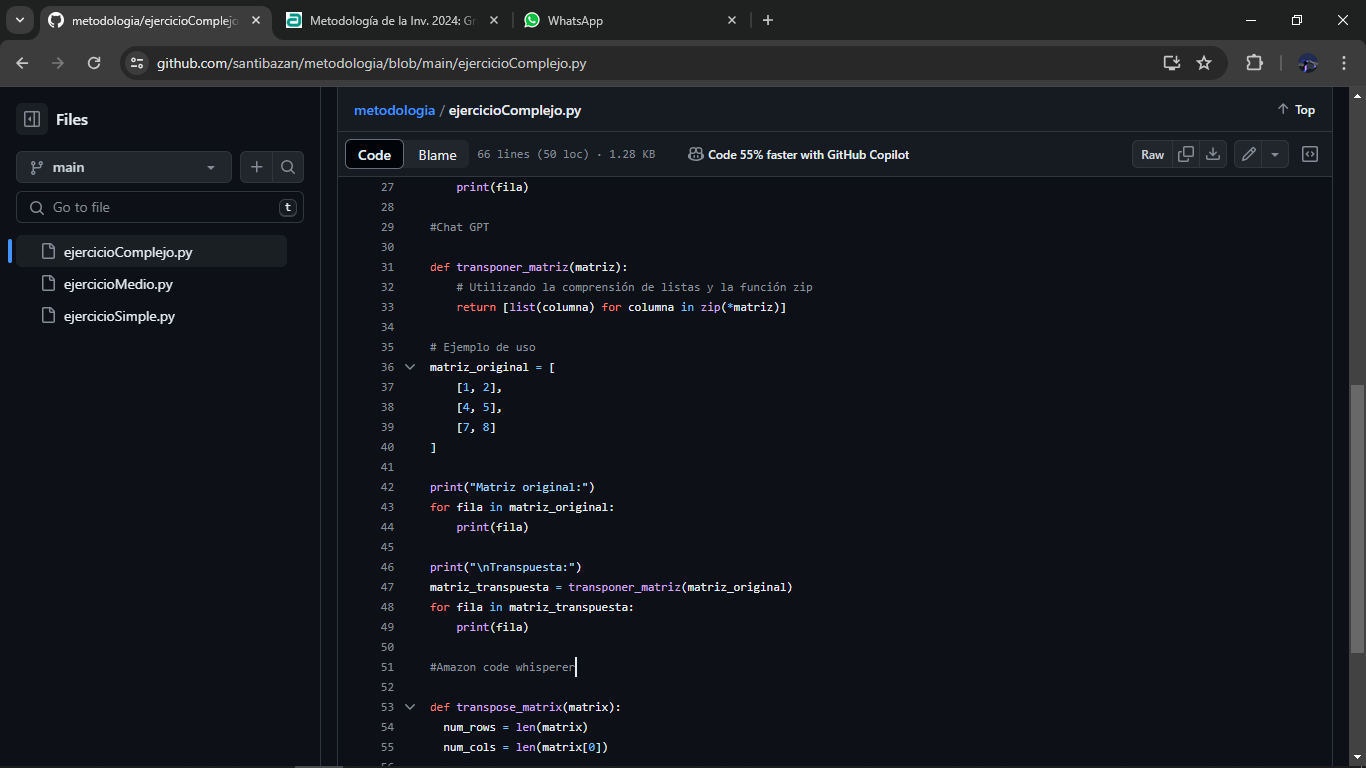
El algoritmo recorre cada elemento de la matriz una vez para construir la matriz transpuesta. La matriz original tiene n filas y m columnas. Por lo tanto, la complejidad total es proporcional al producto de n y m, es decir, O(n\*m).



**Código 2 (Chat GPT):**

* Complejidad: O(n\*m)

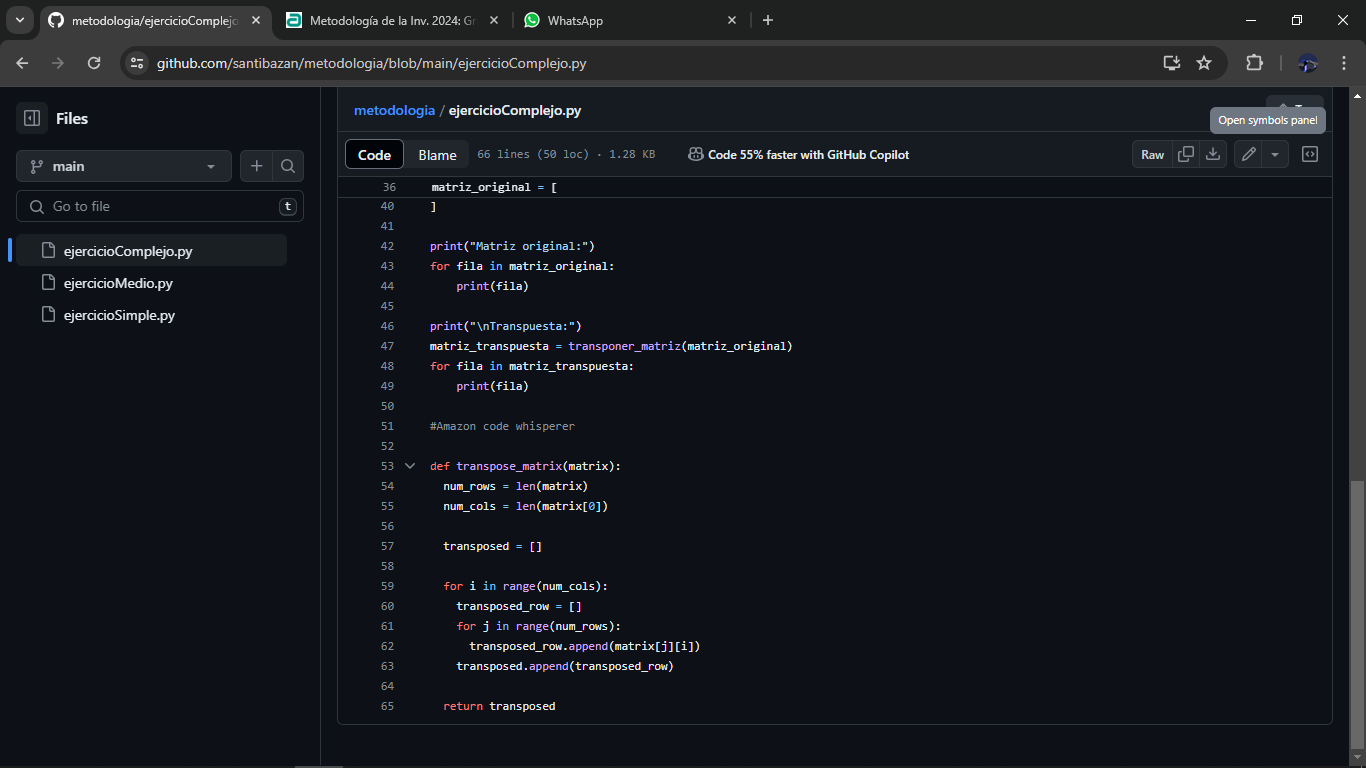
El algoritmo utiliza comprensión de listas y la función zip para transponer la matriz. Aunque la sintaxis es más concisa, internamente, el algoritmo aún recorre cada elemento de la matriz una vez. Por lo tanto, la complejidad es la misma que en el código 1, O(n\*m).



**Código 3 (Amazon CodeWhisperer):**

* Complejidad: O(n\*m)

Este algoritmo también recorre cada elemento de la matriz una vez para construir la matriz transpuesta. Al igual que los otros dos algoritmos, su complejidad es proporcional al producto de n y m, es decir, O(n\*m).



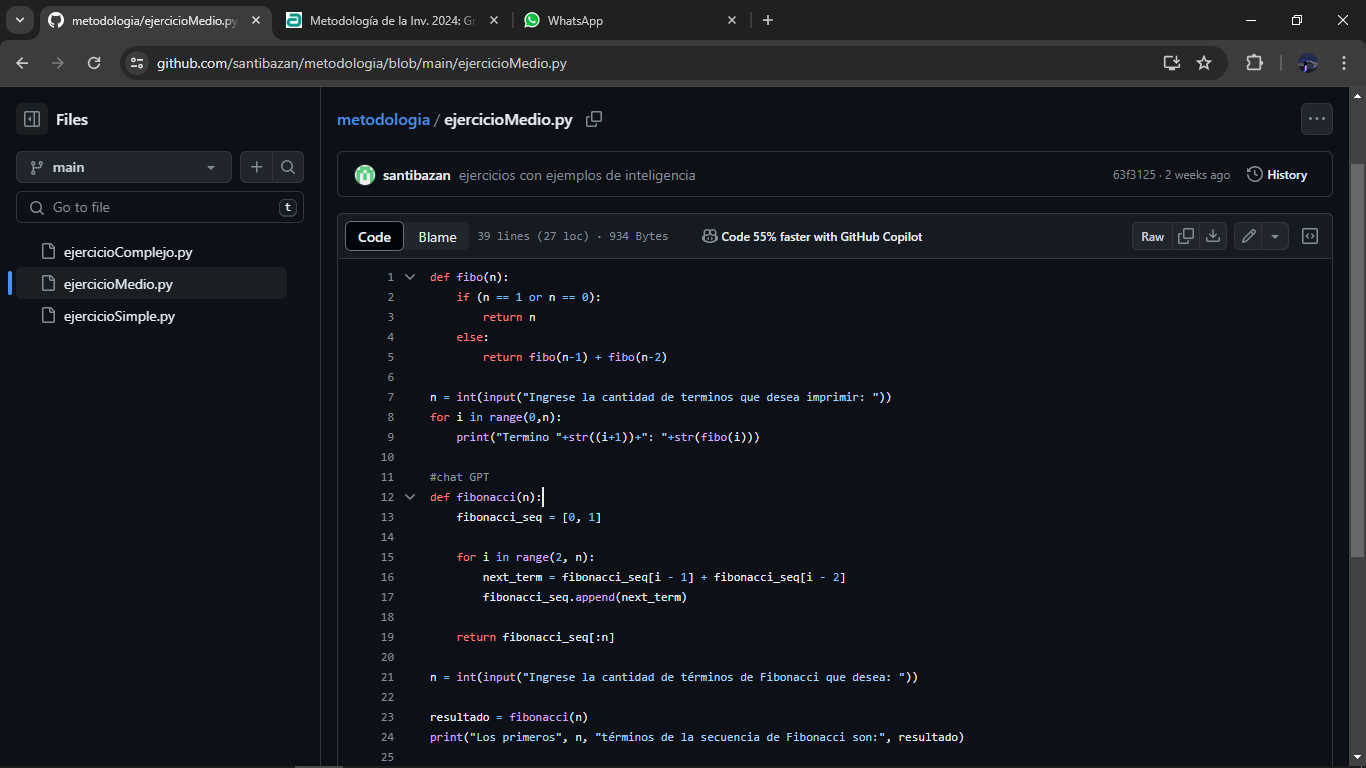
En resumen, los tres algoritmos tienen la misma complejidad temporal, ya que todos requieren recorrer cada elemento de la matriz una vez para transponerla.

**Ejercicio 3: Devolver los primeros n números de la secuencia de Fibonacci**

**Código 1 (Hecho por nosotros):**

* Complejidad: O(2^n)

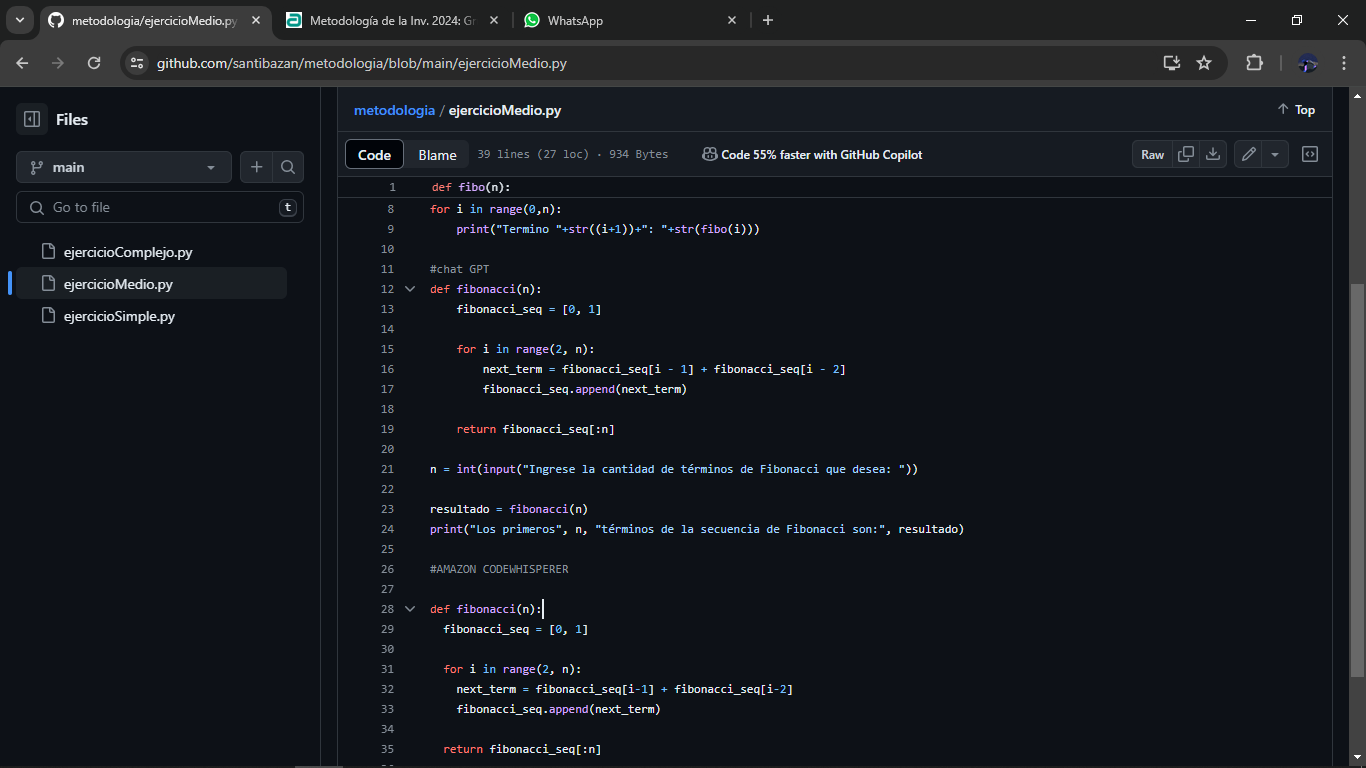
Este algoritmo utiliza recursión para calcular los términos de la secuencia de Fibonacci. En cada llamada recursiva, se realizan dos llamadas adicionales (una para “fibo(n-1)” y otra para “fibo(n-2)”). Esto resulta en una complejidad exponencial, ya que el número de llamadas recursivas crece exponencialmente con n.



**Código 2 (Chat GPT):**

* Complejidad: O(n)

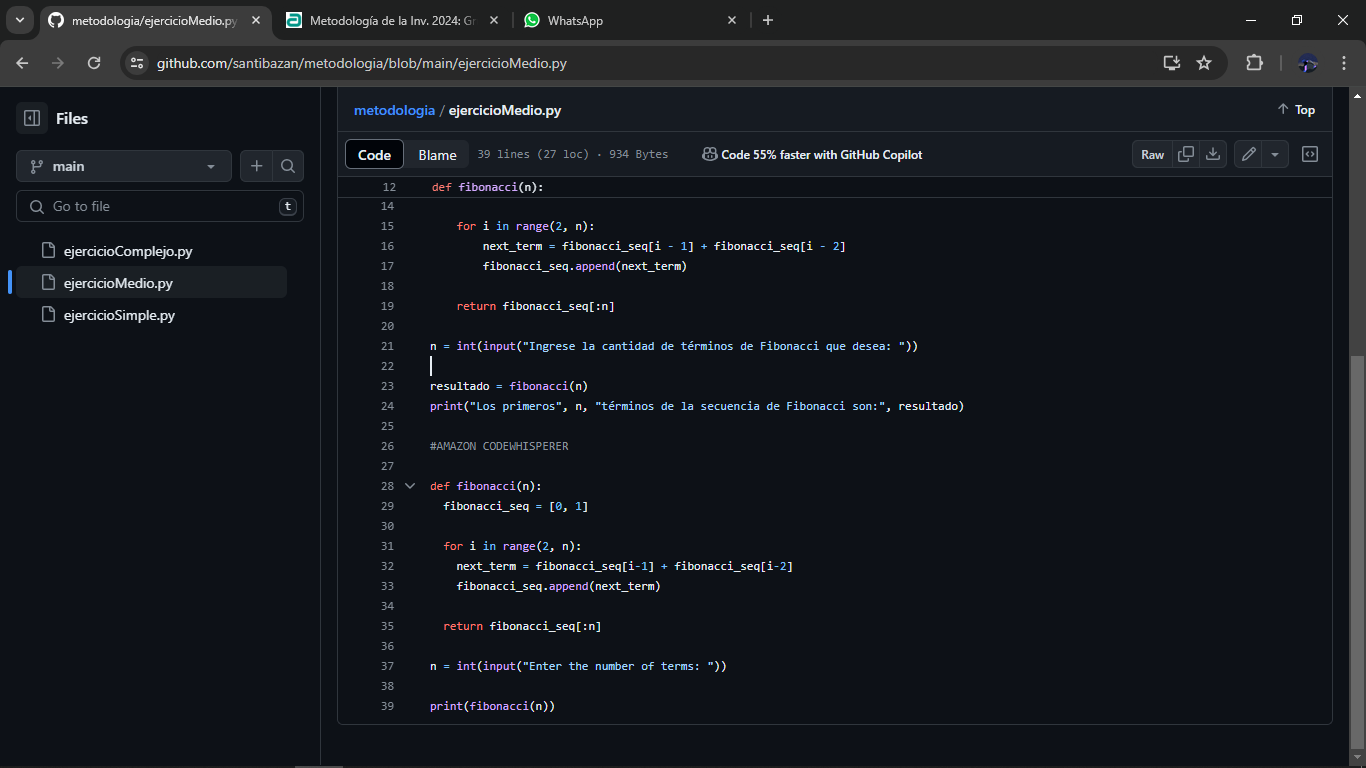
Este algoritmo utiliza un bucle “for” que recorre desde 2 hasta n-1. En cada iteración del bucle, calcula el siguiente término sumando los dos términos anteriores. La complejidad de este algoritmo es lineal, ya que el tiempo de ejecución aumenta linealmente con el tamaño de la entrada n.



**Código 3 (Amazon CodeWhisperer):**

* Complejidad: O(n)

Este algoritmo es similar al código proporcionado por el chat GPT. Utiliza un bucle “for” que recorre desde 2 hasta n-1, calculando cada término de la secuencia de Fibonacci sumando los dos términos anteriores. Como resultado, la complejidad es lineal, O(n).



En resumen, los algoritmos proporcionados por el chat GPT y Amazon CodeWhisperer son mucho más eficientes en términos de tiempo de ejecución para generar los términos de la secuencia de Fibonacci en comparación con el algoritmo proporcionado por nosotros, ya que tienen una complejidad lineal O(n), mientras que el algoritmo que hicimos tiene una complejidad exponencial O(2^n).

Gracias a estos ejemplos podemos concluir que CodeWhisperer es una herramienta que se desenvuelve casi al mismo nivel que chat GPT que es la inteligencia mas usada en estos días, y dado a que es una herramienta que tiene un año de ser lanzada, podemos concluir que a futuro va a ser una de las herramientas mas utilizadas por los programadores del rubro.

**Líneas de Código**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hecho por Nosotros | Chat GPT | CodeWhisperer |
| Ejercicio 1 | 10 | 15 | 14 |
| Ejercicio 2 | 27 | 18 | 12 |
| Ejercicio 3 | 9 | 12 | 11 |

Agregar grafico de barras