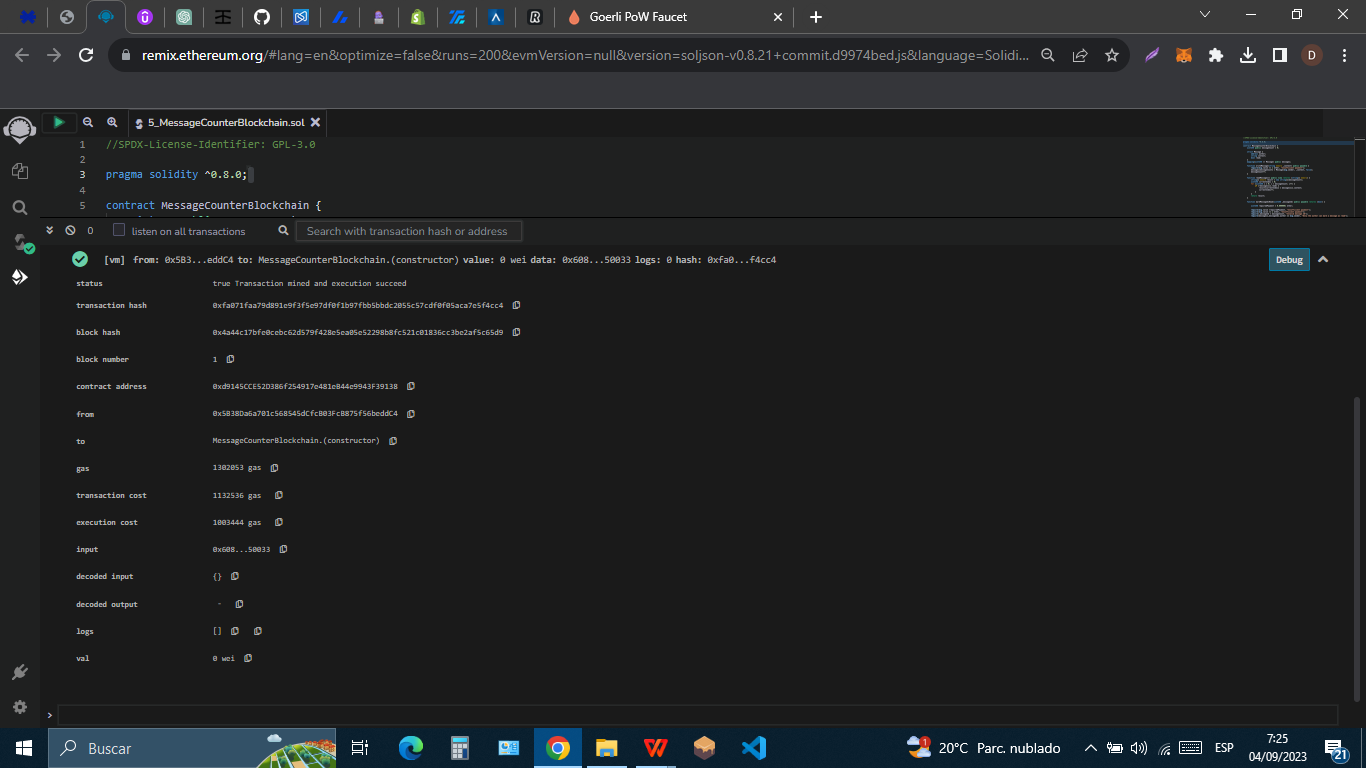
**MÓDULO 4 PRÁCTICA 3 APP CONTADOR MENSAJES ANCLADOS EN LA BLOCKCHAIN**

\*No he utilizado metamask, ya que requiere muchas acciones payable, gas etc, por eso lo he realizado con una VM que provee de cuentas con 100 ethers\*

El contrato creado (basado en el smart contract de la práctica 2) contiene un **contador de mensajes** y estos mensajes son anclados en la blockchain.

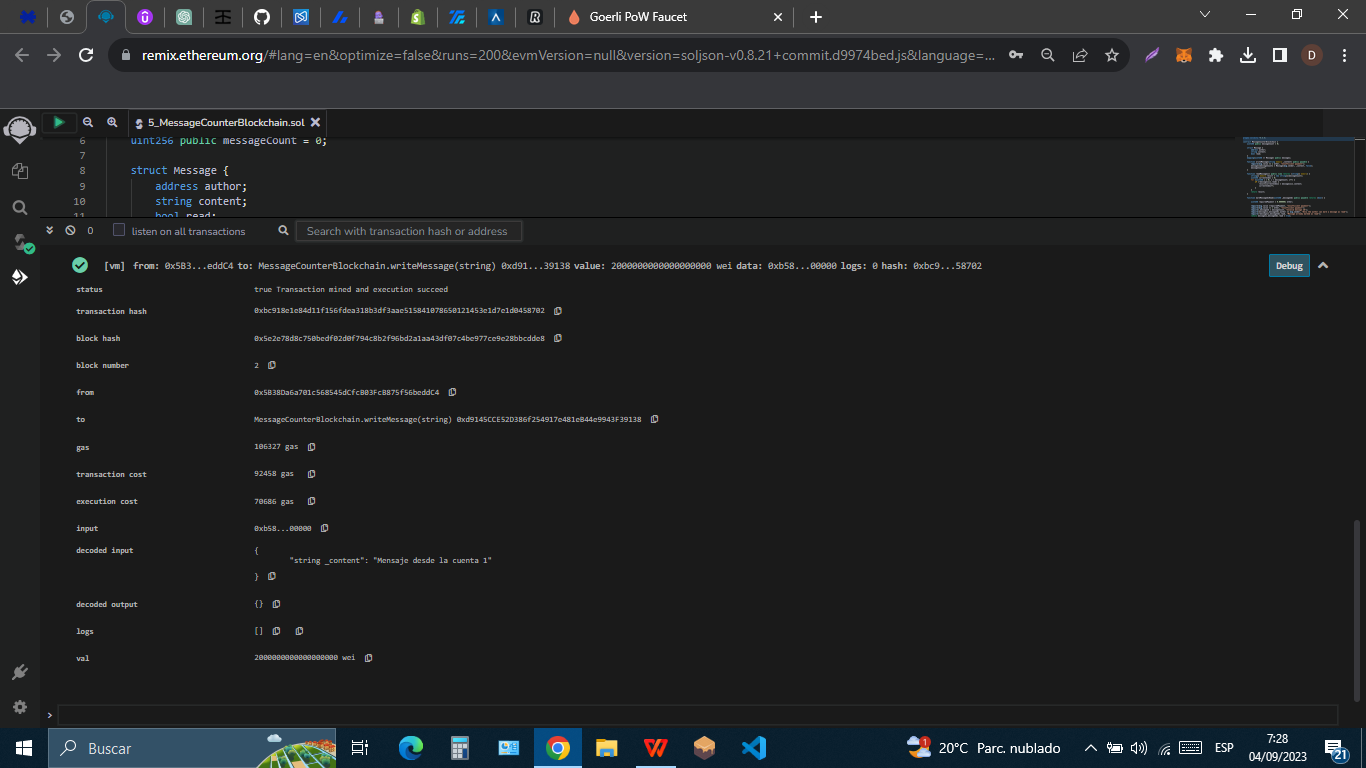
**El contrato tiene las siguientes funciones:**

**-*Desplegar el contrato:*** aquí desplegaré el contrato para más tarde poder realizar las funciones

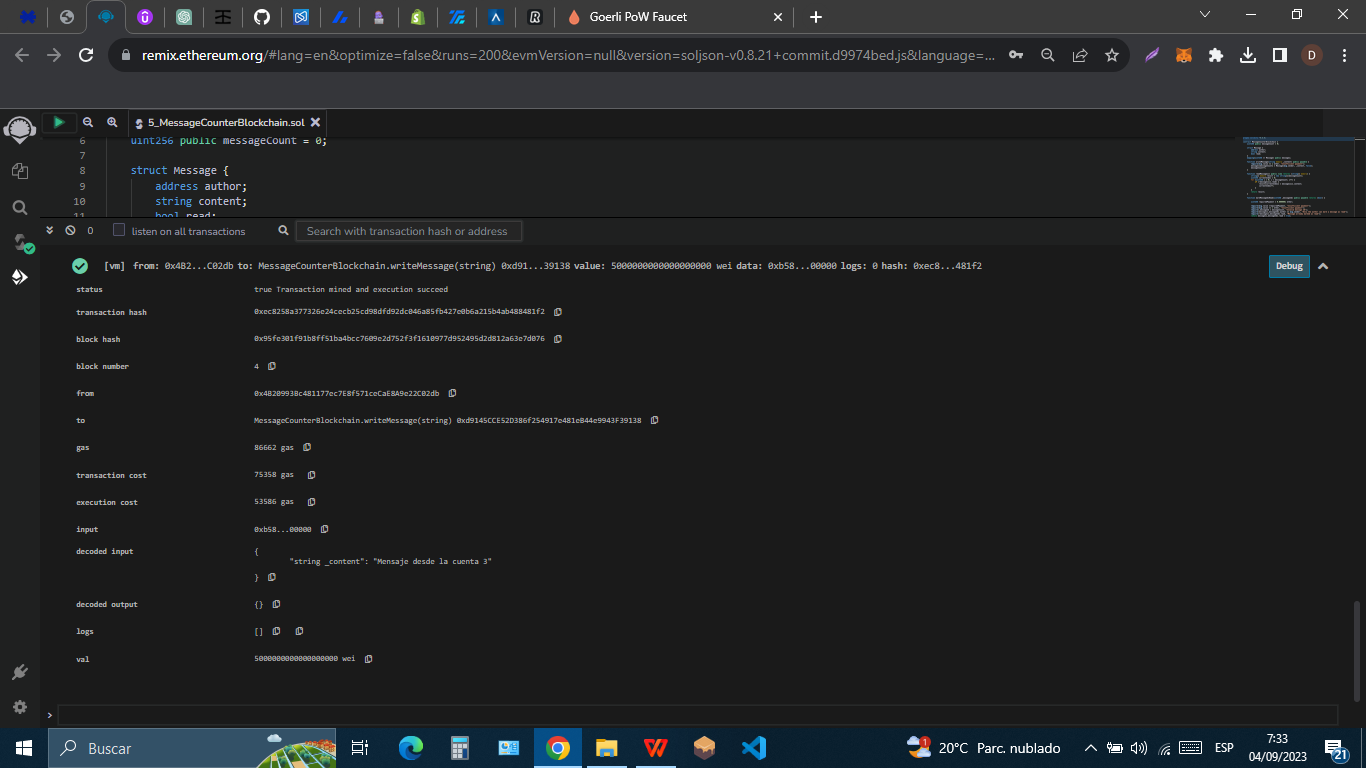


**- *writeMessage*:** Esta función permite a los usuarios escribir mensajes en la blockchain.

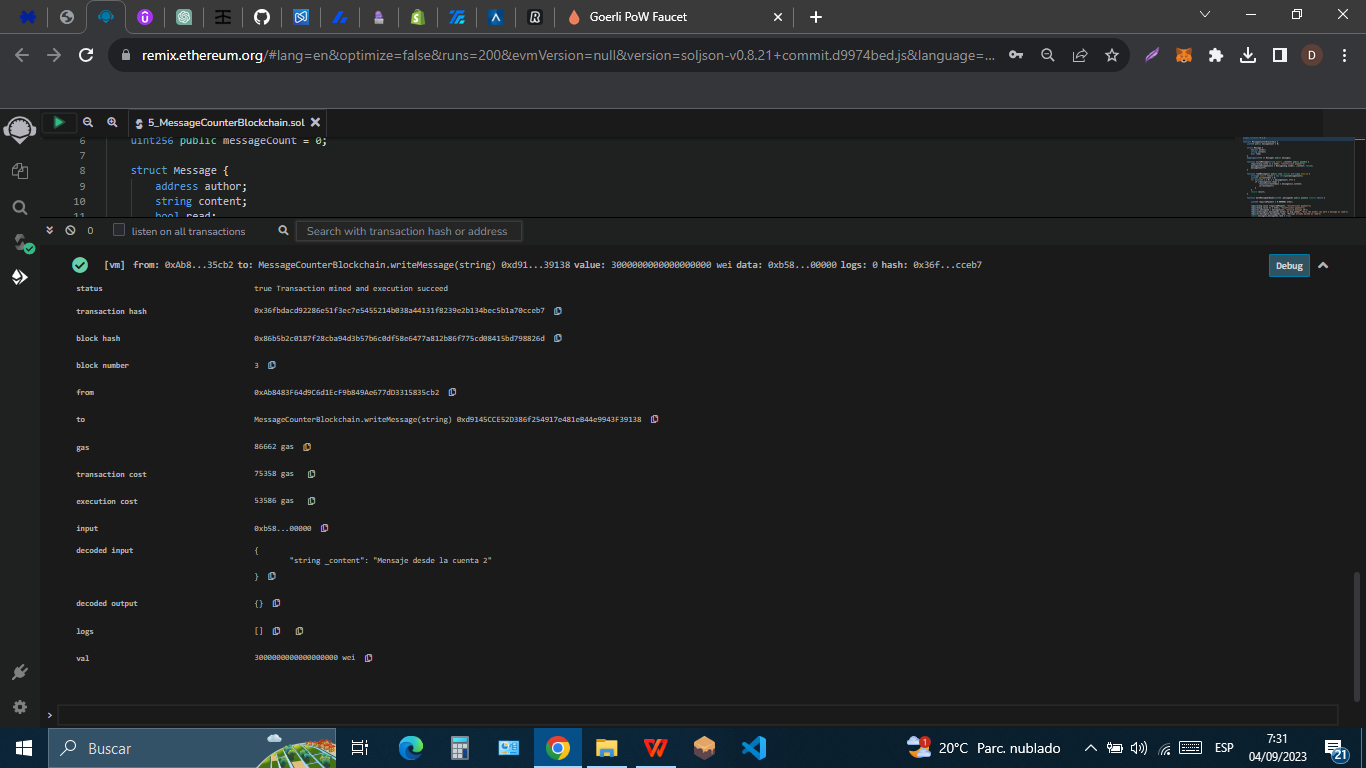
Aquí he escrito un mensaje desde la cuenta 1



Aquí he escrito un mensaje desde la cuenta 2



Aquí he escrito un mensaje desde la cuenta 3

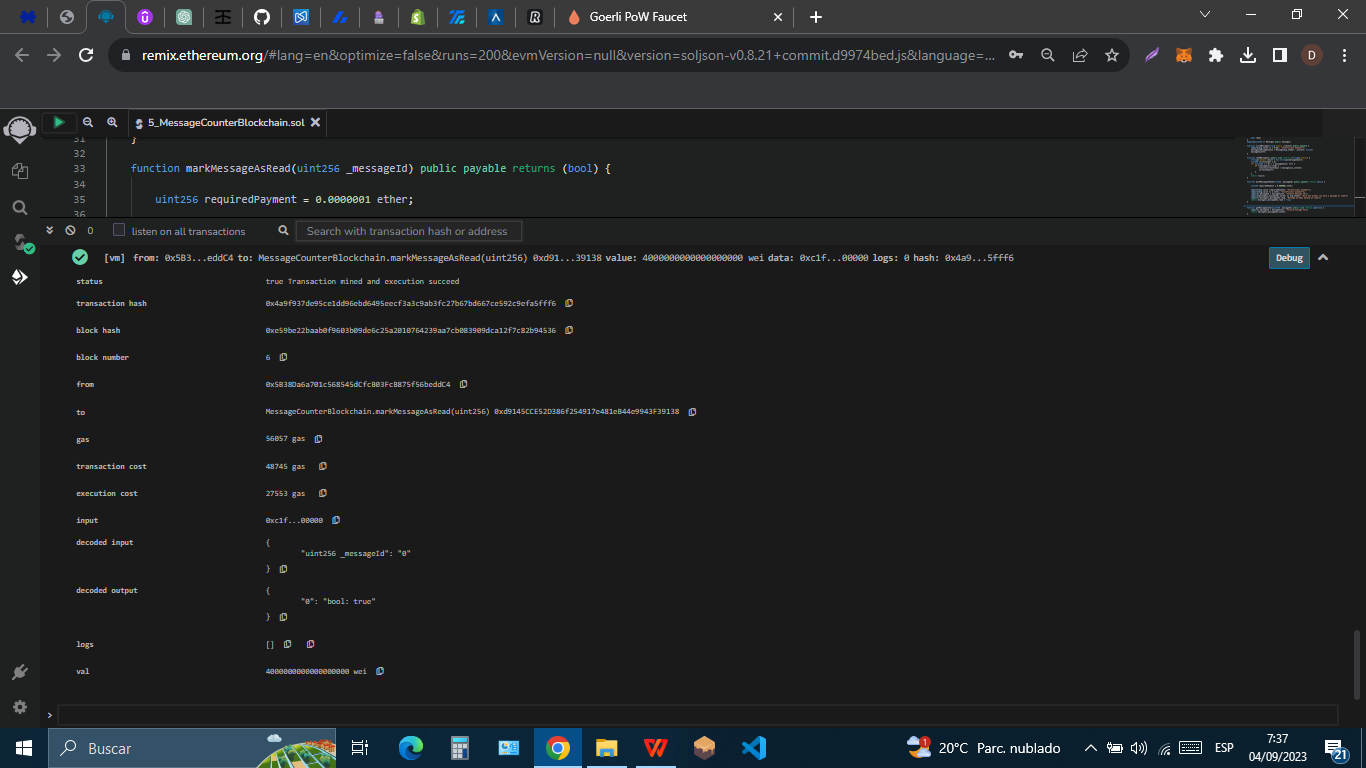


**- *readMessages* :** Esta función permite a los usuarios leer los mensajes que no han sido leídos ya que una vez se han escrito se anclan automáticamente en la blockchain, por eso he eliminado la función ‘***readUnreadMessages***’.

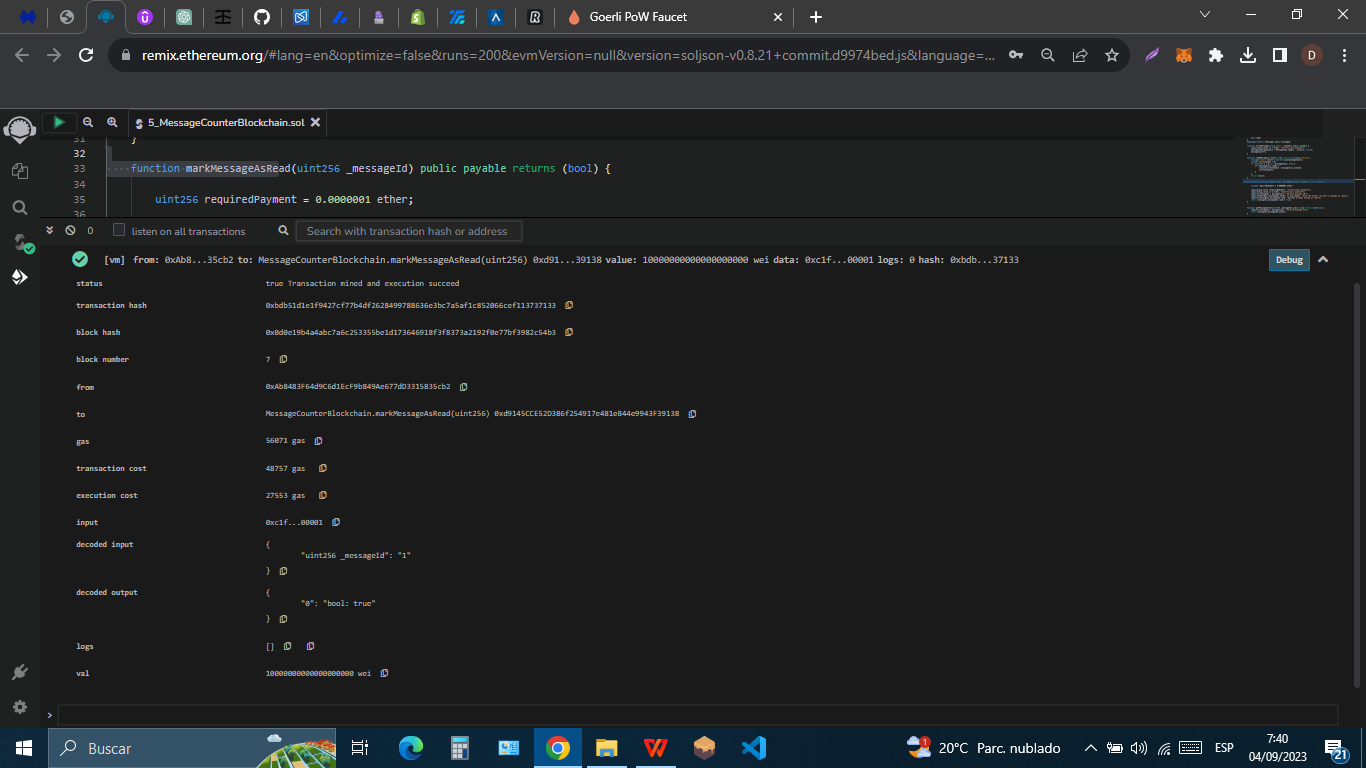
Como primero marque los mensajes comoleídos no salen pero si funciona.

**- *markMessageAsRead* :** Esta función permite a los usuarios que son autores de un mensaje marlarclo como leído

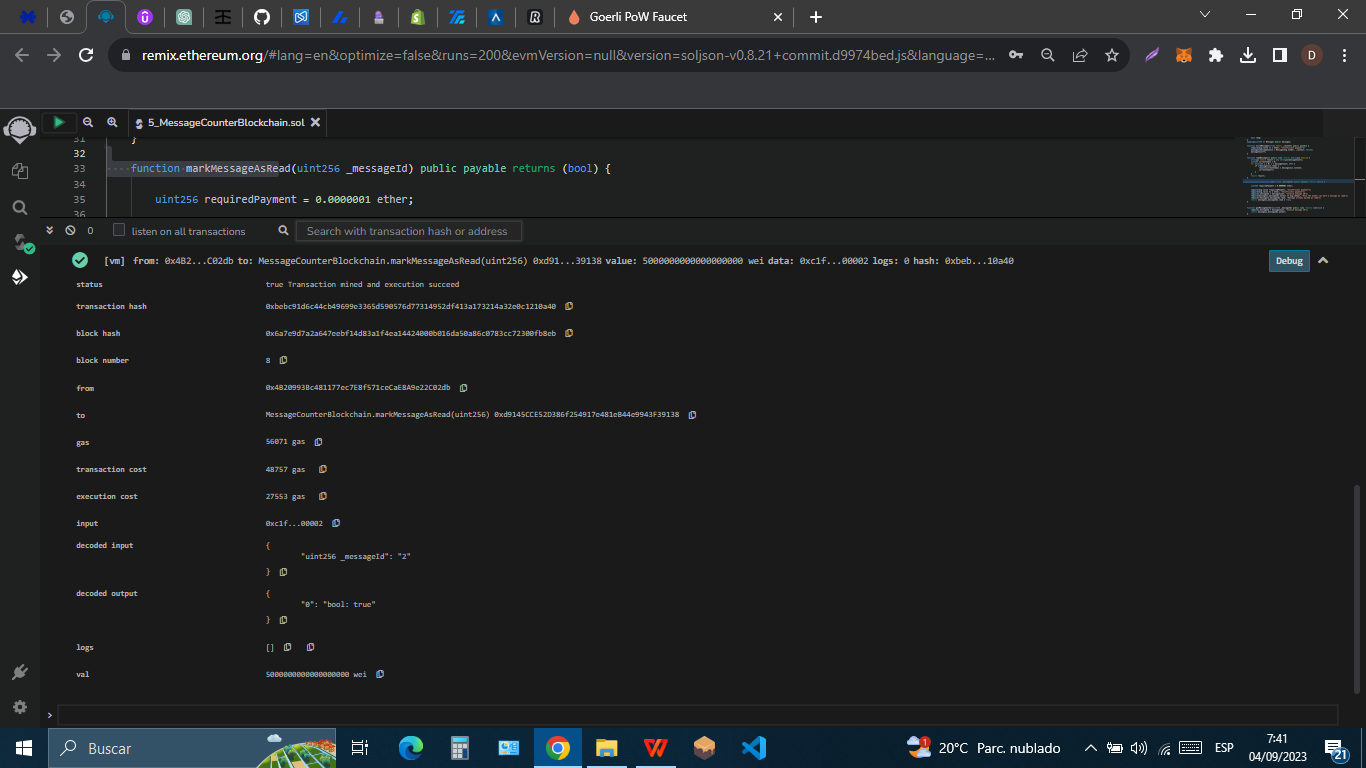
Mensaje de la cuenta 1 leído



Mensaje de la cuenta 2 leído

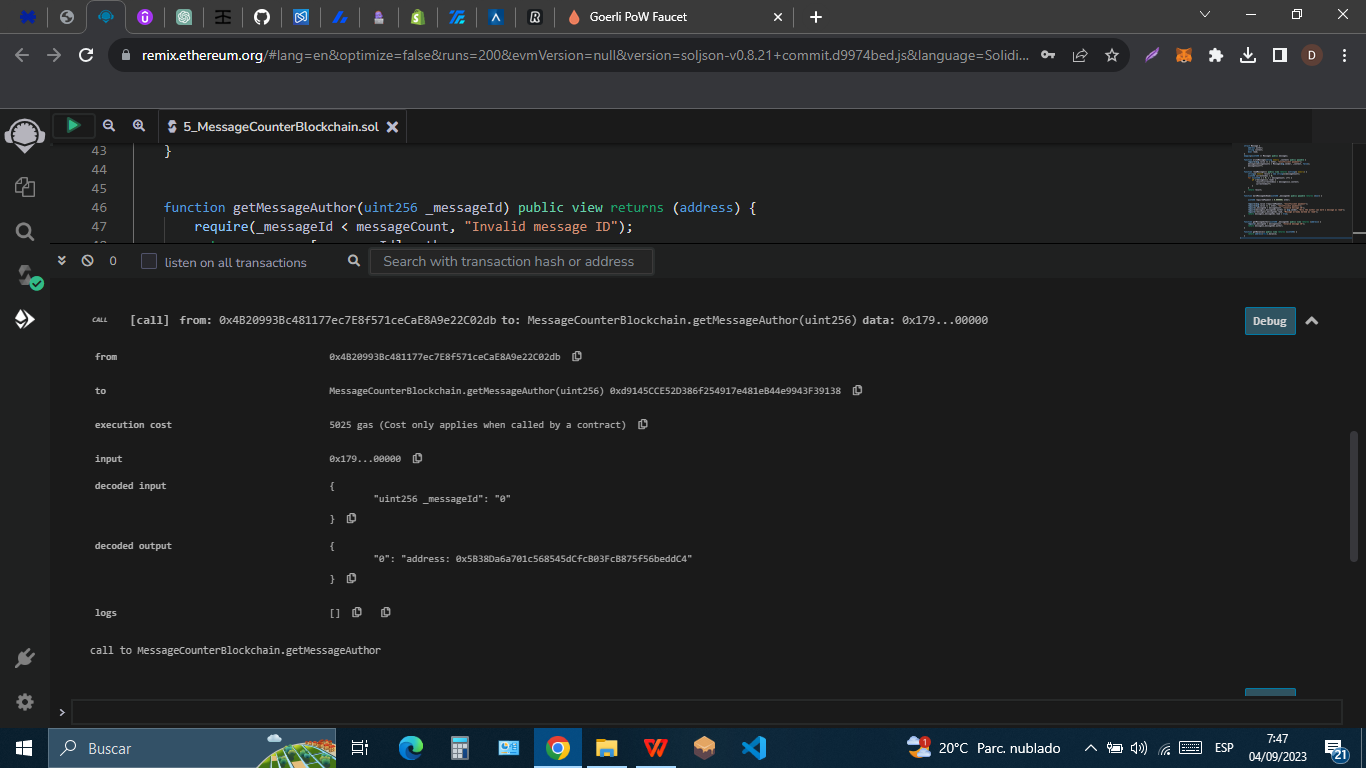


Mensaje de la cuenta 3 leído

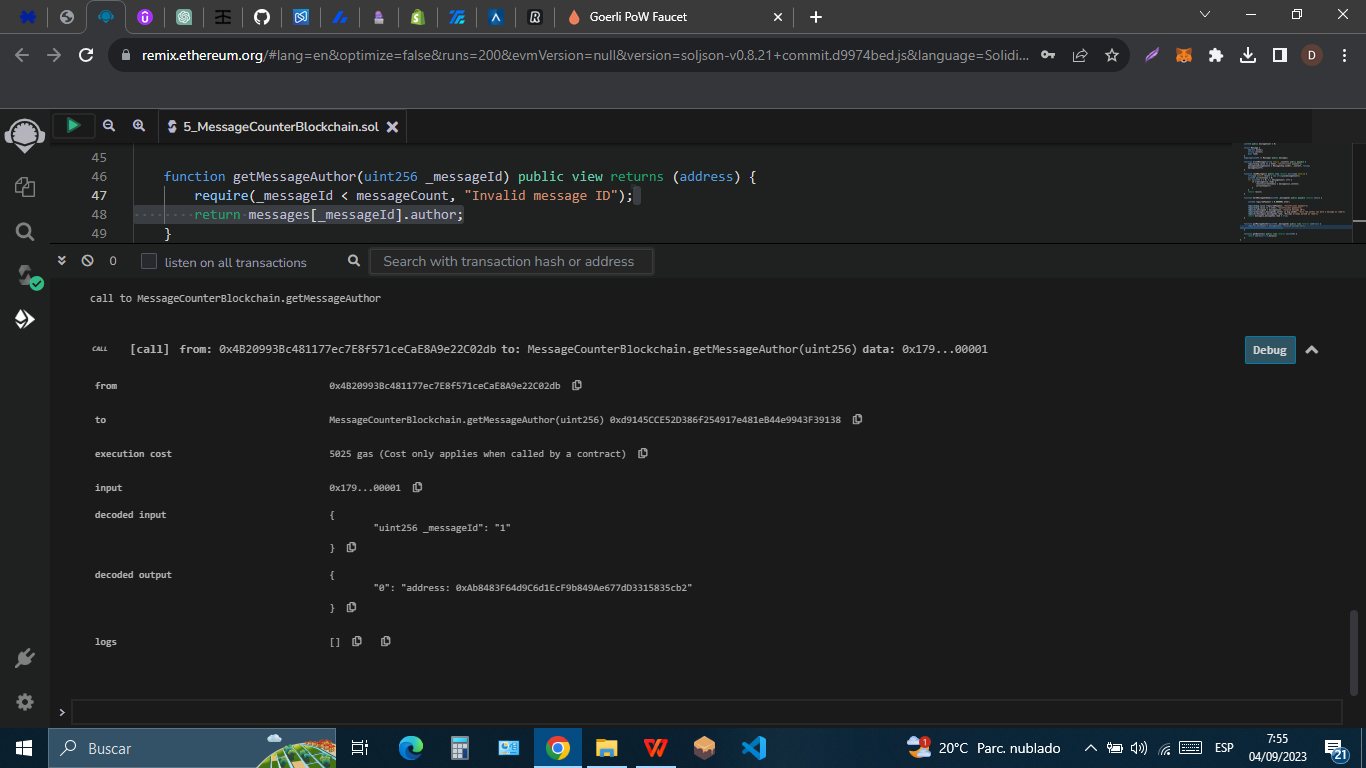


**- *getMessageAuthor* :** Esta función permite a los usuarios obtener la dirección del autor de un mensaje.

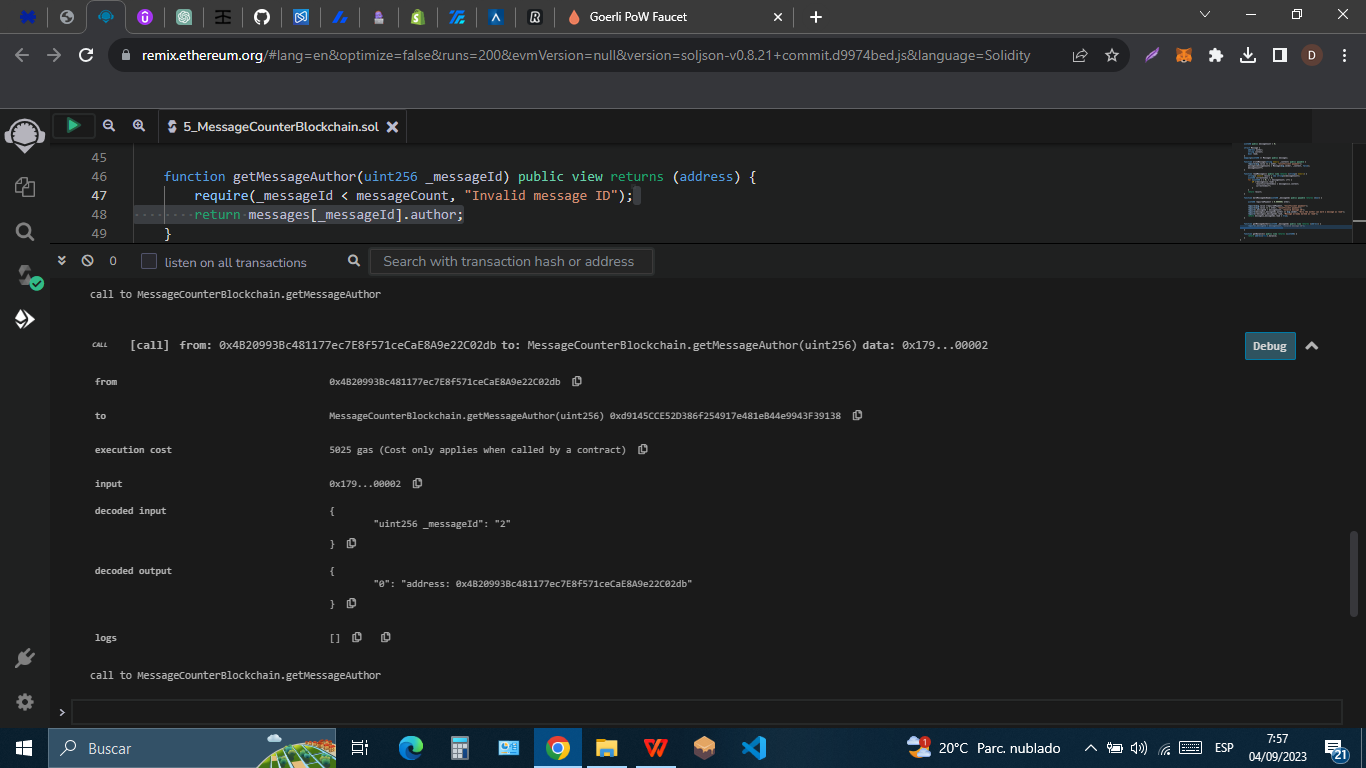
Address del autor del mensaje 1



Adress del autor del mensaje 2

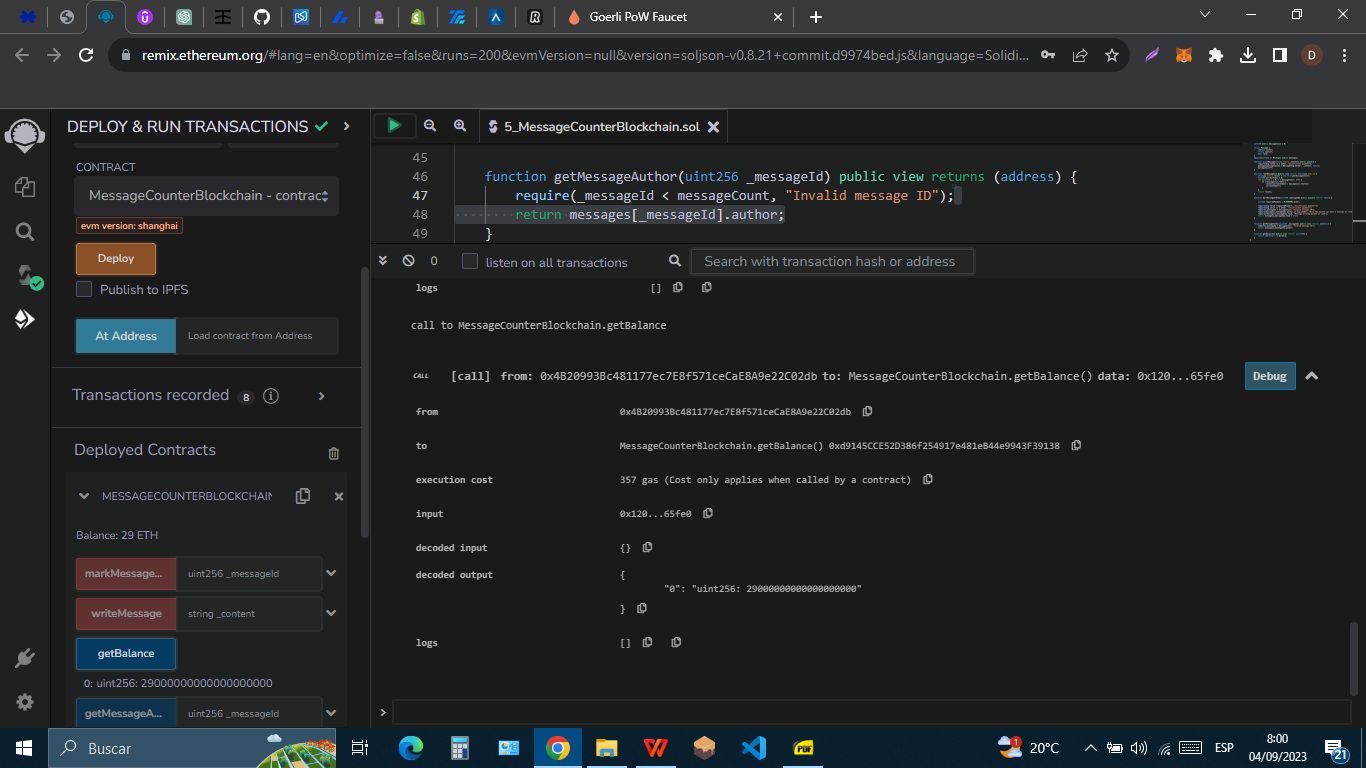


Adress del autor del mensaje 3



**- *getBalance* :** Esta función permite a los usuarios obtener el balance del contrato.

Aquí está el balance de todas las funciones realizadas con éxito de tipo payabley el balance total



**- *messageCount* :** Esta variable almacena el número de mensajes que se han escrito en la blockchain.

