GUÍA DEL ASESOR BLOCKCHAIN

LÍNEA BLOCKCHAIN		
II. NÚCLEO ESPECÍFICO	**e. Desafíos económicos de la blockchain**	
Sesión 15	**Identidad auto soberana y Zero-knowledge Proof**	**Tiempo: 3 hora**

Con antelación verificar si se cuenta con:

Aula virtual	**X**	**Accesos**	**X**	**Presentación**	**X**	**Recursos**		**Cámara y micrófono**	**X**

Python
Ganache
Microsoft Visual C++ 14.0
Visual studio Code

Objetivos

- Entender cómo es el proceso de despliegue de los contratos inteligentes en un ambiente como ganache o en rinkeby por medio de scripts creados en Python.

Introducción al tema

En esta sesión vamos a abordar el tema de identidad autosoberana que propone que los individuos puedan administrar y presentar sus activos digitales utilizando billeteras digitales y que puedan compartirlos e intercambiarlos con garantías de seguridad y maximización de privacidad.

También es importante hacer esta pregunta ¿Es posible demostrar que algo es verdad sin desvelar los datos que lo prueban? Es lo que propone la tecnología 'Zero Knowledge Proof', o Prueba de Conocimiento Cero, una técnica que emplea algoritmos criptográficos para que varias partes puedan verificar la veracidad de una información sin necesidad de compartir los datos que la componen.

Finalmente desde una perspectiva técnica vamos a usar el contrato básico que permite usar el contrato de hello world usado en sesiones anteriores para desplegarlo en ganache e infura. De esta forma más allá de la aproximación técnica, los participantes podrán entender las interacción básicas con los contratos inteligentes que se pueden definir como desplegarlo, escribir y leer.

Infura es una plataforma que proporciona un conjunto de herramientas e infraestructuras que permiten a los desarrolladores llevar fácilmente su aplicación blockchain de la prueba, a la implementación a escala, con acceso simple y confiable a Ethereum e IPFS.

En otras palabras Infura proporciona una puerta de acceso a la red Ethereum sin tener que ejecutar su propio nodo Ethereum.

Método: Propuesta pedagógica

La sesión se realizará de forma sincrónica a distancia a través de la plataforma TEAMS.

- Primera parte: charla magistral de un invitado especial para todos los participantes de la línea blockchain, se realizará con apoyo audiovisual y una ronda de preguntas para resolver las dudas de los participantes presentes.
  Duración: 1 hora.

- Segunda parte: los participantes se dirigen a sus respectivos grupos con su asesor asignado. El asesor contextualizará y profundizará el tema de la sesión, y responderá a las preguntas de los participantes que tengan relación con la conferencia. Se realizarán las actividades previstas con acompañamiento del asesor blockchain, utilizando demos para reforzar conocimientos a través de la práctica.
  Duración: 2 horas.

Habilidades y competencias desarrolladas

- Poder desplegar un contrato en una red de pruebas local
- Poder desplegar un contrato en una red de pruebas rinkeby
- Usar python para crear scripts que permitan interactuar con los contratos inteligentes

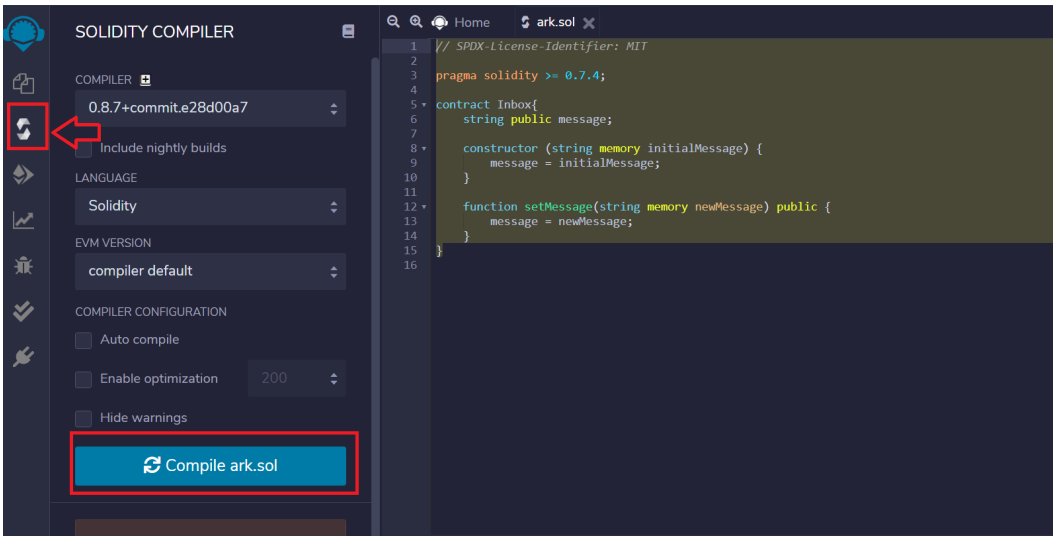
Recursos

- https://infura.io/
- Video: https://drive.google.com/drive/folders/1eWDZX4UNRwv_Nf6Ezdj0AXveZFMKIzOa?usp=sharing
- Scripts: https://drive.google.com/drive/folders/1juCnlCH5mhyDCwg8F_hhwZqPOthbB5Ft?usp=sharing
-

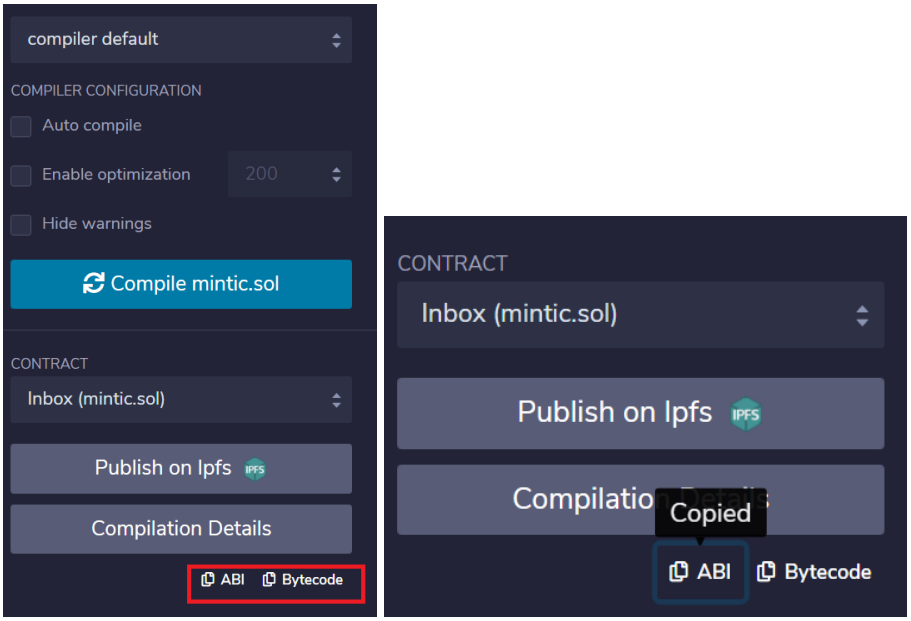
Instrucciones

Actividad 1: Compilar contrato y obtener ABI y Bytecode

Para la actividad es necesario contar con el Bytecode que es el código que se compilado en lenguaje de máquina que le dice a la EVM que hace nuestro contrato inteligente y también el ABI que es la interfaz para comunicarnos desde nuestro script con el contrato.



Una vez compilado en la parte inferior de esta misma ventana encontraremos el ABI y el Bytecode, los cuales debemos copiar en un bloc de notas.



El ABI copia sería igual al de la siguiente imagen

```
[
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "initialMessage",
                "type": "string"
            }
        ],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "inputs": [],
        "name": "message",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "newMessage",
                "type": "string"
            }
        ],
        "name": "setMessage",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    }
]
```

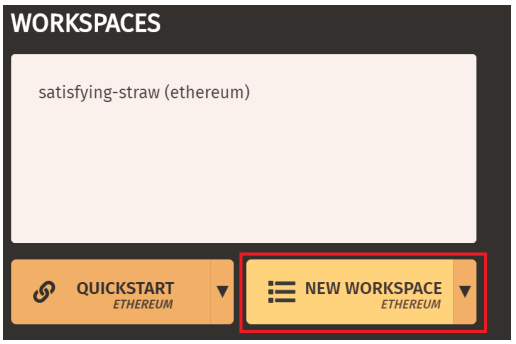

Y el bytecode es un archivo mucho más extenso pero solo necesitamos buscar el “object” al final


```
    "linkReferences": {},
    "object":
"60806040523480156200001157600080fd5b506040516200080e3803806200080e833981810160405281019062000037919062000185565b80600090805190602001906200004f92919062000057565b50506200035a565b828054620000
65906200026b565b90600052602060002090601f016020900481019282620000895760008555620000d5565b82601f10620000a457805160ff1916838001178555620000d5565b82800160010185558215620000d5579182015b828111156
20000d457825182559160200191906001019062000b7565b5b50905062000e4919062000e8565b5090565b5b8082111562000103576000816000905550600101620000e9565b5090565b60006200011e620001188462000 1ff565b6200
01d6565b9050828152602081018484840111156200013d576200013c6200033a565b5b6200014a84828562000235565b50939250505050565b600082601f8301126200016a576200016962000335565b5b81516200017c848260208601620000
107565b91505092915050565b6000602082840312156200019e576200019d62000344565b5b600082015167ffffffffffffffff811115620001bf57620001be6200033f565b5b620001cd8482850162000152565b91505092915050565b60
00620001e2620001f5565b9050620001f08282620002a1565b919050565b6000604051905090565b600067ffffffffffffffff8211156200021d576200021c62000306565b5b620002288262000349565b9050602081019050919050565b6
0005b838110156200025557808201518184015260208101905062000238565b8381111562000265576000848401525b50505050565b600060028204905060018216806200028457607f821691505b602082108114156200029b576200029a
620002d7565b5b50919050565b620002ac8262000349565b810181811067ffffffffffffffff82111715620002ce57620002cd62000306565b5b80604052505050565b7f4e487b7100000000000000000000000000000000000000000000
000000000000600052602260045260246000fd5b7f4e487b7100000000000000000000000000000000000000000000000000000000600052604160045260246000fd5b600080fd5b600080fd5b600080fd5b600080fd5b6000601f19601f83
01169050919050565b6104a4806200036a6000396000f3fe608060405234801561001057600080fd5b50600436106100365760003560e01c8063368b87721461003b578063e21f37ce14610057575b600080fd5b610055600480360381019
06100509190610230565b610075565b005b61005f61008f565b60405161006c91906102b2565b60405180910390f35b806000908051906020019061008b92919061011d565b5050565b6000805461009c90610388565b80601f0160208091
0402602001604051908101604052809291908181526020018280546100c890610388565b80156101155780601f106100ea57610100808354040283529160200191610115565b820191906000526020600020905b815481529060010190602
0018083116100f857829003601f168201915b505050505081565b82805461012990610388565b90600052602060002090601f01602090048101928261014b5760008555610192565b82601f1061016457805160ff19168380011785556101
92565b82800160010185558215610192579182015b82811115610191578251825591602001919060010190610176565b5b50905061019f91906101a3565b5090565b5b808211156101bc57600081600090555060010161 01a4565b5090565
b60006101d36101ce846102f9565b6102d4565b9050828152602081018484840111156101ef576101ee61044e565b5b6101fa848285610346565b50939250505050565b600082601f83011261021757610216610449565b5b81356102278482
602086016101c0565b91505092915050565b60006020828403121561024657610245610458565b5b600082013567ffffffffffffffff81111561026457610263610453565b5b61027084828501610202565b91505092915050565b6000610
2848261032a565b61028e8185610335565b935061029e818560208601610355565b6102a78161045d565b840191505092915050565b600060208201905081810360008301526102cc8184610279565b905092915050565b60006102de6102
ef565b90506102ea82826103ba565b919050565b6000604051905090565b600067ffffffffffffffff8211156103145761031361041a565b5b61031d8261045d565b9050602081019050919050565b600081519050919050565b600082825
260208201905092915050565b82818337600083830152505050565b60005b83811015610373578082015181840152602081019050610358565b83811115610382576000848401525b50505050565b600060028204905060018216806103a0
57607f821691505b602082108114156103b4576103b36103eb565b5b50919050565b6103c38261045d565b810181811067ffffffffffffffff821117156103e2576103e161041a565b5b80604052505050565b7f4e487b710000000000000
0000000000000000000000000000000000000000000600052602260045260246000fd5b7f4e487b7100000000000000000000000000000000000000000000000000000000600052604160045260246000fd5b600080fd5b600080fd5b6000
80fd5b600080fd5b6000601f19601f830116905091905056fea26469706673582212202d76b4113b05f67ea929d116d4df8d08dd08c71e54c98d58f58000a4ceb115d764736f6c63430008070033",
```

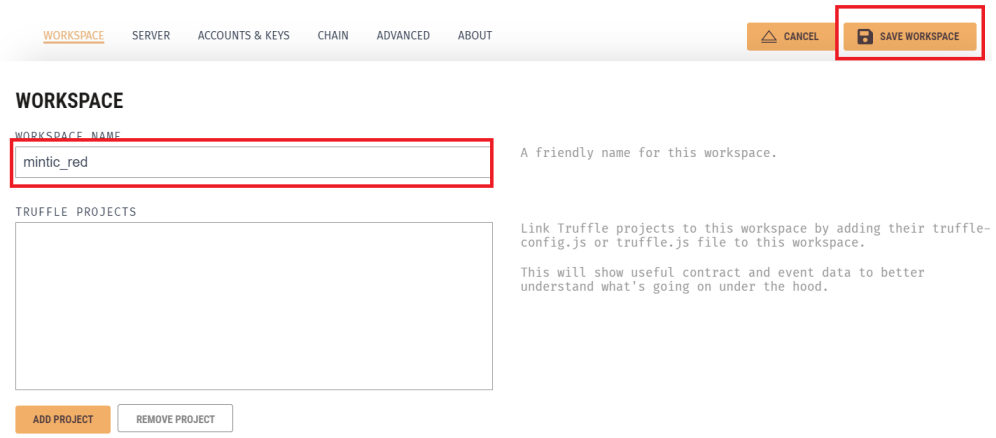

Estos dos elementos ya se encontrarán en los scripts de python que se van a usar en esta sesión.

Actividad 2: Desplegar contratos en ganache

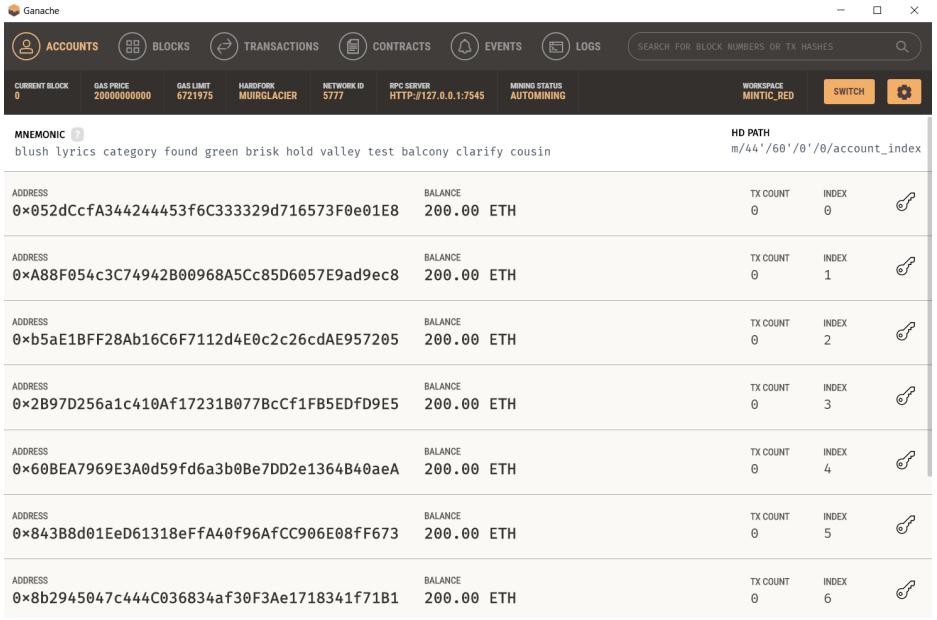
- Al iniciar ganache debes elegir la opción NEW WORKSPACE, este proceso puede tardar unos 3 minutos en completarse.



- Le asignamos un nombre que en este caso puede ser “mintic_red” y le damos en la opción SAVE WORKSPACE



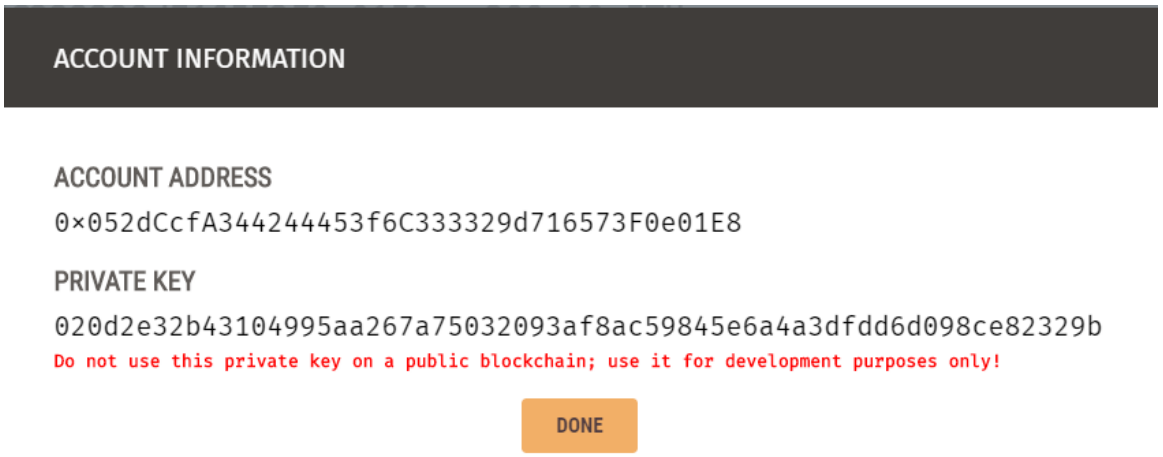
- Debería aparecer una ventana como la que se ve a continuación que es similar a las cuentas que nos creaba remix al momento de usar la JavaScript VM



- En la parte superior se pueden desplazar por la opciones BLOCKS, TRANSACTIONS etc para mostrarles a los asistentes que como es una “red” nueva no tiene ningún registro.

- A continuación necesitarás un editor de texto/archivos, recomendamos Visual Studio Code, uno de los editores gratuitos de código abierto más popularesVaya a https://code.visualstudio.com/, descargue e instale la última versión de Visual Studio Code para su sistema operativo.
- Una vez instalado Visual Studio Code, se deben usar los scripts que se encuentran en el siguiente .rar . Solamente tienen que descomprimirlo, proceder a abrir Visual Studio Code y usar la opción Open / Open Folder y seleccionar la carpeta ganache, debería verse así.

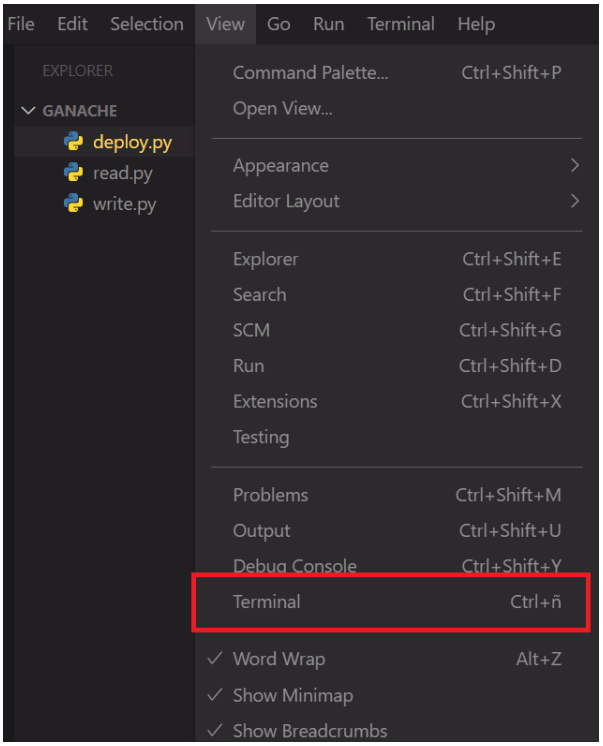
- A continuación en el archivo deploy debes cambiar la información de la línea 49 y 50 por su cuenta y llave privada de ganache con la siguiente información :



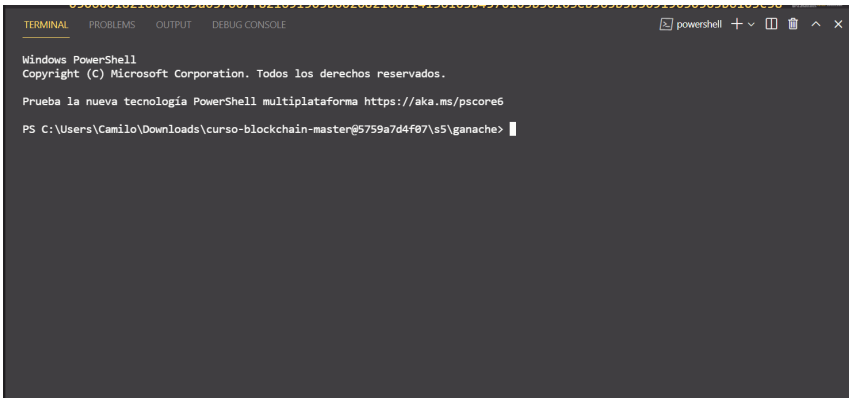
- Debería verse así, teniendo en cuenta la información de la imagen anterior y es fundamental no olvidarse de guardar los cambios en el archivo usando el atajo (CTRL + S).

```
49  account_a = "0x052dCcfA344244453f6C333329d716573F0e01E8"
50  key_a = "020d2e32b43104995aa267a75032093af8ac59845e6a4a3dfdd6d098ce82329b"
```

- Ya estamos listos para desplegar el contrato en ganache, para hacerlo debemos abrir una terminal en visual studio code, para ello debemos ir al menú superior y seleccionar view -> Terminal o usar el atajo (CTRL + Ñ)



- Y debería aparecer una terminal como en la siguiente imagen



- Ya estamos listos para desplegar el contrato inteligente, para ello la terminal debemos digitar el siguiente comando, para verificar que esté ***instalada la librería web3, aunque este paso se debió realizar en la sesión anterior.***

```
pip install web3
```

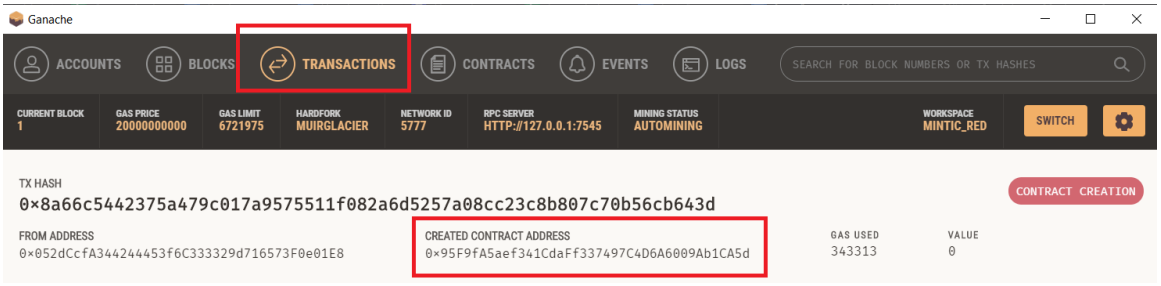
Y seguido vamos a ejecutar el script deploy.py, para lo cual necesitamos digitar el siguiente comando.

```
python deploy.py
```

- Si no se presenta ningún error se debería ver por consola un hash de transacción

```
PS C:\Users\Camilo\Downloads\curso-blockchain-master@5759a7d4f07\s5\ganache> python deploy.py
0x8a66c5442375a479c017a9575511f082a6d5257a08cc23c8b807c70b56cb643d
PS C:\Users\Camilo\Downloads\curso-blockchain-master@5759a7d4f07\s5\ganache>
```

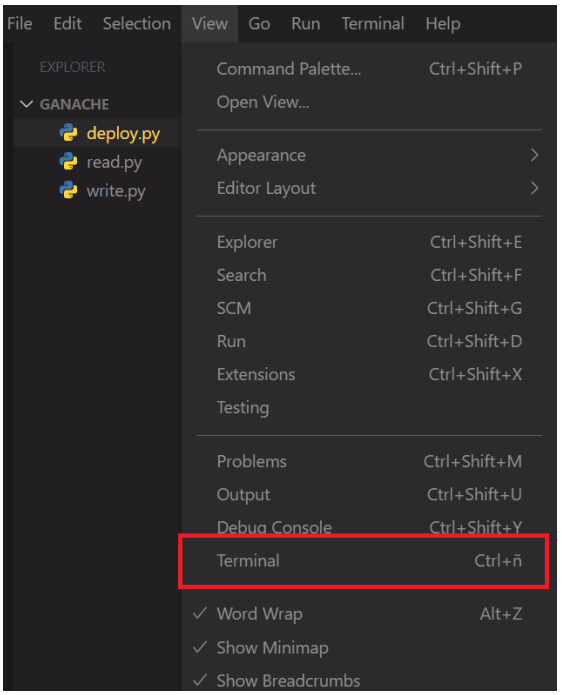
- Ahora en ganache vamos a darnos cuenta que tenemos una transacción de ”CONTRACT CREATION” y con esa dirección de contrato (que va a ser diferente en cada caso) vamos a trabajar en el script read.py y write.py



- Al igual que se hizo anteriormente debemos cambiar la información de account_a, key_a, contract address según corresponda en nuestro ganache en nuestros archivo **read.py y write.py** (ya teníamos la información de account_a y key_a en el archivo deploy.py). Deberían quedar así.

```
account_a = "0x052dCcfA344244453f6C333329d716573F0e01E8"
key_a = "020d2e32b43104995aa267a75032093af8ac59845e6a4a3dfdd6d098ce82329b"
contract_address = "0x95F9fA5aef341CdaFf337497C4D6A6009Ab1CA5d" #cambiar por la dirección de
sus contratos
```

- Teniendo en cuenta que nuestro contrato Inbox se despliega con un mensaje por defecto podríamos usar el script read.py, para ello tenemos que abrir nuevamente la terminal o usar el atajo (CTRL + Ñ)



- Y seguido vamos a ejecutar el script read.py, para lo cual necesitamos digitar el siguiente comando.

```
python read.py
```

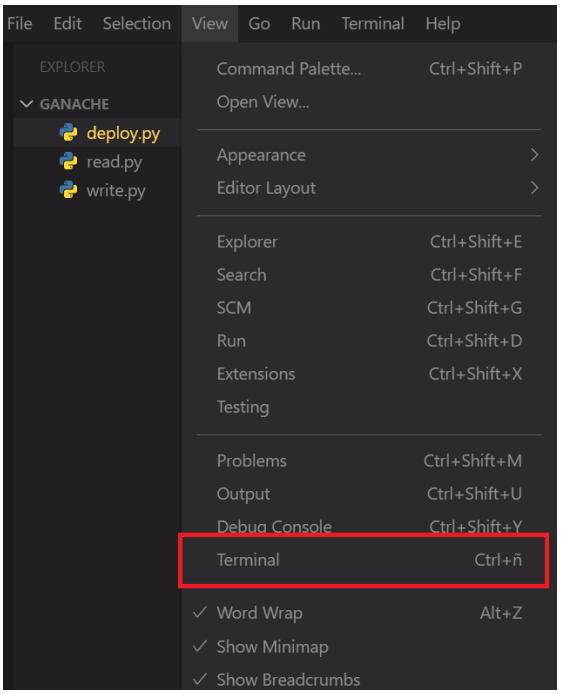
- Si no se presenta ningún error se debería ver por consola el mensaje con el que se desplegó el contrato

```
PS C:\Users\Camilo\Downloads\curso-blockchain-master@5759a7d4f07\s5\ganache> python read.py
Primer mensaje
```

- Ahora solo nos hace falta usar el script write.py, para ello debemos modificar la línea 69 agregando en este caso “Nuevo Mensaje”

```
69      tx = contract.functions.setMessage("Nuevo mensaje").buildTransaction(parameter)
```

- Para ello tenemos que abrir nuevamente la terminal o usar el atajo (CTRL + Ñ)



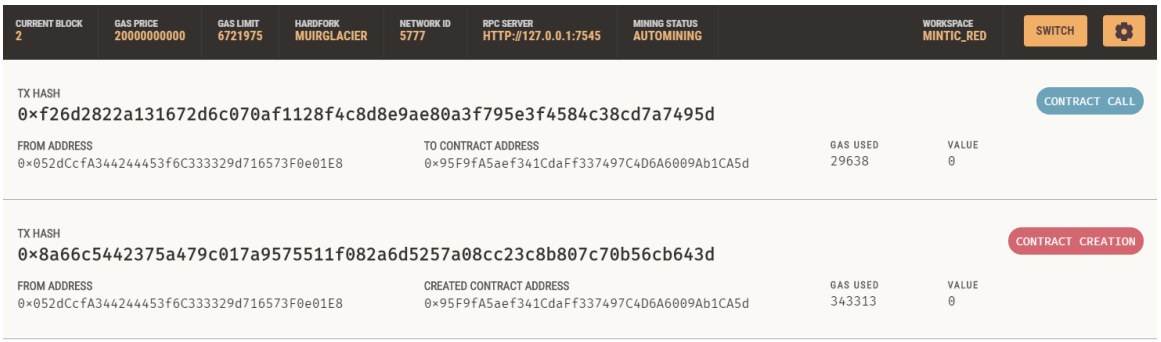
- Y seguido vamos a ejecutar el script write.py, para lo cual necesitamos digitar el siguiente comando.

```
python write.py
```

- Si no se presenta ningún error se debería ver por consola el mensaje con el que se desplegó el contrato

```
PS C:\Users\Camilo\Downloads\curso-blockchain-master@5759a7d4f07\s5\ganache> python write.py
0xf26d2822a131672d6c070af1128f4c8d8e9ae80a3f795e3f4584c38cd7a7495d
```

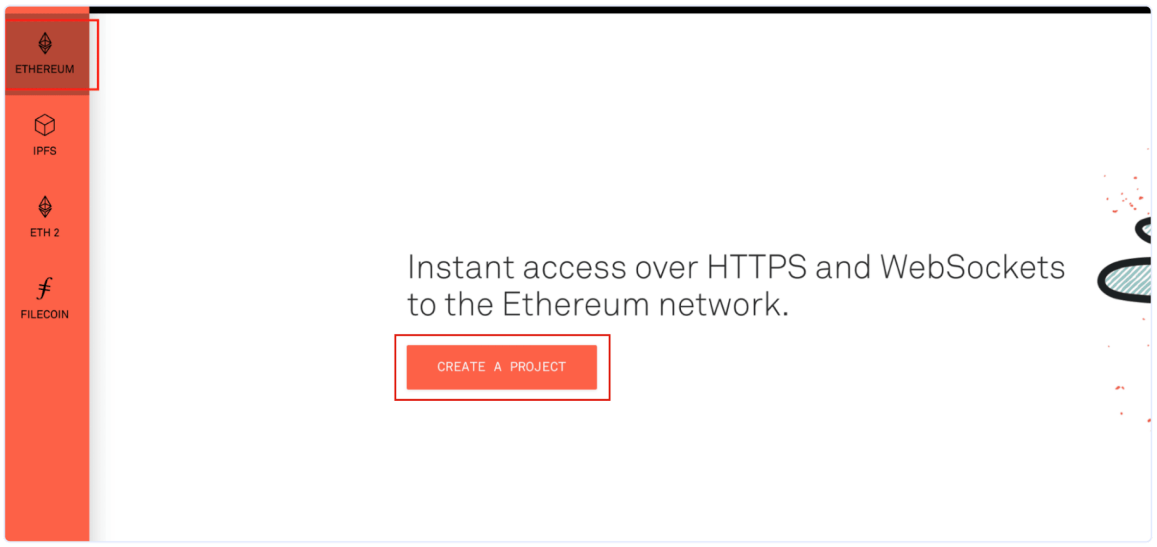
- Como las transacciones que escriben datos en la blockchain generan información para registrar en blockchain en este caso veremos en Ganache que tenemos registrada la transacción.



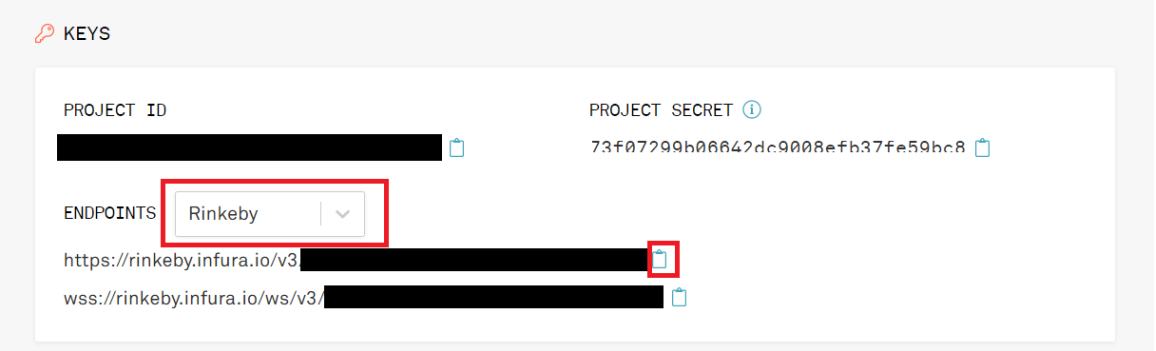
Actividad 3: Desplegar contratos en rinkey usando infura.

Regístrate en Infura

- Dirígete a https://infura.io/product/ethereum, presiona en Sign Up, y rellena tus datos. Una vez que hayas verificado tu dirección de correo electrónico, deberías poder iniciar sesión en el sitio.
- Una vez iniciada la sesión,selecciona el botón Ethereum en el menú de la izquierda, y selecciona 'Create a Project', llámalo 'Mintic' y presiona Create.



- Una vez que estés en tu proyecto, en la configuración del proyecto, cambia el desplegable 'ENDPOINTS' a 'Rinkeby', copia y toma nota de tu endpoint HTTPS endpoint que lo vamos a utlizar para agregarlo a los scripts.



- En los scripts que usastes anteriormente debes cambiar la línea 5 por la URL del ENDPOINTS de infura, debería verse algo así

```
url = "https://rinkeby.infura.io/v3/d96f4893fad8420696838c8f06aa38a5"
```

-
- En este caso vamos a usar los mismo script de ganache pero tenemos que cambiar account_a, key_a por los datos de nuestro metamask y podemos proceder a realizar los mismos pasos que en ganache pero esta vez vamos a poder ver las transacciones en el rinkeby scan.

En el siguientes enlaces encuentran los scripts para trabajar con ganache y rinkeby además del video para que puedan seguir todos los pasos.

- Video: https://drive.google.com/drive/folders/1eWDZX4UNRwv_Nf6Ezdj0AXveZFMKIzOa?usp=sharing
- Scripts: https://drive.google.com/drive/folders/1juCnlCH5mhyDCwg8F_hhwZqPOthbB5Ft?usp=sharing
-

BALANCE DEL DÍA

- Con la instalación de ganache, el participante podrá conectar de forma práctica todos los conceptos y los procesos abordados de forma teórica en las anteriores sesiones.
- El participante cuenta con un ambiente de pruebas para comenzar a desplegar sus contratos inteligentes sin necesidad de usar las redes públicas de Ethereum.