

BC Práctica 2

Autor: Samuel López Iglesias
Daniel Balea Muñoz
Alexandre Sousa Cajide

1. Código React

El código de React se encarga de interactuar con Metamask, IPFS y el contrato inteligente de Ethereum. A continuación, se destacan algunos puntos clave del código:

- **Inicialización de Web3:** El código utiliza la biblioteca Web3 para interactuar con Metamask y establecer una conexión con la red Ethereum.
- **Contrato Inteligente:** Se utiliza el contrato inteligente `SistemaVotacion` en la red Ethereum. Se establece la conexión con el contrato y se define su interfaz en el código.
- **Subir a IPFS con Pinata:** Las imágenes de los candidatos se cargan en IPFS utilizando Pinata. Se utiliza la API de Pinata para fijar archivos en IPFS, y la URL resultante (el Hash del archivo) se almacena en el contrato junto con el nombre del candidato.
- **Votar:** Los usuarios pueden votar por un candidato, lo que se refleja en el contrato y actualiza localmente la cantidad de votos.
- **Obtener Ganador:** Se puede obtener el ganador de la votación consultando el contrato.

2. Contrato Inteligente

El contrato inteligente `SistemaVotacion` en Solidity define la lógica principal del sistema de votación. Aquí se resumen sus principales funciones:

- **Agregar Candidato:** Permite agregar nuevos candidatos al sistema con un nombre único.
- **Votar:** Los usuarios pueden votar por un candidato específico.
- **Obtener Información:** Proporciona funciones para obtener información sobre candidatos, el total de candidatos y el ganador de la votación.

3. Uso de Pinata para Subir a IPFS

En el proceso de subir imágenes a IPFS, el código utiliza Pinata, un servicio de alojamiento para archivos en la red IPFS. A continuación, se explica el proceso:

1. **Preparación del Archivo:** El código utiliza la biblioteca axios para enviar una solicitud POST a la API de Pinata. Antes de enviar la solicitud, se prepara un formulario de datos que contiene el archivo de imagen del candidato.
2. **Envío a Pinata:** La solicitud POST se envía a la URL de la API de Pinata con la información del formulario. En los encabezados de la solicitud se incluyen las claves de API de Pinata para la autenticación.
3. **Obtención del Hash de IPFS:** Pinata procesa la solicitud, fija el archivo en IPFS y devuelve el hash único de IPFS asociado al archivo. Este hash se utiliza para construir la URL de acceso a la imagen en IPFS.

El uso de Pinata simplifica el proceso de carga de archivos a IPFS al proporcionar una interfaz fácil de usar y gestionar la complejidad subyacente de la red IPFS.

4. Demostración

En primer lugar, una vez te conectas a la página web se solicitará la conexión con una cuenta de Metamask. Una vez se cumple este requisito, se podrá comenzar a seleccionar a los diferentes candidatos para la votación.

Cada vez que se añada a un candidato habrá que pagar un costo de transacción por realizar la función correspondiente del smart contract, como se puede ver en la Figura 1.

Una vez hecho esto, y tras transcurrir unos pocos segundos, la imagen seleccionada será subida a IPFS y se mostrará en la página (obteniéndola de IPFS), como se puede ver en la Figura 2

Una vez finalicemos de añadir a todos los candidatos podremos proceder con las votaciones presionando el botón Votar. Este botón llamará a la función correspondiente del contrato, pidiendo, al igual que al añadir un candidato, la confirmación de la correspondiente transacción. Esto permitirá aumentar en uno la cantidad de votos del candidato, como se puede ver en la Figura 3.

Finalmente, si pulsamos el botón Obtener Ganador, se nos mostrará por pantalla el candidato ganador de las elecciones, con sus datos correspondientes (Figura 4).

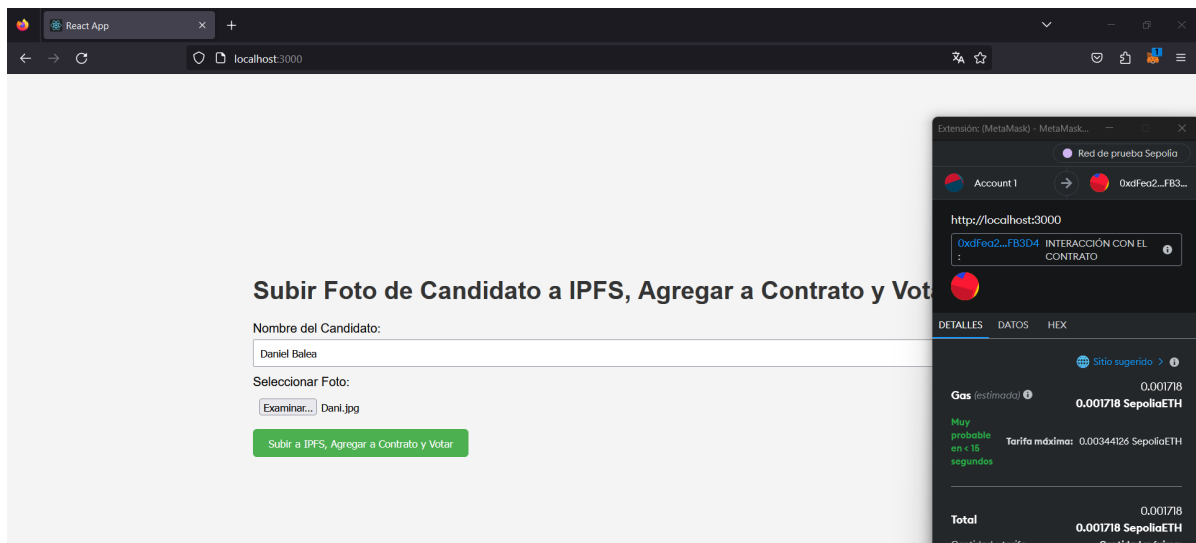


Figura 1: Proceso de Subir Imagen a IPFS con Pinata



Figura 2: Imagen obtenida de IPFS

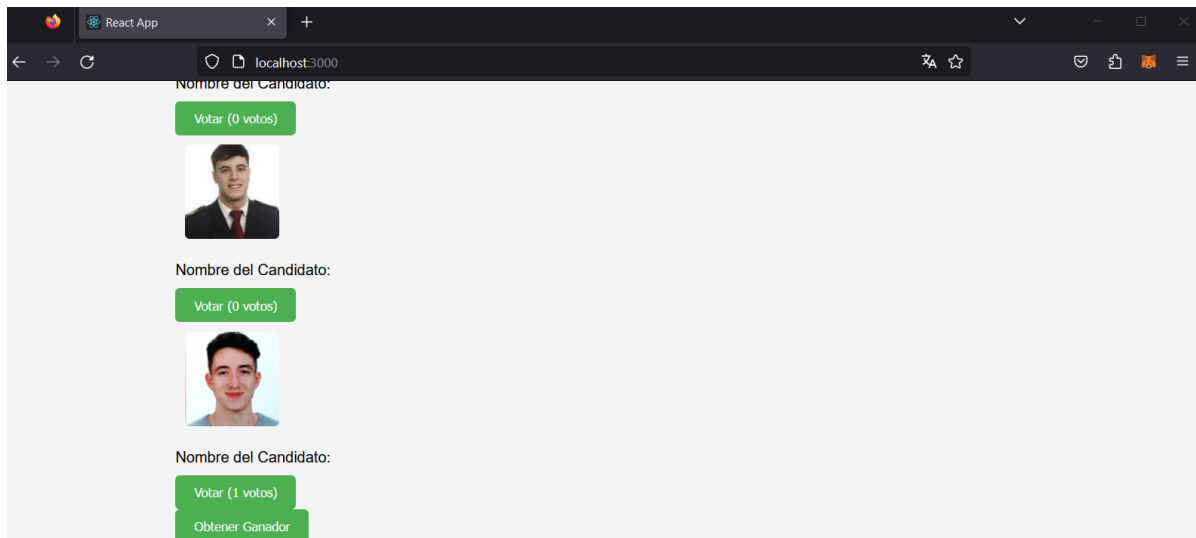


Figura 3: Votación de un candidato

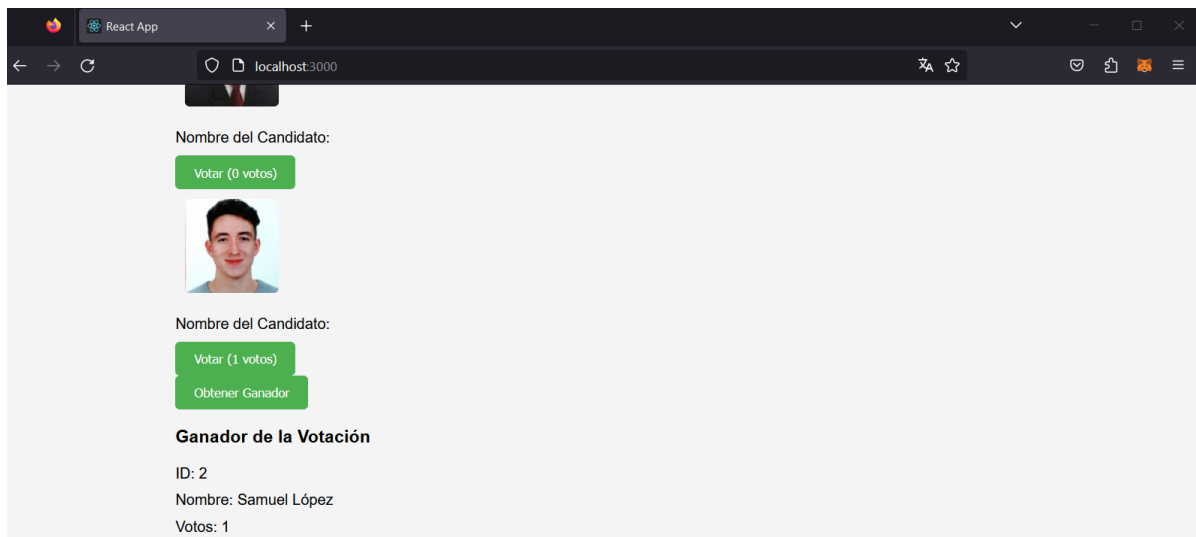


Figura 4: Ganador de la votación