Caso de uso:

Plataforma de Crowdfunding con NFTs y archivos alojados en IPFS

Diego Pérez Martínez y Martín Alonso Pérez

El crowdfunding es un mecanismo colaborativo de financiación donde una comunidad contribuye económicamente para que un proyecto pueda llevarse a cabo. Sin embargo, las plataformas tradicionales dependen de intermediarios centralizados que almacenan la información del proyecto y gestionan los pagos. Esto plantea riesgos como censura, pérdida de datos, comisiones elevadas o falta de transparencia.

Para evitar estos problemas, proponemos un sistema de crowdfunding basado en **blockchain** e **IPFS**, donde:

- El **owner** del proyecto puede subir un documento explicativo a IPFS (PDF, video, presentación, imagen…), garantizando su persistencia y resistencia a la censura.
- Cada usuario que contribuya recibirá un NFT único cuyo **metadata JSON** se aloja en IPFS, lo que permite incluir información como el importe aportado, la dirección del contribuidor y el enlace al archivo principal del proyecto.
- El contrato inteligente gestiona las aportaciones, retiros, reembolsos y vincula a cada usuario con su NFT.

Este enfoque permite un sistema más **seguro, transparente y descentralizado**, eliminando intermediarios y garantizando trazabilidad en todas las contribuciones.

Descripción del caso de uso

Imaginemos un creador que desea financiar el desarrollo de un videojuego independiente. Para explicar su proyecto, sube un documento de presentación (ej.: *Whitepaper.pdf*) a IPFS. De esta manera, ningún servidor central puede eliminarlo o modificarlo.

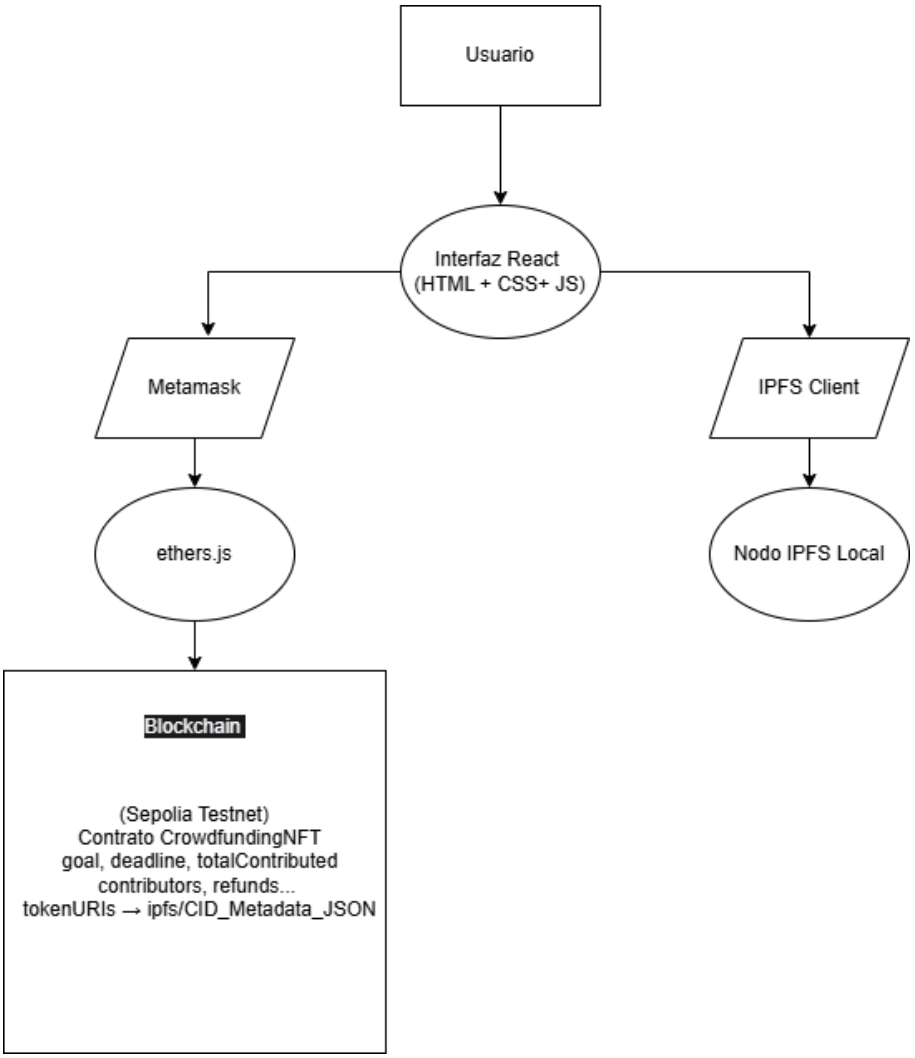
Los usuarios que quieran apoyar el proyecto pueden enviar fondos en ETH al contrato inteligente. A cambio de su contribución reciben un NFT único que podrá consultar en IPFS. Este NFT funciona como:

- Certificado público de participación.
- Posible utilidad futura (descuentos, acceso anticipado, contenido exclusivo).
- Elemento coleccionable del proyecto.

Si la campaña alcanza el objetivo económico antes de la fecha límite, el owner puede retirar los fondos. En caso contrario, los participantes pueden recuperar su dinero (*refund*).

Arquitectura de comunicaciones

El siguiente esquema resume la interacción entre los componentes del sistema:



La arquitectura de comunicaciones del sistema propuesto permite que todos los actores involucrados en la campaña de crowdfunding interactúen de forma segura y descentralizada utilizando blockchain e IPFS. En este modelo intervienen tres capas principales: la capa de interfaz web (front-end), la capa de almacenamiento distribuido (IPFS) y la capa de ejecución lógica y financiera (blockchain).

En primer lugar, los usuarios interactúan con la plataforma desde un navegador web, donde se ejecuta una aplicación desarrollada con React. Esta interfaz permite al **owner** subir el archivo explicativo del proyecto y a los participantes enviar contribuciones y consultar su NFT. Para poder firmar transacciones y comunicarse con la blockchain, la aplicación utiliza **MetaMask**, que expone la API `window.ethereum`, permitiendo la conexión de una wallet real o de pruebas.

La comunicación con la blockchain se gestiona mediante la librería **ethers.js**, que facilita la invocación de funciones del contrato inteligente desplegado en la red de Ethereum (en este caso, una testnet como Sepolia). Estas operaciones incluyen registrar el hash IPFS del

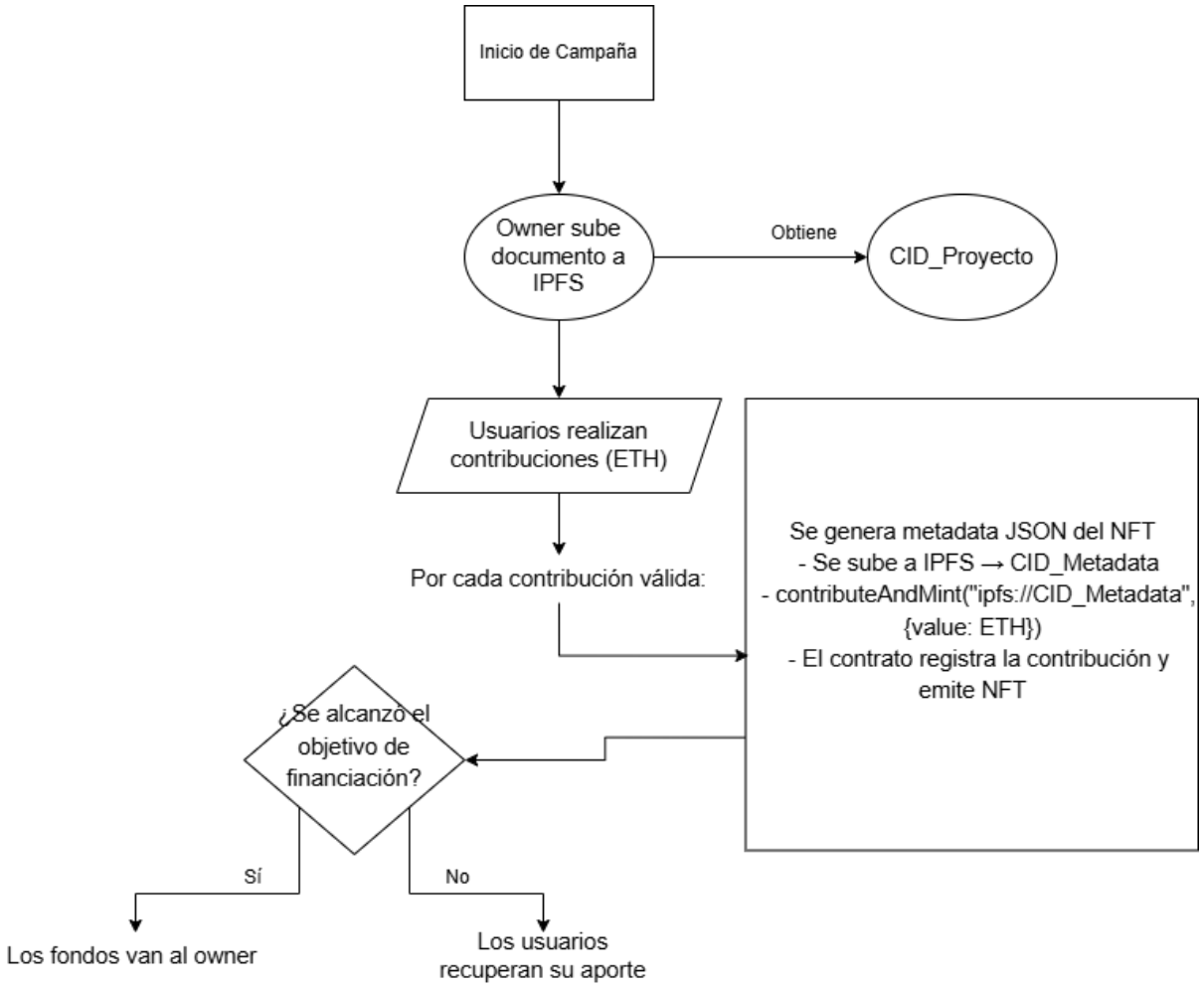
archivo del proyecto, realizar contribuciones económicas, mintear los NFTs y, al finalizar la campaña, retirar fondos o solicitar reembolsos.

De forma paralela, el sistema utiliza un nodo de **IPFS** (ya sea local o remoto) para almacenar tanto el documento explicativo del proyecto como los metadatos JSON asociados al NFT entregado a cada contribuidor. La aplicación se comunica con este nodo a través de **kubo-rpc-client** o alguna librería equivalente, que permite subir archivos, obtener el CID resultante y consultarlos posteriormente mediante un gateway.

Gracias a esta arquitectura distribuida, ningún servidor central controla el contenido ni las transacciones. El acceso a los documentos del proyecto y a los NFTs se realiza mediante enlaces del tipo: http://localhost:8080/ipfs/<CID>

Con este diseño, la integridad del contenido está garantizada por IPFS y la veracidad de las transacciones por la blockchain, logrando así una plataforma de crowdfunding más **resistente a la censura, transparente y verificable**.

Funcionamiento



El funcionamiento de la plataforma de crowdfunding se organiza en varias fases que abarcan desde el inicio de la campaña hasta la retirada de fondos o la devolución del dinero a los participantes.

En primer lugar, el **owner** del proyecto genera un archivo explicativo (por ejemplo, un documento PDF, una presentación o una imagen promocional) en el que detalla los objetivos del proyecto y el uso previsto de los fondos recaudados. Este archivo se sube a la red **IPFS**, obteniendo un identificador único o **CID**. Dicho CID se registra en el contrato inteligente mediante la función `setCampaignFile()`, quedando almacenado como `campaignFileIPFS` y asociado de forma inmutable y verificable a la campaña sin depender de servidores centralizados.

Una vez iniciada la campaña de crowdfunding, los usuarios pueden apoyar la propuesta enviando una contribución económica en **ETH** mediante la función `contributeAndMint()`. Antes de ejecutar dicha transacción, la aplicación genera un fichero **metadata JSON**, que describe el NFT que recibirá el usuario, incluyendo información como:

- la dirección del contribuidor,
- el importe aportado en WEI,
- el CID del archivo principal del proyecto (`campaignFileIPFS`),
- nombre y descripción del NFT,
- y, opcionalmente, una imagen o recurso multimedia vinculado al proyecto.

Este archivo JSON se sube a IPFS y se obtiene un **CID de metadatos**, el cual es enviado como parámetro a `contributeAndMint()`. El contrato registra la contribución, actualiza los totales de recaudación y finalmente **acuña un NFT ERC-721** cuyo `tokenURI` corresponde al CID de este metadata almacenado en IPFS. De este modo, el NFT queda registrado en la blockchain como prueba pública, verificable y descentralizada de participación en la campaña.

Durante toda la campaña, el contrato controla automáticamente la cantidad total recaudada y la compara con el objetivo económico establecido (`goal`). Una vez que la campaña termina, pueden darse dos escenarios:

Objetivo alcanzado:
Si la cantidad recaudada es igual o superior al objetivo, el owner puede ejecutar la función `withdrawFunds()`, que transfiere al propietario del proyecto todos los fondos acumulados. Gracias a la transparencia de la blockchain, el proceso evita la manipulación o retención indebida de fondos.

Objetivo no alcanzado antes de la fecha límite:
Si la campaña finaliza sin haber alcanzado el objetivo, los usuarios que hayan participado podrán recuperar sus aportaciones mediante la función `refund()`. El contrato devuelve a cada participante exactamente la cantidad que aportó, sin intervención del owner ni intermediarios.

Gracias a esta lógica autónoma implementada en el contrato inteligente, la plataforma evita la necesidad de terceros de confianza y proporciona un sistema de financiación colectiva más **seguro, trazable, resistente a la censura y completamente transparente** para todos los participantes.

Lecciones aprendidas

Durante la realización de la práctica nos dimos cuenta que no se podía usar node desde un docker para conectarse con otro docker de ipfs. Con lo cual hay que descargarlo en wsl (ya que en Windows no es muy recomendable).

El navegador usa por defecto ETH , con lo cual lanza un error por pantalla al estar la cuenta de Metamask en Sepolia, por lo cual tuvimos que cambiar el código para obligar a usar esta última.

Tuvimos que modificar la configuración del docker de ipfs para que funcionase, ya que el comando para habilitar el CORS no nos funcionó como debería. La configuración no permitía las acciones HTTP (GET,POST,PUT,DELETE) por defecto y tuvimos que modificarlo en el config del docker ipfs y no habilitaba la autorización del CORS.

Explicación vídeos

En los vídeos podemos apreciar la visión del owner y de un donador para el crowdfunding.
Observamos como el owner introduce su cuenta metamask, sube a ipfs el archivo para generar los NFTs (archivo donde se explicaría el objetivo del crowdfunding y diciendo que eres donador) y hace una donación, haciendo click en el enlace puede ver su NFT JSON con su dirección de cuenta, cuanto dinero ha donado y un enlace a un documento, este será el que el owner subió previamente a IPFS.
Al llegar al límite establecido, recargando la página el owner puede retirar el dinero recaudado para la causa establecida.

En la visión del donador, podemos observar como conecta la cuenta de metamask, hace una donación y observa el NFT JSON que apunta al archivo subido por el owner, en este caso al ser ordenadores distintos y estar todo en local salvo blockchain, no puede ver el archivo IPFS. Al llegar al límite de dinero no le aparece el botón para retirar el dinero recaudado porque no es el owner.