



UNIVERSIDADE DA CORUÑA

Facultad de Informática

Práctica 2: Almacenamiento y distribución con IPFS Ejercicio 2

Autores:
Héctor Núñez Fernández
Fernando Antonio Fung Cen
Santiago Novo Rodríguez

Profesora:
Paula Fraga Lamas

A Coruña, Diciembre 2023

ÍNDICE

1. Análisis y Definición del Escenario.....	2
2. Diseño	3
3. Arquitectura	5
4. Funcionamiento del sistema	6
5. Implementación del laboratorio	7
6. Referencias.....	12

1. Análisis y Definición del Escenario

Para la realización de este segundo ejercicio, y en cierta medida el trabajo tutelado, se plantea la interconexión de una red de estaciones meteorológicas para agricultura privadas instaladas en granjas o fincas, que permitan realizar el intercambio de información más precisa y en tiempo real como soporte, y al final de termino, una mejora al sistema de red de meteorología nacional.

En cierta medida, resulta en una ampliación de las capacidades de mediciones meteorológicas actuales, ya que tradicionalmente las estaciones de medición cubren grandes extensiones de distancias lo cual no otorga mucha precisión en las mismas a medida que se alejan de la estación, aunado a ello existen zonas que por las condiciones geográficas como la altura o la nula presencia humana no existen como tales puntos de medición que den un buen grado de certeza en las previsiones.

Para ello, es necesario la interconexión de los equipos existentes en las granjas a través de unos servicios que serán implementados y que tendrán acceso a una conexión a internet (ya sea satelital o a través de una conexión a la red celular o conexión internet por un ISP regular) para su exposición.

Específicamente, las mediciones obtenidas de los sensores en las estaciones meteorológicas son enviados en tiempo real a un intervalo especificado a cada modelo y marca del equipo hacia un servicio que actuara como colector y lo almacena en una base de datos local que estará interconectadas con otros pares, y paralelamente se guardara una pequeña metadata como referencia de esta información dentro de un contrato inteligente a fin de poder tener un traza y catalogo en tiempo y procedencia de los registros de medición disponibles.

A su vez, el contrato inteligente contendrá registrado todas las estaciones meteorológicas que participen dentro de esta red, otorgando una propiedad de unicidad y que además no permitan a terceros maliciosos interrumpen o distorsionen la información que puedan existir dentro de la misma. Por otro lado, permite también la trazabilidad e inmutabilidad de los datos ya que el almacenamiento de la información es a través de contratos inteligentes, haciendo de esto fidedigna transparente en acceso a terceros que requieran la información y que a su vez estarán restringidos al acceso solo aquellos que están dentro del contrato e identificados como entidades de solo lectura.

Implementado este nuevo sistema disminuirá el área que cubre una estación meteorológica con relación a un mayor número de ellas conectadas por zona, garantizando con ello que en los lugares alejados existan al menos un punto de medición de referencia, reduciendo la imprecisión que conlleva actualmente.

En resumidas cuentas, el sistema propuesto está basado en blockchain que nos permita recopilar la información y depositarla en un almacenamiento descentralizado, y luego a partir de una aplicación FrontEnd remota que recopila la información periódicamente, los procese y genere una respectiva presentación reducida de la información procesada para un público general, y paralelamente para aquellas entidades que requieran una información completa de las mediciones que están almacenadas en la que incluya toda la metadata disponible, proveerles un acceso el acceso a correspondiente siempre y cuando estén registrados como entidades de solo lectura en el contrato.

2. Diseño

El contrato inteligente contendrá las siguientes especificaciones:

a) Datos: dispondrá de los siguientes registros.

- Lista de estaciones meteorológicas: array con las direcciones e información técnica de todas estaciones meteorológicas que ingresaran datos de mediciones climatológicas.
- Lista de nodos de lectura: array con las direcciones de entidades externas con fines de investigación que tienen acceso a una mayor cantidad de datos existentes en la red.
- Lista de mediciones: array que contiene organizado por fecha y un hash único las respectivas mediciones cargadas en el almacenamiento descentralizado, y además que están asociadas a una estación meteorológica.
- Administrador: contiene la dirección (address) del usuario que despliega el contrato, y es el único que podrá ejecutar funciones sensibles relacionadas al control de las estaciones meteorológicas y nodos de lectura.

b) Funciones: a nivel general contendrá las siguientes funciones.

- `AgregarEstacionMeteorologica`: permite agregar una estación meteorológica con su información técnica.
- `AgregarEnteSoloLectura`: permite agregar una entidad participante para el acceso de los datos directos en el almacenamiento.
- `InsertarMedicion`: permite insertar un registro como identificador que apunte a la nueva medición depositada en el almacenamiento descentralizado.
- `ObtenerTodasMediciones`: permite obtener todas las referencias de las mediciones almacenadas de todas las estaciones meteorológicas.
- `ObtenerMedicion`: permite obtener las referencias de las mediciones almacenadas de una estación meteorológica específica.
- `RemoverEnteSoloLectura`: permite remover una entidad participante para el acceso de los datos directos en el almacenamiento.

c) Reglas de operación: se tienen estipulados las siguientes reglas del negocio a aplicar en el contrato.

- No pueden existir estaciones meteorológicas duplicadas.
- No pueden existir entes de solo lectura duplicadas.
- Solo el propietario del contrato puede agregar estaciones meteorológicas y entes de solo lectura.
- Solo el propietario del contrato puede remover entes de solo lectura.
- Solo las estaciones meteorológicas pueden agregar registros de mediciones.
- Los entes de solo lectura solo pueden obtener las referencias de las mediciones almacenadas.
- No se permite remover ni las estaciones meteorológicas, ni las mediciones ingresadas asociadas a la estación respectiva si la misma es desincorporada.
- El registro de la medición debe contener la fecha y hora de la generación de la misma, así como también mantener un identificador único a la información guardada en el almacenamiento descentralizado.

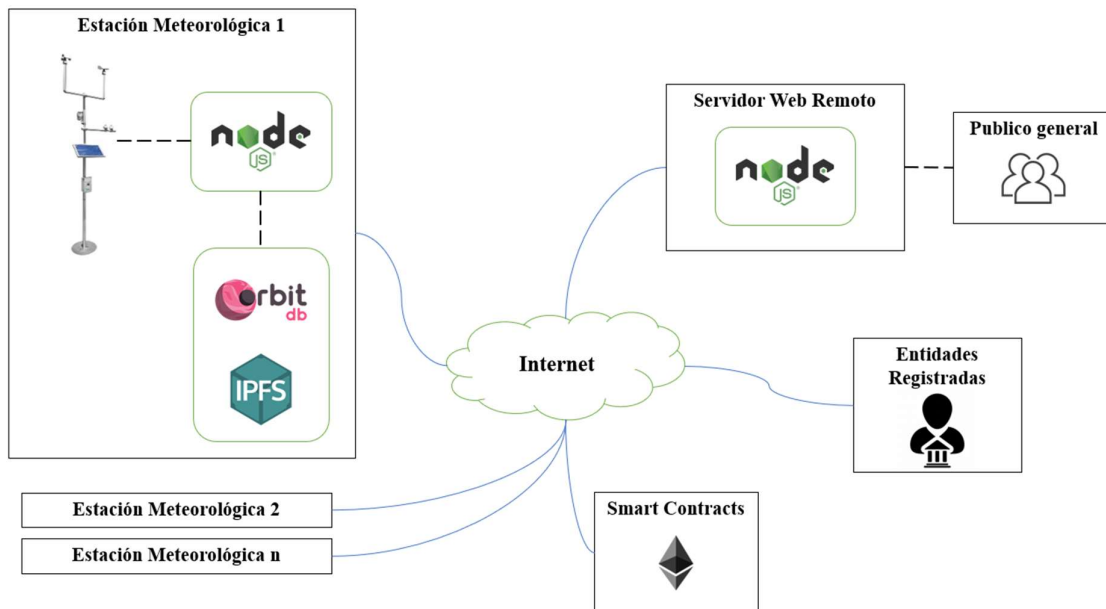
d) Usuarios del sistema: tiene contemplado los siguientes.

- Administrador del contrato inteligente quien crea el contrato y controla el ingreso de las estaciones y las entidades de solo lectura.
- Estaciones meteorológicas instalados en las granjas.
- Entes de solo lectura representados por investigadores, instituciones gubernamentales, etc.

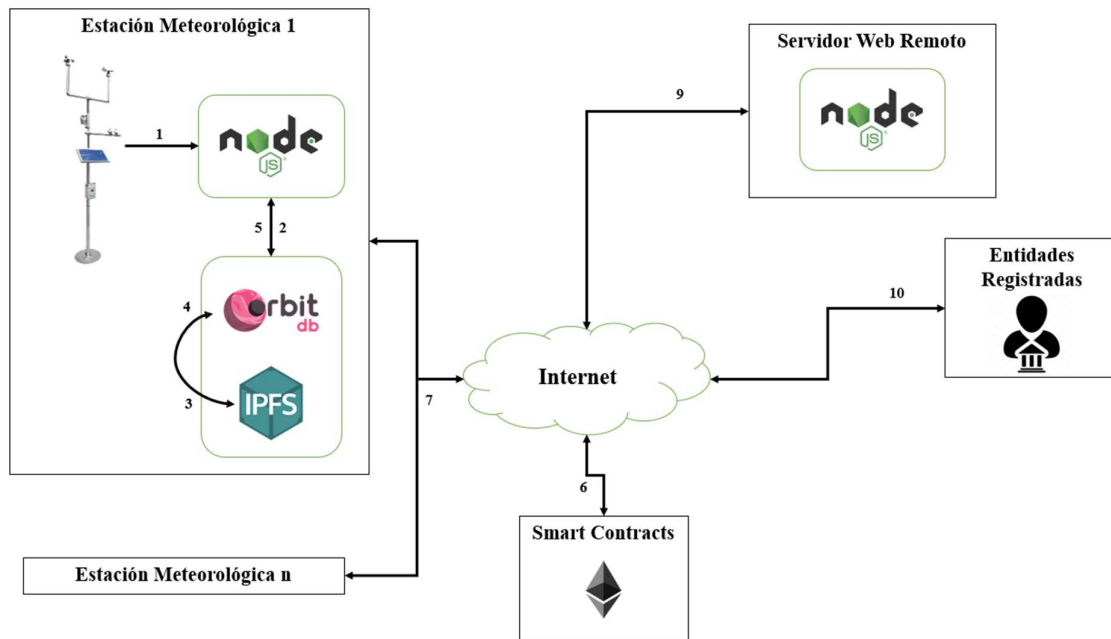
3. Arquitectura

El sistema propuesto contendrá los siguientes componentes:

- Servidor Web remoto: empleado para la muestra de la información de las mediciones meteorológicas a través de una interfaz gráfica de usuario.
- Almacenamiento descentralizado: utilizando IPFS para permitir la transferencia en una red de archivos P2P adecuado a este tipo de sistemas, ya que cada estación meteorológica representaría un nodo cuya información es única y está asociada a ella, estará implementado en conjunto con OrbitDB para sincronizar los datos con otros nodos y poder tener replica de la información por seguridad a fallos en alguno de los nodos o disponibilidad en el acceso al mismo.
- Contrato inteligente: empleado para dar acceso al sistema a las entidades que realizan las mediciones meteorológicas, así como también el registro de las referencias de la información en la red descentralizadas; y para que otros puedan realizar solo lectura de datos con fines de investigación.
- Interfaz de la estación meteorológica: es un aplicativo empleado en las estaciones meteorológicas con el fin de que permita cargar a la red de almacenamiento descentralizado la información pertinente, y el registro de la metadata del mismo en el contrato inteligente.



4. Funcionamiento del sistema



Teniendo en cuenta que los respectivos componentes están operando de manera correcta, el proceso parte con el envío de información proveniente de la estación meteorológica (paso 1), esta es recibida por una interfaz que lo escribe en un archivo csv y posteriormente lo almacena en un nodo IPFS local (paso 2). El IPFS local recibe y guarda el archivo (paso 3), el cual al estar integrado con OrbitDB se inicia un proceso de replicación de datos con otros nodos subscritos mediante el proceso de publicación/suscripción por la nueva entrada y a su vez trae una copia de nuevas entradas de estos nodos (paso 4 y 7). En paralelo, la interfaz ejecuta el método de ingresar la metadata de un nuevo registro relacionado a las mediciones en el respectivo smartContract (paso 6).

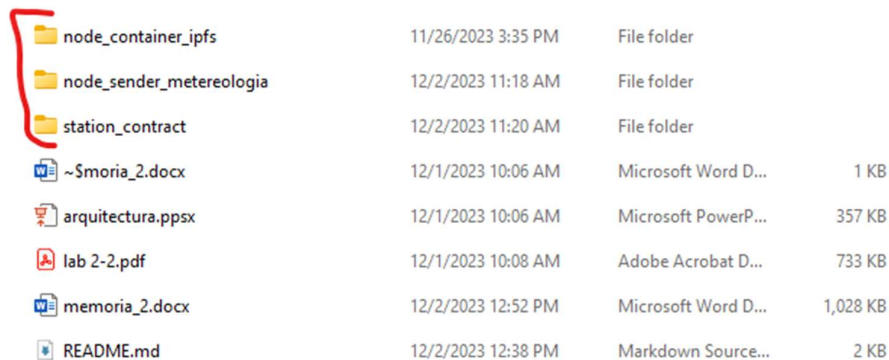
Por otra parte, cuando el servidor web remoto ejecuta la tarea periódica de actualización de datos de medición, este accede al smartContract (paso 9) y toma solamente los registros nuevos, bajo las cuales las procesa y guarda en una BD local el resultado, a fin de reducir la sobrecarga del procesamiento de archivos en cada ejecución de la tarea y que se disponga de datos previamente preparados y procesados para mostrar solo al público general.

Como último punto, las entidades registradas de solo lectura al requerir la información, esta consulta el smartContract para saber los archivos disponibles y la ubicación de ella (paso 10), en la que posteriormente a través de las interfaces correspondientes implementadas por ellos pueden obtenerlas.

5. Implementación del laboratorio

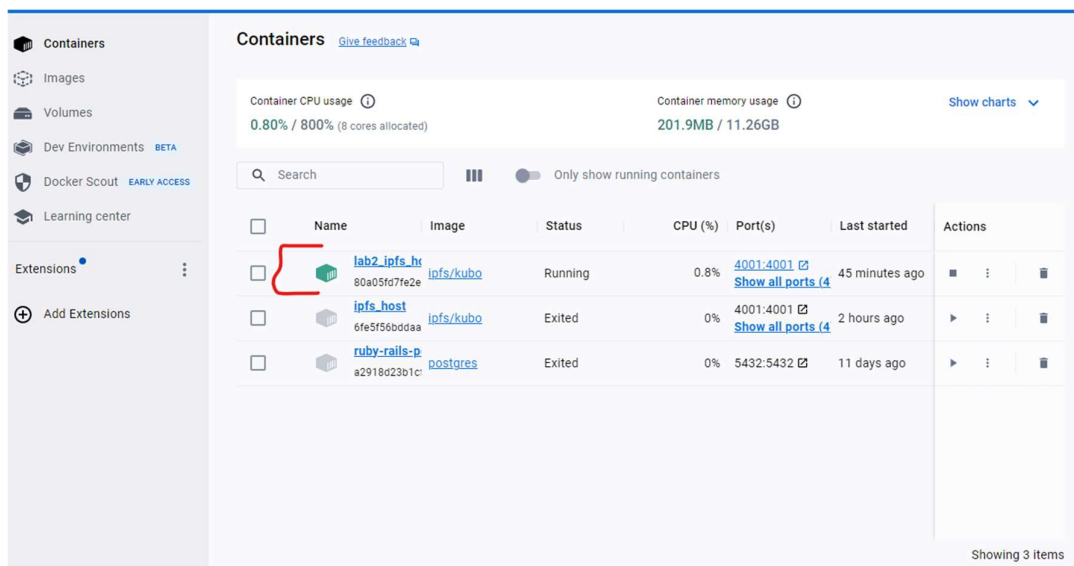
La realización del ejercicio 2 fue implementado en una maquina bajo sistema operativo Windows 11, en la cual, se recalca que existirán variaciones en el despliegue del mismo con respecto al instructivo dado en clases, dado al histórico de dificultades indicadas en la memoria del ejercicio 1, como lo son el problema de CORS tanto a nivel del dominio y en los métodos. La carpeta que contiene está disponible en https://github.com/ff-repo/bc_udc_labs/tree/main/lab2/ejercicio_2

Así pues, a continuación tal como se muestra en la siguiente imagen, las tres carpetas indicadas corresponden a la solución a entregar, y en el mismo orden en que se muestran, node_container_ipfs contiene las carpetas compartidas con el container en docker de la aplicación IPFS, node_sender_meteorologia corresponde a un aplicativo FrontEnd basado en React para permitir la carga de archivos al container de IPFS, y por ultimo dentro de la carpeta station_contract se dispone del .json con la especificacion de la interfaz con IPFS y el contrato inteligente empleado para el despliegue del IPFS, aunado a lo anterior el archivo README.md contiene los comando empleados para el despliegue de cada uno de los componentes (IPFS container y aplicación FrontEnd)



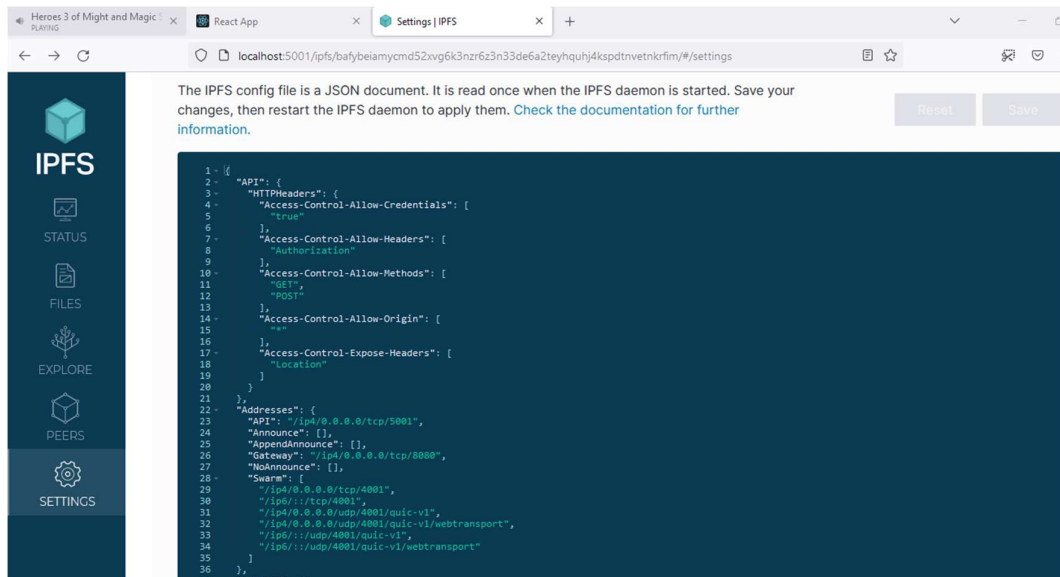
node_container_ipfs	11/26/2023 3:35 PM	File folder	
node_sender_meteorologia	12/2/2023 11:18 AM	File folder	
station_contract	12/2/2023 11:20 AM	File folder	
~\$morla_2.docx	12/1/2023 10:06 AM	Microsoft Word D...	1 KB
arquitectura.ppsx	12/1/2023 10:06 AM	Microsoft PowerP...	357 KB
lab 2-2.pdf	12/1/2023 10:08 AM	Adobe Acrobat D...	733 KB
memoria_2.docx	12/2/2023 12:52 PM	Microsoft Word D...	1,028 KB
README.md	12/2/2023 12:38 PM	Markdown Source...	2 KB

Empleando los comandos indicados dentro del README.md se despliega un container de IPFS como se muestra en la siguiente imagen.

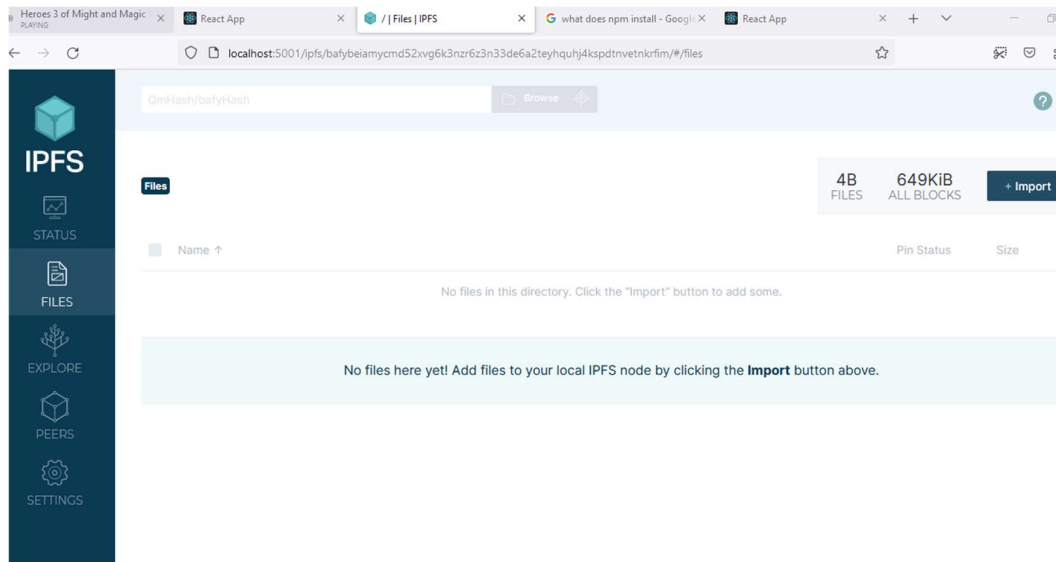


Containers Give feedback							
Container CPU usage [ⓘ]				Container memory usage [ⓘ]		Show charts	
0.80% / 800% (8 cores allocated)				201.9MB / 11.26GB			
<input type="text" value="Search"/> <input type="checkbox"/> Only show running containers							
<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	lab2_ipfs_hc 80a05fd7fe2e	ipfs/kubo	Running	0.8%	4001:4001 🔗 Show all ports (4)	45 minutes ago	🛑 🔗 🗑️
<input type="checkbox"/>	ipfs_host 6fe5f5ebddaa	ipfs/kubo	Exited	0%	4001:4001 🔗 Show all ports (4)	2 hours ago	▶ 🔗 🗑️
<input type="checkbox"/>	ruby-rails-p a2918d23b1c	postgres	Exited	0%	5432:5432 🔗	11 days ago	▶ 🔗 🗑️
Showing 3 items							

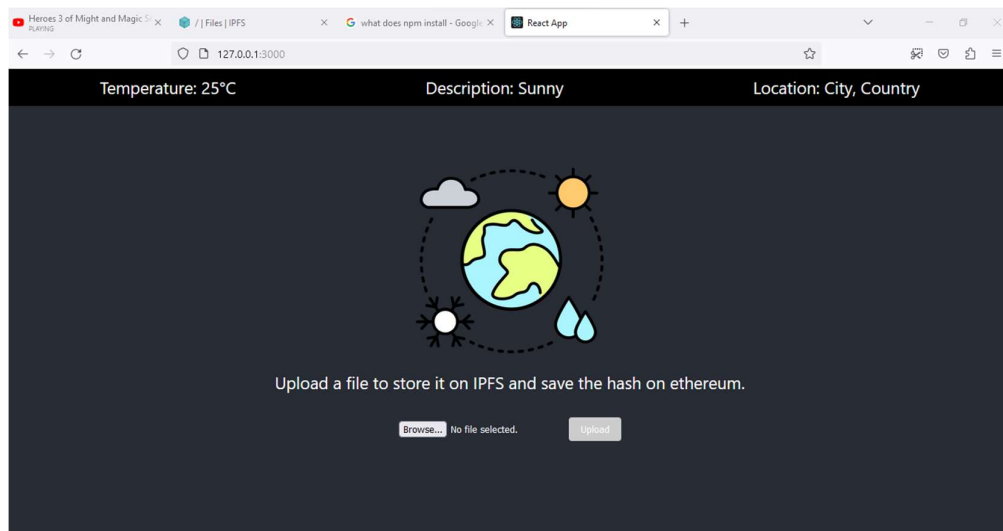
Los permisos en relación a CORS (no recomendados en un ambiente de producción), se establecieron de esta forma:



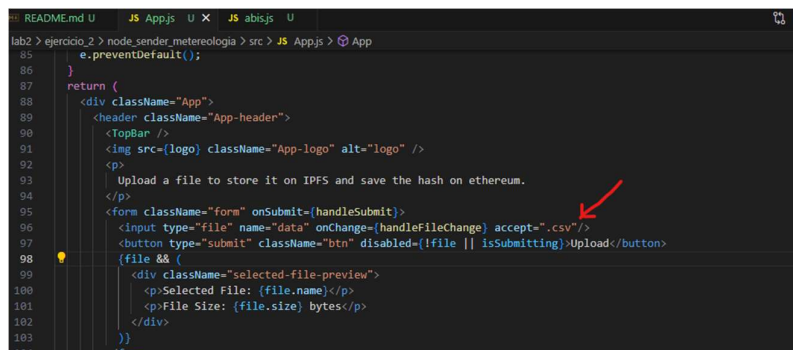
En esta otra imagen se puede apreciar que el IPFS no contiene archivos.



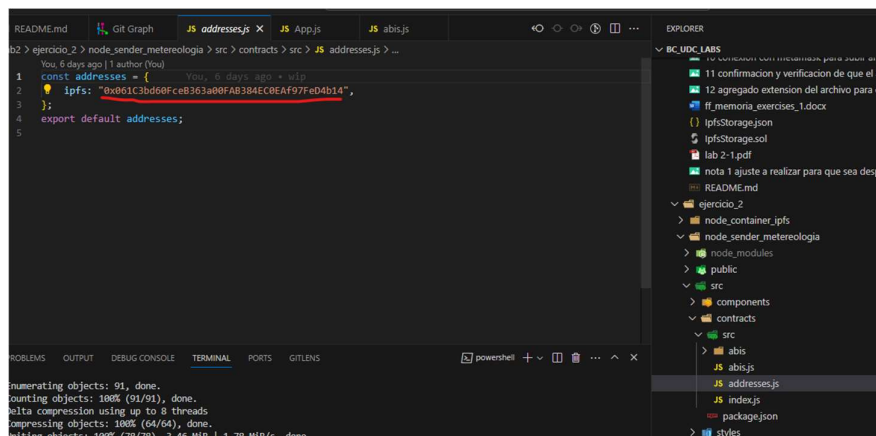
El despliegue de la aplicación FrontEnd se muestra en la siguiente vista.



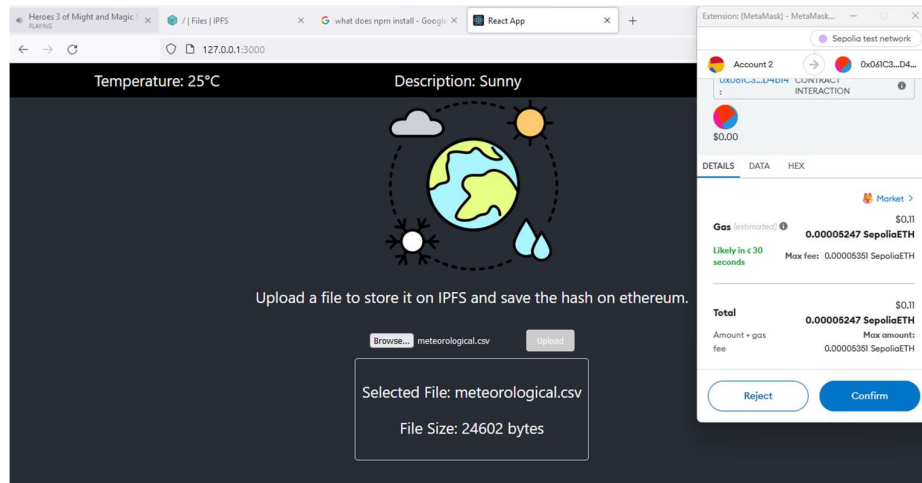
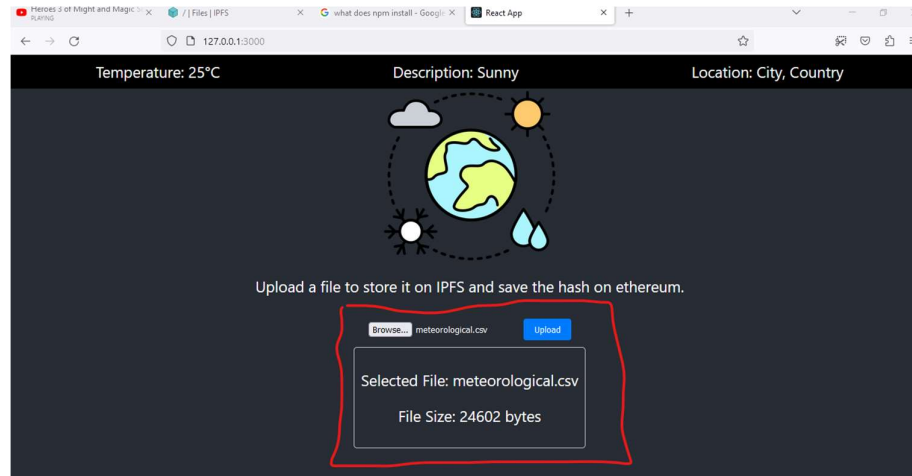
Dado que los datos enviados por una estación meteorológica es mejor almacenarlo en forma de CSV, se implementó un filtro para permitir solo dicho tipo de archivo, como se observa en la siguiente imagen.



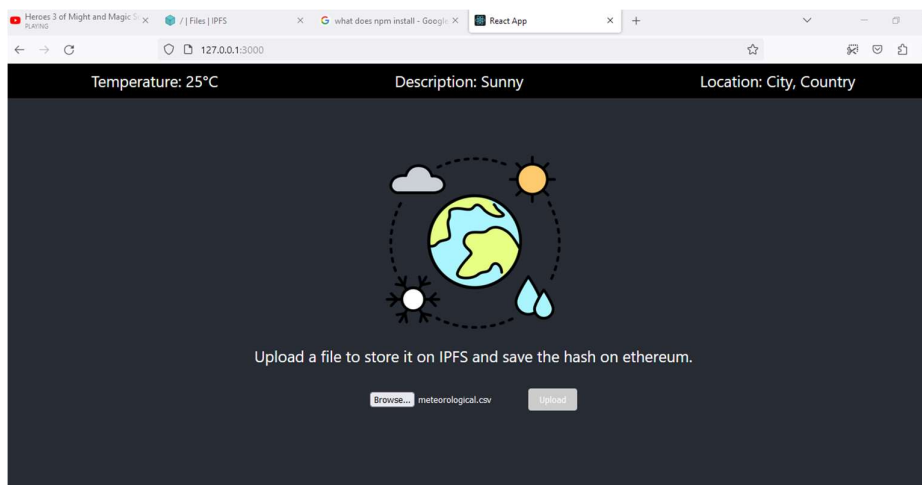
Se debe tener en cuenta reemplazar la dirección por el obtenido en el despliegue del contrato tal como fue realizado en el ejercicio 1 del laboratorio 2.



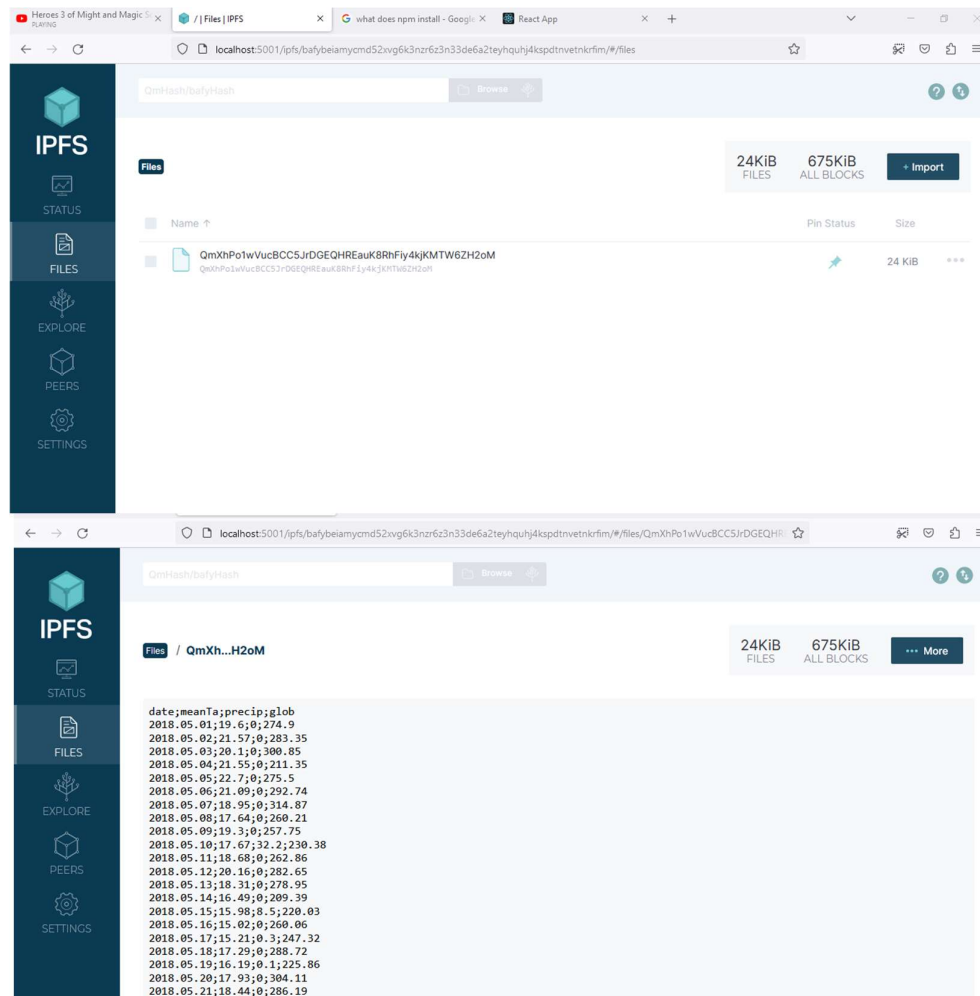
Siguiendo, realizamos una prueba seleccionando un archivo CSV modelo para cargarlo al IPFS.



Una vez aceptada la operación, después de la subida del archivo se reinician los campos del FrontEnd.



Mientras que en el IPFS container se puede apreciar el archivo cargado y su contenido.



6. Referencias

<https://www.darrera.com/wp/es/producto/3r-aws200-estacion-meteorologica-agricultura/>

<https://www.maherelectronica.com/sensores-agricolas-clima/estacion-meteorologica-para-agricultura/>