

Memoria - PTI

Javier Armaza Bravo / @estudiantat.upc.edu
Sergi Colomer Segalés / @estudiantat.upc.edu
Sofía López Olivares / @estudiantat.upc.edu
Pablo Rosas Roda / @estudiantat.upc.edu

Marzo del 2024

ÍNDICE

1. Resumen propuesta	3
2. Objetivos tecnológicos del proyecto y trabajos existentes	4
2.1. Objetivos tecnológicos del proyecto	4
2.2. Trabajos existentes/punto de partida	4
3. Propuesta de proyecto	5
3.1. Descripción propuesta	5
3.2. Explicar trabajo a hacer, desglosar trabajos en varios paquetes de trabajo	6
3.3. Explicar relación entre los paquetes de trabajo	7
3.4. Plan de trabajo (esquema temporal (Gantt))	8
3.5. Resultado que se espera obtener	8
4. Relación de la propuesta con la asignatura	9
5. Potencial de innovación	9
6. Paquetes de trabajo	10
7. Grupo del proyecto	12
8. Referencias	13

1. Resumen propuesta

La idea general es hacer un sistema de almacenamiento, como los que ya existen hoy en día como por ejemplo Drive de Google, pero con diferencias que hagan que quieras usar nuestro producto y no el de los demás .

Queremos crear un servicio de almacenamiento de datos privado con almacenamiento en nodos distribuidos para garantizar la disponibilidad de los datos en todo momento. A estos datos se podrá acceder a través de un portal en el que tendrás que confirmar tu identidad con unas credenciales. Una vez dentro podrás consultar, subir y/o bajar todo tipo de archivos, estos mismos archivos se podrán hacer públicos a través de una funcionalidad de "compartir" y podrás hacer un seguimiento de la popularidad de tu archivo, es decir podrás consultar el número de veces que otros usuarios han accedido a tu archivo para consultarlo y/o bajarlo.

2. Objetivos tecnológicos del proyecto y trabajos existentes

2.1. Objetivos tecnológicos del proyecto

- Crear un servidor de disco descentralizado utilizando un cluster privado de IPFS para gestionar nodos de almacenamiento.
- Dockerizar los nodos del IPFS Cluster para facilitar su control a través de Kubernetes, permitiendo la automatización de la creación y destrucción de nodos.
- Desarrollar una página web con autenticación de usuarios para que estos puedan acceder al servicio y gestionar sus archivos.
- Implementar el front-end de la página web utilizando React, y el back-end para el control de credenciales con Node.js y MongoDB.
- Configurar un servidor web con Nginx para servir la página web y comunicar las diferentes partes del proyecto, como el back-end y la gestión de Kubernetes.
- Investigar y potencialmente implementar la tecnología blockchain para registrar la popularidad de archivos compartidos entre usuarios.

2.2. Trabajos existentes/punto de partida

- Investigación previa sobre IPFS y su implementación en clusters privados.
- Estudio de Docker y Kubernetes para la dockerización y gestión automatizada de los nodos.
- Experiencia previa en el desarrollo web utilizando React, Node.js y MongoDB para el front-end y back-end respectivamente.
- Conocimiento de Nginx y su configuración como servidor web.
- Estudio de la tecnología blockchain y su aplicación en el registro de la popularidad de archivos compartidos.

Además, buscando trabajos existentes se ha encontrado lo siguiente que tienen relación con el proyecto:

En el apartado de [Referencias](#) están listados todos los artículos que hemos encontrado del desarrollo sobre la temática del proyecto.

3. Propuesta de proyecto

3.1. Descripción propuesta

Se propone como proyecto la creación de un servidor de disco descentralizado, es decir la creación de un sistema que permita gestionar el almacenamiento de datos que no se guardan en un solo lugar, sino que se distribuyen entre varios dispositivos en una red.

El sistema se desarrolla mediante un cluster privado de IPFS, con el objetivo de mantener un control total sobre todos los nodos dedicados al almacenamiento. Esta configuración nos permite garantizar redundancia al tener el contenido replicado y proporcionar espacio adicional de almacenamiento al distribuir la información en diversos nodos. Además, cada uno de estos nodos será dockerizado para facilitar su gestión mediante Kubernetes, esto con la finalidad principal de automatizar en cierta medida la creación y destrucción de nodos. Dependiendo de si se quiere redundancia o almacenamiento, se tendrá que replicar contenedores en ejecución para no perder datos o crear nodos nuevos a partir de una imagen de partida.

Los clientes podrán acceder al sistema a través de una página web que contará con autenticación de usuarios. Por lo tanto, además de implementar un sistema de inicio de sesión con credenciales, será necesario desarrollar la funcionalidad para almacenar los archivos asociados a cada cuenta de usuario. Para llevar a cabo esta tarea, el front-end se desarrollará con React para la parte gráfica, mientras que para el control del sistema de credenciales se utilizarán Node.js y MongoDB.

Para habilitar esta página web se establecerá un servidor web utilizando Nginx. Este servidor también funcionará como un nodo central para comunicar las distintas áreas de trabajo del proyecto, actuando como intermediario entre el backend de la base de datos y la gestión de Kubernetes para proporcionar al usuario la información requerida.

Además, si el tiempo lo permite, se implementará tecnología blockchain para registrar la popularidad de un archivo cuando se comparte con otros usuarios. Esto permitirá mantener un registro transparente y seguro de la actividad de intercambio de archivos en la plataforma.

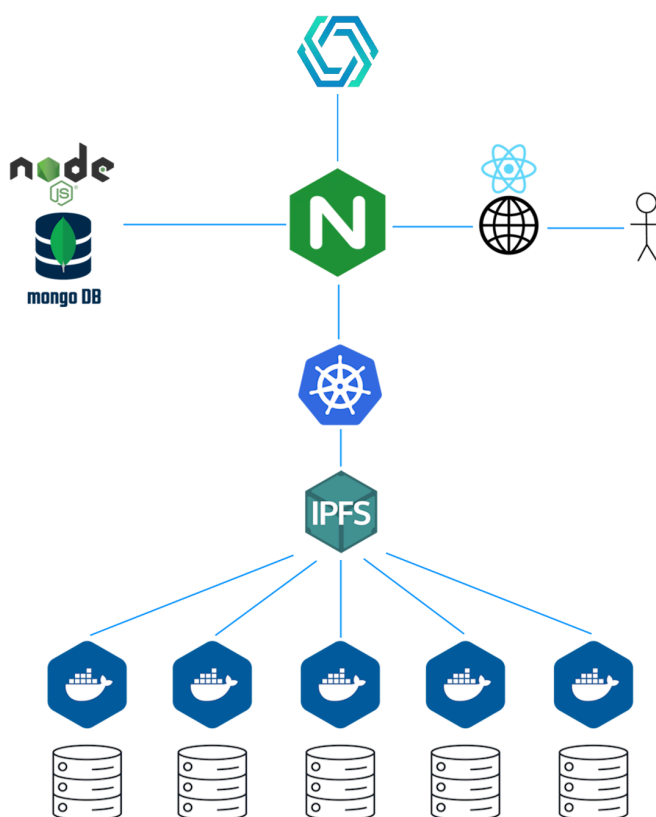
3.2. Explicar trabajo a hacer, desglosar trabajos en varios paquetes de trabajo

La división del trabajo se ha desglosado en 5 paquetes de trabajo, lo cuales son:

1. Creación de los nodos e interacción de los nodos de IPFS
En este paquete se realizará la configuración y despliegue de nodos IPFS en un cluster privado para almacenamiento descentralizado, garantizando sincronización y redundancia de datos.
2. Uso de NginX para separar las cargas de trabajo
Se desarrolla Nginx como servidor web para dirigir el tráfico entre partes del sistema, como la página web, servidor de autenticación y nodos IPFS.
3. Creación del servidor de autenticación
Tal cual como se menciona, se desarrollará un servidor para gestionar la autenticación de usuarios, con manejo de credenciales y verificación de identidad antes de acceder al sistema.
4. Creación del FrontEnd
En este punto se desarrollará la interfaz de usuario utilizando React para permitir a los usuarios autenticados acceder y gestionar sus archivos e interactuar con otras partes del sistema.
5. Implementación de contenedores con Kubernetes.
Se implementarán contenedores con Kubernetes para distribuir la carga de trabajo en los nodos de IPFS, garantizando disponibilidad y rendimiento óptimo del sistema.
6. Creación de un sistema de popularidad de archivos.
Se implementará un sistema de popularidad de archivos basado en blockchain.

3.3. Explicar relación entre los paquetes de trabajo

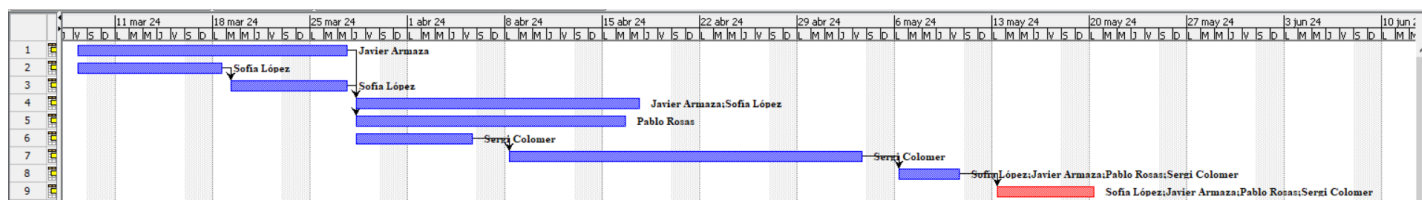
Los paquetes de trabajo están estrechamente interrelacionados para garantizar el funcionamiento eficiente del sistema de almacenamiento descentralizado. La creación y configuración de los nodos IPFS proporciona la base para la distribución y redundancia de datos. El servidor Nginx actúa como punto central para dirigir el tráfico entre las diferentes partes del sistema, asegurando una comunicación fluida entre el frontend, el servidor de autenticación y los nodos IPFS. El servidor de autenticación es esencial para la seguridad y gestión de usuarios, interactuando con el servidor Nginx para gestionar el acceso a los recursos. El frontend desarrollado con React permite a los usuarios interactuar con el sistema, comunicándose con el servidor de autenticación y el servidor Nginx para acceder a los recursos necesarios. Por último, la implementación de contenedores con Kubernetes garantiza la escalabilidad y gestión eficiente de los recursos del sistema, distribuyendo la carga de trabajo en los nodos IPFS para mantener su disponibilidad y rendimiento óptimo.



Esquema conceptual de las relaciones entre las tecnologías y la interacción del usuario.

3.4. Plan de trabajo (esquema temporal (Gantt))

		Nombre	Duracion	Inicio	Terminado	Predecesores	Nombres del Recurso
1		Servidor Nginx	14 days	08-03-24 8:00	27-03-24 17:00		Javier Armaza
2		Base de datos	7 days	08-03-24 8:00	18-03-24 17:00		Sofía López
3		Back End	7 days	19-03-24 8:00	27-03-24 17:00	2	Sofía López
4		frond	15 days	28-03-24 8:00	17-04-24 17:00	3	Javier Armaza;Sofía López
5		IPFS	14 days	28-03-24 8:00	16-04-24 17:00	1	Pablo Rosas
6		Docker	7 days	28-03-24 8:00	05-04-24 17:00		Sergi Colomer
7		Kubernetes	20 days	08-04-24 8:00	03-05-24 17:00	6	Sergi Colomer
8		Conexion	5 days	06-05-24 8:00	10-05-24 17:00	7	Sofía López;Javier Armaza;...
9		Blockchain	5 days	13-05-24 9:00	20-05-24 9:00	8	Sofía López;Javier Armaza;...



3.5. Resultado que se espera obtener

Esperamos obtener un sistema de almacenamiento de datos funcional que use todas las tecnologías previamente mencionadas, así como que sea fácilmente escalable por si se necesita almacenar más cantidad de datos. Al final el resultado esperado sería que el usuario sea capaz de almacenar documentos y otros tipos de archivo, como podría ser una imagen. También deberá tener la capacidad para poder consultar y/o bajar este archivo en cualquier momento que lo desee. Finalmente se espera una funcionalidad adicional que permita al usuario saber la cantidad de veces que se ha consultado un archivo compartido por este.

4. Relación de la propuesta con la asignatura

La propuesta de proyecto cuenta con tecnologías enseñadas en las sesiones de laboratorio. Entre las cuales se encuentran Docker, Kubernetes y posiblemente Blockchain, por lo que la propuesta del proyecto abarca varios aspectos de la asignatura, desde el desarrollo de aplicaciones web hasta la implementación de tecnologías emergentes como blockchain y la utilización de contenedores para la gestión de infraestructura.

5. Potencial de innovación

Con este proyecto queremos crear un sistema de almacenamiento distribuido en nodos, que sea resiliente y que sea expandible gracias a una arquitectura que permita crecer. Al realizarse el almacenamiento en nodos de IPFS en vez de directamente en discos, si queremos aumentar la capacidad de la red, en vez de tener que implementar un sistema el cual se tiene que reconfigurar manualmente, solamente se tendrá que desplegar la imagen de IPFS en la máquina. Al desplegarse automáticamente se sincronizará con el resto de los nodos, y el equipo en ese momento estará disponible para ser usado. Esto nos puede permitir adaptarnos a la demanda de forma elástica. IPFS ha sido diseñado para redes de muchos elementos, lo que hace que no tenga problemas para ser escalada. Además, al usar IPFS Cluster, tendremos respaldo de los archivos, haciendo que el sistema sea a prueba de errores.

Estas características nos permiten poder ofrecer una alta disponibilidad de forma eficiente, y una escalabilidad rápida, lo que nos permite mejorar nuestras condiciones que ofrecemos al cliente, pudiendo reclamar un pago mayor por el servicio. Además, al poder escalar rápidamente, podemos hacer compra de equipamiento con menor antelación, facilitando las previsiones de obtención de equipamiento.

Estas características no serán demasiado interesantes para el usuario, ya que la tecnología y esas características no suelen ser los puntos de interés cuando el usuario se está decidiendo entre dos productos, pero puede ser un sistema muy interesante para proveedores de servicios de almacenamientos, ya que puede pasar que el sistema propuesto les sea más útil que el sistema actual que tengan.

6. Paquetes de trabajo

Tal y como se ha explicado previamente en el apartado [3.2](#), los paquetes de trabajo que forman nuestro proyecto son los siguientes:

PT1 - Creación de los nodos e interacción de los nodos de IPFS

En este primer paquete tenemos la intención de tener diferentes nodos de almacenaje en un sistema descentralizado como es el de IPFS. Por lo que el punto de partida es nulo y como objetivo esperamos tener el sistema funcionando de forma local.

Para ello utilizaremos la implementación de IPFS Cluster para poder tener los nodos controlados en una red local.

Esta tecnología implica la necesidad de dockerizar cada nodo para poder ser utilizado por el cluster.

PT2 - Uso de NginX para separar las cargas de trabajo

En este segundo paquete de trabajo tenemos la intención de tener un sistema más cercano al funcionamiento final, por eso, crearemos un servidor web el cual se comunicará con nuestra previa implementación de IPFS. Esto nos permitirá acceder a los recursos de forma remota aunque limitada ya que, en este punto, todavía no tendremos autenticación de usuarios ni interfaz gráfica.

A medida que avance el proyecto, Nginx se encargará de manejar la comunicación entre los diferentes sectores de trabajo.

PT3 - Creación del servidor de autenticación

Este punto de trabajo está estrechamente relacionado con el punto anterior. Aquí lo que se realizará es un sistema de login con diferentes usuarios para poder acceder de una forma más cómoda a los diferentes recursos.

Para poder realizar esta implementación se utilizará una base de datos que crearemos y gestionaremos con Node.js y MongoDB.

PT4 - Creación del FrontEnd

En este paquete de trabajo se seguirá mejorando el de los dos anteriores. El objetivo propuesto es realizar todo lo que es la interfaz gráfica de la aplicación, tanto del login como de la gestión de archivos una vez entrado al servidor. Por este motivo, este paquete de trabajo se irá realizando paralelamente a los siguientes.

PT5 - Implementación de contenedores con Kubernetes.

En este paquete esperamos poder mejorar el sistema de IPFS creado inicialmente. Para ello utilizaremos kubernetes, el cual nos permitirá gestionar los diferentes nodos de almacenamiento, de la misma forma que crear nuevos y destruir los ya existentes en caso de ser necesario.

Por lo que el objetivo de este paquete de trabajo será conseguir que nuestro sistema sea fácilmente escalable y ampliable a nuestro gusto.

PT6 - Creación de un sistema de popularidad de archivos.

Este último paquete de trabajo se centrará en la creación de un sistema que sea capaz de almacenar la popularidad de un archivo en concreto, ya sea por la cantidad de veces que este se ha compartido, la cantidad de visualizaciones o la cantidad de descargas.

Esto lo realizaremos con la tecnología blockchain, la cual nos permitirá almacenar en todos los nodos el autor del fichero y la popularidad que ha generado.

Cada paquete de trabajo cuenta con su respectiva comprobación de los servicios ofrecidos y la implementación realizada para asegurar su correcto funcionamiento a lo largo del desarrollo de todo el proyecto.

7. Grupo del proyecto

7.1. Rol de los participantes

Paquete de trabajo	Javier	Sergi	Sofía	Pablo
PT1 (IPFS)		Ayudante		Responsable
PT2 (NginX)	Responsable			
PT3 (BBDD)			Responsable	
PT4 (FrontEnd)	Responsable		Ayudante	
PT5 (Kubernetes)	Ayudante	Responsable		
PT6 (Blockchain)	Ayudante	Ayudante	Responsable	Ayudante

7.2. Organización y gestión del proyecto

El grupo se reunirá para poder compartir los avances y discutir los problemas de forma semanal. Los miembros del grupo nos hemos reservado una franja horaria cada domingo para este propósito.

Para la correcta organización y planificación del proyecto contamos con un repositorio en GitLab y con un servidor de Discord, por el cual se realizarán las reuniones y se podrán comentar los diferentes puntos del proyecto en diferentes hilos de conversación para tenerlo todo más organizado.

8. Referencias

Deploy IPFS Cluster with Kubernetes

Tipo Página Web

Resumen In my previous post I have discussed about deploying IPFS Cluster with docker. In this post I'm gonna discuss about deploying IPFS Cluster with Kubernetes. All the deployments which related to this post available in gitlab. Please clone the repo and continue the post.

URL <https://medium.com/rahasak/deploy-ipfs-cluster-with-kubernetes-c4cd8d64b7c8>

Set up server infrastructure with IPFS Cluster

Tipo Página Web

Resumen If you want to install IPFS in a server environment and offer IPFS as a service, you should look at IPFS Cluster as a way to scale your IPFS deployment beyond a single IPFS daemon. IPFS Cluster provides data orchestration across a swarm of IPFS daemons by allocating, replicating, and tracking a global pin-set distributed among multiple peers. This makes it significantly easier to manage multiple IPFS nodes and ensure that data is available across an internal network.

URL <https://docs.ipfs.tech/install/server-infrastructure/>

Deploy a private IPFS network in 5 steps

Tipo Página Web

Resumen IPFS (InterPlanetary File System) is a peer-to-peer distributed file system that stores and retrieves files in a BitTorrent-like way. The default installation of IPFS connects your machine to the global distributed network. In some cases (privacy, confidentiality) a private IPFS network is preferred over connecting to the public IPFS network.

URL https://medium.com/@s_van_laar/deploy-a-private-ipfs-network-on-ubuntu-in-5-steps-5aad95f7261b

Run IPFS in a Docker container**Tipo** Página Web

Resumen In recent years, Docker and a few other projects have redefined how we run server applications. In the future, we might be running containerized apps in our personal devices. At its core, this fast-paced improvement is a combination of good interfaces to standardize how to do things, and great tooling to make using containers easy.

URL <https://blog.ipfs.tech/1-run-ipfs-on-docker/>

Running an IPFS node as a Docker container on a private network**Tipo** Página Web

Resumen IPFS (InterPlanetary File System) is a decentralized storage system designed to store data in a way that is distributed, with any desired degree of redundancy, with a capability of isolating a random segment of the network without use of functionality. The integrity of data is cryptographically protected. And that makes it a highly promising technology.

URL <https://medium.com/@borepstein/running-an-ipfs-node-as-a-docker-container-on-a-private-network-4c0186aeca61>