



IPFS, la nueva web resiliente

Administración de Sistemas Informáticos en Red

3 de Abril - 29 de Mayo del 2020

Nicky Kennedy Rufino Correa



Índice

Propuesta De Trabajo	5
Proyecto IPFS	5
Objetivo Principal	5
Investigación	5
Implementación	5
Objetivo Secundario	5
Descripción General Del Objeto	6
¿Qué es IPFS?	6
Cómo funciona IPFS	7
Direccionamiento de contenidos	8
Lib2p2	9
Intercambio	9
Routing	9
Network	9
MFS	10
Gráficos acíclicos dirigidos (DAG)	11
Tablas hash distribuidas (DHT)	12
IPLD	12
¿Qué es IPNS?	13
Cómo funciona IPNS	14
¿Que es IPFS Cluster?	16
Cómo funciona IPFS-Clúster	16
Descripción Técnica De La Solución Adoptada	17
Red IPFS + IPNS Privada & IPFS-Cluster	17
Funcionamiento IPFS & IPFS-Cluster	18
IPFS	18
Comandos Básicos	19
ipfs init	19
ipfs add	19
ipfs cat	20
ipfs get	20
ipfs ls	20
ipfs refs	20
Comandos De Estructura De Datos	21



ipfs object	21
ipfs files	21
Advanced Commands	22
ipfs pin	22
ipfs repo	22
IPFS-Cluster	23
Comandos útiles	23
Plan De Instalación y Configuración	24
Instalacion Windows	24
GO	24
IPFS	24
Instalación Linux	25
Git	25
GO	25
IPFS	26
IPFS Cluster	27
Opcional	27
WebUI	27
Configuración	28
Red IPFS Privada	28
Swarm Key	28
Bootstrap	29
Systemd	30
IPFS-Cluster	31
Configuración	31
Systemd	33
Problemas Encontrados Y Solución Adoptada	33
Instalación IPFS vía comandos PowerShell	33
Problemas de perfil PowerShell	33
Solución	34
Problemas al abrir PowerShell	34
Solución	34
Conclusión	35
Puntos positivos	35
Puntos negativos	35
A mejorar	35
Anexo	36



Definiciones	36	
B	36	
F	36	
H	36	
I		36
K		36
P		36
S		37
U		37
Bibliografía		37
IPFS		37
IPLD & IPNS & DNSLink		38
IPFS Cluster		38
Otros links interesantes		38
FCT		39
Bloque 1		39
Empresa		39
Departamentos		39
Rol En La Empresa		39
Objetivos		40
Bloque 2		40
Actitud profesional		40
Carrera profesional		41
Retos y etapas		41
Proyección laboral		41
Bloque 3		42
Desarrollo		42
Ámbitos		42
Beneficios a la empresa		43



Propuesta De Trabajo

Proyecto IPFS

En este proyecto investigaremos cómo funciona una red descentralizada como IPFS y cuáles son sus beneficios y desventajas.

La idea central es instalar una red IPFS en una subred para poder explotar las posibilidades que nos puede brindar al tener un sistema descentralizado en una zona de trabajo cooperativo.

Objetivo Principal

Este proyecto me permitirá investigar no sólo cómo funciona una red descentralizada sino también cómo funciona en conjunto con Hashes y Blockchain ampliando así mis conocimientos en transacciones en línea de forma segura y anónima.

Consta de dos partes:

- Investigar el funcionamiento de IPFS.
- Implementar IPFS en una subred privada.

Investigación

En la fase de investigación no sólo debo aprender cómo funciona IPFS sino también cuáles son las especificaciones técnicas de la implementación en una subred y si es posible implementarla correctamente.

Implementación

Pondré en práctica lo investigado para poder verificar si su implementación sería exitosa o no. En caso de que no fuera exitosa tendría que investigar una solución o cambiar la dirección del proyecto para que pueda adaptarse a la situación.

Objetivo Secundario

Como objetivo secundario el objetivo es investigar la implementación de los proyectos de los compañeros de curso a la red IPFS, haciendo que su página web esté disponible para su consulta en la nueva red. Además no sólo implementar los demás proyectos sino que investigar su viabilidad e integridad.

Descripción General Del Objeto

¿Qué es IPFS?

Primero que nada, buenos días. **IPFS** es un **protocolo** que funciona como un sistema distribuido para almacenar y acceder a archivos, sitios web, aplicaciones y datos que hacer posible descargar un archivo desde muchas ubicaciones que no son administradas por una organización.

En un caso común cuando deseamos hacer una consulta se hace a un servidor central en la cual la **URL** se vería de este tipo:

```
https://en.wikipedia.org/wiki/ejemplo
```

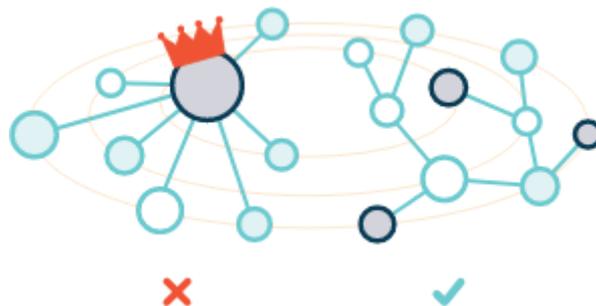
Con IPFS en vez de la petición hacer a un servidor central la petición se hace sobre otros ordenadores que contengan la información en la cual la URL se vería de esta forma:

```
/ipfs/QmXoypizjW3WkLwHCnL72vedjQkDDP1mXWo6uc/wiki/ejemplo.html
```

Esa mezcla de letras después de **"/ipfs/"** se llama un **identificador de contenido** (CID), en lugar de estar basado en la ubicación IPFS direcciona un archivo por lo que hay en él o por su contenido. La información se busca por el identificador anterior, el cual es un **hash** criptográfico del contenido de esa dirección. El hash es único para el contenido del que procede haciendo que te permita verificar que tienes lo que pediste.

Cuando usas IPFS no solo descargas archivos de otra persona, tu computadora también ayuda a distribuirlos haciendo que tenga las siguientes ventajas:

- Soporta un Internet resistente.
- Hace que sea más difícil censurar el contenido.
- Puede acelerar la web cuando está lejos o desconectado.



Cómo funciona IPFS

IPFS es una red de almacenamiento **punto a punto (p2p)**. El contenido es accesible a través de pares, ubicados en cualquier parte del mundo, que podrían transmitir información, almacenarla o hacer ambas cosas. IPFS sabe cómo encontrar lo que pide a través de su dirección de contenido, en lugar de su ubicación.

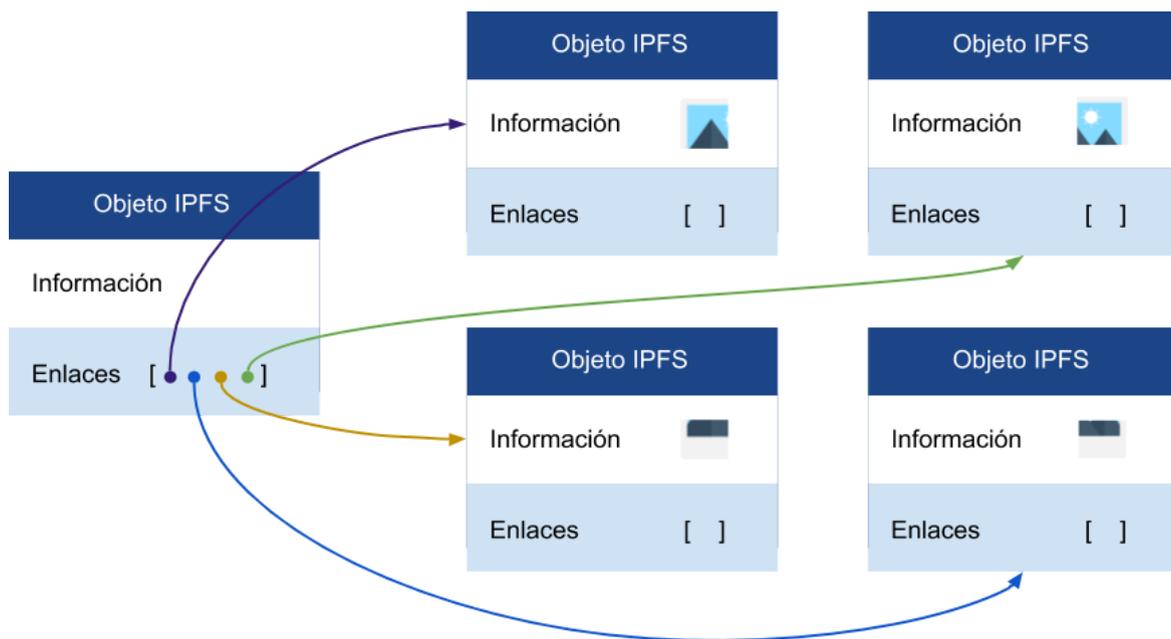
Hay tres principios fundamentales para entender el IPFS:

- Identificación única a través del **direccionamiento de contenidos**
- Vinculación de contenido a través de **gráficos acíclicos dirigidos (DAG)**
- Detección de contenido a través de **tablas hash distribuidas (DHT)**

Los datos se guardan dentro de lo que se llama objeto IPFS el cual puede almacenar un máximo de 256 **kB** de información, y en el cual también soporta links.

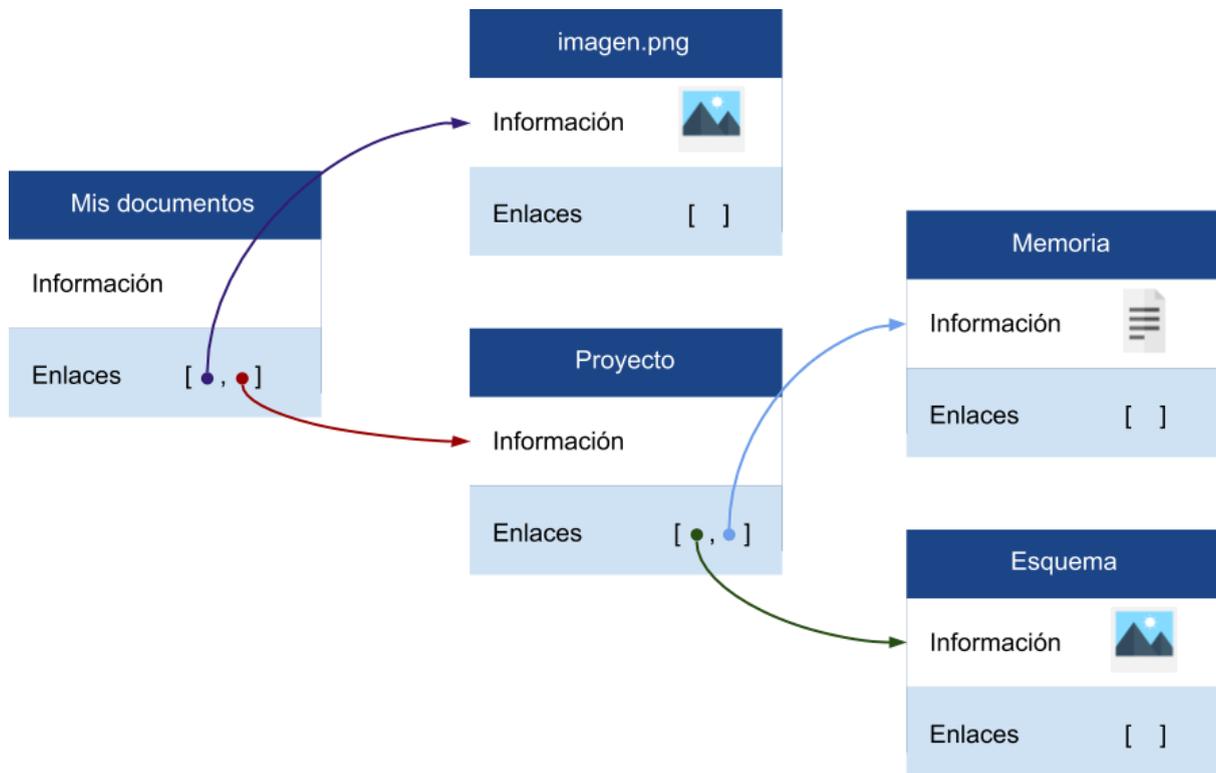


En el caso de que la información sea demasiado grande lo que se hace es compartimentar la información en otros objetos IPFS el cual apuntan a un nuevo objeto, el nodo principal.

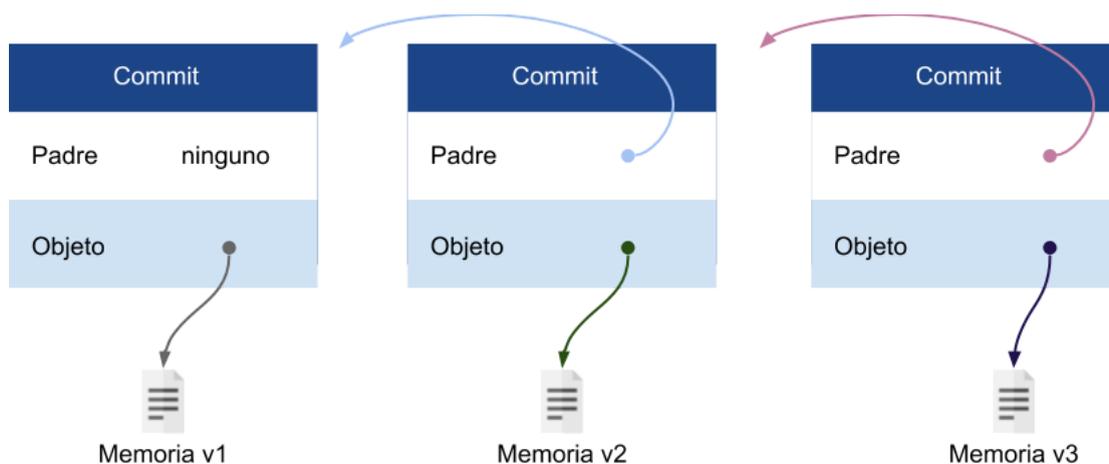


Direccionamiento de contenidos

Es la manera en la que se busca esa información en la red IPFS. Nos permite identificar un objeto mediante su hash, generando así un CID único para ese objeto, esto nos permite que cuando dos objetos son idénticos generarán el mismo CID haciendo así que la red no tenga duplicados de un mismo objeto.



Esto nos permite que los objetos de la red sean inmutables, es decir, siempre tendrán el mismo CID y no se podrá cambiar ese objeto asegurando así su integridad. Por fortuna IPFS permite que un objeto tenga versiones, es decir, al actualizar un objeto se genera un campo más que indica su versión anterior del mismo. Esto no permite ver todas las versiones de un documento haciendo que por ejemplo podamos ver la evolución de una página web.





En el protocolo HTTP cuando un objeto se mueve de lugar suele devolver un error bastante común el 404, ya que en vez de buscar el contenido buscamos su ubicación y si el contenido se mueve, esa ubicación no se actualiza. Un ejemplo:

<http://ejemplo.com/index.html>

Busca en un dirección



<http://10.20.30.40/index.html>

Mientras que en el protocolo IPFS se busca la información por su contenido que es convertido en hash, en otras palabras, se busca la información por su identificador de contenido (CID).

<http://ejemplo.com/index.html>

Busca por contenido



</ipfs/QmXoyvizjW3/index.html>

Lib2p2

Es una pila de red y una biblioteca modularizada a partir del Proyecto IPFS, y agrupadas por separado para que otras herramientas las usen. Es la herramienta la cual usan las aplicaciones distribuidas no solo para intercambiar información, sino que también permite enrutar los peers, descubrir nuevos peers y permite un amplio abanico de protocolos de transporte.

Intercambio

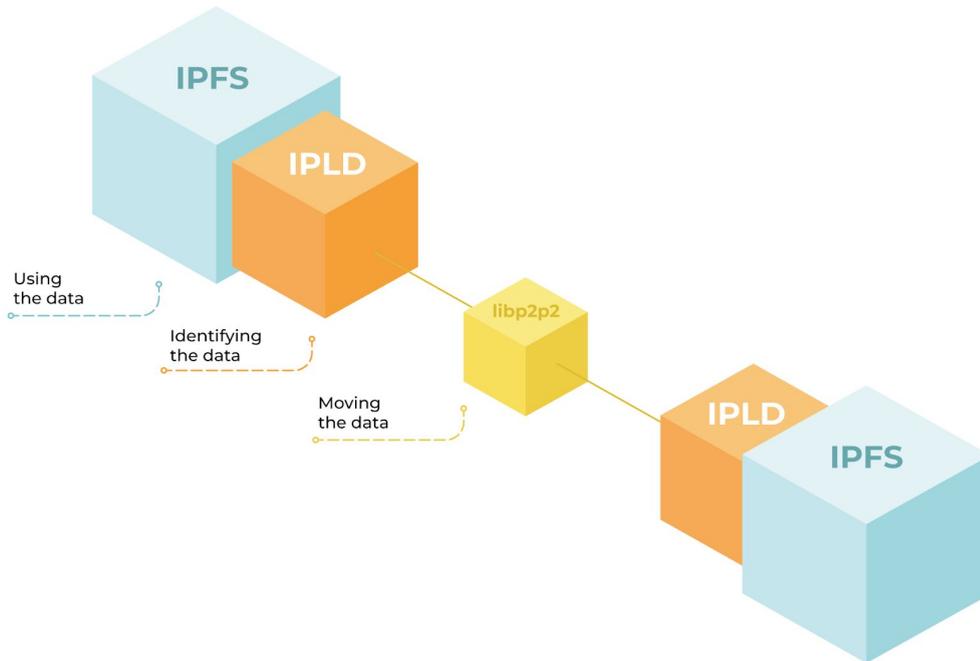
Permite intercambiar información por BitTorrent, Bitswap, **FTP** y **HTTP**.

Routing

Para enrutar el contenido dispone de varias opciones como Gossip, Chord, Kad DHT, mDNS y Delegated.

Network

Para los protocolos de intercambios de mensajes disponemos de CJDNS, UDT, RLPx, uTP, WebRTC, QUIC, TCP y WebSockets.



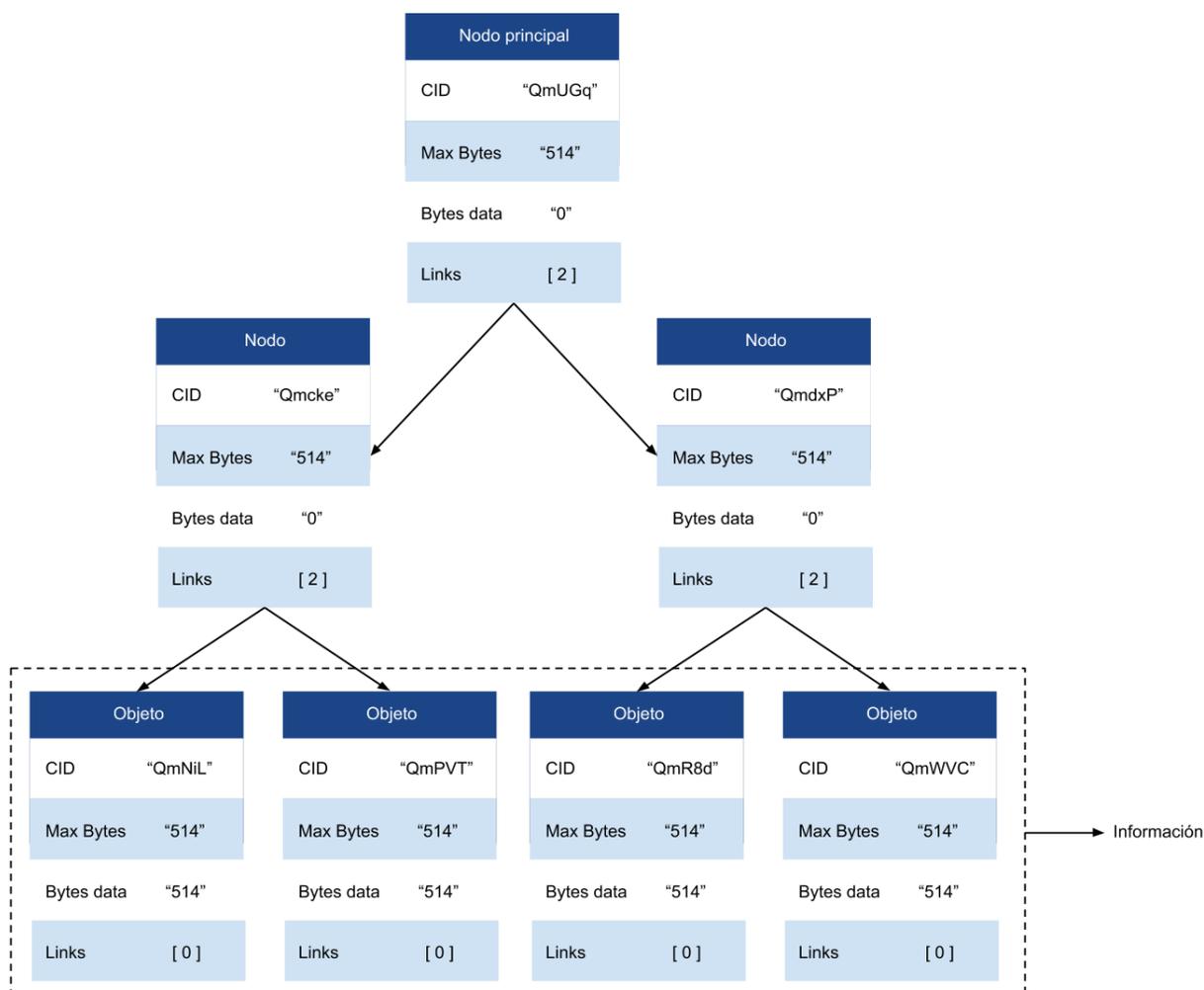
MFS

Como he mencionado unas líneas más arriba los objetos IPFS son direccionados por contenido e inmutables, haciendo que pueda ser complicado editarlos. **MFS** (Mutable File System) viene a solucionar ese problema, sistema de archivos normal basado en nombres que permite tratar archivos como lo haría con un sistema de archivos normal basado en nombres, en el cual se puede agregar, eliminar, mover y editar archivos. MFS hará todo el trabajo de subirlo a la red IPFS y actualizar los hashes si algún archivo se modifica.



Gráficos acíclicos dirigidos (DAG)

Se utiliza DAG de Merkle, que son DAG donde cada nodo tiene un identificador único que es un hash del contenido del nodo. ¿Te suena familiar? Esto se refiere al concepto CID que cubrimos en la sección anterior. Dicho de otro modo: identificar un objeto de datos por el valor de su hash es el direccionamiento de contenido.

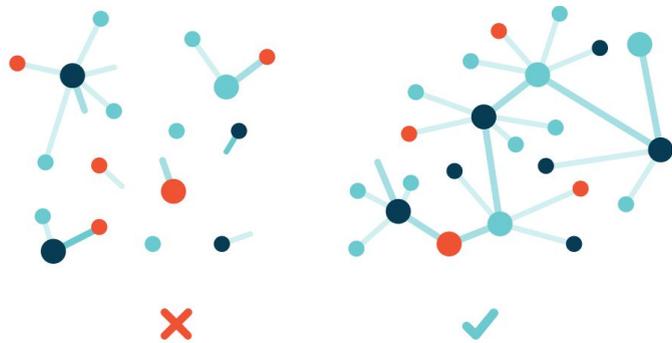


IPFS utiliza un DAG de Merkle que está optimizado para representar directorios y archivos. Es la estructura en la cual se guarda la información, tiene forma de pirámide descendente donde la información se guarda en el último piso, los pisos superiores son referencias a otros nodos pudiendo así compartimentar la información en pequeñas porciones de datos.

Si actualiza un sitio web, solo los archivos actualizados recibirán nuevas direcciones de contenido. Su versión anterior y su nueva versión pueden hacer referencia a los mismos bloques para todo lo demás. Esto puede hacer que la transferencia de versiones de grandes conjuntos de datos sea más eficiente porque solo necesita transferir las partes que son nuevas o han cambiado en lugar de crear archivos completamente nuevos cada vez.

Tablas hash distribuidas (DHT)

Para encontrar qué pares hospeda el contenido que busca, IPFS utiliza una tabla hash distribuida DHT. Una tabla hash es una base de datos de claves para valores. Una tabla hash distribuida es una en la que la tabla se divide entre todos los pares de una red distribuida.



Estos tres elementos trabajan juntos para crear lo que conocemos como el protocolo de IPFS. En el cual se combina con los hashes únicos del direccionamiento de contenido para identificar los objetos individualmente el cual se combina con DAG, ya que DAG nos proporciona la estructura en la cual se va a almacenar los hashes. Y para finalizar se utiliza DHT para obtener eficiencia a la hora de encontrar el contenido en los nodos, haciendo que baje la latencia ya que puede coger la información de diferentes nodos y organizarlos.

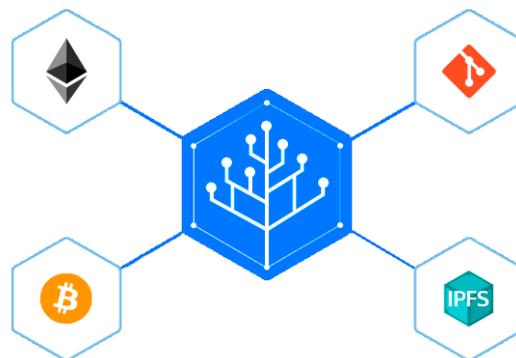
IPLD

IPLD es un conjunto de estándares e implementaciones para crear estructuras de datos descentralizadas que son universalmente direccionables y vinculables. Estas estructuras nos permitirán hacer para los datos lo que las URL y los enlaces hicieron para las páginas web HTML.

Permite trabajar a través de los límites del protocolo. El punto es que IPLD proporciona bibliotecas que hacen que los datos subyacentes sean interoperables entre herramientas y protocolos de forma predeterminada.

IPLD, es el modelo de datos de la web direccionable por contenido. Nos permite tratar todas las estructuras de datos vinculadas a hash como subconjuntos de un espacio de información unificado, unificando todos los modelos de datos que vinculan datos con hashes como instancias de IPLD.

Nos permite identificar el contenido y además nos permite generar enlaces de cualquier aplicación distribuida que incluya hashes para que es disponible en cualquier lugar, haciendo así que pueda obtener links de .bit, ethereum, bitcoin, git, etc.





¿Qué es IPNS?

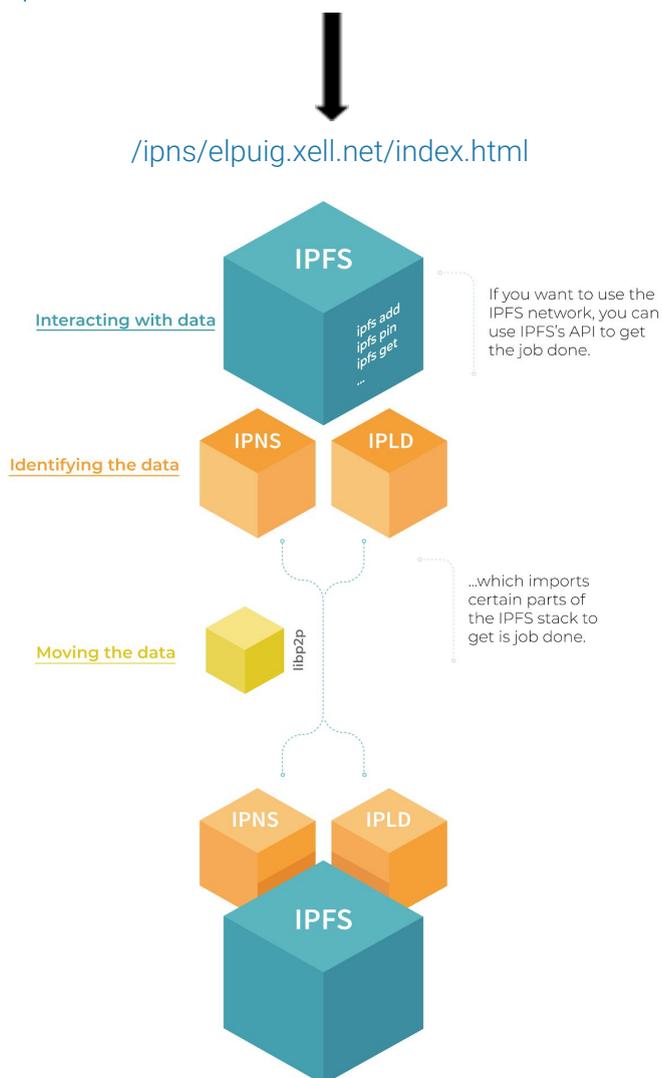
IPFS (InterPlanetary Name System) utiliza direccionamiento basado en contenido; crea una dirección de un archivo basada en los datos contenidos en el archivo. Desafortunadamente, tener una larga cadena de letras y números al azar no es lo ideal si tratamos de alojar un sitio web o compartir un enlaces. Si queremos compartir las direcciones IPFS con alguien, no sólo es difícil de leer, también se necesita dar a la persona un nuevo enlace cada vez que actualice el contenido. Ej:

`/ipfs/QmbezGequPwcsWo8UL4wDF6a8hYwM1hmbzYv2mnKkEWaUp`

El interplanetario Name System (IPNS) resuelve estos problemas mediante la creación de direcciones legibles por humanos que se pueden actualizar. Además IPNS puede apuntar a otros objetos como DNS, .onion, .bit para que esté disponible en IPFS.

También se puede usar **DNSLink** junto a IPNS para almacenar los links a nombre de dominios tradicionales. Es decir podemos pasar de:

`/ipns/QmbezGequPwcsWo8UL4wDF6a8hYwM1hmbzYv2mnKkJALL/index.html`





Cómo funciona IPNS

IPNS se puede implementar de muchas maneras, pero su implementación actual utiliza la tabla hash distribuida (DHT). Como consecuencia, solo la asignación más reciente de cada URI a su hash correspondiente está disponible para la resolución, olvidando cualquier asignación histórica. Esto no es bueno desde la perspectiva de archivo, ya que las versiones anteriores de un archivo pueden seguir existiendo en el almacén IPFS, pero se pierden sus asignaciones de URI correspondientes.

Recordemos que los Hashes son el resultado directo del contenido por lo tanto si el contenido cambia también lo hará el hash, haciendo que si quisieras compartir un "link" lo tengas que copiar y volver a pasarlo a una persona cada vez cambies un archivo. IPNS por lo tanto es una manera de mantener un solo link que apunta a la última versión de un archivo. Ej:

```
cisco@TestA2:~/ipfs/html$ ipfs add -r web
added QmTz7oiqq6TDiMgutDvREFFt57Xxyq7qoPwGkpui4Mptjv web/cat.jpg
added QmTYVRjFn9NNtDzw2RzWVHUFaKBANGGNTLjdzNqDqJDdML web/index.html
added QmdGfX8eXyrefuP3UwjTiMQteimubjyXxs8ZAhyFKmYf5J web/pagina1.html
added Qmee8hAuQAAzK8NultQGfPe8ymVQNwAr5sAU7ZGrA3CN3S web/style.css
added QmYphVrH2KMLm1Q6jcanNesrUSNXoRzLaUwtgqky454LXg web
473.87 KiB / 473.87 KiB [=====] 100.00%
cisco@TestA2:~/ipfs/html$ ipfs name publish QmYphVrH2KMLm1Q6jcanNesrUSNXoRzLaUwtgqky454LXg
Published to QmW7XPwXWMJMtST74eGwTtcJuZ59vxuKFh53VQ49GzJALL: /ipfs/QmYphVrH2KMLm1Q6jcanNesrUSNXoRzLaUwtgqky454LXg
```

Para acceder a la página web lo hacemos de esta manera:

</ipfs/QmW7XPwXWMJMtST74eGwTtcJuZ59vxuKFh53VQ49GzJALL/index.html>

Si cambiáramos algo la página web, lo tendríamos que volver a subir a IPFS con ipfs add -r y volver a ejecutar ipfs name publish con el nuevo Hash:

```
cisco@TestA2:~/ipfs/html$ ipfs add -r web
added QmTz7oiqq6TDiMgutDvREFFt57Xxyq7qoPwGkpui4Mptjv web/cat.jpg
added QmWt8do1xyieYuKogr9VxwSZFUoscreMZPn6hQ9fMAV4Zi web/index.html
added QmdGfX8eXyrefuP3UwjTiMQteimubjyXxs8ZAhyFKmYf5J web/pagina1.html
added Qmee8hAuQAAzK8NultQGfPe8ymVQNwAr5sAU7ZGrA3CN3S web/style.css
added Qmba6yxfTskqGLhtoor2nnoATyRQKvrz9ywmYcqZ98fmjn web
473.91 KiB / 473.91 KiB [=====] 100.00%
cisco@TestA2:~/ipfs/html$ ipfs name publish Qmba6yxfTskqGLhtoor2nnoATyRQKvrz9ywmYcqZ98fmjn
Published to QmW7XPwXWMJMtST74eGwTtcJuZ59vxuKFh53VQ49GzJALL: /ipfs/Qmba6yxfTskqGLhtoor2nnoATyRQKvrz9ywmYcqZ98fmjn
```

Como podemos ver al usar IPNS el link se mantiene el mismo aunque el contenido de actualiza.





¿Que es IPFS Cluster?

En una red p2p donde la información se transmite a toda la red, la información está distribuida y no en una máquina central. En las redes p2p si una máquina se desconecta la información no estará disponible para las demás máquinas. Es aquí cuando entra IPFS Cluster, ya que nos proporciona una manera de replicación de datos en una red de enjambre IPFS, permitiéndonos conectarnos a los pares que disponen de la información y replicando a todos los pares de esa misma red para preservar la información.



IPFS Cluster no está integrado con el paquete de IPFS, es decir, es un modelo que podemos acoplar a IPFS para mejorar la eficiencia de la red y la integridad de los datos.

Cómo funciona IPFS-Clúster

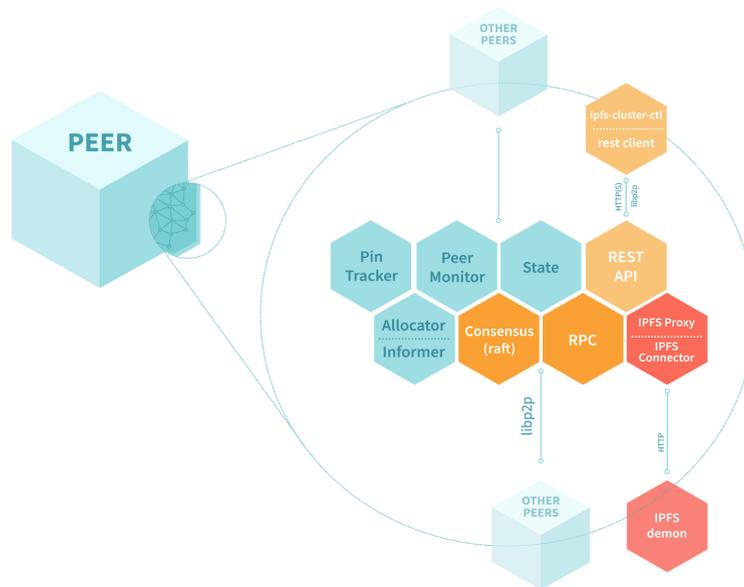
IPFS Cluster consiste en tres paquetes binarios que nos administran nuestro Cluster y la red.

- `ipfs-cluster-service` ejecuta un Cluster peer (similar al demonio `ipfs`) usando un archivo de configuración y almacenando información en el disco.
- `ipfs-cluster-ctl` se usa para comunicarse con un Cluster peer y realizar acciones como fijar CID de IPFS al Cluster.
- `ipfs-cluster-follow` ejecuta un seguidor Cluster peer. Se puede utilizar como un sustituto de `ipfs-cluster-service` para este caso de uso específico y combina algunas funciones de servicio y `ctl`.

IPFS Clúster es una aplicación totalmente distribuida, `ipfs-cluster-service` ejecuta un Clúster peer y todos los peers son iguales. Los Clusters peers forman una red libp2p (privada) separada y aislada. Esta red no interactúa con la red principal de IPFS, ni con otras redes privadas de IPFS y se usa únicamente para que los peers del Clúster puedan comunicarse y operar, pero a diferencia de IPFS, no está pensado para bootstraps públicos.

Esto significa que los Clúster peers normalmente necesitarán sus propios bootstrappers que pueden ser cualquier otro Clúster peer, es decir, otro Clúster en la red. Aunque a veces también funcionan con algún mDNS (multicast DNS).

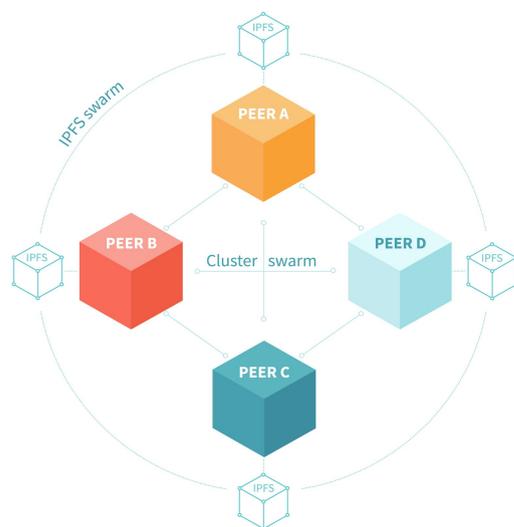
Está sería una ilustración de como funciona un Clúster junto a IPFS, el cual no altera en ningún momento el funcionamiento normal de IPFS:



Descripción Técnica De La Solución Adoptada

Red IPFS + IPNS Privada & IPFS-Cluster

Con IPFS no se puede limitar la conexión a una subred en sí, sin embargo, podemos configurar los nodos IPFS para que busques uno o varios nodos en específico, utilizando el **framework** de **Bootstrap**. Además si queremos que nuestra red IPFS sea auténticamente privada para que otros nodos o incluso nodos maliciosos no puedan conectar con nuestra red utilizaremos **Swarm Key** para la generación de claves de enjambre. Con este método conseguiremos tener una red totalmente privada y segura. También utilizaremos IPNS para encontrar la última versión de una página sin tener que preocuparnos por los hashes generados al actualizar un documento.+



Además de una red privada se implementará **IPFS Cluster** el cual que nos permitira tener disponibilidad y redundancia automatizada de datos en IPFS ya que proporciona orquestación de datos a través de un enjambre de demonios IPFS asignando, replicando y rastreando un conjunto de pines global distribuido entre varios pares.



Funcionamiento IPFS & IPFS-Cluster

IPFS

Una vez inicializado IPFS e iniciado el daemon, podremos añadir archivos a IPFS de la siguiente manera (por defecto IPFS ancla este objeto en el disco para que no sea borrado). Los demás host de la misma red podrán acceder al contenido pero este caso al ser un red privada no hay replicación de datos así que si un host se desconecta esta información no estará disponible.

```
ipfs add [archivo] | add -r [carpeta]
> added QmX4RWwYytRR63138cvnPQeZ5A2tXX9dwtzc65P57nQ7BM README.txt
> 22 B / 22 B [=====] 100.00%
```

Si queremos ver lo que hemos subido a la red o los objetos que tenemos anclados simplemente utilizamos:

```
ipfs pin ls
> bafkreigbnk77avf5g77chnb3yc6nzd2ucfuzr5ktjkeibcz2nkm3a2uyqa indirect
> bafkreicklkgjyxjnhhcuop26ozhp12b3gjz673gk5sr6xhd3w4f3wn4kru indirect
> bafkreifw4stpl3p7juzs3kzxqc5lno52ee2lsztp5i4kg6y2a62wbtoxba indirect
```

Si queremos eliminar lo que no tenemos anclado y que ocupa espacio en el disco:

```
ipfs repo gc
```

Si queremos subir nuestra página web y habilitar IPNS lo hacemos de la siguiente manera. Subimos la carpeta a IPFS, cogimos el hash del directorio raíz y lo publicamos. Podremos acceder con el hash que sigue después de "Publish to":

```
ipfs add -r web
> added QmTz7oiqq6TDiMgutDvREFFt57XXyq7qoPwGkpui4Mptjv web/cat.jpg
> added QmWt8d0lxYieYuKogr9VxwSZFUoscreMZPn6hQ9fMAV4Zi web/index.html
> added QmdGfX8eXyrefuP3UwjTiMQteimubjyXxs8ZAhyFKmYf5J web/pagina1.html
> added Qmee8hAuQAAzK8NultQGFPe8ymVQNwAr5sAU7ZGrA3CN3S web/style.css
> added Qmba6yxftskqGLhtoor2nnoATyRQKvrz9ywmYcqZ98fmjn web
> 473.91 KiB / 473.91 KiB [=====] 100.00%
ipfs name publish Qmba6yxftskqGLhtoor2nnoATyRQKvrz9ywmYcqZ98fmjn
> Published to QmW7XPwXWMJMtST74eGwTTcJuZ59vxuKFh53VQ49GzJALL:
> /ipfs/Qmba6yxftskqGLhtoor2nnoATyRQKvrz9ywmYcqZ98fmjn
```



Si queremos usar MFS, para crear una carpeta lo podemos hacer de esta manera

```
ipfs files mkdir /[nombre]
```

Para listar los objetos MFS lo podemos hacer de esta manera:

```
ipfs files ls
```

Inicializa el repositorio regularmente.

```
cat README.txt | echo "Some text" | ipfs files write -e -p /"Space  
Cats"/README.txt
```

Comandos Básicos

ipfs init

Nos permite inicializar el repositorio cargando toda la configuración necesaria para usar IPFS. Ejemplos:

Inicializa el repositorio regularmente.

```
ipfs init
```

Inicializa el repositorio sin descargar y anclar la ayuda al repositorio

```
ipfs init -e
```

ipfs add

Añadimos un archivo o un directorio al repositorio de IPFS. Algunas opciones más utilizadas:

-r: añade recursivamente.

-w: crea un directorio y mete el archivo dentro de ese directorio creado.

-s: especifica el tamaño de los bloques al fragmentar los datos, dependiendo del tamaño de los bloques de fragmentación dos archivos idénticos pueden tener dos hashes diferentes. Por defecto el tamaño es de 256 * 1024 bytes. Ejemplos:

Añade cat al repositorio creando un directorio y metiendo el archivo cat.jpg dentro.

```
ipfs add cat.jpg -w
```

Añade la carpeta recursivamente

```
ipfs add -r ~/web
```



ipfs cat

Muestra los datos de un objeto IPFS. Ejemplos:

Muestra el contenido del objeto

```
ipfs cat QmWpMDJZCbdbRQvwFqMkzkH23DdLVQVeTGw4F5EZX2wG9V
```

Guardar un objeto IPFS y que se llame **algo**.

```
ipfs cat QmWpMDJZCbdbRQvwFqMkzkH23DdLVQVeTGw4F5EZX2wG9V > algo
```

ipfs get

Nos permite descargar los contenido de IPFS a nuestro almacenamiento local.

Ejemplos:

Descarga el archivo en un carpeta en concreto.

```
ipfs get -o~/Descargas QmWpMDJZCbdbRQvwFqMkzkH23DdLVQVeTGw4F5EZX2wG9V
```

Descarga el archivo comprimido en .tar

```
ipfs get -a QmWpMDJZCbdbRQvwFqMkzkH23DdLVQVeTGw4F5EZX2wG9V
```

Descarga el archivo comprimido en .gz

```
ipfs get -C QmWpMDJZCbdbRQvwFqMkzkH23DdLVQVeTGw4F5EZX2wG9V
```

ipfs ls

Listar el contenido del directorio para objetos del sistema de archivos Unix. Algunas opciones más utilizadas:

- v:** imprimir encabezados de tabla (hash, tamaño, nombre).
- s:** habilite la transmisión experimental de entradas de directorio a medida que se recorren

ipfs refs

Lista de enlaces (referencias) de un objeto. Algunas opciones más utilizadas:

- e:** emitir formato de borde: ``<from> -> <to>``.
- r:** enumere recursivamente enlaces de nodos secundarios.

También podemos usar la opción local para listar todas las referencias locales.

```
ipfs refs local
```

Nota:



Las referencias que salen con **ipfs refs local** si no están ancladas (**pin**) en nuestro IPFS el recolector de basura las eliminará, haciendo que si tengamos documentos importantes que no están anclados los mismos se eliminarán:

Ejemplo y comparativa con **ipfs ls**.

```
cisco@TestA1:~$ ipfs ls -v -s QmNxXqkVsCWspWzpjajHtEXF1psiZZcQxcEcT3ig1GYwEw
Hash                               Size      Name
QmTarVBRCs3LTLCnQJSuy5BH1padCPTuLyZgMTkNC8Bmp4 -        bar/
QmCReN2Xh5zFkCXJn77bLiKo2*5r6nqfGvYGM8ouiQTCg3 11       hello.txt
cisco@TestA1:~$ ipfs refs -e -r QmNxXqkVsCWspWzpjajHtEXF1psiZZcQxcEcT3ig1GYwEw
QmNxXqkVsCWspWzpjajHtEXF1psiZZcQxcEcT3ig1GYwEw → QmTarVBRCs3LTLCnQJSuy5BH1padCPTuLyZgMTkNC8Bmp4
QmTarVBRCs3LTLCnQJSuy5BH1padCPTuLyZgMTkNC8Bmp4 → QmWfujbkfFsozy5exuswceBALAEDmWT2DxP3aeURQ3Xv9c
QmWfujbkfFsozy5exuswceBALAEDmWT2DxP3aeURQ3Xv9c → QmXsn1gLxFOqRxFa2QkLYsquStYR4MgSeLvshHbcrECPt6
QmNxXqkVsCWspWzpjajHtEXF1psiZZcQxcEcT3ig1GYwEw → QmCReN2Xh5zFkCXJn77bLiKo2*5r6nqfGvYGM8ouiQTCg3
```

Comandos De Estructura De Datos

ipfs object

Podemos interactuar con los objetos de IPFS. Comandos más utiles:

Salida de los enlaces apuntados por el objeto especificado.

```
ipfs object links QmNxXqkVsCWspWzpjajHtEXF1psiZZcQxcEcT3ig1GYwEw
```

Obtenga estadísticas para el nodo DAG nombrado por <hash>.

```
ipfs object stat QmNxXqkVsCWspWzpjajHtEXF1psiZZcQxcEcT3ig1GYwEw
```

Descarga el archivo comprimido en .gz

```
ipfs get -C QmWpMDJZCbdbRQvWfQmKzkH23DdLVQVeTGw4F5EZx2wG9V
```

ipfs files

Es una API para manipular objetos IPFS como si fueran un sistema de archivos Unix. Nos permite manejar IPFS de forma **MFS** para la comodidad del usuario. Estas son las opciones más usadas:

files ls -l: nos muestra el hash que tiene ese objeto.

files write -e: crear el archivo si no existe.

files write -p: crear un directorio si no está asignado a ninguno.

files rm -r: borra una carpeta de forma recursiva.

Podemos listar todas nuestras carpetas de esta manera (si no sale nada es que no hay nada creado):

```
ipfs files ls -l /"Space cats"
```

Con este comando podemos obtener las estadísticas de la carpeta (si el nombre del archivo tiene espacios se debe de poner entre comillas).



```
ipfs files stat /"Space cats"
```

Podemos crear carpetas MFS de esta manera (si el nombre del archivo tiene espacios se debe de poner entre comillas):

```
ipfs files mkdir /"Hello World!"
```

Podemos borrar carpetas de forma recursiva de esta manera (si el nombre del archivo tiene espacios se debe de poner entre comillas):

```
ipfs files rm -r /"Hello World!"
```

Descarga el archivo comprimido en .gz

```
cat README.txt | echo "Space cats convention" | ipfs files write -e -p  
/"Space Cats"/README.txt
```

Podemos ver el contenido de los archivos con este comando:

```
ipfs files read /"Space Cats"/README.txt
```

Advanced Commands

ipfs pin

Ancla de desancla objetos en nuestro disco local.

Verificar que los bloques de datos no esten corruptos:

```
ipfs pin ls
```

Anclamos un objetos de forma recursiva:

```
ipfs pin add -r
```

Desvinculamos objetos de forma recursiva.:

```
ipfs pin rm -r
```

ipfs repo

Nos permite administrar el repositorio de IPFS. Los dos comandos más relevantes serían:

Obtenemos las estadísticas del repositorio utilizado actualmente:

```
ipfs repo stat
```

Realice un barrido recolectando la basura en el repositorio, se considera basura todos aquellos bloques que no están en un pin:

```
ipfs repo gc
```



Verificar que los bloques de datos no esten corruptos:

```
ipfs repo verify
```

Obtenemos las estadísticas del repositorio utilizado actualmente:

```
ipfs repo stat
```

IPFS-Cluster

Una hemos inicializado IPFS-Cluster y el daemon, para replicar un objeto lo que debemos de hacer es este comando:

```
ipfs-cluster-ctl add text.txt
> added QmfM2r8seH2GiRaC4esTjeraXEachRt8ZsSeGaWTPLYMoG text.txt
```

Podemos saber en cuantos Cluster está nuestro archivo simplemente con el siguiente comando:

```
ipfs-cluster-ctl status QmfM2r8seH2GiRaC4esTjeraXEachRt8ZsSeGaWTPLYMoG
QmfM2r8seH2GiRaC4esTjeraXEachRt8ZsSeGaWTPLYMoG:
> TestA2   : PINNED | 2020-05-29T18:26:07.221113479+02:00
> TestA1   : PINNED | 2020-05-29T16:26:07.251292253Z
```

Comandos útiles

Podremos ver cual es la id que tiene nuestro Cluster:

```
ipfs-cluster-ctl id
> 12D3KooWNyQ3SvsQMbZ4ugbqeXbCpXBtPTWS4eJZWuVmxquies2V | ClusterA2 | Sees 1
other peers
```

Añade un directorio recursivamente:

```
ipfs-cluster-ctl add -r text.txt
```

Con este comando podremos administrar los hashes en nuestro cluster, anclando los, eliminandolos o listando los respectivamente:

```
ipfs-cluster-ctl pin [add / rm / ls] QmfM2r8seH2GiRaC4esTjeraXEachRt8ZsSaW
```

Con este comando listar los pares conectados o eliminar algún par respectivamente:

```
ipfs-cluster-ctl peers [ ls / rm ] QmfM2r8seH2GiRaC4esTjeraXEachRt8ZsSeGaW
```



Plan De Instalación y Configuración

Instalacion Windows

GO

Primero debemos descargar la última versión de GO en su pagina de descargas <https://golang.org/dl/>. A continuación seguimos los pasos del instalador para instalar GO el sistema.

IPFS

Primero debemos descargar la distribución de IPFS que vamos a usar, en mi caso vamos a usar la versión más reciente con implementación del código GO en la siguiente enlace <https://dist.ipfs.io/#go-ipfs>.

1. Descomprima el archivo y muévelo a un lugar útil:

```
Expand-Archive -Path go-ipfs-v0.5.0.zip -DestinationPath D:\Archivos de programa\IPFS\
```

2. Muévase a la carpeta y compruebe que funciona:

```
cd D:\Archivos de programa\IPFS\go-ipfs\  
.\ipfs.exe --version  
> ipfs version 0.5.0
```

Aunque ya podemos usar IPFS en este momento, es mejor agregar un alias a PS para poder trabajar con más comodidad.

3. Imprima el directorio de trabajo actual y cópielo en el portapapeles:

```
cd D:\Archivos de programa\IPFS\go-ipfs\  
.\ipfs.exe --version  
> Path  
> ----  
> C:\Archivos de programa\IPFS\go-ipfs
```



4. Agregue la dirección que acaba de copiar a PowerShell agregándole al final del archivo almacenado en: WindowsPowerShell.

```
Add-Content
D:\Usuarios\<Usuario>\Documents\WindowsPowerShell\profile.ps1
"[System.Environment]::SetEnvironmentVariable('PATH',`$Env:PATH+';D:\Archivos de Programa\IPFS\go-ipfs')"
```

5. Cierre y vuelva a abrir la ventana de PowerShell. Pruebe que su ruta IPFS está configurada correctamente yendo a su carpeta de inicio y preguntando a IPFS por la versión::

```
cd ~
ipfs --version
> ipfs version 0.5.0
```

Instalación Linux

Git

1. Instalamos git simplemente con el siguiente comando:

```
sudo apt install git
```

GO

1. Primero descargamos última versión de GO: <https://golang.org/dl/>

```
wget -c https://dl.google.com/go/go1.14.3.linux-amd64.tar.gz
```

2. Descomprimir GO:

```
sudo tar -C /usr/local -xzf go1.14.3.linux-amd64.tar.gz
```

2. Creamos la carpeta de Go:

```
mkdir ~/go
```

3. Editamos el archivo ~/.bashrc añadiendo al final de archivo el siguiente contenido:

```
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```



4. Actualice el archivo `.bashrc` y verifique la versión Go:

```
source ~/.bashrc
go version
> go version go1.14.2 linux/amd64
```

4. Por último borramos las carpetas que no necesitamos:

```
sudo rm -r go1.14.3.linux-amd64.tar.gz
```

IPFS

1. Download `go-ipfs_v0.5.0_linux-amd64.tar.gz` from GitHub:

```
wget
https://github.com/ipfs/go-ipfs/releases/download/v0.5.0/go-ipfs_v0.5.0_linux-amd64.tar.gz
```

2. Descomprima el archivo:

```
tar -xvzf go-ipfs_v0.5.0_linux-amd64.tar.gz
> x go-ipfs/install.sh
> x go-ipfs/ipfs
> x go-ipfs/LICENSE
> x go-ipfs/LICENSE-APACHE
> x go-ipfs/LICENSE-MIT
> x go-ipfs/README.md
```

3. Vaya a la carpeta y ejecute el script de instalación:

```
cd go-ipfs
sudo bash install.sh
> Moved ./ipfs to /usr/local/bin
```

4. Compruebe que IPFS se ha instalado correctamente:

```
ipfs --version
> ipfs version 0.5.0
```

4. Por último borramos las carpetas que no necesitamos:

```
sudo rm -r go-ipfs_v0.5.0_linux-amd64.tar.gz go-ipfs
```



IPFS Cluster

1. Crear un archivo en el directorio `/etc/systemd/system/` con el siguiente contenido:

```
git clone https://github.com/ipfs/ipfs-cluster.git
```

2. Entramos en la carpeta y ejecutamos los siguientes comandos para instalar ipfs-cluster, se instalará en `$GOPATH/bin`:

```
cd ~/ipfs-cluster
go install ./cmd/ipfs-cluster-service
go install ./cmd/ipfs-cluster-ctl
go install ./cmd/ipfs-cluster-follow
```

3. Podemos comprobar la instalación con el siguiente comando:

```
ipfs-cluster-service --version
> ipfs-cluster-service version 0.13.0-next
ipfs-cluster-ctl --version
> ipfs-cluster-ctl version 0.13.0-next
```

4. Por último borramos las carpetas que no necesitamos:

```
sudo rm -r ipfs-cluster/
```

Opcional

WebUI

Para obtener WebUI (interfaz gráfica en el navegador) para administrar IPFS debemos seguir los siguientes pasos.

Nota:

En caso de que haya borrado la lista bootstrap que permite conectar con la red pública de IPFS ejecute este comando antes:

```
ipfs bootstrap add --default
```

1. Debemos inicializar con **ipfs init** si no has hecho, una vez inicializado debemos arrancar el daemon.

```
ipfs bootstrap add --default
```



2. Ahora lo que debemos hacer es descargar WebUI en la máquina con el siguiente comando:

```
ipfs pin add
bafybeidatz2hli6fgu3zul5woi27ujesdf5o5a7bu622qj6ugharciwjg
```

3. Una vez añadido simplemente borramos la lista bootstrap y la volvemos a poner como red privada, para conectar con WebUI simplemente ponemos en el navegador:

```
http://127.0.0.1:5001/webui
```

4. Borramos cualquier archivo que haya podido descargar:

```
ipfs repo gc
```

Configuración Red IPFS Privada

Swarm Key

1. Inicializamos los nodos de todas las máquinas con el siguiente comando:

```
ipfs init
> initializing IPFS node at /home/cisco/.ipfs
> generating 2048-bit RSA keypair...done
> peer identity: Qmb2ycbkZoEcj1lSDAYREzXLtyCrSFvqy1dQahUDCHZfTZ
> to get started, enter:
> ipfs cat /ipfs/QmQPeNsJPYVWPFVDVHb77w8G42Fvo15z4bG2X8D2GhfbSXC/readme
```

2. Descargamos el generador de swarm key de git:

```
go get -u github.com/Kubuxu/go-ipfs-swarm-key-gen/ipfs-swarm-key-gen
```

3. Generamos el swarm key en el nodo principal:

```
ipfs-swarm-key-gen > ~/.ipfs/swarm.key
```

4. Copiamos el archivo generado en el nodo principal al resto de nodos:

```
scp ~/.ipfs/swarm.key [usuario]@[IP]:~/.ipfs/
```



Bootstrap

1. Eliminar todas las entradas de bootstrap de todos los nodos (incluso en el nodo principal):

```
ipfs bootstrap rm --all
> removed
/dnsaddr/bootstrap.libp2p.io/p2p/QmNnooDu7bfjPFoTZYxMNLWUQJyrVwtbZg5gBMjTezGA
JN
> removed
/dnsaddr/bootstrap.libp2p.io/p2p/QmQCU2EcMqAqQPR2i9bChDtGNJchTbq5TbXJJ16u19uL
Ta
> removed
/dnsaddr/bootstrap.libp2p.io/p2p/QmbLHAnMoJPWSCR5Zhtx6BHJX9KiKNN6tpvbUcqanj75
Nb
```

2. Podemos ver los resultados en el siguiente apartado:

```
ipfs config show
> [...]
> "AutoNAT": {},
> "Bootstrap": null,
> [...]
```

3. Agregar la dirección IP y el Peer Identity (dirección hash) del nodo principal a cada uno de los nodos (cambiar incluso en el nodo principal):

```
ipfs bootstrap add /ip4/<IP nodo principal>/tcp/4001/ipfs/<Hash Peer identity
del nodo Principal>

ipfs bootstrap add /ip4/<IP nodo principal>/tcp/4001/p2p/<Hash Peer identity
del nodo Principal>
```

Para ver la ip de la máquina lo podemos hacer con

```
hostname -I
```

Y el Peer Identity con:

```
ipfs config show | grep "PeerID"
```



4. Forzamos que Lib2p2p solo busque en nuestra red privada:

```
export LIBP2P_FORCE_PNET=1
```

Systemd

1. Crear un archivo en el directorio `/etc/systemd/system/` con el siguiente contenido:

```
[Unit]
Description=IPFS Daemon
After=syslog.target network.target remote-fs.target nss-lookup.target
[Service]
Type=simple
ExecStart=/usr/local/bin/ipfs daemon
User=<<Usuario en cual ha instalado IPFS>>
Restart=on-failure
[Install]
WantedBy=default.target
```

2. Recalcamos el demonio de systemd:

```
sudo systemctl daemon-reload
```

3. Habilitamos el servicio ipfs con el siguiente comando:

```
sudo systemctl enable ipfs.service
```

4. Ahora para iniciar el servicio de forma manual y comoda, tambien se iniciara al encender el equipo:

```
sudo systemctl start ipfs
```



IPFS-Cluster

Configuración

1. Iniciar el Clúster del nodo principal y de todos los demás:

```
ipfs-cluster-service init
> 2020-05-28T13:01:41.496+0200 INFO  config  config/config.go:481  Saving
configuration
> configuration written to /home/cisco/.ipfs-cluster/service.json.
> 2020-05-28T13:01:41.499+0200 INFO  config  config/identity.go:73  Saving
identity
> new identity written to /home/cisco/.ipfs-cluster/identity.json
> new empty peerstore written to /home/cisco/.ipfs-cluster/peerstore.
```

2. Ahora debemos modificar el archivo `~/ipfs-cluster/service.json` en el resto de nodos para que tengan la misma clave, haciendo que los Clusters se puedan comunicarse:

```
sed -i '/"secret"/c\"secret\": \"[Clave Secreta]\",' service.json
```

Podemos ver esta clave simplemente haciendo un `cat` en el **nodo principal**:

```
cat ~/.ipfs-cluster/service.json
> "cluster": {
>   "peername": "TestA1",
>   "secret": "14dba9e7f51905451ba8a94ffef43e2ff97bb3409caa61d713581fce5317",
```

3. Ahora lo que debemos hacer es iniciar el daemon del Cluster en el nodo principal, podemos hacerlo de dos maneras distintas, de forma manual o con **systemd**:

```
ipfs-cluster-service daemon
```

4. Para iniciar el daemon del Clúster en los demás nodos podemos hacer de dos maneras, de forma manual o editando `~/ipfs-cluster/peerstore`:

```
ipfs-cluster-service daemon --bootstrap /ip4/[IP nodo
principal]/tcp/9096/ipfs/[Hash ID del cluster]
```



Para ver la **ID del Cluster** que tenemos en el servidor podemos ejecutar el siguiente comando (el daemon del cluster tiene que estar activado):

```
ipfs-cluster-ctl id

> 12D3KooWSFVXSkWukiWfSCQRtodXga2pShRL4YSTRFcpN6irwUMj | TestA1 | Sees 1
other peers
> > Addresses:
>/ip4/127.0.0.1/tcp/9096/p2p/12D3KooWSFVXSkWukiWfSCQRtodXga2pShRL4YSTRFcpN6ir
wUMj
>/ip4/192.168.1.94/tcp/9096/p2p/12D3KooWSFVXSkWukiWfSCQRtodXga2pShRL4YSTRFcpN
6irwUMj
>/ip4/217.125.160.188/tcp/61734/p2p/12D3KooWSFVXSkWukiWfSCQRtodXga2pShRL4YSTR
FcpN6irwUMj
> > IPFS: Qmc8299UEPY9m6tjj3VrzDK8PSKHkoKMiTePR2PbWHLmwd
>/ip4/127.0.0.1/tcp/4001/p2p/Qmc8299UEPY9m6tjj3VrzDK8PSKHkoKMiTePR2PbWHLmwd
>/ip4/192.168.1.94/tcp/4001/p2p/Qmc8299UEPY9m6tjj3VrzDK8PSKHkoKMiTePR2PbWHLm
d
>/ip4/217.125.160.188/tcp/55038/p2p/Qmc8299UEPY9m6tjj3VrzDK8PSKHkoKMiTePR2PbW
HLmwd
>/ip6>:::1/tcp/4001/p2p/Qmc8299UEPY9m6tjj3VrzDK8PSKHkoKMiTePR2PbWHLmwd
```

Para editar el archivo simplemente copiamos lo siguiente:

```
echo "/ip4/[IP nodo principal]/tcp/9096/ipfs/[Hash ID del cluster]" >>
~/.ipfs-cluster/peerstore
```

Y ahora podemos iniciar el daemon sin necesidad de poner opciones de inicio:

```
ipfs-cluster-service daemon
```

Al iniciar el daemon en todos los nodos, podemos utilizar el siguiente comando para comprobar que de verdad se han conectado:

```
ipfs-cluster-ctl status
```

```
cisco@TestA1:~$ ipfs-cluster-ctl status
QmWpMDJZCbdbRQvwFqMkzkH23DdLVQVeTGw4F5EZx2wG9V:
> TestA3           : PINNED | 2020-05-28T13:54:48.534007363Z
> TestA2           : PINNED | 2020-05-28T13:54:48.424102033Z
> TestA1           : PINNED | 2020-05-28T15:54:48.435612975+02:00
```



Systemd

1. Crear un archivo en el directorio `/etc/systemd/system/` con el siguiente contenido:

```
[Unit]
Description=IPFS Daemon
After=syslog.target network.target remote-fs.target nss-lookup.target ipfsw
[Service]
Type=simple
ExecStart=/home/cisco/go/bin/ipfs-cluster-service daemon
User=<<Usuario en cual ha instalado IPFS-Cluster>>
Restart=on-failure
[Install]
WantedBy=default.target
```

2. Recalcamos el demonio de systemd:

```
sudo systemctl daemon-reload
```

3. Habilitamos el servicio ipfs con el siguiente comando:

```
sudo systemctl enable cluster.service
```

4. Ahora para iniciar el servicio de forma manual y comoda, tambien se iniciara al encender el equipo:

```
sudo systemctl start cluster
```

Problemas Encontrados Y Solución Adoptada

Instalación IPFS vía comandos PowerShell

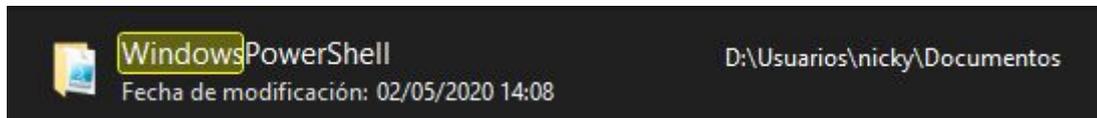
Problemas de perfil PowerShell

Al intentar guardar en el perfil nos dirá que no existe:

```
PS C:\Users\nicky> Add-Content D:\Usuarios\nicky\Documentos\WindowsPowerShell\profile.ps1 "[System.Environment]::SetEnvironmentVariable('PATH', $Env:PATH+';D:\Archivos de Programa\IPFS\go-ipfs')"
Add-Content : No se puede encontrar una parte de la ruta de acceso
'D:\Usuarios\nicky\Documentos\WindowsPowerShell\profile.ps1'.
En línea: 1 Carácter: 1
+ Add-Content D:\Usuarios\nicky\Documentos\WindowsPowerShell\profile.ps ...
~
+ CategoryInfo          : ObjectNotFound: (D:\Usuarios\nic...ell\profile.ps1:String) [Add-Content], DirectoryNotFo
undException
+ FullyQualifiedErrorId : GetContentWriterDirectoryNotFound,Microsoft.PowerShell.Commands.AddContentCommand
```

Solución

Creamos la carpeta donde vamos a guardar nuestro perfil:



Para crear un perfil con el siguiente comando en PS:

```
PS C:\Users\nicky> New-Item -ItemType File $PROFILE

Directorio: D:\Usuarios\nicky\Documentos\WindowsPowerShell

Mode                LastWriteTime         Length Name
----                -
-a-----          02/05/2020    14:02           0 Microsoft.PowerShell_profile.ps1
```

Podemos ver la lista de perfiles con el siguiente comando:

```
PS C:\Users\nicky> $PROFILE | Format-List -Force

AllUsersAllHosts      : C:\Windows\System32\WindowsPowerShell\v1.0\profile.ps1
AllUsersCurrentHost   : C:\Windows\System32\WindowsPowerShell\v1.0\Microsoft.PowerShell_profile.ps1
CurrentUserAllHosts  : D:\Usuarios\nicky\Documentos\WindowsPowerShell\profile.ps1
CurrentUserCurrentHost : D:\Usuarios\nicky\Documentos\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
Length                : 79
```

Problemas al abrir PowerShell

Al guardar el nuevo perfil y abrir PS nos salta un error debido a que el sistema no puede ejecutar scripts en la consola.

```

: : No se puede cargar el archivo D:\Usuarios\nicky\Documentos\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
porque la ejecución de scripts está deshabilitada en este sistema. Para obtener más información, consulta el tema
about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170.
En línea: 1 Carácter: 3
+ . 'D:\Usuarios\nicky\Documentos\WindowsPowerShell\Microsoft.PowerShell ...
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

```

Solución

Abrir PS como administrador y escribir lo siguiente.

```
PS C:\Windows\system32> Get-ExecutionPolicy
Restricted
PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías
exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en
https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "N"): s
PS C:\Windows\system32>
```

Cerrar PS y volver abrir normalmente.

```
PS C:\Users\nicky> ipfs --version
ipfs version 0.5.0
```



Conclusión

En conclusión puede decir que he cumplido con los objetivos principales del proyecto e incluso los objetivos secundarios. He creado un red descentralizada y distribuida que permite la replicación de datos aumentando así su fiabilidad e integridad de los datos en la red. He de decir que me ha faltado tiempo para poder experimentar e implementar lo que tenía en mente, por eso aún que mis objetivos iniciales se han cumplido siento que toda via el trabajo no está acabado y todavía falta mucho que implementar y pulir al proyecto.

Puntos positivos

Al hacer un proyecto con esta tecnología he aprendido no solo como funciona un red descentralizada sino que ahora se lo conceptos básicos de blockchain y hashes permitiendo así desarrollarme más en el mundo de las redes.ç

Otro punto a favor es que la documentación y algunos tutoriales están en inglés he podido mejorar ese idioma aumentando aún más mis capacidades hacia el futuro laboral.

Puntos negativos

Al ser un tecnología relativamente nuevo y que aún está en desarrollo no solo es difícil encontrar información sobre cómo funciona en profundidad IPFS y sus módulos. La poca información que hay son de páginas webs antiguas que no son compatibles con las versiones actuales, la mayoría de información se puede conseguir en su rama oficial de GitHub en las notas de los propios desarrolladores y proyectos en curso, esto supone un dificultad adicional no solo porque la información escasea sobre el funcionamiento sino que hablas con términos avanzados del proyecto.

A mejorar

El proyecto aún que haya cumplido las expectativas no significa que sea perfecto, tanto mi proyecto actual como IPFS pueden ser mejoras sustancialmente. Mi parte se podría haber mejorado con la inclusión de un chat online instantáneo por medio de una red privada IPFS como puede ser, Pubsub, Orbit Chat o Knockout. De esa manera los compañeros de clase podrian chatear y compartir los links IPFS de una manera más cómoda.

Por otro lado IPFS como ya he dicho es una tecnología relativamente nueva y tendremos que espera a que mejoren no solo implementando nuevas funcionalidad sino mejoran y rediseñando las que ya tiene.



Anexo

Definiciones

B

Bootstrap: es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.

F

Framework: es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

H

Hash: es una función computable mediante un algoritmo tal que: Tiene como entrada un conjunto de elementos, que suelen ser cadenas, y los convierte en un rango de salida finito, normalmente cadenas de longitud fija.

HTTP (Protocolo de transferencia de hipertexto): es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.

I

IPFS (InterPlanetary File System): es un protocolo y una red diseñados para crear un método p2p direccionable por contenido para almacenar y compartir hipermedia en un sistema de archivos distribuidos.

IPFS-Cluster: es una aplicación distribuida que funciona como un sidecar para los pares de IPFS, manteniendo un conjunto de pines de clúster global y asignando inteligentemente sus elementos a los pares IPFS.

Identificadores de contenido (CID): es una etiqueta que se utiliza para apuntar al material en IPFS, no indica dónde se almacena el contenido, pero forma un tipo de dirección basada en el contenido en sí.

FTP (Protocolo de Transferencia de Archivos): es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

K

Kilobyte (kB): es una unidad de almacenamiento de información y equivale a 10^3 .



P

Protocolo: es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física.

Peer to peer (p2p): es una red de ordenadores en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí.

Pubsub (publicación-suscripción): es un patrón de mensajería bastante estándar en el que los editores no saben quién, si alguien se suscribe a un tema determinado.

S

Swarm key: es una clave encriptada específicamente para las conexiones en un enjambre de nodos. Solo los nodos con el mismo swarm key se podrán comunicarse entre sí.

U

URL (Uniform Resource Locator): es un identificador de recursos uniforme (URI), cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo.

Bibliografía

IPFS

- <https://ipfs.io/>
- <https://docs.ipfs.io/install/>
- <https://labs.eleks.com/2019/03/ipfs-network-data-replication.html>
- <https://hackernoon.com/understanding-ipfs-in-depth-1-5-a-beginner-to-advanced-guide-e937675a8c8a>
- https://es.wikipedia.org/wiki/Sistema_de_archivos_interplanetario
- <https://www.itdo.com/blog/que-es-ipfs-world-wide-web-descentralizada/>
- https://medium.com/@s_van_laar/deploy-a-private-ipfs-network-on-ubuntu-in-5-steps-5aad95f7261b
- <https://discuss.ipfs.io/t/can-i-make-a-private-ipfs-network-of-public-external-ips-this-would-require-my-own-gateway-as-well/5509>
- <https://docs.ipfs.io/how-to/modify-bootstrap-list/>
- <https://docs.ipfs.io/reference/cli/#ipfs>
- <https://www.youtube.com/watch?v=C6WfDTqXGR8>
- <https://www.youtube.com/watch?v=VyGqaPjKoZ8>



- https://www.youtube.com/watch?v=oiEhyw17_OI
- <https://www.youtube.com/watch?v=5Uj6uR3fp-U&t=4s>
- <https://www.youtube.com/watch?v=BA2rHlbB5i0>
- <https://github.com/ipfs-inactive/faq/issues/230>
- <https://discuss.ipfs.io/t/how-to-create-a-private-network-of-ipfs/339>
- <https://github.com/ipfs/iptb>
- <https://discuss.ipfs.io/t/webui-not-working-in-private-network-configuration/6348/2>
- <https://github.com/ipfs/go-ipfs/issues/4174>
- <https://github.com/ipfs/go-ipfs/issues/4907>
- <http://www.programmersonsought.com/article/3906831330/>
- <https://geeks-world.github.io/articles/465599/index.html>

IPLD & IPNS & DNSLink

- <https://hackernoon.com/understanding-ipfs-in-depth-3-6-what-is-interplanetary-naming-system-ipns-9aca71e4c13b>
- <https://dnslink.io/>
- <https://hackernoon.com/understanding-ipfs-in-depth-2-6-what-is-interplanetary-linked-data-ipld-c8c01551517b>
- <https://ipld.io/>
- <https://github.com/ipld/ipld>

IPFS Cluster

- <https://cluster.ipfs.io/>
- <https://cluster.ipfs.io/cluster/>
- <https://github.com/ipfs/ipfs-cluster>
- <https://labs.eleks.com/2019/03/ipfs-network-data-replication.html>
- <https://medium.com/towardsblockchain/setting-up-your-first-distributed-private-storage-network-on-ipfs-part-1-a6ff15222b90>
- <https://stackoverflow.com/questions/52529664/setting-up-ipfs-cluster-on-docker-environment>

Otros links interesantes

- <https://medium.com/textileio/build-a-decentralized-chat-app-with-knockout-and-ipfs-fccf11e8ce7b>
- <https://blog.ipfs.io/25-pubsub/>
- <https://github.com/orbitdb/orbit-db>
- <https://github.com/orbitdb/orbit>
- <https://medium.com/textileio/build-a-decentralized-chat-app-with-knockout-and-ipfs-fccf11e8ce7b>

FCT

Bloque 1

Empresa

La empresa en la cual podría desarrollar mi proyecto de síntesis tendría que ser una empresa enfocada a las redes, ya la empresa podría dar soporte a mi proyecto no solo para crear una red descentralizada sino para desarrollar aplicaciones para la red descentralizada.

Esto podría abrir una vía a la empresa de meterse en las nueva web trayendo así muchas oportunidades negocio y desarrollo.

Además de que las empresas tienen que arriesgar e innovar en el ámbito de la tecnología, esto podría suponer una ventaja ya que sería de las primeras empresas en investigar para mejorar la web mundial no en un modelo de negocio como hoy en día sino para mejorar para todo el mundo.

Departamentos

Los departamentos que tendría que tener la empresa serían como muchas de las que ya existen hoy en día. Creo que constaría de 5 departamentos principales en los cuales serán cruciales para el crecimiento de la empresa.

- **RRHH:** para la selección de candidatos aptos para la empresa y para hacer la gestión administrativa del personal.
- **IT:** para proteger la empresa en la ciberseguridad, realizar soporte y mantenimiento a los usuarios de la empresa
- **I+D:** para identificar las oportunidades, planificar la innovación y el desarrollo de proyectos.
- **Administración y Contabilidad:** para gestionar las facturas de la empresa, liquidación de impuestos y la gestión de cobros y pagos.
- **Dirección general:** para definir qué modelo de negocio será la empresa y para organizar los diferentes departamentos de la empresa.

Rol En La Empresa

El rol que adoptaría en la empresa seguramente sería de un empleado del departamento de IT o I+D, que dispongo de conocimientos en los dos campos.

Haciendo así que pueda participar en los departamentos sin ningún inconveniente para la empresa y para mí.



Haciendo así que pudiera ganar experiencia en la industria y el sector de la informática y el desarrollo.

Esto me concedería más oportunidades no solo de ascender en la propia empresa sino también poder optar a una multinacional grande.

Objetivos

Los objetivos que espero lograr al estar en un empresa sería poder dominar mi ámbito para así no solo poder rendir más en la empresa sino también ir pudiendo ascender en la empresa debido a mis conocimientos y proyectos realizados pudiendo crecer como persona y compañero de trabajo.

Entre los conocimientos que podría aportar a la empresa sería la de programación en Java, pudiendo así desarrollar aplicaciones tanto como en móviles y ordenadores, dando versatilidad a la hora de llevar a cabo un proyecto extenso.

Además poseo conocimientos de redes que le resultaría interesantes, ya que este proyecto me ha permitido investigar sobre el blockchain.

El valor añadido que le daría a la empresa en contrarme sería en que soy una persona que puede aportar y soportar en los proyectos, ya que poseo ciertos conocimientos en el mundo de la programación y redes, además de poseer ciertas aptitudes que le podrían interesar a la empresa.

Bloque 2

Actitud profesional

Creo que poseo algunas actitudes proposicionales interesantes aunque de momento no las haya podido llevar a cabo en un puesto de trabajo en la industria y el sector de la informática.

Respecto a mi profesión siento un cierto sentimiento de aventura ya que todavía me queda una larga vida y experiencias laborales por conocer haciendo que a la hora de buscar un trabajo sienta una montaña rusa de emociones en la cual me ha sentir algunas veces de que un puesto de trabajo en concreto me podría venir a la perfección, hacerme sentir agusto o que sería el puesto de trabajo y tipo de trabajo ideal para mi.

Otra veces siento que cierto puesto de trabajo no está hecho para mi, haciendo que no sepa muy bien cómo enfrentar la situación.

Pero al final siempre mantengo una actitud positiva respecto al futuro que me espera, haciendo sentir que puedo con cualquier cosa que me proponga hacer.

Carrera profesional

La carrera profesional ideal para mí sería una donde me sienta agusto en mi puesto de trabajo y obviamente domine lo que esté haciendo para poder rendir al máximo de mi capacidad.

En una empresa espero poder desarrollarme como un empleado de programación o soporte informático aprendiendo mi oficio de forma que me permita en un par de años ascender de puesto a jefe de proyectos o coordinador de proyectos, esto sería un gran salto para mí ya que me impulsa tanto laboralmente como personalmente ya que sentiría una sensación de que voy avanzando en mi vida y los proyectos que tengo pensado.

Retos y etapas

En primer lugar para mí poder encontrar un trabajo en el mundo de la informática ya es todo un reto para mí, ya que al ser mi primera experiencia en el mundo laboral enfocado al mundo y el sector informático pues ya me supone un reto que debo superar.

Cuando consiga superar este reto conlleva que empezaría una nueva etapa para mí que obviamente también se llenaría de retos a superar para poder ser avanzando en la carrera de la vida. Y creo que después de superar este primer reto en mi vida tanto laboral como personal se vería afectado para mejor ya que sentiré que todo irá a mejor y podré empezar a vivir mi vida como de verdad me gustaría.

A partir de ahí mi objetivo en la empresa no sería ascender sino obtener el máximo que conocimientos y experiencia antes de poder ascender en la empresa o incluso cambiar a otra empresa. Después de lo dicho mi objetivo sería llegar a una empresa multinacional en la cual tenga varios departamentos y países en donde operar. Ya que esto supondría una oportunidad de viajar a otros lugares del mundo aunque sea por trabajo.

Una vez esté en una multinacional el objetivo sería ir escalando hasta llegar a un punto donde sienta que estoy muy cómodo con el entorno y la carga de trabajo que me rodean.

Proyección laboral

Mi proyección laboral como he dicho antes consta de tres partes. Una de ellas es la empresa, en la cual la me gustaría estar en una multinacional que opera en varios países del mundo ya que esto me permitiría aunque sea por trabajo pudiendo así mejorar mis capacidades a la hora de hablar un idioma o incluso aprender nuevos.

Otra es mi puesto de trabajo, el cual no considero que debería de tener un puesto alto en una empresa para ser o sentirse exitoso, ya que para mí el éxito consiste en hacer bien el trabajo y ser un buen compañero de trabajo, así que me gusta ser jefe de proyectos o coordinador ya



que me gusta dirigir proyectos, siento que tengo aptitudes de líder pero tampoco quiero sobresalir.

Y el punto final es que considero que mi ámbito es el ideal para mí, ya sea carga de trabajo, compañeros o ambiente. Si me siento agusto el dinero y los ascensos no es lo que más me importa ya que considero estar agusto y estable antes que estresado o incómodo en un puesto de trabajo.

Bloque 3

Desarrollo

En la empresa me gustaría desarrollar cualquier proyecto que sea relacionado con la programación o la administración de sistemas o incluso algún proyecto de red ya que es donde me siento más cómodo para desarrollar un proyecto.

Además de que ya que poseo conocimiento también me sé desarrolla bastante bien en este entorno porque he aprendido a buscar la información que deseo para conseguir lo que deseo, esto me hace ser un empleado que sabe resolver sus problemas antes que pedir ayuda a todas horas.

De todos modos creo que sería capaz de desarrollar cualquier proyecto o actividad con las herramientas necesarias o la ayuda necesaria ya que uno de mis puntos fuertes que aprendo rápido y se busca de forma eficaz la información que deseo.

En este caso mi proyecto se ha enfocado no en la programación sino en la red y en internet. Eso demuestra que soy polivalente para trabajar en varias actividades o sectores.

Ámbitos

Mi proyecto se centra en el ámbito de la red y comunicación, en el cual podemos comunicarnos en una red que no necesita el protocolo HTTP aunque lo soporte, IPFS es un nuevo protocolo de intercambio de información en una red descentralizada optimizando así la obtención de información y la integridad del mismo.

Mi proyecto también se podría decir que está en el ámbito de la innovación ya que como he mencionado IPFS es un nuevo protocolo desarrollado para sustituir el protocolo HTTP haciendo así que la web evolucione. Sin embargo, IPFS no está ni mucho menos terminado y todavía le queda mucho recorrido para sustituir al protocolo HTTP.

Por último también se incluye en el ámbito social, ya que IPFS se centra en hacer una internet mejor no para las empresas sino para los usuarios, permitiendo ser anónimos si se desea, que los gobiernos no puedan censurar algún contenido porque lo considere inconveniente y por último ayuda a preservar la información ya que un contenido no se puede borrar y además podemos ver las versiones anteriores que ha tenido un documento.



Beneficios a la empresa

IPFS como ya he mencionado no solo es un sucesor viable al protocolo HTTP, sino que el contenido que está en la web no es duplicada ya que IPFS no busca por donde está el contenido como lo hace el protocolo HTTP sino que el propio contenido es un hash pero que sirve como link para obtenerlo. Además que a diferencia de HTTP, IPFS permite ver las versiones de los documentos que pertenecen en la red, esto es debido a que como ya he dicho el link es el propio contenido así que si el contenido cambia también lo hace el hash.

La empresa donde trabajaba se podría beneficiar de esta tecnología de varias razones, la primera es que la información al no estar en un solo sitio como los servidores del protocolo HTTP, la información está distribuida en la red. Esto ahorraría dinero en almacenamiento a las empresas con servidores dedicados, además de que si la información está en todos los dispositivos se eliminaría la necesidad de contratar un hosting para alojar el contenido en la web, esto haría que la empresa ahorraría más dinero en el alquiler y mantenimiento de un hosting de terceros. Otra de las ventajas que podría obtener la empresa es que el contenido no podría ser censurado por los gobiernos así que toda la información está disponible en todo momento para cualquier usuario.

Por último que se beneficiaría no solo a las empresas sino también a todos los usuarios es que la información al estar distribuida por la red esto causaría un gran descenso en el consumo de ancho de banda. Esto no satura la red en ciertos puntos e incluso podríamos rebajar las cuotas contratadas o usar ese ancho de banda extra con otros propósitos.