

# La web distribuida: el protocolo IPFS

Fundamentos de Redes

---

José María Martín Luque

Adolfo Soto Werner

19 de noviembre de 2017

# Introducción

---

HTTP es el protocolo de comunicación utilizado actualmente para la transferencia de información en la web.

El desarrollo de HTTP comenzó en 1989 en el CERN por parte de Tim Berners-Lee.

La primera definición de HTTP/1.1, la versión más utilizada apareció en 1997.

Es un protocolo antiguo que no fue diseñado sin vistas al futuro y pensando Internet a menor escala y no tal y como lo vemos hoy en día.

# Problemas de HTTP y la web actual

---

# Not Found

The requested URL /hola was not found on this server.

---

*Apache/2.4.18 (Ubuntu) Server at dgiimcloud.ml Port 443*

HTTP es frágil porque para acceder al contenido se depende de una serie de servidores concretos.

Si alguno de ellos falla, no se puede acceder al contenido.

Si un servidor cambia de dirección o deja de estar disponible, los enlaces que apuntan hacia él dejan de funcionar.

El creador de Pinboard estima que alrededor del 5 % de los enlaces que se almacenan en este servicio dejan de funcionar cada año.



La web actualmente está altamente centralizada. La práctica totalidad de los usuarios de Internet dependen de una serie de servicios concretos.

Una caída del servicio de Google en 2013 provocó una reducción mundial del tráfico de Internet del 40 %.

No hay que irse muy lejos.

La *hipercentralización* también facilita el control de las comunicaciones de los ciudadanos por parte de los Gobiernos.

¿Cuál sería el coste aproximado por haber distribuido el vídeo más visto de YouTube?

(Suponiendo un coste 1 céntimo/GB distribuido y que siempre se reproduce a 720p)

El vídeo en 720p pesa 61,1MB.

El vídeo se ha reproducido al menos 4.000.000.000 de veces.

$$((61,1 \cdot 4,000,000,000) \div 1024) \cdot 0,01 = 2.621.093,75\text{€}$$

## **La web distribuida**

---

La idea de la web distribuida es eliminar la dependencia en una serie de servidores centrales haciendo que cualquier receptor sea a la vez emisor de información.

Un ejemplo en la naturaleza que ilustra esta idea es la de la ramificación de las neuronas.

Blockchain: Ethereum

Comunicación: BitMessage

Bases de datos: IPDB

Web descentralizada: Tor

Difusión y almacenamiento de información: IPFS

...



# El protocolo IPFS

---

IPFS es un sistema de archivos distribuido.

Los nodos IPFS se conectan entre sí para transferir datos.

El protocolo está dividido en un serie de sub-protocolos.

# Identidades

Gestiona la generación y verificación de la identidad de los nodos.

Cada nodo se identifica por un `NodeID` que es el *hash* de su clave pública.

IPFS no utiliza una función *hash* concreta.

```
// Hash criptográfico
type NodeId Multihash
type Multihash []byte

// Claves
type PublicKey []byte
type PrivateKey []byte

type Node struct {
    NodeId NodeID
    PubKey PublicKey
    PriKey PrivateKey
}
```

Listing 1: Definición de Nodo.

El subsistema de red de IPFS incluye funciones para:

- Transporte
- Fiabilidad
- Conectividad
- Integridad
- Autenticidad

IPFS puede utilizar cualquier red: no depende ni asume acceso a IP.

IPFS utiliza una Tabla Hash Distribuida (DSHT).

```
type IPFSRouting interface {  
    // Obtiene la dirección de un nodo concreto  
    FindPeer(node NodeId)  
  
    // Almacena un pequeño dato en la DSHT  
    SetValue(key []byte, value []byte)  
  
    // Obtiene un pequeño dato de la DSHT  
    GetValue(key []byte)  
  
    // Anuncia que el nodo puede distribuir un dato grande  
    ProvideValue(key Multihash)  
  
    // Obtiene una serie de peers distribuyendo un dato grande  
    FindValuePeers(key Multihash, min int)  
}
```

Listing 2: Interfaz de la DSHT.

## Intercambio de bloques: el protocolo BitSwap

BitSwap es el protocolo de intercambio de bloques de IPFS. Está inspirado en BitTorrent.

Los *peers* buscan conseguir un conjunto de bloques (`want_list`) y tienen otro conjunto de bloques que ofrecer (`have_list`).

BitSwap actúa como una especie de *mercado*.

Normalmente el intercambio de archivos no es complementario.

Los nodos envían bloques a sus *peers* de forma optimista.

Hay que protegerse de los *leeches*.

Un sistema de créditos resuelve el problema:

- Los *peers* realizan un seguimiento de su saldo con otros nodos.
- Los *peers* envían bloques a otros *peers* probabilísticamente.

Si un nodo decide no enviar bloques a un *peer* el nodo lo ignorará durante un tiempo.



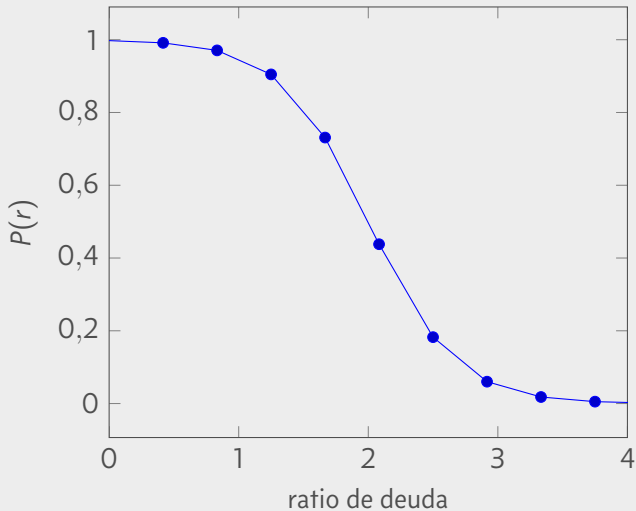
La elección de una función debe procurar varias cosas.

Una función que es efectiva en la práctica es una sigmoide, escalada por una *ratio de deuda*.

$$r = \frac{\text{bytes\_enviados}}{\text{bytes\_recibidos} + 1}$$

Dado  $r$  la probabilidad de enviar a un deudor es:

$$P(r) = 1 - \frac{1}{1 + \exp(6 - 3r)}$$



**Figura 1:** Representación de la función de probabilidad  $P(r)$ .

Los nodos mantienen un registro de las transferencias con otros nodos.

Cuando se inicia una conexión entre dos nodos intercambian su libro mayor.

Si no concuerdan, el libro mayor se destruye. Si la pérdida es intencionada puede que los nodos decidan negar el intercambio de información.

```
type Ledger struct {  
    owner NodeId  
    partner NodeId  
    bytes_sent int  
    bytes_recv int  
    timestamp Timestamp  
}
```

Listing 3: Implementación del libro mayor.

Esquema de conexión con un *peer*:

1. Conexión: los *peers* envían sus libros mayores hasta que estén de acuerdo.
2. Envío: los *peers* intercambian sus listas y los bloques.
3. Cierre: los *peers* desactivan la conexión.
4. Ignorado: un *peer* es ignorado si el nodo decide no enviar información.

La implementación del sistema de archivos de IPFS es una generalización de la estructura de datos de Git.

Utilizar un GAD de Merkle nos aporta:

- Direcccionamiento de contenido
- Resistencia a la manipulación
- Unicidad

Los objetos IPFS pueden ser recorridos con una API de rutas *string* con el siguiente formato:

```
// Formato
/ipfs/<hash-del-objeto>/<nombre-del-objeto>

// Ejemplo
/ipfs/XLYkgq61DYaQ8NhkcqyU7rLcnSa7dSHQ16x/foo.txt
```

Listing 4: Rutas en IPFS.

El objeto `blob` contiene una unidad de datos direccionable

Representa un archivo sencillo.

```
{  
  "data": "datos aquí",  
}
```

Listing 5: Estructura JSON de un `blob`.

## Archivos: `list`

El objeto `list` representa un archivo grande, formado a partir de varios blobs.

A diferencia de `blob`, `list` tiene enlaces.

```
{
  // Las listas pueden tener un array de objetos como datos
  "data": ["blob", "list", "blob"],

  // Las listas no tienen nombres en los enlaces
  "links": [
    { "hash": "nccSbCKcEWHAJ9wj0M72", "size": 38231 },
    { "hash": "2dUB2OS507rJBI57n7Hs", "size": 110 },
    { "hash": "BHHGh0pUmwiAyjN5mbTY", "size": 6468 },
  ]
}
```

Listing 6: Estructura JSON de un `list`.



El objeto `tree` representa un directorio. Es un *mapeo* de nombres a *hashes*.

```
{
  // Los árboles pueden tener un array de objetos como datos
  "data": ["blob", "list", "blob"],

  // Las listas no tienen nombres en los enlaces
  "links": [
    { "hash": "nccSbCKcEWHAJ9wj0M72", "name": "test1", "size":
      38231 },
    { "hash": "2dUB20S507rJBI57n7Hs", "name": "test2", "size":
      110 },
    { "hash": "BHHGh0pUmwiaYjN5mbTY", "name": "test3", "size":
      6468 },
  ]
}
```

Listing 7: Estructura JSON de un `tree`.

El objeto `tree` representa una instantánea en el historial de versiones de cualquier objeto.

```
{
  "data": {
    "type": "tree",
    "date": "2017-11-12 19:58:00",
    "message": "El mensaje del commit."
  },
  "links": [
    { "hash": "nccSbCKcEWHAJ9wj0M72", "name": "parent", "size": 38231 },
    { "hash": "2dUB20S507rJBI57n7Hs", "name": "object", "size": 110 },
    { "hash": "BHHGh0pUmwiAyjN5mbTY", "name": "author", "size": 6468 },
  ]
}
```

Listing 8: Estructura JSON de un commit.

Tal y como hemos descrito hasta ahora IPFS se accede a los archivos por su *hash*.

Si el contenido del archivo cambia, el *hash* también.

Necesitamos una forma de acceder a un archivo con una única ruta aunque su contenido cambie.

Para ello se define IPNS: InterPlanetary Name Space.

Asignamos a cada usuario un espacio de nombres mutable en  
`ipns/<NodeId>`.

Se pueden utilizar servicios acortadores de nombres o métodos que permiten codificar una cadena en una serie de palabras fáciles de recordar.

## **IPFS en la actualidad**

---

El 29 de abril de 2017 el Gobierno turco bloqueaba el acceso a Wikipedia en el país.

Pocos días después en el blog oficial de IPFS se anuncia que se ha realizado una instantánea de la Wikipedia turca y que estaba disponible a través de IPFS.

Actualmente se realizan copias semanales de Wikipedia en Árabe, Inglés, Kurdo y Turco.

## Web del referéndum ilegal sobre la independencia de Cataluña (2017)

La historia del referéndum ya la conocemos todos.

El Gobierno catalán utilizó IPFS para informar sobre el referéndum y permitir la consulta de los centros de votación.

Se evita el bloqueo de la web: dominio internacional y contenido distribuido.