
Blockchain-based reputation system for peer reviewing



Master's Thesis

Viktor Jacynycz García

Directed by

Samer Hassan Collado and Antonio Sánchez Ruiz-Granados

Máster en ingeniería informática

Masters degree in software engineering

Facultad de Informática

Universidad Complutense de Madrid

Curso 2017/2018

Document made with T_EX_S v.1.0+. modified by Viktor Jacynycz García

This document is prepared for duple-side printing.

Blockchain-based reputation system for peer reviewing

Memoria que presenta para optar al master en ingeniería informática

Viktor Jacynycz García

Directed by

Samer Hassan Collado and Antonio Sánchez Ruiz-Granados

**Máster en ingeniería informática
Masters degree in software engineering
Facultad de Informática
Universidad Complutense de Madrid**

Curso 2017/2018

Copyright ©Viktor Jacynycz García

Al duque de Béjar
y
a tí, lector carísimo

*I can't go to a restaurant and
order food because I keep looking
at the fonts on the menu.
Donald Knuth*

Agradecimientos

I find your lack of faith disturbing

Darth Vader, Star Wars: A New Hope.

Groucho Marx decía que encontraba a la televisión muy educativa porque cada vez que alguien la encendía, él se iba a otra habitación a leer un libro. Utilizando un esquema similar, nosotros queremos agradecer al Word de Microsoft el habernos forzado a utilizar \LaTeX . Cualquiera que haya intentado escribir un documento de más de 150 páginas con esta aplicación entenderé a quié nos referimos. Y lo decimos porque nuestra andadura con \LaTeX comenzó, precisamente, después de escribir un documento de algo más de 200 páginas. Una vez terminado decidimos que nunca más pasaríamos por ahí. Y entonces caímos en \LaTeX .

Es muy posible que hubiéramos llegado al mismo sitio de todas formas, ya que en el mundo académico a la hora de escribir artículos y contribuciones a congresos lo más extendido es \LaTeX . Sin embargo, también es cierto que cuando intentas escribir un documento grande en \LaTeX por tu cuenta y riesgo sin un enlace del tipo “*Author instructions*”, se hace cuesta arriba, pues uno no sabe por donde empezar.

Y ahí es donde debemos agradecer tanto a Pablo Gervís como a Miguel Palomino su ayuda. El primero nos ofreció el código fuente de una programación docente que había hecho unos años atrás y que nos sirvió de inspiración (por ejemplo, el fichero `guionado.tex` de \TeX IS tiene una estructura casi exacta a la suya e incluso puede que el nombre sea el mismo). El segundo nos dejó husmear en el código fuente de su propia tesis donde, además de otras cosas más interesantes pero menos curiosas, descubrimos que aún hay gente que escribe los acentos españoles con el `\’{\i}`.

No podemos tampoco olvidar a los numerosos autores de los libros y tutoriales de \LaTeX que no sólo permiten descargar esos manuales sin coste adicional, sino que también dejan disponible el código fuente. Estamos pensando en Tobias Oetiker, Hubert Partl, Irene Hyna y Elisabeth Schlegl, autores del famoso “The Not So Short Introduction to \LaTeX 2 ϵ ” y en Tomás Bautista, autor de la traducción al español. De ellos es, entre otras muchas cosas, el entorno `example` utilizado en algunos momentos en este manual.

También estamos en deuda con Joaquín Ataz López, autor del libro “Creación de ficheros \LaTeX con GNU Emacs”. Gracias a él dejamos de lado a WinEdt y a Kile, los editores que por entonces utilizábamos en entornos Windows y Linux respectivamente, y nos pasamos a emacs. El tiempo de escritura que nos ahorramos por no mover las manos del teclado para desplazar el cursor o por no tener que escribir `\emph` una y otra vez se lo debemos a él; nuestro ocio y vida social se lo agradecen.

Por último, gracias a toda esa gente creadora de manuales, tutoriales, documentación de paquetes o respuestas en foros que hemos utilizado y seguiremos utilizando en nuestro quehacer como usuarios de \LaTeX . Sabéis un montón.

Y para terminar, a Donal Knuth, Leslie Lamport y todos los que hacen y han hecho posible que hoy puedas estar leyendo estas líneas.

Resumen

...
...

... Resumen chulo

Índice

Agradecimientos	IX
Resumen	XI
I Primera parte	1
1. Introduction	3
1.1. Publications systems	3
En el próximo capítulo	3
2. Technology	5
2.1. IPFS	5
2.1.1. Distributed Hash Tables	5
2.1.2. BitTorrent - File sharing	6
2.1.3. Git - Version control system	6
II Apéndices	7
A. Anexo 1: Reuniones del equipo	9
A.1. Reunión del 07 de Septiembre de 2017	9
A.2. Reunión del 08 de Septiembre de 2017	10
Bibliografía	11

Índice de figuras

Índice de Tablas

Parte I

Primera parte

Descripción de la primera parte

Este es el backtext de la descripción de la primera parte

Capítulo 1

Introduction

Frasemolona

Vik

1.1. Publications systems

Citamos algo para que aparezca en la bibliografía...

Y también ponemos el acrónimo CVS para que no crujá.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en “release” mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 2

Technology

*The needs of the many outweigh the
needs of the few*

Spock - The Wrath of Khan

2.1. IPFS

IPFS stands for Interplanetary File System. It is a peer-to-peer file-sharing protocol that uses a cryptographic hashes to store files in a distributed network. IPFS works very similar to HTTP protocol but in a BitTorrent way. It can be seen as a giant git repository where everyone can store, share and exchange files[1].

IPFS merges three main ideas: Distributed Hash Tables, BitTorrent, Git and Self-Certified Systems.

2.1.1. Distributed Hash Tables

A distributed hash table(*DHT*) is a decentralized structure that works very similar to a hash table. Hash tables are used to identify items in a database. The table performs simple mathematical operations generating a random string called hash. The hash acts as a pointer that directs to the data, this allows the user to find data directly instead of looking through the entire database[2].

In a distributed hash table, any node can use a hash as a key to retrieve data. This system includes a data structure called “keyspace” that is a set of all possible keys, which is split up across the nodes in the system. The mapping of the keys is made by another function that describes the distance from one key to another. All the nodes have an identifier and a set of identifiers pointing to all its neighbors nodes. If a node is removed from the network, only a small portion of the data must be recovered by other nodes[2].

This system makes *DHTs* scalable, fast and robust. It is used by frameworks such as Tapestry [3], Chord [4], Kelips [5], Kademlia [6] and IPFS [1]. These platforms are similar in cost and performance if they are tested in a large enough network. They behave very fast when it comes to searching for a key through massive networks of nodes[7], that's why it is used by IPFS to create its distributed file system.

2.1.2. BitTorrent - File sharing

BitTorrent [8] is a P2P file sharing system used worldwide. In this system, files are divided into very small chunks of data, and are shared in a peer-to-peer network. Each peer aims to maximize its download rate by connecting to low latency peers. In BitTorrent's network, peers with high upload rate will get higher download rate, so the key is balancing the network bandwidth between downloading and uploading files[9].

IPFS uses three main features from BitTorrent's protocol[1]:

- BitTorrent's data exchange protocol rewards nodes who contribute to the network, and punishes the ones who don't.
- BitTorrent tracks the availability of file chunks, sending the rarest first rather than sending the most common ones.
- IPFS uses PropShare[10] bandwidth allocation strategy to improve BitTorrent's behavior facing exploitable scenarios.

2.1.3. Git - Version control system

Git is a distributed version control system (*DVCS*)[11]. Git was born in 2005 when the development process of the Linux kernel lost its version control system.

Parte II

Apéndices

Apéndice A

Anexo 1: Reuniones del equipo

...

...

RESUMEN: ...

A.1. Reunión del 07 de Septiembre de 2017

En la primera reunión del equipo se hicieron las presentaciones de los integrantes, y se discutieron las posibles ideas que se podrían implementar como proyecto en la *hackathon*. El tema principal sobre el que se discutía era el impacto social del proyecto, y que las métricas de la *hackathon* así lo exigían.

Realizamos una tormenta de ideas en la que surgieron las siguientes:

- **Plataforma de publicación de artículos académicos distribuida:** Implementar un sistema de publicación de artículos para compartir a través de la comunidad científica utilizando IPFS. Esta plataforma pretende eliminar los costes para el acceso a los artículos que imponen las empresas que se encargan de publicarlos y se benefician por ello. Implementar una plataforma totalmente descentralizada para compartir los artículos de divulgación científica conseguirá que el conocimiento de la investigación académica sea público y accesible por todos.
- **Wikipedia distribuida con modelos de gobernanza:** La idea de este proyecto inicialmente era descentralizar la plataforma de Wikipedia a través de IPFS y añadir algún modelo de gobernanza y de reputación para las revisiones de los artículos. EL problema es que es un proyecto muy complejo para implementarlo en sólo un mes, y

haría falta un equipo bastante grande y la colaboración de la propia Wikipedia para llevarlo a cabo.

- **Aplicación de contactos para homosexuales en países donde son colectivos reprimidos:** En países como Rusia, los colectivos LGBT son reprimidos hasta el punto de que expresar su sexualidad puede ser un peligro para su seguridad personal. Esta idea trataba de poner en contacto de la manera más anónima y discreta posible a esas personas sin exponerse a los riesgos que ello conlleva.
- **ONG distribuida:** Esta plataforma pretendía ofrecer una bolsa de dinero en la que las personas iban realizando donaciones. Cada semana los donantes votaban dónde se iban a invertir el dinero mediante un sistema de votos.
- **Plataforma de intercambio de conocimientos de programación distribuida:** Stack Exchange es una de las web más importantes en la comunidad informática. Esta solución propone una alternativa totalmente distribuida mediante blockchain.
- **Plataforma de toma de decisiones distribuida:** La toma de decisiones en comunidades reprimidas es bastante difícil. Mediante una aplicación de toma de decisiones en blockchain (como la que tiene Loomio), se pueden ofrecer una herramienta para que estas personas en riesgo de exclusión se hagan oír.
- **Plataforma de crowdfunding para *wistleblowers*:** El problema de las plataformas de crowdfunding es que una vez que se financia el proyecto, el usuario sólo puede ver el final del producto esperando que lo que ha financiado sea como prometen los desarrolladores. Esta plataforma propondría una alternativa con varios entregables en función del dinero que se vaya consiguiendo.

...

A.2. Reunión del 08 de Septiembre de 2017

Una vez que el equipo ha decidido el proyecto que vamos a afrontar, nos reunimos para ir decidiendo poco a poco las funcionalidades que habría de tener nuestra plataforma. Algunas de ellas son:

Bibliografía

*Y así, del mucho sleep y del poco dormir,
se le secó el cerebro de manera que vino
a perder el juicio.*

Miguel de Cervantes Saavedra

- [1] J. Benet, “Ipfs-content addressed, versioned, p2p file system,” *arXiv pre-print arXiv:1407.3561*, 2014.
- [2] A. Kaluszka, “Distributed hash tables,” 2010.
- [3] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment,” *IEEE Journal on selected areas in communications*, vol. 22, no. 1, pp. 41–53, 2004.
- [4] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.
- [5] I. Gupta, K. Birman, P. Linga, A. Demers, and R. Van Renesse, “Kelips: Building an efficient and stable p2p dht through increased memory and background overhead,” *Peer-to-Peer Systems II*, pp. 160–169, 2003.
- [6] P. Maymounkov and D. Mazieres, “Kademlia: A peer-to-peer information system based on the xor metric,” in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.
- [7] J. Li, J. Stribling, T. M. Gil, R. Morris, and M. F. Kaashoek, “Comparing the performance of distributed hash tables under churn,” in *Iptps*, vol. 4. Springer, 2004, pp. 87–99.
- [8] B. Cohen, “Incentives build robustness in bittorrent,” in *Workshop on Economics of Peer-to-Peer systems*, vol. 6, 2003, pp. 68–72.

- [9] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “The bittorrent p2p file-sharing system: Measurements and analysis,” in *IPTPS*, vol. 5. Springer, 2005, pp. 205–216.
- [10] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, “Bittorrent is an auction: analyzing and improving bittorrent’s incentives,” in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 243–254.
- [11] L. Torvalds and J. Hamano, “Git: Fast version control system,” *URL* <http://git-scm.com>, 2010.

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

*–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

