

Grupe, Nicolás

Lozada Freytes, Marcos

Almacenamiento de archivos en la nube descentralizado, basado en Blockchain

**Tesis para la obtención del título de grado de
Ingeniero de Sistemas**

Directora: Fragueiro, Carolina

Documento disponible para su consulta y descarga en Biblioteca Digital - Producción Académica, repositorio institucional de la Universidad Católica de Córdoba, gestionado por el Sistema de Bibliotecas de la UCC.



[Esta obra está bajo una licencia de Creative Commons Reconocimiento-No Comercial-Sin Obra Derivada 4.0 Internacional.](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Trabajo Final

Grupe, Nicolás - Lozada Freytes, Marcos



**UNIVERSIDAD
CATÓLICA
DE CÓRDOBA**
JESUITAS

Ingeniería de Sistemas

Facultad de Ingeniería

Universidad Católica de Córdoba

Año 2019

Índice

Índice	1
Presentación del tema	5
Resumen	6
Abstract	6
Diagnóstico	7
Objetivos	9
General	9
Globales	9
Específicos	10
Marco teórico	11
Campo de acción	12
Análisis de mercado	12
Elaboración de la encuesta	13
Sección 1	13
Sección 2	13
Sección 3	14
Resultados obtenidos de la encuesta	14
Plataformas existentes	19
QubistryFS	19
Sia	22
Conclusión	23
Tecnologías analizadas para el desarrollo	23
Blockchain	23
DApp (aplicación descentralizada)	25
Ethereum	26
Solidity	26
Smart Contracts	27
Etherscan	28
Remix IDE	29
ConsenSys	29
ConsenSys Labs	30
MetaMask	31

Truffle Suite	31
Truffle	31
Ganache	32
Infura	33
IPFS - InterPlanetary File System	34
JavaScript	35
jQuery	36
Lodash	36
Web3.js	37
Node.js	38
npm (Node Package Manager)	39
CryptoJS	40
Webpack	40
HTML	41
HTML5	41
CSS	42
CSS3	43
Bootstrap	44
TerserPlugin	44
Babel	45
Git	46
GitHub	46
Trello	47
Metodología ágil utilizada para el desarrollo: Scrum.	48
Visual Studio Code (VS Code)	49
Propuesta de solución	51
Alcance funcional	52
Definición de actores	52
Requerimientos	52
Requerimientos funcionales	53
Requerimientos no funcionales	53
Casos de Uso	55
Usuario	55
Use Case: Editar perfil	56
Use Case: Ver listado de archivos	56
Use Case: Subir archivo	57
Use Case: Cifrar archivo	58
Use Case: Descargar archivo	58

Use Case: Descifrar archivo	59
Use Case: Generar cuenta	60
Diagrama de flujo	61
Subir un archivo	61
Descargar un archivo	62
La aplicación	63
Pantallas de la aplicación del usuario	63
Home	63
Home Pop Up Login MetaMask	64
Home Pop Up MetaMask	65
Home con selector de archivos	66
Home Pop UP MetaMask Confirmar Transacción	67
Home Pop UP MetaMask Rechazar Transacción	68
Home con archivos descargados	69
About Us	70
Enviar correo electrónico	71
Testing	73
Beneficios post-implantación	73
Modelo de Negocio	74
Descripción de los bloques	75
Propuesta de valor	75
Segmento de clientes	76
Canales	76
Relación con los clientes	77
Actividades clave	77
Recursos clave	77
Asociados clave	77
Estructura de costos	78
Vías de ingreso	78
Impacto Social	79
Posibilidades de Crecimiento	80
Conclusión	82
Anexos	83
Test Cases - Balloon	83
Anteproyecto	102

Diagrama de Gantt	105
Glosario	106
Fuentes	107

Presentación del tema

En la actualidad existen diversas aplicaciones para almacenar archivos en la nube. La primera de ellas fue Dropbox y al tiempo se le sumaron otras como OneDrive de Microsoft, iCloud de Apple y la más popular en la actualidad, Google Drive. Todas permiten almacenar cualquier tipo de archivos (documentos, pdf, videos, fotos, etc) en internet. Esto trae grandes beneficios, y entre ellos, que no hace falta tener que llevar tu computadora o tu pendrive a otro lado si no que se puede consultar los archivos desde cualquier lado que tengas acceso a internet. Por lo general, estas plataformas permiten ocupar cierta memoria de forma gratuita y en algunos casos sin ningún costo. Pero ¿por qué nos permitirían almacenar información, ocupando espacio en sus servidores, de forma gratuita?

Algunos podrán considerar que nada es gratis en la vida, y en ese caso, ¿qué estamos brindando a estas empresas a cambio de espacio de almacenamiento en sus servidores? La respuesta es: NUESTRA INFORMACIÓN.

Estas grandes empresas almacenan nuestra información, que se las estamos brindando nosotros mismos, en sus servidores, permitiendo así que ellos sepan qué cosas nos gustan y de esa forma poder mostrarnos anuncios que puedan llegar a interesarnos por ejemplo, y quién sabe qué más pueden llegar a hacer con ella.

El presente trabajo tiene como objetivo desarrollar una plataforma web, denominada "Balloon". Esta plataforma permitirá, al usuario, subir archivos a la nube pero en una red descentralizada (utilizando la Blockchain de Ethereum). Es decir, no va a almacenar la información del usuario en servidores de las empresas anteriormente mencionadas si no que en nodos (dispositivos que formen parte de la red descentralizada como computadoras, celulares, etc). Este tipo de productos se los denomina Dapp por su significado en inglés (Decentralized App - aplicación descentralizada).

Asimismo, la Dapp, brinda la posibilidad al usuario, de tener sus archivos cifrados y encriptados generando una seguridad superior a las nubes centralizadas, y permitiendo tener una correcta y rápida administración de los mismos por el usuario en su cuenta y que no haya alteración alguna del archivo.

El proyecto surge por iniciativa propia en base a creencias y diferencias encontradas a nivel de almacenamiento de archivos en nube (Cloud). Además, se trabajó buscando todos los enfoques técnicos necesarios para poder implementar las herramientas obtenidas gracias a los conocimientos y aptitudes adquiridas a lo largo de toda nuestra formación profesional.

Resumen

Hoy en día, mucha gente utiliza los servicios de nube para subir sus archivos personales, como Google Drive, iCloud, entre otras. Si bien estas grandes empresas permiten que nuestra información sea “privada”, nada los frena a ellos a acceder a nuestros archivos y aprender un poco de nosotros para luego poder ofrecernos, en otros sitios que son de las mismas empresas, mostrarnos publicidades que puedan llegar a interesarnos.

Nosotros decidimos buscar una alternativa que nos permita no almacenar nuestra información en sus servidores y, como entusiastas de la tecnología de Blockchain, realizar un proyecto que consiste en desarrollar un sistema de almacenamiento de archivos en nube descentralizado, utilizando para este fin protocolos existentes que utilizan tecnología P2P (Peer-to-Peer) y Blockchain: IPFS (InterPlanetary File System) y Ethereum (ETH).

Abstract

Now a days, lots of people uses cloud services to upload personal files, such us Google Drive, iCloud, between others. Even though this large companies allow us to upload information in a secure way, nothing stops them to access our files and learn something about us so then they can show us, in their other websites, publicity that we may like, for example.

We decided to search for an alternative way that can allow us to store information not using their servers and, as we are enthusiasts about Blockchain technology, realize a project that consists on developing a system that will allow us to store files in the cloud but in a decentralized way, using for this existing protocols with the technology P2P (Peer-to-Peer) and Blockchain: IPFS (InterPlanetary File System) and Ethereum.

Diagnóstico

Hoy en día la desconfianza a la que se enfrenta la gente al utilizar internet como medio para almacenar su información, en la mayoría de los usuarios y de los casos es demasiado alta.

Una persona muchas veces cree que las corporaciones como Google, Facebook y demás no hacen uso de la información de sus usuarios, pero eso deja mucho que pensar. En primer lugar, resulta complejo encontrar una forma de administrar que datos o información usan de nosotros estas partes o entidades, muchas veces damos click en “Aceptar terminos y condiciones” sin saber que por detrás ellos aclaran que pueden hacer uso de nuestros datos o información y nosotros no leemos, pero nos enfrentamos a tales cuestiones o inseguridades que pueden atentar hacia nuestra información de la que somos dueños y no es de nuestro interés que otros la utilicen.

Luego de una larga puesta en mesa de diferentes cuestiones que merodeaban en torno a esta temática y haciendo referencia a la seguridad de la información, habiendo estudiado en los últimos años la tecnología Blockchain como base de datos para tener una buena trazabilidad de las transacciones que en ella se llevan a cabo, almacenamiento seguro y transparente, de gran fiabilidad para el usuario, nos pusimos el desafío de ver si se podía utilizar la misma para enfrentar y darle una solución óptima a esta problemática de los usuarios actuales y que todos enfrentamos en el día a día.

Como la utilización de Blockchain hoy en día hace uso de varios métodos de seguridad, las aplicaciones que existen suelen no ser tan amigables para los usuarios tradicionales, por lo que día tras día se intenta crear nuevas implementaciones que preserven la seguridad pero que sean cada vez más User Friendly, es decir fáciles de usar para el usuario final (independientemente de quién sea) sin perder los principios que viene a traer la Blockchain.

En definitiva y en base a lo relevado y analizado con diferentes terceros, al ver que era posible llevar a cabo la creación de un almacenamiento descentralizado y que aún no había sido desarrollado de la manera en que nosotros lo deseábamos hacer, utilizando Blockchain, particularmente la de Ethereum (ETH) que permite la utilización de Smart Contracts (Contratos Inteligentes) como administrador de transacciones e IPFS (InterPlanetary File System) para

almacenar la información de manera segura sin intermediarios y promocionando el principio Peer to Peer (Entidad a Entidad) justamente para evitar desfases de la información o uso por terceros que no son de la incumbencia del usuario.

En base a estas tecnologías, y con uso de herramientas de Consensys Labs, se apuntó lograr el desarrollo web de un almacenamiento descentralizado basado en estas tecnologías sin perder los principios de las mismas y enfocado en la seguridad del usuario y de su información que hoy en día es muy valiosa.

A modo de conclusión de los problemas, y para que el entendimiento de la realidad que abordaremos sea el correcto, los separaremos en dos grandes grupos. Por un lado, el que presentan los usuarios que almacenan su información, y por el otro, el que presentan las entidades que permiten almacenar la información.

- Usuarios que desean almacenar su información:
 - Desconfianza a la hora de almacenar su información por miedo que otro la utilice.
 - Si la información del usuario está en la nube, ¿se puede perder?
 - No le alcanza el espacio disponible para almacenar los archivos al usuario.
 - Al aceptar los términos y condiciones, ¿el usuario puede confiar en que su información seguirá siendo privada?

- Entidades que permiten almacenar información:
 - Cobran por sus servicios, sobre todo por el espacio de almacenamiento. Pero los usuarios no se dan cuenta que les están brindando su información, que es mucho más cara.
 - No garantizan la integridad y seguridad de los datos de sus usuarios.
 - Existen prejuicios y noticias sobre grandes entidades como Facebook o Google que utilizan la información de sus usuarios, generando desconfianza de estos hacia estas entidades centralizadas nombradas aquí arriba como muchas otras más.

Con estas cuestiones bien claras nos vimos motivados a tratar de sacar estas entidades intermediarias que almacenan los datos en sus servidores de los cuales ellos son los administradores, como por ejemplo Google y Facebook, entre otros. Buscando quitar a estas empresas del medio y promoviendo el tráfico Peer to Peer (P2P) es que decidimos ir tras el diseño de esta Dapp a fines de almacenar descentralizadamente archivos de usuarios los cuales ellos son los únicos dueños, se encuentran encriptados y guardados en nodos a los que solo el usuario tiene la posibilidad de acceder cuando tenga interés o necesidad de hacerlo.

Objetivos

General

El objetivo de esta dapp es permitirle al usuario poder tener un almacenamiento de archivos en la nube descentralizado, basado en blockchain, de su interés; los mismos serán cifrados y quedarán almacenados en una red descentralizada llamada IPFS, que mediante un Smart Contract alojado en la Blockchain de ETH, almacene un índice al archivo guardado permitiendo tener una correcta y rápida administración de los mismos por el usuario en su cuenta y que no haya alteración alguna del archivo.

Globales

- Brindar una plataforma que facilite la gestión segura y almacenamiento de archivos para los usuarios.
- Fomentar la utilización de Blockchain como herramienta para almacenar información.
- Eliminar entidades intermediarias a la hora de alojar archivos, que puedan llegar a hacer uso de los mismos.
- Concluir con las problemáticas relacionadas a la seguridad de la información o desconfianza de los usuarios que almacenan archivos en la nube.
- Brindar un caso de uso fácil de la tecnología Blockchain, sin ser difícil de utilizar por los usuarios, evitando que sea poco amigable la interacción y sin eliminar los principios que viene a traer esta tecnología.

Específicos

- Asegurar al usuario un almacenamiento 100% privado y fiable.
- Garantizar el registro de cuando un archivo ha sido guardado, siendo la transacción 100% visible a través del timestamp (sello de tiempo/fecha) de la misma.
- Garantizar que los archivos almacenados permanecen en la red IPFS mediante el Smart Contract que se aloja en la Blockchain de ETH.
- Garantizar que el proyecto funcione en la Testnet de ETH.
- Ver transacciones en tiempo real a través de Etherscan.
- Desarrollar un proyecto completo de desarrollo de software para aplicar todo lo aprendido durante la carrera.
- Elaborar una aplicación web Open Source, con intención de buscarle una manera de capitalizar en un futuro.
- Lograr entender e interiorizar el proceso de desarrollo de software, desde la etapa de requerimientos hasta la puesta en productivo de la aplicación final.
- Aprender a utilizar las tecnologías actuales para el desarrollo web, más específicamente en torno al Blockchain.
- Conocer nuevas herramientas que se puedan utilizar en un futuro para la administración y gestión de proyectos.
- Trabajar respetando las condiciones y tiempos de entrega predeterminados, similar a una experiencia profesional completa y real.

Marco teórico

Al comienzo de este proyecto, buscamos si hasta el momento existían algunas aplicaciones o páginas web, ya sean nacionales o internacionales, que resolvieran nuestro principal problema, almacenar archivos de un usuario particular en la nube de manera fiable, sin intermediarios y utilizando una red descentralizada de nodos que alojen una Blockchain existente (en nuestro caso ETH por sus beneficios) con registros de transacciones sobre archivos guardados.

El resultado de la búsqueda fue positivo. Encontramos algunas soluciones nacionales, aunque ninguna de ellas cubría de manera completa nuestro problema, y las que más se acercaban no funcionaban de la manera a la que apuntaba nuestro modelo de negocios. Por otro lado, encontramos una propuesta funcional y activa en Europa, la cual parte sobre una idea similar a la nuestra que es el almacenamiento de archivos en blockchain, pero no utiliza los mismos principios ya que utilizan una blockchain creada por ellos, no como este trabajo que pretende utilizar una blockchain con prestigio a fin de brindar seguridad y evitar posibles altercados en la información y con nuestra propuesta hacia el usuario.

En la sección “Plataformas”, se realizará una breve reseña de cada una de ellas, haciendo hincapié en las fallas o puntos débiles de las alternativas nacionales, que nuestra aplicación deberá corregir o mejorar para convertirse en una solución genuina a la problemática planteada al comienzo de este proyecto. Esta investigación y análisis de alternativas preliminares generó una mayor motivación para desarrollar el proyecto. Principalmente debido a dos factores, el primero es la falta de un producto local que cubra las necesidades que planteamos. Mientras que el segundo factor es el hecho que a nivel global no existe una aplicación planteada de manera similar con características similares o los mismos principios y las grandes entidades que almacenan en la nube poseen una cantidad de usuarios interesante lo que nos motiva a ir detrás de esta idea.

Campo de acción

Almacenar archivos en la nube es algo que está en constante crecimiento, cada vez más personas lo hacen, y cada vez, cada persona almacena más archivos. Mientras más memoria tenemos más la vamos a ocupar. Podemos ver esto bastante claro en que hay varias grandes empresas que ofrecen estos servicios. Entre ellas, la plataforma que más se utiliza es la que ofrece Google, Google Drive, que hace un par de años, decidieron separar lo que era el almacenamiento de fotos del resto de archivos y ahí es cuando apareció Google Fotos que te permite almacenar tus fotos sin límite de espacio, por lo que deja el espacio de almacenamiento de tu Google Drive libre para los otros tipos de archivos.

Con esto en mente, hay que tener ahora en cuenta las personas, ya sean físicas o jurídicas, que deseen empezar a almacenar información confidencial, privada. Sobre todo, las empresas que deseen utilizar almacenamiento en nube sobre archivos muy delicados seguramente tengan sus dudas en proveer este tipo de información a los servidores de Google o cualquier otra de estas grandes empresas. Ahí es donde entra nuestra propuesta, el almacenamiento en nube descentralizado para que la información no sea almacenada en servidores centralizados brindando la confidencialidad y seguridad que ese tipo de archivos necesitan.

Análisis de mercado

En orden de saber si otras personas pensaban como nosotros y qué opinaban de acuerdo de nuestra posible solución, se realizó una encuesta con la herramienta Google Forms de Google. Dicha encuesta se compartió por distintos grupos de Whatsapp intentando diversificar en las personas que la respondieron, en distintas edades y de distintos rubros, tecnológicos y no tecnológicos, todo esto para mejorar la calidad de las respuestas de la encuesta.

Elaboración de la encuesta

La encuesta fue dividida en 3 secciones redireccionando a las personas que respondieron en caso de que sus respuestas sean negativas a la hora de preguntar si conocían cuestiones técnicas como Blockchain. Dentro de cada sección distintas preguntas.

Sección 1

Almacenamiento Descentralizado

El objetivo de esta dapp es permitirle al usuario de la misma poder tener un almacenamiento descentralizado de archivos de su interés, los mismos podrán ser encriptados o no a la hora de guardarse y quedarán almacenados en una red descentralizada con un Smart Contract que almacene un índice al archivo guardado permitiendo tener una correcta administración de los mismos y que no haya alteración alguna del archivo.

- ¿Confiarías en una aplicación para guardar archivos de suma importancia en internet?
 - Si
 - No
- Utilizas algún servicio de almacenamiento en la nube como Google Drive, iCloud, DropBox?
 - Si
 - No
- Estas de acuerdo que empresas como Google tengan acceso a los archivos que guardas en su nube?
 - Si
 - No

Sección 2

- ¿Estás al tanto de la tecnología de Blockchain?
 - Si
 - No
 - Solo un poco
- ¿Sabes lo que es un Smart Contract?
 - Si
 - No
 - Solo un poco
- ¿Entendés el concepto de bases de datos descentralizadas?
 - Si

- No
- Tal vez

Sección 3

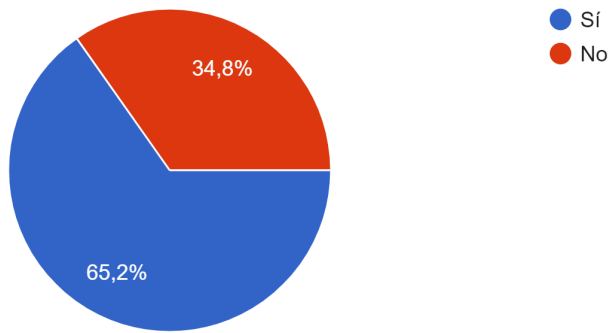
- Utilizarías un nuevo almacenamiento en la nube que te garantice el 100% de confidencialidad de archivos sin permitir que empresas como Google accedan a tu información?
 - Si
 - No
 - No me interesa
- Pagarías por subir tus archivos a este nuevo almacenamiento en la nube descentralizado?
 - Si
 - No
 - Tal vez

Resultados obtenidos de la encuesta

La encuesta fue respondida por más de 100 personas y se obtuvo una gran diversidad de respuestas, de gran valor para la investigación. La mayoría de las respuestas confirmó la hipótesis de la necesidad de tener como opción una plataforma que permita el almacenamiento en nube descentralizado. Se puede observar en los siguientes gráficos las respuestas de los encuestados.

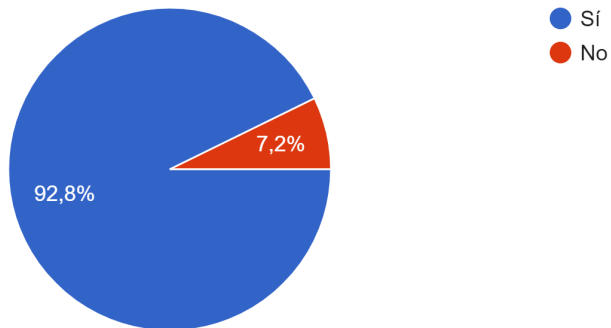
Confiarías en una aplicación para guardar archivos de suma importancia en internet?

138 respuestas



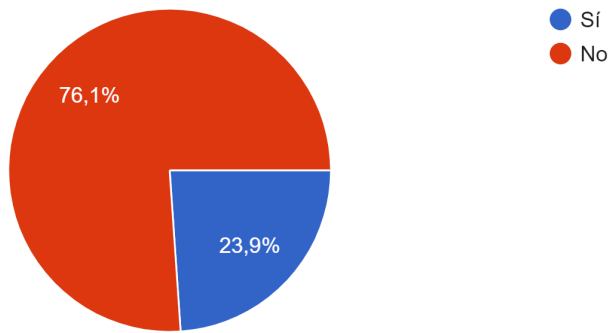
Utilizas algún servicio de almacenamiento en la nube como Google Drive, iCloud, DropBox?

138 respuestas



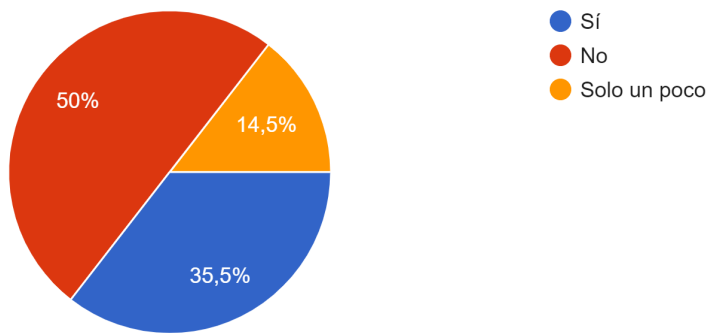
Estas de acuerdo que empresas como Google tengan acceso a los archivos que guardas en su nube?

138 respuestas



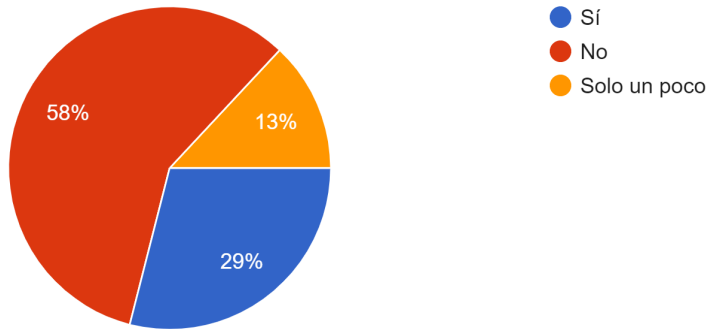
Estas al tanto de la tecnología de Blockchain?

138 respuestas



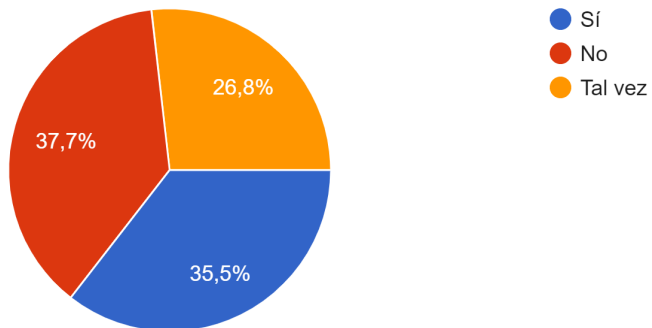
Sabes lo que es un Smart Contract?

138 respuestas



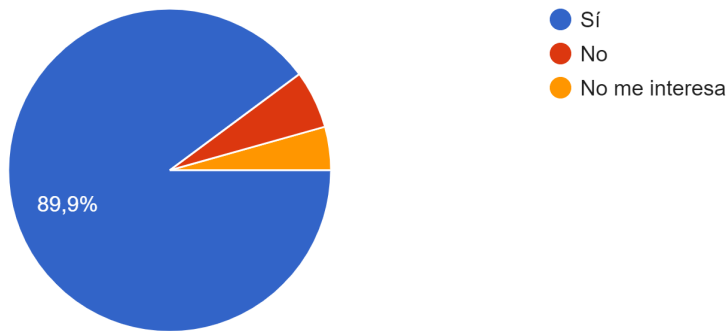
Entendés el concepto de bases de datos descentralizadas?

138 respuestas



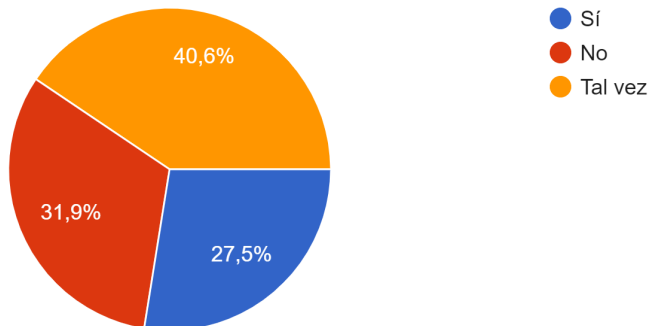
Utilizarías un nuevo almacenamiento en la nube que te garantice el 100% de confidencialidad de archivos sin permitir que Google accedan a tu información?

138 respuestas



Pagarías por subir tus archivos a este nuevo almacenamiento en la nube descentralizado?

138 respuestas



Los datos finales que se obtuvieron y observan, son que mucha gente actualmente almacena archivos en la nube, más precisamente el 92,8% de nuestros encuestados, en plataformas de Google, Dropbox y otras. Muchos, un 76,1% no están de acuerdo con no conocer las intenciones de las grandes empresas al brindarles nuestra información a cambio de espacio en sus servidores. El 50% de las personas no conoce lo que es la tecnología de Blockchain y los beneficios que tiene, como los Smart Contracts son el 58%, y unos pocos

entienden de cuando hablamos de descentralización de la información (el 37,7%). Pero la mayoría (el 89,9%) está de interesada en una plataforma de almacenamiento con esta tecnología. Sin embargo, el 31,9% no están dispuestos a pagar por mantener sus archivos de la forma más segura y confidencial, pero del 68,1% restantes, son el 40,6% los que tal vez pagarían y solo el 27,5% afirma que si.

En conclusión, como propulsores a desarrollar e implementar un sistema de almacenamiento descentralizado orientado hacia el segmento propuesto (Blockchain, red IPFS, Smart Contracts, entre otras tecnologías), se observó como positivo que las personas que lo utilizarían, quieran y gocen un sistema así, ya que les brindaría excelente seguridad a sus archivos, confianza personal en el sistema y tranquilidad a sus pensamientos o prejuicios de “quien es el dueño” de su información, que muchas veces no es de interés para los usuarios que alguien vea aparte de ellos la misma. A su vez, se tomó como amenaza que no conozcan la tecnología Blockchain, ya que es algo realmente nuevo y novedoso que viene a traer grandes prestaciones, pero se sabe que las personas de a poco va a ir profundizando más en el tema, juntando mas ganas de adentrarse en este segmento tecnológico de base de datos seguras y distribuidas.

Plataformas existentes

QubistryFS

Es una sección de la página web de Qubistry¹ para permitir a los usuarios que naveguen el sitio subir archivos a una blockchain de test.

Este sistema no está planteado para que diferentes usuarios tengan sus archivos en su propia cuenta, las transacciones se realizan en un mismo sitio y con una misma clave para subir los archivos. Se suelen apreciar al navegar por la sección algunos pequeños bugs que llevan a que el usuario piense que no funciona la pagina y vuelva a cargarla.

¹ <https://www.qubistry.co/#>

Si bien QubistryFS implementó un apartado en un sitio web para que los usuarios suban imágenes principalmente y las mismas queden almacenadas en una blockchain, el mismo presenta ciertas contras como que todos los que entran a la página pueden ver los archivos subidos y si disponen de la clave puedan abrirlos con el link al mismo. Tampoco existe la funcionalidad de de encriptado de archivos ni la de subir diferentes tipos de archivos.

A continuación veremos cómo es el apartado que el usuario ve al deslizarse por la página de Qubistry:



Qubistry^{FS}

¿Querés guardar un archivo o una foto en Blockchain?

Podemos desde este nuevo módulo de QubistryFS guardar un archivo o una foto en blockchain y probar la función para luego recuperar dichos datos desde el IPFS y el hash del Blockchain. Nos podremos dar cuenta entonces que la información guardada, no solo lo estará para siempre, sino que también nadie la podrá modificar. ¿Querés compartir una foto con un amigo? Subila al blockchain y envíale luego el código para que la pueda ver, y así ya serían dos más que usan el blockchain como sistema descentralizado de información. Cuando te pida password utilizá 1234. En esta aplicación estamos usando el Testnet de Ethereum Rinkeby, y el Gas para enviar la información corre por cuenta nuestra.

[Subí un archivo](#)

Una vez que el usuario selecciona en “Subí un archivo”, la página redirige a la persona hacia otra sección en la que el mismo puede completar los siguientes datos:



El sistema QubistryFs, nos irá mostrando los siguientes datos a medida que la gente va guardando fotos o archivos, todo en una misma página, para todos las personas que hayan interactuado en el sitio:

- Título documento
- Fecha y hora de la transacción
- Wallet address firmante
- Hash SHA256 del documento o foto
- Descargar desde IPFS

Sia

Sia² es una plataforma de almacenamiento en la nube basada en blockchain y en la descentralización. Sia permite que cualquier persona o entidad pueda alojar o acceder a almacenamiento encriptado a través de la plataforma. Utilizando el libro público de blockchain se verifican todos los contratos, las pruebas de almacenamiento y las transacciones.

Sia divide, cifra y distribuye los datos a través de su red descentralizada para almacenarlos. Sia permite que los usuarios “alquilen” el espacio de su disco duro que no utilizan. De esta forma el coste general de almacenamiento se reduce, mientras que en otras plataformas centralizadas cobran aproximadamente 10 dólares mensuales el TeraByte, en Sia 1 Tera se puede conseguir por 1 o 2 dólares al mes.

Los servicios de la red se pagan con la propia moneda, Siacoin. Es decir, cualquiera que desee usar espacio de almacenamiento de Sia deberá adquirir la moneda para pagar con ella. Y aquellos que ofrecen su espacio libre del disco duro recibirán a cambio la misma moneda.

Ambas partes pueden establecer un acuerdo sobre el almacenamiento y para eso redactar un contrato de archivo, que es un tipo de smart contract. A diferencia de los datos de los usuarios que se almacenan en los discos duros de los proveedores, los contratos permanecen guardados en la blockchain de Siacoin. Estos contratos de archivo definen los términos de un acuerdo de almacenamiento y aseguran que cada parte cumpla con las obligaciones ya establecidas. El contrato original es verificable e inmutable una vez que se ha enviado a la red.



² <https://sia.tech/>

Conclusión

De los proyectos analizados anteriormente, el que más se parece a nuestra idea de solución es Sia en cuanto al modelo de negocio que ofrece, donde cada usuario tiene sus archivos almacenados sin que nadie pueda acceder a ellos si no es con su clave privada.

A diferencia de Sia, en este trabajo no se creará una red propia de Blockchain si no que se utilizará la Blockchain de Ethereum que ya tiene una espalda en el rubro, junto con sus Smart Contracts, lo que se considera más seguro y asegura no haya inconvenientes o alteraciones con el sistema. Tampoco se ofrecerá a los usuarios intercambiar almacenamiento de sus ordenadores o nodos, en cambio se usará una red de nodos llamada Infura, de Consencys Labs.

Respecto a QubistryFs, si bien existen tecnologías en común, ellos ofrecen una plataforma para compartir archivos con otros usuarios, por lo que los modelos de negocio son diferentes.

Tecnologías analizadas para el desarrollo

En la siguiente sección se detallarán las tecnologías que se decidieron incorporar en el Trabajo Final dado a que se consideran las más novedosas y óptimas en torno a las temática afrontada.

Para esta sección se ha tomado como base libros, foros como StackOverflow o Consensys para tomar herramientas a fin de hacer el desarrollo lo más profesional posible y con los mejores instrumentos a la hora de poner en pie la idea.

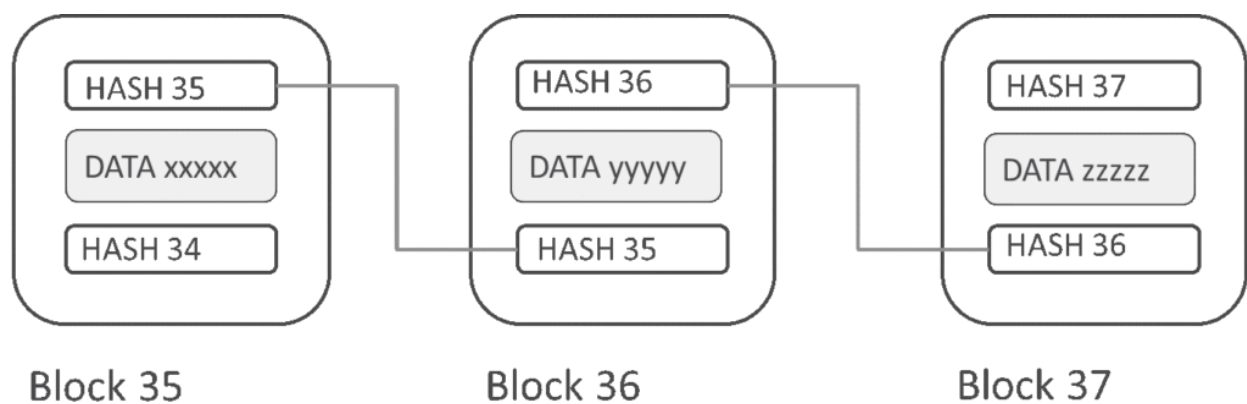
Blockchain

Una Blockchain o cadena de bloques es una estructura de datos utilizada para crear un libro contable descentralizado. Una Blockchain se compone de bloques de forma serializada. Un bloque contiene un conjunto de transacciones, un hash del bloque anterior, marca de tiempo (Timestamp, que indica cuándo se creó el bloque), recompensa de bloque (block reward), número de bloque, etc. Cada bloque contiene un hash del bloque anterior, creando así una

cadena de bloques vinculados entre sí. Cada nodo en la red contiene una copia de la cadena de bloques.

La prueba de trabajo (Proof of Work), la prueba de participación (proof of stake), etc, son varios protocolos de consenso utilizados para mantener la cadena de bloques segura. Dependiendo del protocolo de consenso, los bloques se crean y agregan a la Blockchain de manera diferente. En la prueba de trabajo, los bloques se crean mediante un procedimiento llamado minería, que mantiene segura la cadena de bloques. En el protocolo de prueba de trabajo, la minería implica resolver acertijos complejos. La Blockchain en la red de Ethereum contiene transacciones de ETH. Los ether se suministran a la red recompensando nuevos ETH a los nodos que extraen bloques con éxito.

La principal ventaja de la estructura de datos de blockchain es que automatiza la auditoría y hace que una aplicación sea transparente y a su vez segura. Puede prevenir el fraude y la corrupción. Se puede usar para resolver muchos otros problemas dependiendo de cómo lo implemente y lo use.



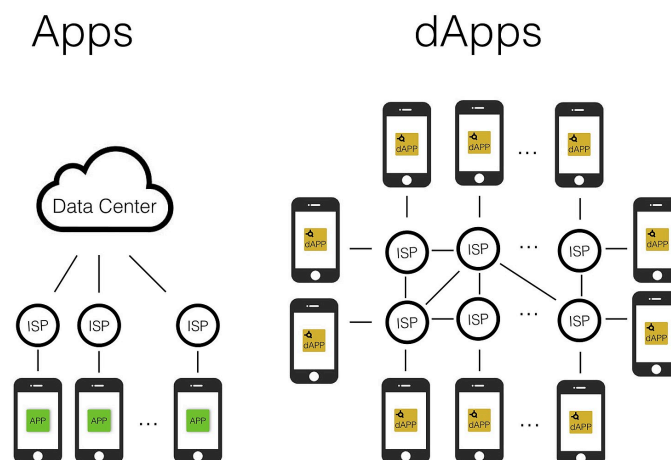
Dentro de las ventajas que ofrece la tecnología Blockchain se encuentran la seguridad, la distribución y descentralización, privacidad, transparencia e integridad y confianza. La confianza entre los pares que transaccionan se da gracias a que confían en la tecnología, no necesitan confiar en el otro. Lo de transparencia se refiere a que es público el libro de transacciones realizadas, pero con la privacidad de que cada transacción se realiza entre 2 claves, sin información del usuario por detrás de una de ellas. Cada usuario cuenta con 2

claves, una privada, con la cual el usuario puede ingresar a sus datos y saber cuánto dinero tiene y puede realizar transacciones, y una clave pública, la cual se usa para que otros hagan una transacción a tu cuenta.

Cuando se habla de la seguridad que ofrece la tecnología Blockchain es porque para alterar las transacciones se deberían cambiar los enlaces, o hash, de las transacciones en un bloque, el cual también contiene el hash del bloque anterior y a su vez se deberían modificar todos los hash posteriores hacia atrás para que estos coincidan, ya que al cambiar un solo carácter cambia todo el hash y por ende todo debería cambiar y coincidir, y todo esto se debería lograr modificar antes que un nuevo bloque se cierre (que son aproximadamente 5 minutos en Ethereum). La manera en que las transacciones son aceptadas es a partir de los mineros que trabajan en la red, ellos buscan cerrar los bloques ya que al hacerlo reciben una recompensa en criptomoneda.

DApp (aplicación descentralizada)

Una Dapp es una aplicación que corre en una red distribuida, con información que participa de manera segura y protegida, y operaciones ejecutadas en un entorno descentralizado a través de una red de nodos. Una aplicación distribuida es código corriendo en una red P2P. Sus principios son: open source, autónomo, seguro (dado a que los archivos están almacenados en la Blockchain), descentralizado, automatizado.



Ethereum

Ethereum es una plataforma descentralizada, que permite implementar DApps sobre ella. Los contratos inteligentes se escriben utilizando el lenguaje de programación Solidity. Los DApps se crean utilizando uno o más contratos inteligentes. En Ethereum, los contratos inteligentes se pueden escribir en varios lenguajes de programación, incluidos Solidity, LLL y Serpent. Solidity es el más popular de esos idiomas. Ethereum tiene una moneda interna llamada ether (ETH). Para implementar contratos inteligentes o llamar a sus métodos, se necesita ether. Puede haber múltiples instancias de un contrato inteligente como cualquier otra DApp, y cada instancia se identifica por su dirección única. Tanto las cuentas de usuario como los contratos inteligentes pueden contener ether.

Ethereum utiliza la estructura de datos Blockchain y el protocolo de consenso de prueba de trabajo (Proof of Work). Se puede invocar un método de contrato inteligente mediante una transacción o mediante otro método. Hay dos tipos de nodos en la red: nodos regulares y mineros. Los nodos regulares son los que solo tienen una copia de la cadena de bloques, mientras que los mineros construyen la cadena de bloques mediante la minería de bloques.



Solidity

Solidity es un lenguaje de programación de alto nivel cuya síntesis es similar a otro de los lenguajes de programación más usados hoy en día: Javascript. Este lenguaje está diseñado

para crear y desarrollar contratos inteligentes que se ejecuten en la Máquina Virtual Ethereum (EVM de sus siglas en inglés).

Este lenguaje es denominado “Turing Complete”, este concepto, ideado por Alan Turing, se refiere, en informática, a aquel lenguaje que tiene un poder computacional equivalente a lo que se denomina una “Máquina de Turing Universal”. Término que se refiere al software que es lo suficientemente hábil como para ejecutar cualquier código definido por el desarrollador.

El IDE oficial que se utiliza para escribir smart contracts con Solidity se llama Remix, sin embargo se encuentran otros IDEs que son compatibles como por ejemplo: Ethereum Studio, Plugin IntelliJ IDEA, Atom, y además se puede agregar un plugin a Visual Studio Code para que sea compatible.



Smart Contracts

Los smart contracts son contratos o acuerdos en un bloque. Hoy en día hay un montón de contratos comunes y corrientes en negocios y compañías. La diferencia de los contratos inteligentes con los contratos comunes es que no hace falta la participación de organizaciones de terceros, siendo el contrato forzado por el código mismo. Un contrato inteligente de Ethereum es un programa que se ejecuta en Ethereum. Un contrato inteligente se ejecuta exactamente como se programó sin ninguna posibilidad de tiempo de inactividad, censura, fraude e interferencia de terceros.

En este ámbito es muy importante destacar la descentralización permitiendo que partes que no tengan confianza entre sí, o no se conozcan, puedan transaccionar entre ellos de manera confiable.

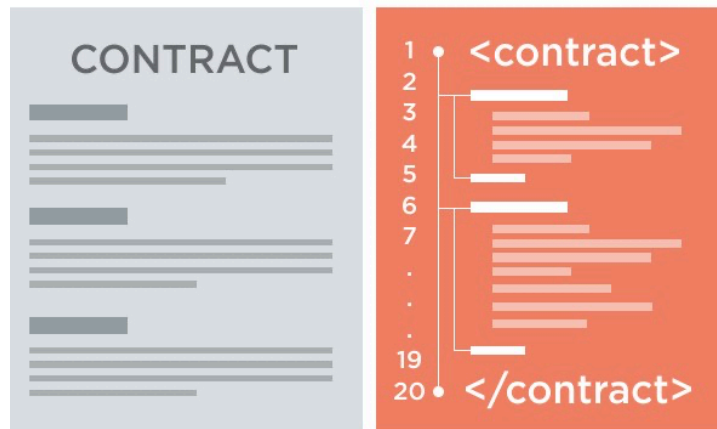
Un contrato inteligente tiene un input y un output (válido o inválido), y funciona bajo esta norma solamente. Cuando este contrato está en la Blockchain es inmutable porque el código lo protege, este código dice como el proceso va a ser - "Code is law" (el código es la ley).

Las ventajas de los contratos inteligentes:

- son automáticos (sin terceras partes, eficientes, rápidos),
- incorruptibles,
- independientes (P2P),
- y su certeza.

Y su principal desventaja es:

- el contrato no es modificable.



Etherscan

Etherscan es una plataforma Block Explorer (Exploradora de bloques), Search (Búsqueda), API (application programming interface o interfaz de programación de aplicaciones) y Analytics (Analíticas) para Ethereum, la plataforma descentralizada de contratos inteligentes (Smart Contracts). Permite ver transacciones en la Blockchain de Ethereum (ETH) en tiempo real, ver si se han realizado correctamente, si se han cancelado y demás!



Remix IDE

Remix IDE³ es una herramienta open source, que permite escribir smart contracts en el lenguaje de Solidity directo desde el browser (Chrome, Firefox, etc.). Remix fue programado en JavaScript y es por eso que soporta tanto su uso en el browser como de manera local. También, Remix permite hacer pruebas (testing), depuración (debugging) y desplegar (deploying) de los smart contracts.



ConsenSys

ConsenSys⁴ es una compañía global pura y exclusivamente dedicada a blockchain. Desarrollan aplicaciones empresariales, invierten en startups, crean herramientas de desarrollo que pertenecen a un área de la empresa llamada "ConsenSys Labs" y ofrecen a su vez educación en blockchain brindando cursos pagos en esta tecnología en "ConsenSys Academy".

³ <http://remix.ethereum.org/>

⁴ <https://consensys.net>



ConsensusSys Labs

ConsensusSys⁵ Labs es el portafolio de herramientas de ConsensusSys, apoyando a emprendedores y desarrolladores de todo el mundo a través de herramientas, talento y capital que ellos ponen en las manos de estos a fin de conseguir más profesionales e interesados en el rubro del Blockchain.

Dentro de las múltiples herramientas brindadas por ConsensusSys a través de ConsensusSys Labs se encuentran:

- MetaMask
- Truffle Suite (Truffle, Ganache y Drizzle)
- uPort



MetaMask

MetaMask⁶ es una aplicación para los browsers que les permite visitar la web distribuida. Le permite ejecutar Ethereum DApps directamente en su browser sin ejecutar un nodo Ethereum

⁵ <https://labs.consensys.net/portfolio/>

⁶ <https://metamask.io/>

completo. MetaMask permite mantener la identidad del usuario de forma segura, ya que proporciona una interfaz de usuario para administrar sus identidades en diferentes sitios y firmar transacciones de blockchain. Se puede instalar el complemento MetaMask en Chrome, Firefox y Opera.

La misión de MetaMask es hacer que Ethereum sea lo más fácil de usar para la mayor cantidad de personas posible lo cual lo asemeja mucho a nuestra idea de proyecto en poder acercar estas tecnologías que parecen tan distantes y difíciles de usar a cualquier posible usuario.



Truffle Suite

Truffle suite⁷ consiste de 3 herramientas (Truffle, Ganache y Drizzle) que vienen a solucionar y a hacer más amigable el desarrollo de aplicaciones con la tecnología de Blockchain de Ethereum. En el desarrollo de esta Dapp de almacenamiento se utilizarán 2 de estas 3 herramientas: Truffle y Ganache.

Truffle

Es un ambiente de desarrollo, testing y pipeline para Blockchain usando la máquina virtual de Ethereum (EVM), intentando facilitar las cosas al desarrollador. Con Truffle se obtiene:

- Compilación de contratos inteligentes incorporada, vinculación, implementación y gestión binaria.
- Pruebas automáticas de contratos para un rápido desarrollo.
- Marco de implementación y migración extensible y programable.

⁷ <https://www.trufflesuite.com/>

- Gestión de redes para implementar en cualquier número de redes públicas y privadas.
- Gestión de paquetes con EthPM y NPM, utilizando el estándar ERC190.
- Consola interactiva para comunicación directa por contrato.
- Canal de construcción configurable con soporte para una estrecha integración.
- Runner de script externo que ejecuta scripts dentro de un entorno Truffle.



Ganache

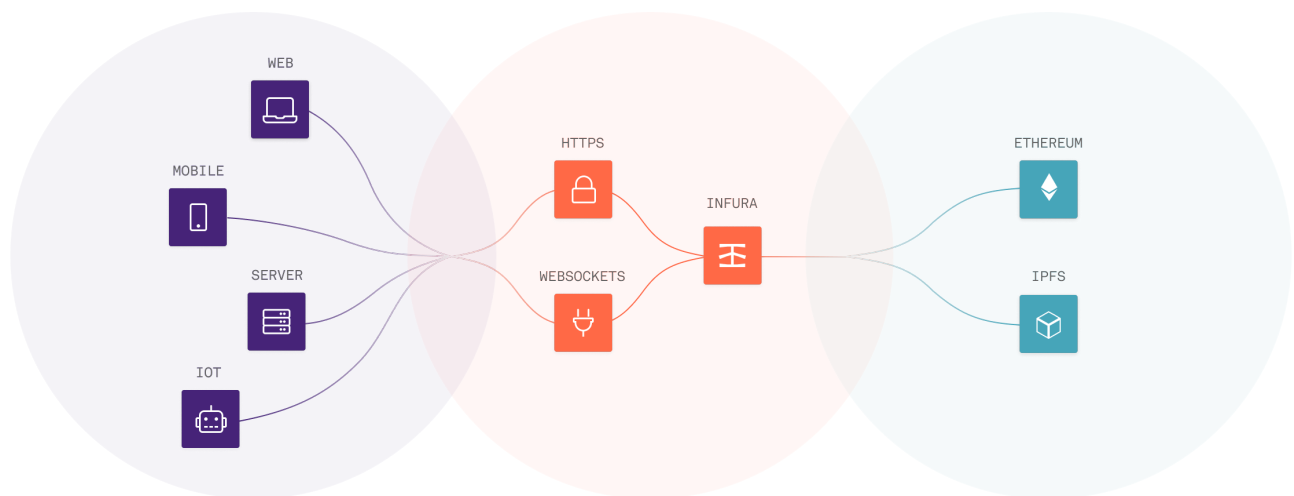
Ganache es una Blockchain personal para desarrolladores de Ethereum que se puede utilizar para deployar contratos, desarrollar las aplicaciones y correr las pruebas. Está disponible como aplicación desktop (o GUI) y también como herramienta de comando de línea (o CLI). Está disponible para Windows, Mac y Linux.



Infura

Infura es una API que proporciona acceso instantáneo a través de HTTPS (Hypertext Transfer Protocol Secure, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP) y WebSockets (es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP) a las redes Ethereum e IPFS. La infraestructura que brinda para su aplicación en aplicaciones de la Web 3.0 es sencilla y amigable para cualquier tipo de usuario.

A continuación se ve un diagrama de como funciona y se comporta el API de Infura resumidamente:



*Imagen obtenida de: <https://infura.io/product>

INFURA

IPFS - InterPlanetary File System

IPFS (InterPlanetary File System) es un sistema de archivos descentralizado. IPFS⁸ utiliza estructuras de datos DHT (tabla hash distribuida) y Merkle DAG (gráfico acíclico dirigido). Utiliza un protocolo similar a BitTorrent para decidir cómo mover los datos por la red. Una de las características avanzadas de IPFS es que admite el control de versiones de archivos. Para lograr el control de versiones del archivo, utiliza estructuras de datos similares a Git.

En IPFS, cada nodo no contiene todos los archivos, sino que almacena los archivos que necesita. Por lo tanto, si un archivo es menos popular, obviamente muchos nodos no lo tendrán; por lo tanto, existe una gran posibilidad de que el archivo desaparezca de la red. Debido a esto, muchas personas prefieren llamar a IPFS una aplicación descentralizada de intercambio de archivos punto a punto (peer to peer o P2P). O bien, puede pensar en IPFS como BitTorrent, que está completamente descentralizado; es decir, no tiene un rastreador y tiene algunas características avanzadas.

Por ejemplo, una descripción general de cómo funciona IPFS sería: cuando se almacena un archivo en IPFS, se divide en fragmentos <256 KB y se generan hashes de cada uno de estos fragmentos. Los nodos en la red contienen los archivos IPFS que necesitan y sus hashes en una tabla hash.

Hay cuatro tipos de archivos IPFS: blob, list, tree y commit. Un blob representa una porción de un archivo real que se almacena en IPFS. Un archivo de tipo list o lista representa un archivo completo ya que contiene la lista de blobs y otras listas. Como las listas pueden contener otras listas, ayuda en la compresión de datos a través de la red. Un tree o árbol representa un directorio ya que contiene una lista de blobs, listas, otros árboles y commits. Y un archivo de confirmación representa una instantánea en el historial de versiones de cualquier otro archivo. Como las listas, los árboles y las confirmaciones tienen enlaces a otros archivos IPFS, forman un DAG Merkle.

Entonces, cuando se quiere descargar un archivo de la red, solo se necesita el hash del archivo de lista IPFS. O si se quiere descargar un directorio, entonces solo se necesita el hash del archivo de árbol IPFS.

⁸ <https://ipfs.io>

Como cada archivo se identifica por un hash, los nombres no son fáciles de recordar. Si se actualiza un archivo, entonces se necesita compartir un nuevo hash con todos los que quieran descargar ese archivo. Para abordar este problema, IPFS utiliza la función IPNS (Inter-Planetary Name System), que permite que los archivos IPFS se señalen con nombres autocertificados o nombres amigables para las personas.



JavaScript

Es un lenguaje de programación interpretado, es decir, que no requiere de compilación sino que es analizado por otro programa, específicamente los navegadores web. Javascript está diseñado para correr en un navegador. En sus comienzos, este lenguaje fue utilizado principalmente para ejecutar acciones del lado del cliente, aunque actualmente también existen formas de utilizarlo del lado del servidor. Su principal objetivo fue agregar interactividad a las páginas web, es decir, dotarlas de cierta lógica para que las acciones ejecutadas por el usuario actualicen la información presentada en la página. En este sentido, le permite a los desarrolladores añadir efectos especiales, dinamismo y vistosidad a las páginas web.

Por otra parte se debe señalar que Javascript está definido como un lenguaje de programación orientado a objetos basado en prototipos, es decir, que en js no existen las clases y por ende el concepto de herencia es distinto. También, se caracteriza por ser débilmente tipado, no hace falta especificar el tipo de dato que se almacena.



jQuery

jQuery es una librería rápida, pequeña y llena de características de JavaScript. Hace que cosas como el desplazamiento y la manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores. Permite que manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX sea más fácil



Lodash

Lodash es una biblioteca de JavaScript que proporciona funciones de utilidad para tareas de programación comunes utilizando el paradigma de programación funcional. Lodash facilita la programación con JavaScript ya que elimina la molestia de trabajar con matrices, arreglos, números, objetos, cadenas, etc. Los métodos modulares de Lodash son excelentes para: iterar de matrices, objetos y cadenas, manipular y probar valores, y crear funciones compuestas.

LO

Web3.js

web3.js es una colección de librerías de JavaScript que te permite interactuar con un nodo remoto o local de Ethereum usando una conexión del tipo HTTP o IPC. La Web3.js interactúa con la cadena de bloques Ethereum. Puede recuperar cuentas de usuario, enviar transacciones, interactuar con smart contracts, entre otras cosas.



Node.js

Node.js⁹ es un entorno en tiempo de ejecución multiplataforma, basado en Javascript y orientado a eventos asíncronos (hace referencia al suceso que no tiene lugar en total correspondencia temporal con otro suceso), diseñado para construir aplicaciones en red escalables (que tienden a crecer con el pasar del tiempo). Esto se debe a su gran capacidad

⁹ <https://nodejs.org/es/>

para manejar conexiones que suceden al mismo tiempo, por cada una de ellas se ejecutará una función que recibe el nombre de “callback”¹⁰.

Node dispone de módulos, similares a librerías Javascript, que son un conjunto de funciones para incluir en la aplicación que se esté desarrollando. Trae por defecto un conjunto de módulos incorporados, que se pueden utilizar sin necesidad de instalación. Uno de ellos es el módulo HTTP, el cual permite transferir datos sobre “Hyper Text Transfer Protocol” (HTTP) y también crear un servidor HTTP que escucha a los puertos del servidor y responde al cliente.

También dispone de un manejador de paquetes llamados “npm” que se instala en el mismo momento que se instala Node. Un paquete en Node.js contiene todos los archivos que se necesitan para que un módulo funcione.



npm (Node Package Manager)

npm¹¹ es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript. Consiste en un conjunto de herramientas esenciales de desarrollo de JavaScript que ayudan a los desarrolladores a ir al mercado más rápido y crear aplicaciones potentes utilizando el código fuente moderno.

npm consta de tres componentes diferentes:

¹⁰ [https://es.wikipedia.org/wiki/Callback_\(informática\)](https://es.wikipedia.org/wiki/Callback_(informática))

¹¹ <https://www.npmjs.com>

- el sitio web (para descubrir paquetes, configurar perfiles y administrar otros aspectos de su experiencia npm. Por ejemplo, puede configurar Orgs (organizaciones) para administrar el acceso a paquetes públicos o privados).
- la interfaz de línea de comandos (CLI - esta se ejecuta desde una terminal, y así es como la mayoría de los desarrolladores interactúan con npm).
- el registro (es una gran base de datos pública de software JavaScript y la metainformación o “meta-information” que lo rodea).

npm se puede usar para adaptar los paquetes de código para nuestras aplicaciones o incorporar los paquetes tal como están, descargar herramientas independientes que se pueden usar de manera inmediata, ejecutar paquetes sin descargar usando npx (este sirve para ejecutar binarios del paquete npm), compartir código con cualquier usuario de npm, en cualquier lugar, restringir el código a desarrolladores específicos, crear Orgs (organizaciones) para coordinar el mantenimiento de paquetes, la codificación y los desarrolladores, formar equipos virtuales usando Orgs, administrar múltiples versiones de código y dependencias de código, actualizar las aplicaciones fácilmente cuando se actualice el código subyacente, descubrir múltiples formas de resolver el mismo rompecabezas y encontrar otros desarrolladores que estén trabajando en problemas y proyectos similares.



CryptoJS

CryptoJS es una colección, cada vez mayor, de algoritmos criptográficos estándar y seguros implementados en JavaScript utilizando las mejores prácticas y patrones. Son rápidos y tienen una interfaz consistente y simple.

Esta librería de JavaScript incluye algoritmos de hash como MD5, SHA-1, SHA-256¹² a SHA-512, entre otras y algoritmos criptográficos como AES¹³, DES, Triple DES, Rabbit, RC4 y algunas más.



Webpack

Webpack es un paquete de módulos JavaScript open source. Este paquete puede transformar activos de front-end como HTML, CSS e imágenes si se incluyen los complementos correspondientes. En otras palabras, webpack es un paquete de módulos estáticos para aplicaciones JavaScript modernas open source. Cuando webpack procesa la aplicación, internamente crea un gráfico de dependencia que asigna cada módulo que el proyecto necesita y genera uno o más paquetes.



HTML

HTML significa lenguaje de marcado de hipertexto o Hyper Text Markup Language, HTML es el lenguaje de marcado estándar para páginas web. Los elementos HTML son los componentes

¹² <https://medium.com/biffures/part-5-hashing-with-sha-256-4c2afc191c40>

¹³ <https://www.techradar.com/news/what-is-aes>

básicos de las páginas HTML¹⁴. Estos elementos HTML están representados por tags o etiquetas <>

HTML es el lenguaje principal de la Web para crear contenido permitiendo que todos lo usen en cualquier lugar y de forma universal.

HTML fue creado por Sir Tim Berners-Lee a fines de 1991, pero no se lanzó oficialmente, hasta que se publicó en 1995 como HTML 2.0. HTML 4.01 se publicó a fines de 1999 y era la versión principal de HTML.

HTML es un lenguaje de marcado muy evolutivo y ha evolucionado con varias versiones actualizadas. Mucho antes de que se incorporen sus estándares y especificaciones revisados, cada versión ha permitido a su usuario crear páginas web de una manera mucho más fácil, con excelentes diseños y hacer que los sitios web sean muy eficientes.



HTML5

HTML5 (lanzado en 2012) es la última versión del lenguaje de marcado de hipertexto o mejor dicho HTML, el código que describe las páginas web. En realidad, son tres tipos de código: el HTML que proporciona la estructura, las hojas de estilo en cascada (CSS, que se describe más adelante) que se encargan de la presentación y JavaScript (del que ya se habló previamente), que hace que las cosas sucedan.

HTML5 ha sido diseñado para ofrecer casi todo lo que desea hacer en línea sin necesidad de software adicional, como complementos del navegador. Hace de todo, desde animación

¹⁴ <https://www.w3.org/html/>

hasta aplicaciones, música y películas, y también se puede utilizar para crear aplicaciones increíblemente complicadas que se ejecutan en su navegador.



CSS

Las hojas de estilo en cascada (CSS o Cascading Style Sheets) son un mecanismo simple para agregar estilo (por ejemplo, fuentes, colores, espacios) a los documentos web.

Con el pasar de los años, se han lanzado al mercado diferentes versiones de CSS, cada una presentando mejoras significativas en torno a las capacidades de manejar el estilo de los desarrollos, las cuales serán listadas aquí abajo:

- CSS (1996) permite al usuario seleccionar el estilo y tamaño de fuente y a su vez cambiar el color del texto y el fondo.
- CSS2 (1998) tiene capacidades que permiten al usuario diseñar el diseño completo de la página.
- CSS3 (1999) permite al usuario crear presentaciones a partir de documentos y seleccionar entre una gama más amplia de fuentes, incluidas las de Google y Typecast. Excepcionalmente, CSS3 permite al usuario incorporar bordes redondeados y usar múltiples columnas. Se considera que CSS3 es más fácil de usar (en comparación con CSS2) porque tiene diferentes módulos más completos y específicos.



CSS3

CSS3 básicamente realiza cambios en cómo algunos elementos visuales son implementados y representados por un navegador. Sin embargo, no es una única especificación enormemente difícil de manejar, a diferencia de CSS2. CSS3 se separa en módulos separados para facilitar el desarrollo. Esto significa que la especificación sale en fragmentos, con módulos más estables que otros.

Algunos de los principales módulos de CSS3 son: Modelo de caja (Box model), valores de imagen y contenido reemplazado, efectos de texto, selectores (Selectors) , fondos y fronteras (Borders), animaciones, interfaz de usuario (IU), diseño de múltiples columnas y transformaciones 2D / 3D.



Bootstrap

Es una herramienta que nos permite crear proyectos responsive o sensibles a cada interfaz para dispositivos móviles en la web con una biblioteca de componentes front-end super completa y la más popular del mundo.

Bootstrap es un kit de herramientas de código abierto para desarrollar con HTML, CSS y JS (JavaScript). Permite al usuario desarrollador crear prototipos rápidamente o crear toda su aplicación con variables Sass (Syntactically Awesome Style Sheets o hojas de estilo sintácticamente impresionantes) y mixins (es una clase que ofrece cierta funcionalidad para ser heredada por una subclase, pero no está ideada para ser autónoma), sistema de cuadrícula receptiva/sensible (responsive grid system), componentes precompilados extensos y complementos potentes creados en jQuery.



TerserPlugin

Un analizador de JavaScript y un kit de herramientas de compresión para ES6 +. ECMAScript v6 (Abreviado como ES6 o ES2015) es el estándar que sigue JavaScript desde Junio de 2015. TerserPlugin es un minificador de código y eso logra reducir el tamaño de los archivos de código, lo que permite que su transmisión a través de internet sea más rápido.



Babel

Babel es un compilador de JavaScript, consiste en una cadena de herramientas que se utiliza principalmente para convertir el código ECMAScript 2015+ en una versión de JavaScript compatible con versiones anteriores en navegadores o entornos actuales y anteriores. Estas son las principales cosas que Babel puede hacer por los usuarios o desarrolladores:

- Transformar la sintaxis.
- Funciones de Polyfill que faltan en su entorno de destino (a través de `@babel/polyfill`).
- Transformaciones de código fuente (codemods¹⁵).

Estos complementos permiten usar una nueva sintaxis, en el presente actual, sin esperar la asistencia del navegador o que salgan actualizaciones. A su vez, Babel intenta usar la menor cantidad de código posible sin depender de un tiempo de ejecución voluminoso.



¹⁵ <https://www.sitepoint.com/getting-started-with-codemods/>

Git

Git¹⁶ es un software gratis y open source de control de versiones. Está pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente y un gran número de colaboradores. La idea principal de Git es llevar registro de los cambios en archivos de un sistema de software y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Git permite clonar repositorios de códigos y crear nuevas ramas a partir de la rama principal, de esa forma permite que siempre haya una rama principal con el código estable y las ramas con nuevas funcionalidades sean añadidas una vez que funcionen de la forma correcta.



GitHub

GitHub¹⁷ es una plataforma web de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, teniendo ya sea una cuenta paga o una cuenta de estudiante.

GitHub permite entonces alojar tu repositorio de código y brinda varias herramientas. Entre las herramientas están el Fork (clonar) y el Pull request. Ya que es por lo general de uso público, otros desarrolladores pueden hacer Fork de tu código para solucionar algún Bug y luego hacer un Pull request, donde le mandan al dueño del repositorio los cambios realizados para que analice si desea realizar esos cambios o no en la rama principal del código.

¹⁶ <https://git-scm.com/about>

¹⁷ <https://github.com/features>



Trello

Trello¹⁸ es una herramienta colaborativa que permite llevar la administración y gestión de proyectos, simulando un tablero con post-its, donde se puede obtener de forma fácil y rápida las tareas asignadas a cada uno, el estado en que están, qué está realizando el resto y en qué parte del proceso completo está cada integrante del grupo. Trello no deja de ser más que un sistema que brinda una ayuda y mayor organización para determinar lo que se tiene que hacer en determinado momento.

Trello es un tablero que está distribuido por columnas, llamadas listas. Cada lista se compone de tarjetas, que representan las tareas a realizar, o que ya fueron realizadas. A su vez, Trello permite integrar otras aplicaciones para brindar más información a cada tarjeta. Entre algunas de estas aplicaciones se puede nombrar Github, que permite relacionar la tarjeta con un commit en un proyecto, adjuntar archivos a partir de Google Drive, checklist para comprobar el proceso de cumplimiento de la propia tarjeta, entre muchas otras opciones.



¹⁸ <https://trello.com/es>

Metodología ágil utilizada para el desarrollo: Scrum.

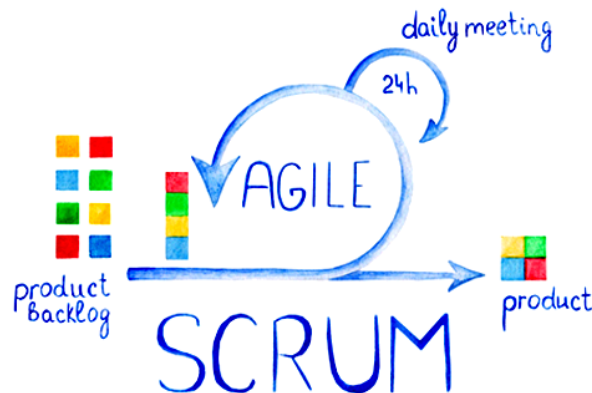
Scrum es una metodología ágil para el desarrollo de software, la cual utiliza Kanban para organizar las distintas historias de usuario y dentro de ellas las tareas a llevar a cabo por los desarrolladores.

Las metodologías ágiles son nuevas formas de organizar el trabajo que buscan adaptarlo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para adaptar el proyecto y su desarrollo a las circunstancias específicas del entorno.

Scrum busca agregar valor en cada etapa, haciendo que con cada una se obtenga siempre un producto que se acerque cada vez más a lo que el cliente desea. La idea de aplicar Scrum es poder establecer un proceso de desarrollo basado en la mejora continua, donde cada etapa produce un aprendizaje que servirá como entrada para la próxima etapa.

Como se mencionó anteriormente, Scrum se basa en dividir por etapas el proceso de desarrollo. En cada una de ellas se llevan a cabo ciertos actos o ceremonias que buscan establecer Scrum de la forma más óptima posible. Entre ellas se pueden encontrar:

- Daily: actividad que se realiza todos los días, en donde cada miembro del equipo cuenta como va con el desarrollo de su tarea. La duración no debe superar los 10/15 minutos.
- Meeting: actividad que se realiza al finalizar el sprint, que generalmente duran entre 1 y 2 semanas, en donde se hace una 'Revisión', 'Retrospectiva' y 'Planificación'.
- Revisión: actividad en donde se exponen las tareas finalizadas en el sprint.
- Retrospectiva: actividad en la que cada miembro del equipo cuenta cómo fue su experiencia en ese sprint, diciendo cosas que le parecieron buenas, si le molestó algo o cosas que se podrían mejorar.
- Planificación: actividad en donde se toman y se analizan las historias que se trabajarán en el próximo sprint.



Visual Studio Code (VS Code)

Visual Studio Code¹⁹ es un editor de código fuente, se caracteriza por ser ligero pero muy potente y performante. Este se ejecuta en el escritorio del usuario y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go, Solidity) y tiempos de ejecución (como .NET y Unity) .

Dentro de las características principales que esta herramienta brinda a sus usuarios, se encuentran:

- Brinda una herramienta llamada IntelliSense, resaltando la sintaxis y autocompletando el código, esta tool proporciona terminaciones inteligentes basadas en tipos de variables, definiciones de funciones y módulos importados.
- Código de depuración directamente desde el editor. Permite iniciar o adjuntar a sus aplicaciones en ejecución y depurar con puntos de interrupción, pilas de llamadas y una consola interactiva.
- Comandos Git incorporados, trabajar con Git y otros proveedores de SCM nunca ha sido tan fácil para los desarrolladores que utilicen esta herramienta.
- Extensible y personalizable. Permite instalar extensiones para agregar nuevos idiomas, temas, depuradores y para conectarse a servicios adicionales. Las

¹⁹ <https://code.visualstudio.com>

extensiones se ejecutan en procesos separados, lo que garantiza que no pondrán lento el editor.

- Deployar e implementar con confianza y facilidad (a través de Microsoft Azure).



Visual Studio Code

Propuesta de solución

Ante el masivo interés de las personas por almacenar archivos en la nube y el crecimiento de esta actividad con el correr de los años desde que se ha puesto en práctica, y a su vez con la inseguridad que hoy existe en internet y más aún con la información que se suele regalar a grandes compañías a cambio de espacio en sus servidores, permitiéndoles recaudar millones con esta y los usuarios perdiendo su intimidad, dignidad, privacidad y demás, es que definitivamente se decidió desarrollar Balloon, un almacenamiento descentralizado disruptivo.

Este se trata de un una aplicación web diseñada para almacenar archivos del usuario, el mismo, podrá subir sus archivos de interés bajo un mínimo costo y acceder a los mismos para que sean descargados de manera gratuita de por vida las veces que el usuario quiera, con su clave del cual él será el único poseedor y utilizando los datos para loguearse correctos podrá acceder a la plataforma adecuadamente y ver su contenido subido y alojado en la aplicación desde cualquier dispositivo, en cualquier parte del mundo.



Alcance funcional

Con Balloon se quiere ofrecer una solución para todo el tipo de usuarios, desde Crypto Holders hasta personas jóvenes o adultas que deseen almacenar archivos en la nube y no tienen interés en que un tercero disponga de su información. A su vez se brindará acceso sencillo a las interacciones con una Blockchain, desde la tenencia de una billetera virtual hasta la solicitud y observación de transacciones en tiempo real con una Blockchain, particularmente la de Ethereum (ETH).

Los usuarios tendrán un formulario de carga en el que seleccionaran un input del ordenador que será el archivo, sea cual sea, que este desee subir y almacenar en la nube, una vez confirmada la transacción y pagado un precio minúsculo para que el procesamiento de la operación se lleve a cabo por la red de nodos colaboradores y que sustentan la red y le dan respaldo, este quedara guardado y aparecerá en la tabla de archivos subidos del usuario logueado. Este usuario, luego podrá descargar los archivos previamente subidos en el ordenador, respondiendo a su demanda cuando el lo solicite. Sólo él dispondrá de acceso a estos archivos ya que los mismos en el momento de carga son encriptados y descryptados en el proceso de descarga, utilizando la información de la billetera digital del usuario logueado, permitiendo que sólo él sea el apto para hacer estas operaciones o solicitudes.

Definición de actores

Los usuarios de Balloon serán todos aquellos interesados en almacenar un archivo en la nube de manera descentralizada. Se tendrá un solo tipo de actor ya que las interacciones se realizarán por usuarios que se comportarán de la misma manera al querer tener sus diferentes archivos en la aplicación web almacenados en una red IPFS con links a los mismos alojados en la Blockchain de Ethereum (ETH) para su descarga cuando el usuario lo solicite.

Requerimientos

En esta sección serán detallados los requerimientos (propiedades o restricciones), determinados con precisión, que el producto de software Balloon va a satisfacer. Serán desglosados en dos diferentes secciones, por un lado, los requerimientos funcionales, que

hacen referencia a aquellos servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Cuando se habla de las entradas, no necesariamente se hace referencia a las entradas de los usuarios solamente. Pueden ser interacciones con otros sistemas, respuestas automáticas, procesos predefinidos.

En algunos casos, los requerimientos funcionales de los sistemas también establecen explícitamente lo que el sistema no debe hacer. Por el otro se encuentran los requerimientos no funcionales, haciendo referencia a aquellos que no se refieren directamente a las funciones específicas brindadas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad, performance, latencia. En palabras más fáciles, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace. Alternativamente, definen restricciones del sistema tales como la capacidad de los dispositivos de entrada/salida y la representación de los datos utilizados en la interfaz del sistema.

Requerimientos funcionales

- Registrar nuevo usuario en Balloon.
- Mostrar una lista de archivos subidos por el usuario.
- Almacenar nuevo archivo en Balloon.
- Cifrar nuevo archivo para ser subido a la nube.
- Descifrar archivo al ser descargado por el usuario.
- Descargar archivo desde la nube, por solicitud del usuario.
- Editar perfil del usuario, desde MetaMask.
- Marcar transacción como completada, el archivo queda subido en la Dapp.
- Login obligatorio para cada usuario específico que desee ingresar a Balloon.
- Seguridad de la cuenta otorgada por la billetera digital, MetaMask.
- Seguridad de los archivos otorgada por los hash y algoritmos de encriptación (AES) utilizados a la hora de subir o descargar archivos.
- Aplicación web, que permite almacenar archivos de manera descentralizada.

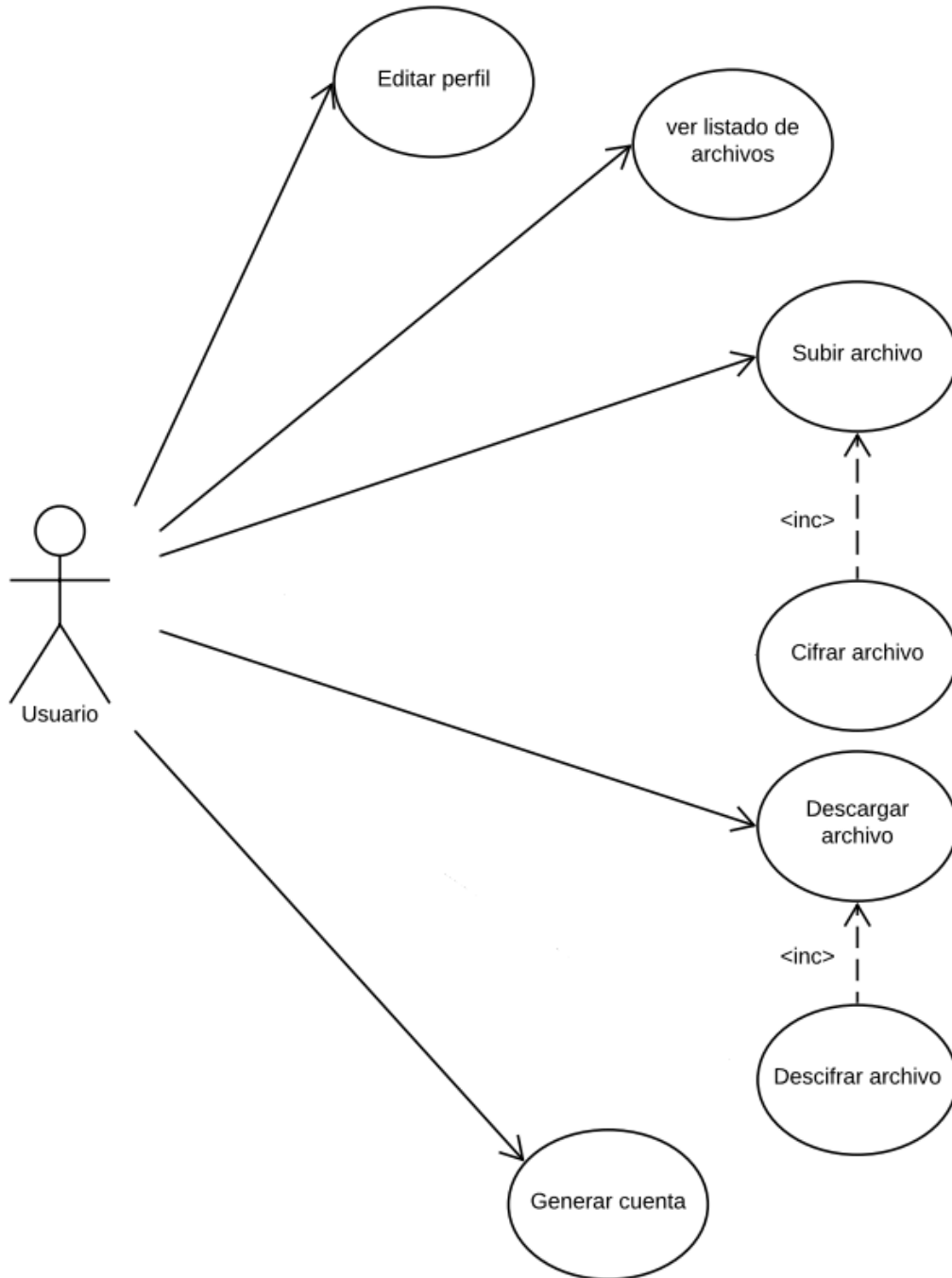
Requerimientos no funcionales

- Almacenar archivos en red IPFS.
- Guardar índice de archivos en Smart Contract, alojado en la Blockchain de Ethereum (ETH).

- Migrar Smart Contract a la Test Net de Ethereum (Blockchain).
- Migrar Smart Contract a la Main Net de Ethereum (Blockchain).
- Cifrar archivo con AES (Advanced Encryption Standard - Estándar de cifrado avanzado).
- Desarrollar front-end con HTML5, CSS3, JS.
- Registrar al usuario con la billetera virtual: MetaMask.

Casos de Uso

Usuario



Use Case: Editar perfil

Nombre del Use Case: Editar perfil	Nro. de Orden: 1
Actor Principal: Usuario	Actor Secundario: no aplica

Tipo de Use Case: Concreto
Objetivo: editar el perfil del usuario
Descripción: el usuario logueado un la dapp podrá editar su perfil (general, conexiones, avanzado, contactos, seguridad y privacidad, redes y acerca) desde su billetera virtual de MetaMask, en el menú de configuración incluido dentro de la misma.

Autor: Nicolás Grupe	FUM: 24/04/2019
-----------------------------	------------------------

Use Case: Ver listado de archivos

Nombre del Use Case: Ver listado de archivos	Nro. de Orden: 1
Actor Principal: Usuario	Actor Secundario: no aplica

Tipo de Use Case: Concreto
Objetivo: ver el listado total de archivos subidos por el usuario
Descripción: el usuario podrá ver una tabla en la página principal de la dapp con el total de archivos subidos por el mismo desde que empezó a interactuar con la esta por primera vez

en su vida hasta la hora actual, los mismos se mostrarán infinitamente hacia abajo, resultando ser todos los que el usuario ha subido hasta el momento. En esta tabla, los archivos estarán referenciados por las siguientes características: Un número o id (#), el título del archivo subido (File Title), la fecha y hora en que se realizó la transacción (Transaction datetime) y un botón de descarga (Download from IPFS) que permitirá la descarga directa del archivo referenciado. El usuario podrá ordenar los mismos por “número o id, #” de menor a mayor y viceversa o también ordenarlos por orden alfabético con el “File Title”.

Autor: Marcos Lozada	FUM: 24/04/2019
-----------------------------	------------------------

Use Case: Subir archivo

Nombre del Use Case: Subir archivo	Nro. de Orden: 1
Actor Principal: Usuario	Actor Secundario: no aplica

Tipo de Use Case: Concreto
Objetivo: subir un archivo a la nube por el usuario
Descripción: el usuario solicitara subir un archivo a la nube, seleccionando el que desee desde su ordenador y el mismo se subirá incluyendo el cifrado del archivo (se ejecuta el UC - Cifrar archivo)

Autor: Marcos Lozada	FUM: 24/04/2019
-----------------------------	------------------------

Use Case: Cifrar archivo

Nombre del Use Case: Cifrar archivo	Nro. de Orden: 1
Actor Principal: Usuario	Actor Secundario: no aplica

Tipo de Use Case: Abstracto
Objetivo: cifrado de un archivo que el usuario desea subir a la nube
Descripción: una vez seleccionado el archivo que se desea subir por el usuario a la nube, le dapp cifrara correctamente el mismo a fin de protegerlo de cualquier amenaza que busque corromper su integridad. El archivo luego estará cifrado y listo para ser subido a la red descentralizada.

Autor: Marcos Lozada	FUM: 24/04/2019
-----------------------------	------------------------

Use Case: Descargar archivo

Nombre del Use Case: Descargar archivo	Nro. de Orden: 1
Actor Principal: Usuario	Actor Secundario: no aplica

Tipo de Use Case: Concreto
Objetivo: descargar un archivo desde la nube por el usuario

Descripción: el usuario solicitara descargar un archivo desde la nube, seleccionando el que desee desde su ordenador al ver en la tabla de archivos subidos previamente, dando click en el botón descargar (Download) y el mismo será descargado incluyendo el descifrado del archivo (se ejecuta el UC - Descifrar archivo)

Autor: Nicolás Grupe

FUM: 25/04/2019

Use Case: Descifrar archivo

Nombre del Use Case: Descifrar archivo	Nro. de Orden: 1
Actor Principal: Usuario	Actor Secundario: no aplica

Tipo de Use Case: Abstracto

Objetivo: descifrado de un archivo que el usuario desea descargar a la nube

Descripción: una vez seleccionado el archivo que se desea descargar por el usuario desde la nube, al hacer click en el botón descargar (Download), le dapp descifrara correctamente el mismo a fin de que el archivo vuelva a estar en el estado original antes de que se haya cifrado previamente. El archivo luego estará descifrado y listo para ser consultado nuevamente por el usuario a fines de su incumbencia.

Autor: Nicolás Grupe

FUM: 25/04/2019

Use Case: Generar cuenta

Nombre del Use Case: Generar cuenta	Nro. de Orden: 1
Actor Principal: Usuario	Actor Secundario: no aplica

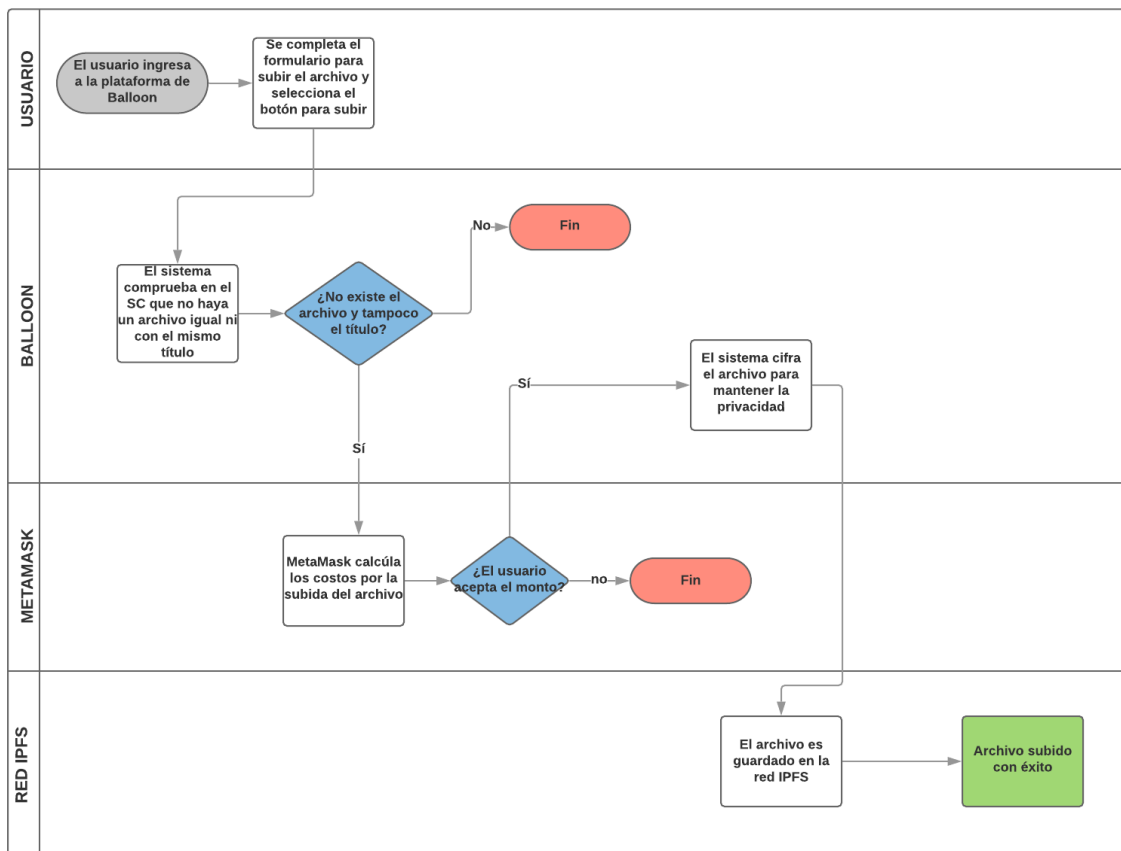
Tipo de Use Case: Concreto
Objetivo: generar una cuenta por el usuario en la billetera virtual MetaMask
Descripción: el usuario, al entrar por primera vez a la dapp Balloon, se le solicitará que cree una cuenta en la billetera digital MetaMask. El mismo, siguiendo unos simples pasos, generará la cuenta con la que va a poder interactuar luego en Balloon, siendo este su usuario.

Autor: Marcos Lozada	FUM: 24/04/2019
-----------------------------	------------------------

Diagrama de flujo

Subir un archivo

A la hora de subir un archivo, el usuario, luego de completar el formulario de subida del archivo (título y cargar el archivo), debe seleccionar el botón “Upload” (subir en inglés). Al subir el archivo, el sistema de Balloon comprueba en el Smart Contract, precisamente en el patrón que se decidió usar llamado Eternal Storage, y , en caso de que no sea el mismo archivo y tenga el mismo título, pasa al sistema de MetaMask, donde se realizará la transacción y consultará al usuario antes de subir el archivo para ver si está de acuerdo con el valor. Si es aprobado, el sistema de Balloon cifra el archivo, con el método AES, y luego el archivo es subido a la red IPFS.

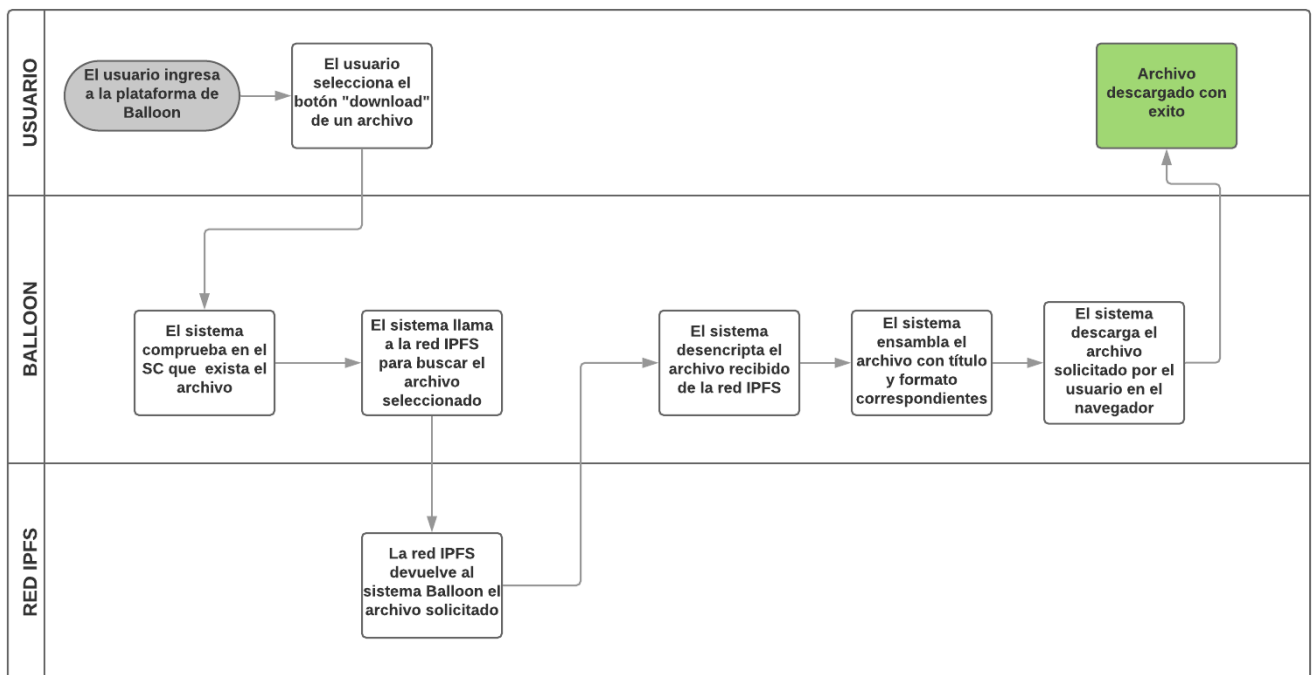


Descargar un archivo

En el momento que el usuario solicite descargar un archivo, previamente cargado al sistema de Balloon, tras el botón que se encuentra al final de cada archivo en la tabla de archivos subidos, el sistema de Balloon va a comprobar que ese archivo exista, luego solicitará, por medio de tener el hash apropiado del archivo seleccionado, a la red IPFS para conseguir el mismo. La red IPFS encuentra el archivo y lo devuelve al sistema de Balloon.

El archivo está originalmente cifrado, con el método de cifrado AES, por lo que el sistema de Balloon va a descifrarlo por medio de la utilización de la clave generada al momento en que el archivo fue cifrado inicialmente cuando el usuario lo subió por primera vez.

Una vez descifrado el archivo, Balloon ensamblará el archivo para que tenga la misma apariencia, en cuanto al título y el formato del archivo, del archivo original subido por el usuario. Luego de ese proceso, el sistema descarga, finalmente, el archivo en el dispositivo del usuario, reflejando en el navegador del dispositivo.



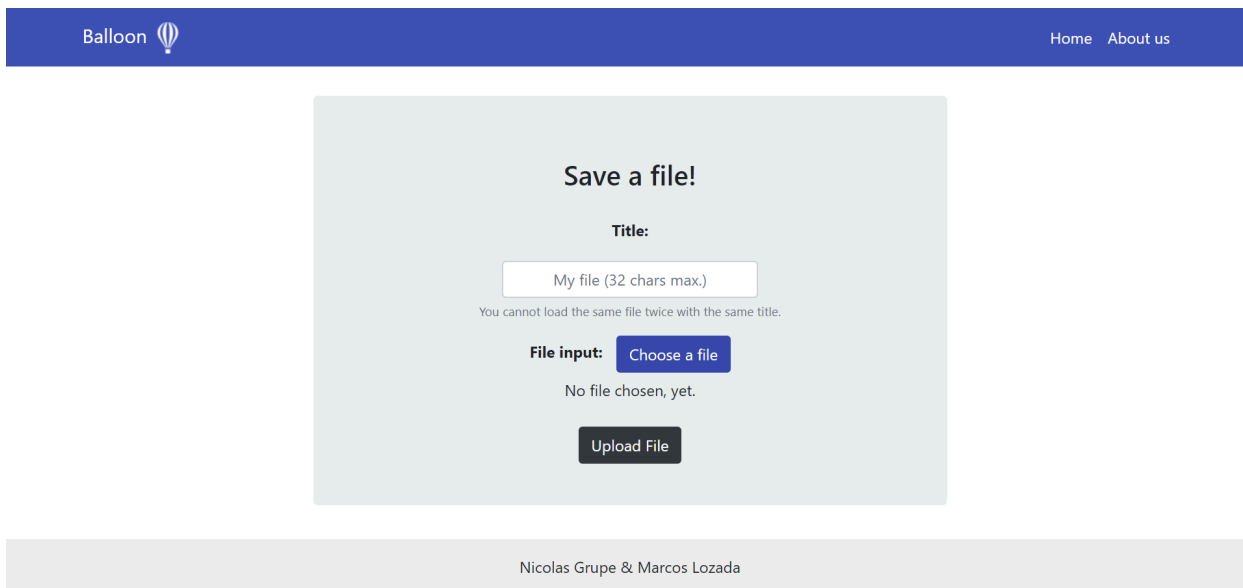
La aplicación

Luego de todo el proceso que implicó llevar a cabo este proyecto y sus largas horas de trabajo, se logró una versión final del mismo. La aplicación final corre localmente, pero se está trabajando para que sea alojada en Heroku o algún hosting adecuado a la brevedad y sea de público acceso.

En esta sección del documento se procederá a explicar las funcionalidades más importantes que posee la plataforma, de forma gráfica mostrando las pantallas principales de la misma, haciendo un recorrido por cada una y a su vez indicando el objetivo de cada una de las pantallas.

Pantallas de la aplicación del usuario

Home

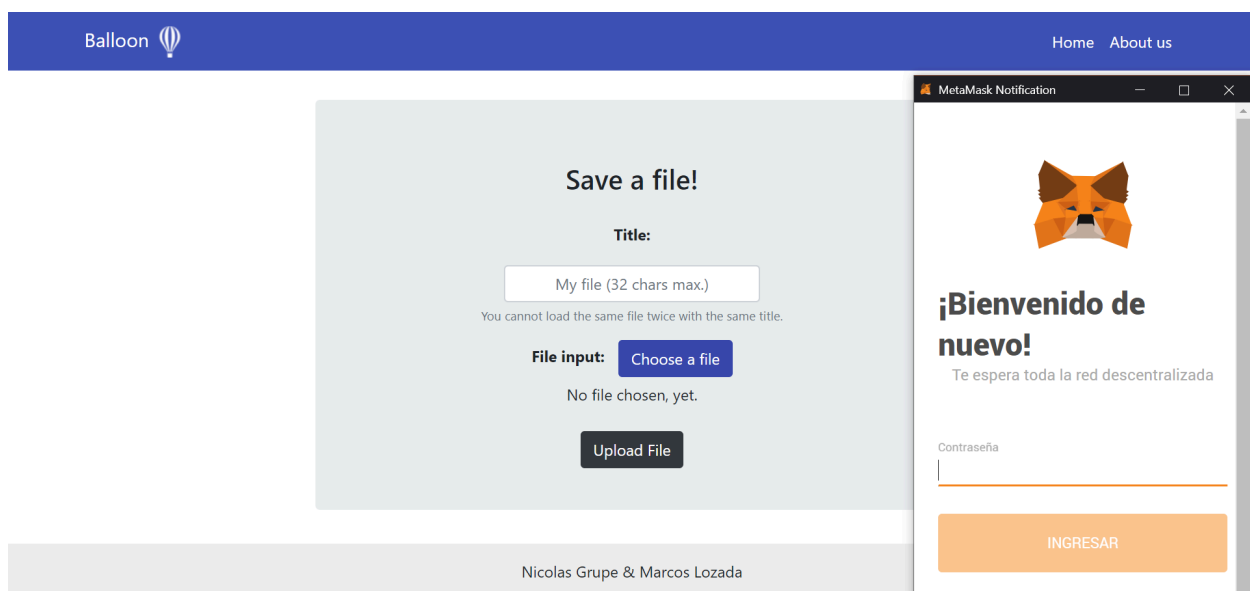


The screenshot shows the home page of the Balloon application. At the top, there is a dark blue header with the Balloon logo and a hot air balloon icon on the left, and navigation links for 'Home' and 'About us' on the right. The main content area is a light gray box titled 'Save a file!'. It contains a form with a 'Title:' label, a text input field with the placeholder 'My file (32 chars max.)', and a feedback message: 'You cannot load the same file twice with the same title.' Below the input field is a 'File input:' label, a blue button labeled 'Choose a file', and a feedback message: 'No file chosen, yet.' At the bottom of the form is a dark gray button labeled 'Upload File'. The footer of the page is a light gray bar with the text 'Nicolas Grupe & Marcos Lozada'.

Esta es la pantalla inicial de la Dapp Balloon. El usuario al ingresar al URL de Balloon, estando logueado en la billetera digital MetaMask, verá la siguiente interfaz en el home de

inicio. En ella se puede encontrar un formulario de carga que invita al usuario a que almacene un archivo de manera descentralizada, solicitando un Título (Title) de 32 caracteres máximo, también que el usuario elija un archivo del ordenador dando click en el botón elegir un archivo (Choose a File) y luego termina el formulario con el botón de subida de archivo (Upload File) al que el usuario le va a dar click una vez que haya seleccionado el archivo y título adecuado. Por último en esta pantalla, en la esquina superior derecha se verán dos botones, Home (que permite recargar o volver al home estando en otra pantalla) y About Us (que es el botón que permite ir a la pantalla que explica sobre qué es Balloon y algunos conceptos básicos para poder utilizar este almacenamiento en la nube).

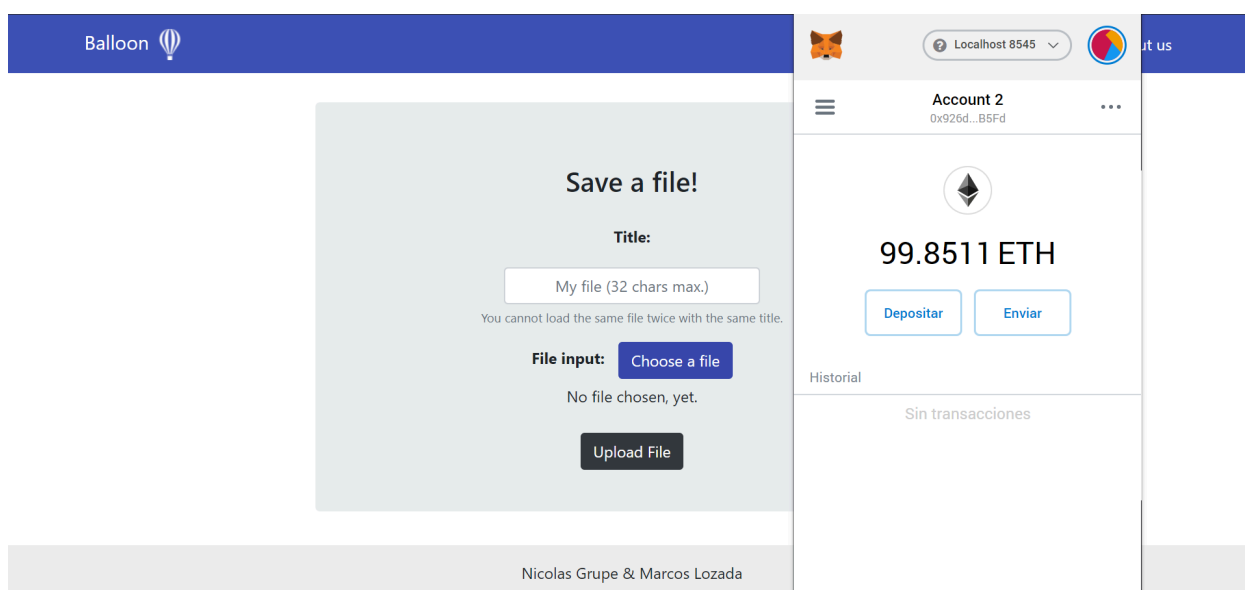
Home Pop Up Login MetaMask



Esta pantalla que se observa aquí arriba, es la que verá el usuario cuando entre por primera vez a la Dapp Balloon o como también cuando el mismo esté ingresando desde otro ordenador o haya cerrado sesión previamente en la billetera digital MetaMask y desee ingresar de nuevo al sistema. Se aparecerá un Pop Up de la billetera digital MetaMask solicitando la contraseña, en el caso de que sea un usuario que haya estado interactuando previamente desde el mismo navegador y en la misma computadora. Otro caso podría ser un usuario que ingresa por

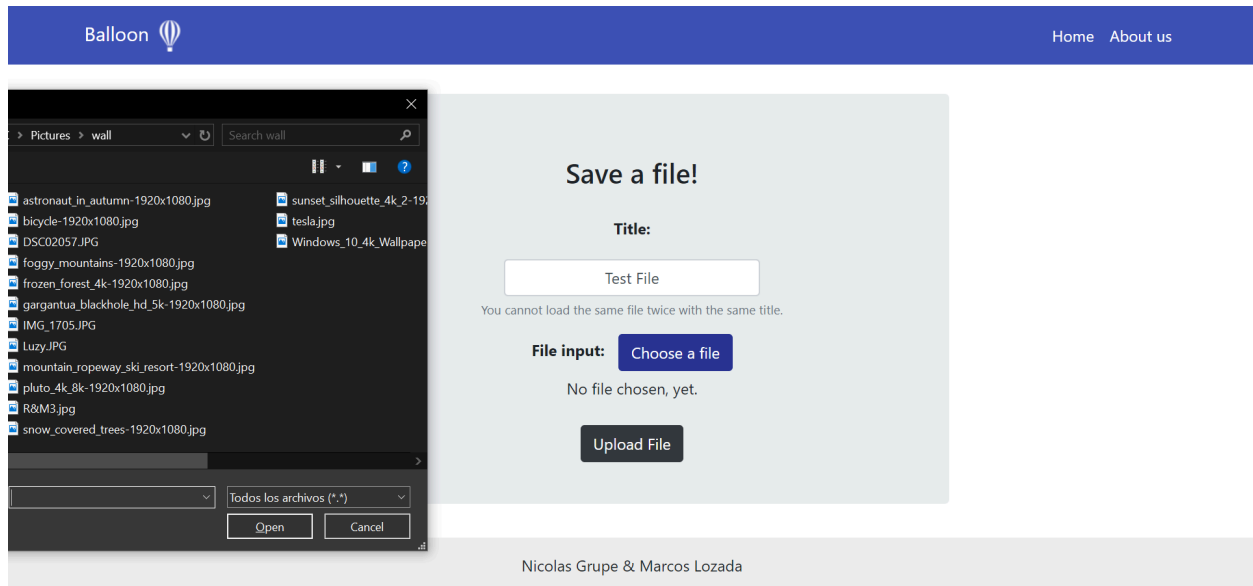
primera vez al sistema, por lo que él mismo desde este Pop Up va a poder seguir los pasos para crear un usuario en MetaMask, que va a ser su usuario de Balloon a su vez, con el que va a interactuar y subir los archivos de su interés. Y por último se encuentra el caso de que un usuario existente desee loguearse desde otro ordenador o navegador, el mismo dentro del Pop Up deberá poner su frase semilla (seed phrase) y seguir los pasos para la correcta importación de su cuenta, pudiendo, en fin, utilizar la Dapp.

Home Pop Up MetaMask



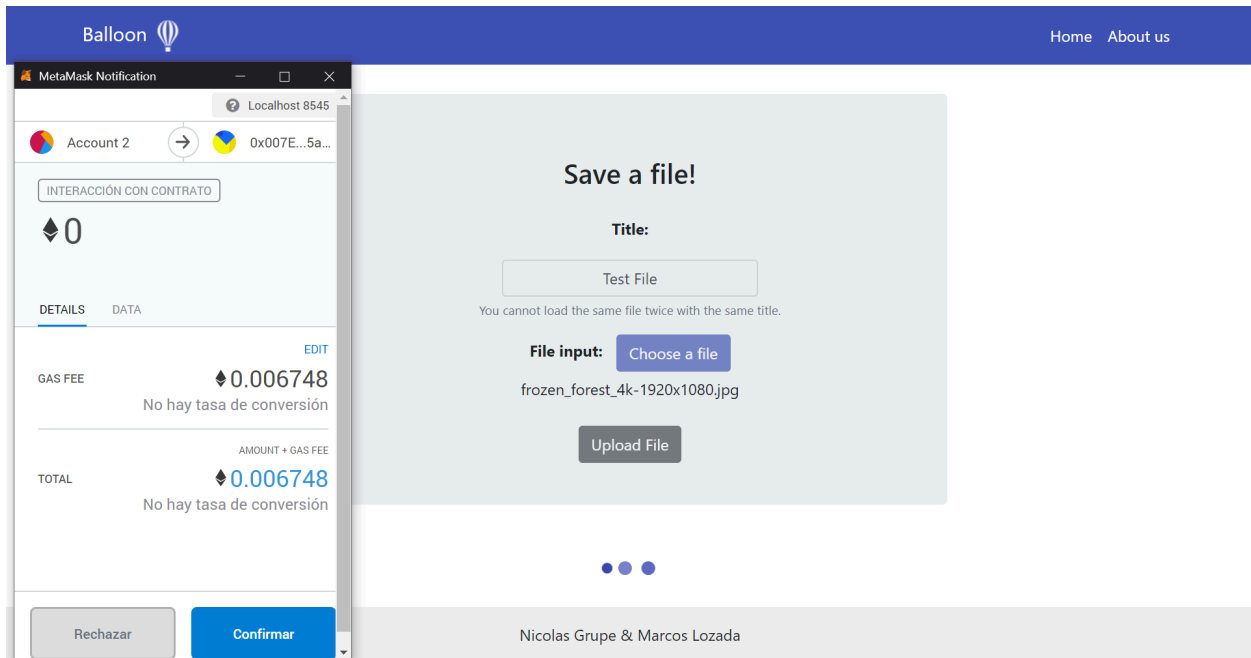
En esta pantalla, un usuario con la billetera digital MetaMask activa y estando logueado con la misma en el navegador, hace click en el icono de MetaMask que es una pequeña cabeza de un zorro, al estar en el URL de Balloon, podrá solicitar en qué estado está su billetera. En la misma se ven detalles del usuario que pueden ser modificados, el balance de la billetera y saldos restantes, a su vez tiene accesos a todas las funciones de MetaMask en sí. Se irán actualizando las operaciones a medida que el usuario va interactuando con la Dapp al subir archivos y el mismo va a poder ir viendo las actualizaciones y transacciones en esta pestaña, lo cual es de gran comodidad.

Home con selector de archivos



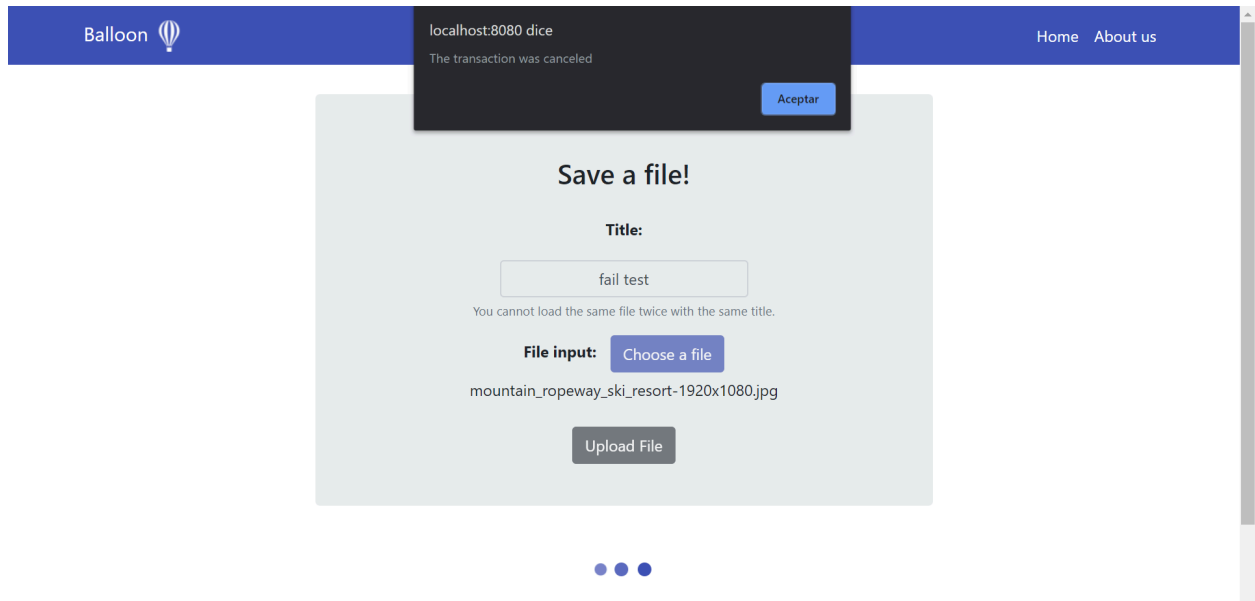
En esta interfaz, se observa el clásico Pop Up del explorador de archivos sobre la pantalla Home original, este se activa cuando el usuario le da click al botón para elegir un archivo del ordenador (Choose a file). En la misma el usuario va a interactuar directamente con el Pop Up en el que va a seleccionar el archivo de interés que desee subir a la Dapp Balloon para que sea encriptado y alojado en la red descentralizada, este archivo podrá ser de cualquier tipo, y con eso se hace referencia a: .zip, .png, .doc, .pdf, .jpg, .exe, .tar y muchos otros más!

Home Pop UP MetaMask Confirmar Transacción

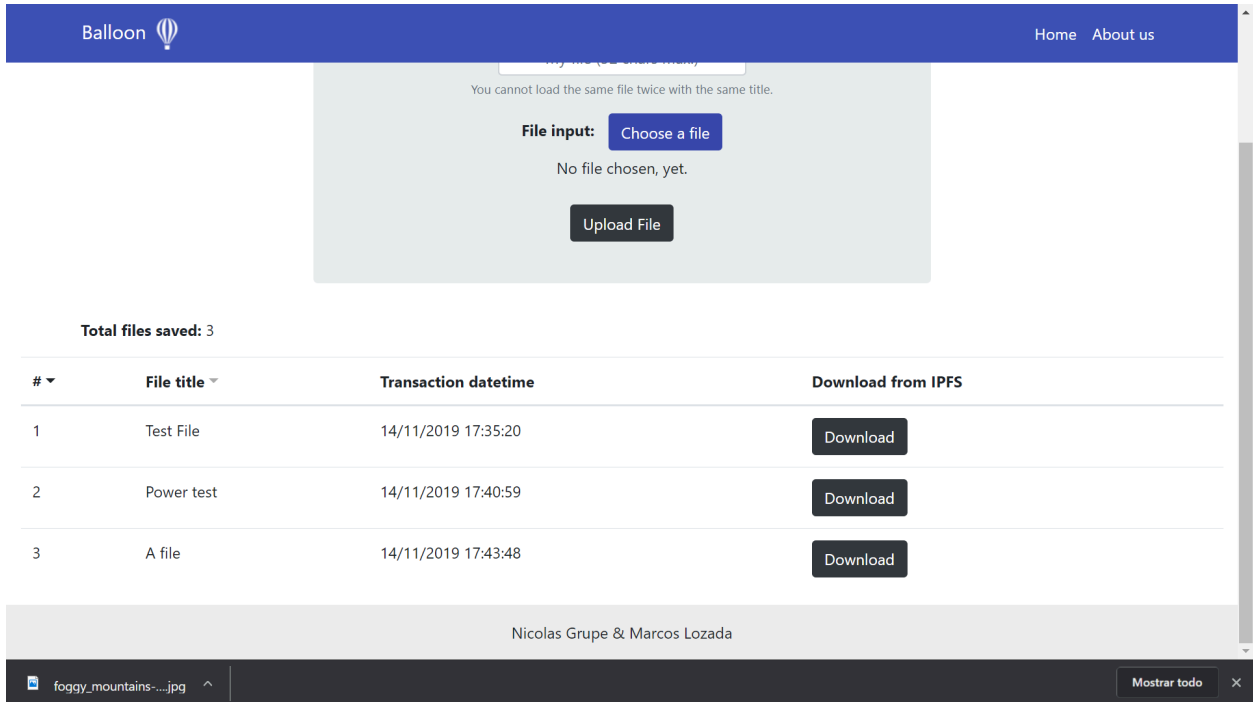


En esta vista, se observa la pantalla Home de fondo, pero la atención va en el Pop Up de la billetera digital MetaMask. El mismo invita al usuario que previamente ha puesto un título en el campo "Title" del formulario, ha elegido un archivo desde el ordenador interactuando con el explorador de archivos al hacer click en el botón de selección de archivos (Choose a file) y por ultimo ha hecho click en el botón para subir el archivo (Upload File) a que el mismo confirme la operación y por ende la transacción que se va a realizar interactuando con el Smart Contract (Contrato Inteligente) de la Blockchain de Ethereum (ETH) en su billetera digital de MetaMask, dado que la misma tiene un costo por el trabajo de procesamiento de los nodos a fin de realizar la operación de almacenamiento del archivo que el usuario ha solicitado en la Dapp.

Home Pop UP MetaMask Rechazar Transacción



En esta vista, en diferencia de la anterior, se observa la pantalla Home de fondo y un mensaje que nos avisa que la transacción ha sido cancelada (The transaction was canceled). Esto se debe a que el usuario, previamente ha cancelado la operación presionando en el botón “Rechazar” de la imagen previa (Home Pop UP MetaMask Confirmar Transacción) dando lugar a la correcta cancelación de la operación y por ende la transacción no se va a realizar, dejando de lado la interacción con el Smart Contract (Contrato Inteligente) de la Blockchain de Ethereum (ETH) a través de la billetera digital de MetaMask del usuario, dado que la misma tiene un costo por el trabajo de procesamiento de los nodos a fin de realizar la operación de almacenamiento del archivo que el usuario ha solicitado en la Dapp, que no ha sido aprobado por el mismo y no desea llevar el almacenamiento del mismo a cabo.

Home con archivos descargados

The screenshot shows the Balloon application interface. At the top, there is a blue header with the Balloon logo and navigation links for Home and About us. Below the header, a central area contains a file upload form with a message: "You cannot load the same file twice with the same title." The form includes a "File input:" label, a "Choose a file" button, the text "No file chosen, yet.", and an "Upload File" button.

Below the upload form, it states "Total files saved: 3". A table displays the list of saved files:

#	File title	Transaction datetime	Download from IPFS
1	Test File	14/11/2019 17:35:20	Download
2	Power test	14/11/2019 17:40:59	Download
3	A file	14/11/2019 17:43:48	Download

At the bottom of the page, there is a footer with the text "Nicolas Grupe & Marcos Lozada". A taskbar at the very bottom shows a file named "foggy_mountains-...jpg" and a "Mostrar todo" button.

La pantalla que se observa aquí arriba, presenta la pantalla Home, que se ha ido actualizando a medida que el usuario ha ido interactuando con Balloon al subir sus archivos de interés. El usuario verá una tabla y a su vez un indicador superior que muestra la cantidad total de archivos que ha ido almacenando el mismo hasta el momento. Dentro de los campos de la tabla se tienen: el ID o número del archivo (#), el título del archivo con el que el usuario decidió almacenarlo (File title), la fecha exacta en que se realizó la transacción (Transaction datetime) y por último un campo de descarga (Download from IPFS) que posee un botón específico para cada archivo de la tabla que al darle click por el usuario el mismo descargara el archivo directamente al ordenador como se ve en la esquina izquierda debajo de la página en la que se ha descargado un archivo por parte del usuario.

About Us



About Balloon

Here you can store your files using Blockchain technology (Ethereum) so that no one can access it but you. Saving a file with Balloon is as easy as completing the file form in the Home section and you will be able to access it for life and from any device.

When you upload a file with Balloon, first, we will cipher it, and after that your file will be saved in the IPFS network (and that is why it is going to be accessible from any place in the world), ensuring that your file is stored in a decentralized way, without third parties, for a life time. You will be able to download your files any time from the chart you will see below the 'Save a file!' form.

If you want to contact us just click [here](#) to send us an e-mail.

What do you need to upload a file?

All you need in order to get started is to have a wallet in [metamask.io](#) and after that you have to [buy Ether](#) so that you can pay for the transaction of uploading your files into the Ethereum Blockchain.

Balloon team



Nicolás Grupe

Systems Engineering student at UCC

[E-Mail Nicolás](#)



Marcos Lozada

Systems Engineering student at UCC

[E-Mail Marcos](#)

Nicolas Grupe & Marcos Lozada

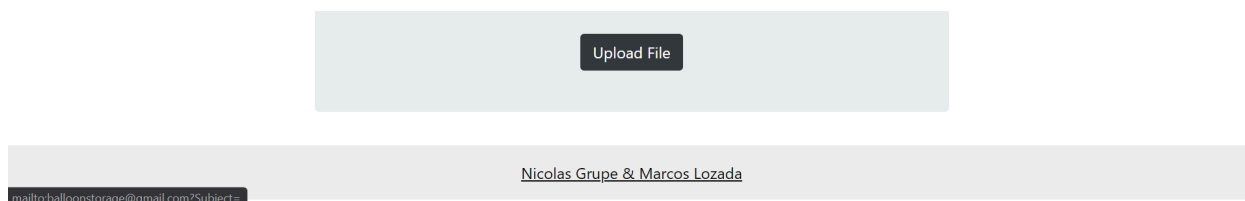
En la pantalla About Us (Sobre nosotros), que es la que se observa aquí arriba, es a la que el usuario accede al hacer click en el botón de la esquina superior derecha que dice "About Us". En esta interfaz, se encuentran tres breves secciones bien detalladas.

La primera, que lleva el título "About Balloon" (Sobre Balloon) consta de una breve descripción acerca de para qué sirve la Dapp Balloon y a su vez también detalla sus características y beneficios que viene a brindar al usuario.

La segunda, que se titula “What do you need to upload a file?” en español: ¿Qué necesitas para subir un archivo?, se brinda fácil acceso, a través de hipervínculos, a las tecnologías que se utilizan para poder interactuar con la Dapp Balloon que son: MetaMask (la billetera digital) y Ether (un crypto activo) de Ethereum (ETH).

La tercera, que se encabeza con la palabra “Balloon team” o “Equipo Balloon” en español, muestra fotos de los fundadores de Balloon, Nicolás Grupe y Marcos Lozada, los roles de ambos y un enlace que permite enviarles un mail directamente a cada uno desde la casilla de correo electrónico del usuario solicitante al hacer click en el mismo, a fin de responder cualquier inquietud del mismo.

Enviar correo electrónico



(Componente footer de Balloon)

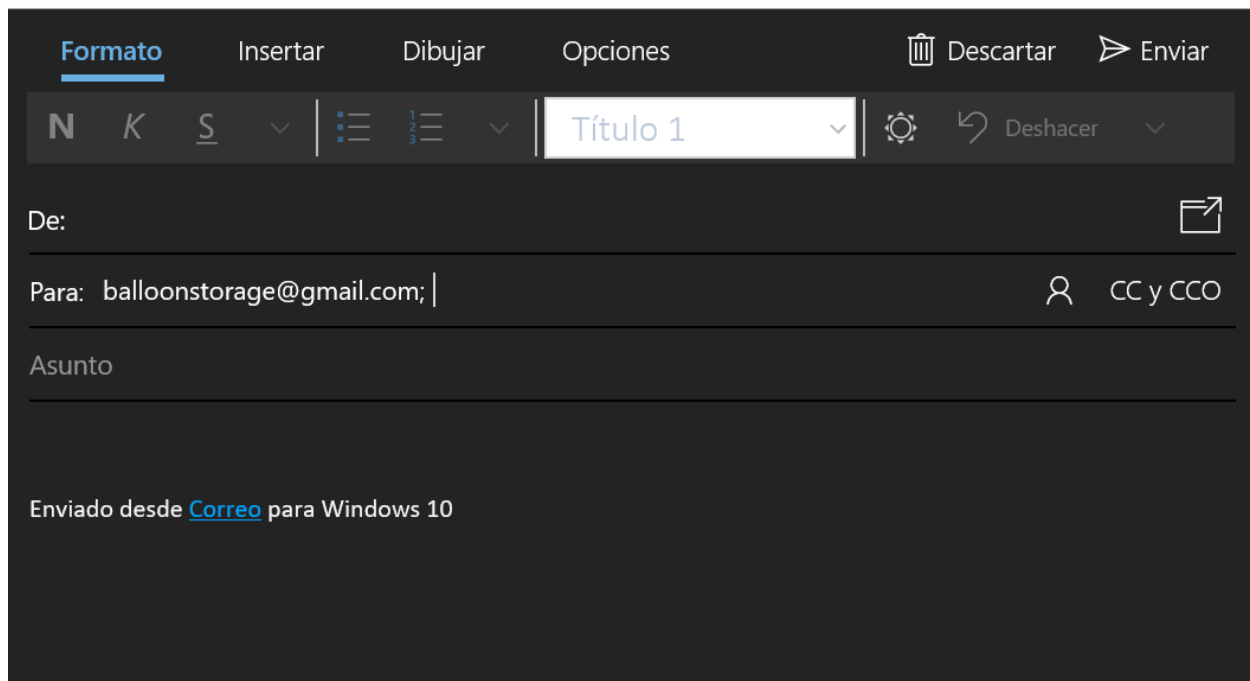
About Balloon

Here you can store your files using Blockchain technology (Ethereum) so that no one can access it but you. Saving a file with Balloon is as easy as completing the file form in the Home section and you will be able to access it for life and from any device.

When you upload a file with Balloon, first, we will cipher it, and after that your file will be saved in the IPFS network (and that is why it is going to be accessible from any place in the world), ensuring that your file is stored in a decentralized way, without third parties, for a life time. You will be able to download your files any time from the chart you will see below the 'Save a file!' form.

If you want to contact us just click [here](#) to send us an e-mail.

(Componente About Balloon en la sección About Us)



En el componente Footer (parte inferior de la plataforma web), donde se podrán encontrar los nombres de los fundadores de Balloon, Nicolás Grupe & Marcos Lozada, es en realidad un link (enlace en español). Lo mismo en la sección About Us, bajo el título de About Balloon, en el último párrafo se encontrará un enlace en color azul que dice “here”. Estos enlaces lo que hacen es indicar al dispositivo que se esté utilizando que debe abrir la aplicación de correos electrónicos del mismo y en la sección de “Para” del correo, ingresar la dirección que se dejara asociada al enlace y será detallada a continuación. Esta dirección fue creada con el propósito de brindar soporte y consultoría a los usuarios de la plataforma. La dirección es “balloonstorage@gmail.com”.

De esta forma el navegador abrirá la aplicación de correo del dispositivo de forma automática ya teniendo listo la dirección de correo para que el usuario se ponga en contacto con nosotros de forma rápida.

Testing

En esta sección se detalla la etapa de pruebas/testing que se llevó a cabo en el desarrollo de la Dapp Balloon, la misma tomó lugar en dos instancias, una con pruebas unitarias de código, programadas con JavaScript para que se corran de manera automática, y la otra constó de una serie de diez casos de prueba donde se invitó a distintas personas para que pusieran a prueba nuestra Dapp, logrando de esa forma mejorar la calidad del producto.

La primera instancia constó principalmente en realizar una serie de pruebas unitarias de código programadas en JavaScript, las mismas testean las funciones que incluye el Smart Contract de la Dapp, en el cual las mismas no pueden fallar ya que las transacciones que opera el mismo son de gran importancia, para lo cual se han automatizado estas pruebas a fin de que corran automáticamente y eviten cualquier error o problema.

La segunda instancia incluyó la realización de diez casos de prueba escritos en planillas especialmente diseñadas para detallar los mismos, las cuales se anexan en este documento al final, dentro de la sección “Anexos” con el título “Test Cases - Balloon”. En esta ocasión se realizaron dos sprints (cada 15 días) específicamente diseñados para testear la Dapp Balloon en el mes de Septiembre. El primer sprint comenzó el 01/09/2019 y terminó el 15/09/2019, mientras que el segundo sprint comenzó el 16/09/2019 y finalizó el 30/09/2019. Luego de cada sprint se entregaron cinco Test Cases y se invitó a personas random para que pusieran a prueba nuestro desarrollo, logrando de esa forma mejorar el feedback y así poder llegar más lejos con la implementación de la Dapp para almacenamiento descentralizado: Balloon.

Beneficios post-implantación

La Dapp Balloon fue desarrollada para brindar una opción de almacenamiento más segura y confiable, sin terceras partes que puedan acceder a la información del usuario, en forma descentralizada. Las funcionalidades que Balloon ofrece, forma de operar y accesibilidad de la aplicación hacen posible que estos problemas se solucionen de la forma más eficiente y práctica posible.

Para el caso de las personas, tanto físicas como jurídicas, Balloon les brinda una herramienta que tiene, dentro de sus beneficios, una puerta de acceso al mundo de la tecnología de Blockchain, tecnología de vanguardia, que trae, entre otros beneficios, que los archivos permanezcan almacenados para toda la vida en la Blockchain de Ethereum, con seguridad y anonimidad, brindando, de esta manera, un espacio de almacenamiento en una nube en la que cualquier usuario podría confiar en que solo él podrá tener acceso a los archivos que vaya a subir desde su cuenta de MetaMask.

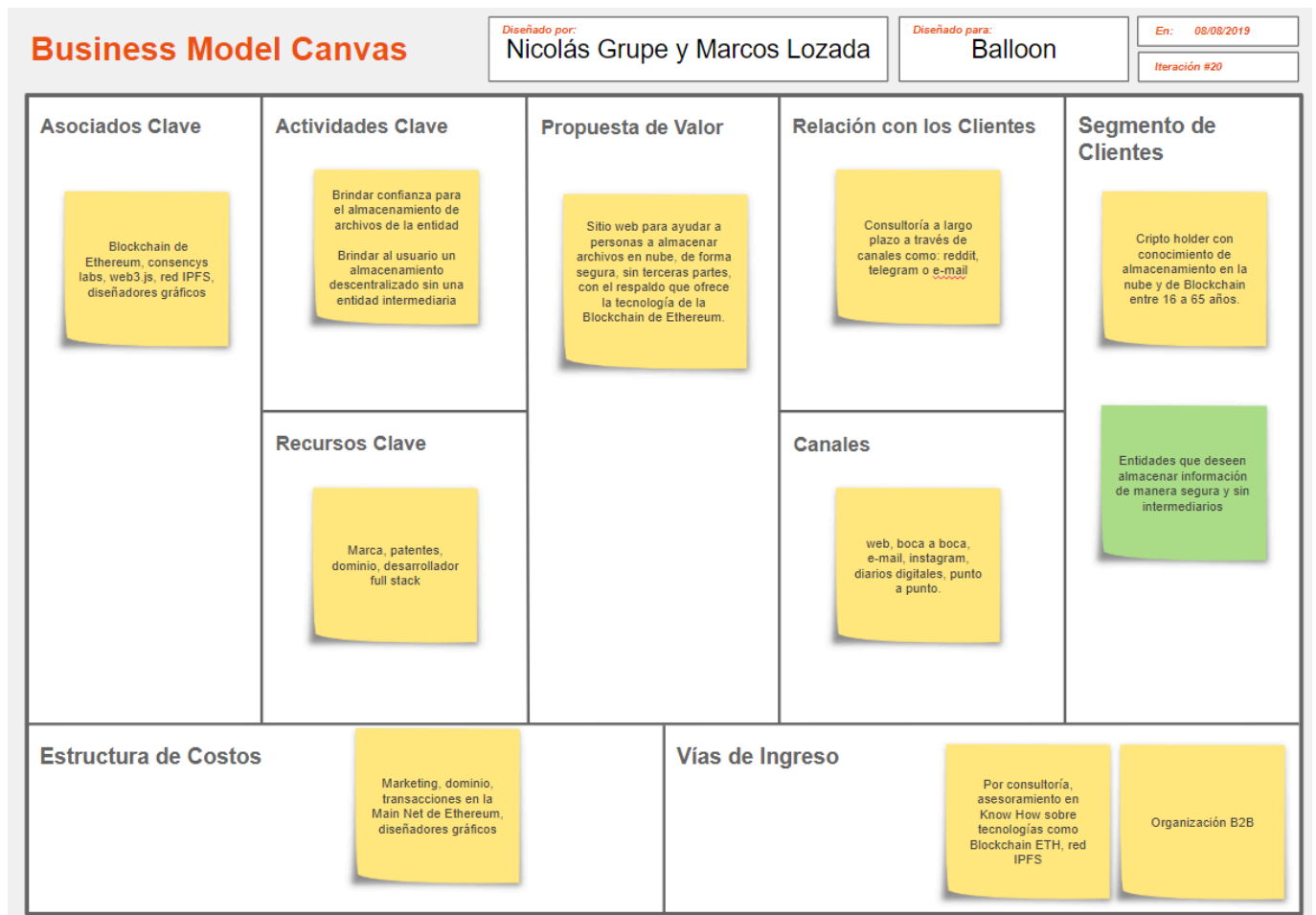
Si bien la subida de los archivos tiene un costo, por el trabajo de procesamiento de los nodos a fin de realizar la operación de almacenamiento del archivo que el usuario ha solicitado, es un pequeño costo a pagar por poder mantener sus archivos seguros de cualquier gran empresa, y de otros usuarios, porque no.

Dejando de lado el hecho del almacenamiento descentralizado en nube en sí, Balloon está ofreciendo un acercamiento, de los pocos que se ven actualmente en el mundo, de permitir que los usuarios interactúen con la tecnología de Blockchain de una forma tan amigable (user friendly), algo que hasta el momento solo se daba para entendidos hackers (persona experta en el manejo de computadoras) utilizando la Blockchain desde terminales a través de líneas de comando.

Modelo de Negocio

Para poder representar el modelo de negocios de la Dapp Balloon, se utilizó una herramienta online llamada BMC (Business Model Canvas - en español: Esquema de modelo de negocios Canvas). Se decidió utilizar dos colores distintos para los Post-it representando así los distintos segmentos de clientes a los que Balloon está orientado.

- Por un lado el amarillo para los cripto holders con conocimiento en almacenamiento en la nube y de Blockchain, entre 16 a 65 años.
- Por el otro, el color verde para las entidades que deseen almacenar información de manera segura y sin intermediarios.



Descripción de los bloques

A continuación se detalla de manera breve a que está orientado cada bloque atacado en este esquema de modelado de negocios estilo Canvas, en torno a nuestra Dapp Balloon, demostrando cada segmento de la mejor manera.

Propuesta de valor

Nuestra propuesta de valor con la Dapp Balloon es brindar un sitio web para ayudar a personas a almacenar archivos en nube, de forma segura, sin terceras partes, con el respaldo que ofrece la tecnología de la Blockchain de Ethereum.

La tecnología de Blockchain cumple exactamente los requerimientos para poder utilizarse como un almacenamiento en nube con la ventaja de hacerlo en forma descentralizada, y en el caso de Balloon, proporcionando como valor agregado, que el archivo se encuentre cifrado, brindando de esta forma mayor seguridad. También, la velocidad de acceso a los archivos a la hora de descargarlos pasa a ser una gran ventaja de almacenarlo en la Blockchain ya que nuestra información se va a encontrar replicada en varios nodos y su acceso va a ser muy rápido.

Segmento de clientes

Balloon busca atacar dos segmentos de clientes puntuales y se cree son los más realistas en torno a esta Dapp en sí:

- Por un lado se tiene a los cripto holders con conocimientos en el almacenamiento en la nube y de Blockchain, entre 16 a 65 años. Estos son aficionados como nosotros en la tecnología de Blockchain y las ventajas que trae respecto a el internet tradicional.
- Por el otro, las entidades que deseen almacenar información de manera segura en la nube y sin intermediarios o terceras partes. Este segmento de clientes apunta tanto a las personas físicas que desean almacenar información sensible de manera segura sin terceras partes, como entidades, o personas jurídicas, que desean almacenar información confidencial de la entidad misma en alguna red, o servicio en nube, que garantice la confidencialidad y seguridad de la información.

Canales

La forma en que la Dapp Balloon va a llegar a los usuarios es bastante clara. Principalmente mediante la web con todos sus medios que se pueden encontrar hoy en día, luego se tiene el boca a boca de nuestros usuarios, vía e-mail difusivos patrocinando Balloon, la red social instagram no se dejara de lado tampoco, varios diarios digitales que nos han preguntado ya y van a ser partners así como también la revista punto a punto.

Relación con los clientes

La relación con nuestros clientes y usuarios se dará pura y exclusivamente por consultoría y asesoramiento a largo plazo a través de canales como: reddit, telegram o email, este último estará anexado en la página para que los usuarios puedan conectarse directamente con nosotros a fin de solventar cualquier inconveniente o mejorar Balloon.

Actividades clave

A fin de lograr la propuesta de valor de la Dapp Balloon, se van a tener que llevar dos actividades clave a cabo:

- Por un lado, brindar absoluta confianza para el almacenamiento de archivos de la entidad, evitando cualquier duda y potenciando la seguridad de nuestros usuarios y de su información.
- A su vez y fundamentalmente, brindar al usuario un almacenamiento descentralizado sin una entidad intermediaria. Permitiendo enlaces directos entre el usuario y su información, sin alteraciones o anomalías entre medio.

Recursos clave

Se necesitarán los siguientes recursos para alcanzar la propuesta de valor de manera adecuada y profesionalmente:

- Una marca, y las respectivas patentes que demuestran nuestra legalidad y ganas de desarrollar algo realmente seguro y profesional.
- Un dominio, exclusivo a nuestra firma Balloon.
- Desarrollador full stack a fin de desarrollar la Dapp y mantenerla en el tiempo de cualquier inconveniente o mejora.

Asociados clave

Nuestros asociados clave que se mostraran como alianzas estratégicas serán aquellos que nos den la confianza total para trabajar juntos, dentro de estas entidades se encuentran:

- La Blockchain de Ethereum (ETH) y herramientas asociadas a esta como Etherscan, Ether, Contratos inteligentes (Smart Contracts).
- Consencys Labs con varias de sus herramientas como MetaMask y Truffle Suite.
- La librería web3.js.
- La red IPFS (InterPlanetary File System)
- Diseñador gráfico apto para brindar profesionalismo absoluto a fin de diseñar de la mejor manera la Dapp Balloon.

Estructura de costos

Al plantear la estructura de costos, se detectó que los costos mayores que existirán a lo largo del proyecto son por el marketing que se va a tener que llevar a cabo y la contratación de un Lic. en Marketing apto para esto, el dominio de Balloon, las transacciones en la Main Net de Ethereum (ETH) que serán un gasto para los founders inicialmente ya que serán realizadas por nosotros para migrar el contrato inteligente (Smart Contract) a Blockchain, el diseñador gráfico apto para diseñar de la mejor manera la Dapp Balloon.

Vías de ingreso

La principal vía de ingreso de nosotros va a ser por consultoría y asesoramiento en Know How sobre tecnologías como Blockchain ETH o red IPFS (InterPlanetary File System). Se cree que las entidades que estén interesadas en Balloon van a querer comunicarse con nosotros a fin de desarrollar soluciones para sus empresas o requerimientos personales que se nutran de estas tecnologías de cabecera como Blockchain de Ethereum (ETH), red IPFS, billeteras digitales como MetaMask, Ether y demás.

Impacto Social

Mediante el desarrollo de este proyecto, se busca ofrecer a las personas privacidad de su información. Actualmente vivimos en un mundo globalizado, donde si una persona sube cualquier tipo de información a internet, ya sea en redes sociales o casillas de mensaje, esa información ya deja de ser de la persona y pasa a estar público, donde cualquiera puede tener acceso a ella y puede hacer uso, o mal uso, de la información.

Se puede ver cómo es que no se puede confiar en empresas gigantes como Facebook Inc., dueña de Facebook, Instagram y Whatsapp, tres de las aplicaciones más utilizadas a nivel mundial, o Google, dueña de YouTube, Gmail, Google Drive, entre otras, muestran publicidad relacionada a cosas que se estuvo buscando en internet y hasta a veces sólo hablando de ellas en voz alta (ya que hay aplicaciones a las que se les brinda acceso a los micrófonos y cámaras de los dispositivos que se usan diariamente por las personas). Facebook, al principio del 2018, tuvo que ir a juicio tras haber violado la privacidad de sus usuarios, confirmando por su fundador y CEO, Mark Zuckerberg, a CNN²⁰.

Cuestiones como la anteriormente explicada lleva a darnos cuenta que la privacidad y la protección de los datos son algo muy importante para las personas, y es eso lo que se trata de brindar a los usuarios de Balloon, implementando para ello el uso de una plataforma que se encarga de encriptar la información y aparte almacenarla de forma cifrada, anónima, sin terceras partes y para toda la vida en la Blockchain de Ethereum.

²⁰ [Nota Mark Zuckerberg a CNN - infobae](#)

Posibilidades de Crecimiento

Tras un análisis sobre las funcionalidades futuras que podría tener la Dapp Balloon y a su vez mejoras no solo en funcionalidad sino a nivel negocio y diseño, se ha detectado y a su vez por lo que nos han dicho personas o entidades cercanas que están al tanto de la Dapp que se está desarrollando, que algunas posibles mejoras a tener en cuenta podrían ser:

- Incluir un buscador dentro de la tabla de archivos, para tener un acceso más rápido a el archivo deseado.
- Desplegar la plataforma en Amazon, para que pueda ser escalable la cantidad de usuarios que utilicen la plataforma al mismo tiempo, suficiente como para poder alcanzar nuestro objetivo de al millón de usuarios por semestre.
- Incorporar un diseñador gráfico al equipo para mejorar la experiencia de usuario, logrando una interfaz más atractiva y con mayor profesionalismo.
- Posibilidad de ofrecer consultoría y asesoramiento en el Know How sobre tecnologías de Blockchain (sobre todo la de Ethereum) y red IPFS (InterPlanetary File System).
- Posibilidad de seleccionar en qué red de nodos desea el usuario almacenar sus archivos, no utilizando solo la de Infura, sino poder elegir entre otras existentes en el mercado. Colocando un selector para esta cuestión en el formulario de carga del archivo.
- Traer datos de la billetera digital MetaMask y que se muestren inyectados en la pantalla de Home de Balloon, así el usuario no tiene que estar entrando a MetaMask cada vez que desea consultar saldos, transacciones y demás que son de interés para el mismo.
- Conseguir incorporar un Lic. en Marketing al equipo para mejorar la experiencia de última milla que se quiere atacar con Balloon, logrando una llegada más efectiva a nuestros futuros clientes y usuarios.

Todas estas mejoras se irán desarrollando a medida que la Dapp Balloon comience a ser utilizada por los crecientes usuarios, de tal forma que atraiga a la mayor cantidad de

interesados posible, y sobretodo, mantenga a aquellos que ya se han unido a la plataforma siendo esto una imagen de conformidad de los mismos con Balloon.

Conclusión

Al finalizar este proyecto se puede concluir que se transitó por una experiencia de aprendizaje muy enriquecedora, donde, llevando a cabo el proceso completo del desarrollo de un producto software, aprendido a lo largo de toda la carrera de Ingeniería de Sistemas, se obtuvo un producto final de muy buena calidad lo cual nos deja muy conformes como desarrolladores del mismo.

Se pasó por un proceso inicial de aprendizaje de un nuevo lenguaje de programación, Solidity, proceso que nos llevó dos meses antes de poder comenzar a programar nuestro Smart Contract, que es también nuestro Back-End. Se realizó, también, el proceso de análisis de requerimientos del proyecto de software, donde se llevaron a cabo distintos métodos de toma de requerimientos, armado de diagramas de caso de uso, granularidades finas, diagramas de flujo, todo lo necesario para que el proceso de desarrollo de software sea más fácil. Y luego, para la etapa de diseño y desarrollo, se utilizaron metodologías ágiles, donde se pudieron hacer pequeñas iteraciones con valor funcional para el cliente (en este caso los desarrolladores en sí) e ir cambiando lo necesario para que se adapte a las necesidades que iban surgiendo. Se incluyó además la etapa de pruebas, que fue llevada a cabo en dos instancias, una con pruebas unitarias de código, programadas con JavaScript para que se corran de manera automática, y también casos de prueba donde se invitó a distintas personas para que pusieran a prueba nuestro código, logrando de esa forma mejorar la calidad del producto.

Para terminar, creemos que el proyecto realizado, tras su completo ciclo de vida, fue muy completo, aparte de complejo, con muy buena calidad, logrando resolver la problemática planteada a lo largo de este documento. Se espera que Balloon pueda ser utilizado y aprovechado como nueva herramienta para almacenar archivos de importancia para las personas de forma descentralizada y segura aprovechando las ventajas que ofrece la tecnología de Blockchain.

Anexos

Test Cases - Balloon

En esta sección del anexo se detallarán brevemente un set de Test Cases que se realizaron en dos sprints (cada 15 días) específicamente diseñados para testear la Dapp Balloon y se llevaron a cabo en el mes de Septiembre. El primer sprint comenzó el 01/09/2019 y terminó el 15/09/2019, mientras que el segundo sprint comenzó el 16/09/2019 y finalizó el 30/09/2019.

TC Nro: 01

Nombre: Iniciar sesión en MetaMask con cuenta activa

Sprint: 1

Fecha: 02/09/2019

Diseño TC: 0:30

Ejecución TC: 0:30

Versión: 1.0 Objetivos: Poder iniciar sesión en el wallet de MetaMask con cuenta activa Descripción: Iniciar sesión en MetaMask con una cuenta ya existente	1	Ingresar en el navegador al localhost en el puerto 8080 (http://localhost:8080/)	El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo una sección para la subida de un archivo. El sistema abre una ventana emergente de MetaMask donde pide ingresar la contraseña.
	2	El usuario ingresa la contraseña	El sistema de MetaMask ingresa correctamente a la cuenta y muestra una nueva pestaña donde pregunta si aceptar conectar MetaMask con Balloon y por debajo dos botones:

<p>Éxito del TC: Lograr iniciar sesión</p>			<p>-aceptar -rechazar</p>
<p>Fracaso del TC: cuenta no existente</p> <p>Ambiente de Testing: dev</p> <p>Pre-requisitos: cuenta en MetaMask ya existente</p> <p>Especificación de los datos de prueba: password : testWallet</p>	3	El usuario selecciona el botón "aceptar"	El sistema comprueba y cierra la ventana emergente de MetaMask y vuelve a la ventana del navegador donde está abierto el localhost.

TC Nro: 02

Nombre: Crear cuenta en MetaMask

Sprint: 1

Fecha: 05/09/2019

Diseño TC: 1:00

Ejecución TC: 1:00

<p>Versión: 1.0</p> <p>Objetivos: Crear</p>	1	Ingresar al localhost en el puerto 8080 (http://localhost:8080/)	El sistema muestra la pantalla principal con una barra de navegación con 2 opciones
---	---	---	---

<p>una cuenta en MetaMask</p> <p>Descripción: Al ingresar a la página web, si el usuario no tiene cuenta en MetaMask va a tener que crear una nueva cuenta.</p>			<p>(home y about us), abajo una sección para la subida de un archivo.</p> <p>El sistema abre una nueva pestaña en el navegador en el sitio web de MetaMask donde explica brevemente lo que es y debajo un botón "Get Started"</p>
<p>Éxito del TC: Cuenta creada con éxito.</p>	2	El usuario selecciona el botón "Get Started".	El sistema de MetaMask muestra una vista con 2 secciones uno con un botón que dice "Importar Wallet" y el otro con un botón que dice "Crear Wallet".
<p>Fracaso del TC: Fallo al crear una cuenta</p>	3	El usuario selecciona el botón "Crear Wallet"	El sistema muestra una vista con una descripción bajo el título de "Help us improve MetaMask" y al final 2 botones que dicen "No Thanks" y "I agree"
<p>Ambiente de Testing: dev</p> <p>Pre-requisitos: Ingresar a https://addons.mozilla.org/es/firefox/addon/ether-metamask/ y primero eliminar la extensión de</p>	4	El usuario selecciona el botón "I agree"	El sistema muestra una vista bajo el título de "Crear Contraseña" y debajo 2 campos para ingresar, en el primero una nueva contraseña con mínimo 8 caracteres y el segundo confirmar la contraseña, luego seleccionar el checkbox de estar de acuerdo con los términos de uso y al último muestra el botón "crear" bloqueado.

<p>MetaMask (con el botón “Eliminar”) y, luego de ser eliminado, volver a instalarlo (con el botón “+ Agregar a Firefox”) o https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaiaoeihlefknodbefgpgknn?utm_source=chrome-ntp-icon en caso de chrome, primero desinstalar la extensión de MetaMask (con el botón “desinstalar”) y, luego de ser desinstalado, volver a instalarlo (con el botón “añadir a chrome”).</p> <p>Especificación</p>	5	El usuario ingresa en el primer input la contraseña con más de 8 caracteres	El sistema permite ingresar la contraseña
	6	El usuario ingresa en el segundo input de nuevo la misma contraseña	El sistema permite ingresar la contraseña
	7	El usuario selecciona el checkbox de estar de acuerdo con los términos de uso	El sistema habilita el botón crear.
	8	El usuario selecciona el botón “crear”	El sistema muestra una vista con el título “Secret Backup Phrase” donde muestra un cuadro borroso con un mensaje que dice “haga clic aquí para revelar palabras secretas” y debajo 2 botones “remind me later” y “siguiente” que se encuentra bloqueado.
	9	El usuario hace click en el cuadro borroso	El sistema muestra 12 palabras random dentro del cuadro y se habilita el botón “siguiente”
10	El usuario selecciona el botón “Siguiente”	El sistema muestra una vista con un recuadro en blanco, debajo 12 botones, dentro de cada botón una de las 12 palabras anteriores que el usuario debe seleccionar en el mismo orden que el mensaje original, y al ultimo un botón “confirmar”, que	

<p>de los datos de prueba: https://metamask.io/</p> <p>Observaciones: -El usuario debe tomar nota de las 12 palabras que MetaMask brinda al momento de crear una nueva cuenta. -Al crear la Wallet, la misma posee 0 (cero) Ether, ya que se la acaba de crear no tiene dinero todavía cargado.</p>			está bloqueado
	11	El usuario debe seleccionar las palabras en el orden del mensaje de 12 palabras anterior	El sistema habilita el botón "confirmar"
	12	El usuario selecciona el botón "confirmar"	El sistema muestra una vista con un botón al final que dice "all done"
	13	El usuario selecciona el botón "all done"	El sistema muestra una vista para permitir conectar Balloon con MetaMask y debajo 2 botones "Cancelar" y "Conectar"
	14	El usuario selecciona el botón "Conectar"	El sistema abre la cuenta de MetaMask.
	15	El usuario hace click en el dropdown justo al lado del logo de MetaMask en la barra de navegación, donde dice "Red principal de Ethereum"	<p>El sistema muestra un drop down que tiene distintas blockchains para elegir:</p> <ul style="list-style-type: none"> - Red principal de Ethereum - Red privada Ropsten - Red de pruebas Kovan - Red privada Rinkeby - Goerli Test Network - Localhost 8545 - Rpc personalizado
16	El usuario selecciona la opción "Localhost 8545"	El sistema muestra la cantidad de Ether que posee la wallet en esa red (cero para este caso).	

TC Nro: 03**Nombre: Importar cuenta en MetaMask con semilla**

Sprint: 1

Fecha: 08/09/2019

Diseño TC: 1:00

Ejecución TC: 00:30

<p>Versión: 1.0</p> <p>Objetivos: Importar una cuenta de MetaMask utilizando la semilla</p> <p>Descripción: Por medio de una frase semilla (o seed phrase) compuesta de 12 palabras aleatorias únicas para cada cuenta, MetaMask permite recuperar una wallet</p> <p>Éxito del TC: recuperación de la wallet</p>	1	Ingresar al localhost en el puerto 8080 (http://localhost:8080/)	El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo una sección para la subida de un archivo. El sistema abre una nueva pestaña en el navegador en el sitio web de MetaMask donde explica brevemente lo que es y debajo un botón "Get Started"
	2	El usuario selecciona el botón "Get Started".	El sistema de MetaMask muestra una vista con 2 secciones uno con un botón que dice "Importar Wallet" y el otro con un botón que dice "Crear Wallet".
	3	El usuario selecciona el botón "Importar Wallet"	El sistema muestra una vista con una descripción bajo el título de "Help us improve MetaMask" y al final 2 botones que dicen "No Thanks" y "I agree"
	4	El usuario selecciona el botón "I agree"	El sistema muestra una vista con el título "Import an Account with Seed Phrase, debajo un cuadro

<p>Fracaso del TC: no se puede recuperar la wallet</p> <p>Ambiente de Testing: dev</p> <p>Pre-requisitos: Ingresar a https://addons.mozilla.org/es/firefox/addon/ether-metamask/, en caso de Firefox, y primero eliminar, o des instalar la extensión de MetaMask (con el botón "Eliminar") y, luego de ser eliminado, volver a instalarlo (con el botón "+ Agregar a Firefox"), o https://chrome.google.com/webstore/detail/metamask/nkbihfbegoaek/ehlefnkodbefgpgknn?utm_source=c</p>			<p>de texto para ingresar la Seed Phrase, luego un campo para ingresar la nueva contraseña (mínimo de 8 caracteres) y otro cuadro de texto para confirmar la contraseña, debajo de eso un checkbox para indicar que está de acuerdo con los términos de uso y por último el botón "Importar" bloqueado</p>
	5	El usuario ingresa la Seed Phrase en el primer cuadro de texto con cada palabra separada con un espacio	El sistema permite ingresar la Seed Phrase
	6	El usuario ingresa la nueva contraseña en el segundo cuadro de texto	El sistema permite ingresar la nueva contraseña
	7	El usuario ingresa la contraseña nuevamente en el último cuadro de texto	El sistema permite ingresar la nueva contraseña nuevamente
	8	El usuario selecciona el checkbox de estar de acuerdo con los términos de uso	El sistema habilita el botón "Importar"
	9	El usuario selecciona el botón "Importar"	El sistema muestra una vista con un botón al final que dice "all done"
	10	El usuario selecciona el botón "all done"	El sistema muestra una vista para permitir conectar Balloon con

<p>hrome-ntp-icon</p> <p>en caso de chrome, primero desinstalar la extensión de MetaMask (con el botón “desinstalar”) y, luego de ser desinstalado, volver a instalarlo (con el botón “añadir a chrome”).</p> <p>Especificación de los datos de prueba:</p> <p>-Seed Phrase: skill pretty million cheap panda guitar light orient dish noise trumpet panel</p> <p>-nueva contraseña: testWallet</p> <p>Observaciones: -Al cambiar de red la Wallet (de Red principal de</p>			MetaMask y debajo 2 botones “Cancelar” y “Conectar”
	11	El usuario selecciona el botón “Conectar”	El sistema abre la cuenta de MetaMask.
	12	El usuario hace click en la el drop down justo al lado del logo de MetaMask en la barra de navegación, donde dice “Red principal de Ethereum”	El sistema muestra un drop down que tiene distintas blockchains para elegir: <ul style="list-style-type: none"> - Red principal de Ethereum - Red privada Ropsten - Red de pruebas Kovan - Red privada Rinkeby - Goerli Test Network - Localhost 8545 - Rpc personalizado
	13	El usuario selecciona la opción “Localhost 8545”	El sistema muestra la cantidad de Ether que posee la wallet en esa red.

Ethereum a localhost) la misma posee 99.8315 ETH.			
---	--	--	--

TC Nro: 04**Nombre: Subir un archivo**

Sprint: 1

Fecha: 10/09/2019

Diseño TC: 1:00

Ejecución TC: 00:30

<p>Versión: 1.0</p> <p>Objetivos: Subir un archivo al sistema Balloon</p> <p>Descripción: Subir un archivo utilizando la wallet de MetaMask</p> <p>Éxito del TC: se logra subir el archivo</p> <p>Fracaso del TC:</p>	1	Ingresar al localhost en el puerto 8080 (http://localhost:8080/)	El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo una sección para la subida de un archivo donde se debe ingresar el título del archivo, debajo hay un botón "Choose a file" para seleccionar un archivo desde el ordenador y debajo un botón "upload file" para subir el archivo.
	2	El usuario ingresa el título en el cuadro de texto "title"	El sistema permite escribir el título
	3	El usuario selecciona el botón "choose a file"	El sistema abre el buscador de archivos

<p>El archivo no se puede subir</p> <p>Ambiente de Testing: dev</p> <p>Pre-requisitos:</p> <p>Especificación de los datos de prueba:</p>	4	El usuario selecciona el archivo que desea subir haciendo doble click	El sistema carga el archivo seleccionado por el usuario, mostrando debajo del botón "choose a file" el nombre del archivo seleccionado
	5	El usuario selecciona el botón "upload file"	El sistema bloquea los campos del formulario y muestra un loading gif mientras se conecta con el sistema de MetaMask. Al conectarse el sistema muestra una ventana de MetaMask que informa los costos de la transacción y debajo 2 botones "rechazar" y "aceptar".
	6	El usuario selecciona el botón "aceptar"	El sistema cierra la ventana emergente de MetaMask y, en el "home" de Balloon, al final de la página, se crea una tabla donde muestra información del archivo subido y al final de la información del mismo un botón "Download"

TC Nro: 05

Nombre: Descargar un archivo

Sprint: 1

Fecha: 14/09/2019

Diseño TC: 0:30

Ejecución TC: 0:30

<p>Versión: 1.0</p> <p>Objetivos: Descargar un archivo subido previamente</p> <p>Descripción: Poder descargar un archivo que se subió anteriormente al sistema</p> <p>Éxito del TC: el archivo se descarga</p> <p>Fracaso del TC: el archivo no se descarga</p>	1	<p>Ingresar al localhost en el puerto 8080 (http://localhost:8080/)</p>	<p>El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo una sección para la subida de un archivo donde se debe ingresar el título del archivo, debajo hay un botón "Choose a file" para seleccionar un archivo desde el ordenador y debajo un botón "upload file" para subir el archivo. Y debajo hay una tabla que contiene los archivos subidos previamente para esa Wallet Address que tiene los campos: "#" (id), file title, transaction datetime, download from ipfs. Este último campo solo tiene botones por cada archivo que dicen "download"</p>
<p>Ambiente de Testing: dev</p> <p>Pre-requisitos: logueado en MetaMask y un archivo ya subido</p> <p>Especificación de</p>	2	<p>El usuario selecciona el botón "download" del archivo correspondiente</p>	<p>El sistema descarga el archivo.</p>

los datos de prueba:			
----------------------	--	--	--

TC Nro: 06**Nombre: Rechazar la subida de un archivo**

Sprint: 2

Fecha: 17/09/2019

Diseño TC: 00:30

Ejecución TC: 00:30

<p>Versión: 1.0</p> <p>Objetivos: Rechazar una subida de archivo</p> <p>Descripción: El usuario desea cancelar la subida del archivo y el sistema debe mostrar un mensaje confirmando que se canceló la transacción</p> <p>Éxito del TC: El archivo no se sube</p>	1	Ingresar al localhost en el puerto 8080 (http://localhost:8080/)	El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo una sección para la subida de un archivo donde se debe ingresar el título del archivo, debajo hay un botón "Choose a file" para seleccionar un archivo desde el ordenador y debajo un botón "upload file" para subir el archivo.
	2	El usuario ingresa el título en el cuadro de texto "title"	El sistema permite escribir el título
	3	El usuario selecciona el botón "choose a file"	El sistema abre el buscador de archivos
	4	El usuario selecciona el archivo que desea subir haciendo doble click	El sistema carga el archivo seleccionado por el usuario, mostrando debajo del botón "choose a file" el nombre del

<p>Fracaso del TC: el sistema no permite cancelar la subida de archivo</p> <p>Ambiente de Testing: dev</p> <p>Pre-requisitos: estar logueado en MetaMask</p> <p>Especificación de los datos de prueba:</p>			archivo seleccionado
	5	El usuario selecciona el botón "upload file"	El sistema bloquea los campos del formulario y muestra un loading gif mientras se conecta con el sistema de MetaMask. Al conectarse el sistema muestra una ventana de MetaMask que informa los costos de la transacción y debajo 2 botones "rechazar" y "aceptar".
	6	El usuario selecciona el botón "rechazar"	El sistema cierra la ventana emergente del MetaMask y muestra, en el "home" de Balloon un popup con el mensaje: "The transaction was canceled"

TC Nro: 07**Nombre: Consultar sección About us**

Sprint: 2

Fecha: 20/09/2019

Diseño TC: 1:00

Ejecución TC: 0:30

<p>Versión: 1.0</p> <p>Objetivos: Acceder a la sección de About us</p>	1	El usuario ingresa a http://localhost:8080/	El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo una sección para la subida de un archivo (cualquier tipo de
--	---	--	--

<p>Descripción: El usuario consulta la sección About us para</p>			<p>archivo). En caso de tener un archivo subido, debajo de la sección para subir el archivo hay una tabla con los archivos subidos.</p>
<p>Éxito del TC: el usuario accede a la sección About us</p> <p>Fracaso del TC: El usuario no accede a la sección About us</p> <p>Ambiente de Testing: dev</p> <p>Pre-requisitos: logueado en MetaMask</p> <p>Especificación de los datos de prueba:</p>	2	<p>El usuario hace click en el botón "About us" en la barra de navegación a la derecha al fondo</p>	<p>El sistema muestra la sección About us que contiene: la barra de navegación con el logo y los botones de Home y About us, debajo un párrafo bajo el título "Welcome to Balloon", seguido de ese otro título "What do you need to upload a file?" con 2 links (metamask.io y una sobre como comprar Ether) y al final un footer que contiene los nombres de los creadores (Nicolás Grupe y Marcos Lozada)</p>

TC Nro: 08

Nombre: Consultar sección Home

Sprint: 2

Fecha: 24/09/2019

Diseño TC: 1:00

Ejecución TC: 0:30

<p>Versión: 1.0</p> <p>Objetivos: Acceder a la sección Home</p> <p>Descripción: Partiendo de la sección About us pasar a la sección Home</p> <p>Éxito del TC: El usuario puede acceder a la sección Home</p> <p>Fracaso del TC: El usuario no puede acceder a la sección Home</p> <p>Ambiente de Testing: dev</p> <p>Pre-requisitos: Estar logueado en MetaMask y haber subido algún archivo</p>	1	<p>El usuario ingresa a http://localhost:8080/about-us.html</p>	<p>El sistema muestra la pantalla About us que contiene una barra de navegación con el logo de Balloon y los botones Home y About us, debajo 2 párrafos con 2 títulos (Welcome to Balloon, What do you need to upload a file?)</p>
	2	<p>El usuario hace click en el botón "Home" en la barra de navegación a la derecha</p>	<p>El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo un formulario para la subida de un archivo (cualquier tipo de archivo) con los campos Title, File input y el botón "Upload File". Debajo una sección con una tabla con primero un texto "Total files saved" que dice la cantidad total de archivos para ese usuario, y luego una tabla con todos los archivos subidos con los campos: id, file title, transaction datetime y download from ipfs.</p>

(cualquier tipo de archivo) previamente			
Especificación de los datos de prueba:			

TC Nro: 09**Nombre: Ordenar la tabla de archivos**

Sprint: 2

Fecha: 26/09/2019

Diseño TC: 1:00

Ejecución TC: 0:30

Versión: 1.0 Objetivos: Ordenar la tabla de los archivos por distintos campos Descripción: Ordenar la tabla de archivos por id o por título de los archivos Éxito del TC: la tabla se ordena	1	El usuario ingresa a http://localhost:8080/	El sistema muestra la pantalla principal con una barra de navegación con 2 opciones (home y about us), abajo una sección para la subida de un archivo y debajo la tabla de archivos subidos.
	2	El usuario selecciona la flecha apuntando hacia abajo color gris a la derecha de File Title en la segunda columna de la tabla de archivos al final de la página	El sistema ordena los archivos de la tabla en orden alfabético según los títulos de los archivos
	3	El usuario selecciona la flecha apuntando hacia abajo color gris a la derecha de # en la primer	El sistema ordena los archivos de la tabla en orden de menor a mayor según los id de los

Fracaso del TC: la tabla no se ordena		columna de la tabla de archivos al final de la página	archivos
Ambiente de Testing: dev			
Pre-requisitos: logueado en MetaMask y haber subido mínimo 3 archivos previamente			
Especificación de los datos de prueba:			

TC Nro: 10**Nombre: Verificar subida de archivo**

Sprint: 2


Fecha: 29/09/2019

Diseño TC: 1:00

Ejecución TC: 0:30

Versión: 1.0 Objetivos: Verificar si el	1	Inmediatamente después de subir el archivo presionar en el teclado F12 (algunos casos Fn+F12)	El sistema abre las herramientas para desarrolladores de Chrome que tiene una barra de navegación
---	---	---	---

<p>archivo se subió encriptado</p> <p>Descripción:</p>			<p>con las opciones: elements, console, source, network, performance, memory, application, security, audit</p>
<p>Verificar en la red ipfs si el archivo se subió</p>	2	El usuario selecciona Console	El sistema muestra la consola de Chrome
<p>encriptado</p> <p>Éxito del TC: al ingresar al link ipfs se debe ver una cadena de caracteres sin sentido</p> <p>Fracaso del TC: al ingresar al link ipfs se debe ver el archivo que se subieron previamente</p> <p>Ambiente de Testing: dev</p> <p>Pre-requisitos: logueado en MetaMask, haber subido un archivo justo antes de ejecutar este TC</p>	3	El usuario hace click en el link de ipfs que se encuentra al lado del mensaje "The file was saved in IPFS: (ipfs link)"	El sistema abre en una nueva ventana el link de ipfs donde está guardado el archivo donde se muestra la información del archivo



Especificación de los datos de prueba:			
--	--	--	--

Anteproyecto

En esta sección del Anexo, se encuentra una descripción, el objetivo principal, los objetivos específicos y la división de etapas que se propusieron y se pactaron cumplir en un principio del año 2019, específicamente en Marzo, para poder alcanzar con nuestra Dapp Balloon hacia fines de Diciembre del 2019.

Este anteproyecto si bien ha mutado para mejor en varios ámbitos, siempre autorizado por los directores del trabajo y en base a nuestras propuestas de mejora, siempre ha sido de conformidad y se cree se ha podido cumplido con el compromiso y logro personal sobre todo, que es lo que nos viene a nutrir este desarrollo innovador con tecnologías disruptivas.

DESCRIPCIÓN

Los alumnos de quinto año de Ingeniería de sistemas: Nicolas Grupe y Marcos Lozada, en motivo de realizar el TRABAJO FINAL para nuestra carrera en la Universidad Católica de Córdoba, se ha propuesto un proyecto que consiste en desarrollar un sistema de almacenamiento de archivos en nube descentralizado, utilizando para este fin protocolos existentes que utilizan tecnología P2P (Peer-to-Peer) y Blockchain: IPFS (InterPlanetary File System) y Ethereum.

IPFS (<https://ipfs.io>) es un protocolo de almacenamiento de archivos peer-to-peer que permite guardar archivos en una red descentralizada. En otras palabras, IPFS proporciona un modelo de almacenamiento en bloque con dirección al contenido de alto rendimiento, y a su vez con un conjunto de hipervínculos direccionados al contenido. IPFS no tiene un solo punto de fallo, y los nodos no necesitan confiar entre sí.

Ethereum es una red de Blockchain (base de datos distribuida) que cuenta con Smart Contracts (Contratos Inteligentes). Los Smart Contracts son piezas de código que se almacenan en la red Blockchain de Ethereum, y que pueden programarse para cumplir funciones específicas a través de un lenguaje de programación llamado Solidity. Una vez que el contrato se guarda en la red de Ethereum, goza de las ventajas de la tecnología Blockchain, la cual asegura la descentralización e inmutabilidad del mismo, brindando fiabilidad y consenso entre nodos que trabajan juntos.

El proyecto propuesto implica la utilización de las tecnologías IPFS y Ethereum para el desarrollo de una DApp (Decentralized App - es una aplicación de computadora que se ejecuta en un sistema informático distribuido) con la capacidad para almacenar archivos del usuario en la red IPFS y guardar un índice de los mismos en un Smart Contract migrado a la Testnet de Ethereum (BC). La DApp permitirá al usuario dos opciones: en primer lugar, cargar un archivo de forma simple sin cifrar, o sino, la segunda opción que consiste en que el mismo esté cifrado con un protocolo como por ejemplo AES si es realmente importante para el usuario, generando el almacenamiento del archivo 100% cifrado.

El front-end de la DApp se realizará en HTML5, CSS3 y JavaScript, utilizando un complemento llamado MetaMask (de Consensys Labs), que permite crear un wallet con MetaMask y automáticamente inyectar la librería Web3 en el browser.

OBJETIVO GENERAL

El objetivo de esta dapp es permitirle al usuario poder tener un almacenamiento de archivos en la nube descentralizado, basado en blockchain, de su interés; los mismos podrán ser cifrados o no a la hora de guardarse y quedarán almacenados en una red descentralizada llamada IPFS, que mediante un Smart Contract alojado en la Blockchain de ETH, almacene un índice al archivo guardado permitiendo tener una correcta y rápida administración de los mismos por el usuario en su cuenta y que no haya alteración alguna del archivo.

OBJETIVOS ESPECÍFICOS

- Asegurar al usuario un almacenamiento 100% privado y fiable.
- Garantizar el registro de cuando un archivo ha sido guardado.
- Garantizar que los archivos almacenados permanecen en la red IPFS mediante el Smart Contract que se aloja en la Blockchain de ETH.
- Garantizar que el proyecto funcione en la Testnet de ETH.
- Ver transacciones en tiempo real a través de Etherscan.
- Presentar nuestro TF el 20/12/2019.

EL DESARROLLO SE DIVIDIRÁ EN 3 ETAPAS QUE SE DESCRIBEN A CONTINUACIÓN:

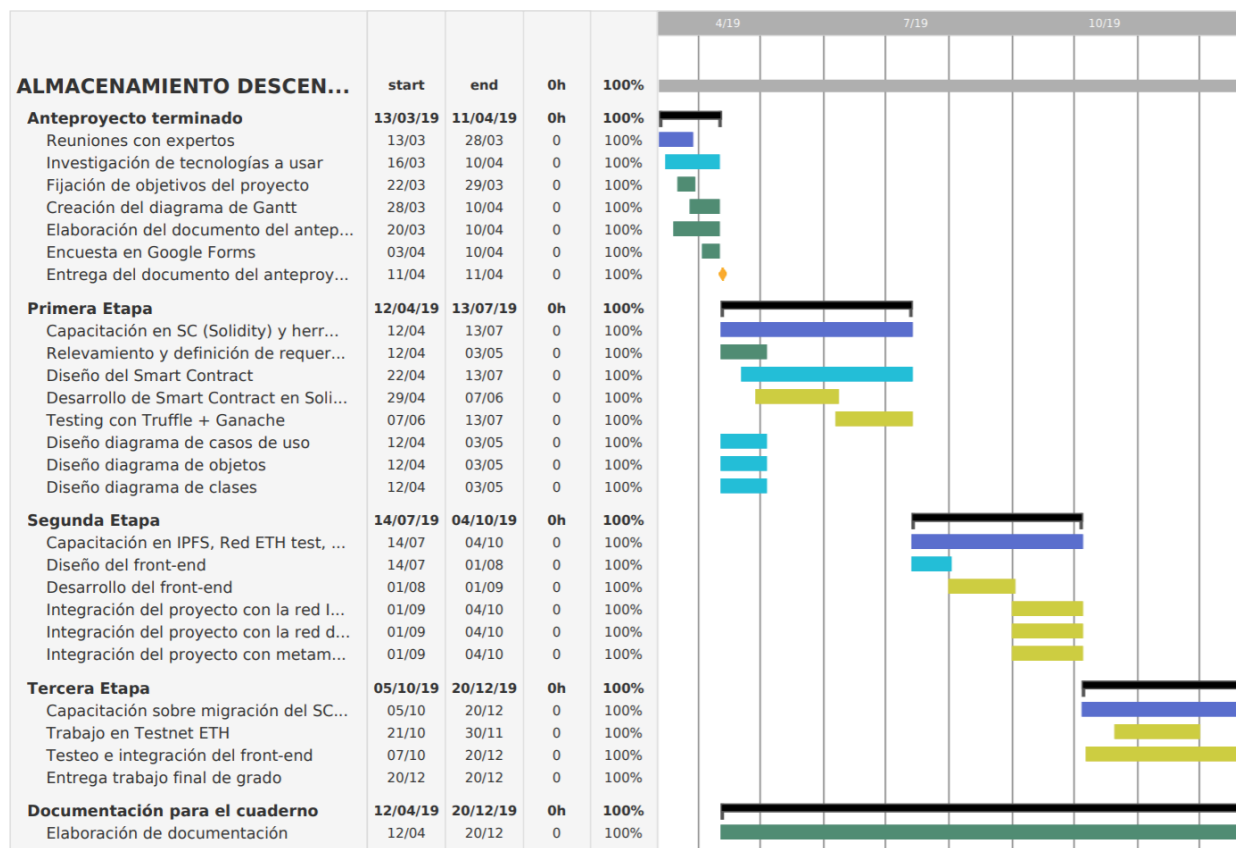
En la primera etapa se incluye el diseño y desarrollo del Smart Contract en Solidity, utilizando como IDEs Remix en primer lugar (para programar en Solidity) y luego testing con Truffle + Ganache (Truffle es un IDE para migración y testing, Ganache permite crear una Blockchain local propia para testing).

La segunda etapa consiste en el desarrollo del front-end y la integración con la red IPFS, la red Ethereum de testing en Ganache y MetaMask.

En la tercera etapa se migraría el contrato y se trabajaría directamente en la Testnet de Ethereum, dando los retoques finales en el código del front-end. Se puede incluir alguna función adicional si el tiempo lo permite.

Diagrama de Gantt

El diagrama de Gantt es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. En nuestro caso, se dividió el proyecto en 4 etapas: anteproyecto, primera etapa, segunda etapa y tercera etapa (cada una de las etapas fue explicada en el anteproyecto en la sección anterior de los anexos).



Glosario

- **User Friendly:** algo que es fácil de usar, es fácil de usar para las personas.
- **Crypto Holder:** personas o entidades con conocimiento sobre crypto activos, que tienen preferentemente alguna moneda digital y están al tanto de las tecnologías que sustentan estas cuestiones.
- **Pop Up:** es una ventana emergente con contenido que aparece de forma repentina en un navegador web o en la pantalla de la computadora del usuario. Se abre para mostrar un contenido complementario, que pueda ser de interés, vinculado a la página y al accionar del usuario o que simplemente busque reflejar publicidad sobre una marca o negocio.
- **Seed (Seed Phrase):** es una lista de palabras que almacenan toda la información necesaria para recuperar una billetera digital, como por ejemplo la de MetaMask. El software de la billetera digital generalmente brindara una frase inicial y le indicará al usuario que la escriba en papel. Si la computadora del usuario se rompe o su disco duro se corrompe, puede descargar el mismo software de billetera digital nuevamente y usar la copia de seguridad en papel para recuperar su billetera virtual.
- **Post-It:** hace referencia a pequeñas hojas de papel autoadhesivo de varias dimensiones, formas y colores.
- **Test Case:** un caso de prueba o “Test Case” es un conjunto de condiciones o variables bajo las cuales un analista determinará si una aplicación, un sistema software (software system), o una característica de éstos es parcial o completamente satisfactoria.
- **Sprint:** es el nombre que va a recibir cada uno de los ciclos o iteraciones que existen dentro de dentro de un proyecto Scrum. Estos ciclos permiten tener un ritmo de trabajo con un tiempo prefijado, siendo la duración habitual de un Sprint unas cuatro semanas, aunque lo que la metodología dice es que debería estar entre dos semanas y un máximo de dos meses. En cada Sprint o cada ciclo de trabajo lo que se consigue es un entregable o incremento del producto, que aporte valor al cliente.

Fuentes

<https://solidity.readthedocs.io/en/v0.5.6/introduction-to-smart-contracts.html>

<https://ipfs.io/>

<https://tech.tribalyte.eu/blog-introduccion-a-la-programacion-de-un-smart-contract>

<https://etherscan.io/>

<https://truffleframework.com/ganache>

<https://www.ethereum.org/>

<https://consensys.net/academy/bootcamp/>

<https://ethgasstation.info/>

<https://truffleframework.com/docs/truffle/overview>

<https://metamask.io/>

<https://sia.tech/sia.pdf>

<https://nodejs.org/en/about/>

<https://www.w3schools.com/js/>

<https://www.w3schools.com/nodejs/>

<https://github.com/ethereum/EIPs/issues/930>

<https://infura.io/product>