Universidad Europea

LAUREATE INTERNATIONAL UNIVERSITIES

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

PRIMERA ENTREGA TRABAJO FIN DE MASTER

VALIDACIÓN DE DOCUMENTOS MEDIANTE BLOCKCHAIN

ALBERTO RODRIGUEZ TENA

CURSO 2018-2019

TÍTULO: VALIDACIÓN DE DOCUMENTOS MEDIANTE BLOCKCHAIN

AUTOR: ALBERTO RODRIGUEZ TENA

TITULACIÓN: MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

DIRECTOR DEL PROYECTO: JUAN ANTONIO PIÑUELA IZQUIERDO

FECHA: MAYO de 2019

Índice

Capítulo 1. INTRODUCCIÓN

1.1 Planteamiento del problema

Uno de los grandes problemas en la actualidad es conocer si la información que recibimos o enviamos mayoritariamente mediante medios electrónicos es recibida de la manera mas adecuada tanto en tiempo como en forma. Este proyecto surge para dar respuesta a esta necesidad, la validación de documentos electrónicos. Aunque, actualmente es cierto que existen medios a través de los cuales se da respuesta a esta necesidad, mediante aplicaciones como *DocuSign* o *HelloSign*, por qué no mezclar esta necesidad con una de las tecnologías que está más en auge actualmente, blockchain y que ofrece como características fundamentales la confianza, la transparencia y la inmutabilidad.

1.2 Objetivos del proyecto

Como se ha comentado en el apartado anterior el objetivo principal de este proyecto es validar documentos electrónicos dejando así a disposición del usuario documentos veraces y fiable.

Esto se llevará a cabo mediante la realización de una aplicación en la que el usuario introducirá aquellas personas que quiera que validen el documento a los cuales les llegará un aviso mediante correo electrónico y estos validarán o no el documento. En posteriores apartados se explicará de manera más detallada como se produce esta validación.

Capítulo 2. TECNOLOGÍA UTILIZADA

En este capítulo vamos a describir las tecnologías que han sido utilizadas para la realización de este proyecto y cuyo concepto es importante para tener una mejor compresión del proyecto.

2.1 Python

Python es un lenguaje de programación de propósito general, orientado a objetos, programación imperativa y en menor medida programación funcional. Debido a las características citadas anteriormente se puede decir que Python es un lenguaje de uso sencillo y de fácil aprendizaje y, por tanto, fue la opción mas idónea de entre los lenguajes de programación para llevar a cabo este proyecto. Fue creado por Guido Van Rossum, con el objetivo de cubrir la necesidad de un lenguaje orientado a objetos de sencillo uso que se podrá utilizar para tratar diversas tareas dentro de la programación, que habitualmente se hacían en Unix usando C. Python es un lenguaje de scripting, independiente de plataforma y como ya se ha comentado orientado a objetos, esto le permite a este lenguaje estar preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso páginas web. Es un lenguaje interpretado esto quiere decir que no se necesita compilar el código fuente para poder ejecutarlo, esto ofrece grandes ventajas como la rapidez de desarrollo y como inconveniente menor velocidad. Python dispone de dos versiones que son usadas en la actualidad la 2.x y la 3.x esta última incluye una serie de cambios que hacen necesario tener que reescribir el código de versiones anteriores. La versión utilizada en este proyecto es 3.6.6.

2.2 Django

Django es un framework (conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente) para aplicaciones web gratuito y open source escrito en Python. Fue desarrollado en principio para gestionar varias páginas orientadas a noticias de World Company de Lawrence, Kansas y fue liberada al público bajo una licencia de BSD (Distribución de Software Berkeley). El principal objetivo que tiene Django es facilitar la creación de sitios web complejos, además, pone bastante hincapié en reusar la conectividad y extensibilidad de componentes de desarrollo rápido y el principio DRY (No te repitas). Python es utilizado en todas las partes de framework, incluso en configuraciones, archivos y en los modelos de datos. Django requiere de una versión de Python 2.5 o superiores en este proyecto se ha utilizado la versión de Python 2.7.6. No se necesitarán por tanto otras bibliotecas de Python para obtener una funcionalidad básica. Para la realización de este proyecto se ha utilizado la versión de Django 1.9.5.

2.3 SQLite

SQLite es una biblioteca escrita en C que implementa un motor de base de datos SQL de dominio público, de alta confiabilidad, incorporado y con todas las funciones. Es compatible con ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Es el motor de base de datos mas utilizado en el mundo. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca de SQLite se enlaza con el programa pasando a ser parte integral del mismo. En su versión 3, que es la que se ha utilizado para este proyecto SQLite permite bases de datos de hasta 2 Terabytes de tamaño y también permite la inclusión de campos de tipo BLOB (Objetos binarios grandes).

2.4 HTML5

HTML5 es el lenguaje de marcado estándar para crear páginas Web. HTML significaHyper Text Markup Language, es la ´ultima versión de HTML.
El término representa dos conceptos diferentes, se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos. Contiene un conjunto mas amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser mas diversas y de gran alcance. Diseñado para ser utilizable por todos los desarrolladores de OpenWeb, esta página referencia numerosos recursos sobre las tecnologías de HTML5, clasificados en varios grupos según su función.

2.5 CSS3

CSS3 es la última versión de CSS, es un lenguaje que define el estilo o la apariencia de las páginas web, escritas en HTML o documentos XML.
Fue creada para separar el contenido de la forma y esto permite a los diseñadores tener un control mayor de la apariencia de la página.

2.6 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. JavaScript es un lenguaje que se utiliza principalmente en su forma del lado cliente, implementado como parte de un navegador web lo que permite mejoras en la interfaz de usuario y páginas web, existe además una forma de JavaScript del lado del servidor. También es significativo su uso en aplicaciones externas a la web. Fue diseñado con una sintaxis muy similar a C, aunque tiene también convenciones y nombres de Java.

2.7 BootStrap

BootStrap es un framework de código abierto para diseño de sitios y aplicaciones web, contiene elementos de diseños basados en HTML y CSS, así como extensiones de JavaScript opcionales. Es conocido como el proyecto mas popular de GitHub, fue desarrollado por Mark Otto y Jacbod Thornton de Twitter como un marco de trabajo que fomentara las herramientas internas.
Está mantenido por Twitter. Es compatible con la mayoría de los navegadores web.

2.8 Blockchain

Blockchain es una base de datos distribuida, protegida criptográficamente, en la que cada nodo tiene una copia de toda la información almacenada. Blockchain nace en 2008 como respuesta a la necesidad de Bitcoin de tener una base de datos que fuera segura, pero sin la necesidad de un administrador de confianza.

Sus principales características son:

Es un sistema transparente en el que todas las transacciones son registradas y se podrían consultar públicamente, pudiendo hacer un seguimiento del funcionamiento de la Plataforma.

Es una red de confianza, la confianza reside en el consenso entre los participantes, sin que exista la figura de un tercero.

Es un sistema inmutable, una vez introducida, la información, no se puede modificar por parte de ningún actor implicado ni ningún administrador.

Es un ledger distribuido, Blockchain es una base de datos descentralizada, protegida criptográficamente, en la que cada nodo tiene una copia de toda la información almacenada.

Es un paradigma de colaboración, se reduce la fricción entre los distintos participantes en base a unas reglas bien definidas y de fácil adopción, sin necesidad de un gran nivel de acuerdo.

2.9 Smart Contract

Contratos verificables y ejecutables por sí mismos, sin la necesidad de un tercero de confianza. Se pueden almacenar sobre Blockchain Smart Contracts, contratos autoejecutables y escritos en lenguaje de programación, ajenos al control de nadie y, por tanto, en los que cualquiera puede confiar. Las partes definirían el objeto del contrato, las acciones que se pueden realizar sobre él y las cláusulas de aplicación. Al guardarse en la Blockchain, este sería inmutable y se ejecutaría solo al cumplirse las condiciones acordadas. Aumenta la seguridad y la eficiencia, así como reduce costes en la ejecución y el mantenimiento.

2.10 Truffle

Es el framework más popular para el desarrollo en Ethereum hoy en día, Truffle ofrece Compilación, enlace y despliegue de Smart contracts desde el propio framework, Depuración y testing automatizado de contratos, framework con scripts de despliegue y migraciones en redes públicas y privadas, acceso a cientos de paquetes externos y gestión con EthPM & NPM Consola interactiva para comunicación directa con los contratos, interacción con contratos mediante scripts externos.

2.11 Infura

Proporciona un acceso seguro, confiable y escalable a las redes Ethereum. La cual nos permite acceder sin problemas a Ethereum a través de los nodos de carga equilibrada de Infura y la arquitectura inteligente de la misma manera que lo haría a través de sus propios nodos. Esta formado por servicios y API en torno a JSON RPC sobre HTTPS y WebSockets que puede usar con sus bibliotecas y marcos favoritos, en las cuatro redes Ethereum.

Capítulo 3. APLICACIÓN

Para la realización de este proyecto se realizará una aplicación en un framework web de Python llamado Django que servirá para poder validar documentos de forma online. Una vez verificado el documento se enviará una transacción a la blockchain para que esta validación quede registrada en ella.

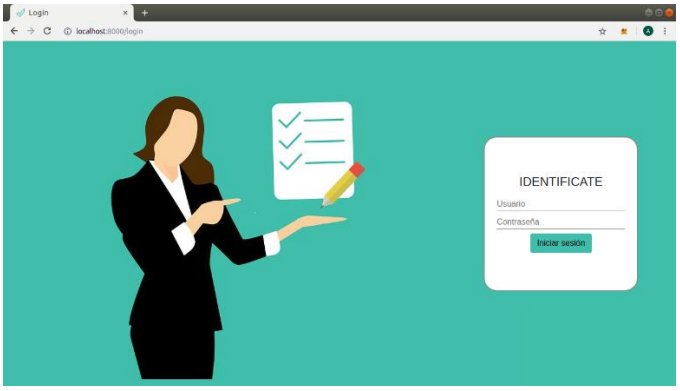
Esta blockchain será una red pública basada en Ethereum ya que es una plataforma open source, descentralizada que permite la creación de acuerdos de contratos inteligentes entre pares, basada en el modelo blockchain. Estas redes además nos ofrecen una alta disponibilidad y un fácil acceso. Debido a que son las redes blockchain mas globalizadas y maduras.

Para comunicarnos con este tipo de redes desde la aplicación utilizaremos una API llamada Infura la cual nos permite poder interactuar con la blockchain tanto a la hora de hacer transferencias como de obtener información de la blockchain.

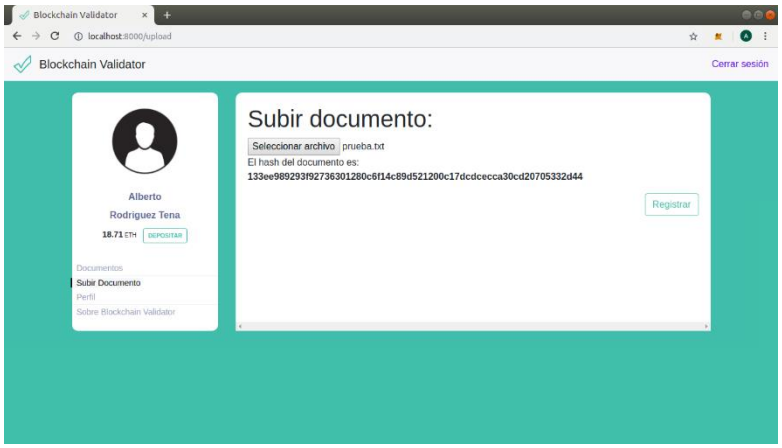
Esta aplicación permitirá al usuario subir documentos de los cuales necesita validación por el resto de las partes implicadas y así confirmar que tengan el aprobado de todas las partes. En el caso de que así sea, esta información se mandará al contrato inteligente desarrollado para este proyecto el cual se despliega dentro de la blockchain. Esta invitación se hará a través de correo electrónico a los usuarios implicados.

3.1 Funcionamiento

La aplicación está pensada para que su utilización sea sencilla y reporte al usuario una experiencia de uso cómodo y rápido. Constará de una página principal en el que el usuario en primera instancia deberá identificarse. Una vez identificado accederá a su página personal donde encontrará los diferentes apartados que la aplicación pone a su disposición. A continuación, se detalla el funcionamiento interno y las bases en las que están fundamentadas las diferentes acciones que son posibles de realizar en la aplicación. En la siguiente imagen se muestra la pantalla de inicio donde el usuario debe registrarse para acceder a la aplicación.



Una vez que el usuario lleve a cabo el proceso de registro y/o identificación tendrá a su disposición la posibilidad de subir documentos esto se realiza mediante la selección del documento por parte del usuario, una vez subido se le calcula y asigna un hash el cual será único y unidireccional, este hash se calcula mediante la función hash "SHA256". Una vez calculado el hash se manda la transacción a la Blockchain. El hash creado se guardará en el Smart contract creado para la aplicación, por tanto, desde ese momento el documento quedará asociado al wallet del usuario el cual se creará en el momento que el usuario se registra en la aplicación haciendo así que si otro usuario intenta subir el mismo documento este será avisado de que este documento ya está registrado por otro usuario (especificando además el nombre del usuario propietario). Esta transacción transcurridos unos segundos será confirmada en la blockchain por los por los nodos validadores de la red. A continuación, se adjunta captura de la aplicación donde se puede observar el hash calculado para subir un documento.



Completado el proceso de registro y confirmación en la blockchain del documento la aplicación permitirá al usuario la posibilidad de enviar a los usuarios una solicitud de colaboración.

Esta solicitud la enviará el usuario propietario del documento a través de una pestaña en la que introducirá el email del usuario al que quiera añadir como colaborador para verificar el archivo. Esto generará un correo de invitación que le llegará al usuario colaborador. En el caso de que este usuario no esté registrado en la aplicación tendrá que registrarse para poder validar o comentar cualquier documento al que haya sido invitado como colaborador.

Esta aplicación no solo ha sido creada para que el documento sea validado por los colaboradores, sino también para permitir al usuario creador del documento la posibilidad de ofrecer documentos con la mayor veracidad posible obteniendo no solo validación por parte de los colaboradores sino también la posibilidad de ver los comentarios que los colaboradores consideran relevantes para la posterior validación del documento. Estas validaciones se mandarán a la blockchain para que también queden reflejadas en ella. Ofreciendo la posibilidad a los usuarios de la aplicación ver aquellos documentos dentro de su perímetro con mayor número de validaciones.

3.2 Estados del documento

Una vez los documentos han sido subidos a la aplicación podrán encontrarse en los siguientes estados que estarán representados por colores en función de la criticidad del estado en el que se encuentren. A continuación se detallan los estados por orden de criticidad partiendo de un nivel menor a uno mayor.

- Documentos pendientes de ser confirmados por la Blockchain cuando un documento se encuentra en este estado implica que los nodos validadores todavía no han confirmado la transacción de registro del documento. Este estado estará representado por el color blanco.

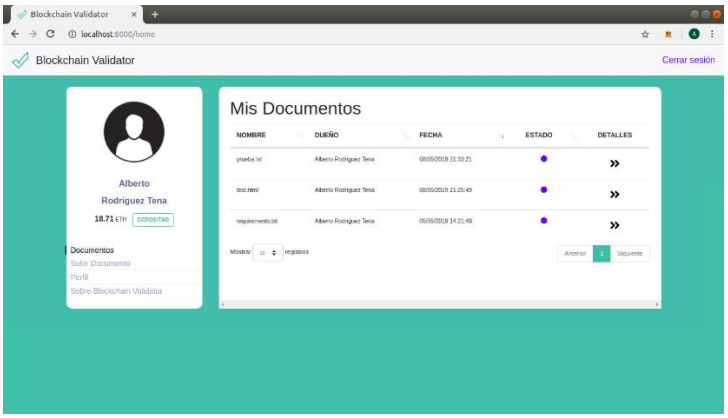
- Documentos confirmados por la blockchain que no tienen colaborador, en este estado se encontraran cuando no se haya solicitado colaboración por parte del usuario propietario del documento. Este estado estará representado por el color azul.

- Documentos que tienen colaboradores, pero estos no han colaborado con el documento. En este estado se encontrarán los documentos en los cuales el usuario propietario haya mandado una solicitud de colaboración y esta solicitud no ha obtenido respuesta. Este estado estará representado por el color morado.

- Documentos validados por colaboradores son los documentos validados por los usuarios colaboradores a los que el usuario propietario del documento ha enviado una solicitud para colaborar con el documento. Este estado estará representado por el color verde.

- Documentos con comentarios de colaboradores que un documento tenga asociado esta anotación implica que el usuario colaborador ha dejado un comentario con aquellos aspectos en los que considere necesario una mejora por parte del usuario propietario del documento. Este estado estará representado por el color rojo.

En la siguiente imagen se observa una captura de la aplicación donde se puede ver el estado de algunos documentos.



Capítulo 4. BIBLIOGRAFÍA

BootStrap. Web sobre bootstrap. https://getbootstrap.com/docs/4.3/getting-started/introduction/
Css3. Web sobre css3.https://devdocs.io/css-color/.
Django. Web oficial de django project. https://docs.djangoproject.com/en/2.2/.
Html5. Web sobre html5. https://www.w3.org/TR/html52/.
SQlite. Web oficial de sqlite. https://www.sqlite.org/
Infura. Web oficial de Infura. https://infura.io/docs
Truffle. Web oficial de Truffle. https://truffleframework.com/docs/truffle/overview.