



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

**SISTEMA DE GESTION DE DONACIONES DE ONGs
CON LA TECNOLOGIA BLOCKCHAIN**

Autor: Lucas Alberto Martínez Luna

Director: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
**SISTEMA DE GESTION DE DONACIONES DE ONGs CON LA TECNOLOGIA
BLOCKCHAIN**


en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2021/22 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Lucas Alberto Martínez Luna

Fecha: 01/07/2022

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón Fecha: 01/07/2022



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

**SISTEMA DE GESTION DE DONACIONES DE ONGs
CON LA TECNOLOGIA BLOCKCHAIN**

Autor: Lucas Alberto Martínez Luna

Director: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Madrid

Agradecimientos

Agradecimiento especial a mi familia, por animarme y apoyarme siempre, y por todo el esfuerzo que han realizado por el desarrollo de mi formación personal y académica.

Agradecer a todos los profesores y compañeros de la Universidad de Comillas que han contribuido a mi formación personal y académica.

Agradecer especial a mi director de TFG, Atilano Fernández-Pacheco, por su comprensión y apoyo durante el desarrollo del proyecto,

Agradecer a la Universidad de Wisconsin-Stout, por su magnífica acogida durante mi intercambio, y a todos los profesores, compañeros y amigos que han hecho de esta una experiencia inolvidable.

Por último, agradecer también a mis amigos que han estado han estado día a día apoyando y animándome para conseguir todos los objetivos que me he propuesto.

SISTEMA DE GESTION DE DONACIONES DE ONGs CON LA TECNOLOGIA BLOCKCHAIN

Autor: Martínez Luna, Lucas Alberto.

Director: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En este Proyecto de Fin de Grado se diseña, desarrolla e implementa una aplicación web basada en la tecnología blockchain para la gestión y manejo de donaciones de las ONGs.

Blockchain, además de no necesitar de terceros para realizar las transacciones, y como consecuencia incrementar el porcentaje de gastos dedicado a las misiones de estas organizaciones, ofrece también una total fiabilidad, seguridad y transparencia en las donaciones.

Para ello, se desarrollará una aplicación web que interactúa con los contratos que se despliegan sobre la red de blockchain, Ethereum, por medio un backend que además de gestionar la base de datos con la información de los proyectos de estas organizaciones, también gestionará todas las peticiones del frontend, así como la comunicación entre los contratos desplegados y la interfaz gráfica.

Palabras clave: Blockchain, Ethereum, Smart Contract, ONG, Web 3.

1. Introducción

Blockchain surge en 2008 gracias al primer documento publicado por Satoshi Nakamoto con el principal objetivo de almacenar y transferir criptomonedas sin la participación de los bancos.

Las transferencias internacionales, aparte de requerir una compleja infraestructura tecnológica, también suponen un coste elevado para las entidades participantes. Blockchain permite agilizar estas transferencias sin perder seguridad ofreciendo una total transparencia y trazabilidad de estas. Esto es posible gracias a la automatización de los pagos, por medio del uso de los Smart Contracts.

Las aplicaciones del blockchain son muchas, una de ellas es la gestión de donaciones de ONGs, aplicación que se pretende realizar y desarrollar a lo largo de este Trabajo Fin de Grado.

2. Definición del proyecto

En este proyecto se diseña un modelo nuevo para la gestión de donaciones de las organizaciones no lucrativas mediante el desarrollo de una aplicación web que se apoya en la tecnología blockchain a través del uso de Smart Contracts.

Para esta aplicación web se va a desarrollar un token por medio de un Smart Contract, con el cual los diferentes usuarios del sistema podrán realizar sus donaciones, quedándose estas almacenadas en la blockchain.

La implementación de este modelo debe servir como impulso para que sean más las personas que realicen donaciones, sabiendo que estas contarán con una mayor seguridad, fiabilidad y transparencia.

La verificación del funcionamiento de la aplicación se va a llevar a cabo en Ganache, este framework ofrece una red local simulada en Ethereum para realizar las pruebas pertinentes.

3. Descripción del sistema

El diseño de la arquitectura de la aplicación que se representa en la Ilustración 1, consta de ciertos elementos fundamentales. En primer lugar, encontramos a los clientes, quienes tienen acceso a la aplicación web y a los recursos almacenados en la blockchain y la base de datos.

En segundo lugar, se encuentra Spring Boot Application, este tiene embebido el web servlet de Tomcat. Además, para el desarrollo web de este proyecto se ha hecho uso de Spring Web quien a su vez hace uso de Spring MVC, este se encarga de soportar todas las peticiones web (HTTP) a la API, y redirigirlas a los componentes que sean necesario.

El framework de Spring Data JPA a través de un conjunto de interfaces y APIs persisten los recursos en una base de datos relacional. Hibernate facilita mediante diferentes Beans el mapeo de atributos en bases de datos, además del modelo de objetos de una aplicación.

Como gestor de la base de datos relacional, se ha escogido PostgreSQL. Este gestor almacenará los recursos de los usuarios, los proyectos y las organizaciones.

Por último, encontramos los componentes relaciones con la tecnología blockchain. El primero de estos elementos es Web3, quien se encarga de establecer comunicación entre el usuario y la blockchain de Ethereum. A través de esta librería se puede acceder directamente a los datos almacenados en la blockchain, en este caso, las donaciones, o podemos acceder a Metamask, para poder así aceptar las transacciones que se quieran realizar.

Dentro de la red de blockchain de Ethereum se encuentra el último componente, Ganache, en el cual se puede realizar la simulación y testeo de la blockchain de Ethereum a través de una red de pruebas local. En este framework se despliegan los Smart Contracts compilados.

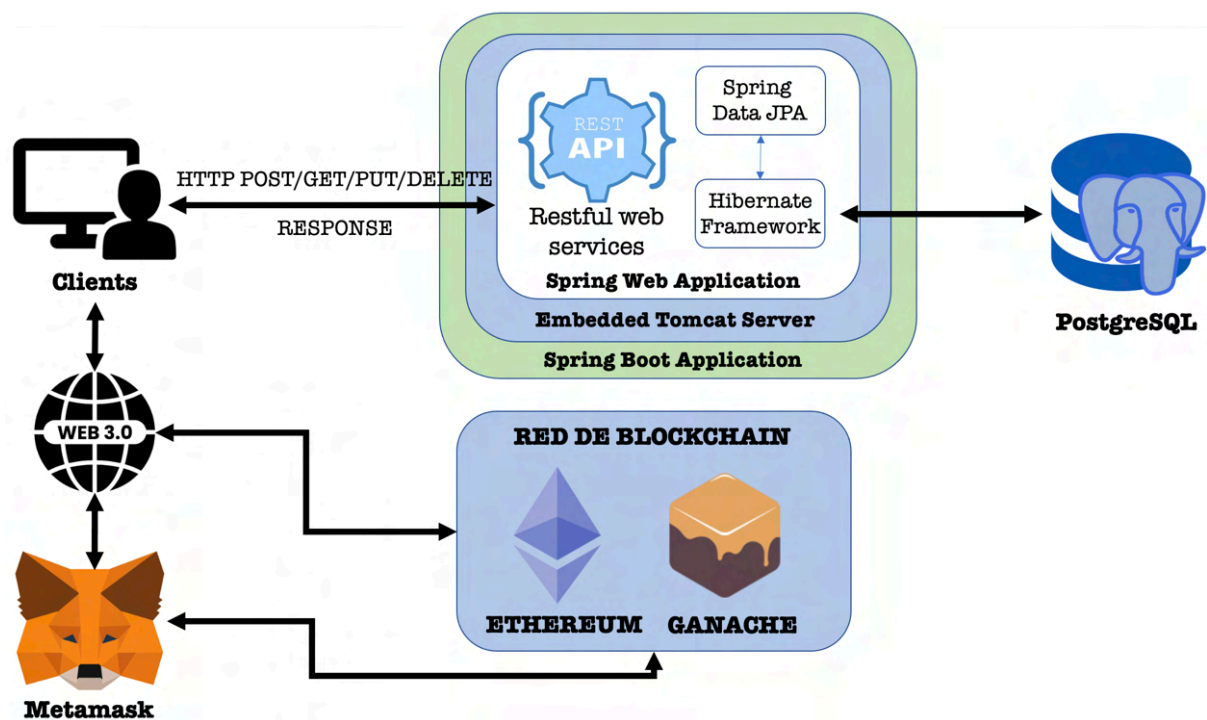


Ilustración 1 – Arquitectura del sistema

4. Resultados

Los resultados obtenidos cumplen con los objetivos propuestos al comienzo del desarrollo de este Trabajo Fin de Grado. La aplicación web cuenta con tres vistas, una primera vista principal que corresponde a la de los usuarios, y dos vistas secundarias correspondientes a la vista de una organización y vista de un administrador para gestionar la creación de proyectos y organizaciones, respectivamente.

Dentro de la vista de un usuario encontramos diferentes pantallas, entre las que destacan la página de Home, la visualización de donaciones y, la visualización de un proyecto con su acceso a donar.

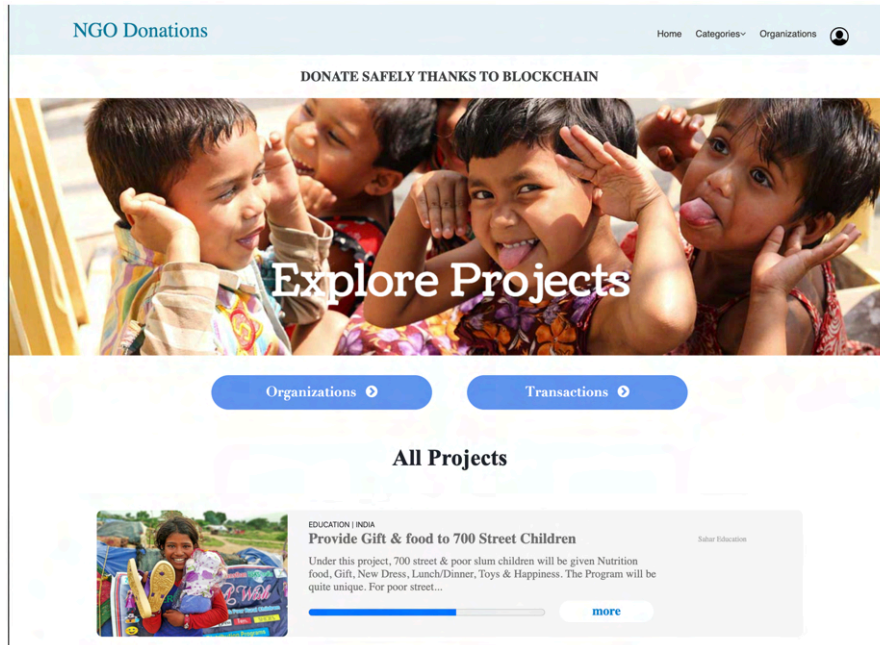


Ilustración 2. Pantalla Home

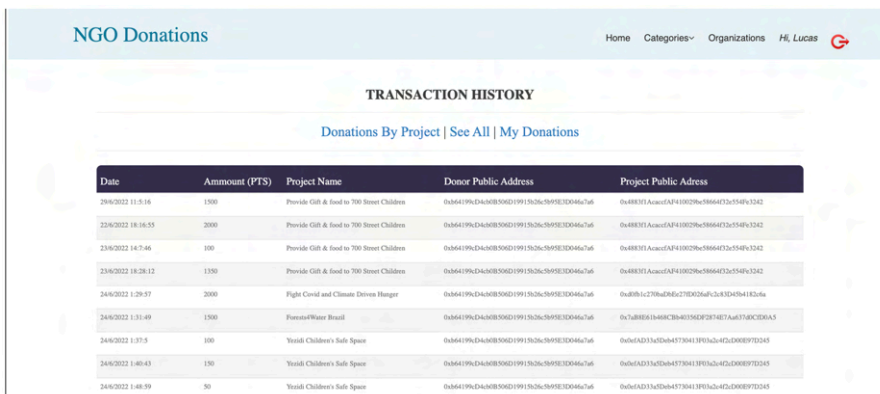


Ilustración 3 – Pantalla de donaciones

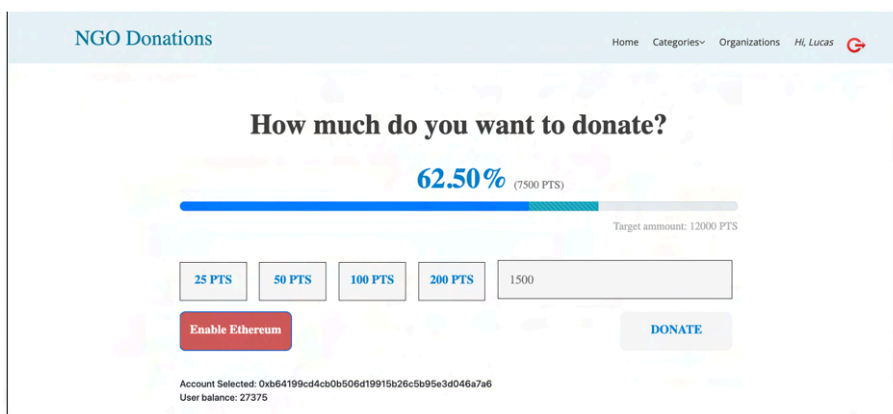


Ilustración 4. Pantalla realizar donación

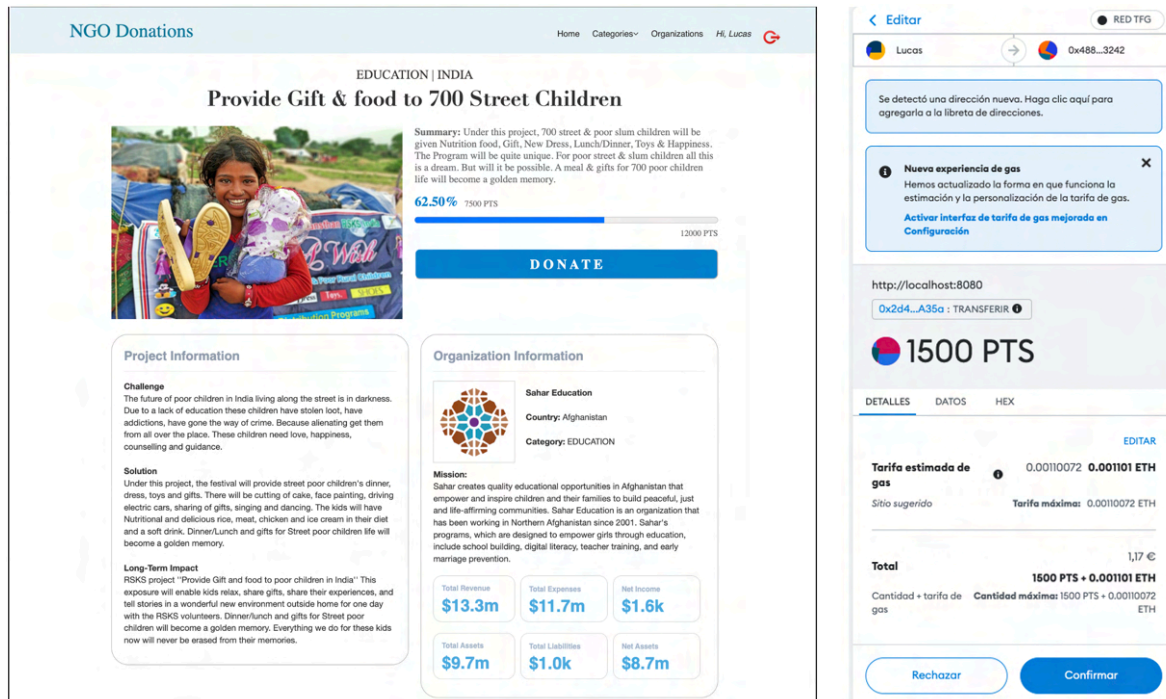


Ilustración 5. Pantalla info proyecto vs Pantalla confirmación de transacción en Metamask

5. Conclusiones

En este Proyecto Fin de Grado se ha conseguido llevar a cabo una primera aproximación de una aplicación web que gestione las donaciones de ONGs con la tecnología Blockchain.

El sistema que se ha desarrollado por medio del despliegue del Smart Contract de un token en la blockchain de Ethereum, ha permitido la realización de donaciones de una forma segura, fácil y transparente, siendo el frontend de la aplicación web el punto de interacción entre un usuario y el sistema.

Este trabajo ha conseguido demostrar que la tecnología blockchain está lista para ser implementada en el ámbito de las donaciones, además de mostrar las numerosas ventajas que esta tecnología aporta.

6. Referencias

- [1] *Claves para entender la tecnología 'blockchain'* (s.f.). Obtenido de BBVA: <https://www.bbva.com/es/claves-para-entender-la-tecnologia-blockchain/>
- [2] *Smart Contracts: ¿Qué son, cómo funcionan y qué aportan?* (s.f.). Obtenido de Bit2me: <https://academy.bit2me.com/que-son-los-smart-contracts/>

[3] RODRÍGUEZ, P. M. (12 de Abril de 2019). *INTRODUCCIÓN A BLOCKCHAIN: ¿QUÉ ES Y CÓMO HA EVOLUCIONADO?* Obtenido de Techedge:
<https://www.techedgegroup.com/es/blog/introduccion-blockchain-explicacion-evolucion>

NGO DONATION MANAGEMENT SYSTEM USING BLOCKCHAIN TECHNOLOGY

Author: Martínez Luna, Lucas Alberto.

Supervisor: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This Final Project design, develop and implements a web application based on blockchain technology for the management and handling of NGOs donations.

Blockchain, doesn't need third parties to perform transactions and as a result it increases the percentage of expenditure devoted to the missions of these organizations. It also offers a total reliability, security and transparency in donations.

For this, a web application will be developed that interacts with the contracts that are deployed on the blockchain network, Ethereum, through a backend that in addition to managing the database with the information of the projects of these organizations, will also manage all the frontend requests, as well as the communication between the deployed contracts and the graphical interface.

Keywords: Blockchain, Ethereum, Smart Contract, NGO, Web 3.

1. Introduction

Blockchain emerged in 2008 thanks to the first document published by Satoshi Nakamoto with the main objective of storing and transferring cryptocurrencies without the participation of banks.

International transfers, apart from requiring a complex technological infrastructure, also involve a high cost for the participating entities. Blockchain allows streamlining these transfers without losing security by offering full transparency and traceability of these transfers. This is possible thanks to the automation of payments, through the use of Smart Contracts.

The applications of blockchain are many, one of them is the management of donations from NGOs, an application that is intended to develop throughout this Final Project.

2. Project definition

This project designs a new model for the management of donations from non-profit organizations through the development of a web application that relies on blockchain technology through the use of Smart Contracts.

For this web application, a token will be developed through a Smart Contract, with which the different users of the system will be able to make their donations, and the donations will be also stored in the blockchain.

The implementation of this model should serve as an impulse for more people to make donations, knowing that they will have greater security, reliability and transparency.

The verification of the application's behavior will be carried out in Ganache, this framework offers a local network simulated in Ethereum to perform the relevant tests.

3. System description

The design of the application architecture depicted in Figure 1 consists of certain fundamental elements. First, we find the clients, who have access to the web application and to the resources stored in the blockchain and the database.

Secondly, there is Spring Boot Application, which has the Tomcat web servlet embedded. In addition, for the web development of this project I will use Spring Web which in turn makes use of Spring MVC, this is responsible for supporting all web requests (HTTP) to the API, and redirect them to the components that are necessary.

Spring Data JPA framework through a set of interfaces and APIs persist resources in a relational database. Hibernate facilitates through different Beans the mapping of attributes in databases, in addition to the object model of an application.

PostgreSQL has been chosen as the relational database manager. This manager will store the resources of users, projects and organizations.

Finally, we find the components related to blockchain technology. The first of these elements is Web3, which is responsible for establishing communication between the user and the Ethereum blockchain. Through this library we can directly access the data stored

in the blockchain, in this case, the donations, or we can access Metamask, in order to accept the transactions that we want to do.

Within the Ethereum blockchain network is the last component, Ganache, in which the simulation and testing of the Ethereum blockchain can be performed through a local test network. The compiled Smart Contracts are deployed in this framework.

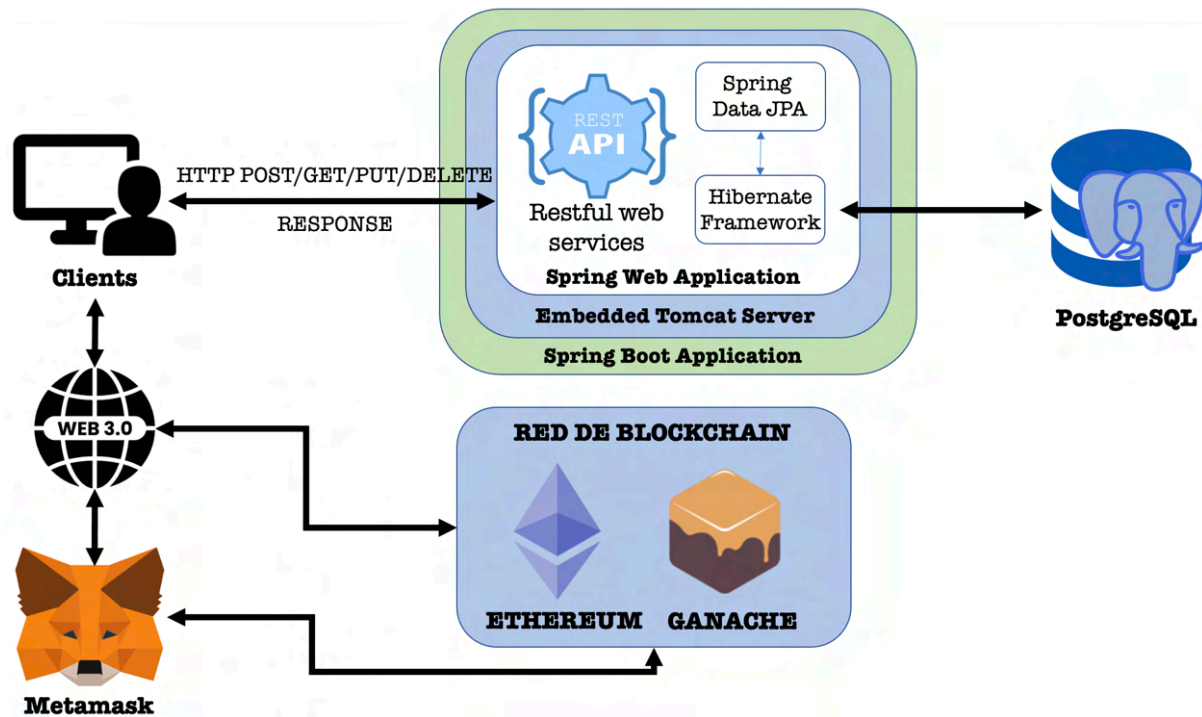


Figure 1. System architecture

4. Results

The results obtained meet the objectives proposed at the beginning of the development of this Final Project. The web application has three views, a first main view corresponding to the user view, and two secondary views corresponding to the view of an organization and an administrator to manage the creation of projects and organizations, respectively.

Within the user's view we find different screens, among which the Home page, the display of donations and the display of a project with access to donate stand out.

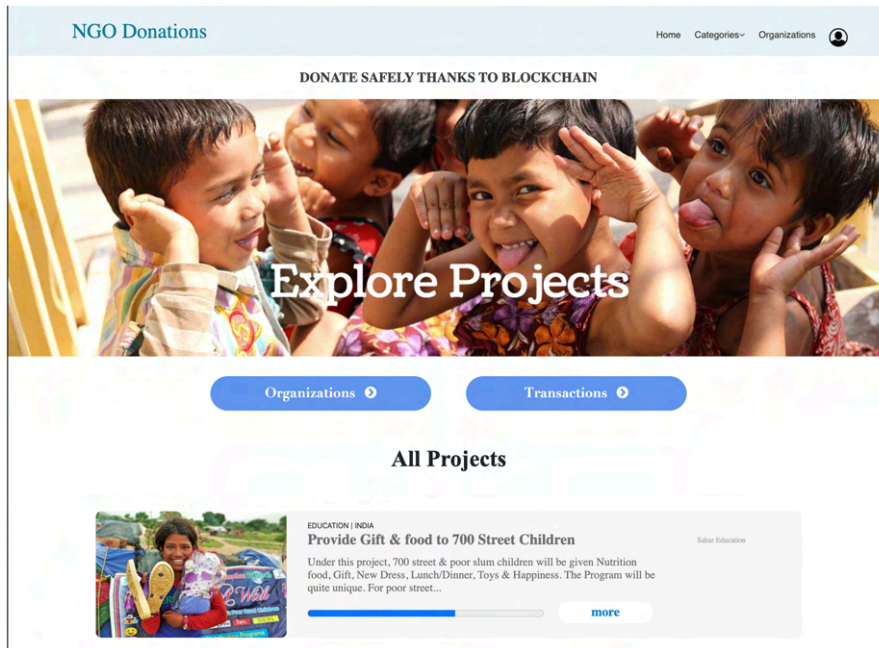


Figure 2. Home screen

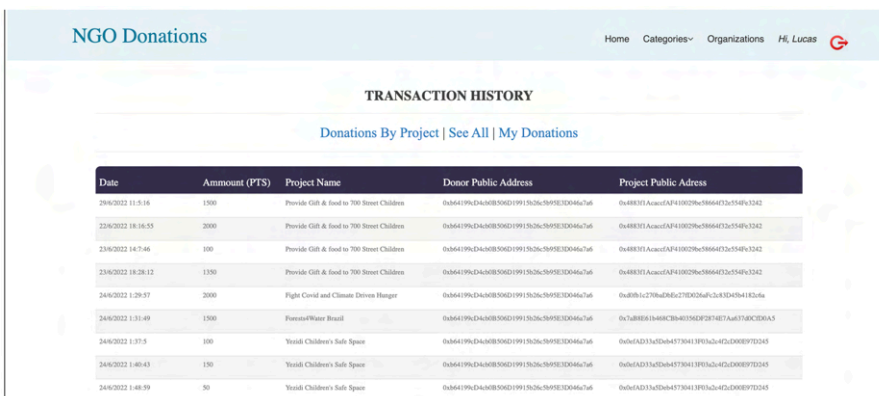


Figure 3. Donations screen

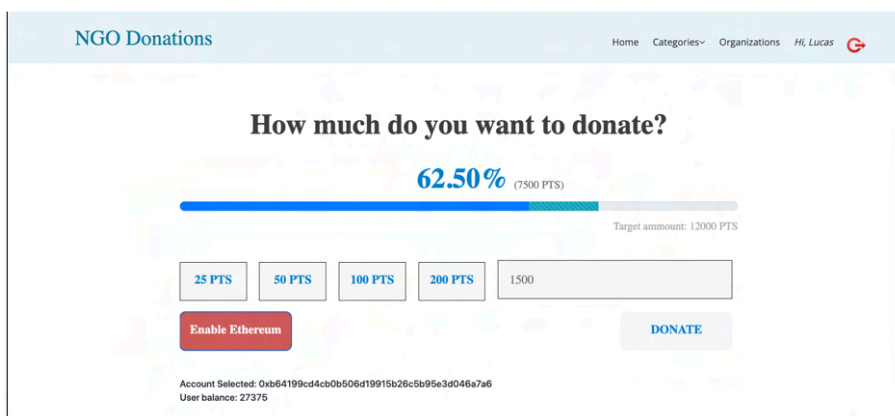


Figure 4. Make a donation screen

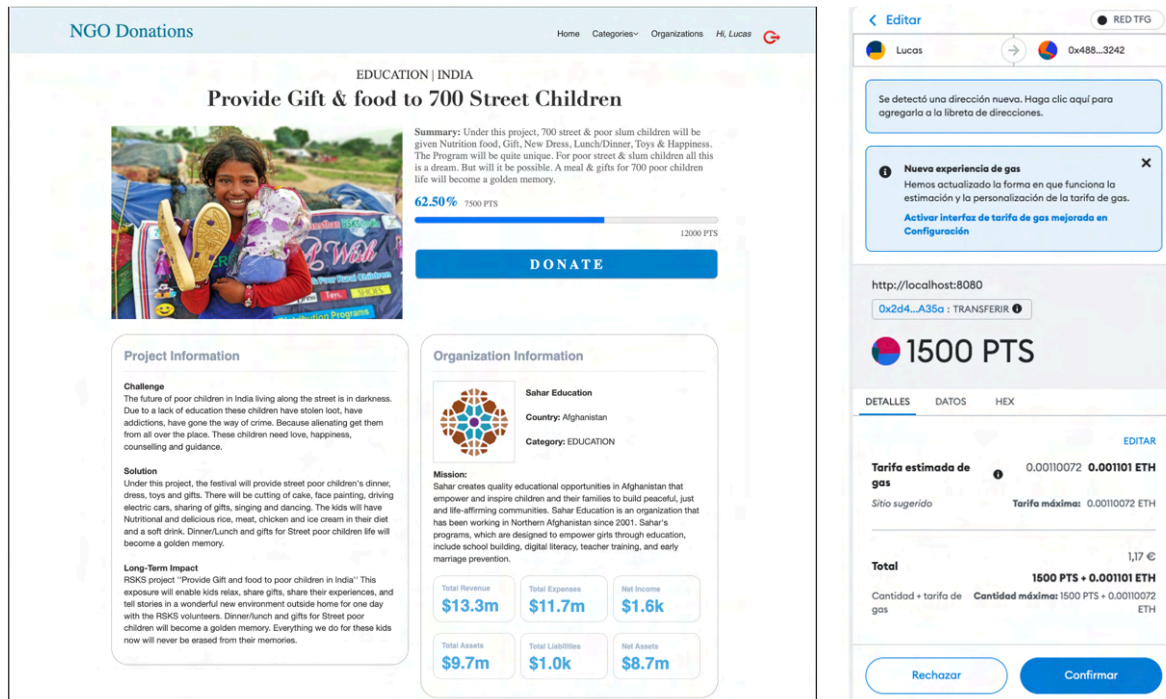


Figure 5. Project Info screen vs transaction confirmation in Metamask

5. Conclusions

In this Final Project I have managed to carry out a first approximation of a web application that manages the donations of NGOs with Blockchain technology.

The system that has been developed through the deployment of the Smart Contract of a token in the Ethereum blockchain, has allowed the realization of donations in a secure, easy and transparent way, being the frontend of the web application the point of interaction between a user and the system.

This work has managed to demonstrate that blockchain technology is ready to be implemented in the field of donations, in addition to showing the many advantages that this technology brings.

6. References

- [1] *Claves para entender la tecnología 'blockchain'* (s.f.). Obtenido de BBVA: <https://www.bbva.com/es/claves-para-entender-la-tecnologia-blockchain/>
- [2] *Smart Contracts: ¿Qué son, cómo funcionan y qué aportan?* (s.f.). Obtenido de Bit2me: <https://academy.bit2me.com/que-son-los-smart-contracts/>

[3] RODRÍGUEZ, P. M. (12 de Abril de 2019). *INTRODUCCIÓN A BLOCKCHAIN: ¿QUÉ ES Y CÓMO HA EVOLUCIONADO?* Obtenido de Techedge:
<https://www.techedgegroup.com/es/blog/introduccion-blockchain-explicacion-evolucion>

Índice de la memoria

CAPÍTULO 1.	INTRODUCCIÓN.....	8
CAPÍTULO 2.	DESCRIPCIÓN DE LAS TECNOLOGÍAS.....	11
2.1	BLOCKCHAIN	11
2.2	ETHEREUM	12
2.3	SMART CONTRACT	14
2.4	WEB3.JS.....	17
2.5	SPRINGBOOT	18
2.6	TECNOLOGÍA WEB	22
2.7	GANACHE.....	25
2.8	REMIX IDE	26
2.9	METAMASK	26
2.10	INTELLIJ	27
2.11	GIT	28
CAPÍTULO 3.	ESTADO DE LA CUESTIÓN	29
3.1	BLOCKCHAIN	29
3.2	ONG	44
3.3	PROYECTOS DE ONGS BASADOS EN BLOCKCHAIN	49
CAPÍTULO 4.	DEFINICIÓN DEL TRABAJO	50
4.1	JUSTIFICACIÓN.....	50
4.2	OBJETIVOS	51
4.3	METODOLOGÍA	51
4.4	PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA	53
CAPÍTULO 5.	SISTEMA DESARROLLADO	65
5.1	ANÁLISIS DEL SISTEMA	65
5.2	DISEÑO.....	68
5.3	IMPLEMENTACIÓN.....	80
CAPÍTULO 6.	ANÁLISIS DE RESULTADOS.....	87
CAPÍTULO 7.	CONCLUSIONES Y TRABAJOS FUTUROS	97

CAPÍTULO 8. BIBLIOGRAFÍA	99
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS.....	105
ANEXO II: MANUAL DE INSTALACIÓN	108
ANEXO II: MANUAL DE USUARIO.....	116

Índice de figuras

Figure 1. System architecture	17
Figure 2. Home screen	18
Figure 3. Donations screen	18
Figure 4. Make a donation screen	18
Figure 5. Project Info screen vs transaction confirmation in Metamask	19
Figura 1. Casos de uso de los Smart Contracts	15
Figura 2. Funcionamiento interno de Web3	17
Figura 3. Funcionamiento Web3	18
Figura 4. Módulos de Spring	19
Figura 5. Funcionamiento de Spring Web	20
Figura 6. Modelo REST API	21
Figura 7. Componentes de la Web	22
Figura 8. Funcionamiento de AJAX	25
Figura 9. Funcionamiento de Blockchain	30
Figura 10. Proceso de validación de firmas digitales	32
Figura 11. Proceso de minería blockchain	33
Figura 12. Tipos de redes existentes	33
Figura 13. Generaciones del blockchain	34
Figura 14. Diferencias entre los protocolos de consenso PoW y PoS	39
Figura 15. Diferencias entre blockchains públicas y privadas	41
Figura 16. Evolución mensual del Bitcoin con el euro de 2016 a 2021	43
Figura 17. Porcentaje de donantes en relación a la población	46
Figura 18. Gastos por ámbito de actividad de las ONGs	48
Figura 19. Ámbito de actuación de las ONGs	48
Figura 20. Los 12 principios del manifiesto Agile	52
Figura 21. Metodología Agile vs Waterfall	53
Figura 22. Panel Kanban con backlog del proyecto	53

Figura 23. Planificación del proyecto	54
Figura 24. Salario medio desarrollador de blockchain	56
Figura 25. Coste del despliegue de los Smart Contracts en Ether	57
Figura 26. Funcionamiento AWS Amplify.....	58
Figura 27. Costes de AWS Amplify	58
Figura 28. Especificaciones de las versiones Tiny Turtle y Happy Hippo de Elephant SQL .59	
Figura 29. Planes de marketing de SoloSEOySEM.....	60
Figura 30. Servicios de la aplicación	66
Figura 31. Petición POST	67
Figura 32. Funcionamiento patrón arquitectónico MVC.....	67
Figura 33. Capas de la aplicación	67
Figura 34. Arquitectura del sistema	69
Figura 35. Ejemplo endpoints API REST.....	70
Figura 36. Diagrama entidad-relación	71
Figura 37. Diagrama caso de uso – cliente	73
Figura 38. Diagrama caso de uso - organización y administrador.....	74
Figura 39. Diagrama de Clases	75
Figura 40. Diagrama de Secuencia - signup	76
Figura 41. Diagrama de Secuencia - donar.....	77
Figura 42. Diagrama de Secuencia - visualización de donaciones	78
Figura 43. Diagrama de navegabilidad – cliente	79
Figura 44. Diagrama de Navegabilidad – organización.....	80
Figura 45. Diagrama de Navegabilidad - administrador.....	80
Figura 46. Smart Contract del Token ERC-20 del proyecto.....	82
Figura 47. Cuenta de Ethereum de un cliente	82
Figura 48. Importación del token ERC-20 en Metamask	83
Figura 49. Correcto despliegue del contrato	87
Figura 50. Correcta ejecución del método transfer() del Smart Contract.....	88
Figura 51. Pantalla Home	89
Figura 52. Pantalla login.....	90

Figura 53. Pantalla organizaciones	90
Figura 54. Pantalla información organización	91
Figura 55. Pantalla información proyecto.....	92
Figura 56. Pantalla donación.....	93
Figura 57. Pantalla confirmar donación en Metamask	94
Figura 58. Pantalla de donaciones	94
Figura 59. Pantalla crear organización.....	95
Figura 60. Pantalla de administración de una organización	96
Figura 61. 8 Objetivos de Desarrollo del Milenio	107
Figura 62. 17 Objetivos de Desarrollo Sostenible	108

Índice de tablas

Tabla 1. Especificación del ordenador.....	55
Tabla 2. Especificación del monitor	55
Tabla 3. Costes software del proyecto	56
Tabla 4. Escenario pesimista de ingresos	62
Tabla 5. Escenario optimista de ingresos.....	63
Tabla 6. Escenario realista de ingresos	64

Índice de códigos

Código 1. Configuración de Web3 y funciones del contrato.....	83
Código 2. Acceso a los métodos del contrato	84
Código 3. Método para obtener el número de bloques generados.....	84
Código 4. Método 1 para obtener las transacciones de un bloque.....	84
Código 5. Método 2 para obtener las transacciones de un bloque.....	85
Código 6. Métodos para calcular la firma y el input de una transacción.....	86

Capítulo 1. INTRODUCCIÓN

Desde el surgimiento de la primera moneda, en Asia Menor alrededor del siglo VII a.C, hasta la sociedad actual, el dinero ha sufrido una importante evolución.

El dinero es un concepto antiguo, que siempre ha estado muy presente en la historia de la humanidad de una forma u otra. Antes de que existiera ningún tipo de moneda se practicaba lo que se denomina ‘trueque’, intercambio de bienes materiales o de servicios entre dos o mas personas por otros objetos o servicios.

La evolución del dinero ha estado siempre marcada por las necesidades de la sociedad. El dinero surgió como respuesta a las dificultades y limitaciones que presentaba el trueque, puesto que para que funcionara era necesario que cada una de las partes quisiera lo mismo que la otra parte ofertaba en un momento determinado.

Antes de establecerse la moneda, hubo otros productos como medio de pago, como el maíz o la sal (de donde proviene el salario). El problema que presentaban estos medios de pago, es la dificultad que presentaban para ahorrar, y es por ello que, como necesidad a este problema surge el oro y plata como medio de pago. Estos materiales preciosos, a pesar de ser efectivos supusieron posteriormente una dificultad a la sociedad presente en aquel momento, incomodidad para manejar grandes cantidades de dinero, y es por ello que surgieron las monedas y los billetes.

La aparición de los primeros bancos durante el Renacimiento, al comienzo del siglo XV, y la fundación del primer banco central, a mediados del siglo XVII, fueron acontecimientos de gran importancia puesto que permitieron controlar el sistema monetario, evitando que se produzcan altos niveles de desempleo o inflación, así como la regulación de la cantidad de dinero en circulación y crédito en la economía.

Las primeras transferencias bancarias fueron realizadas a finales del siglo XIX por la famosa empresa estadounidense, Western Union, a través de telegramas. Posteriormente la aparición

de Internet, a finales del siglo XX, y su rápido desarrollo, supuso importantes innovaciones en las prestaciones ofrecidas por las entidades de depósito.

Desde que los bancos comenzaron a ofrecer servicios de transferencia, estos han buscado siempre revolucionar el proceso, especialmente por que a medida que más gente empezaba a utilizar las transferencias bancarias incrementaba la necesidad de mejorar la rapidez de estas, y es por ello que, tras la aparición y evolución de internet, este se convirtió en una parte fundamental para los bancos, puesto que consiguió agilizar el proceso a través de la realización de transferencias online. Además, esto ayudó a reducir los costes de envío, así como a garantizar la seguridad.

Hoy en día, gracias a rápida evolución de internet, contamos con numerosas herramientas o aplicaciones que nos facilitan todos estos procesos como pueden ser, bizum, venmo, cashapp o probablemente la más famosa de todas, PayPal. Estas aplicaciones surgieron ante la necesidad de la sociedad por una mayor comodidad, rapidez y seguridad. Todo esto ha sido posible también gracias al desarrollo de las diferentes técnicas de verificación que hacen posible validar la identificación de los usuarios.

El efecto del cambio tecnológico en el desempeño financiero demuestra que la incorporación de innovaciones tecnológicas es un componente esencial de las estrategias que los bancos deben seguir para obtener una ventaja competitiva.

Blockchain surge en 2008 gracias al primer documento publicado por Satoshi Nakamoto con el principal objetivo de almacenar y transferir criptomonedas sin la participación de los bancos, sin embargo, algunos de los mayores propulsores de esta tecnología han sido precisamente estos.

La principal aplicación de blockchain para los bancos es, en las transferencias internacionales, puesto que estas, aparte de requerir una compleja infraestructura tecnológica, también suponen un coste elevado para las entidades participantes. Además, blockchain permite agilizar estas transferencias sin perder seguridad.

Una de las mayores ventajas que ofrece blockchain es la automatización de los pagos, por medio del uso de los Smart Contracts, además de ofrecer una total transparencia y trazabilidad de las transacciones.

Las aplicaciones del blockchain son muchas, una de ellas es la gestión de donaciones de ONGs, aplicación que se pretende realizar y desarrollar a lo largo de este Trabajo Fin de Grado.

La credibilidad de las ONG ha disminuido durante el transcurso de los últimos años. Tal y como se describe en un estudio¹ realizado en 2016 por el grupo de INDESOL obtuvo dos datos reveladores, el primero de ellos es que solo el 52% de las ONGs no transparenta de manera pública sus fuentes de financiamiento y el manejo de sus recursos, el segundo de ellos es que el 36% de las ONGs no cuentan con estados financieros. Además, entre el año 2014 y 2017 la credibilidad de estas disminuyó del 66% al 53%.

Estos datos hacen que el blockchain sea especialmente interesante implementarse en las ONGs. Gracias a la blockchain se podrán realizar donaciones de cualquier valor y desde cualquier parte del mundo, con la confianza de que estas van a alcanzar su destino final, debido a la transparencia que ofrece esta tecnología para la ejecución y el seguimiento de las transacciones. Además, gracias al uso de los Smart Contracts que se van a desarrollar a lo largo del proyecto, se reducirán los costes al no existir la necesidad de terceros.

Por todo esto, el principal objetivo que persigue este proyecto es la creación de una aplicación web para la gestión de donaciones de ONGs, mediante el uso de Smart Contracts. Con el uso del concepto de Blockchain se pretende resolver problemas de seguridad y transparencias de las donaciones, así como mejorar el sistema y agilidad de estas.

1 [10] (Cota, ¿Confiar o no confiar en las ONG? He ahí el dilema, 2019)

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

A continuación, se muestran las herramientas que en un principio se van a emplear a lo largo del proyecto.

2.1 *BLOCKCHAIN*

Blockchain puede definirse como un concepto nuevo y revolucionario que pretende garantizar la seguridad, transparencia y privacidad de los usuarios, pero a pesar de ello, blockchain hace uso de una tecnología básica como son las conexiones P2P. En las conexiones P2P, que aparecieron por primera vez en 1996, cada uno de los nodos puede actuar tanto como cliente como servidor. Además de esto blockchain también incorporó un método de encriptación para garantizar la confidencialidad de los datos, así como funciones hash que garantizan la unicidad e integridad de los datos.



Blockchain se basa en una red descentralizada donde los datos se encuentran almacenados en cada uno de los nodos (ordenadores) que conforman la blockchain, solucionando así problemas de alteración y falsificación de datos. El análogo de la blockchain podría ser un libro de cuentas público con copias distribuidas en los diferentes nodos, por lo que, en caso de haber una modificación maliciosa, esta debería realizarse en cada uno de los nodos, lo que resulta prácticamente imposible.

Las transacciones de la red blockchain han de pasar en primer lugar por un proceso de validación, donde la mayoría de los nodos han de confirmar la validez de esta, para poder ser asignadas a bloques que serán inmutables. Estos bloques contienen un hash único.

2.2 ETHEREUM

Ethereum es una blockchain pública de código abierto, por lo cual sus datos, software y desarrollo se encuentran de forma pública y pueden ser usados por cualquier usuario.

Ethereum es además una plataforma programable porque permite a los desarrolladores utilizarla para la creación de sus propias aplicaciones descentralizadas (Dapp).



Esta plataforma fue lanzada en 2015 por Vitalik Buterin, su criptomoneda nativa es el Ether (ETH), y este token puede ser usado en cualquier transacción que use este software.

El valor de esta criptomoneda, Ether, es volátil, en 2016 el Ether cotizaba entorno a 1\$, mientras que a principios de 2018 su valor se encontraba cerca de los 1400\$², su valor actual se encuentra en 1230\$³. A pesar de existir cientos de criptomonedas, esta es una de las pocas que tiene una capitalización significativa junto con Bitcoin.



² [43] (¿Qué es Ethereum y cómo funciona? , s.f.)

³ [20] Dólar (Ethereum en Dólar estadounidense, 2022)

Bitcoin y Ethereum a pesar de presentar muchas similitudes también presentan varias diferencias. Bitcoin y Ethereum utilizan el modelo de “PoW” (Proof of work), el algoritmo de HashCash, pero utilizan distintas funciones hash, Bitcoin utiliza la función SHA-256, mientras que Ethereum utiliza SHA3-256. Mientras que Bitcoin fue la primera criptomoneda revolucionando el sistema financiero, Ethereum intentó ir más allá aprovechando esta tecnología para aplicarla a aplicaciones y contratos financieros descentralizados. La blockchain de Bitcoin solo permite y es usada como medio de intercambio para enviar y recibir transacciones, mientras Ethereum tiene la gran ventaja que permite la implementación de Smart Contracts y la creación de Dapps. Estos programas o aplicaciones que se pueden desarrollar en la red de Ethereum necesitan de su criptomoneda nativa, Ether. Ethereum es también la principal red utilizada para la creación de tokens, los cuales utilizan principalmente el estándar ERC-20.

La red de Bitcoin genera aproximadamente un bloque cada 10 minutos, mientras que Ethereum cada 10-20 segundos. La escalabilidad de Bitcoin se encuentra limitada a unas 7-8 transacciones por segundo mientras que Ethereum llega hasta 16-20 transacciones por segundo.⁴

Ethereum tiene programado para el 8 de junio de 2022 pasar a ETH 2.0, esta actualización dará a lugar diversos cambios. En primer lugar y el principal cambio que va a sufrir Ethereum es un cambio en el consenso que ha estado utilizando. Hasta antes de la actualización Ethereum ha utilizado el consenso PoW (Proof of work), sin embargo, tras el 8 de junio de 2022 se actualizará al consenso PoS (Proof of Stake). Ethereum también duplicará las recompensas de su programa de recompensas por errores.

La actualización a ETH 2.0 tiene como gran objetivo reducir la cantidad de energía consumida por las cadenas de bloques, así como reducir la emisión de ETH en un 90%⁵, además de ofrecer una mayor seguridad y escalabilidad. Con el uso del consenso de PoW era necesario el uso de minería, lo que implica ordenadores de alta gama y un gran consumo de energía para poder

4 [18] (Diferencias entre Bitcoin y Ethereum , s.f.)

5 [21] (Ethereum se acerca a su actualización ETH 2.0 a PoS en junio , 2022)

resolver operaciones matemáticas complejas. Con esta actualización y nuevo uso de consenso la minería será reemplazada por validadores, quienes prometen Ethers para validar y asegurar la red, y son recompensados posteriormente por ello.

2.3 SMART CONTRACT

Cuando se desea realizar un pago de forma física el proceso es sencillo y seguro, pero esto se complica cuando lo que se desea es realizarlo de manera electrónica. Al realizar pagos online surgen diferentes preguntas tal que, ¿Cómo puedo asegurarme que tienes el dinero que dices tener? o ¿Cómo puedo saber que el dinero que vas a transferir pasa efectivamente de tus manos a las mías? Es por ello que hemos necesitado confiar en diversas entidades. En primer lugar, en los bancos, quienes certifican el dinero de las personas y ejecuten los movimientos de estas, y en segundo lugar los gobiernos, quienes controlan y regulan a los bancos. Por ello la aparición de los Smart Contracts ha sido tan importante, puesto que nos permite disponer de un sistema donde no existe la necesidad de ninguna entidad para certificar y regular las transacciones, y que estas sean irreversibles, es decir, que nadie pueda alterarlas ni deshacerlas. Los Smart Contracts permiten la automatización de los pagos de un modo seguro.

Las principales plataformas para el desarrollo y ejecución de los Smart Contracts son las siguientes⁶:

- Ethereum: los Smart Contracts son desarrollados principalmente en Solidity y ejecutados en la EVM.
- Hyperledger: sistemas open source llevado a cabo por Linux Foundation, caracterizada por ser una plataforma flexible para el desarrollo de los Smart Contracts.
- Counterparty: Utiliza la blockchain de Bitcoin y permite el despliegue de contratos inteligentes sobre esta.

6 [62] ('Smart contracts': contratos inteligentes para formalizar acuerdos en la era digital , s.f.)

- Polkadot: esta plataforma es famosa por la capacidad de albergar parachains, blockchains de terceros que se añaden a la principal, para realizar más transacciones de lo habitual.

Un Smart Contract es simplemente un trozo de código, donde se establecen las condiciones que han de cumplir las transacciones para que estas puedan ser validas. Esto es una gran ventaja, puesto que evita la necesidad de terceros, elimina la limitación entre fronteras y reduce costes y tiempo de constitución.

En los Smart Contracts definen un conjunto de reglas para su posterior compilación. Una vez compilados se procede al despliegue de estos sobre la blockchain, donde se convierten en inmutables y se quedan almacenados en la blockchain desplegada. El despliegue de estos además conlleva un coste, así como cada vez que se ejecute, incluso si no puede llegar a completarse porque no satisface el conjunto de reglas. El coste que conlleva cada contrato es variable y estable directamente relacionado con la cantidad de recursos que requiere el contrato, por ellos es necesario desarrollar el código con la mayor simplicidad posible y con la menor cantidad posible de funciones.

El uso de los Smart Contracts convierte a las transacciones en transparentes, irreversibles y rastreables. Estos tienen innumerables casos de uso, algunos de estos se ilustran en la Figura 6.



Figura 6. Casos de uso de los Smart Contracts

Las características más destacadas de los Smart Contracts son:

- Autoejecutables
- Inmutabilidad
- Verificables y seguros
- Reducción de costes
- Rapidez
- Autonomía

2.3.1 SOLIDITY

El lenguaje que se va a emplear en este Trabajo Fin de Grado es Solidity. Este lenguaje es el más famoso y utilizado para la creación de los Smart Contracts.

Solidity junto con Vyper son los lenguajes más activos soportados por la blockchain de Ethereum. Para los desarrolladores más avanzados existe otro lenguaje, Yul, soportado por la blockchain de Ethereum.

Solidity es un lenguaje de programación influenciado por los siguientes lenguajes, C++, Python y JavaScript y su ejecución se da a lugar en la Ethereum Virtual Machine (EVM).



Solidity utiliza un lenguaje de tipado estático, lo que significa que la comprobación del tipo de variable se da durante la compilación, y no durante la ejecución.

Cuando se desarrolla un contrato con este lenguaje es necesario especificar la versión de compilación.

Este lenguaje es compatible con⁷:

- Herencia: es posible heredar de otros contratos.
- Uso de bibliotecas: es posible la creación de funciones estáticas en otros lenguajes de programación orientados a objetos.
- Posibilidad de la creación y uso de tipos complejos de variables.

Por otro lado, Vyper es un lenguaje mucho mas restrictivo pues busca que sus contratos sean mas seguros y fáciles de auditar.

2.4 WEB3.JS

Una vez se han definido los conceptos de Blockchain, Ethereum y Smart Contract, la siguiente pregunta sería como se puede conectar a esta red para poder obtener y cambiar datos. La respuesta y solución a esta pregunta es conectándonos a un nodo (ordenador) que esté conectado a la red. No es necesario que el nodo al que nos conectemos sea de nuestra propiedad, puesto que puede ser ajeno, lo único que se necesitará será la URL que lo identifica en Internet.

Tal y como se ilustra en la Figura 7, para conectarse al nodo localizado y deseado es necesario hacerlo a través de un JSON-RPC, protocolo de llamada a procedimiento remoto codificado en JSON. En dicho JSON se pasan las variables de procedimiento, parámetros para él mismo y el ID del request, una vez finalizada la ejecución en el nodo, este nos mandará la respuesta.⁸

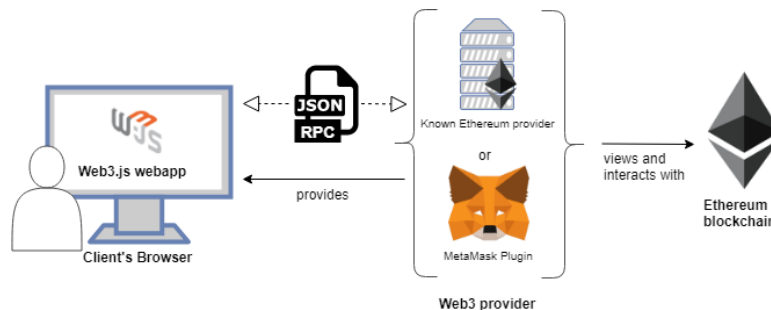


Figura 7. Funcionamiento interno de Web3

7 [30] (Joshua, 2022)

8 [33] (Lasa, 2019)

Debido a la incomodidad y dificultad que puede suponer todo el proceso descrito con anterioridad surgió Web3js.

Web3 es una colección de bibliotecas de JavaScript que permite interactuar con un nodo de Ethereum local o remoto por medio de HTTP, IPC o WebSocket. Web3 puede considerarse como un puente entre las Dapps y la blockchain.

Para poder conectarse a un nodo, en primer lugar, es necesario importar web3 por medio a través de un ‘require’. Posteriormente, se establece la variable del proveedor. En tercer lugar, se inicializa la variable web3Provider para obtener los servicios de Ethereum Networks, y finalmente el objeto con el que vamos a obtener acceso a las funciones de web3. ⁹

Con la Figura 8 se pretende ilustrar el funcionamiento de web3.

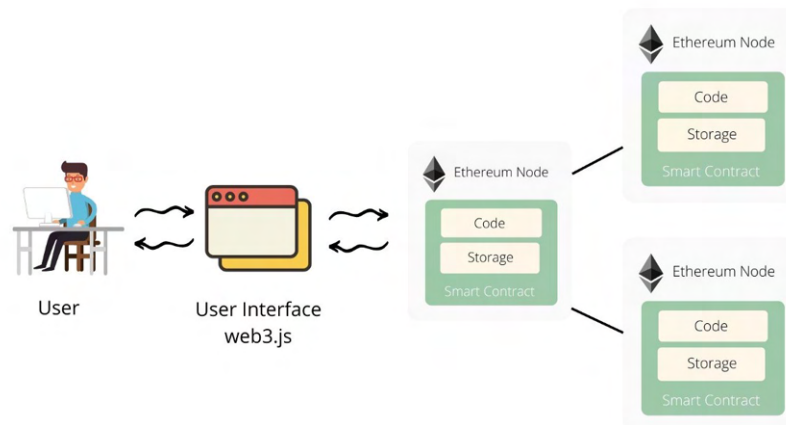


Figura 8. Funcionamiento Web3

2.5 SPRINGBOOT

El framework de SpringBoot es un módulo gratuito del proyecto de Spring, y es el ecosistema de desarrollo web backend Java mas utilizado en el mundo.

La infraestructura de SpringBoot pretende eliminar la mayor parte del trabajo de configuración de aplicaciones basadas en Spring, proporcionando un conjunto de herramientas que agilizan y facilitan la construcción de estas.



A continuación, en la Figura 9, se muestran los diferentes módulos de Spring, cuyo módulo principal es el Core Container, puesto que es el encargado de la creación y gestión de los Beans, así como del manejo de propiedades y dependencias.¹⁰

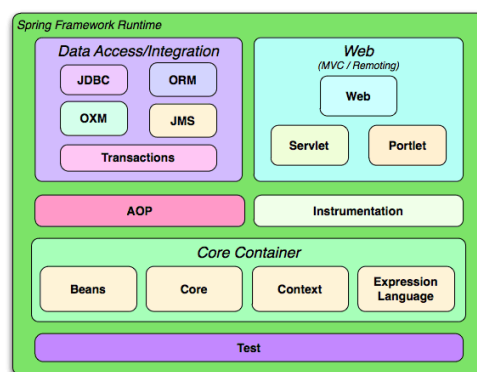


Figura 9. Módulos de Spring

SpringBoot trabaja sobre la JVM, lo que permite desarrollar, migrar o reutilizar código de cualquier proyecto, así como ser ejecutado en los principales sistemas operativos.

Actualmente existe una tendencia al desarrollo de aplicaciones sobre una arquitectura de microservicios. Esta arquitectura es destacada por la flexibilidad que ofrece al separar las responsabilidades en diferentes microservicios consiguiendo así aplicaciones especializadas y fáciles de reutilizar. Además, posteriormente se pueden migrar entre los diferentes modelos de negocios.

10 [46] (Pahino, 2020)

SpringBoot aparte de la versatilidad que ofrece también puede usarse para el desarrollo de aplicaciones web tradicionales, donde el HTML antes de ser entregado al navegador web es generado dinámicamente en el lado del servidor.

Algunas de las mayores ventajas y funcionalidades que nos ofrece SpringBoot son:

- Gestor de configuraciones.
- Seguridad con OAuth2.
- Capa acceso a bases de datos.
- Creación de APIs.
- Inyección de dependencias.
- Testing y monitorización.

SpringBoot provee una serie de contenedores web servlet para poder desplegar la aplicación automáticamente solo con correr el main de la aplicación. Dentro de los contenedores web podemos escoger entre Tomcat, Undertow o Jetty, estos vienen embebidos como dependencias. SpringBoot tiene embebido Apache Tomcat. Además, para el desarrollo web de este proyecto se ha hecho uso de Spring Web quien a su vez hace uso de Spring MVC, este se encarga de soportar todas las peticiones web (HTTP) y redirigirlas a los componentes que sean necesario, tal y como se ilustra en la Figura 10. Además, Spring MVC es considerada la columna vertebral de los servicios web RESTful.

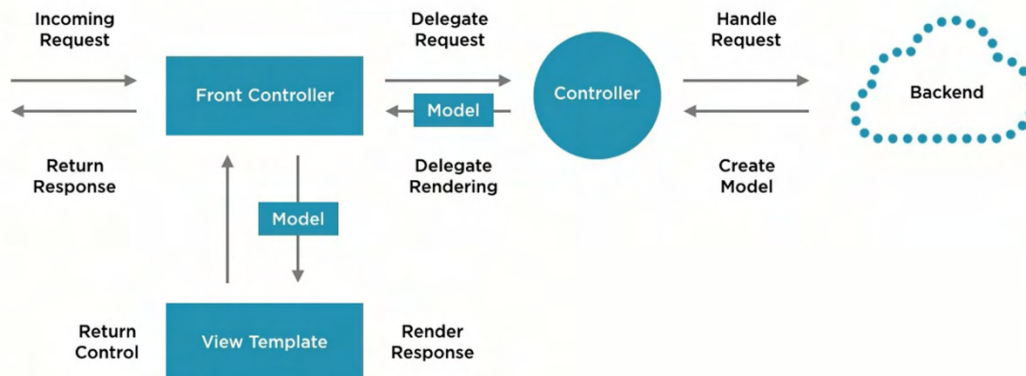


Figura 10. Funcionamiento de Spring Web

SpringBoot además permite elegir entre Maven o Gradle como gestores de dependencias. Como gestor de dependencias para este proyecto se ha hecho uso de Maven. Otras características importantes que presenta Maven son:

- Multiplataforma.
- Creación de plugins modificables, lo que resulta en mecanismo para ser extensible.
- Compatibilidad con multitud de IDEs.
- Facilidad para reusar códigos y librerías.

2.5.1 API

SpringBoot impulsa la creación de nuestras propias API para poder desplegar así nuestros servicios REST, de tal forma que los clientes que lo consuman puedan conectarse por medio del protocolo HTTP. Las API tienen como principal objetivo el intercambio de datos entre diferentes tipos de software para automatizar procedimientos, así como desarrollar nuevas funcionalidades.

Las API pueden desarrollarse en diferentes lenguajes de programación y además es fundamental el desarrollo de una documentación clara para su posterior implementación.

Algunos de los formatos utilizados para el intercambio de información entre sistemas son XML, YAML y JSON.

Se ha utilizado REST como patrón de diseño de la API, donde se maneja la transferencia de documentos a través de una serie de características específicas de la interfaz de usuario que se han de cumplir para que puedan ser reconocidos por los servidores.

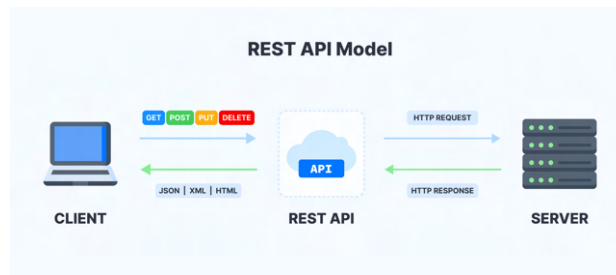


Figura 11. Modelo REST API

Existen además otros protocolos entre los cuales se encuentran, XML-RPC, JSON-RPC y SOAP.

Actualmente, existen cuatro categorías para las API:

- API pública: conocidas como API abiertas, suelen estar totalmente disponibles y accesibles para su uso.
- API privada: disponible únicamente para los sistemas internos de una organización y oculta para cualquier usuario externo.
- APIs de socios: disponible para miembros de una alianza comercial, cuando no se encuentra disponible es necesario solicitar una autorización especial.
- API compuesta: posibilidad para soportar varios sistemas de integración y combinar todos los datos.

2.6 TECNOLOGÍA WEB

A continuación, se van a describir algunas de las tecnologías empleadas para el desarrollo web de la aplicación.

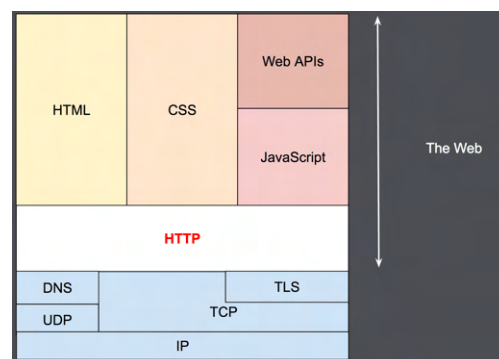


Figura 12. Componentes de la Web

2.6.1 HTTP

Hypertext Transfer Protocol mayormente conocido como HTTP, es un protocolo cliente-servidor y es la base de los intercambios de datos en la Web. HTTP permite la realización de peticiones de datos y recursos por parte de los clientes.

El protocolo HTTP ha ido evolucionando con el tiempo y es un protocolo de capa de aplicación que transmite sobre el protocolo TCP. Se usa para el envío de elementos de hipertexto como HTML e hipermedia.

2.6.2 HTML

Para el desarrollo de aplicación web de este proyecto también se ha hecho uso de HTML, lenguaje empleado para el desarrollo de páginas web. La programación en este lenguaje es por medio del uso de etiquetas que posteriormente el navegador interpreta. HTML es usado para la creación y estructura de una página web, mientras que el diseño debe llevarse a cabo con CSS.

2.6.3 CSS

CSS es un lenguaje que tiene como objetivo el diseño y presentación de las páginas web. Este es desarrollado junto con HTML. Gracias a CSS puedes crear un conjunto de instrucciones para decidir como quieres mostrar la información. Debido a su tediosa programación, se han desarrollado diferentes frameworks, entre los que destaca Bootstrap, que proporcionan una mayor comodidad para el diseño de aplicaciones web.

2.6.4 POSTGRESQL

Las bases de datos relacionales son una colección de datos que se organizan en un conjunto de tablas, donde estos datos pueden ser accedidos o modificados de diferentes maneras sin necesidad de reorganizar las tablas. SQL es el lenguaje utilizado para la creación y acceso de las bases de datos relacionales. Por el contrario de las bases relacionales, las no relacionales no poseen un identificador para relacionar dos o más conjuntos de datos.

Para este Trabajo Fin de Grado se ha decidido utilizar la base de datos relacional PostgreSQL. Para el manejo de esta base de datos se ha hecho uso de la interfaz gráfica pgAdmin4. Esta interfaz además de ser gratuita es también fácil de utilizar y compatible con todos los sistemas operativos.



Postgres tiene como mayor competidor a MySQL, sin embargo, una de las ventajas que ofrece Postgres con respecto a MySQL es que la primera soporta queries de mayor complejidad, además de tener mayor rendimiento con grandes volúmenes de datos. Por otro lado, mientras que MySQL no puede garantizar al 100% el cumplimiento de ACID (atomicidad, consistencia, aislamiento y durabilidad), Postgres si que lo cumple en su totalidad.

Además, para este proyecto se ha decidido también hacer uso de ElephantSQL. Esta es un servicio hosting para almacenar la base de datos desarrollada. Además, se encarga de las tareas administrativas como la instalación o actualización de las versiones, así como del manejo de las copias de seguridad.

2.6.5 JAVASCRIPT

Javascript es un lenguaje de programación que tiene como objetivo principal la interacción dinámica de los sitios web, por ello no hace uso de ningún compilador. Javascript permite responder a eventos causados por los usuarios y realizar modificaciones en el DOM. El código javascript se ejecuta en los navegadores. Las funcionalidades de este lenguaje son muchas, desde detectar errores en los formularios hasta hacer modificaciones de elementos de una página web.

2.6.5.1 AJAX

Esta tecnología fue desarrollada en javascript y ha supuesto uno de los mayores avances en el desarrollo web. AJAX permite el intercambio de información con el servidor sin necesidad de recargar la página.

Esta tecnología permite que las aplicaciones funcionen de forma asíncrona, procesando las request en segundo plano. A continuación, en la Figura 13 se muestra el funcionamiento de esta.

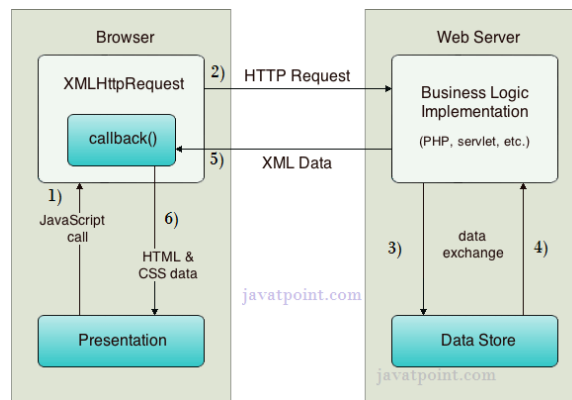


Figura 13. Funcionamiento de AJAX

2.7 GANACHE

El framework Ganache es un software desarrollado para la simulación y testeo de la blockchain de Ethereum a través de una red de pruebas local sencilla y fácil de usar. Esta esta disponible para su descarga en la página oficial de truffle.



Ganache ofrece la posibilidad de utilizar un quickstart workspace con 10 cuentas, cada una de ellas con su clave pública y privada y su dirección de Ethereum. Cada una de estas cuentas cuenta con un saldo inicial de 100 ETH. Una vez iniciado este modo se puede ver la dirección del servidor RPC, así como las transacciones y bloques almacenados en la blockchain.

Por otro lado, Ganache ofrece también la posibilidad de crear un workspace, opción que se ha decidido adoptar en este Trabajo Fin de Grado. Este modo además de ofrecer configuraciones personalizadas, permite guardar las configuraciones y despliegues de contratos que se hayan realizado. Algunas de las configuraciones más importantes son: hostname, número de puerto, network id, número de cuentas, saldo inicial de las cuentas, límite o precio del gas.

2.8 REMIX IDE

Remix es un IDE basada en una aplicación web y de código abierto, cuya funcionalidad principal es el desarrollo de los Smart Contracts de Ethereum. Remix trae consigo integrado un compilador y entorno en tiempo de ejecución para Solidity sin componentes orientados al servidor.



El IDE de Remix está pensado para el lenguaje de programación Solidity y, su uso es admitido tanto en navegador como en el ordenador para su ejecución en local.

Las principales ventajas que ofrece Remix:

- Entorno fácil y sencillo para el desarrollo de Smart Contracts en Solidity.
- Permite la selección de versión del compilador de los Smart Contracts.
- Facilita el despliegue de los contratos sobre la blockchain.
- Permite conectividad con Metamask.
- Fácil accesibilidad para el ABI y address del contrato.
- Fácil comprobación del correcto funcionamiento de las funciones creadas en el contrato.

Otros IDEs existentes son EthFiddle (basado en web), VSC (escritorio) y Atom (escritorio).

2.9 METAMASK

Metamask es una extensión o plugin gratuito de Google Chrome que permite y facilita la interacción de los usuarios con las Dapps de Ethereum. Metamask además de hacer las funciones de wallet para Ethereum también lo hace de los tokens ERC-20, además la importación de estos en la billetera es extremadamente sencilla.



La billetera virtual para criptomonedas Metamask hace uso de la librería web3js, cuya finalidad como se ha comentado con anterioridad es la de la creación de aplicaciones web con capacidad de interactuar con la blockchain de Ethereum. Metamask ofrece también la posibilidad de conectarse a diferentes testnet de Ethereum, así como a otras blockchain.

2.10 INTELLIJ

Aunque existen multitud de IDEs para el desarrollo de aplicaciones, en este Trabajo Fin de Grado se ha decidido utilizar IntelliJ. Esta plataforma creada por Jet Brains es un IDE inteligente para el desarrollo Java.



Actualmente existen dos versiones, Community Edition (open source) y Ultimate (comercial). IntelliJ soporta además una gran cantidad de tecnologías como, Groovy, Javascript, Spring, Hibernate, Android... Son innumerables las tecnologías que soporta. Algunas de las ventajas que ofrece este IDE son:

- Integración con sistemas de control de versiones.
- Autocompletado de código.
- Refactorización inteligente.
- Fiabilidad y robustez.
- Automatización de la compilación al ejecutar el código.

2.11 GIT

Git es un software diseñado para el control de versiones. Git tiene como objetivo mejorar el mantenimiento de versiones de aplicaciones, teniendo en cuenta la eficiencia, la confiabilidad y compatibilidad de estas cuando existe un gran número de archivos de código fuente. Sin embargo, un inconveniente que Git presenta es que, al estar instalado en local, no puede leer ediciones por otros miembros de un equipo en tiempo real.



2.11.1 GITHUB

En el Trabajo Fin de Grado se va a utilizar la plataforma de Github para almacenar el código desarrollado, ya que ofrece una mayor comodidad respecto a git, al no necesitar una interfaz de línea de comandos.



Capítulo 3. ESTADO DE LA CUESTIÓN

3.1 BLOCKCHAIN

Blockchain es una tecnología que está cambiando el mundo revolucionando la forma en que se hacen los negocios. Pero, ¿qué es exactamente blockchain y cómo funciona?

En términos simples, blockchain puede verse como un libro de contabilidad digital o un sistema de mantenimiento de registros distribuido que mantiene un registro seguro de transacciones digitales y la información relacionada con estas.

Blockchain surge por primera vez en 2008, dentro del proyecto Bitcoin. El objetivo de este proyecto era poder garantizar la seguridad, transparencias y privacidad de los usuarios ante empresas que controlan la industria de Internet por medio de servidores de almacenamiento y gestión de los datos particulares de clientes.

Blockchain surgió gracias al desarrollo de la criptografía avanzada, la cual se combinó con una de las tecnologías mas simples existentes, las redes P2P.

Blockchain es considerada una base de datos transaccional, puesto que se almacenan en ella un historial de todas las transacciones en lugar de su contenido como se hace actualmente. Todos los nodos u ordenadores conectados a la blockchain tienen acceso a este registro de transacciones.

Como se ha comentado con anterioridad blockchain facilita el proceso de registro de transacciones, pero también facilita el seguimiento de activos en una red. Un activo puede ser tangible (una casa, coche, dinero efectivo, terreno) o intangible (patentes, derechos de autor, marcas). Prácticamente cualquier activo puede ser rastreado y comercializado en una red de blockchain, reduciendo así el riesgo y coste para todos los involucrados.

Blockchain es importante porque obtiene información y proporciona datos inmediatos, compartidos y completamente transparentes.

Las transacciones son irreversibles una vez grabadas en la blockchain, es decir, no pueden ser eliminadas ni modificadas.

Las características de la tecnología blockchain hacen que sea especialmente interesante su implementación para este proyecto, puesto que los usuarios pueden ver todos los detalles de una transacción de principio a fin, lo que le permite generar mayor confianza y eficiencia, además de obtener más oportunidades.

3.1.1 FUNCIONAMIENTO DEL BLOCKCHAIN

El funcionamiento del blockchain es sencillo, y se ilustra en la Figura 14. En primer lugar, un usuario inicia una transacción desde la wallet de la blockchain, para que posteriormente dicha transacción llegue a la base de datos distribuida y se confirme la identidad del usuario. El tipo de transacción, como una transferencia de criptomonedas, viene incluido en el código.

En segundo lugar, se crea un bloque que representa a la transacción, y este se transmite al resto de nodos de la red. Los nodos de la red aprueban la validez de la transacción, para que este pueda ser añadido a la cadena. Esto proporciona un registro transparente e indeleble de las transacciones. Por último, se ejecuta la transacción.

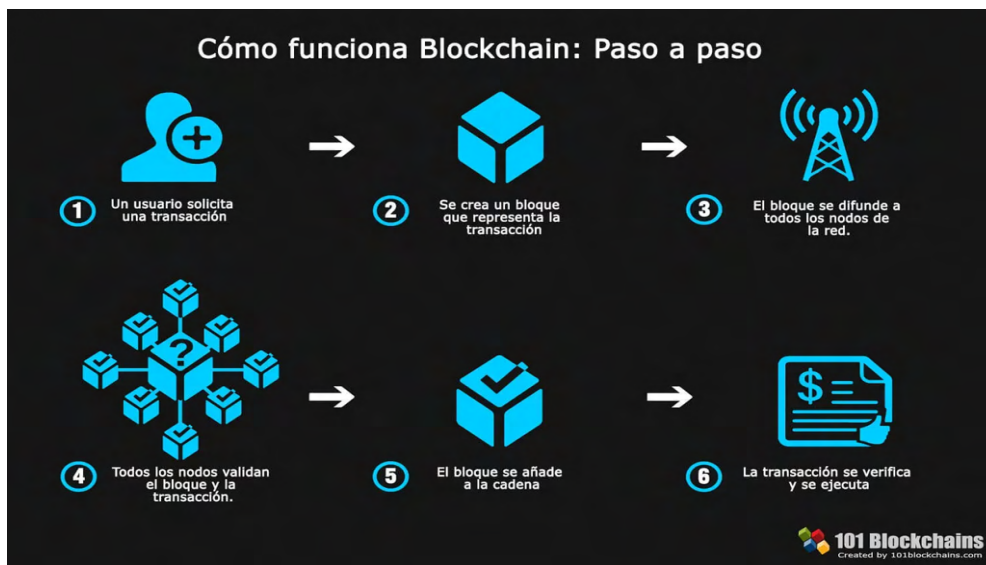


Figura 14. Funcionamiento de Blockchain

Blockchain fue posible gracias al gran desarrollo que se dio en la criptografía. Las funciones Hash juegan un papel importante en Blockchain puesto que permite vincular bloques entre sí, además de conservar la integridad de los datos de cada bloque. Las funciones hash al contrario que la encriptación asimétrica, es de sentido único o unidireccional, proporcionando una huella digital. Además, el algoritmo hash más utilizado por las diferentes blockchains suelen ser el algoritmo hash SHA-256.

Blockchain además de las funciones hash, también requiere de criptografía de clave pública o asimétrica (RSA). Esta hace uso de un par de claves, clave privada y clave pública, obteniendo una identificación única. La clave privada es usada para firmar y autorizar las transacciones asociadas a la clave pública y poder gastar los fondos de una cuenta. Por otro lado, la clave pública se utiliza para poder ser localizado en la red y recibir y verificar transacciones. Las firmas en las transacciones pueden ser validadas por la clave pública, mediante operaciones matemáticas complejas, sin necesidad de conocer la clave privada utilizada. Es de vital importancia que la clave privada no sea compartida y nadie más tenga conocimiento de esta. Además, con este sistema se protege la anonimidad y privacidad de los usuarios.

En muchas de las grandes blockchains, como Bitcoin, se utiliza la criptografía de base elíptica, entre las que destaca secp256k1.¹¹ Esta es un método de criptografía asimétrica basada en el uso de curvas elípticas. Este tipo de criptografía reduce el tamaño de las claves conservando el mismo nivel de seguridad.

El proceso que se sigue cuando se quiere verificar la validez de una firma en una transacción es el siguiente. Supongamos que un usuario, en este caso Alice, quiere enviar criptomonedas a Bob. En primer lugar, Alice crea una transacción con su clave pública, la cantidad a enviar y la dirección de Bob, y crea además una firma digital para dicha transacción, dicha firma es un hash. Para que Alice pueda verificar que es la dueña de la dirección debe firmar el hash con su clave privada. Una vez Alice ha firmado con su clave privada, debe transmitirlo al resto de la red. En este punto cualquier nodo u ordenador de la red puede verificar la validez de la

¹¹ [59] (Secp256k1, 2012)

transacción. Una vez que la transacción ha sido aprobada por consenso, se almacenan en el bloque correspondiente. Una vez se ha cerrado un bloque la transacción se convierte en parte del blockchain y Bob pasa a ser el propietario de los fondos que le fueron transferidos por Alice.

A continuación, en la Figura 15, se pretende ilustrar el proceso de validación de una firma descrito anteriormente.

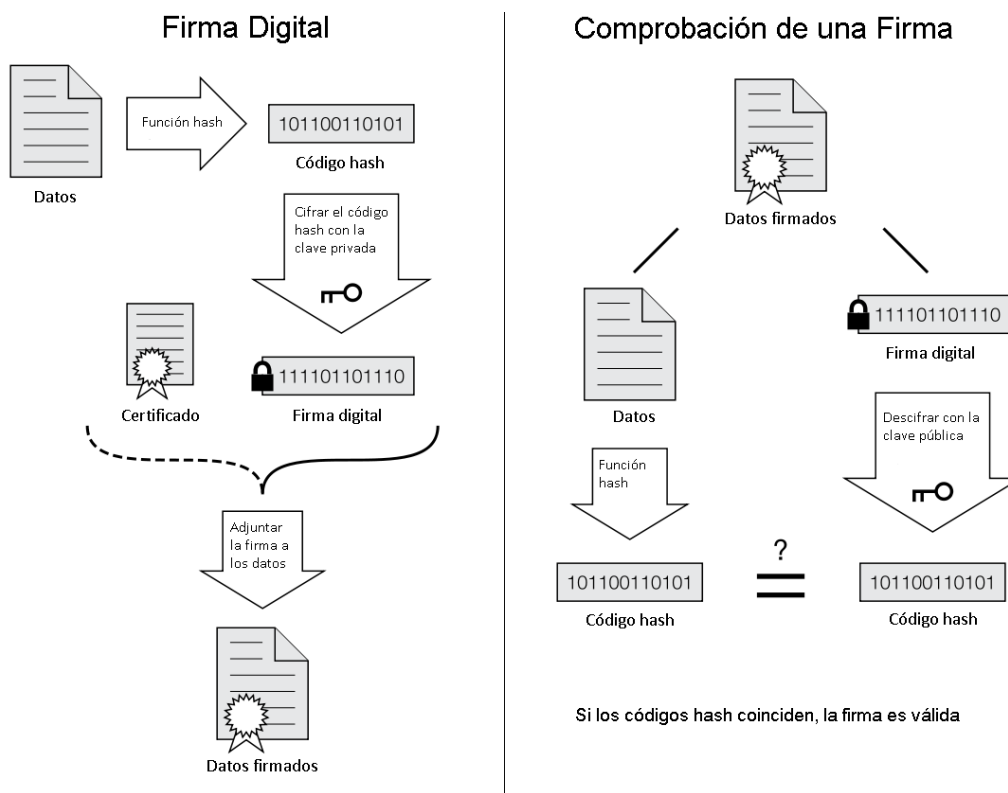


Figura 15. Proceso de validación de firmas digitales

La minería es también uno de los elementos más importantes de las blockchains. Estas garantizan que solo las transacciones legítimas sean verificadas en la blockchain. Los mineros utilizan su poder computacional para registrar transacciones en la blockchain, obteniendo recompensas por ello, tal y como se ilustra en la Figura 16.

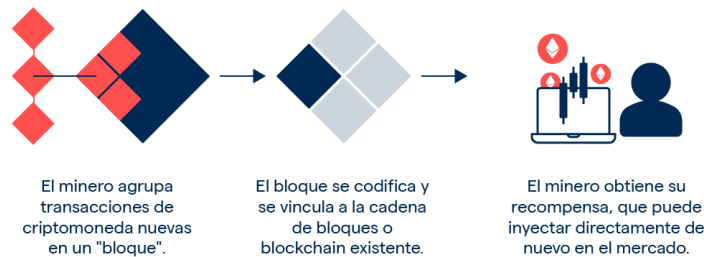


Figura 16. Proceso de minería blockchain

Es también de vital importancia distinguir y reconocer las diferencias existentes entre los distintos tipos de redes. Tal y como se puede observar en la Figura 17, en las redes centralizadas todos los nos están conectados y dirigidos bajo una sola autoridad. En las redes descentralizadas, no hay una autoridad única que dirija a todos los nodos, sino que estos tienen entidades individuales. Por último, en las redes distribuidas, cada nodo es independiente y están todos interconectados entre sí.

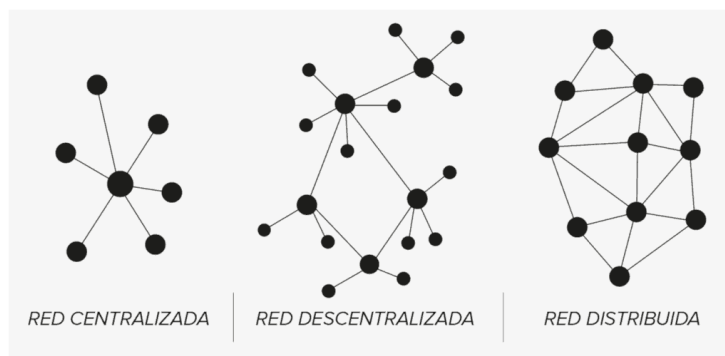


Figura 17. Tipos de redes existentes.

3.1.2 GENERACIONES DEL BLOCKCHAIN

Desde el surgimiento de la primera criptomoneda, Bitcoin, en 2008 hasta la aparición de aplicaciones descentralizadas, podemos encontrar las siguientes generaciones:

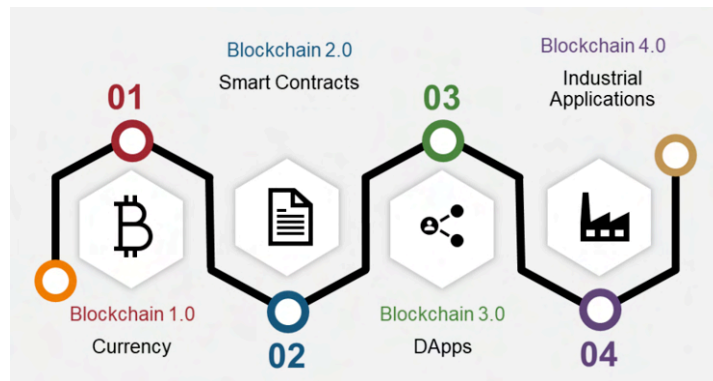


Figura 18. Generaciones del blockchain

3.1.2.1 Blockchain 1.0

La primera generación de blockchain surge con Bitcoin haciendo posible realizar transacciones financieras a través de criptomonedas. Las criptomonedas son un medio de intercambio que se crean y almacenan electrónicamente en la cadena de bloques utilizando técnicas criptográficas, controlando la creación de dinero y permitiendo la verificación de la transferencia de fondos. Una de las características principales fue eliminar las interacciones de terceros con todo tipo de transferencias de divisas. Además, esta primera generación está basada en un sistema de validación Proof of Work.

3.1.2.2 Blockchain 2.0

La segunda generación introdujo los Smart Contracts o contratos inteligentes gracias a Ethereum, esta también consolidó la idea de Proof of Stake, y sentó las bases de las Dapps a través del estándar ERC-20 para el uso de tokens. Los contratos inteligentes son pequeños programas informáticos o lógica preconstruida que residen en la cadena de bloques. Estos son programas informáticos autónomos que se ejecutan automáticamente y se activan automáticamente cuando se cumplen las condiciones requeridas. Uno de los mayores beneficios de esta tecnología es que una vez desplegados nadie puede alterar los contratos inteligentes. Los contratos inteligentes reducen los costos de ejecución y validación. Prevenir el fraude y garantizar la transparencia de los contratos.

3.1.2.3 Blockchain 3.0

La tercera generación de blockchain vino marcada por la aparición de las Dapps, acrónimo de aplicación descentralizada. Básicamente, hay almacenamiento distribuido, es decir, el código de backend pasa a ejecutarse en nodos distribuidos, en lugar de en servidores centralizados como en las aplicaciones tradicionales. Las interfaces de aplicaciones descentralizadas, por otro lado, están alojadas en almacenamiento descentralizado como Ethereum Swarm, que es una combinación de frontend junto a los contratos. Aquí, los contratos siguen una lógica descentralizada.

3.1.2.4 Blockchain 4.0

La cuarta y última generación se encuentra aún en desarrollo y tiene como objetivo buscar soluciones y enfoques nuevos que hagan que las empresas de consultoría de blockchain sean aplicables para cumplir con las demandas comerciales. Blockchain 4.0 significa hacer que Blockchain 3.0 sea aplicable en los negocios de la vida real.

3.1.3 ALGORITMOS DE CONSENSO

Los algoritmos de consenso son mecanismos que permiten la coordinación entre los usuarios y máquinas de una red distribuida. Estos son la raíz de la tecnología blockchain y son usados para seleccionar el estado correcto del registro de una transacción. Para ello deben cumplirse dos condiciones, la primera, que los nodos lleguen a un valor de salida, y la segunda, que exista un acuerdo donde todos los nodos lleguen al mismo valor de salida.

Estos algoritmos suelen llevar a cabo el siguiente proceso:

1. En primer lugar, se elige a un nodo para que proponga el siguiente valor de salida válido.
2. Los nodos de la red escuchan y validan la propuesta, para proponerla como siguiente valor válido.
3. Por último, se llega a un consenso sobre el valor único de salida.

Además, en los algoritmos de consenso suelen aparecer tres actores:

- Proponentes: líderes o coordinadores de las proposiciones de valor de salida.
- Aceptantes: escuchan, validan y responder a las solicitudes de los proponentes.
- Aprendices: solo leen el valor final de la propuesta.

Los algoritmos de consenso fueron creados para dar respuesta y solución a problemas como la pérdida o retraso de un mensaje. La solución más evidente podría ser otorgar un valor máximo al tiempo de entrega, sin embargo, los sistemas síncronos nos son prácticos en el mundo real, puesto que los ordenadores o nodos de la red pueden perder conexión, dejar de funcionar, tener diferente latencia de red o simplemente llegar desordenados. Por lo tanto, la solución debe ser un sistema asíncrono que soporte mejor los posibles fallos que podamos encontrarnos en la red.

En 1985, 3 científicos expertos en la computación distribuida y criptografía, publicaron el trabajo “Impossibility of Distributed Consensus with One Faulty Process”¹², donde tal y como expresa Adrián Rodríguez en Medium¹³ se plantea la siguiente cuestión “¿hay algún algoritmo de consenso determinista que sea seguro, tolerante a fallos y que siempre termine?”, llegando a la conclusión que los sistemas asíncronos además de ser más complejos de desarrollar, solo pueden satisfacer 2 de las siguientes tres características: seguridad, tolerancia a fallos y terminación. Normalmente se buscan blockchains seguras y tolerantes a fallos, por eso la terminación suele ser sacrificada, otorgando tiempos de valores de espera.

Sin embargo, puede llegar a haber actores maliciosos que alteren la mayoría y por tanto el resultado del consenso, y es ahí cuando surge la necesidad de crear algoritmos que sean capaces de combatir los fallos bizantinos.

Los algoritmos DLS y PBFT surgieron como necesidad de encontrar un algoritmo tolerante a fallos bizantinos y a su vez que se adaptara a un ambiente asíncrono. Sin embargo, estos no eran aplicables a sistemas escalables, y es ahí donde entra el documento técnico publicado por Satoshi Nakamoto.

12 [35] (MICHAEL J. FISCHER, 1985)

13 [54] (Rodríguez A. , 2019)

Bitcoin hizo que el proceso del consenso fuera probabilístico, donde los nodos en lugar de decidir el valor, calculan la probabilidad de que el valor se correcto.

A continuación, se listan los algoritmos de consenso más importantes desarrollados hasta el momento:

- Proof of Work (PoW)
- Proof of Stake (PoS)
- Delegated Proof of Stake (DPoS)
- Leased Proof of Stake (LPoS)
- Proof of Elapsed Time (PoET)
- Práctica de tolerancia a faltas bizantinas
- Tolerancia a faltas bizantina simplificada
- Tolerancia a faltas bizantina delegada
- Grafo de acíclico dirigido
- Proof of Activity
- Proof of Importance
- Proof of Capacity
- Proof of Burn
- Proof of Weight

3.1.3.1 Proof of Work, PoW

Proof of work fue el primer algoritmo de consenso utilizado por las blockchains, y sigue siendo utilizado en la actualidad por muchas de estas. En este consenso con la creación de cada bloque nuevo será necesario resolver un acertijo matemático mediante prueba y error, cuyo resultado es un hash. La resolución de estos acertijos es llevada a cabo por lo mineros. Una vez resuelto, se crea el nuevo bloque, confirmándose las transacciones contenidas en este.

El coste computacional para la resolución de estos acertijos es elevado y se incrementa con el crecimiento de la red, sin embargo, la comprobación del resultado es muy sencilla. Cuando un

acertijo es resuelto, el 51% de la red debe comprobar y dar por el resultado, una vez conseguido el consenso, el bloque queda asegurado y se compartido con el resto de nodos.

En este algoritmo de consenso los mineros obtienen menos recompensa con el tiempo, para prevenir ataques del 51%.

Las principales características de este protocolo son:

- Protocolo seguro, la seguridad esta directamente relacionada con el número de mineros que formen la red.
- Protocolo sencillo y fácil de implementar, facilidad del mantenimiento del software.
- Protocolo adaptable a las necesidades hardware.
- Resistente a ataques DDoS.
- Alta consumición de energía por el gran trabajo computacional requerido.

3.1.3.2 Proof of Stake, PoS

El protocolo de consenso Proof of Stake fue creado para aportar una mayor seguridad y escalabilidad en las redes implementadas. Este protocolo junto con el descrito anteriormente, PoW, son los más utilizados por las blockchains.

En este protocolo los mineros son elegidos aleatoriamente, pero no todos pueden participar, puesto que deben cumplir unos ciertos requisitos como tener una cierta cantidad de monedas o tener un cierto tiempo de participación en la red.

Este protocolo incentiva la tenencia de monedas para tener una mayor probabilidad de ser elegidos en el proceso de validación y poder así recibir ganancias por el trabajo realizado.

Este protocolo pretende solucionar algunos de los problemas que presenta el protocolo PoW, entre los que destacan:

- Falta de escalabilidad y velocidad. PoS evita la alta latencia que el proceso de minería agregaba. En el protocolo Proof of Stake las verificaciones son realizadas por nodos

que disponen de una alta tenencias de monedas, cuyo resultado es la consecución de una mayor escalabilidad y velocidad en la red.

- Uno de los mayores problemas que presenta Proof of Work son los ataques del 51%, puesto que, si un atacante consigue el 51% del poder de cómputo, puede modificar la blockchain como le parezca. Sin embargo, en este protocolo, PoS, el atacante necesitaría el 51% de las monedas, y aun así al realizar un ataque de este tipo, saldría perdiendo puesto que el valor de la moneda caería. Con esto se consigue una mayor seguridad en la red.
- Descentralización de la red. Este protocolo tiene como objetivo diversificar y democratizar el acceso a los participantes en las tareas de la red.
- Uno de los mayores problemas que presenta PoW es el alto consumo de energía que es necesario para el proceso de la minería. Por ello, PoS cambia el proceso minero por un proceso de participación.



Figura 19. Diferencias entre los protocolos de consenso PoW y PoS

3.1.4 CLASIFICACIÓN DE LA BLOCKCHAIN

Actualmente existen distintas clasificaciones de blockchain, cada una de ellas con características y capacidades únicas. Estas pueden ser públicas, privadas o híbridas.

3.1.4.1 Blockchain públicas

Este fue el primer tipo que existió, y se encuentran accesibles públicamente desde Internet. Estas mantienen sus datos, software y desarrollo de forma pública. Las principales blockchain públicas son: Ethereum, Bitcoin, Dash o Zcash. Estas, al ser públicas deben garantizar la seguridad de que ningún usuario pueda alterar con o sin intención el funcionamiento de su sistema.

Con el fin de mantener el funcionamiento y descentralización de su red, las blockchain públicas deben tomar ciertas medidas frente a los posibles ataques que puedan recibir:

- Protocolos de consenso robustos, para evitar que se añadan bloques erróneos en la blockchain.
- Protección ante DDoS: peticiones o conexiones utilizando un gran número de ordenadores o direcciones IP.
- Tolerancia a fallas bizantinas, es decir, fallas de consenso, validación o verificación de datos, en definitiva, cualquier falla del sistema.
- Protección ante ataque doble gasto, en el cual un atacante puede utilizar unas mismas monedas múltiples veces.
- Protección ante ataque de 51%, lo cual permite al ciberdelincuente obtener la mayor cantidad de recompensas por minería o realizar ataques de doble gasto.

Por otro lado, encontramos que las cuatro principales características de este tipo de blockchain son:

- Permiten a cualquiera formar parte de ella.
- Funcionamiento transparente y abierto.
- No existe ninguna autoridad central que regule su funcionamiento.
- El mantenimiento económico depende del sistema en si mismo, en general de la minería y el cobro comisión por transacción.

3.1.4.2 Blockchain privadas

Tras la aparición de las primeras blockchains y su rápida evolución y expansión, muchas empresas empezaron a interesarse por ellas, lo que conllevó al desarrollo de las blockchain privadas. Estas cuentan con los mismos elementos que las públicas, con la diferencia que las privadas necesitan una unidad central que controle o regule las acciones, funciones y permisos dentro de la blockchain.

La unidad que controla las acciones del sistema es la que debe dar acceso a los usuarios. Además, el software utilizado suele ser privado, aunque hay también otros que utilizan los softwares libres. El mayor blockchain privado existente es el proyecto desarrollado por la Fundación Linux, conocido como Hyperledger. Otras blockchains privadas importantes son Corda y Quorum.

Las tres principales características de estas redes son:

- Acceso restringido y necesidad de autorización por la unidad central.
- Acceso restringido al libro de transacciones.
- El mantenimiento económico depende de la empresa que este llevando a cabo el proyecto.

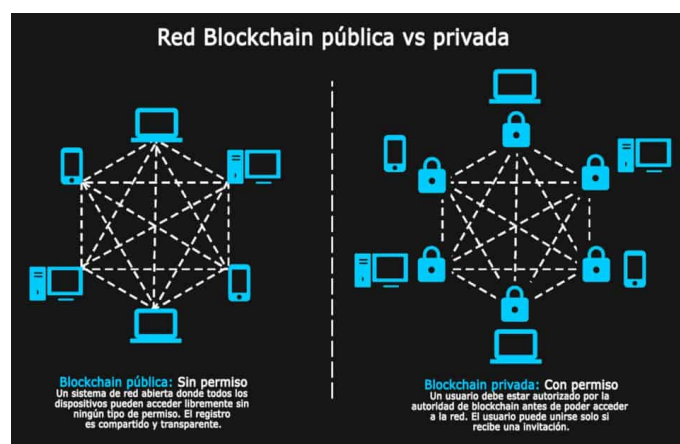


Figura 20. Diferencias entre blockchains públicas y privadas

3.1.4.3 Blockchain híbrida

Las blockchain híbridas son también conocidas como blockchain federadas y fusiona la blockchain privada con la pública con el objetivo de aprovechar lo mejor de estos dos tipos. Para ello sigue estando presente el control al acceso de recursos de la red, pero el libro con transacciones se encuentra abierto al público.

Este tipo de blockchain resulta de gran conveniencia y utilidad para empresas o gobiernos que quieran almacenar y compartir datos de una manera segura. Algunos de los casos de uso de este tipo de blockchain son en los bancos, cadenas de suministro o sector sanitario.

Por todo ello, las principales características que definen a este tipo de blockchain son:

- Acceso restringido y necesidad de autorización por la unidad central.
- Funcionamiento transparente y abierto.
- No existe las criptomonedas ni la minería.

3.1.5 ESTADO ACTUAL DE LA CRIPTOMONEDAS

Es importante conocer las diferencias entre criptomonedas y tokens. En blockchain los tokens representan activos de manera digital y son generados a partir de código, en formato Smart Contracts. Los tokens pueden ser usados de múltiples formas, algunas de ellas pueden ser, como medio de pago, para representar una acción en una empresa o para participar en un evento. Existen dos grandes categorías, los tokens fungibles y los no fungibles. Los tokens fungibles, FT, están basados en el estándar ERC-20, y son intercambiables y fraccionables. Por otro lado, los tokens no fungibles, NFT, siguen el estándar ERC-721, son tokens únicos y no se pueden dividir ni ser usados para ser su intercambio.

Las criptomonedas por tanto son tokens fungibles, entre las cuales podemos diferenciar los siguientes tipos:

- Currency token: criptomonedas.
- Utility token: unidad de cuenta de un proyecto.
- Security token: pueden funcionar como contrato de inversión o como acciones.

- Asset token: activos del mundo real como coches u obras de arte.

Como se ha comentado anteriormente, Bitcoin fue la primera criptomoneda creada. Desde entonces se han creado cientos de ellas, pero todas ellas presentan los mismos problemas:

- Alta volatilidad, lo que impide su total aceptación y consolidación como unidad de valor y su reserva. Esto se ve reflejado en el gráfico de la Figura 21.¹⁴

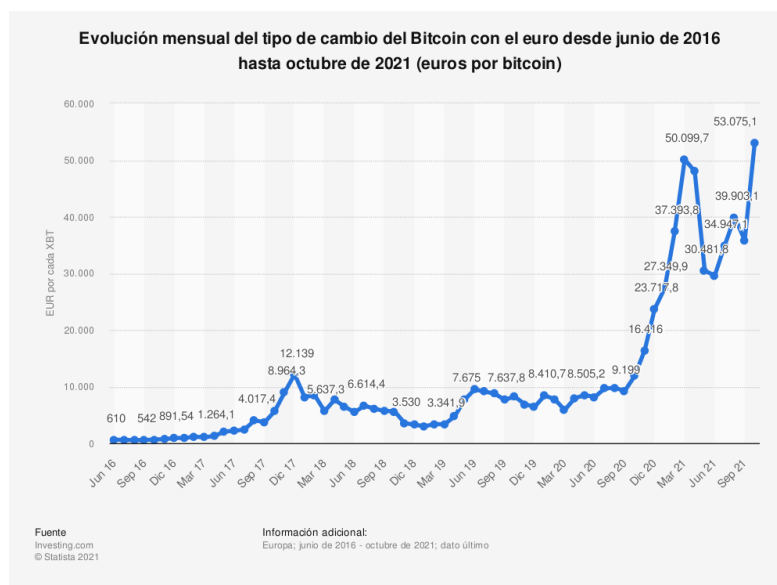


Figura 21. Evolución mensual del Bitcoin con el euro de 2016 a 2021

- Anonimidad, puesto que esto puede favorecer el uso de estas para fine ilícitos, como pueden ser la evasión fiscal o el blanqueo de capitales.
- Riesgos operacionales, los robos de Bitcoin se han sucedido con el paso de los años llegando a entre los 3 y 4 millones. Los proveedores de servicio de custodia de Bitcoins no están sujetos a supervisión, por tanto, reclamar el robo de estos es complicado, lo que genera inseguridad.
- Eficiencia y consumo de la minería. Es por ello que se están desarrollando soluciones alternativas como el protocolo de consenso, PoS.

3.1.5.1 Tokens ERC-20

Como se ha mencionado con anterioridad, los tokens pueden representar cualquier activo, y es por ello que debe existir un estándar robusto que facilite la creación de estos.

ERC-20 es un estándar para el desarrollo de tokens fungibles, donde el token tiene que tener el mismo tipo y valor que el otro token.

Algunas de las funcionalidades que ofrece este estándar son:

- Transferencias de tokens entre cuentas.
- Obtener el saldo actual del token de una cuenta.
- Obtener el suministro total del token de la red.
- Aprobar las cantidades que pueden gastar cuentas de terceros de una cuenta.

3.2 ONG

Las Organizaciones No Gubernamentales, conocidas mayoritariamente como ONG, son entidades que trabajan por el bienestar de la sociedad, mediante su participación en diferentes causas, como la proporción de alimentos, medicamentos, comida, educación u oportunidades de trabajo.

Aunque técnicamente cualquier organización privada puede ser definida como una ONG, estas organizaciones deben cumplir en la práctica tres requisitos.

- Independencia de poderes públicos y partidos políticos.
- Causas llevadas a cabo deben ser por el bien social.
- Sin ánimo de lucro.

Estas organizaciones que trabajan constantemente por la mejoría de la sociedad son necesarias para el desarrollo de cualquiera estas. Los gobiernos no pueden llegar a cubrir todas las necesidades de la sociedad y es por ello que necesitamos la ayuda del trabajo que ofrecen las ONGs.

A veces, las ONG también trabajan junto con los gobiernos o las fuerzas armadas para ayudar a abordar los desafíos del desarrollo. Lo hacen a través de técnicas como micro-donaciones, capacitación o ayudando a desarrollar la capacidad de la comunidad.

Aunque anteriormente ya existían organizaciones que defendían las mismas causas y propósitos que las ONGs, es en 1945 tras la finalización de la Segunda Guerra Mundial cuando la Organización de Naciones Unidas (ONU) creó la definición de ONG¹⁵, “cualquier grupo de ciudadanos voluntarios sin ánimo de lucro que surge en el ámbito local, nacional o internacional, de naturaleza altruista y dirigida por personas con un interés común”. Además, realizan una búsqueda de las diferentes acepciones que encontramos dentro del sector no lucrativo:

- Organizaciones No Gubernamentales, ONG.
- Organizaciones No Lucrativas, ONL o Entidades No Lucrativas, ENL.
- Organizaciones del Tercer Sector, OTS.
- Organizaciones de Voluntariado, OV.
- Organizaciones de Intervención Social, OIS.
- Organizaciones de Economía Social, OES.

Para que los proyectos de las ONGs puedan llevarse a cabo estos deben ser financiados. Por ello, las ONGs pueden hacerlo con fondos públicos o fondos privados.

- Fondos públicos: Ayudas o subvenciones que reciben. Las ONGs pueden solicitar a la administración central fondos, que estas tienen específicamente reservados para este tipo de causas.
- Fondos privados: donación de la población o de organizaciones privadas.
 - Cuotas de socios.
 - Apadrinamiento.
 - Financiación colectiva.

15 [23] (Francisco Ganga Contreras, s.f.)

- Legados solidarios.
- Patrocinio privado, ayudas económicas a cambio de publicidad de las instituciones que han colaborado.
- Mecenazgo.

El porcentaje de financiación de las ONGs, según un informe de ESADE y Fundación "la Caixa"¹⁶, es mayoritariamente a través de fondos públicos con un 60% frente al 40% de fondos privados. Además como se puede observar en el gráfico ofrecido en este mismo informe, Figura 22, España se encuentra muy alejada de países europeos como Finlandia, Austria o Suiza cuando se habla de número de donantes.

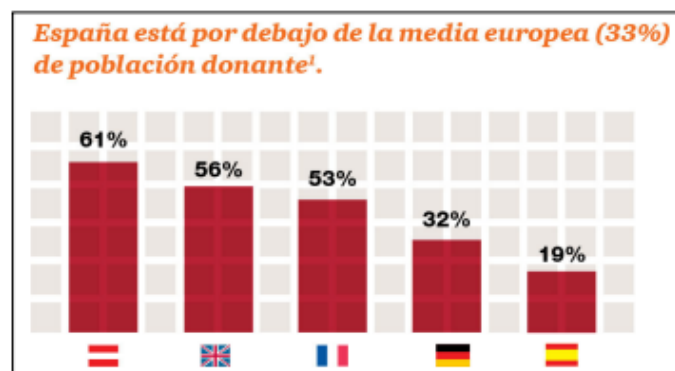


Figura 22. Porcentaje de donantes en relación a la población

Las ONG pueden distinguirse según su orientación y su área de influencia.

Según su orientación encontramos 4 grandes categorías:

- Caridad: Estas son causas que se enfocan en apoyar a los grupos en desarrollo a través de diversas actividades, como puede ser el reparto de ropa, medicamentos, comida o herramientas para el autoabastecimiento. Además, también se incluyen colaboraciones contra desastres, catástrofes humanitarias o guerras.

16 [32] ("La reacción del Tercer Sector Social al entorno de crisis", 2019)

- Servicios: Tienen como objetivo ofrecer servicios públicos que el estado no puede siempre ofrecer. Esto incluye entre otros servicios en la educación, planificación familiar o salud.
- Participativas: Proyectos que se realizan a nivel local de autoayuda, donde la comunidad colabora ofreciendo servicios, bienes o capital.
- Defensa o empoderamiento: Este tipo de causas luchan por cambios en el sistema político, social o económico. Se incluye también la lucha en defensa de los animales o de los desalojos, además de luchar por los colectivos minoritarios a los cuales se les niegan ciertos derechos.

Según su área de influencia encontramos los siguientes tipos:

- Entidades con base comunitaria.
- ONG ciudadanas.
- ONG nacionales.
- ONG internacionales.

Existen diferentes formas de colaborar con una ONG, entre las que destacan las siguientes tres alternativas:

- Aporte mensual.
- Aporte puntual.
- Donaciones en especie.

3.2.1 ESTADO ACTUAL EN LA GESTIÓN DE DONACIONES DE ONG.

El trabajo de las ONG es complejo y desafiante, pero también permite hacer una contribución valiosa a la vida o comunidad de otra persona.

El personal de las ONGs puede estar formado por voluntarios, o por empleados remunerados quienes conforman un gran porcentaje de estas.

Cómo se relaciona el trabajo de las ONG con el marketing empresarial es una pregunta clave para las empresas sociales.

Actualmente, son cada vez más las organizaciones que emplean más recursos económicos al marketing con el objetivo de conseguir un mayor número de socios y de donantes.

El gasto en personal sigue siendo significativo reduciendo así los recursos económicos únicamente dedicados a las causas.

A continuación, se muestra el gráfico realizado en un estudio de la “Situación Actual de las ONG en España” por la Fundación Lealtad¹⁷, Figura 23, mostrando la distribución del gasto por ámbito de actividad promedio.

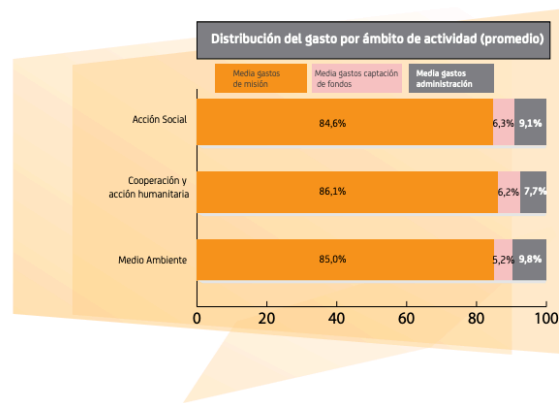


Figura 23. Gastos por ámbito de actividad de las ONGs

Por último, en la Figura 24 se muestra el porcentaje del ámbito principal de las organizaciones obtenido de la misma organización.



Figura 24. Ámbito de actuación de las ONGs

3.3 PROYECTOS DE ONGS BASADOS EN BLOCKCHAIN

Actualmente existen diferentes ONGs que está intentando incorporar la tecnología blockchain, entre las que destacan las pioneras ComGo e Itwillbe, además de otras como Click That Matters o Endaoment. ComGo intenta ofrecer sus servicios a las ONGs para que estas puedan hacer uso de su tecnología.

ComGo nació con el objetivo de mejorar la confianza, la transparencia y la capacidad de auditar todo el proceso de las donaciones económicas. Esta plataforma está basada en la tecnología Hyperledger de IBM Blockchain. Además, la plataforma ComGo permite que los fondos se puedan rastrear y las actividades se puedan monitorizar y validar en tiempo real, elemento que es clave. Además, esta entidad ha conseguido reducir los gastos de auditoría y desarrollar modelos de gestión más eficientes.

La aplicación que se pretende desarrollar a lo largo del proyecto tendrá como objetivo hacer uso de la tecnología blockchain para mejorar la transparencia de las donaciones, y el uso que se hace de estas. Se hará uso de la blockchain de Ethereum en lugar de la tecnología de Hyperledger, y se implementaran a su vez algunas mejoras que ahora mismo carece la plataforma.

La principal diferencia que existe entre la blockchain de Ethereum y la blockchain de Hyperledger es, que Ethereum es una blockchain publica, lo que significa que cualquier persona puede ver las transacciones que se realizan en ella, obteniendo así una mayor transparencia. Por contrario, Hyperledger es una blockchain privada, lo que implica que solo miembros autorizados pueden ver las transacciones que se llevan a cabo. Otra diferencia que existe entre ellas es el lenguaje que utilizan los Smart Contracts, en Ethereum los contratos son desarrollados en Solidity, mientras que en Hyperledger los contratos se desarrollan en Golang. Por último, los objetivos de ambas blockchains son diferentes, el objetivo de Hyperledger es la creación de aplicaciones B2B, mientras que el objetivo de Ethereum es el desarrollo de Dapps con múltiples tareas.

Capítulo 4. DEFINICIÓN DEL TRABAJO

En este capítulo se pretende explicar el porqué de la realización del proyecto. Además, se explicarán los objetivos que persigue el proyecto, la metodología que se ha adoptado, la planificación llevada a cabo y la estimación económica del proyecto.

4.1 JUSTIFICACIÓN

Blockchain es una tecnología nueva e innovadora, sus aplicaciones son muchas y algunas de ellas aún por descubrirse.

Esta tecnología va mas allá del Bitcoin, se puede aplicar a prácticamente a todos los ámbitos, almacenamiento en la nube, registro y verificación de datos, identificación digital, sistema de votación...

Este Trabajo Fin de Grado tiene en específico el objetivo de aplicar esta tecnología al sistema de gestión de donaciones de ONGs, puesto que, a parte de presentar altos gastos de mantenimiento de infraestructura, no terminan de generar suficiente confianza en la población española.

Una de las mayores ventajas que ofrece blockchain es la automatización de los pagos por medio del uso de los Smart Contracts, lo que reduce los altamente los costes, además de ofrecer una total transparencia y trazabilidad de las transacciones generando una mayor confianza.

Otras plataformas existes, como ComGo, siguen teniendo el mismo déficit de transparencia al hacer uso de una blockchain privada como Hyperledger, lo que implica que solo miembros autorizados pueden ver las transacciones que se llevan a cabo.

Por todo ello, se ha decidido desarrollar una aplicación en la blockchain pública de Ethereum, con el objetivo de mejorar la eficiencia y transparencia de la gestión de donaciones de ONGs.

4.2 OBJETIVOS

Como se ha explicado anteriormente, el proyecto consiste en el desarrollo de una aplicación web para la gestión de donaciones de ONGs, mediante el uso de la tecnología blockchain y los Smart Contracts. Por ello los principales objetivos son los que citan a continuación:

- Desarrollo del frontend de la aplicación web, mediante el cual el usuario interactuará con el sistema. El usuario podrá acceder a diversos proyectos que se estén llevando a cabo y observar los fondos necesarios y restantes para la finalización de este. El usuario tendrá la posibilidad de realizar la donación al proyecto que más atractivo le parezca con la cantidad que desee. Además, el usuario tendrá acceso a su lista de donaciones y ver el estado de los proyectos de sus donaciones.
- Creación de los Smart Contracts necesarios para la autorización y ejecución de las donaciones.
- Creación del backend, que sea capaz de conectar el frontend con los Smart Contracts desplegados en la blockchain. Y posteriormente creación de una red de pruebas local que simule el comportamiento del sistema para verificar su correcto funcionamiento.

4.3 METODOLOGÍA

Al comienzo del desarrollo de un proyecto una de las decisiones más importantes que se toman es la metodología que se debe adoptar, clave fundamental para que el proyecto finalice de forma satisfactoria. Dentro de las metodologías encontramos los dos marcos de trabajo más utilizados, ‘agile’ y ‘cascada’ (Waterfall).

Agile es una metodología más colaborativa enfocada a los cambios e imprevistos que se puedan encontrar, para ello busca dividir el trabajo y las tareas del proyecto de manera que estas sean flexibles. Las tareas se dividen en sprints, entregas de un periodo de entre 2 y 4 semanas. Esta metodología tiene como prioridad la satisfacción del cliente. En el manifiesto Agile se describen los siguiente cuatro puntos primordiales:

- Valorar más a los individuos y sus interacciones que a los procesos y las herramientas.
- Valorar más el software funcionando que la documentación exhaustiva.
- Valorar más la colaboración con el cliente que la negociación contractual.
- Valorar más la respuesta ante el cambio que seguir un plan.

Además, el manifiesto de la metodología Agile recoge los siguientes 12 principios que se ilustran en la siguiente figura.



Figura 25. Los 12 principios del manifiesto Agile

Por otro lado, Waterfall es una metodología más tradicional basándose en el desarrollo secuencial del proyecto. El proyecto comienza con una fase de análisis y diseño del proyecto donde además se realiza también la especificación de requisitos. Posteriormente se lleva a cabo su implementación. Una vez implementado el sistema, se comprueba su correcto funcionamiento mediante la realización de diferentes test, unitarios, integración, regresión... En esta metodología es fundamental la parte inicial del proyecto donde se lleva a cabo la planificación de este.

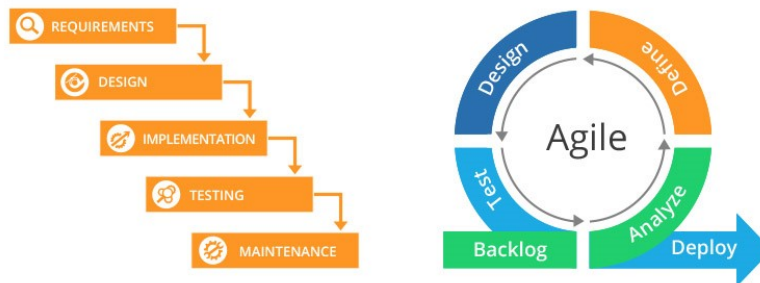


Figura 26. Metodología Agile vs Waterfall

El proyecto fin de grado ha de realizarse a lo largo del curso académico 2021/22, teniendo como plazo máximo el mes de julio, por ello se empleará la metodología Waterfall, la cual se basa en tareas secuenciales de duración fija. En el proyecto se intentará adoptar a su vez características de la metodología Agile, con reuniones periódicas con el director. Además, se hará uso de un panel Kanban en Trello para llevar a cabo el seguimiento de las diferentes tareas y subtareas del proyecto. Este panel se muestra en la siguiente figura.

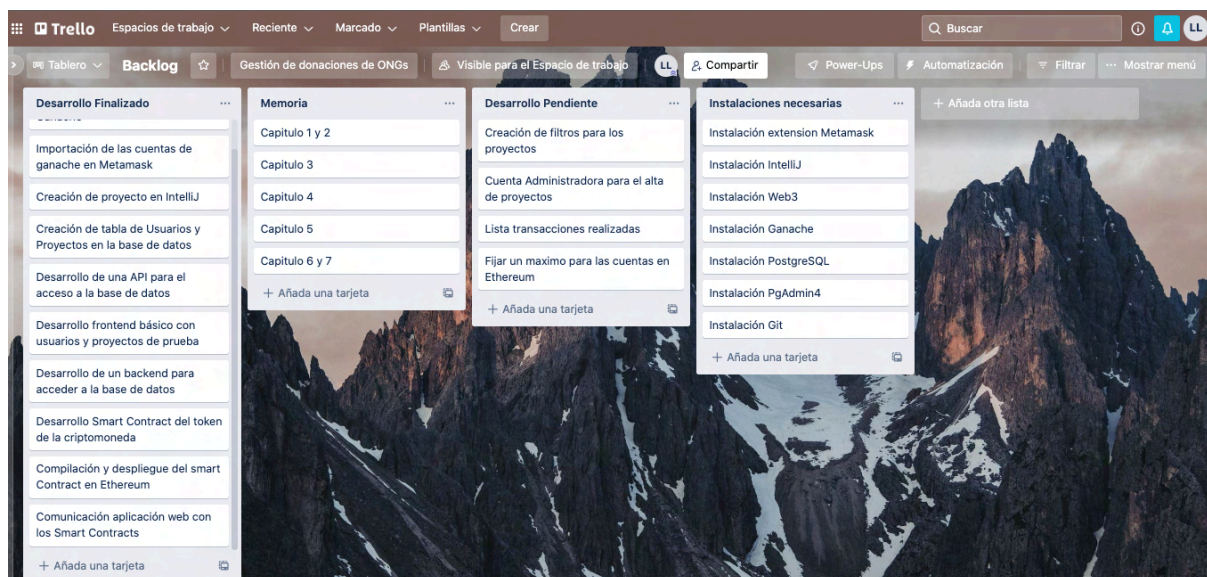


Figura 27. Panel Kanban con backlog del proyecto

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

Respecto a la planificación del proyecto con el objetivo de llevar a cabo con éxito el proyecto se definieron las siguientes tareas:

- Análisis e instalación de herramientas: definición y análisis de requisitos, así como el estudio del funcionamiento de la blockchain, junto con la instalación de los programas a emplear.
- Diseño: diseño lógico (arquitectura software) de los requisitos especificados en la etapa anterior.
- Desarrollo del sistema: programación software del sistema junto con pruebas unitarias.
- Test y simulación del sistema: simulación del sistema en una red de pruebas para verificar su correcto funcionamiento
- Memoria: desarrollo de la memoria del TFG.

A continuación, se muestra el diagrama de Gantt desarrollado con las diferentes tareas que se van a realizar durante el proyecto con sus respectivos tiempos asociados. Los tiempos asociados son flexibles, este diagrama tiene como principal objetivo mostrar la secuencia de las tareas a realizar y la importancia de estas.

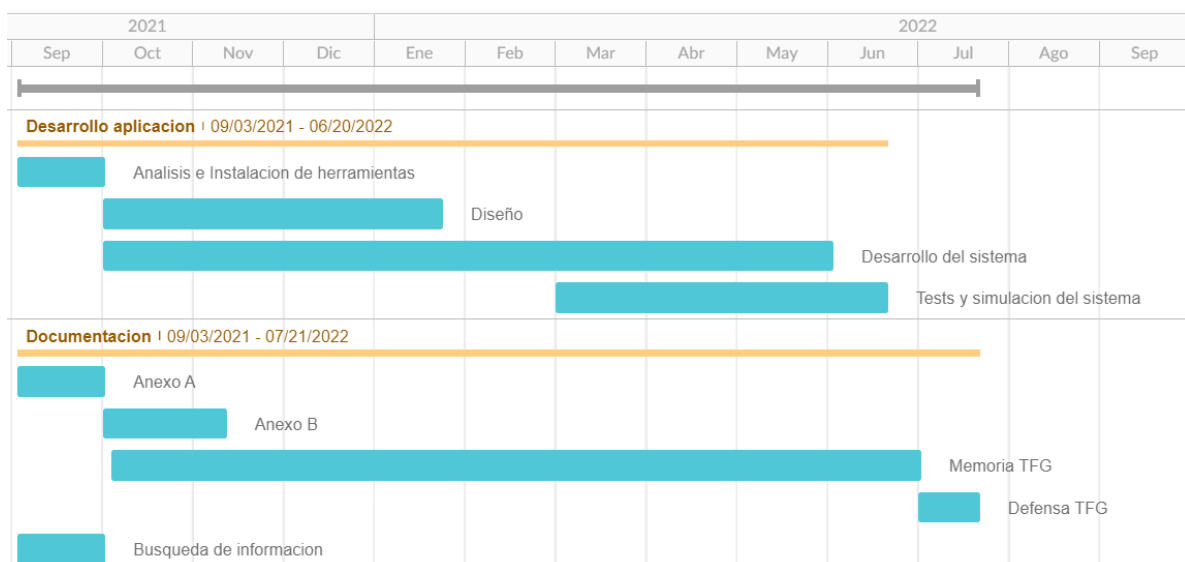


Figura 28. Planificación del proyecto

Para finalizar este capítulo, en este apartado se va a desarrollar el aspecto más importante antes de iniciar cualquier proyecto, la viabilidad económica. Para ello se va a realizar una estimación

económica del coste de desarrollo del proyecto, el coste de explotación y además se plantearán tres escenarios de ingresos.

En primer lugar, encontramos los costes del proyecto. Estos están compuestos por los costes hardware, el coste en personal para el desarrollo software y el coste del software.

- Costes hardware

MacBook Pro 2017 13-inch TouchBar	
Precio	2,200 €
Resolución	2560 x 1600
Almacenamiento	256 GB
Memoria RAM	8 GB
Velocidad del procesador	3,1 GHz
Tarjeta gráfica	Intel Iris Plus Graphics 650
Sistema operativo	macOS Catalina
Procesador	Core i5-7267U

Tabla 1. Especificación del ordenador

Monitor 23" LED FHD HP EliteDisplay E243M	
Precio	275 €
Resolución	FHD (1920 x 1080 a 60 Hz)
Tiempo de respuesta	GtG de 5 ms (con Overdrive)
Tipo de pantalla	IPS con retroiluminación LED
Tratamiento de la pantalla	Antirreflejo

Tabla 2. Especificación del monitor

- Costes en personal

El coste del desarrollo software de la aplicación se va a proceder a calcularlo teniendo en cuenta el número de horas empleadas a la realización del proyecto y el salario medio por hora de un desarrollador de cadenas de bloques según un estudio realizado por Talent, el cual se aprecia en la Figura 29.

Numero de horas = 400h

Salario medio = 71.79\$ = 68.80€

$$\text{Coste desarrollo} = 400 \times 68.80 = 27520\text{€}$$



Figura 29. Salario medio desarrollador de blockchain

- Coste software

Concepto	Detalle	Importe
Intellij IDEA Ultimate	Licencia para estudiantes anual	0 €
Github	Gratis	0 €
Metamask	Gratis	0 €
Remix IDE	Gratis	0 €
ElephantSQL Tiny Turtle	Gratis	0 €
Ganache (Truffle Suite)	Gratis	0 €

Tabla 3. Costes software del proyecto

También será necesario llevar a cabo sobre la red de Ethereum real el despliegue de los contratos, en vez de utilizar una red de pruebas. Actualmente el precio del Ether oscila entorno a los 1,130\$. Tal y como se puede apreciar en la Figura 30., el despliegue de los contratos supone 0.0209 y 0.0105 ETH, por tanto, el coste de despliegue será de $1130 * (0,0209 + 0.0105)$, es decir, 35.51\$ o 33.84€.

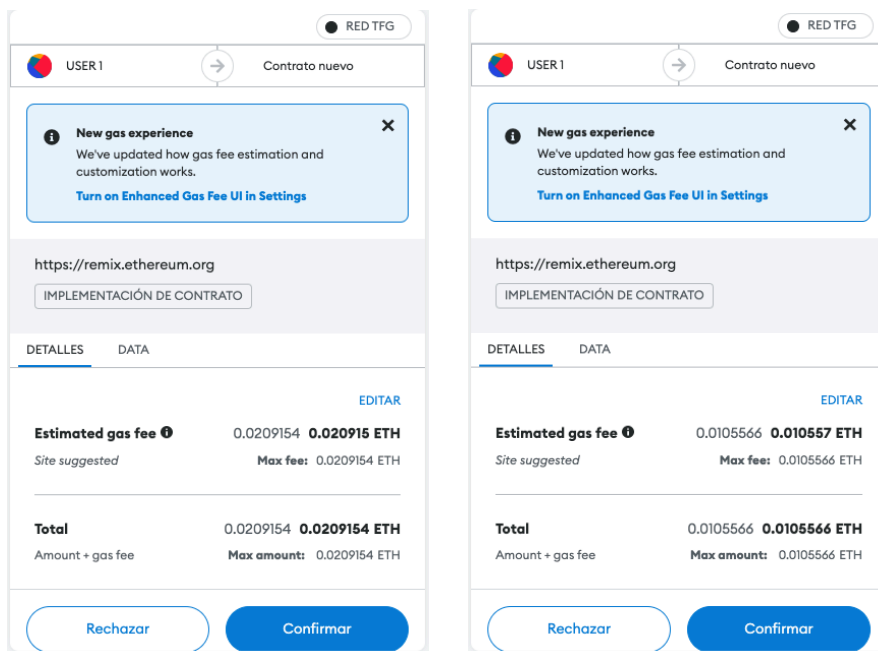


Figura 30. Coste del despliegue de los Smart Contracts en Ether

Por tanto, el coste total del proyecto es el siguiente:

$$\begin{aligned}
 & \text{Costeproyecto} \\
 &= \text{Costehardware} + \text{Costesenpersonal} \\
 &+ \text{CostedespliegueSmartContracts} = (2,200 + 275) + 27,520 + 33.84 \\
 &= 30,028.84\text{€}
 \end{aligned}$$

En segundo lugar, se va a calcular los gastos de explotación u operativos, estos son aquellos gastos necesarios para que se pueda dar el correcto desarrollo del proyecto.

Amazon Web Services, AWS, ofrece servicios web para el hosting de aplicaciones web renderizadas del lado del servidor. Dentro de los distintos servicios que ofrecen se pretende

utilizar AWS Amplify. Esta plataforma permite crear de manera rápida y sencilla aplicaciones web. El funcionamiento de este servicio se ilustra en la Figura 31.

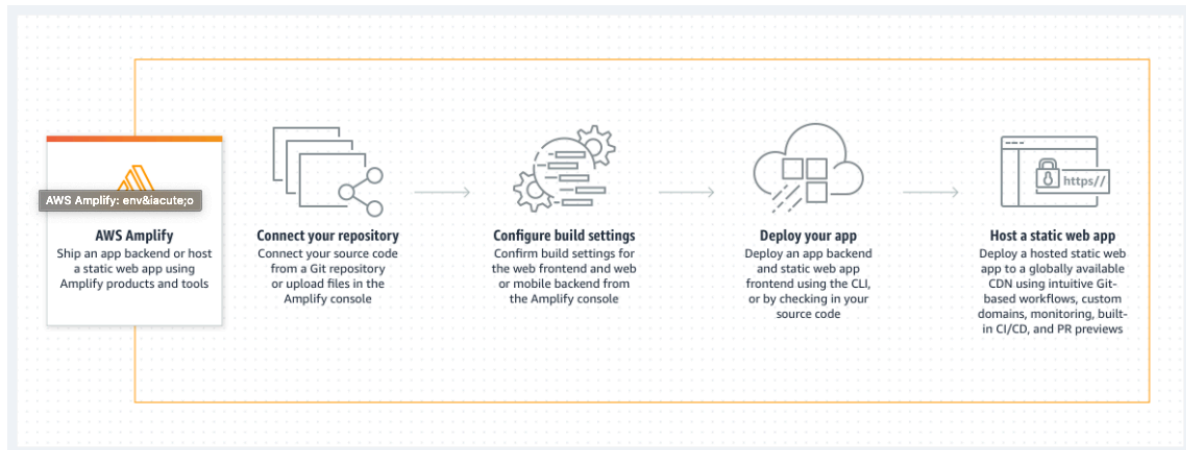


Figura 31. Funcionamiento AWS Amplify

Comparado con otras plataformas los costes de este servicio son muy económicos.

Alojamiento web estático: pago por uso	
Creación e implementación	
	0,01 USD por minuto de compilación
ALOJAMIENTO	
	0,023 USD por GB de almacenamiento al mes
	0,15 USD por GB de uso
<i>Se incluyen sin costo adicional varios sitios por proyecto y certificados SSL públicos</i>	

Figura 32. Costes de AWS Amplify

Para el cálculo de este coste se va a suponer que la aplicación web tiene 5,000 usuarios activos por día y se actualiza 2 veces por mes, un tamaño de aplicación web de 150 MB, y un tamaño promedio de la página solicitada de 2 MB. Los costes mensuales de creación e implementación no son significativos (<2\$). Por otro lado, los costes mensuales de alojamiento se muestran a continuación.

$GB_{deusopormes} = tamaño_{de la aplicación web} * número_{de creaciones mensuales}$

$$= 5,000 * \left(\frac{2}{1024}\right) * 30 = 293GB$$

$GB_{de almacenamiento pormes}$

$= tamaño_{de la aplicación web} * número_{de creaciones mensuales}$

$$= \left(\frac{150}{1024}\right) * 2 = 0,29GB$$

$Cargomensual_{de alojamiento} = 293 * 0,15 + 0,19 * 0,023 = 44\$$

Por tanto, el cargo anual por alojamiento es de 528\$ o 503€

También será necesario actualizar el plan de base de datos que se está utilizando en ElephantSQL. En un primer lugar se había utilizado la versión Tiny Turtle, esta instancia solo permite 5 conexiones concurrentes y un máximo de 20 MB de almacenamiento de datos. Por ello se va a escoger la versión Happy Hippo, la cual permite cientos de conexiones simultáneas, posee 4 GB de RAM y 100 GB para almacenamiento de datos. Tal y como se ilustra en la Figura 33, esta tiene un coste total de 99 \$ por mes, lo que supone un gasto anual de 1,188\$ o 1132€.

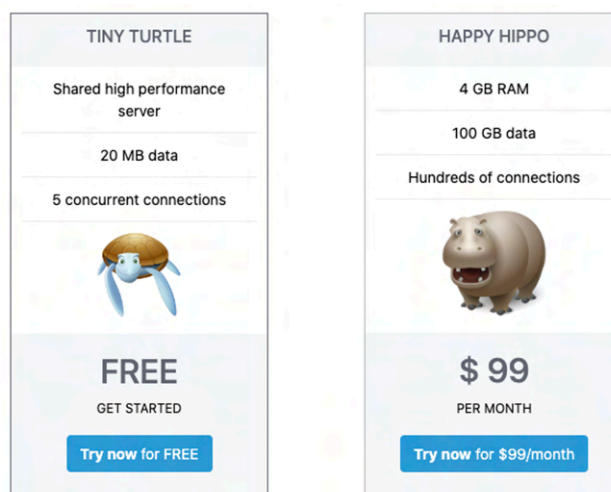


Figura 33. Especificaciones de las versiones Tiny Turtle y Happy Hippo de Elephant SQL

Dentro de los costes de explotación, también se va a analizar cuanto sería el coste de llevar a cabo una estrategia de marketing. La estrategia de marketing se va a realizar en Google. Search

Engine Marketing, SEM, es todo lo que podemos hacer en Google para promocionar nuestros productos o páginas web. Esto se divide en dos categorías, PPC (publicidad) y SEO (Search Engine Optimization).

SEO consiste en crear contenido atractivo o mejorar la página web para que esta se encuentre ubicada en posiciones más altas en los buscadores. Esta herramienta de marketing además de ser gratuita es la mejor para atraer visitas a la página web del proyecto.

Por otro lado, los costes de Google Ads son por click, CPC, y se puede establecer un presupuesto máximo mensual. El CPC en la industria software es de 3.14€.

Para la gestión de estos se contactará con una agencia de Google Ads, en este caso, SoloSEOySEM, para mejorar el posicionamiento SEO y para la gestión de publicidad de Google Ads.

Posicionamiento SEO	Publicidad Google Ads / SEM
por sólo 100€ /mes	por sólo 100€ /mes
Análisis de la estructura de la web	Campañas personalizadas según necesidades
Indexación completa de la tienda	Establecimiento de los objetivos determinando el presupuesto publicitario
Optimización del código fuente	Análisis de los términos claves
Solicitud de texto para la home	Creación de la Campaña
Análisis de las familias de producto más vendidas	Diseño de anuncios
Análisis de los productos más demandados	Control y ajuste de las inversiones
Diseño de fichas de producto optimizadas	Optimización de los parámetros de configuración
Implantación de objetivos y conversiones	Maximizar el número de clics
Seguimiento y control de las visitas web	Seguimiento de las conversiones web
Alta en Directorios	Informe mensual de resultados
Gestión de Intercambio de Enlaces	Creación de nuevos anuncios y revisión de los ya existentes
Informe mensual de conversiones/ventas	Resultados inmediatos y medibles
Servicio mensual de actualización de código	No incluye el precio de los clics
Soporte técnico telefónico, email, chat	
+ INFO	+ INFO

Figura 34. Planes de marketing de SoloSEOySEM

Sin embargo, como se aprecia en la Figura 34, el plan de gestión de Google Ads de SoloSEOySEM no incluye el CPC. Se estima que el número de click por día es de 20, un total

de 600 clicks mensuales. Suponiendo un coste de 3.14€¹⁸ (coste medio de CPC software) el coste de estos sería de $600 * 3,14 = 1,884€$. Además, como se ha comentado con anterioridad Google Ads permite fijar un presupuesto máximo, para esta aplicación dicho presupuesto se definirá en 1900€.

Por todo ello, el coste mensual total de marketing sería de $1,884 + 100 + 100 = 2084€$ y el coste anual de 25,008€

Por último, es necesario el mantenimiento software de la aplicación, para ello se creará un grupo compuesto por 5 desarrolladores software. Cada uno de estos dedicará únicamente 100 horas al año y recibirán un salario de 45€, en total un gasto de $100 * 5 * 45 = 22500€$. Además, se les proporcionará un ordenador con las especificaciones de la Tabla 1, suponiendo un gasto de $2200 * 5 = 11000€$. El alquiler de oficina supone un gasto de 290€/mes¹⁹, es decir, 3480€ anuales.

La previsión de gasto total de explotación en 5 años es de 274,115€.

Gastodeexplotacion(5)

$$\begin{aligned} &= \text{Gastos materiales} \\ &+ (\text{marketing} + \text{alojamiento servidor} + \text{gestor BBDD} + \text{oficina} \\ &+ \text{mantenimiento aplicación}) * 5 \\ &= 11,000 + (25,008 + 503 + 1,132 + 3,480 + 22,500) * 5 = 274,115€ \end{aligned}$$

Por último, como se ha comentado con anterioridad se va a llevar a cabo el desarrollo de tres escenarios de ingresos, pesimista, optimista y realista, comparándolo con los gastos iniciales del proyecto, 30,028.84€, y los gastos de explotación calculados para 5 años, 274,115€. Estos gastos hacen un total de 304,143.84€

18 [40] (¿Cuánto cuesta una campaña de Google Ads? , s.f.)

19 [4] (Alquiler Despachos Madrid , s.f.)

La mayor parte de financiación proviene de las donaciones individuales y las cuotas mensuales aportada por los socios. Únicamente las donaciones de los socios permiten la estabilidad, previsión y planificación económica.

Para el cálculo de los ingresos se va a tomar como referencia los datos de España del año 2020, donde un 13% de la población eran donantes puntuales y un 24% donantes recurrentes, socios.

Los donantes recurrentes realizaron una aportación media de 144€. La población de España en 2020 era 47.35 millones de personas, lo que supone que 11.36 millones de personas donaron, consiguiéndose así una recaudación de 1,636.4 millones de euros anuales.

- Pesimista

En este escenario la aportación de los socios es de 120 €, en lugar de los 140 € promedio que se registraron en 2020. El número de socios del primer año sería de 5,000, y gracias a las estrategias de marketing cada año incrementaría en 1%. En este escenario el total de donaciones según se indica en la Tabla 4, sería de 3,060,603, por tanto, únicamente haría falta emplear el 10% de los donativos para cubrir los gastos.

Año	Número de socios	Recaudación
1	5,000	600,000
2	5,050	606,000
3	5,101	612,060
4	5,152	618,181
5	5,203	624,362
Total		3,060,603

Tabla 4. Escenario pesimista de ingresos

- Optimista

En este escenario la aportación de los socios es de 130 €. El número de socios del primer año sería de 15,000, y gracias a las estrategias de marketing cada año incrementaría en 5%. En este escenario el total de donaciones según se indica en la Tabla 5, sería de 10,774,981, por tanto, haría falta emplear solo el 3% de los donativos para cubrir los gastos.

Año	Número de socios	Recaudación
1	15,000	1,950,000
2	15,750	2,047,500
3	16,538	2,149,875
4	17,364	2,257,369
5	18,233	2,370,237
Total		10,774,981

Tabla 5. Escenario optimista de ingresos

- Realista

En este escenario la aportación de los socios es de 125 €. El número de socios del primer año sería de 10,000, y gracias a las estrategias de marketing cada año incrementaría en 2%. En este escenario el total de donaciones según se indica en la Tabla 6, sería de 6,505,050, por tanto, haría falta emplear solo entorno al 5% de los donativos para cubrir los gastos.

Año	Numero de socios	Recaudación
1	10,000	1,250,000
2	10,200	1,275,000
3	10,404	1,300,500

4	10,612	1,326,510
5	10,824	1,353,040
Total		6,505,050

Tabla 6. Escenario realista de ingresos

Anteriormente en la Figura 23, se ha mostrado los porcentajes del gasto de donaciones, donde aproximadamente es el 85% el porcentaje dedicado a la misión.

Tras la elaboración de los tres escenarios de ingresos se concluye que es viable el desarrollo del proyecto, y se espera poder aumentar el porcentaje de los gastos dedicados a las causas de las ONGs en un 10%, dedicando así un total del 95% de estos.

Capítulo 5. SISTEMA DESARROLLADO

En este capítulo se va a describir el sistema desarrollado en detalle, para ello, se va a dividir el capítulo en tres partes, análisis del sistema, diseño de la aplicación e implementación.

5.1 ANÁLISIS DEL SISTEMA

Dentro de la programación se pueden encontrar distintos paradigmas o estilos de programación software, entre los que destacan, el paradigma imperativo, declarativo, la programación estructurada o la programación orientada a objetos.

Para el desarrollo de la aplicación del proyecto de Fin de Grado se ha llevado a cabo, en base a la estructura de datos y del conjunto de componentes del sistema, el paradigma de programación orientada a objetos (POO), donde el sistema es un conjunto de objetos que interactúan entre sí.

Con el fin de identificar los principales componentes estructurales del sistema y las relaciones entre estos, se ha desarrollado en primer lugar el diseño arquitectónico de la aplicación.

Una de las primeras etapas del proceso de desarrollo ha sido la elaboración de la arquitectura global del sistema. Además, con el fin de conseguir la arquitectura más adecuada para la aplicación se estudiaron los requisitos no funcionales. Estos son, rendimiento, disponibilidad, seguridad, escalabilidad y mantenibilidad.

Por todo ello, se ha decidido llevar a cabo una arquitectura cliente-servidor, obteniendo adicionalmente todas las ventajas de una arquitectura por capas. El sistema reparte las tareas entre diferentes servicios y se obtendrá acceso a estos a través de peticiones HTTP a la API.

Los servicios con los que cuenta el sistema desarrollado son, servicio de usuarios, servicio de organizaciones y servicio de proyectos. Estos se inician una vez se despliega la aplicación tal y como se aprecia en la Figura 35.

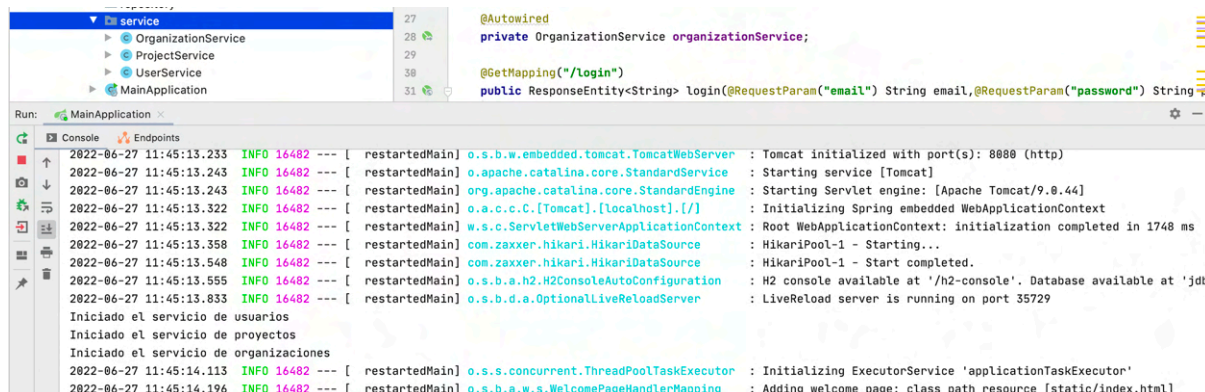


Figura 35. Servicios de la aplicación

La comunicación de cada uno de estos servicios se da a través de la API que se ha desarrollado. La arquitectura que se ha empleado para el desarrollo de la API es la arquitectura REST. Esta arquitectura, crea servicios web aprovechando el diseño de las peticiones HTTP. Se basa en el patrón cliente-servidor donde el cliente realiza peticiones sobre recursos del sistema y el servidor responde con el estado de la petición y el recurso solicitado. Existen 9 peticiones HTTP para interactuar con las APIs, de los cuales los 4 primeros son los más conocidos y utilizados:

- GET: obtener un recurso.
- POST: crear un nuevo recurso,
- PUT: actualizar un recurso.
- DELETE: borrar un recurso.
- HEAD: obtener un recurso sin el cuerpo de la respuesta.
- CONNECT: acceso al servidor identificado por el recurso.
- OPTIONS: describe las opciones de comunicación para el recurso de destino.
- TRACE: prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso objetivo proporcionando un mecanismo de debugging.
- PATCH: modificación parcial de un recurso.

Estas peticiones, están formadas por un ‘HTTP verb’ que define la operación a realizar, un header con la información de la petición, una ruta hacia el recurso y un body con los datos de la petición. A continuación, se muestra un ejemplo de petición POST, Figura 36.

```

var data = {"email": mail,
  "password": password,
  "name": name,
  "lastname": lastname,
  "publicaddress": publicaddress,
  "type": typw
};
fetch( input: 'http://localhost:8080/api/user/',
  init: {
    method: 'POST',
    body: JSON.stringify(data),
    headers: {
      'Content-Type': 'application/json',
      'Accept' : 'application/json',
    }
  }
})

```

Figura 36. Petición POST

La aplicación a su vez sigue el patrón MVC, Modelo, Vista y Controlador, que son las capas o componentes en las que se organiza la aplicación web. El modelo está formado por los datos del dominio, así como lógica de negocio y el acceso a la base de datos. La vista se encarga de generar la interfaz de la aplicación. Por último, el controlador actúa como intermediario entre la aplicación y el cliente/usuario. El funcionamiento de este patrón se muestra en la Figura 37.

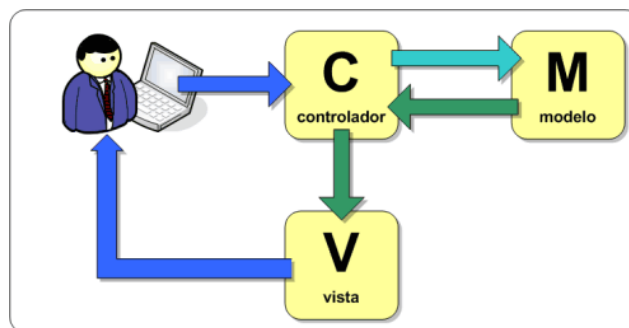


Figura 37. Funcionamiento patrón arquitectónico MVC

Por último, se muestran las capas de la aplicación del sistema desarrollado en la Figura 38.

```

└─ packages
  └─ configuration
    └─ AppConfiguration
  └─ controller
    └─ exception
    └─ RestProjectController
  └─ model
    └─ Organization
    └─ Project
    └─ User
    └─ ValidationErrors
  └─ repository
    └─ OrganizationRepository
    └─ ProjectRepository
    └─ UserRepository
  └─ service
    └─ OrganizationService
    └─ ProjectService
    └─ UserService
    └─ MainApplication

```

Figura 38. Capas de la aplicación

5.2 DISEÑO

En este apartado se va a explicar el diseño y modelado del sistema a desarrollar por medio de diferentes diagramas. Estos diagramas explican y permiten entender la estructura software, y el comportamiento de los usuarios con la aplicación.

5.2.1 DISEÑO DE ARQUITECTURA

El diseño de la arquitectura de la aplicación que se representa en la Figura 39, consta de ciertos elementos fundamentales. En primer lugar, los usuarios o clientes, quienes tienen acceso a la aplicación web y a los recursos almacenados en la blockchain y la base de datos.

En segundo lugar, encontramos Spring Boot Application, quien tiene embebido el contenedor web servlet de Apache Tomcat. Para el desarrollo web de este proyecto se ha hecho uso de Spring Web quien a su vez hace uso de Spring MVC, este se encarga de soportar todas las peticiones web (HTTP) a la API, y redirigirlas a los componentes que sean necesario. Además, Spring MVC es considerada la columna vertebral de los servicios web RESTful. Como se ha explicado anteriormente las llamadas a la API comunican los diferentes servicios de la aplicación.

Spring Data JPA (Java Persistence API), es un framework de Java que a través de un conjunto de interfaces y APIs persisten los recursos en una base de datos relacional. Además, Hibernate es una herramienta de mapeo objeto-relacional que mediante diferentes Beans facilitan el mapeo de atributos en bases de datos, además del modelo de objetos de una aplicación.

Otro de los elementos que encontramos en la arquitectura es el gestor de la base de datos relacional, de los cuales se ha escogido PostgreSQL. Este gestor almacenará los recursos de los usuarios, los proyectos y las organizaciones.

Por otro lado, encontramos los componentes relacionados con la tecnología blockchain. El primero de estos elementos es Web3, quien se encarga de establecer comunicación entre el usuario y la blockchain de Ethereum. A través de esta librería podremos acceder directamente

a los datos almacenados en la blockchain, en este caso, las donaciones, o podemos acceder a Metamask, para poder así aceptar las transacciones que se quieran realizar.

Dentro de la red de blockchain de Ethereum se encuentra el último componente, Ganache, en el cual se puede realizar la simulación y testeo de la blockchain de Ethereum a través de una red de pruebas local. En este framework se despliegan los Smart Contracts compilados.

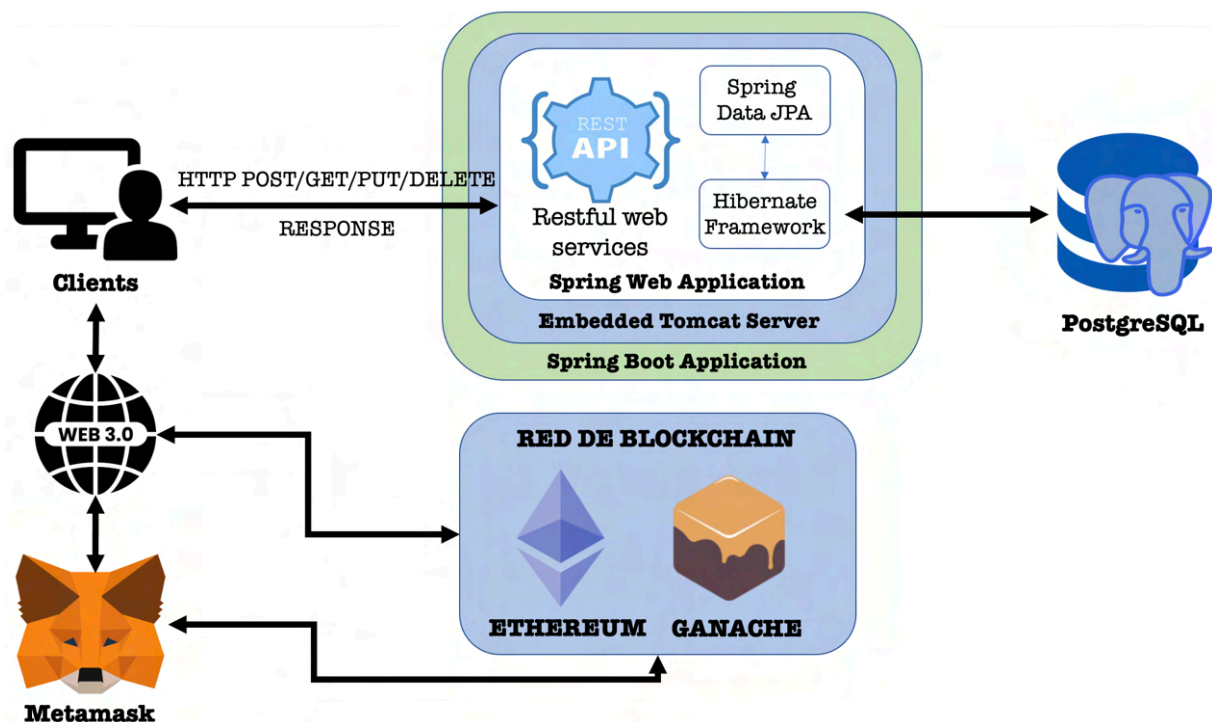


Figura 39. Arquitectura del sistema

5.2.2 API ENDPOINTS

Como se ha comentado con anterioridad, la comunicación de cada uno de estos servicios se da a través de la API. La arquitectura que se ha empleado para el desarrollo de la API es la arquitectura REST. Los 4 métodos de peticiones HTTP o 'HTTP verbs' más utilizados para interactuar con la REST API son, GET/POST/DELETE/PUT.

Para el desarrollo de los endpoints se han seguido los principios de las API Rest, los cuales se ilustran pro medio del ejemplo de la Figura 40.

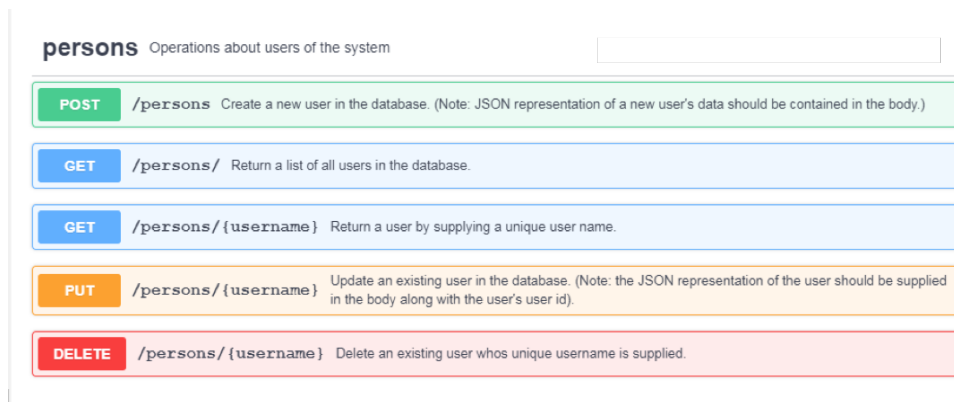


Figura 40. Ejemplo endpoints API REST

Teniendo en cuenta dichos principios, los endpoints a desarrollar en este proyecto son los siguientes:

- Servicio de Usuarios

GET - /users/ Lista con los usuarios

POST - /users/ Creación nuevo usuario

GET - /users/{email} Usuario específico

PUT - /users/{email} Actualizar usuario

- Servicio de Proyectos

GET - /projects/ Lista de los proyectos

POST - /projects/ Crea nuevo proyecto

GET - /project/{address} Info proyecto

PUT - /project/{address} Actualizar

GET - /project/{date} Proyectos por fecha

GET - /project/{category} Proyectos por categorías

GET - /project/{organization} Proyectos por organización

PUT - /project/{address}/status Actualizar estado del Proyecto (pendiente/terminado)

- Servicio de Organizaciones

GET - /organizations/ Lista organizaciones

POST - /organizations/ Crear organizacion

GET - /organizations/{id} Organización

PUT - /organizations/{id} Actualizar

5.2.3 ESTRUCTURA DE DATOS

La gestión de datos como se ha explicado anteriormente, se va a llevar a cabo en la base de datos de PostgreSQL y la blockchain de Ethereum. Con el objetivo de mostrar las relaciones existentes entre las diferentes tablas que conforman la base de datos se ha llevado a cabo un diagrama entidad-relación, Figura 41.

En este diagrama se representa la relación entre los usuarios, proyectos y organizaciones. En primer lugar, los usuarios pueden donar a diferentes proyectos, y los proyectos pueden ser donados por diferentes usuarios. La relación entre estas dos entidades se da a través de donar, la cual se almacena en la blockchain de Ethereum, para que cualquier usuario pueda tener acceso a las donaciones que se hayan realizado, obteniendo así una mayor transparencia.

Al ejecutarse el contrato TokenERC20.sol se almacena en la blockchain, la publicaddress del usuario, la publicaddress del proyecto y la cantidad que se haya donado.

Por último, se muestra la relación entre los proyecto y organizaciones. Un proyecto corresponde a una organización, mientras que una organización puede llevar a cabo múltiples proyectos

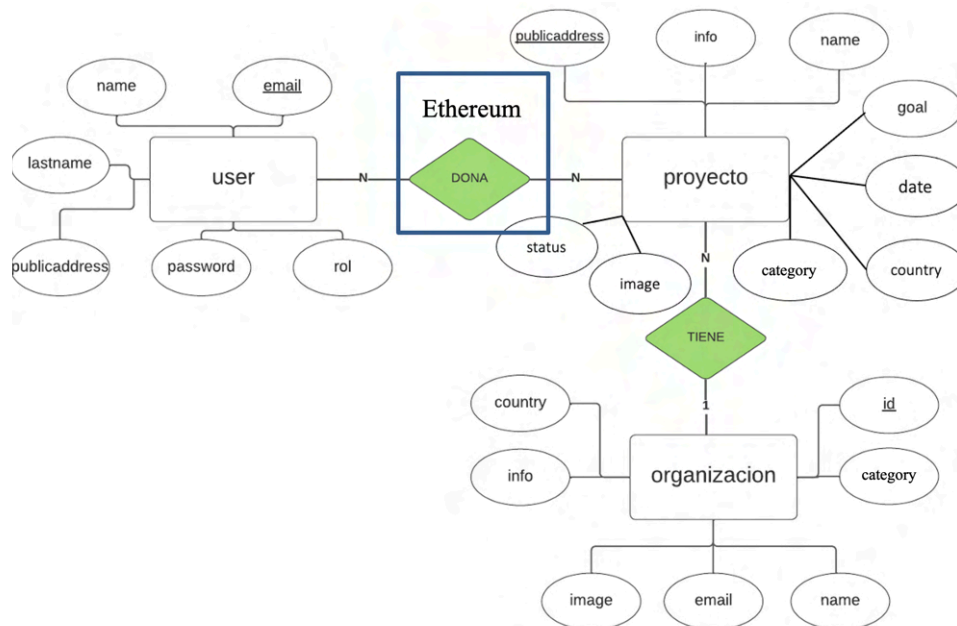


Figura 41. Diagrama entidad-relación

5.2.4 DIAGRAMAS DE CASOS DE USO

Para representar los casos de uso, expresión de los requisitos funcionales, se han llevado a cabo diagramas UML de comportamiento.

El primer diagrama de caso de uso, Figura 42, tiene como actores principales a los clientes. En función de si el cliente ha iniciado o no sesión en la aplicación, podrá acceder o no a diferentes funciones.

En primer lugar, un usuario que no haya iniciado sesión podrá ver los proyectos de las ONGs y opcionalmente los podrá filtrar por categorías o ver una información más detallada acerca de este. El usuario tendrá acceso a su vez a las organizaciones y toda su información, así como los proyectos que estén llevando a cabo. El usuario tendrá acceso a todas las donaciones que se hayan realizado, y podrá ver las donaciones que se hayan realizado a un proyecto en específico. Por último, los usuarios que no hayan iniciado sesión tendrán la posibilidad de hacerlo, y en caso de no contar con una cuenta, podrán crearla.

En segundo lugar, los usuarios que sí hayan iniciado sesión, dispondrán de funcionalidades extra, como editar su perfil, donar, o ver las donaciones que hayan realizado.

Para el acceso a las donaciones, puesto que se encuentran almacenadas en Ethereum, es necesario obtenerlas por medio de este sistema. Además, cuando un usuario done, deberá primero conectarse a su cuenta de Ethereum en Metamask, y posteriormente si el saldo de la cuenta es mayor al del valor de la donación, esta se realizará.

En último lugar, encontramos al sistema de datos, el cual se encarga de actualizar el estado de los proyectos, con el objetivo de mostrar a los usuarios únicamente aquellos proyectos que no hayan sido completados.

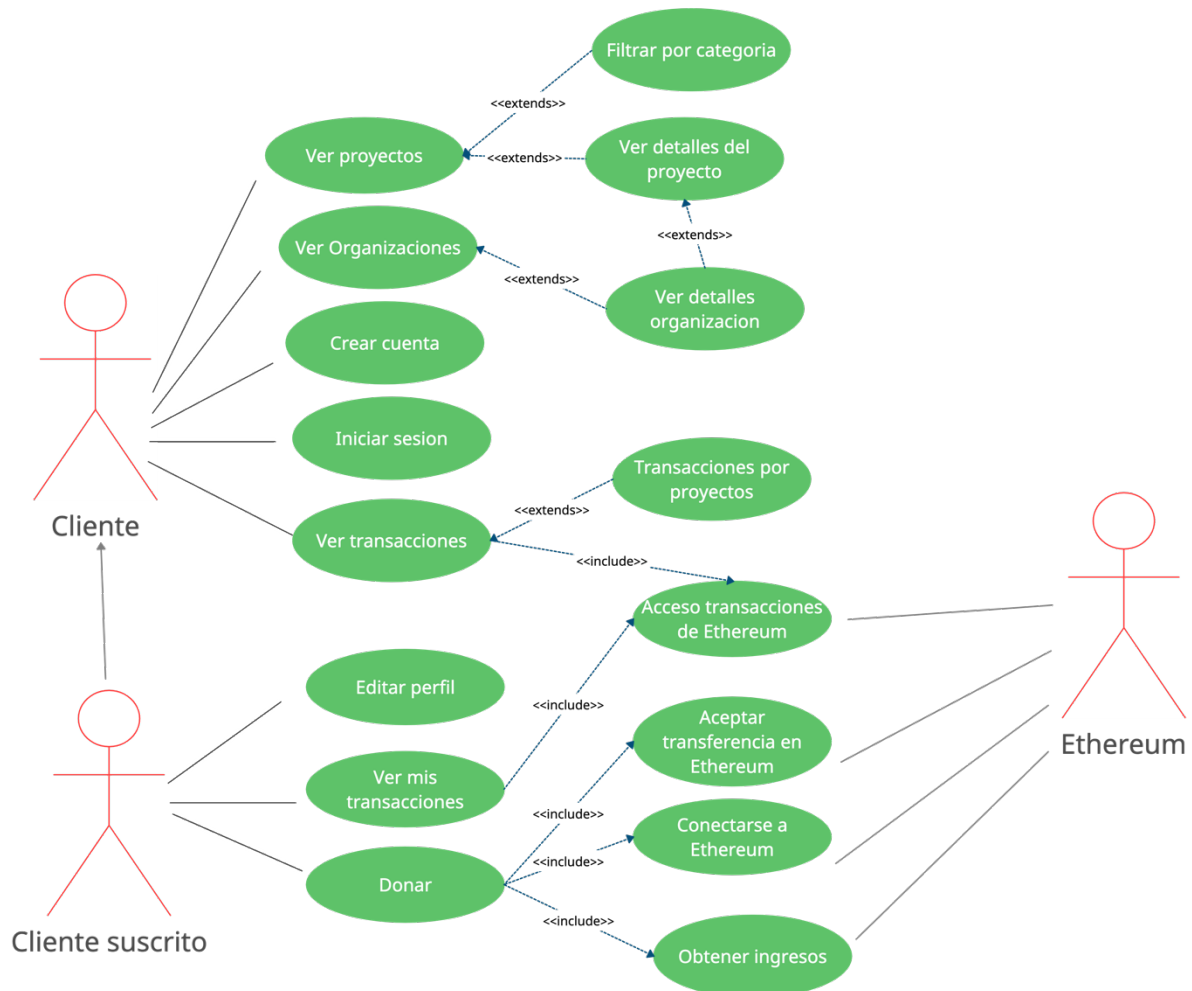


Figura 42. Diagrama caso de uso – cliente

El segundo diagrama, Figura 43, tiene como actores principales a las organizaciones y el administrador.

Las organizaciones podrán ver sus proyectos, así como editarlos. Además, podrán también crear o eliminar proyectos.

Los administradores, tendrán acceso a las mismas funcionalidades que las organizaciones, puesto que tendrán acceso a los proyectos de estas, y gestionarán además la edición, creación y eliminación de las organizaciones.

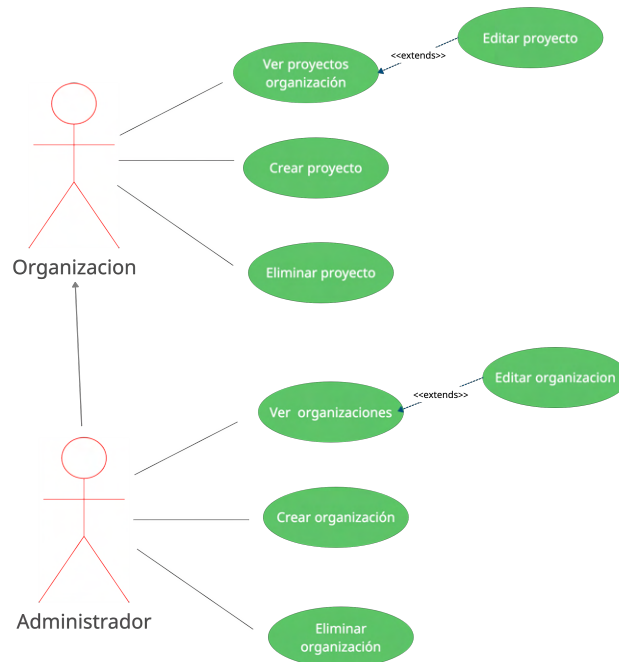


Figura 43. Diagrama caso de uso - organización y administrador

5.2.5 DIAGRAMAS DE CLASE

Una vez el diseño de la arquitectura y el diagrama de los casos de uso han sido realizados, se ha procedido a la realización de un diagrama de alto nivel que describa los diferentes tipos de objetos del sistema y las relaciones que existan entre estos. También se ilustran los atributos y operaciones de las clases.

El diagrama realizado, Figura 44, cuenta con 7 clases. En primer lugar, está la clase usuario, de la cual en función del que estos tengan derivan en clientes, organizaciones o administradores.

En segundo lugar, encontramos la clase CuentaEthereum, de la cual dispondrán proyectos y clientes/donantes.

Las organizaciones se relacionan con los proyectos, puesto que, una organización tendrá o no diversos proyectos y un proyecto estará asociado a una organización.

Por último, a pesar de no ser una clase, el contrato ERC20Token.sol, representaría la clase donación, que relaciona a los usuarios con los proyectos. Este contrato relaciona dichas clases con los atributos, from, correspondiente al donante, y el atributo, to, que corresponde a la dirección pública del proyecto, además de contener el atributo con la cantidad de la donación.

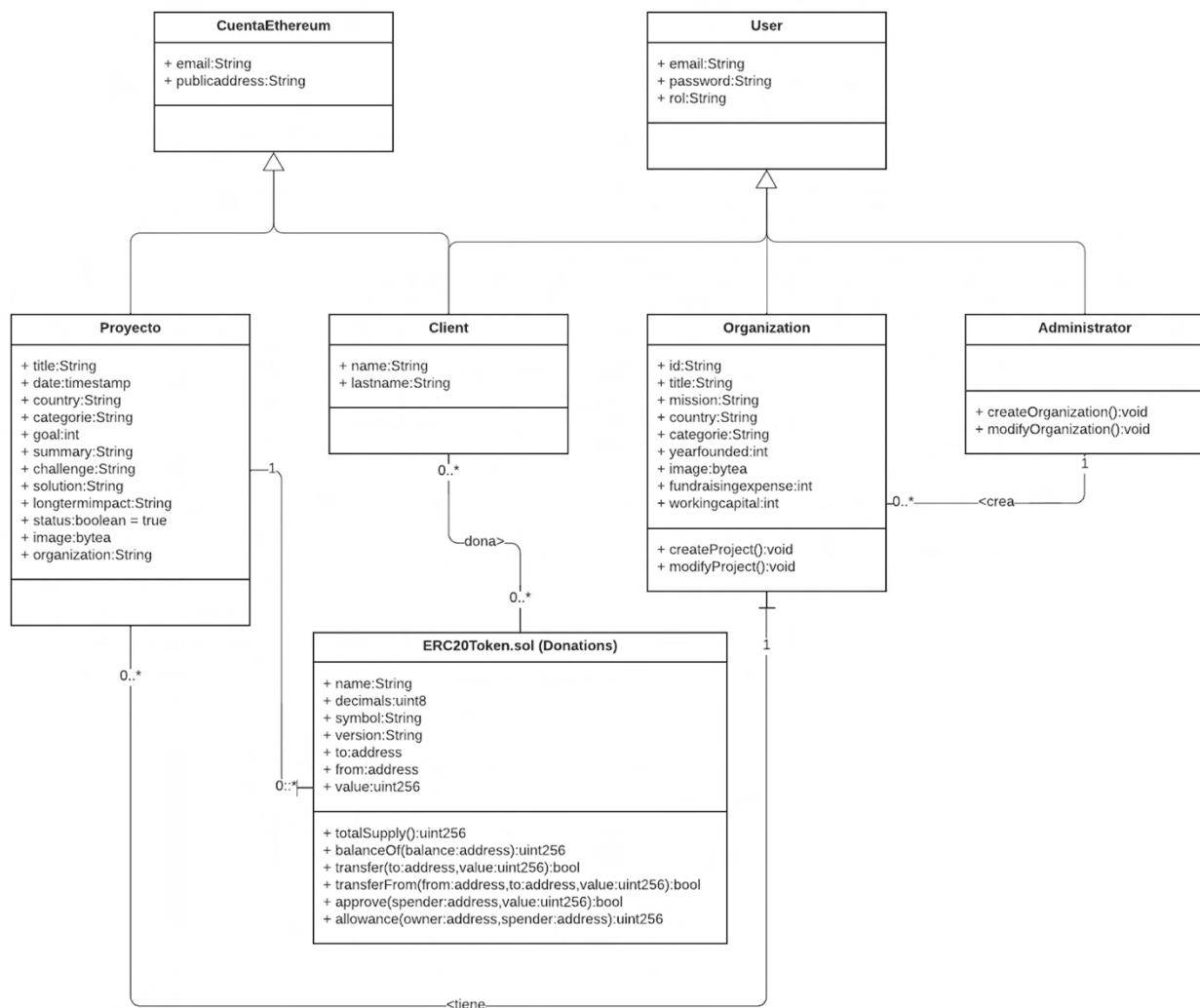


Figura 44. Diagrama de Clases

5.2.6 DIAGRAMA DE SECUENCIA

Se han desarrollado tres diagramas de secuencia de tres casos de uso, que muestran la interacción de diferentes objetos de la aplicación. El objetivo de estos es conocer el funcionamiento de las interacciones de los usuarios con el sistema.

El primer diagrama de secuencia desarrollado es, signup o registro, Figura 45. En este diagrama un cliente se registrará en el sistema. Para ello, en primer lugar, el usuario rellenará sus datos, y posteriormente si el usuario no existe se procederá a registrarlo.

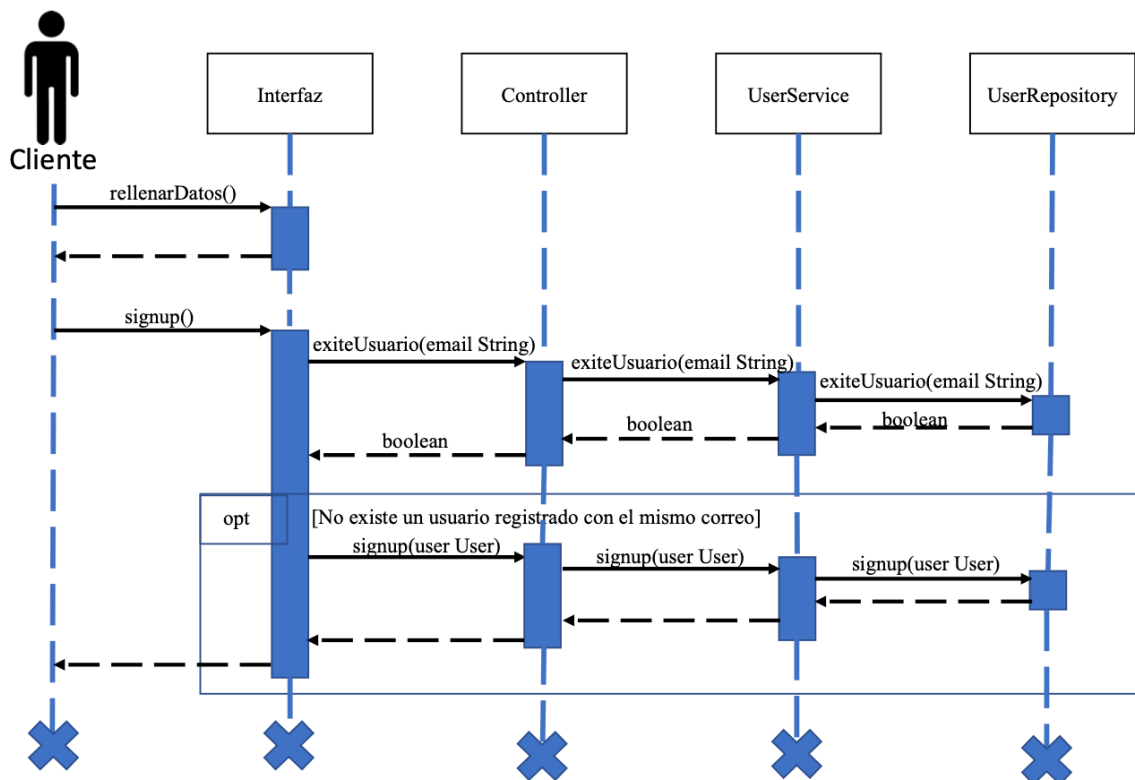


Figura 45. Diagrama de Secuencia - signup

El segundo diagrama desarrollado ha sido, el caso de uso de donar, . El donante verá en primera instancia todos los proyectos disponibles. En segundo lugar, el usuario seleccionará un proyecto a donar. En último lugar, el usuario realizará la donación. Para que la donación sea realizada, el usuario se conectará a Metamask con su cuenta de Ethereum, se obtendrá el balance de este, y si su saldo es mayor que la cantidad a donar se transferirá la donación. Para que la donación sea transferida el client deberá aceptar y confirmar la transacción en Metamask.

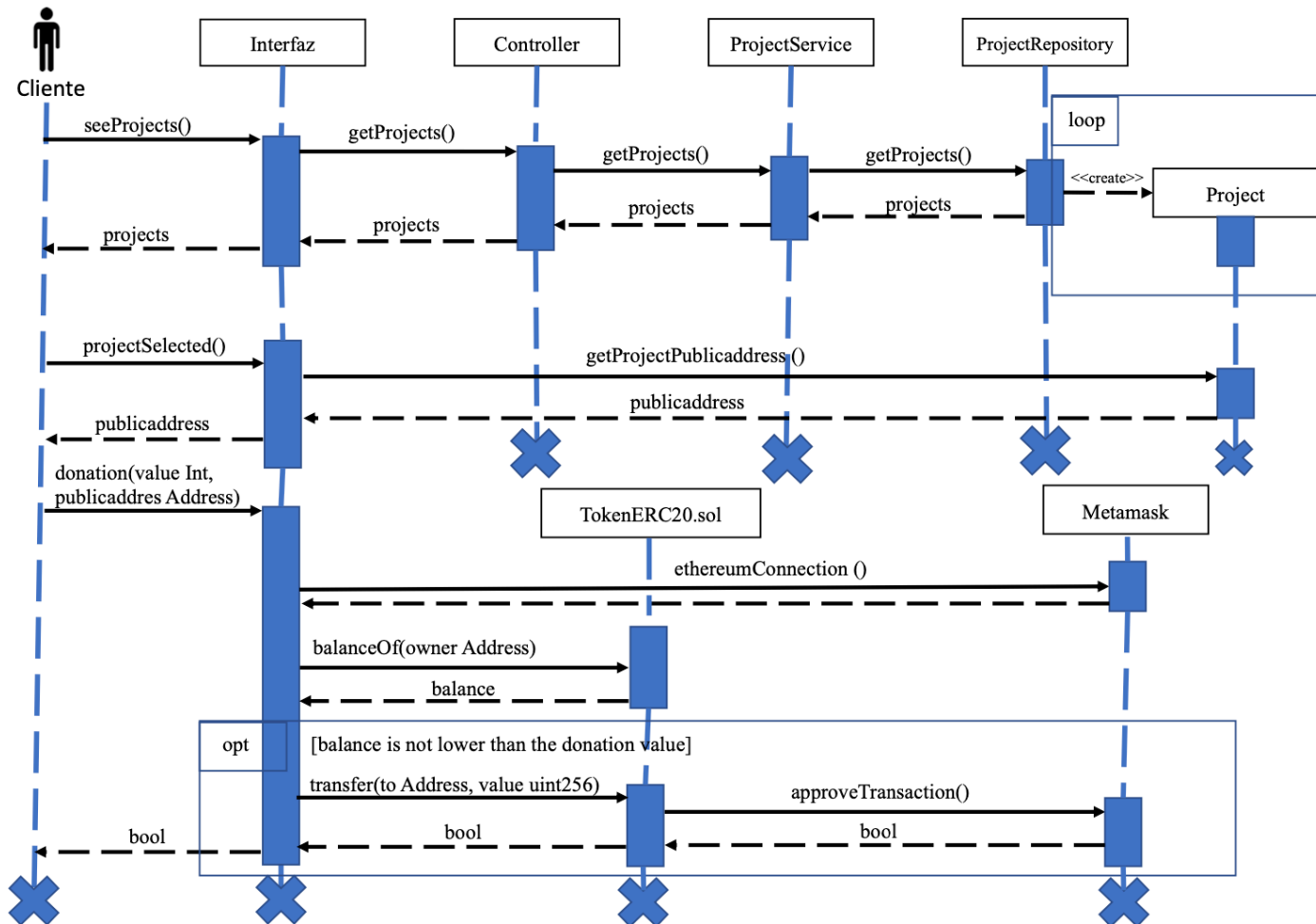


Figura 46. Diagrama de Secuencia - donar

El último diagrama que se ha desarrollado ha sido la visualización de las donaciones, Figura 47. En este caso de uso el usuario solicita ver las donaciones que se hayan realizado. Para ello, en primer lugar, se realiza una conexión con Ethereum, y posteriormente se calcula el número de bloques total. En segundo lugar, se realiza un bucle ('loop'), donde de cada bloque se extrae la transacción y sus datos. En tercer lugar, se crea la clase donación donde se almacena la información de la transacción obtenida de Ethereum. Por último, al no guardarse en Ethereum el nombre del proyecto, este se obtiene en la base de datos, a través de su dirección pública, y se almacena en la clase de donación creada con anterioridad, para que esta sea devuelta al usuario.

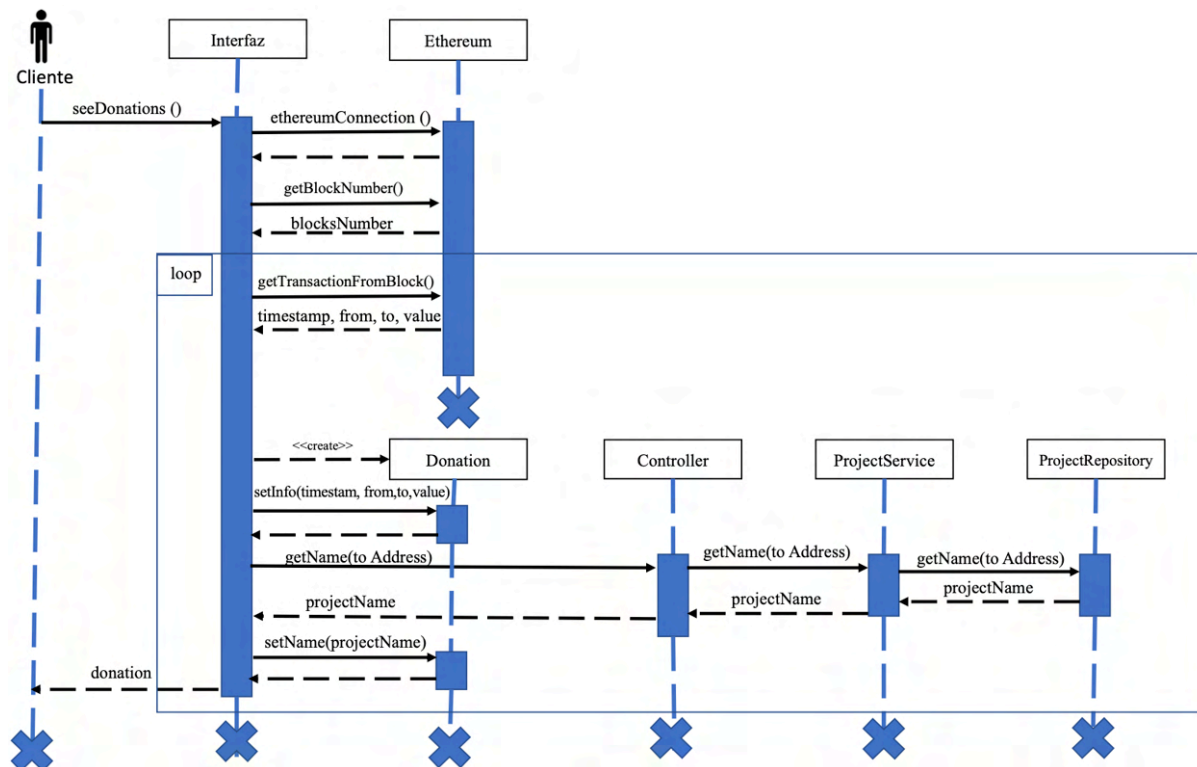


Figura 47. Diagrama de Secuencia - visualización de donaciones

5.2.7 DIAGRAMA DE NAVEGABILIDAD

Por último, se han creado 3 diagramas de navegabilidad, según el rol de usuario que tengan, clientes, organizaciones o administradores. Estos diagramas en forma de árbol, ofrecen una representación visual explícita de los contenidos de la aplicación.

El primer diagrama, Figura 48, muestra el diagrama de navegabilidad de los clientes. Estos pueden desde la página de Home a diferentes recursos. En primer lugar, pueden acceder al inicio de sesión, desde el cual también se puede acceder a la página de creación de cuenta o de recuperación de contraseña.

En segundo lugar, desde Home los clientes podrán también acceder a las donaciones, dentro de las cuales podrán ver las que hayan realizados ellos o las donaciones específicas de un proyecto.

Los clientes tendrán la posibilidad de ver los proyectos y sus categorías, así como una información más detallada de estos, donde tendrán la posibilidad de donar.

Por último, los clientes pueden acceder a las organizaciones existentes en la aplicación y a los proyectos que estén llevando a cabo dichas organizaciones.

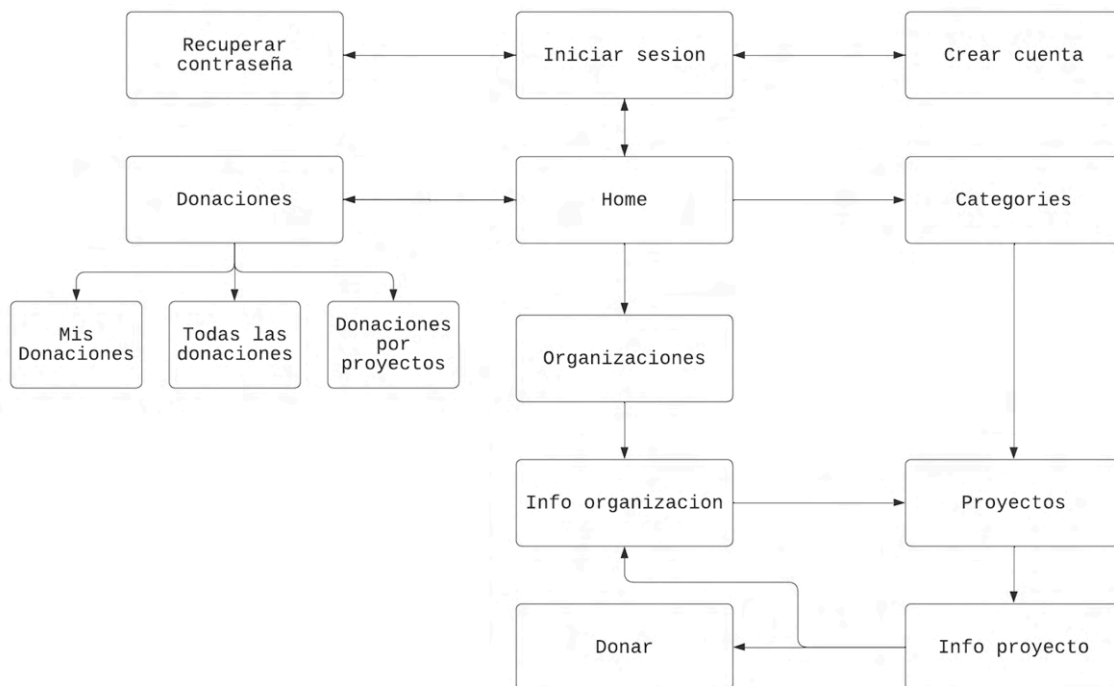


Figura 48. Diagrama de navegabilidad – cliente

El segundo diagrama de navegabilidad, Figura 49, representa los contenidos a los que tendrán acceso las organizaciones. Estas tendrán acceso a los proyectos que estén llevando a cabo, y podrán eliminar o editar dichos proyectos, además crear nuevos proyectos.

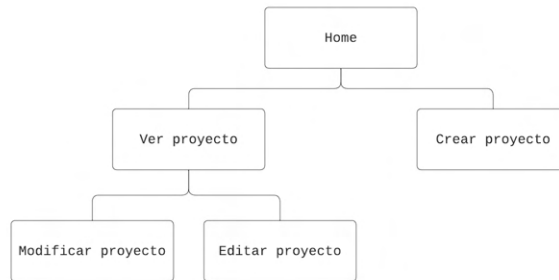


Figura 49. Diagrama de Navegabilidad – organización

El último diagrama de navegabilidad, Figura 50, representa los contenidos a los que tendrán acceso los administradores. Estos podrán crear nuevas organizaciones, así como ver, editar o eliminar las existentes.

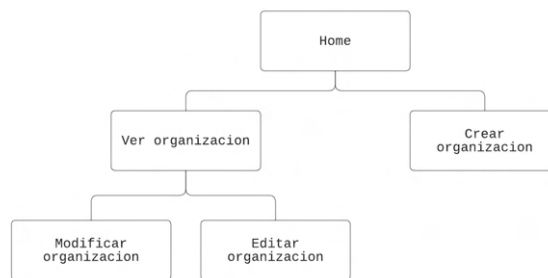


Figura 50. Diagrama de Navegabilidad - administrador

5.3 IMPLEMENTACIÓN

Como se ha explicado anteriormente, los tokens no deben confundirse con las criptomonedas, en este caso los ETH. Los tokens son activos digitales a los que se les atribuye un valor. ERC20 es el protocolo estándar de Ethereum que define las propiedades y funcionalidad de un token. Dentro de estos, encontramos tres grandes categorías, los tokens de utilidad, los tokens de seguridad y los tokens de intercambio. Los primeros dan acceso a un producto actual o futuro y están destinados y diseñados explícitamente para una aplicación, los segundos representan inversiones, y por ellos están sujetos a diversas regulaciones y marcos legales, y los últimos son utilizados como medio de intercambio.

A continuación, se muestra el Smart Contract del token ERC-20 desarrollado para la gestión de donaciones de ONGs de este proyecto.

```
pragma solidity ^0.4.4;

contract Token {

    /// @return total amount of tokens
    function totalSupply() constant returns (uint256 supply) {}

    /// @param _owner The address from which the balance will be retrieved
    /// @return The balance
    function balanceOf(address _owner) constant returns (uint256 balance) {}

    /// @notice send `_value` token to `_to` from `msg.sender`
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transfer(address _to, uint256 _value) returns (bool success) {}

    /// @notice send `_value` token to `_to` from `_from` on the condition it is approved by `_from`
    /// @param _from The address of the sender
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {}

    /// @notice `msg.sender` approves `_addr` to spend `_value` tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @param _value The amount of wei to be approved for transfer
    /// @return Whether the approval was successful or not
    function approve(address _spender, uint256 _value) returns (bool success) {}

    /// @param _owner The address of the account owning tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @return Amount of remaining tokens allowed to spent
    function allowance(address _owner, address _spender) constant returns (uint256 remaining) {}

    event Transfer(address indexed _from, address indexed _to, uint256 _value);
    event Approval(address indexed _owner, address indexed _spender, uint256 _value);
}

contract StandardToken is Token {

    function transfer(address _to, uint256 _value) returns (bool success) {
        if (balances[msg.sender] >= _value && _value > 0) {
            balances[msg.sender] -= _value;
            balances[_to] += _value;
            Transfer(msg.sender, _to, _value);
            return true;
        } else { return false; }
    }

    function transferFrom(address _from, address _to, uint256 _value) returns (bool success) {
        if (balances[_from] >= _value && allowed[_from][msg.sender] >= _value && _value > 0) {
            balances[_to] += _value;
            balances[_from] -= _value;
            allowed[_from][msg.sender] -= _value;
            Transfer(_from, _to, _value);
            return true;
        } else { return false; }
    }

    function balanceOf(address _owner) constant returns (uint256 balance) {
        return balances[_owner];
    }

    function approve(address _spender, uint256 _value) returns (bool success) {
        allowed[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);
        return true;
    }

    function allowance(address _owner, address _spender) constant returns (uint256 remaining) {
        return allowed[_owner][_spender];
    }

    mapping (address => uint256) balances;
    mapping (address => mapping (address => uint256)) allowed;
    uint256 public totalSupply;
}
```

```

contract ERC20Token is StandardToken {
    function () {
        throw;
    }

    string public name;
    uint8 public decimals;
    string public symbol;
    string public version = 'H1.0';

    function ERC20Token() {
        balances[msg.sender] = 10000000;
        totalSupply = 10000000;
        name = "NGO Donation";
        decimals = 0;
        symbol = "PTS";
    }

    function approveAndCall(address _spender, uint256 _value, bytes _extraData) returns (bool success) {
        allowed[msg.sender][_spender] = _value;
        Approval(msg.sender, _spender, _value);

        if(!_spender.call(bytes4(bytes32(sha3("receiveApproval(address,uint256,address,bytes)"))), msg.sender, _value, this, _extraData)) {
            throw;
        }
        return true;
    }
}

```

Figura 51. Smart Contract del Token ERC-20 del proyecto

Este contrato es un conjunto de 6 funciones, que al seguir el estándar ERC-20 puede ser reconocido e identificado por otros Smart Contracts, lo que permite que su uso e interacción sea fluido. Las 4 funcionalidades básicas que se pueden realizar con este contrato son las siguientes:

- Obtener el suministro total
- Obtener el saldo de una cuenta en específico
- Transferir el token de una cuenta a otra
- Aprobar el uso del token como activo monetario

A los usuarios, se les asignará una cuenta de Ethereum, sobre una red simulada, Ganache, en los que dispondrán de una clave pública y otra privada, debiendo esta última ser importada en Metamask.

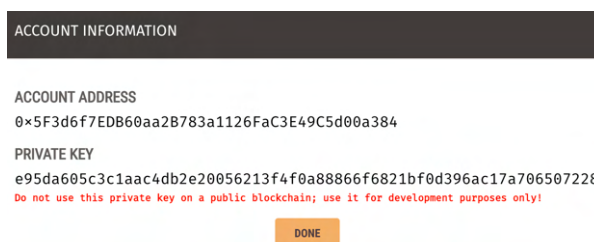


Figura 52. Cuenta de Ethereum de un cliente

El contrato, una vez compilado y desplegado sobre la blockchain, debe ser importado en la billetera de Metamask, con su dirección.

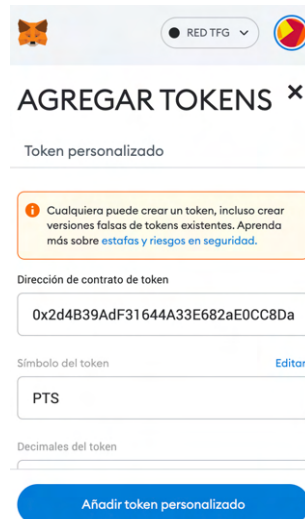


Figura 53. Importación del token ERC-20 en Metamask

Como se ha comentado con anterioridad, Web3 gestiona las conexiones JSON-RPC entre los Smart Contracts y el servidor web, por ello para poder realizar donaciones, en primer lugar, debe configurarse Web3, siendo el proveedor ganache, y en segundo lugar mediante una de las funciones que provee esta librería se configura el contrato. Esta última función, hace uso del ABI que se genera al compilarse el contrato, y de la dirección en la que se almacena al desplegarse. Esta configuración se ilustra en el Código 1.

```
// Initialize Web3
if (typeof web3 !== 'undefined') {
  web3 = new Web3(web3.currentProvider);
} else {
  web3 = new Web3(new Web3.providers.HttpProvider('http://localhost:7545'));
}

// Set the Contract
var contract = new web3.eth.Contract(contractAbi, contractAddress);
```

Código 1. Configuración de Web3 y funciones del contrato

Una vez han sido configurados los parámetros de Web3 y del contrato, se puede acceder a sus métodos para transferir y realizar donaciones, tal y como se indican en el Código 2.

En segundo lugar, el método `web3.eth.getBlock(n)`, a pesar de no obtener el input de la transacción, obtiene otros parámetros, Código 5, entre los que destacan el timestamp de la transacción o el límite de gas.

```

> web3.eth.getBlock(25).then(r => {console.log(r)})
< ▶ Promise {<pending>}
VM427:1
{number: 25, hash: '0x8267304d2f8953c1046b3f710a35d71f804b52104da910eec8f5c4ab8e3e0723',
  parentHash: '0x6bbd2dbb5645b5ce34c14c8edecb1116feba1d3f83feb22b05da6e9ed0d8ea4c', mixHash:
  h: '0x0000000000000000000000000000000000000000000000000000000000000000', nonce: '0x00000
  000000000', ...}
  difficulty: "0"
  extraData: "0x"
  gasLimit: "6721975"
  gasUsed: "36715"
  hash: "0x8267304d2f8953c1046b3f710a35d71f804b52104da910eec8f5c4ab8e3e0723"
  logsBloom: "0x00000000000000000000000000000000000000000000000000000000000000008000000000000000"
  miner: "0x0000000000000000000000000000000000000000000000000000000000000000"
  mixHash: "0x0000000000000000000000000000000000000000000000000000000000000000"
  nonce: "0x0000000000000000"
  number: 25
  parentHash: "0x6bbd2dbb5645b5ce34c14c8edecb1116feba1d3f83feb22b05da6e9ed0d8ea4c"
  receiptsRoot: "0x90c87a9d97c0803fee6ef6ca78b73ef46e94ac15a0ea0aed5f20e9e7471837bc"
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccdd41ad312451b948a7413f0a142fd40d49347"
  size: 1000
  stateRoot: "0x90a7bd573539ed0ba85ae0aea0bb2676cf2dc1d5c84bc161db21d85539d06a8b"
  timestamp: 1656030645
  totalDifficulty: "0"
  ▶ transactions: ['0xb0c0f489b60f5ff89f5e24181ce197c4b39cc5f128049c39dcb57b21406d0b95']
  transactionsRoot: "0x4e6e75f2460af4789dca638d320ba9688b7636d0612865da0e6c67743be95ff4"
  ▶ uncles: []
  ▶ [[Prototype]]: Object
  
```

Código 5. Método 2 para obtener las transacciones de un bloque

Antes de poder decodificar los valores del input, hay que tener en cuenta que a estos se les añade la firma del ABI del contrato, por tanto, en primer lugar, se debe conocer el valor de esta firma. Esta puede ser conocida gracias a la función `web3.eth.encodeFunctionSignature()`, la cual recibe como parámetros el nombre de la función y, el nombre y tipo de las variables.

Una vez conocida la firma, hay que extraérsela al input, tal y como se muestra en el Código 6, para poder así decodificar el input, gracias a la función `web3.eth.abi.decodeParameters()`, y obtener el valor de los parámetros del contrato ejecutado, en este caso `_to` y `_amount`

Capítulo 6. ANÁLISIS DE RESULTADOS

En este capítulo se van a destacar los resultados más relevantes tras el desarrollo del sistema y se va a realizar un análisis crítico de los mismos.

El objetivo principal que ha perseguido este proyecto es la creación de una aplicación web intuitiva para los usuarios, en la que la gestión de las donaciones se lleve a cabo con la tecnología blockchain.

Para la gestión de donaciones como se ha explicado en capítulos anteriores, se ha creado un Smart Contract. El resultado de este contrato es la posibilidad de intercambio de tokens entre diferentes usuarios. La verificación de que el contrato ha sido desplegado correctamente se muestra a continuación en la Figura 54.

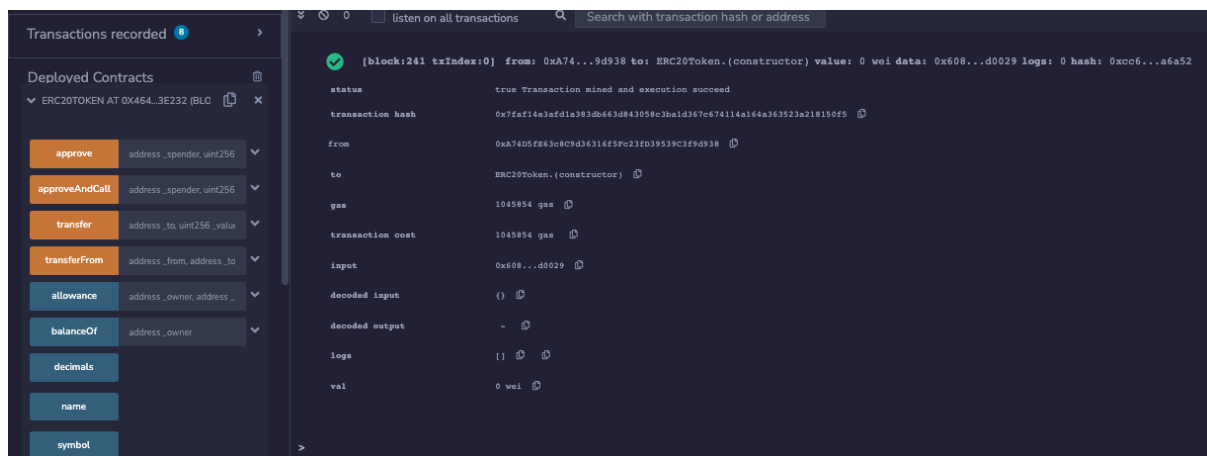


Figura 54. Correcto despliegue del contrato

Además, gracias a Remix IDE, se puede verificar el correcto comportamiento de las funciones implementadas en el contrato. A continuación, en la Figura 55, se muestra la correcta ejecución del método transfer, el cual será utilizado para la realización de donaciones. En la siguiente ejecución, se puede observar la dirección del usuario que ejecuta el contrato, el gas utilizado, el coste de la transacción y, el input que contiene la dirección a la que va destinada la transacción y la cantidad de esta.

```
[block:242 txIndex:0] from: 0xA74...9d938 to: ERC20Token.transfer(address,uint256) 0x464...3e232 value: 0 wei data: 0xa90...00064 logs: 1 hash: 0x148...93b7d

status true Transaction mined and execution succeed
transaction hash 0xe6c634054e7f9bdd0736c3acbef1826b16ba0b4280acb26519d131e64499a32b
from 0xA74D5fE63c8C9d36316f5Fc23fD39539C3f9d938
to ERC20Token.transfer(address,uint256) 0x464DB25a37199f338c8fAb3A15D2FE044093e232
gas 51691 gas
transaction cost 51691 gas
input 0xa90...00064
decoded input {
  "address_to": "0x98d57bdC035407f9F825f7A04c48939BC30cB079",
  "uint256_value": "100"
}
decoded output -
logs [
  {
    "from": "0x464DB25a37199f338c8fAb3A15D2FE044093e232",
    "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
    "event": "Transfer",
    "args": {
      "0": "0xA74D5fE63c8C9d36316f5Fc23fD39539C3f9d938",
      "1": "0x98d57bdC035407f9F825f7A04c48939BC30cB079",
      "2": "100",
      "_from": "0xA74D5fE63c8C9d36316f5Fc23fD39539C3f9d938",
      "_to": "0x98d57bdC035407f9F825f7A04c48939BC30cB079",
      "_value": "100"
    }
  }
]
```

Figura 55. Correcta ejecución del método transfer() del Smart Contract

A continuación, se van a explicar las principales pantallas de navegación de la aplicación.


En primer lugar, se encuentra la página Home, Figura 56, común para todos los usuarios del sistema. En esta pantalla, inicialmente se cargan 5 proyectos, pero posteriormente por medio del botón inferior, ‘See 5 more’, se podrán ir visualizando más. La información mostrada de cada uno de los proyectos, es simplemente una breve descripción, puesto que si el usuario quisiera conocer más acerca de ese proyecto podría hacerlo a través del botón, ‘more’.

En esta pantalla a través de la barra de navegación se podrán filtrar los proyectos por categorías, además de tener acceso a las diferentes organizaciones existentes. Por último, si el usuario no estuviera logeado se tendrá acceso a la página de login, y si hubiera iniciado sesión, el usuario tendría la posibilidad de desconectarse.

Por último, los usuarios encontrarán también a su disposición un botón, ‘transactions’, para poder ver todas las donaciones que se hayan realizado.

NGO Donations
Home Categories Organizations

DONATE SAFELY THANKS TO BLOCKCHAIN




Explore Projects

Organizations

Transactions

All Projects




EDUCATION | INDIA

Provide Gift & food to 700 Street Children

Sahar Education

Under this project, 700 street & poor slum children will be given Nutrition food, Gift, New Dress, Lunch/Dinner, Toys & Happiness. The Program will be quite unique. For poor street...

more




FOOD SECURITY | GUATEMALA

Fight Covid and Climate Driven Hunger

Develop Africa, Inc.

Guatemala has one of highest rates of child malnutrition in the world. Covid and climate change are adding to the problem. Working with community health workers, we combat child...

more




ENVIRONMENT | BRAZIL

Forests4Water Brazil

Reef Check Malaysia

If there's one thing that the pandemic has taught us, it's how interdependent we all are, and how what happens in one place can affect others, thousands of miles away. Take rainforests....

more




HUMAN RIGHTS | IRAQ

Yezidi Children's Safe Space

High Atlas Foundation

FYF provides an environment for children to be safe and enjoy enriching activities, despite their harsh environment. As survivors and witnesses to the 2014 genocide living in protracted...

more



ENVIRONMENT | MALAYSIA

Protect Coral Reefs in Malaysia

Tracambi

Reef Check Malaysia works to conserve Malaysia's ocean and marine ecosystems. Malaysia is part of the Coral Triangle, home to more than 75% of the world's coral species. Our Projects...

more

SEE 5 MORE

Figura 56. Pantalla Home

La segunda pantalla que se va a mostrar, es la página de login del usuario, Figura 57. Tanto administradores, como organizaciones, como clientes podrán iniciar sesión en esta página. Los clientes que aún no se encuentren registrados podrán acceder a la página de signup, o continuar accediendo a los recursos del sistema como 'guests'.

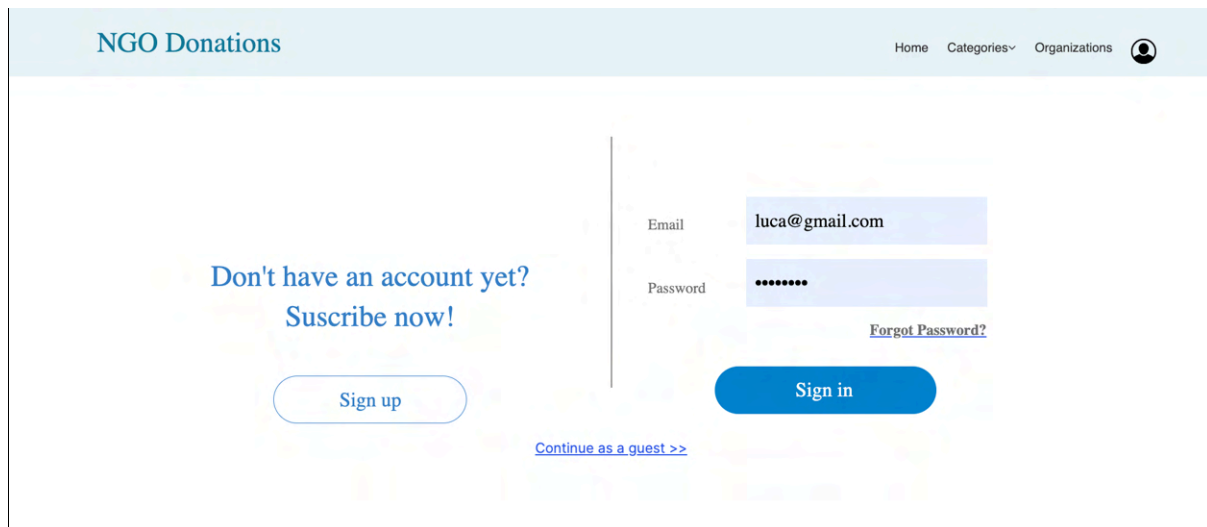


Figura 57. Pantalla login

La siguiente pantalla con la que cuenta el sistema es, la pantalla de organizaciones, Figura 58. Los usuarios podrán ver todas las organizaciones en esta pantalla y si desean conocer más acerca de alguna de estas podrán clicar y ver una información más detallada.

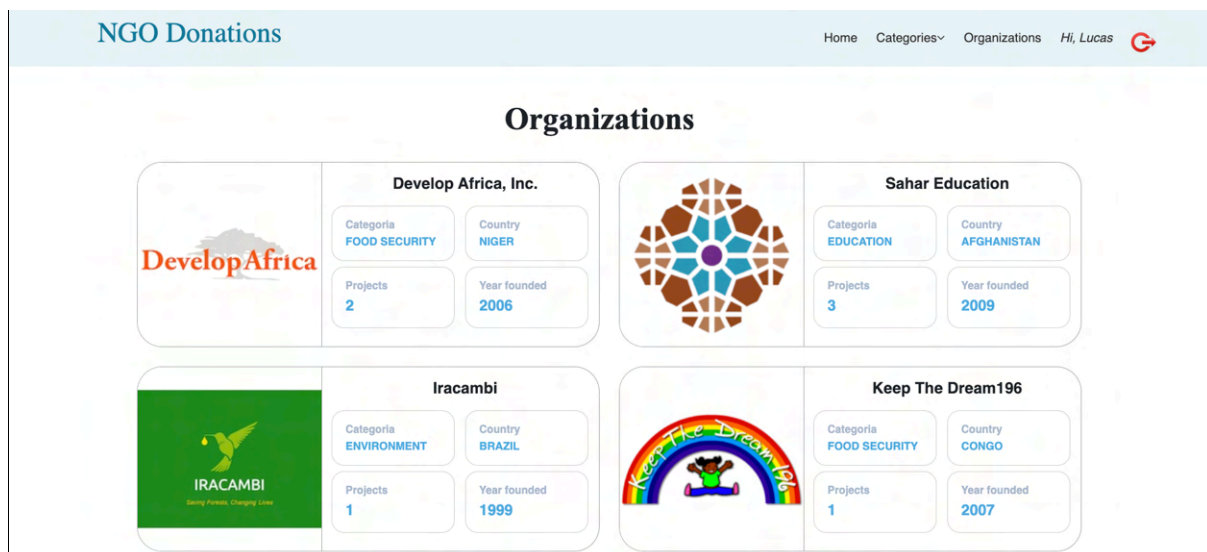



Figura 58. Pantalla organizaciones

En la siguiente pantalla, Figura 59, los usuarios podrán ver una información más detallada de la organización, tales como los gastos e ingresos que estas tienen, además de ver los proyectos lanzados por las mismas. Si el usuario quisiera conocer más sobre la información de algunos de los proyectos que se están llevando a cabo por la organización podría hacer por medio del botón, 'more'.

NGO Donations
Home Categories Organizations Hi, Lucas

Afghan Institute of Learning

Organization Information



Mission: The Afghan Institute of Learning (AIL) works to empower all Afghans who are needy, especially women and children, providing them the knowledge and skills to care for themselves. AIL is expanding access to quality education and healthcare through community based programming, enabling communities to develop the capacity of their people. The goal is to create a foundation of quality education and health systems throughout Afghanistan which meet the needs of local people now and in the future.

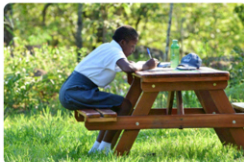
Year founded: 1995

Country: AFGHANISTAN

Categorie: EDUCATION

Total Revenue \$36.8m	Total Expenses \$26.7m	Net Income \$10.1m	Program Expense 81.5%	Administrative Expense 5.45%
Total Assets \$114.8m	Total Liabilities \$29.1m	Net Assets \$85.7k	Fundraising Expense 14.04%	Working Capital 3.21 yrs

Projects (3)




EDUCATION | ZIMBABWE

Build the USAP Community School in Zimbabwe

Education Matters is building a 11th and 12th grade high school for 90 high-achieving, low-income Zimbabwean students to access quality education that values critical thinking,...

Afghan Institute of Learning

[more](#)




EDUCATION | KENYA

Empower, Engage, Educate in Rural Kenya!

Imagine students in rural Kenya using computers and reading books! The Kenya Connect Learning Resource Center engages and empowers students and teachers from our 55 partner schools...

Afghan Institute of Learning

[more](#)



EDUCATION | DOMINICAN REPUBLIC

Offer Hope for 300 Youth

The Bienvenido Project provides holistic care for over 300 impoverished Dominican and Haitian youth in the areas of food, education, health care, advocating for the child's rights...

Afghan Institute of Learning

[more](#)

Figura 59. Pantalla información organización

La siguiente pantalla desarrollada muestra la información completa de un proyecto en concreto. Se puede leer un pequeño resumen del proyecto, así como el desafío que afrontan y la solución que ofrecen, además de un pequeño párrafo con el impacto que tendrá el proyecto a largo plazo.


Aparte se podrá observar un pequeño resumen de la información de la organización que lo está llevando a cabo.

En último lugar los clientes o donantes podrán ver el porcentaje del objetivo a cumplir y acceder a la pantalla de donar, en caso que quisieran realizar una donación.

NGO Donations
Home Categories Organizations Hi, Lucas

EDUCATION | INDIA

Provide Gift & food to 700 Street Children



Summary: Under this project, 700 street & poor slum children will be given Nutrition food, Gift, New Dress, Lunch/Dinner, Toys & Happiness. The Program will be quite unique. For poor street & slum children all this is a dream. But will it be possible. A meal & gifts for 700 poor children life will become a golden memory.

62.50% 7500 PTS

12000 PTS

DONATE

Project Information

Challenge
The future of poor children in India living along the street is in darkness. Due to a lack of education these children have stolen loot, have addictions, have gone the way of crime. Because alienating get them from all over the place. These children need love, happiness, counselling and guidance.

Solution
Under this project, the festival will provide street poor children's dinner, dress, toys and gifts. There will be cutting of cake, face painting, driving electric cars, sharing of gifts, singing and dancing. The kids will have Nutritional and delicious rice, meat, chicken and ice cream in their diet and a soft drink. Dinner/Lunch and gifts for Street poor children life will become a golden memory.

Long-Term Impact
RSKS project "Provide Gift and food to poor children in India" This exposure will enable kids relax, share gifts, share their experiences, and tell stories in a wonderful new environment outside home for one day with the RSKS volunteers. Dinner/lunch and gifts for Street poor children will become a golden memory. Everything we do for these kids now will never be erased from their memories.

Organization Information

Sahar Education

Country: Afghanistan

Category: EDUCATION

Mission:
Sahar creates quality educational opportunities in Afghanistan that empower and inspire children and their families to build peaceful, just and life-affirming communities. Sahar Education is an organization that has been working in Northern Afghanistan since 2001. Sahar's programs, which are designed to empower girls through education, include school building, digital literacy, teacher training, and early marriage prevention.

Total Revenue

\$13.3m

Total Expenses

\$11.7m

Net Income

\$1.6k

Total Assets

\$9.7m

Total Liabilities

\$1.0k

Net Assets

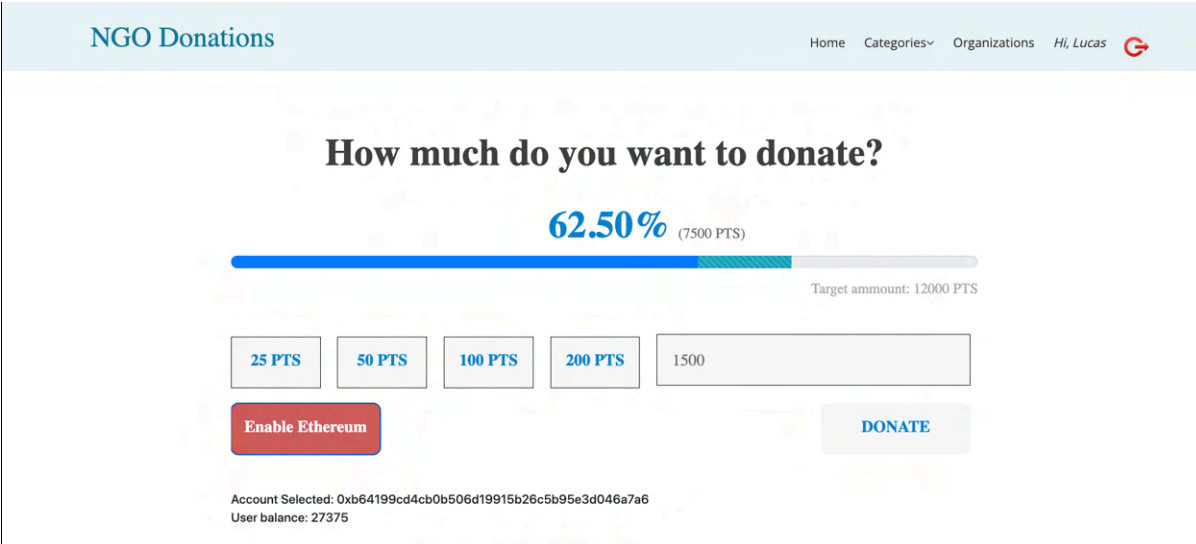
\$8.7m

Figura 60. Pantalla información proyecto

Solo los clientes o donantes tendrán acceso a la siguiente pantalla, Figura 61, cuando deseen realizar una donación. Los donantes podrán seleccionar algunas de las cantidades predeterminadas o podrán introducirla ellos mismos, siendo mostrado cuánto sería el nuevo porcentaje.

Además, los clientes deberán primero conectarse a su cuenta de Ethereum a través de la billetera virtual de Metamask, y una vez estos se han conectados, se mostrará su dirección pública y el saldo del que disponen.

Por último, los usuarios podrán realizar la donación si el valor de la donación introducido es menor que el saldo con el que cuentan.



NGO Donations

Home Categories Organizations Hi, Lucas

How much do you want to donate?

62.50% (7500 PTS)

Target ammount: 12000 PTS

25 PTS 50 PTS 100 PTS 200 PTS 1500

Enable Ethereum DONATE

Account Selected: 0xb64199cd4cb0b506d19915b26c5b95e3d046a7a6
User balance: 27375

Figura 61. Pantalla donación

Los clientes tendrán también acceso a una pantalla para confirmar la transacción en Metamask, tal y como se ilustra en la Figura 62.

En dicha pantalla, los usuarios verán el precio del gas que tendrán que pagar para realizar la donación y el botón para confirmar la transacción y que los fondos sean transferidos.

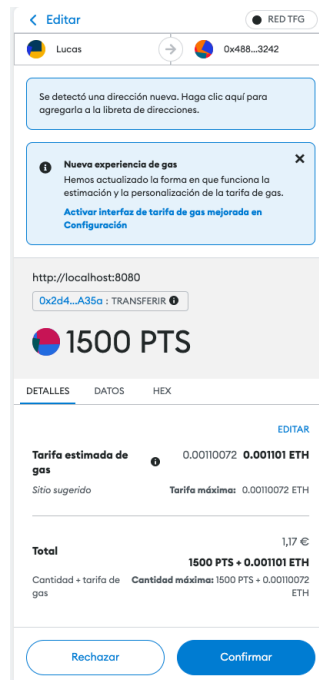


Figura 62. Pantalla confirmar donación en Metamask

Como última pantalla a la que tendrán acceso los clientes encontramos la pantalla de donaciones, Figura 63. En esta pantalla, los usuarios podrán acceder a todas las donaciones realizadas, a las donaciones realizadas por proyectos y, si han iniciado sesión podrán ver también sus donaciones.

NGO Donations							
				Home	Categories	Organizations	Hi, Lucas
TRANSACTION HISTORY							
Donations By Project See All My Donations							
Project Name: Provide Gift & food to 700 Street Children, Goal: 12000PTS, Achieved: 7500PTS							
Date	Amount (PTS)	Project Name	Donor Public Address	Project Public Address			
22/6/2022 18:16:55	2000	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c0B506D19915826c5b95E3D046a7a6	0x4883f1AcaccfAF410029bc58664f32e554F63242			
23/6/2022 14:7:46	100	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c0B506D19915826c5b95E3D046a7a6	0x4883f1AcaccfAF410029bc58664f32e554F63242			
29/6/2022 11:5:16	1500	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c0B506D19915826c5b95E3D046a7a6	0x4883f1AcaccfAF410029bc58664f32e554F63242			
23/6/2022 18:28:12	1350	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c0B506D19915826c5b95E3D046a7a6	0x4883f1AcaccfAF410029bc58664f32e554F63242			
24/6/2022 12:43:34	50	Help Feed Hungry Children in Zimbabwe	0x28549c8B1fA5c07ED38ba67dc3C77317e4f382	0x4883f1AcaccfAF410029bc58664f32e554F63242			
28/6/2022 23:21:21	2500	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c0B506D19915826c5b95E3D046a7a6	0x4883f1AcaccfAF410029bc58664f32e554F63242			
Project Name: Fight Covid and Climate Driven Hunger, Goal: 23500PTS, Achieved: 3500PTS							

Figura 63. Pantalla de donaciones

Otra de las pantallas desarrolladas en la aplicación web es, la pantalla de para crear organizaciones, Figura 64, a la cual solo tendrán acceso los administradores. Una vez el administrador haya completado todos los campos, la organización será subida a la base de datos.

Administration of organizations

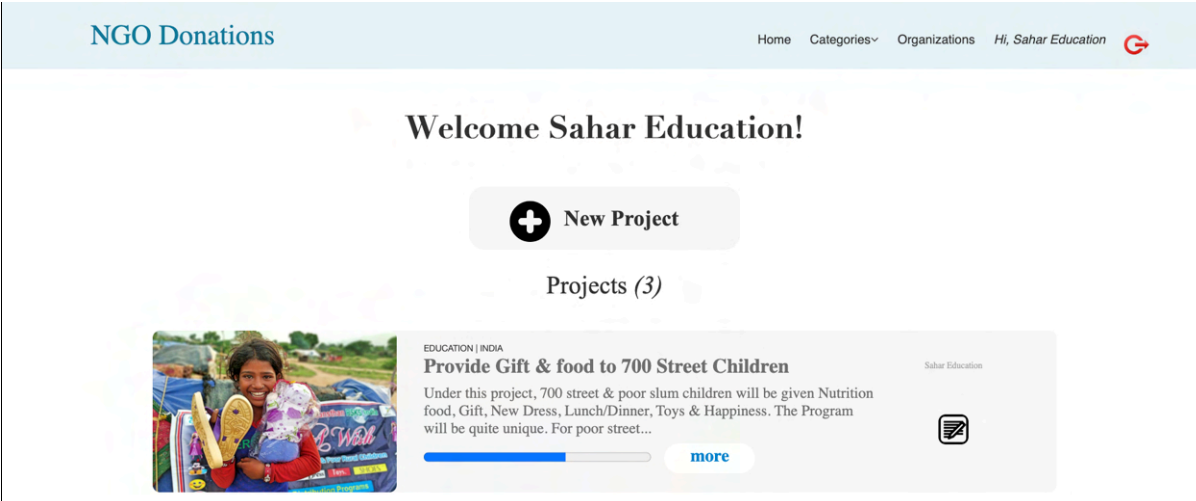
Start a organization

<small>Organization Name</small> <input type="text" value="Name of the organization..."/>	<small>Country</small> <input type="text" value="Country of the organization..."/>
<small>Categorie</small> <input type="text" value="The category of the organization's projects will be..."/>	<small>Year founded</small> <input type="text" value="The year when the organization was founded..."/>
<small>Logo</small> <input type="text" value="Place the organization's logo..."/>	
<small>Email</small> <input type="text" value="Place the organization's email..."/>	
<small>Password</small> <input type="text" value="Place the organization's password..."/>	
<small>Total revenue</small> <input type="text" value="What is the organization's total revenue?"/>	<small>Total assets</small> <input type="text" value="What is the organization's total assets?"/>
<small>Total expenses</small> <input type="text" value="What is the organization's total expenses?"/>	<small>Total liabilities</small> <input type="text" value="What is the organization's total liabilities?"/>
<small>Net income</small> <input type="text" value="What is the organization's net income?"/>	<small>Net assets</small> <input type="text" value="What is the organization's net assets?"/>
<small>Program expenses</small> <input type="text" value="What is the percentage of the organization's program exp"/>	<small>Administrative expenses</small> <input type="text" value="What is the percentage of the organization's administrativ"/>
<small>Fundraising expenses</small> <input type="text" value="What is the percentage of the organization's fundraising t"/>	<small>Working Capital</small> <input type="text" value="How long (in years) would work the organization with the"/>
<small>Mission</small> <input style="height: 40px;" type="text" value="Write a short paragraph describing the mission of the organization..."/>	

Figura 64. Pantalla crear organización

La última pantalla que se va a mostrar, corresponde con la pantalla de las organizaciones, a la cual solo estas tienen acceso, Figura 65. En esta pantalla las organizaciones, en este caso *Sahar Education*, podrá editar o crear un nuevo proyecto, en el cual tendrá que rellenar un formulario parecido al enseñado anteriormente cuando un administrador crea una organización.

Al editar un proyecto la organización solo podrá modificar ciertos atributos.



The screenshot displays the 'NGO Donations' interface. At the top, there is a navigation bar with 'Home', 'Categories', 'Organizations', and 'Hi, Sahar Education'. The main content area features a 'Welcome Sahar Education!' heading, a '+ New Project' button, and a 'Projects (3)' section. The first project listed is 'Provide Gift & food to 700 Street Children' by Sahar Education. It includes a description: 'Under this project, 700 street & poor slum children will be given Nutrition food, Gift, New Dress, Lunch/Dinner, Toys & Happiness. The Program will be quite unique. For poor street...'. A progress bar is visible below the description, and a 'more' button is located to its right.

Figura 65. Pantalla de administración de una organización

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

En este Proyecto Fin de Grado se ha conseguido llevar a cabo una primera aproximación de una aplicación web que gestione las donaciones de ONGs con la tecnología Blockchain.

Las funcionalidades que se marcaron el apartado de Objetivos han sido todas abarcadas, y se han conseguido desarrollar además otras funcionalidades adicionales, con el objetivo de conseguir un sistema más completo.

El sistema que se ha desarrollado por medio del despliegue del Smart Contract de un token en la blockchain de Ethereum, ha permitido la realización de donaciones de una forma segura, fácil y transparente. Estas donaciones las realizan los donantes en el frontend desarrollado que ha tenido como objetivo ser lo más fácil de usar posible ('user-friendly').

Con este trabajo se ha conseguido demostrar que la tecnología blockchain está lista para ser implementada en el ámbito de las organizaciones no lucrativas, y las numerosas ventajas que esta tecnología aporta.

Debido a la limitación de tiempo, quedan como trabajos futuros:

- Mejorar la seguridad del servicio de usuarios, para evitar vulnerabilidades contra amenazas como la modificación o acceso no autorizado. Para ello sería recomendable utilizar el módulo de Spring Security. Este módulo permitiría incorporar seguridad en la aplicación por medio de controles de acceso por URL, actuando como cortafuegos de la aplicación web. Este control permitiría limitar las opciones de peticiones HTTP que pudieran ejecutar los usuarios sobre la aplicación según su rol. Además, este módulo del proyecto de Spring ofrece la posibilidad de codificar la contraseña, por medio de la función hash SHA-256. Se utilizaría el estándar de autorización de APIs, OAuth2.0. Este estándar permite compartir información sin necesidad de tener que compartir la identidad.

- Despliegue de contratos sobre la red de Ethereum. Como se ha comentado anteriormente debido a la limitación económica se ha llevado el despliegue del contrato sobre una red local de pruebas en la red de Ethereum que ofrece Ganache. Por ello, una vez se ha demostrado la viabilidad de utilizar esta tecnología, queda como paso futuro implementarlo dentro de la red de Ethereum o sobre el desarrollado de una blockchain propia.
- Implementar un mayor número de pruebas para el testing. El testing del software es un proceso fundamental durante el desarrollo, por ello se han llevado a cabo distintos tipos de testing, tales como pruebas unitarias, de integridad, funcionales y de humo. Sin embargo, sería conveniente llevar a cabo también otro tipo de pruebas como las pruebas de rendimiento, para medir la velocidad, fiabilidad y escalabilidad de la aplicación, o de aceptación, donde el usuario final tendría que verificar el correcto comportamiento del sistema.
- Actualmente la opción de ver las donaciones por proyectos se visualiza en la página de donaciones, pero queda como trabajo futuro incluir dichas donaciones en la página de información de los proyectos.
- Incluir un sistema de socios dentro del sistema. Actualmente solo se ha desarrollado un sistema para admitir donaciones puntuales, por ello queda como trabajo futuro implementar un sistema/servicio que permita a los usuarios ser socios y realizar donaciones periódicas. Estas cuotas como se ha explicado anteriormente en el capítulo 4, en el apartado de Planificación y Estimación Económica, permitirán realizar la planificación económica, consiguiendo así la estabilidad económica de la aplicación.
- Por último, queda como trabajo a futuro complementar la página web con una aplicación Android/iOS. Para ello sería recomendable el uso del framework de Flutter, puesto que ofrece un atractivo toolkit para el desarrollo de interfaces de usuario para aplicaciones móviles con diferentes sistemas operativos.

Capítulo 8. BIBLIOGRAFÍA

- [1] (s.f.). Obtenido de Endaoment: <https://app.endaoment.org/>
- [2] *5 diferencias entre los Objetivos de Desarrollo del Milenio y los Objetivos de Desarrollo Sostenible* . (s.f.). Obtenido de Unicef: <https://www.unicef.es/noticia/5-diferencias-entre-los-objetivos-de-desarrollo-del-milenio-y-los-objetivos-de-desarrollo>
- [3] AEFr. (6 de Julio de 2021). *¿Cómo son las donaciones a las ONG en España?* . Obtenido de aefundraising: <https://www.aefundraising.org/como-son-las-donaciones-a-las-ong-en-espana/>
- [4] *Alquiler Despachos Madrid* . (s.f.). Obtenido de Centronegocio: <https://centronegocio.net/servicios/alquiler-despachos-madrid.html>
- [5] Arias, E. R. (23 de Diciembre de 2019). *Origen del dinero*. Obtenido de Economipedia: <https://economipedia.com/historia/origen-del-dinero.html>
- [6] *Breve historia del dinero: del trueque al ahorro*. (s.f.). Obtenido de Principal: <https://www.principal.cl/ciclo-vida/breve-historia-del-dinero-0>
- [7] Chandler, A. (s.f.). *¿Qué es un token de seguridad y STO?* . Obtenido de Changelly: <https://changelly.com/blog/es/que-es-un-token-de-seguridad-y-sto/>
- [8] *Claves para entender la tecnología 'blockchain'* . (s.f.). Obtenido de BBVA: <https://www.bbva.com/es/claves-para-entender-la-tecnologia-blockchain/>
- [9] contributors, M. (8 de Diciembre de 2020). *Generalidades del protocolo HTTP* . Obtenido de <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- [10] Cota, E. (2 de Septiembre de 2019). *¿Confiar o no confiar en las ONG? He ahí el dilema*. Obtenido de Resiliente Magazine: <https://resilientmagazine.com/confiar-o-no-confiar-en-las-ong-he-ahi-el-dilema-audidores-sociales-asociados/>
- [11] Cota, E. (2019). *¿Confiar o no confiar en las ONG? He ahí el dilema*. Resiliente Magazine.
- [12] *Create your own Ethereum ERC20-Token* . (7 de Noviembre de 2018). Obtenido de Axom: <https://axom-software.com/en/blog/2019/07/blockchain/1562846735/create-your-own-ethereum-erc20-token>
- [13] Cristina Farras, A. M. (15 de Mayo de 2018). *Del trueque a la criptomoneda: una breve historia del intercambio* . Obtenido de Caixa Bank: <https://www.caixabankresearch.com/es/economia-y-mercados/politica-monetaria/del-trueque-criptomoneda-breve-historia-del-intercambio>
- [14] *Cuántos tipos de blockchain existen* . (s.f.). Obtenido de Bit2me Academy: <https://academy.bit2me.com/cuantos-tipos-de-blockchain->

- [26] González-Páramo, J. M. (2019). *LA DIGITALIZACIÓN DEL DINERO*. Obtenido de https://www.boe.es/biblioteca_juridica/anuarios_derecho/abrir_pdf.php?id=ANU-M-2019-10025700278
- [27] *Historia de las ONG* . (19 de Septiembre de 2018). Obtenido de Ayuda en Accion: <https://ayudaenaccion.org/blog/solidaridad/historia-de-las-ong/#:~:text=C%C3%B3mo%20nacieron%20las%20ONG&text=El%20t%C3%A9rmino%20Organizaciones%20No%20Gubernamentales,lucha%20por%20el%20sufragio%20femenino> .
- [28] *Historia y evolución del dinero*. (10 de Septiembre de 2019). Obtenido de Creditea: <https://www.creditea.es/blog/historia-y-evolucion-del-dinero/#:~:text=El%20dinero%20es%20todo%20activo,con%20los%20primeros%20asentamientos%20humanos>.
- [29] *How AJAX works?* . (s.f.). Obtenido de javaTpoint: <https://www.javatpoint.com/how-ajax-works>
- [30] Joshua. (25 de Enero de 2022). *LENGUAJES DE CONTRATO INTELIGENTE* . Obtenido de Ethereum: <https://ethereum.org/es/developers/docs/smart-contracts/languages/>
- [31] Kandefer, K. (28 de Octubre de 2021). *Datos completos de los sueldos de los desarrolladores de la cadena de bloques* . Obtenido de DevSkiller: <https://devskiller.com/es/blockchain-developer-salary/>
- [32] *La reacción del Tercer Sector Social al entorno de crisis*". (Septiembre de 2019). Obtenido de ESADE y fundacion "la Caixa": <https://prensa.fundacionlacaixa.org/wp-content/uploads/2019/09/36796.pdf>
- [33] Lasa, A. (15 de Junio de 2019). *Introducción a Web3js (parte 1)* . Obtenido de Medium: <https://albertolasa.medium.com/introducci%C3%B3n-a-web3js-en-ethereum-e9c0b20a2013>
- [34] López, J. (s.f.). *Las distintas generaciones de Blockchain*. Obtenido de AudiTech: <https://auditech.es/las-distintas-generaciones-de-blockchain/>
- [35] MICHAEL J. FISCHER, N. A. (1985). *Impossibility of Distributed Consensus with One Faulty Process*. Obtenido de <https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf>
- [36] Moreno, S. (Mayo de 2016). *Situación Actual de las ONG en España*. Obtenido de Fundación Lealtad: https://www.fundacionlealtad.org/wp-content/uploads/2016/05/Situacion-actual-ONG_web.pdf
- [37] *Objetivos de Desarrollo del Milenio: se cumplen 3 metas antes de finalizar 2015* . (s.f.). Obtenido de Unicef: <https://www.unicef.es/noticia/objetivos-de-desarrollo-del-milenio-se->

- <https://javarevisited.blogspot.com/2017/10/differences-between-requestparam-and-pathvariable-annotations-spring-mvc.html#axzz7WO4745NC>
- [48] Pilehchiha, S. (23 de Mayo de 2022). *ERC-20 TOKEN STANDARD* . Obtenido de Ethereum: <https://ethereum.org/en/developers/docs/standards/tokens/erc-20/>
- [49] *Precios de AWS Amplify* . (s.f.). Obtenido de Amazon Web Services, AWS: <https://aws.amazon.com/es/amplify/pricing/>
- [50] *Qué es un 'token' y para qué sirve* . (23 de Diciembre de 2021). Obtenido de BBVA: <https://www.bbva.com/es/que-es-un-token-y-para-que-sirve/>
- [51] RAMÍREZ, L. (28 de Marzo de 2022). *¿Qué es un token y qué tipos existen?* . Obtenido de Iebs: <https://www.iebschool.com/blog/tipos-de-token-que-es-finanzas/>
- [52] *Redes de Criptomonedas ¿Qué tipos hay?* . (s.f.). Obtenido de TradeWithUs: <http://crypto.tradew.us/educacion-capacitacion-conocimiento-universidad-avanzado-curso-gratis/tecnologia-blockchain/redes-de-criptomonedas-que-tipos-hay/>
- [53] *Revisando las tres generaciones de la Tecnología Blockchain* . (s.f.). Obtenido de Kevin Melgarejo: <https://kevinmelgarejo.com/revisando-las-tres-generaciones-de-la-tecnologia-blockchain/>
- [54] Rodriguez, A. (20 de Septiembre de 2019). *¿Qué es un algoritmo de consenso?* . Obtenido de Medium: <https://adr-rod87.medium.com/qu%C3%A9-es-un-algoritmo-de-consenso-ea1d8e99f348>
- [55] Rodriguez, N. (20 de Septiembre de 2018). *Algoritmos De Consenso: La Raíz De La Tecnología Blockchain* . Obtenido de 101 Blockchain: <https://101blockchains.com/es/algoritmos-de-consenso-blockchain/>
- [56] Rodriguez, N. (4 de Septiembre de 2018). *La Mejor Guía Sobre La Tecnología Blockchain: Una Revolución Para Cambiar El Mundo* . Obtenido de 101 Blockchains: <https://101blockchains.com/es/tecnologia-blockchain/#prettyPhoto>
- [57] RODRÍGUEZ, P. M. (12 de Abril de 2019). *INTRODUCCIÓN A BLOCKCHAIN: ¿QUÉ ES Y CÓMO HA EVOLUCIONADO?* . Obtenido de Techedge: <https://www.techedgegroup.com/es/blog/introduccion-blockchain-explicacion-evolucion>
- [58] Santaella, J. (30 de Mayo de 2022). *¿Qué son los token ERC-20 de Ethereum y por qué son tan útiles?* Obtenido de Economía3: <https://economia3.com/token-erc20-ethereum-que-es-criptomonedas/>
- [59] *Secp256k1*. (10 de Diciembre de 2012). Obtenido de Bitcoin Wiki: <https://es.bitcoin.it/wiki/Secp256k1>

- [60] Simões, C. (24 de Marzo de 2022). *¿Qué es la criptografía? ¿Cuál es su papel en Blockchain?* . Obtenido de Itdo: <https://www.itdo.com/blog/que-es-la-criptografia-cual-es-su-papel-en-blockchain/>
- [61] *Smart contracts, ¿qué son y para qué sirven?* . (27 de Mayo de 2022). Obtenido de Santander: <https://www.santander.com/es/stories/smart-contracts>
- [62] *'Smart contracts': contratos inteligentes para formalizar acuerdos en la era digital* . (s.f.). Obtenido de Iberdrola: <https://www.iberdrola.com/innovacion/smart-contracts>
- [63] *Smart Contracts: ¿Qué son, cómo funcionan y qué aportan?* . (s.f.). Obtenido de Bit2me: <https://academy.bit2me.com/que-son-los-smart-contracts/>
- [64] wendy.tong. (18 de Enero de 2022). *La criptografía Blockchain: la columna vertebral de la seguridad Blockchain* . Obtenido de Phemex: <https://phemex.com/es/academy/la-criptografia-blockchain>

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

En el año 2000, 189 países se reunieron para desarrollar los Objetivos de Desarrollo del Milenio (ODM), en la sede central de las Naciones Unidas. Estos objetivos, Figura 66, nacieron con el propósito de ser alcanzados para el año 2015.

Según el Informe ODM 2012²⁰, 3 importantes objetivos, la erradicación de pobreza, el acceso al agua potable y la mejora de vida en los barrios marginales fueron alcanzados tres años antes de la fecha establecida.

Según los datos ofrecidos en dicho informe el porcentaje de personas con menos de 1.25\$ al día se vio reducido en más de la mitad, entre 1990 y 2010. También se menciona en el informe haber alcanzado la meta en 2010 de reducir en la mitad el porcentaje de personas sin acceso a fuente de agua. Por último, en 2012 se redujo de un 39% a un 33% el porcentaje de habitantes que viven en barrios marginales. También se señala que se consiguió la igualdad en la educación primaria entre niños y niñas.



Figura 66. 8 Objetivos de Desarrollo del Milenio

20 [37] (Objetivos de Desarrollo del Milenio: se cumplen 3 metas antes de finalizar 2015 , s.f.)

El alcance de los logros de los ODM había sido desigual, la lucha contra el cambio climático, la desertificación o la pérdida de biodiversidad había sido mínimo, por ello, en septiembre de 2015, 193 países miembros de Naciones Unidas aprobaron la nueva Agenda 2030. Esta agenda tiene recogidos 17 Objetivos de Desarrollo Sostenible²¹, Figura 67, con año límite 2030.



Figura 67. 17 Objetivos de Desarrollo Sostenible

Estos 17 objetivos tienen como reto alcanzar de forma sostenible el desarrollo para todas las personas. Los 5 principales cambios más significativos con respecto a los ODM, tal y como recoge Unicef²², son los siguientes:

- Sostenibilidad, seguir un modelo sostenible.
- Equidad, enfoque con parámetros que reflejan mejor la realidad.

21 [44] (Objetivos de Desarrollo Sostenible, s.f.)

22 [2] (5 diferencias entre los Objetivos de Desarrollo del Milenio y los Objetivos de Desarrollo Sostenible, s.f.)

- Universalidad, metas enfocadas no solo a países en desarrollo.
- Compromiso, todos los países estarán comprometidos con los ODS.
- Alcance, los ODM tenían 21 metas mientras los ODS 169 metas que alcanzar.

Tal y como se ilustra en la Figura 67, los ODS se pueden agrupar en 5 grupos, conocidos como las '5 P', Prosperidad (Prosperity), Planeta (Planet), Personas (People), Paz (Peace), Cooperación (Partnership).

Este proyecto, al colaborar con ONGs cuyas misiones o fines abarcan todos los 17 objetivos, contribuye en gran medida a la consecución de estos. Por esto, uno de los grandes objetivos de este proyecto ha sido reducir los gastos de mantenimiento y publicidad que han tenido hasta hoy en día las ONGs, pudiendo así colaborar en mayor medida con todos estos objetivos propuestos.

Además, este proyecto tiene como objetivos principales el número 9, Industria, innovación e infraestructura, el 16, paz, justicia e instituciones sólidas, el 12, Producción y consumo responsables y el 8, trabajo decente y crecimiento económico. Todo esto gracias al uso de la tecnología blockchain, que además de aumentar la transparencia y trazabilidad de las donaciones, también al hacer uso de redes descentralizadas reduce gastos de infraestructura.

ANEXO II: MANUAL DE INSTALACIÓN

Se desarrolla en el presente Anexo un manual de instalación con el objetivo que todo el mundo pueda probar la aplicación web desarrollada en este Trabajo de Fin Grado.

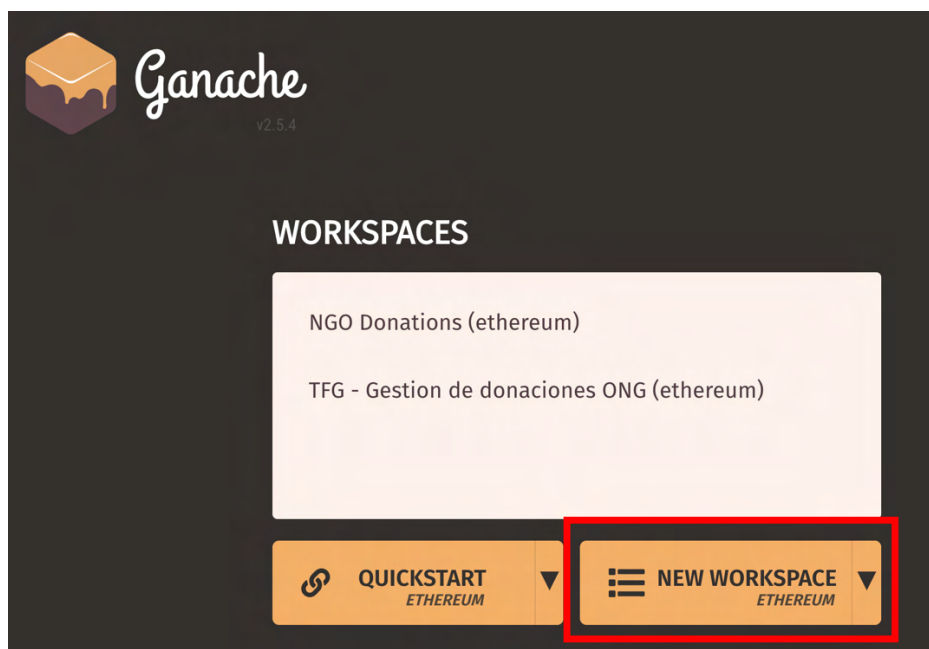
Se desarrolla en el presente Anexo un manual de instalación con el objetivo que todo el mundo pueda probar la aplicación web desarrollada en este Trabajo de Fin Grado.

CONFIGURACIÓN GANACHE

En primer lugar, es necesario instalarse Ganache para poder desplegar los contratos y llevar a cabo el testeo de la aplicación.

Link de descarga de Ganache: <https://trufflesuite.com/ganache/>

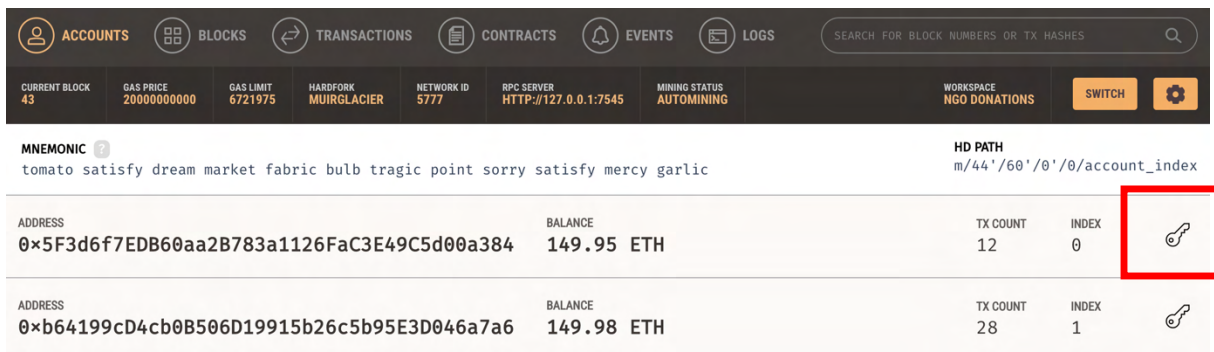
Una vez se haya instalado ganache se deberá crear un nuevo workspace.



Se deberá configurar las secciones, 'SERVER' y 'ACCOUNTS & KEYS' tal y como se indican en las siguientes imágenes.

<p>SERVER</p> <p>HOSTNAME <input type="text" value="127.0.0.1 - lo0"/></p> <p>PORT NUMBER <input type="text" value="7545"/></p> <p>NETWORK ID <input type="text" value="5777"/></p>	<p>ACCOUNTS & KEYS</p> <p>ACCOUNT DEFAULT BALANCE <input type="text" value="150"/></p> <p>TOTAL ACCOUNTS TO GENERATE <input type="text" value="99"/></p>
---	---

Una vez se haya creado el nuevo workspace, este se encontrará disponible con las mismas cuentas de Ethereum siempre que lo necesitemos. Además, podremos acceder desde allí a las claves privadas de las cuentas para poder importarlas posteriormente en Metamask.



The screenshot shows a dashboard with various metrics and a table of accounts. The table has columns for ADDRESS, BALANCE, TX COUNT, and INDEX. A red box highlights a key icon in the rightmost column of the first row.

ADDRESS	BALANCE	TX COUNT	INDEX	
0x5F3d6f7EDB60aa2B783a1126FaC3E49C5d00a384	149.95 ETH	12	0	
0xb64199cD4cb0B506D19915b26c5b95E3D046a7a6	149.98 ETH	28	1	

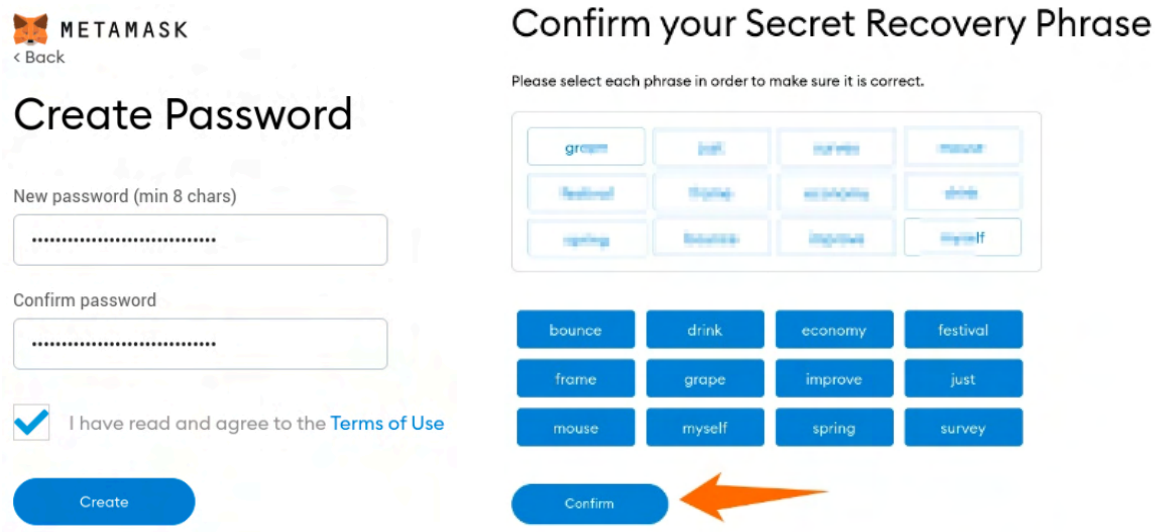
CONFIGURACIÓN METAMASK

En segundo lugar, en caso de no tener instalado el navegador de Google Chrome se deberá instalar para poder hacer uso de la extensión de Metamask, cuyo link se adjunta a continuación.

Link Metamask: <https://metamask.io/download/>

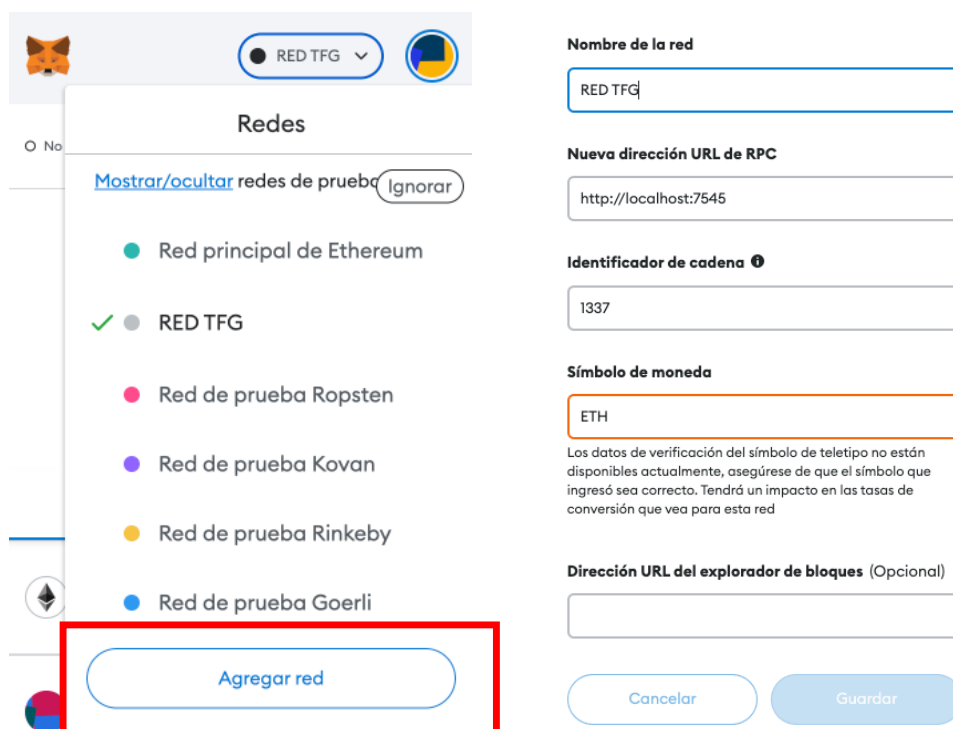
Una vez se tiene acceso a dicha extensión, se deberá Importar una billetera o en caso contrario crear una nueva. En caso de no disponer se deberá escoger la opción de crear billetera, y posteriormente se podrá elegir o no compartir algunos datos de manera anónima con Metamask.

Un momento crucial en la creación de una billetera de Metamask es a la hora de crear una contraseña. Se deberá introducir una contraseña, y posteriormente Metamask te hará crear una frase secreta de recuperación con unas palabras aleatorias que será fundamental no perder y almacenarlo de forma segura.

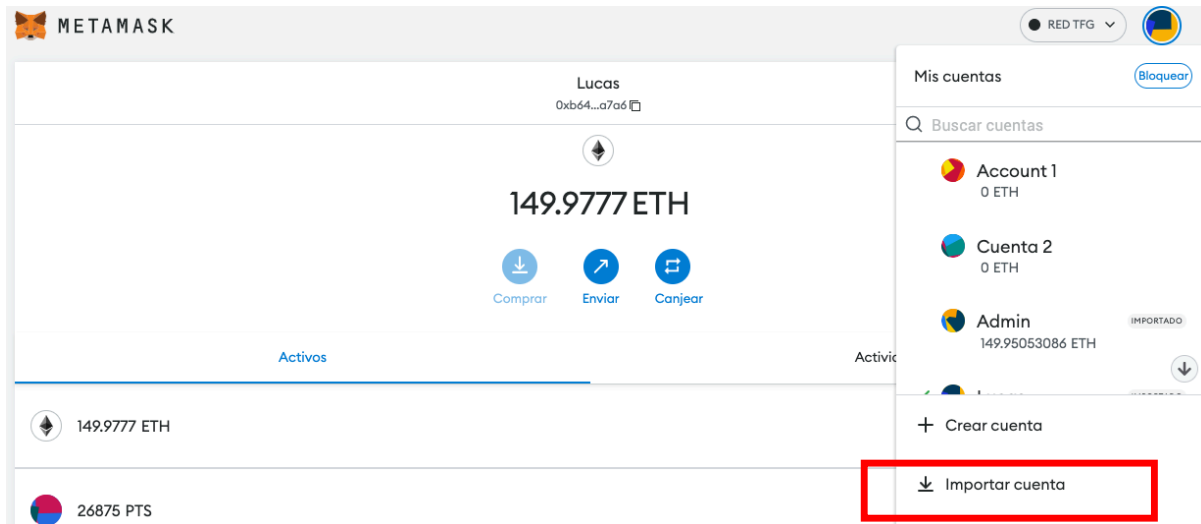


Una vez establecida la contraseña y la frase de recuperación, Metamask se encontrará disponible para ser utilizada.

Para este proyecto será necesario agregar en Metamask una nueva red, para sincronizar la red local de Ganache que se ha creado con anterioridad. Se deberá introducir el puerto que se ha indicado previamente en Ganache.



Una vez se ha creado la red sincronizada con Ganache, se deberá proceder a la importación de las cuentas de esta red simulada.



Metamask pedirá que se introduzca la clave privada de la cuenta que se quiere importar, la cual se ha indicado anteriormente como obtenerla.

Importar cuenta

Las cuentas importadas no se asociarán con la frase secreta de recuperación de la cuenta original de MetaMask. Más información sobre las cuentas importadas [aquí](#)

Seleccionar tipo

Pegue aquí la cadena de clave privada:

Cancelar

Importar

ACCOUNT INFORMATION

ACCOUNT ADDRESS

0xb64199cD4cb0B506D19915b26c5b95E3D046a7a6

PRIVATE KEY

4b47c27a54a054984ec6f5d1b516f4d76439ca5a314ec2fadc5df799c9465f

Do not use this private key on a public blockchain; use it for development purposes only!

DONE

Una vez configurado ganache, Metamask e importado las cuentas correspondientes, se procederá al despliegue del Smart Contract del Token.

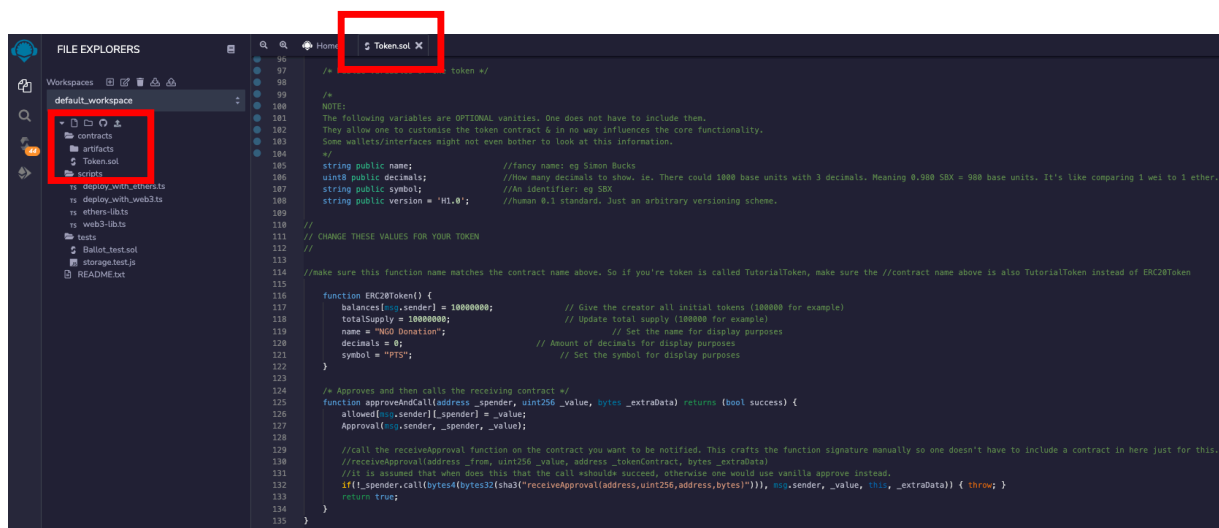
DESPLIEGUE SMART CONTRACT

Para el despliegue del Smart Contract se recomienda utilizar Remix, IDE online que facilita la compilación y migración del contrato sobre Ethereum.

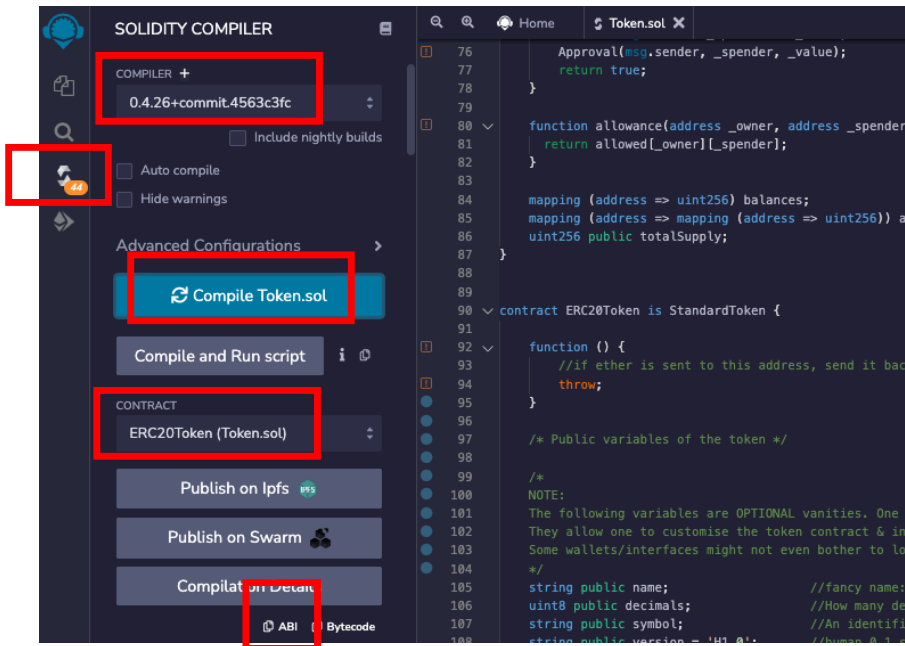
URL Remix: <https://remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.7+commit.e28d00a7.js>

Smart Contract: <https://github.com/luca1015/NGO-Donations/tree/master/src/contracts>

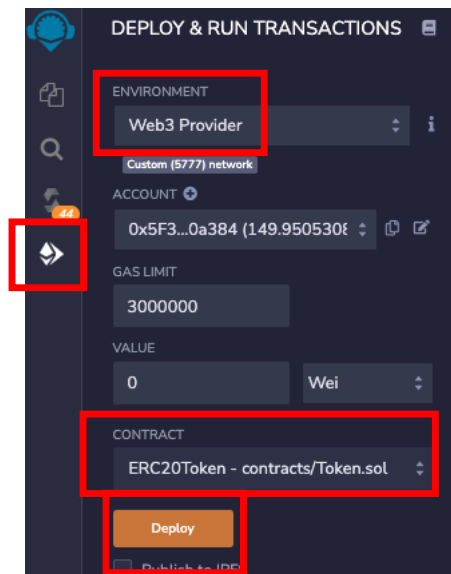
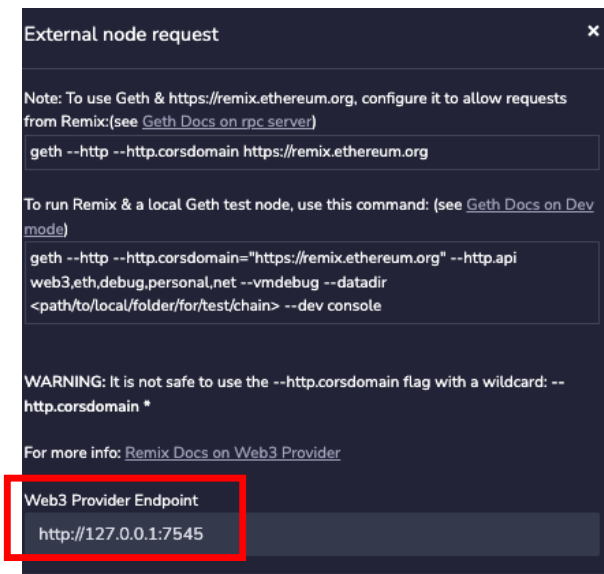
En primer lugar, se debe copiar el Smart Contract adjuntado, pegarlo en la carpeta de contracts de Remix y abrir dicho archivo.



Posteriormente se debe elegir el contrato a compilar y la versión del compilador, 0.4.26. Una vez compilado se debe copiar el ABI, para posteriormente ser capaces de acceder a las funciones del contrato.



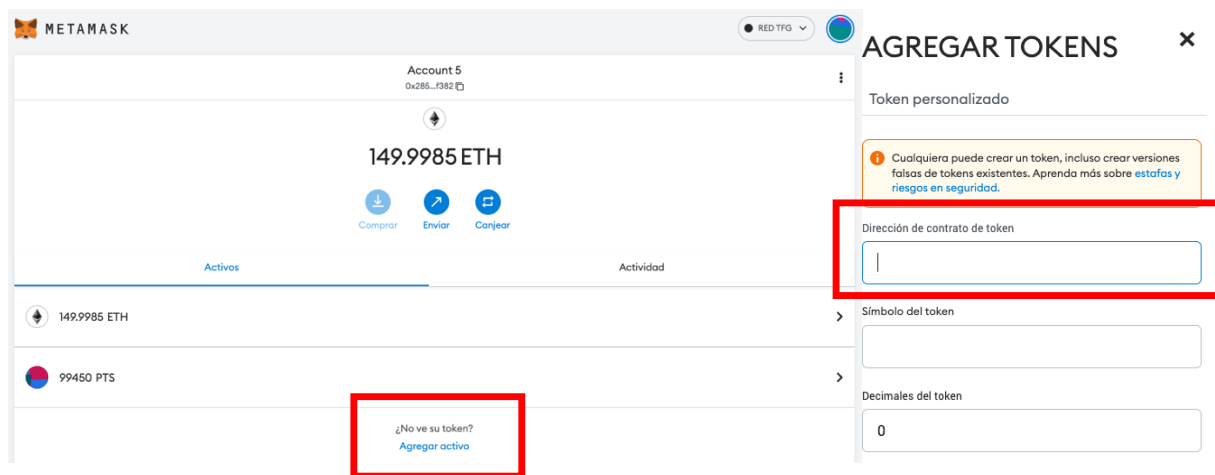
Remix también ofrece la posibilidad desplegar el contrato en la red de pruebas de Ethereum creada en Ganache. Para esto, se debe elegir en ‘ENVIRONMENT’ la opción de Web3 Provider, poner como endpoint la dirección IP de la red en ganache, con el número de puerto correspondiente y como último paso clicar en ‘DEPLOY’. Para realizar esto se debe tener abierto el workspace creado en Ganache.



Una vez desplegado el contrato se obtendrá una dirección la cual se usará para poder acceder al contrato en el backend. Además, se podrá comprobar el funcionamiento del mismo en el IDE de Remix.

TOKEN DESPLEGADO

El contrato desplegado da como resultado la creación de un Token. Dicho token debe ser importado en la billetera de Metamask a través de la dirección que se ha obtenido al desplegarlo.



Por último, se deberá modificar el archivo javascript, Contract.js (una vez se haya clonado el proyecto de GitHub). Cuando se define la variable contract, esta recibe dos variables en la función utilizada, contractABI y contractAddress. Estas dos variables deberán ser redefinidas con los nuevos valores, el ABI del contrato al compilarse y la address generada al desplegarse.

```
// Set the Contract  
var contract = new web3.eth.Contract(contractAbi, contractAddress);
```

BASE DE DATOS

Como bien se ha explicado durante la memoria, la gestión de usuarios, proyectos y organizaciones se va a llevar a cabo en un base de datos. La base de datos que se ha escogido y de la que se ha programado todas las configuraciones para su acceso ha sido PostgreSQL.

Por ello es necesario instalarse una versión superior o igual a Postgres SQL 12.4. Una vez se ha instalado PostgreSQL se debe importar la base de datos, que se denomina NGODonations.sql, y se encuentra en la siguiente dirección:

URL Database: <https://github.com/luca1015/NGO-Donations>

Se aconseja realizar la importación de la base de datos a través del framework PgAdmin4. El proceso para importar la base es sencillo y se detalla a continuación:

- Cree una base de datos nueva (dentro del apartado Base de Datos) que se llame: NGO Donations.
- Haga click derecho encima de la base de datos y pulse donde aparezca Restaurar/Restore, en el nombre del archivo debe poner la ruta de la base de datos a importar, y en nombre del rol debe seleccionar postgres.
- Por último, haga click en restaurar y la base de datos se habrá guardado correctamente. (Compruebe que tenga las siguientes tablas: users, projects, organizations).

CLONACIÓN DEL PROYECTO

El primer paso antes de clonar el proyecto, es asegurarse que se tiene la versión de Java 14 instalado, o en su defecto una versión superior.

En segundo lugar, se procede a la clonación del proyecto ubicado en GitHub sobre el IDE que mejor le parezca, para el desarrollo de este trabajo se ha utilizado IntelliJ.

Proyecto en GitHub: <https://github.com/luca1015/NGO-Donations>

Por último, se debe tener en cuenta que el archivo de contract.js debe ser modificado con la nueva dirección y ABI del Smart Contract.

ANEXO II: MANUAL DE USUARIO

En este Anexo se ha desarrollado un manual de usuario, con el fin que estos puedan visualizar la interacción con el sistema y las distintas pantallas a las que tiene acceso.

Existen tres tipos de acceso, cliente, organización y administrador. En función del rol de usuario, las pantallas y acceso a recursos variará.

El usuario deberá tener instalado los siguientes dos elementos:


- Navegador web de Google Chrome
- Extensión Metamask

El acceso a la aplicación será por medio de la siguiente URL: <http://localhost:8080/>.

A continuación, se procede a mostrar las pantallas de la aplicación web.

- **Vista Administrador**

En primer lugar, el administrador visualizará las organizaciones existentes, pudiendo crear nuevas.





NGO Donations Home Categories Organizations Hi, admin

Administration of organizations

+ New Organization

Organizations (8)

Organization Name	Logo	Category	Country	Projects	Year founded
Develop Africa, Inc.		FOOD SECURITY	NIGER	2	2006
Sahar Education		EDUCATION	AFGHANISTAN	3	2009

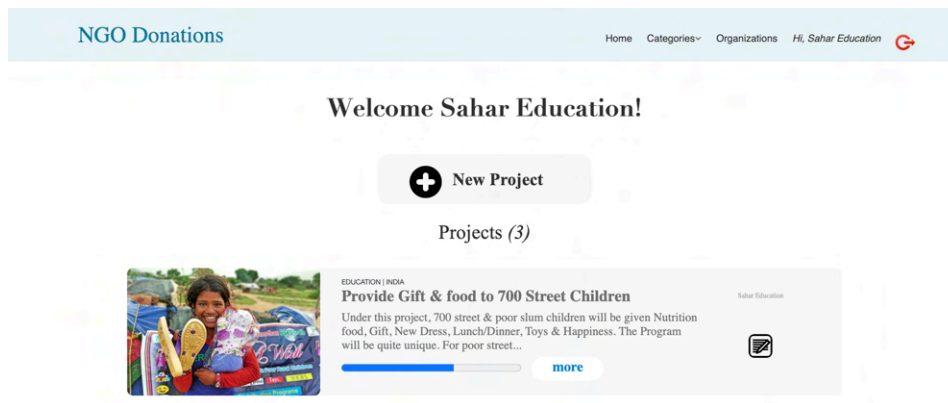
El administrador podrá crear una organización al completar los datos de la siguiente pantalla:

Start a organization

<p>Organization Name</p> <input type="text" value="Name of the organization..."/>	<p>Country</p> <input type="text" value="Country of the organization..."/>
<p>Categorie</p> <input type="text" value="The category of the organization's projects will be..."/>	<p>Year founded</p> <input type="text" value="The year when the organization was founded..."/>
<p>Logo</p> <input type="text" value="Place the organization's logo..."/>	
<p>Email</p> <input type="text" value="Place the organization's email..."/>	
<p>Password</p> <input type="text" value="Place the organization's password..."/>	
<p>Total revenue</p> <input type="text" value="What is the organization's total revenue?"/>	<p>Total assets</p> <input type="text" value="What is the organization's total assets?"/>
<p>Total expenses</p> <input type="text" value="What is the organization's total expenses?"/>	<p>Total liabilities</p> <input type="text" value="What is the organization's total liabilities?"/>
<p>Net income</p> <input type="text" value="What is the organization's net income?"/>	<p>Net assets</p> <input type="text" value="What is the organization's net assets?"/>
<p>Program expenses</p> <input type="text" value="What is the percentage of the organization's program exp"/>	<p>Administrative expenses</p> <input type="text" value="What is the percentage of the organization's administrati"/>
<p>Fundraising expenses</p> <input type="text" value="What is the percentage of the organization's fundraising i"/>	<p>Working Capital</p> <input type="text" value="How long (in years) would work the organization with the"/>
<p>Mission</p> <input type="text" value="Write a short paragraph describing the mission of the organization..."/>	
<p>Create Organization</p>	

- **Vista Organización**

Las organizaciones tendrán acceso a la lista de proyectos que estén llevando a cabo y podrán editar los mismos, además de crear nuevos.

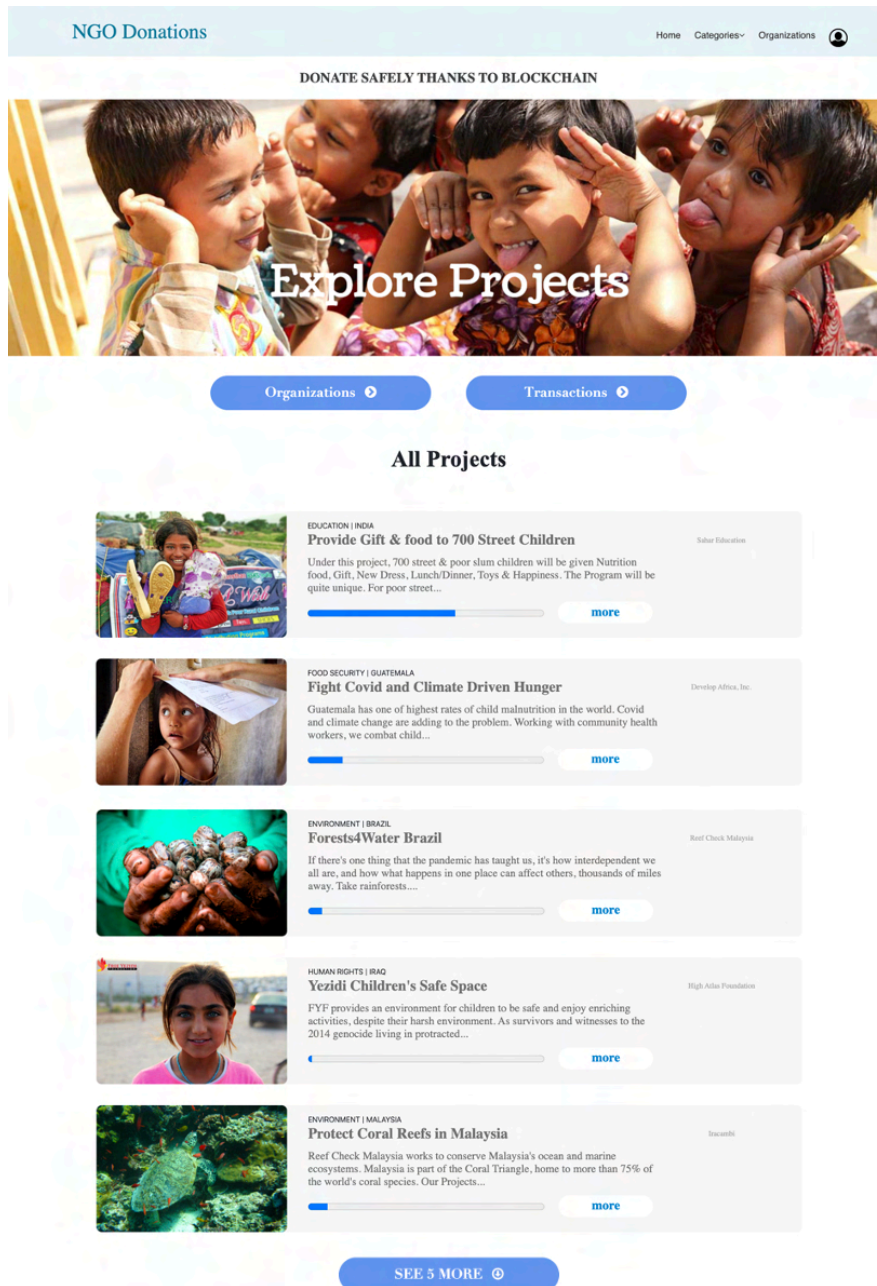


Edit Provide Gift & food to 700 Street Children

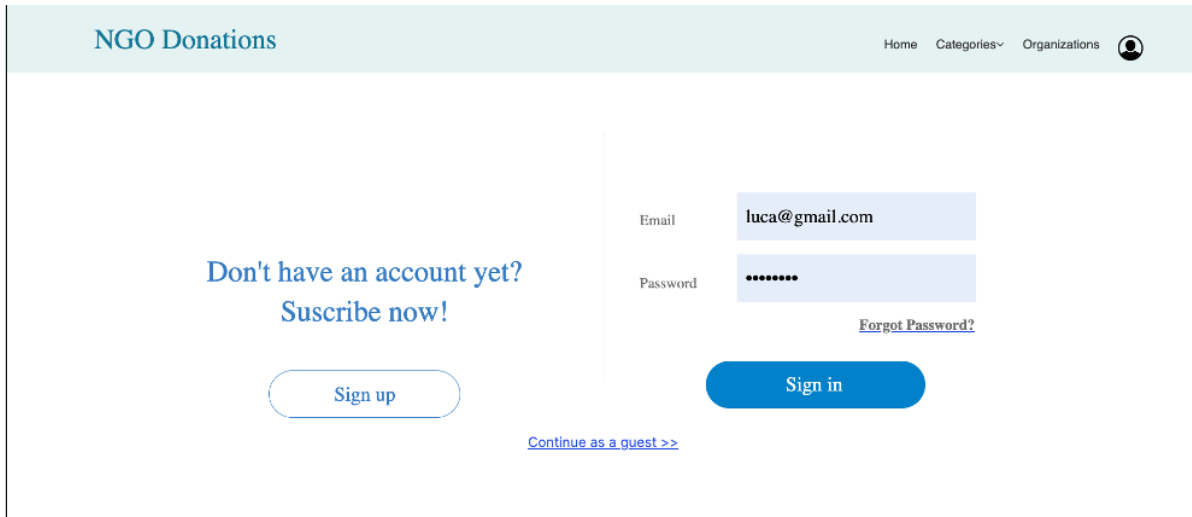
Project Name	Country
<input type="text" value="Provide Gift & food to 700 Street Children"/>	<input type="text" value="INDIA"/>
Public Address	
<input type="text" value="0x4883f1Acacaf4f410029be58664f32e554Fe3242"/>	
Status	Creation Date
<input type="text" value="Not Finished"/>	<input type="text" value="2022-06-22T08:58:33.577Z"/>
Categorie	Goal (PTS)
<input type="text" value="EDUCATION"/>	<input type="text" value="12000"/>
Summary	
<input type="text" value="Under this project, 700 street & poor slum children will be given Nutrition food, Gift, New Dress, Lunch/Dinner, Toys & Happiness. The Program will be quite unique. For poor street & slum children all this is a dream. But will it be possible. A meal & gifts for 700 poor children life will become a golden memory."/>	
Challenge	
<input type="text" value="The future of poor children in India living along the street is in darkness. Due to a lack of education these children have stolen loot, have addictions, have gone the way of crime. Because alienating get them from all over the place. These children need love, happiness, counselling and guidance."/>	
Solution	
<input type="text" value="Under this project, the festival will provide street poor children's dinner, dress, toys and gifts. There will be cutting of cake, face painting, driving electric cars, sharing of gifts, singing and dancing. The kids will have Nutritional and delicious rice, meat, chicken and ice cream in their diet and a soft drink. Dinner/Lunch and gifts for Street poor children life will become a golden memory."/>	
Long Term Impact	
<input type="text" value="RSKS project 'Provide Gift and food to poor children in India' This exposure will enable kids relax, share gifts, share their experiences, and tell stories in a wonderful new environment outside home for one day with the RSKS volunteers. Dinner/lunch and gifts for Street poor children will become a golden memory. Everything we do for these kids now will never be erased from their memories."/>	

- **Vista Usuario**

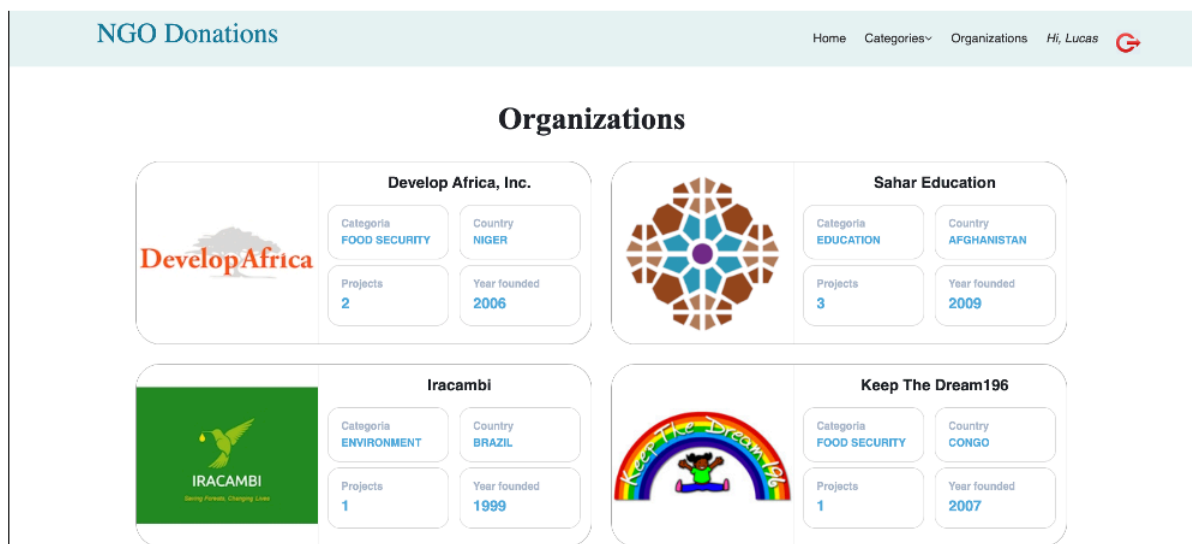
Por último, encontramos la vista de un usuario. En primer lugar, el usuario accederá al menú de Home donde dispondrá de diferentes funcionalidades.



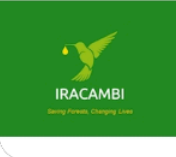



El usuario podrá iniciar sesión, a través de esta pantalla, en caso de no contar con una cuenta, se podrá acceder a la pantalla de registro, o en caso de no querer continuar con el inicio de sesión se puede acceder como 'guest'.



A continuación, se muestra la pantalla de las organizaciones existentes en el sistema desarrollado.




Organization	Logo	Category	Country	Projects	Year founded
Develop Africa, Inc.		FOOD SECURITY	NIGER	2	2006
Sahar Education		EDUCATION	AFGHANISTAN	3	2009
Iracambi		ENVIRONMENT	BRAZIL	1	1999
Keep The Dream196		FOOD SECURITY	CONGO	1	2007

EL usuario si lo desea también podrá acceder a una información más detallada de la organización y de los proyectos que esta está llevando a cabo.

NGO Donations
Home Categories Organizations Hi, Lucas

Afghan Institute of Learning

Organization Information



Mission: The Afghan Institute of Learning (AIL) works to empower all Afghans who are needy, especially women and children, providing them the knowledge and skills to care for themselves. AIL is expanding access to quality education and healthcare through community based programming, enabling communities to develop the capacity of their people. The goal is to create a foundation of quality education and health systems throughout Afghanistan which meet the needs of local people now and in the future.

Year founded: 1995

Country: AFGHANISTAN

Categorie: EDUCATION

Total Revenue

\$36.8m

Total Expenses

\$26.7m

Net Income

\$10.1m

Program Expense

81.5%

Administrative Expense

5.45%

Total Assets

\$114.8m

Total Liabilities

\$29.1m

Net Assets

\$85.7k


Fundraising Expense

14.04%

Working Capital

3.21 yrs

Projects (3)




EDUCATION | ZIMBABWE

Build the USAP Community School in Zimbabwe

Afghan Institute of Learning

Education Matters is building a 11th and 12th grade high school for 90 high-achieving, low-income Zimbabwean students to access quality education that values critical thinking...

[more](#)




EDUCATION | KENYA

Empower, Engage, Educate in Rural Kenya!

Afghan Institute of Learning

Imagine students in rural Kenya using computers and reading books! The Kenya Connect Learning Resource Center engages and empowers students and teachers from our 55 partner schools...

[more](#)



EDUCATION | DOMINICAN REPUBLIC

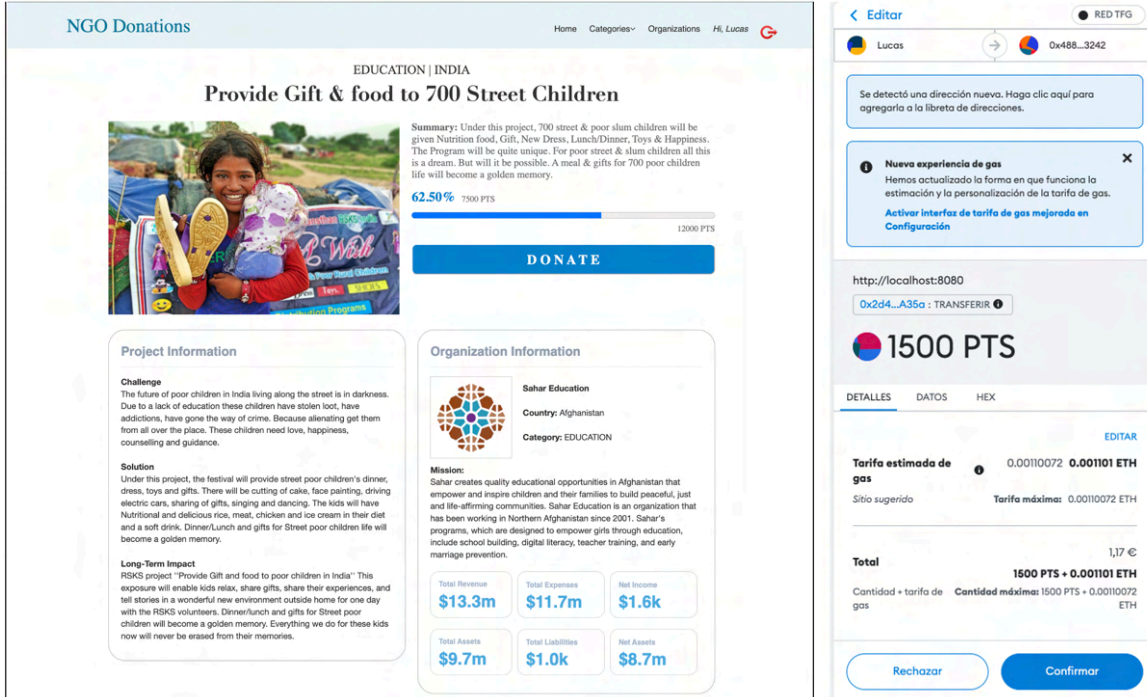
Offer Hope for 300 Youth

Afghan Institute of Learning

The Bienvenido Project provides holistic care for over 300 impoverished Dominican and Haitian youth in the areas of food, education, health care, advocating for the child's rights...

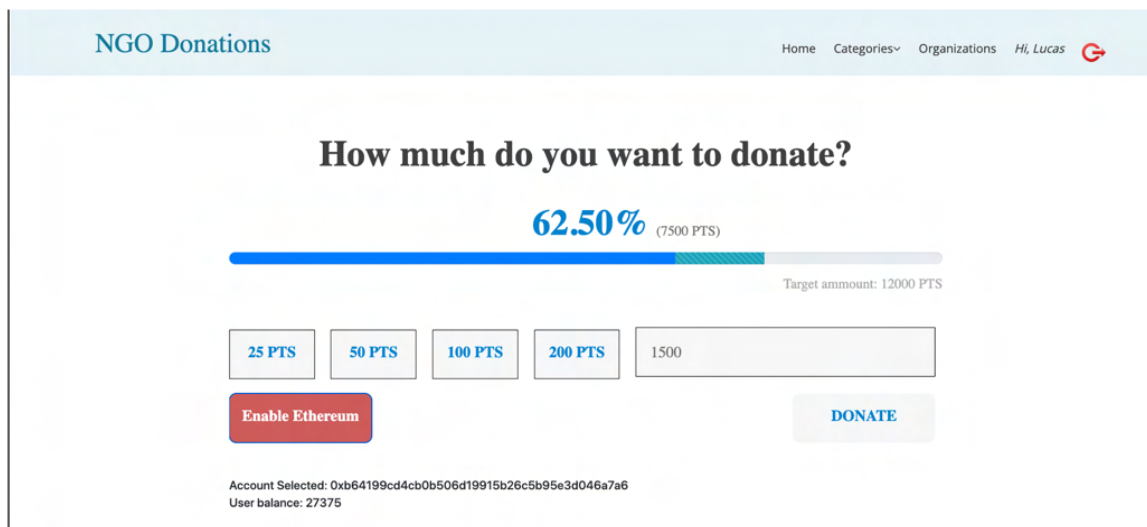
[more](#)

Un usuario podrá ver la información completa de un proyecto para posteriormente si lo desea realizar una donación.



The image shows two side-by-side screenshots from a mobile application. The left screenshot displays a project page for 'Provide Gift & food to 700 Street Children' under the 'EDUCATION | INDIA' category. It features a progress bar at 62.50% (7500 PTS) towards a target of 12000 PTS, a 'DONATE' button, and sections for Project and Organization Information. The right screenshot shows a mobile wallet interface with a notification about gas price updates, a balance of 1500 PTS, and transaction details for a gas purchase, including a 'Confirmar' button.

Cuando usuario desee realizar una donación, deberá en primer lugar, introducir la cantidad de esta, y en segundo lugar, conectarse a su cuenta de Ethereum por medio de Metamask para poder aceptar la transacción y que esta se realice.



The image shows a 'How much do you want to donate?' form. At the top, it displays '62.50% (7500 PTS)' with a progress bar and a 'Target amount: 12000 PTS'. Below this, there are buttons for '25 PTS', '50 PTS', '100 PTS', and '200 PTS', followed by a text input field containing '1500'. There is an 'Enable Ethereum' button and a 'DONATE' button. At the bottom, it shows 'Account Selected: 0xb64199cd4cb0b506d19915b26c5b95e3d046a7a6' and 'User balance: 27375'.

Por último, se muestran las pantallas de donaciones. En primera instancia, se mostrarán todas las donaciones realizadas. En segundo lugar, se mostrará las donaciones por proyectos, para que cualquier persona que lo desee pueda ver cuantas donaciones ha recibido un determinado proyecto.

NGO Donations
Home Categories Organizations Hi, Lucas

TRANSACTION HISTORY

[Donations By Project](#) | [See All](#) | [My Donations](#)

Date	Amount (PTS)	Project Name	Donor Public Address	Project Public Address
29/6/2022 11:5:16	1500	Provide Gift & food to 700 Street Children	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
22/6/2022 18:16:55	2000	Provide Gift & food to 700 Street Children	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
23/6/2022 14:7:46	100	Provide Gift & food to 700 Street Children	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
23/6/2022 18:28:12	1350	Provide Gift & food to 700 Street Children	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
24/6/2022 1:29:57	2000	Fight Covid and Climate Driven Hunger	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0xd8fb1c270baD8Ec27D026aFc2c83D4584182c6a
24/6/2022 1:31:49	1500	Forests4Water Brazil	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x7aB8E61b468CBb40350DF2874E7Aa637d0CD0A5
24/6/2022 1:37:5	100	Yezidi Children's Safe Space	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x0eFAD33a5Deb45730413f03a2e42cD00E97D245
24/6/2022 1:40:43	150	Yezidi Children's Safe Space	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x0eFAD33a5Deb45730413f03a2e42cD00E97D245
24/6/2022 1:48:59	50	Yezidi Children's Safe Space	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x0eFAD33a5Deb45730413f03a2e42cD00E97D245

NGO Donations
Home Categories Organizations Hi, Lucas

TRANSACTION HISTORY

[Donations By Project](#) | [See All](#) | [My Donations](#)

Project Name: Provide Gift & food to 700 Street Children, Goal: 12000PTS, Achieved: 7500PTS

Date	Amount (PTS)	Project Name	Donor Public Address	Project Public Address
22/6/2022 18:16:55	2000	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
23/6/2022 14:7:46	100	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
28/6/2022 11:5:16	1500	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
23/6/2022 18:28:12	1350	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242
24/6/2022 12:43:34	50	Help Feed Hungry Children in Zimbabwe	0x28549c8B1bfAec075D38ba6f5dc3C77317e4f582	0x4883f1AcacAF410029bc58664f32e554Fc3242
28/6/2022 23:21:21	2500	Help Feed Hungry Children in Zimbabwe	0xb64199cD4c80B506D19915b26c5b95E3D046a7a6	0x4883f1AcacAF410029bc58664f32e554Fc3242

Project Name: Fight Covid and Climate Driven Hunger, Goal: 23500PTS, Achieved: 3500PTS