Facultad Regional Rosario

Proyecto Final 2022

Ingeniería en Sistemas de Información

APLICACIÓN DESCENTRALIZADA

ADMINISTRADOR DE FONDOS COMUNES

Comisión: 503 ISI

Autor:

- Raselli, Gianfranco - 46472 (gianrase4@gmail.com)

Tutores:

- Ing. José Luis Albano
- Ing. Silvia Stortoni
- Ing. Valeria Aguzzi

Rosario, Mayo 2023

ÍNDICE

RESUMEN

El sistema de administración de fondos comunes será una aplicación descentralizada que correrá en la cadena de bloques (Blockchain) de Ethereum.
La función principal de la aplicación será llevar un registro de los activos (Ether) que las diferentes cuentas de Ethereum ponen a disposición para una causa común. Los mismos serán gestionados mediante contratos inteligentes (Smart Contracts), que son porciones de código que residen en la cadena de bloques mediante los cuales se implementará la lógica de negocio.
La aplicación contará con distintos usos prácticos, la misma permitirá gestionar los fondos de donaciones, pensiones, seguros, campañas de inversiones, entre varios usos más que se le puede asignar en el uso diario de cada grupo de personas.
Por lo tanto, el alcance funcional comprende desde la creación de un fondo común (una instancia de un contrato inteligente) por parte de uno o más administradores del mismo, pasando por la inserción de fondos (Ether) en dicho smart contract por parte de las diferentes cuentas/personas interesadas, hasta la utilización de dichos recursos (lo cual tendrá diferentes usos dependiendo del tipo de fondo; es decir, sea un fondo de donaciones, seguro, campañas de inversión, entre otros).
Nuestro sistema contará con las ventajas de una Blockchain pública:

- Descentralización: Un blockchain descentralizado añade una red de igual a igual a las características de seguridad existentes típicas de las bases de datos de un blockchain. Los miembros de esta red no tienen que confiar ni conocerse entre sí, sino que cada integrante obtiene una copia del mismo registro de contabilidad del blockchain.
- Transparencia: La transparencia en blockchain se consigue publicando las reglas con las que se define el funcionamiento de blockchain. Esto se logra haciendo público el código del software necesario para ejecutar blockchain y generando una comunidad de nodos y desarrolladores que siguen este principio de transparencia.
- Inmutabilidad: Ningún participante puede cambiar o falsificar una transacción unavez grabada en el libro mayor compartido. Si el registro de una transacción incluye un error, se debe añadir una nueva transacción para revertir el error, pero ambas transacciones serán visibles.
- Privacidad: Las direcciones blockchain no están ligadas a las identidades de las personas que controlan cada una de las direcciones blockchain. Para poder operar en un blockchain público es necesario disponer del par de claves pública y privada que permiten controlar la dirección blockchain.

- Trazabilidad: Blockchain permite recorrer la cadena de bloques y trazar todas las operaciones que se han realizado sobre una determinada dirección; o retroceder en el tiempo y revisar las transacciones que se hicieron en una fecha determinada explorando todos los bloques generados en la fecha indicada.

Nuestro sistema seguirá los principios de dicha tecnología para escribir las reglas lógicas sobre las que se regirá nuestra aplicación. Las mismas serán escritas en un lenguaje de alto nivel orientado a contratos denominado Solidity. Dicha lógica de negocio interactúa con una aplicación Web/Mobile desacoplada, donde el frontend (cliente) interactúa con la API (servidor) utilizando la tecnología Web3, lo que permitirá también escalar dicha aplicación en base a la demanda de sus usuarios mediante auto-escalado de instancias.

PALABRAS CLAVES

Aplicación Descentralizada – Ethereum – Cadena de Bloques – Donaciones – Fondos Comunes – Campañas – Finanzas – MetaMask – Contratos Inteligentes - Inversiones.

OBJETIVO

El objetivo del proyecto es construir una aplicación descentralizada que brinde a empresas o personas físicas un medio para administrar dinero, el cual puede ser aportado por varias partes diferentes, de forma totalmente transparente y trazable. Para ello, la aplicación web, que será la interfaz con la que un usuario interactúa, permitirá comunicarse con el backend (un contrato inteligente) y de esta manera configurar diferentes parámetros a la instancia de dicho smart contract que regularán las reglas de la administración del fondo. Los principales ítems configurables son:

- Si se permitirá a cualquier usuario de la red de Ethereum aportar dinero al fondo o si dicha acción sólo estará restringida a personas específicas (address).
- Cómo se realizará el proceso de extracción/transferencia del dinero del smart contract:
    - Si solo puede realizarlo el/los dueño/s del contrato inteligente.
    - Si cualquier persona que haya aportado al fondo puede retirar/transferir dinero del mismo.
    - Si cualquier persona que haya aportado al fondo puede retirar/transferir dinero del mismo con previa autorización del dueño del fondo o mediante una solicitud que será votada por las personas que hayan aportado dinero al mismo.

MARCO

La aplicación surge debido a que es habitual que cuando varias personas aportan dinero para una cierta causa común ocurran irregularidades con el manejo y la utilización de la misma por las personas que se encargan de administrarla. Esto es particularmente importante cuando la cantidad de personas involucradas en la causa es muy grande, pero también puede ocurrir en grupos más reducidos (por ejemplo cuando se junta dinero en un grupo de amigos).
Por lo tanto, la aplicación busca solventar y brindar transparencia en la utilización de este dinero, siendo una gran herramienta tanto para empresas como para personas individuales.

Definiciones:

- **Aplicación descentralizada o dApp:**
  - Las aplicaciones descentralizadas, también conocidas como "dApps" o "dapps", son aplicaciones digitales que se ejecutan en una red blockchain de computadoras en lugar de depender de una sola computadora. Podemos hacer una analogía de la comparación entre las apps tradicionales y las dApps haciendo alusión a las diferencias que existen entre un sistema centralizado y un sistema distribuido. El hecho de ser descentralizadas hace que estén libres del control y de la interferencia de una sola autoridad.
- **Blockchain:**
  - Blockchain es un libro mayor compartido e inmutable que facilita el proceso de registro de transacciones y de seguimiento de activos en una red de negocios. Un activo puede ser tangible (una casa, un auto, dinero en efectivo, terrenos) o intangible (propiedad intelectual, patentes, derechos de autor, marcas). Prácticamente cualquier cosa de valor puede ser rastreada y comercializada en una red de blockchain, reduciendo el riesgo y los costos para todos los involucrados.
- **Web 2.0**:
  - El término Web 2.0 se utiliza para referirse a sitios web que permiten a las personas colaborar y compartir información en línea de formas que antes no eran posibles. Este tipo de sitios web suelen considerarse interactivos, colaborativos o con contenidos generados por los usuarios. Sitios web como Facebook, Twitter, YouTube y Wikipedia son ejemplos de la web 2.0. La web

2.0 también se considera una forma de cultura participativa, lo que significa que adopta muchas formas, como el activismo social de base para el cambio global, la colaboración.

- **Web 3.0**:
  - La Web 3.0 es la tercera generación de servicios de Internet para páginas web y aplicaciones. Se centrará en el uso de una comprensión de datos basada en máquinas para proporcionar una Web semántica y datificada. El objetivo final de la Web 3.0 es crear sitios web más inteligentes, conectados yabiertos.
- **Ethereum:**
  - Ethereum en sí mismo es una plataforma digital que se basa en la tecnología blockchain o cadena de bloques. Su objetivo es convertirse en una blockchain capaz de ejecutar aplicaciones descentralizadas. Para lograr esto, este proyecto cuenta con una blockchain y una criptomoneda con características únicas. Entre ellas la capacidad de usar y crear smart contracty nuevos tokens. Ambas son poderosas funcionalidades, que permiten quese erija como una de las blockchain más completas y poderosas del criptomundo.
- **Ether:**
  - La moneda de la red se denomina Ether (ETH), y al igual que Bitcoin (BTC), el Ether se caracteriza por ser una criptomoneda que puede ser usada como método de pago entre pares. Otra de las semejanzas con Bitcoin y otras criptomonedas es que no está controlada por ningún gobierno u organismo regulador. Su desarrollo está marcado por la Ethereum Foundation, su Core Team y la comunidad que soporta y apoya. Otro punto importante, es que utiliza el protocolo de consenso Proof-of-Work (PoW), usando el algoritmo Ethash. Aunque esto puede cambiar en el corto plazo con el lanzamiento de Ethereum 2.0 y el salto a ser una criptomoneda usando el protocolo Proof of Stake (PoS).
- **Exchange:**
  - Son plataformas o mercados digitales que permiten intercambiar monedas digitales por dinero fiat y/u otras criptomonedas o mercancías.

- **NFT**:
    - Son las siglas en inglés de token no fungible (Non Fungible Token). Los NFT son un activo "inimitable" en el mundo digital que puede ser comprado y vendido como cualquier otro tipo de propiedad, pero no tienen forma tangible en sí mismos.
- **Solidity**:
    - Solidity es un lenguaje de programación orientado a objetos para escribir contratos inteligentes. Se utiliza para implementar contratos inteligentes en varias plataformas blockchain, la más destacada, Ethereum.
- **Contratos inteligentes o Smart Contracts:**
    - Los contratos inteligentes son líneas de código autoejecutables con los términos de un acuerdo que se establece entre un comprador y un vendedor verificados y ejecutados automáticamente a través de una red informática.
    - Nick Szabo los define como “protocolos de transacción computarizados que ejecutan los términos de un contrato”. Así, al ser implementados en cadenas de bloques hacen que las transacciones sean rastreables, transparentes e irreversibles.
- **Dirección cripto o Crypto Address:**
    - Una dirección cripto (de bitcoin, ether u otra criptomoneda) es un código alfanumérico que indica un posible destino para un pago de la criptomoneda que estés operando. Cada criptomoneda tiene su propia estructura de direcciones, por lo que no es posible abonar, por ejemplo, ethers en una dirección Bitcoin.
    - Todas las direcciones tienen asociada una llave privada y una llave pública: la **llave privada** corresponde a la contraseña con que firmas una transacción. Por lo tanto, es el acceso a los fondos que haya en una dirección. Quien tenga acceso a la llave privada tendrá acceso a las criptomonedas que haya en ella. Por su parte, la **llave pública** permite que lared pueda corroborar que una transacción cualquiera fue, efectivamente, firmada por la llave privada asociada a una dirección específica, y así corroborar que la transacción es válida.

Matriz FODA

Fortalezas

- **Descentralización de operaciones**: La aplicación correrá en la blockchain, lo que representa que las operaciones no dependen de un único ente (por ejemplo un banco) y las transacciones son directas entre usuarios.
- **Transparencia de transacciones**: Las operaciones realizadas en la blockchain quedan en un registro por lo que permite realizar un seguimiento de la utilización de los fondos.
- **Escalable y no dependiente de un único servidor**: La aplicación escala ya que no depende de un único nodo, lo que además agrega robustez y estabilidad.
- **Cada transacción puede ser pública y/o anónima**: Las transacciones quedan registradas pero no necesariamente se sabe quién es el usuario que la realiza, a menos que el usuario anuncie la Address que le corresponde.

Oportunidades

- **Nuevas oportunidades de negocio no exploradas**: Mucha gente todavía no tiene conocimientos del sector por lo que el nivel de utilidad que tiene la blockchain actualmente es desconocido.
- **Ser líderes en un área no desarrollada**: Debido a la cantidad de aplicaciones centralizadas, es posible marcar una impronta en lo que se conoce como Web 3.0.

Debilidades

- **Tecnología relativamente nueva**: Genera una necesidad de conocimiento que muchas veces es difícil encontrar.
- **Los costos pueden ser impredecibles**: Los costos son impredecibles debido a que el precio del “gas” o “fee” depende de la cantidad de transacciones que se estén dando en la blockchain al momento de querer realizar una operación, esto será resuelto con el lanzamiento de ETH 2.0, la cual se estipula llegará para finales de 2022.
- **Experiencia de Usuario limitada**: Las transacciones actualmente no son instantáneas, por lo que a veces se puede reflejar cierta demora entre el envío y la recepción de fondos.

Amenazas

- **Mercado debilitado por volatilidad**: Los usuarios tienen mucha incertidumbre sobre el estado de la blockchain debido a un mercado volátil donde el precio de los activos es determinado por oferta y demanda de sus propios usuarios.
- **Estafas e incertidumbre en el "entorno cripto"**: Se han dado a conocer muchas estafas lo que genera cierto escepticismo sobre las nuevas aplicaciones dentro de la Web 3.0, por lo que es necesario trabajar en una marca transparente y confiable.

VENTAJA COMPETITIVA

Actualmente existen múltiples plataformas de gestión de fondos en la Web 2.0, entre ellas aplicaciones muy conocidas como pueden ser Kickstarter e Indiegogo, aunque éstas no están basadas en la blockchain, por lo que resta transparencia y descentralización.

- **Kickstarter**
- **Indiegogo**
- **AirFunding**
- **DonorPerfect**
- **Cafecito**

Aún no es un sector muy explorado en la blockchain ya que la mayoría de las aplicaciones están focalizadas en Exchanges y NFTs, aunque sí podemos considerar a BitGive, la cual es una DAO para realizar donaciones.

- **BitGive**

Buscamos combinar lo mejor de ambos mundos, en el caso de BitGive el enfoque está dado en donaciones sin fines de lucro, pero aplicaciones como Kickstarter o Indiegogo buscan potenciar no sólo proyectos sin fines de lucro, sino también financiar startups o proyectos indie que requieren de un capital inicial del cuál quizás sus fundadores no disponen, pero no cuentan con las ventajas de transparencia y solidez que provee la Blockchain.
Con este proyecto deseamos implementar las ventajas que nos provee la Web 3.0 para poder colaborar y desarrollar un mundo más descentralizado y transparente en el que no debemos depender de una organización externa o las limitaciones de un gobierno de turno para poder fondear un proyecto o realizar donaciones a una organización sin fines de lucro además de irrumpir en un mercado en el que todavía no se ha realizado tanto énfasis y no se han descubierto las grandes ventajas que hoy encontramos gracias a las nuevas tecnologías.

INFRAESTRUCTURA

Hardware

No se requerirá de ningún tipo de Hardware propio para implementar la aplicación en producción debido a que el backend, escrito en Solidity, será desplegado en la red descentralizada de Ethereum mediante la conexión con un nodo de la red, la cual se podrá implementar utilizando una herramienta gratuita como es Infura.
Por otro lado, el frontend (aplicación web) será desplegado en el sistema de archivos distribuido IPFS, el cual nos brindará la principal ventaja de que dicha aplicación será almacenada en varios nodos (computadoras de la red) descentralizadas.
De esta manera, tendremos una aplicación descentralizada completa, la cual no podrá ser atacada a través de un servidor centralizado.

Software

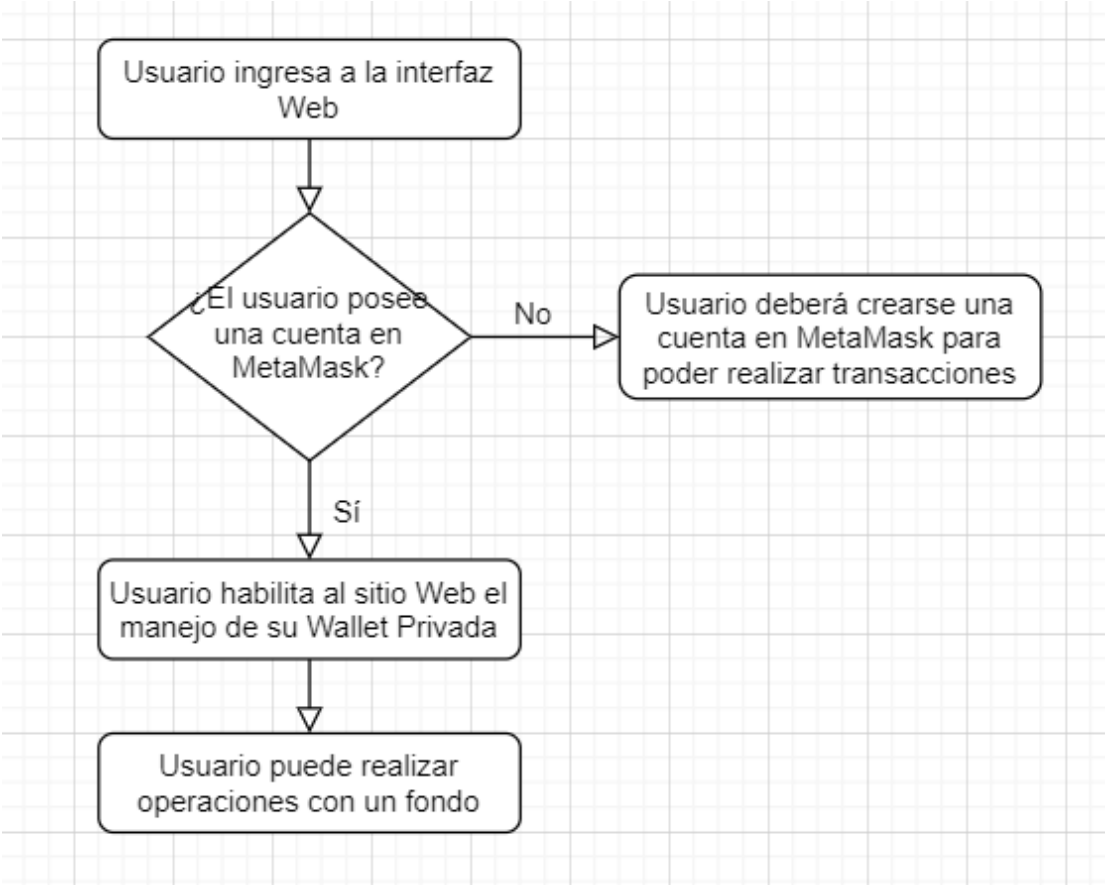
Los programas o softwares necesario para realizar el proyecto serán:

- Remix: Remix es un entorno integrado de desarrollo (IDE) basado en un navegador que integra un compilador y un entorno en tiempo de ejecución para Solidity sin los componentes orientados al servidor.
- Visual Studio Code: Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. Cuenta con una gran cantidad de extensiones que nos ayudarán en nuestro desarrollo.
- Infura: Es un conjunto de herramientas para que cualquiera pueda crear una aplicación que se conecte a la cadena de bloques Ethereum. Interactúa con la cadena de bloques Ethereum y ejecuta nodos en nombre de sus usuarios.
- IPFS: InterPlanetary File System, es un sistema de archivo descentralizado que busca garantizar la seguridad, privacidad y resistencia a la censura de tus datos.
- Ganache: Ganache es un software que nos proporciona una red de pruebas local súper sencilla e intuitiva.
- Rinkeby Testnet: es una red de prueba de Ethereum que nos permitirá desplegar nuestra aplicación en la misma para hacer diferentes pruebas sin gastar dinero (ethers) real.
- Truffle: Truffle es un conjunto de herramientas que nos permitirá crear aplicaciones sostenibles y profesionales utilizando la Máquina Virtual Ethereum (Ethereum Virtual Machine, EVM).

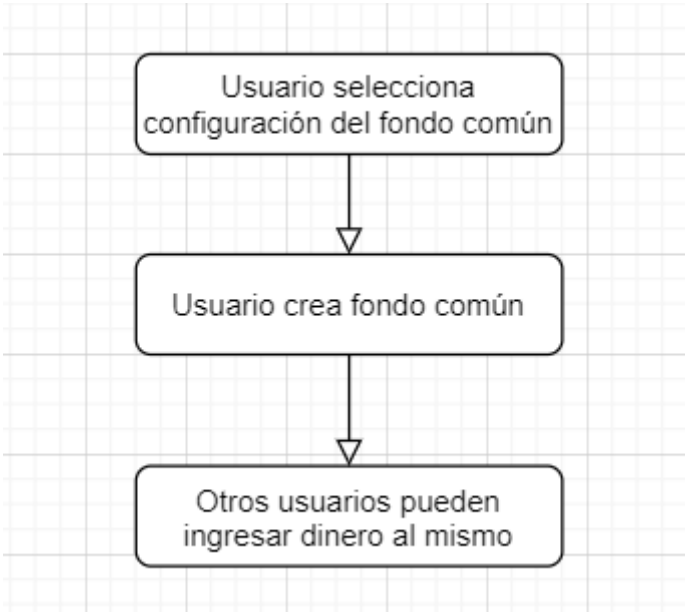
- Hardhat: Es un entorno de desarrollo para compilar, desplegar, testear y depurar un software de Ethereum. Nos ayudará a gestionar y automatizar tareas recurrentes.
- MetaMask: MetaMask es un software de criptomoneda que es instalado como extensión de un navegador web. Este es utilizado para interactuar con la plataforma de blockchain Ethereum.

DIAGRAMAS DE ACTIVIDADES

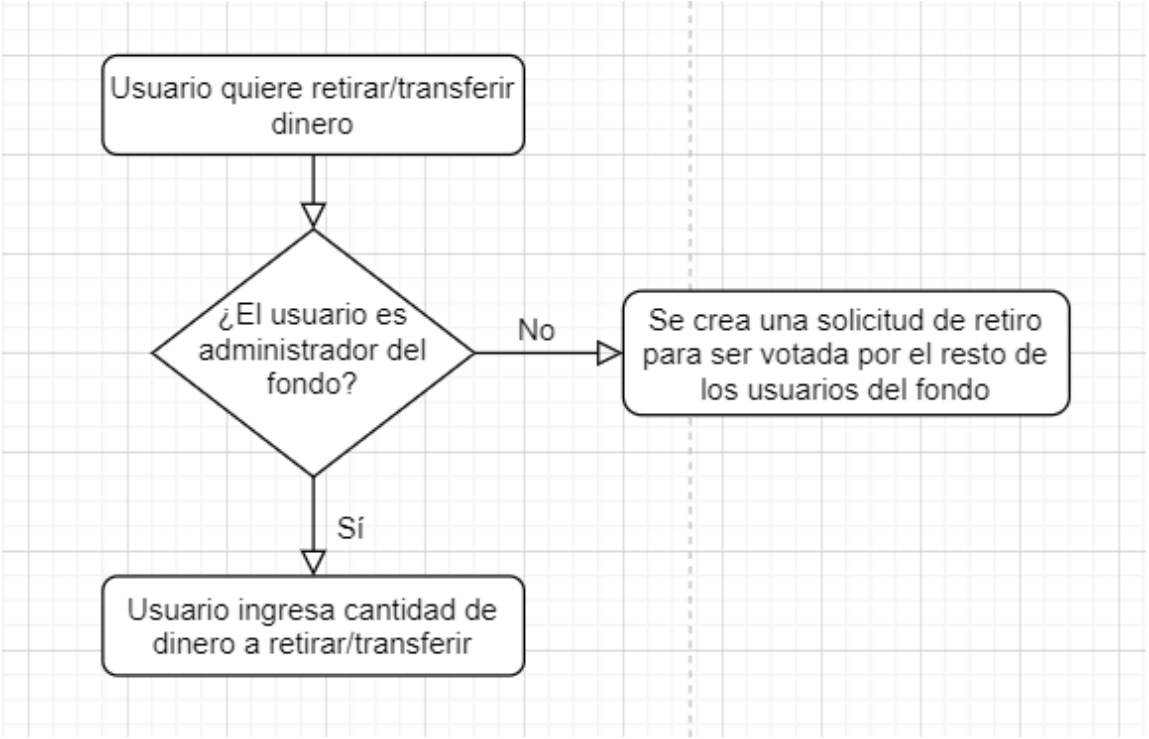
Conexión con el sitio Web



Creación de un fondo común



Retiro de dinero del fondo



ANÁLISIS DE FUNCIONES

La función principal de la aplicación será llevar un registro de los activos (Ether) que las diferentes cuentas de Ethereum ponen a disposición para una causa común. Los mismos serán gestionados mediante contratos inteligentes (Smart Contracts), es decir, mediante porciones de código que residen en la cadena de bloques mediante los cuales se implementará la lógica de negocio.

FundFactory

Este Smart contract será el encargado de llevar un registro de todos los fondos creados, por lo tanto, el mismo poseerá una función que le permitirá crear los mismos a cualquier persona. A su vez, será el dueño del contrato de FundToken, esto le permitirá vender los mismos como así también recibir los pagos en dichos tokens cuando se quiera crear un nuevo fondo.

Variables

- FundToken public immutable fundToken
- uint256 public fundTokenPrice
- uint256 public constant createFundPrice = 1
- Fund[] public deployedFunds

Eventos

- NewFundTokenPrice(uint fundTokenPrice)
- FundTokensBuyed(address indexed buyer, uint fundTokensBuyed)
- NewFund(address fundAddress, string name, string description, address indexed creator, uint256 createdAt)

Funciones

- changeFundTokenPrice(uint256 _newFundTokenPrice) public onlyOwner
- buyFundTokens(uint256 _fundTokens) public payable
- withdrawMoney() public onlyOwner
- createFund(

    string memory _name,

    string memory _description,

    address[] memory _managers,

    bool _managersCanBeAddedOrRemoved,

bool _managersCanTransferMoneyWithoutARequest,
bool _requestsCanBeCreated,
bool _onlyManagersCanCreateARequest,
bool _onlyContributorsCanApproveARequest,
uint256 _minimumContributionPercentageRequired,
uint256 _minimumApprovalsPercentageRequired) public

- getDeployedFundsCount() public view returns (uint256)
- getDeployedFunds() public view returns (Fund[] memory)

FundToken

Dicho contrato inteligente heredará del Smart Contract ERC20 de OpenZeppelin. El cual es un proyecto de Software libre que se encarga de realizar la implementación a alto nivel del estándar de Token ERC20.
El ERC-20 introduce un estándar para los tokens funcionales, es decir, tienen una propiedad que hace que cada token sea exactamente igual (en tipo y valor) que otro token.
Mediante dichos tokens (FundToken) las personas podrán crear nuevos fondos pagando por realizar dicha función una cantidad de los mismos. En nuestro caso, hemos definido que la creación de fondos cueste 1 FundToken.
Como fue mencionado anteriormente, nuestro contrato heredará todas las funcionalidades y variables definidas en el contrato heredado (las cuales le permitirán a los poseedores de tokens administrarlos). Sin embargo, a continuación describiremos solamente las características agregadas en nuestro FundToken contract.

Funciones

- decimals() public pure override returns (uint8)
- mint(address _account, uint256 _amount) public onlyOwner
- burn(address _account, uint256 _amount) public onlyOwner

Fund

Finalmente tenemos el contrato que permitirá administrar cada fondo creado. Este será el contrato inteligente más extenso y con mayor lógica.
Cuando un nuevo fondo es creado mediante el FundFactory, la dirección (address) del Fund contract instanciado (creado) es almacenada en dicha fábrica de contratos.

Variables

- string public name
- string public description
- address public immutable creator
- uint256 public immutable createdAt = block.timestamp
- address[] public managers
- mapping(address => bool) public isManager
- bool public immutable managersCanBeAddedOrRemoved
- address[] public contributors
- mapping(address => uint256) public contributions
- uint256 public totalContributions
- bool public immutable managersCanTransferMoneyWithoutARequest
- Request[] public requests
- bool public immutable requestsCanBeCreated
- bool public immutable onlyManagersCanCreateARequest
- bool public immutable onlyContributorsCanApproveARequest
- uint256 public immutable minimumContributionPercentageRequired
- uint256 public immutable minimumApprovalsPercentageRequired;

Eventos

- NewManager(address indexed manager)
- RemoveManager(address indexed manager)
- Contribute(address indexed contributor, uint256 value)
- Transfer(address indexed sender, address indexed to, uint256 value)
- NewRequest(string description, address indexed petitioner, address indexedrecipient, uint256 valueToTransfer)
- ApproveRequest(uint256 indexed requestIndex, address indexed approver)
- FinalizeRequest(uint256 indexed requestIndex, uint256 transferredValue)

Funciones

- addNewManagers(address[] memory _managers) public
- removeManager(uint256 _index) public
- managersCount() public view returns (uint256)
- getManagers() public view returns (address[] memory)
- contribute() public payable
- contributeFor(address _for) public payable
- contributorsCount() public view returns (uint256)
- getContributors() public view returns (address[] memory)

- balance() public view returns (uint256)
- transfer(address _to, uint256 _value) public
- createRequest(string memory _description, address _recipient, uint256

  _valueToTransfer) public
- requestsCount() public view returns (uint256)
- approveRequest(uint256 _index) public
- finalizeRequest(uint256 _index) public nonReentrant
- _contribute(address _contributor) private

ESTUDIO DE FACTIBILIDAD OPERACIONAL, TÉCNICA Y LEGAL

Factibilidad económica

Para determinar la factibilidad económica realizaremos una comparativa de los costos y beneficios que se reflejan en el proyecto.

Costos

- Dado que el mismo es llevado a cabo por un grupo de estudiantes que no cuentan con una retribución económica directa por las tareas realizadas, no habrá costos implicados en RRHH.
- Dado que los Smart Contracts serán subidos en una blockchain pública no habrá que pagar para usar la misma. Los únicos costos asociados a este ítem serán en los que se incurrirán al realizar la transacción para desplegar dichos contratos (un solo pago con tarifas relativamente bajas).
- Hosting interfaz web: la misma será alojada en un servicio de hosting gratuito como Heroku o Netlify.
- Publicidad: de momento no se derivarán nuevos costos asociados a la imagen de la aplicación. Pero es una opción que se deberá tener en cuenta para un futuro.

Beneficios

- Cada vez que una persona quiera crear un fondo mediante la FundFactory deberá pagar al Smart Contract con un FundToken. El precio en token para crear un nuevo fondo será fijo, lo que variará será el precio a los que se venderán dichos tokens. Por lo tanto, los beneficios serán dependientes del precio que la oferta y la demanda del mercado asignen a los mismos. El contrato inteligente será el único encargado de crear o quemar nuevos tokens.

Más allá de que los beneficios económicos del proyecto son un poco inciertos de momento, dado los costos casi inexistentes en los que se incurrirá para llevar a cabo el mismo, contar con una mínima cantidad de usuarios utilizando la aplicación los beneficios ya serán mayores que los costos. Por lo tanto, el proyecto es completamente factible económicamente.

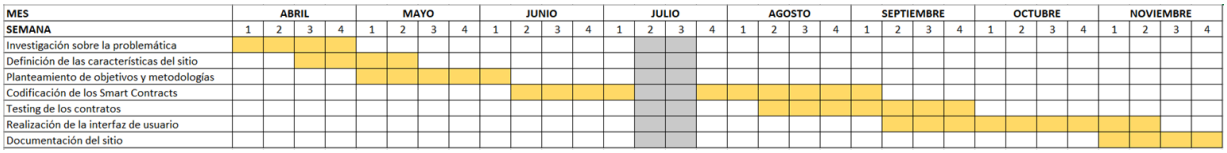
Factibilidad técnica

Desde el punto de vista técnico, el proyecto es completamente factible. Las tecnologías que se elegirán e implementarán son las más completas y lo mejor que ofrece el mercado para los requerimientos del proyecto. Además, los integrantes del equipo se encuentran totalmente capacitados para trabajar con las mismas.

Factibilidad legal

En este punto se tendrá en consideración que todos los frameworks utilizados para el desarrollo de la aplicación como la librerías utilizadas tanto en los contratos inteligentes como en el frontend poseen una licencia MIT. La misma es una licencia de software libre permisiva lo que significa que impone muy pocas limitaciones en la reutilización y por tanto posee una excelente compatibilidad de licencia. Además, permite reutilizar software dentro del software propietario. En consecuencia, la factibilidad legal del proyecto presentado está completamente asegurada para la empresa y no tendrá motivo alguno de alarmarse por estas cuestiones legales.

DIAGRAMA DE GANTT



MES	ABRIL				MAYO				JUNIO				JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE			
SEMANA	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Investigación sobre la problemática																																
Definición de las características del sitio																																
Planteamiento de objetivos y metodologías																																
Codificación de los Smart Contracts																																
Testing de los contratos																																
Realización de la interfaz de usuario																																
Documentación del sitio																																

https://docs.google.com/spreadsheets/d/1_rfZPxQTqRnXUfqlwmJ9CVaoZ4bAhKth/edit#gid=1692303644

PRESUPUESTO

Nombre del proyecto: Administrador de fondos comunes

Tipo de proyecto: Aplicación descentralizada

Red de despliegue: Blockchain de Ethereum

Funciones que se podrán realizar en la aplicación:

- Comprar FundTokens (lo que le permitirá crear un nuevo fondo)
- Crear nuevos fondos personalizables (Smart Contract que cuenta con varias reglas predefinidas que permitirá controlar el dinero que varios usuarios aportan para una causa común)
- Consultar los fondos creados
- Consultar los fondos creados por uno mismo
- Aportar dinero (ethers) a los fondos que crea conveniente
- Administrar los fondos en los que es un manager
- Crear solicitudes para retirar dinero
- Votar por aprobar una solicitud de retiro de dinero (en caso que sea un contribuyente o un administrador del mismo)

Interfaz de usuario: Aplicación Web realizada con VueJS

Ventajas que brinda la tecnología Blockchain:

- Descentralización
- Transparencia
- Inmutabilidad
- Privacidad
- Trazabilidad instantánea
- Mayor eficiencia y velocidad
- Seguridad reforzada

Fecha estimada de finalización: Noviembre de 2022

Instalaciones y equipos a utilizar: los mismos serán proveídos por los propios integrantes del equipo (serán autosuficientes en este punto)

Cantidad de personas abocadas al proyecto inicialmente:

- Miño, Julian (renuncia)
- Raselli, Gianfranco (a cargo de la continuación del proyecto)
- Romaniuk, Federico (renuncia)

Cantidad de horas estimadas de trabajo: 750 horas (150 días)

Duración total del proyecto: 30 semanas

Costos asociados al proyecto:

Costos asociados al proyecto	Gas utilizado	ETH/Gas	USD/ETH	Costo en USD
Costo energético (computadora)		-		$ 50
Honorarios por trabajador (1 persona)				$ 2,000
Pago de servicios		-		$ 100
Despliegue FundFactory	6400000	0.000000015	$ 1,600	$ 154
		-		
Costos totales				$ 2,304

Monetización del proyecto: La monetización del proyecto presentado consta de un pago por única vez y posteriormente una suscripción mensual de bajo coste a un servicio de mantenimiento de la página. En este nos encargaremos de cualquier problema presentado durante el transcurso de su funcionamiento y puesta en marcha, como también pequeños ajustes que deban de surgir durante su implementación en la organización de la facultad.

Precio inicial del FundToken:

- Cantidad estimada de fondos creados en el año inicial: 360 (30 nuevos fondos en promedio por mes)
- Se planea recuperar la inversión realizada en el proyecto en el lapso de 2 años. Por lo tanto, en el primer año se debería recuperar aproximadamente $1.152 USD
- Precio del FundToken = $1.152 USD / 360 fondos estimados por año = $3,2 USD ≈ 0,00196 ETH (a precio actual)

DOCUMENTACIÓN DE ANÁLISIS Y DISEÑO

Listado y especificaciones de requerimientos

Requerimientos funcionales

- El sistema permitirá a un usuario de la red de Ethereum crear un nuevo fondo común (instanciar un contrato inteligente) del cual será el administrador.
  - Los usuarios podrán crear la cantidad de fondos que deseen con la única restricción que deberán abonar 1 FundToken por cada uno que se instancie.
- El sistema permitirá a un usuario de la red de Ethereum que desea crear un nuevo fondo configurar una serie de parámetros iniciales del mismo que determinarán las reglas del contrato inteligente que se instanciará.
  - Cuando un usuario cree un nuevo fondo deberá configurar los siguientes parámetros (que luego no podrán ser modificados por la seguridad del fondo): tipo de fondo, nombre, los administradores, si los administradores pueden ser agregados o removidos, si los administradores pueden transferir dinero sin una solicitud, si las solicitudes pueden ser creadas, si solo los administradores pueden crear una solicitud, si solo los contribuyentes pueden aprobar una solicitud, el mínimo porcentaje de contribución requerido para aprobar una solicitud, el mínimo porcentaje de aprobaciones requerido para finalizar una solicitud.
- El sistema permitirá a un usuario de la red de Ethereum ingresar dinero (ether) en un fondo común particular.
  - Un usuario puede transferir dinero a un fondo en su nombre e incluso lo puede hacer en nombre de otra entidad.
- El sistema permitirá a un usuario de la red de Ethereum autorizado retirar/transferir dinero (ether) de un fondo común al que haya aportado anteriormente.
  - En caso que un usuario sea administrador de un fondo y de que los mismos puedan retirar dinero sin una solicitud (parámetros a configurar al crear el fondo), la entidad podrá transferir dinero del fondo común hacia otra entidad.
- El sistema permitirá a un usuario de la red de Ethereum realizar una solicitud de retiro de dinero de un fondo al que haya aportado anteriormente.
  - En caso de que el fondo permita la creación de solicitudes, una entidad podrá solicitar la transferencia de cierta cantidad de dinero del fondo creando una, que luego deberá ser votada por los contribuyentes.
- El sistema permitirá a un usuario de la red de Ethereum votar el retiro de dinero de un fondo por parte de otro contribuyente que lo haya solicitado.

- En caso de que un contribuyente alcance el porcentaje mínimo de contribución requerido para aprobar una solicitud, podrá votar a favor de la transferencia de dinero de una solicitud creada previamente.

- El sistema permitirá a un usuario de la red de Ethereum administrador de un fondo delegar la administración del mismo a otras personas.
  - En caso que la opción de agregar o remover nuevos administradores, la entidad creadora podrá relegar la responsabilidad en otro usuario.
- El sistema mostrará un registro de todas las operaciones del contrato inteligente del fondo realizadas.
  - Cada fondo mostrará un listado de todas las transferencias realizadas, los solicitudes creadas, los administradores a cargo, las contribuciones realizadas, entre todos los demás datos del fondo.

Requerimientos no funcionales

- La aplicación estará disponible siempre debido a que correrá en una red descentralizada y la misma estará distribuida en diferentes nodos (máquinas).
  - La aplicación estará a salvo de la caída de un servidor en una arquitectura centralizada.
- La aplicación mantendrá la privacidad de cada usuario dado que los mismo solos estarán identificados en el mismo con una dirección de la red de Ethereum (anonimato).
  - Como se sabe en una aplicación descentralizada toda la información es pública para lograr la transparencia, pero al mismo tiempo la información es anónima porque los usuarios que realizan las transacciones en la red solo están identificados por una dirección (por ej. 0x2b4d87eff06f22798c30dc4407c7d83429aa9abc) sin revelar la verdadera identidad.
- Toda la interacción de la interfaz web con el backend será llevada a cabo directamente por los usuario mediante la utilización de una wallet privada (MetaMask).
  - Las transacciones que se lleven a cabo con la red de Ethereum y el backend en NodeJS deberán ser firmadas directamente por los usuarios mediante la billetera MetaMask que deberán tener instalada en el navegador para poder utilizar la aplicación.
- Toda la actividad que realizan los usuarios con el contrato inteligente del fondo común será rastreable y nunca podrá ser eliminada de la red de Ethereum.

- Cada transacción que el usuario firma con MetaMask y que es incluida en la cadena de bloques por los mineros de la red de Ethereum, nunca podrá ser eliminada de la misma para brindar mayor seguridad, dado que la información estará distribuida en millones de nodos que participan en la red.

Arquitectura del producto de software

La arquitectura utilizada es una arquitectura híbrida:

- Se utilizó una arquitectura descentralizada, utilizando las ventajas de la red de Ethereum, para almacenar los datos cruciales para el funcionamiento de la aplicación.
- Por cuestiones de costos, se utilizó una arquitectura cliente-servidor centralizada para almacenar los datos que no sean imprescindibles para el uso del sistema.

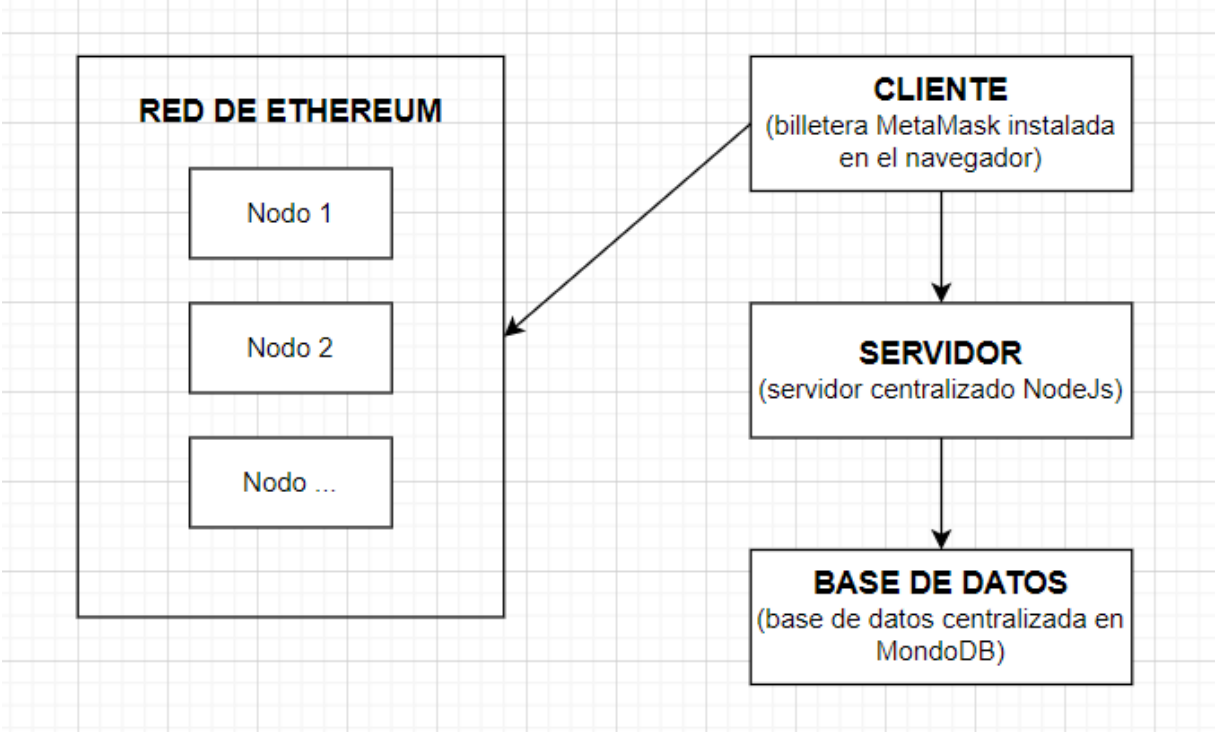


Diagrama de clases (contratos inteligentes Ethereum - Solidity)

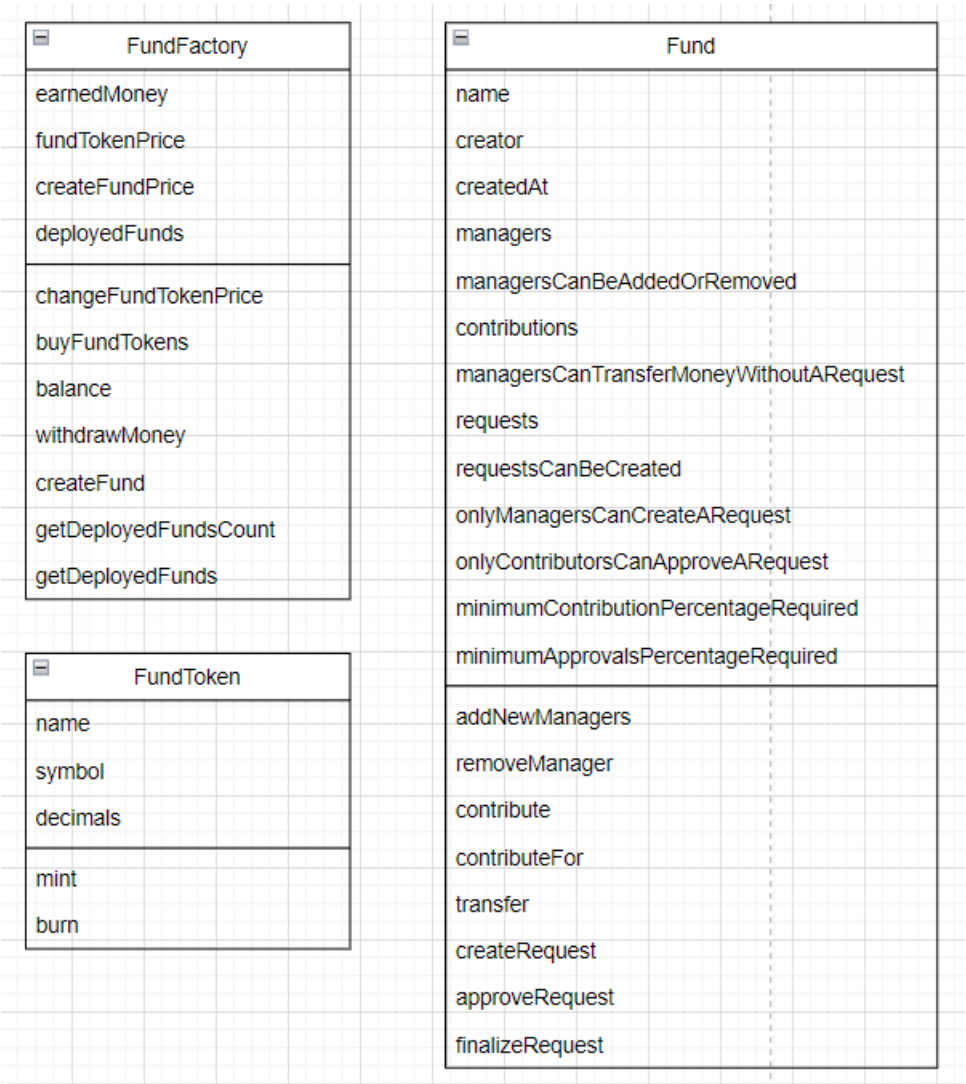
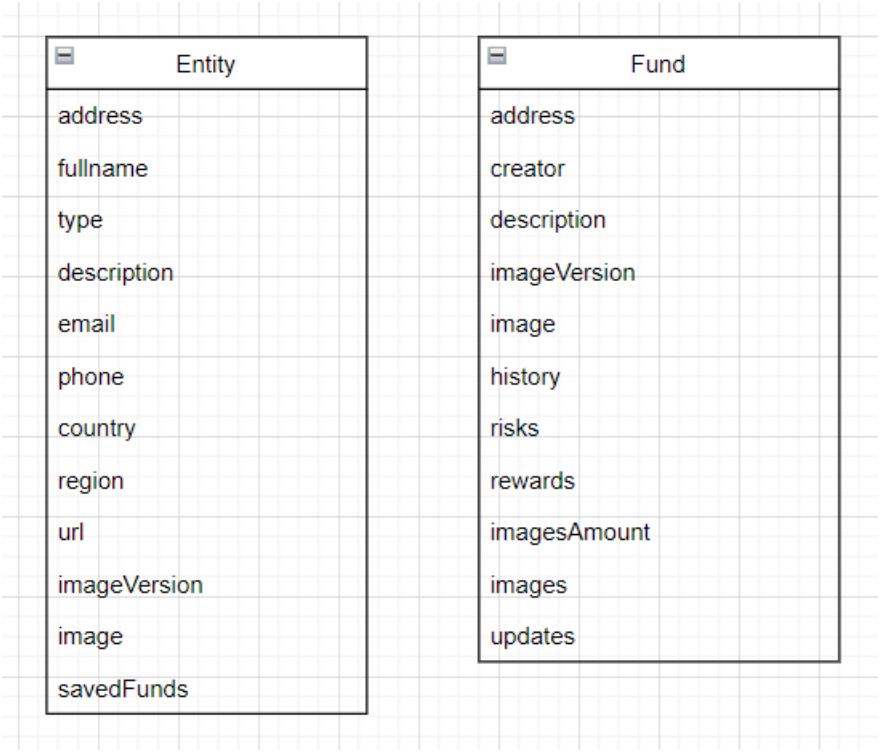


Diagrama de clases (base de datos – MongoDB)



Estándares de desarrollo

- Programación frontend: HTML, CSS, JavaScript, VueJS versión 3.0 o superior.
- Programación backend: NodeJS versión 18.6 o superior.
- Programación Ethereum: Lenguaje Solidity versión 0.8.16 o superior.
- Base de datos: MongoDB versión 6.0 o superior.
- Billetera para conectar con el sitio web: MetaMask.
- Red descentralizada: Cadena de bloques de Ethereum.

Conclusión personal

La conclusión a la que podemos arribar; luego de realizado todo el análisis, diseño y finalmente la implementación del sistema, es que resulta crucial tener un amplio conocimiento de diferentes arquitecturas de desarrollo de software para ser capaz de aprovechar las ventajas particulares de cada una de ellas. Como así también, se deben conocer cuales son sus debilidades (como por ejemplo en nuestro caso, la cadena de bloques nos brinda muchas cualidades positivas pero una desventaja es el costo de las transacciones, por lo tanto, incorporamos un servidor centralizado para las funciones menos importantes de la aplicación) para buscar soluciones alterativas que sean mejor en ese ítem que la tecnología elegida. Es evidente que esto no es sencillo de conseguir porque se debe tener mucho conocimiento de varias tecnologías, por lo que es clave antes de emprender un proyecto hacer una investigación exhaustiva en la Web de diferentes soluciones a problemas conocidos, y luego, a medida que se va desarrollando el trabajo, ir aprendiendo más acerca de las herramientas previamente seleccionadas.

MANUAL DE USUARIO

Acerca de Nosotros

(https://proyecto-final-blockchain.netlify.app/nosotros)

¿Quiénes somos?

La aplicación comenzó como proyecto llevado a cabo para la materia Proyecto Final de la carrera Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional (UTN) ubicada en Rosario, Argentina.

¿Qué tipo de aplicación es?

El sistema de administración de fondos comunes es una aplicación descentralizada que funciona sobre la cadena de bloques (Blockchain) de Ethereum.

¿Cuál es la función de la aplicación?

La función principal de la aplicación es llevar un registro de los activos (Ether) que las diferentes cuentas de Ethereum (entidades) pongan a disposición para una causa común. Los mismos serán gestionados mediante contratos inteligentes (Smart Contracts), que son porciones de código que residen en la cadena de bloques mediante los cuales se implementó la lógica de negocio.

¿Qué usos se le puede dar a la aplicación?

La aplicación cuenta con distintos usos prácticos, la misma permite gestionar los fondos de donaciones, pensiones, seguros, campañas de inversiones, entre varios usos más que cada entidad le puede asignar personalizando diferentes parámetros a su medida durante la creación de los mismos.

¿Qué ventajas nos proporciona la cadena de bloques?

Descentralización

Una cadena de bloques descentralizada añade una red de igual a igual a las características de seguridad existentes típicas de las bases de datos de una blockchain. Los miembros de esta red no tienen que confiar ni conocerse entre sí, sino que cada integrante obtiene una copia del mismo registro de contabilidad de la misma.

Transparencia

La transparencia se consigue publicando las reglas con las que se define el funcionamiento de la cadena de bloques. Esto se logra haciendo público el código del software necesario para

ejecutar la red y generando una comunidad de nodos y desarrolladores que siguen este principio de transparencia.

Inmutabilidad

Ningún participante puede cambiar o falsificar una transacción una vez grabada en el libro mayor compartido. Si el registro de una transacción incluye un dato no deseado, se debe añadir una nueva transacción para revertir el error, pero ambas transacciones serán visibles.

Privacidad

Las direcciones de las cuentas no están ligadas a las identidades de las personas. Para poder operar en la cadena de bloques es necesario disponer del par de claves pública y privada.

Trazabilidad

La cadena de bloques permite realizar un recorrido de todas las operaciones que se han realizado sobre una determinada dirección, o retroceder en el tiempo y revisar las transacciones que se hicieron en una fecha determinada explorando todos los bloques generados en la fecha indicada.

Preguntas Frecuentes

(https://proyecto-final-blockchain.netlify.app/preguntasFrecuentes)

¿Qué se necesita para comenzar a utilizar la aplicación?

Todo lo que se necesita para comenzar a interactuar con la aplicación es contar con la billetera MetaMask instalada en el navegador.

La aplicación detecta automáticamente si la billetera está instalada, en caso de ser así, le aparecerá un botón en la esquina superior derecha que le permitirá vincular una cuenta de MetaMask con el sistema.

¿Qué es MetaMask?

MetaMask es una billetera y un navegador al mismo tiempo, sirve para almacenar e intercambiar activos digitales y para explorar nuevas aplicaciones en la web descentralizada.

https://metamask.io/

¿Cómo instalar MetaMask?

Dirígete a instalar MetaMask y dale al botón «Install MetaMask for Chrome», luego sigue los pasos para culminar la instalación.

¿Para qué sirve un FundToken?

El FundToken es la moneda interna que se puede comprar en la aplicación. La misma funciona como un ticket para poder crear un fondo.

¿Cómo comprar FundTokens?

1. Desde cualquier parte de la aplicación dale al botón «Comprar FundTokens» en la barra de navegación superior. En caso de estar navegando desde un dispositivo móvil debe dirigirse a la parte inferior, y darle al botón «FundToken».
2. En la ventana emergente indique la cantidad de FundTokens que desea comprar.
3. Dale al botón «Comprar» para confirmar la compra (para poder proceder con la compra debes estar conectado a la aplicación con una cuenta de MetaMask).
4. Finalmente, se le abrirá una nueva ventana de MetaMask indicandole el costo de la compra, para finalizar la misma debe darle al botón «Confirmar».

¿Quiénes pueden crear un fondo?

Cualquier persona, empresa o entidad que disponga de una cuenta en la billetera de MetaMask puede crear e interactuar con cualquier fondo.

¿Cómo crear un nuevo fondo?

1. Desde cualquier parte de la aplicación dale al botón «Crear fondo» en la barra de navegación superior. En caso de estar navegando desde un dispositivo móvil debe dirigirse a la parte inferior, y darle al botón «Crear».
2. Aquí debe personalizar los parámetros del fondo a crear según sus necesidades (una vez creado no se pueden modificar). Para facilitarle la tarea, puede seleccionar tres diferentes tipos de fondo (con sus respectivos parámetros ya configurados) al comienzo del formulario. En caso de que dichos tipos no se adapten a tus preferencias, puede seleccionar «Fondo personalizado» y configurar todos los parámetros a su gusto. Luego, debe darle al botón «Crear fondo» (para eso debe disponer de al menos 1 FundToken en su cuenta de MetaMask).

¿Qué tipos de fondos se pueden crear?

Existen tres tipos de fondos que, al momento de crear un nuevo fondo, ya cuentan con varios parámetros configurados para su uso específico.

Fondo de Amigos

Este tipo de fondo está pensado para que pequeños grupos de personas (amigos, compañeros) que desean recaudar dinero para alguna causa común.

Fondo de Campaña

Este tipo de fondo está pensado para que aquellas entidades con una idea, pero que no dispongan de los recursos necesarios, puedan crear un fondo para que otras personas puedan financiarla a cambio de beneficios.

Fondo de Donación

Este tipo de fondo está pensado para aquellas entidades que deseen recaudar dinero para alguna causa benéfica.

Fondo Personalizado

En caso de que ninguno de los tipos de fondos anteriores se adapten a sus necesidades, puede personalizar completamente todos los parámetros de su fondo a su gusto.

MANUAL DE INSTALACIÓN

URL página de Inicio:

https://proyecto-final-blockchain.netlify.app/inicio

URL página para crear nuevo fondo:

https://proyecto-final-blockchain.netlify.app/crearFondo

URL página de todos los fondos desplegados:

https://proyecto-final-blockchain.netlify.app/fondos

URL repositorio del código fuente:

https://github.com/GianfrancoRaselli/proyecto-final-2022

BIBLIOGRAFÍA

- ¿Qué es el blockchain y qué ventajas aporta a las empresas?. Recuperado de https://www.eaeprogramas.es/blog/negocio/finanzas-economia/que-es-el-blockchain-y-que-ventajas-aporta-las-empresas
- Los cuatro beneficios que ofrece el uso del blockchain en los mercados de capitales. Recuperado de https://idbinvest.org/es/blog/digitalizacion-y-conectividad/los-cuatro-beneficios-que-ofrece-el-uso-del-blockchain-en-los
- Ethereum: qué es, cómo nació y cuáles son sus ventajas. Recuperado de https://www.rankia.com/blog/blockchain-criptomonedas-bitcoin-ethereum/3683082-ethereum-que-como-nacio-cuales-son-sus-ventajas
- ¿Qué es Ethereum?. Recuperado de https://es.cointelegraph.com/learn/what-is-ethereum-a-beginners-guide-to-eth-cryptocurrency
- Todo sobre Solidity, el lenguaje de programación de Ethereum. Recuperado de https://www.thepowermba.com/es/blog/todo-sobre-solidity-el-lenguaje-de-programacion-de-etherium
- KICKSTARTER. Recuperado de https://www.kickstarter.com/
- INDIEOGO. Recuperado de https://www.indiegogo.com/
- DONORPERFECT. Recuperado de https://www.donorperfect.com/
- CAFECITO. Recuperado de https://cafecito.app/
- BitGive. Recuperado de https://www.bitgivefoundation.org/