Proyecto Final 2022

APLICACIÓN DESCENTRALIZADA

ETHEREUM

ADMINISTRADOR DE FONDOS COMUNES

Comisión: 503 ISI

Integrante:

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

ÍNDICE

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 indexed recipient, 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)