



Documento Diseño y Análisis Funcional
E-ZUM

HOJA DE CONTROL

Proyecto	E-ZUM		
Entregable	Documento diseño y análisis funcional		
Autor	Pablo Linares Garrido		
Versión/Edición	0100	Fecha Versión	15/03/2023
Aprobado por		Fecha Aprobación	DD/MM/AAAA
		Nº Total de Páginas	13

REGISTRO DE CAMBIOS

Versión	Motivo del Cambio	Responsable del Cambio	Fecha
001	Versión inicial	autor	

1	INTRODUCCIÓN.....	4
1.1	OBJECTIVOS	4
2	Actores.....	4
3	Especificación de Casos de uso del Sistema	5
3.1	Roles (perfiles de usuario)	5
3.2	Listado Casos de uso del Sistema	5
3.3	Diagrama UML.....	6
4	Interfaces y pantallas	7
5	Detalle Casos de uso del Sistema	8
6	Entidades del sistema	12
6.1	Diagrama Entidad Relación E/R.....	12
7	Arquitectura de la solución.....	13

1 INTRODUCCIÓN

Este documento detalla las especificaciones funcionales para E-ZUM, la cual es una aplicación descentralizada que otorga al usuario diferentes ventajas relacionadas con el envío de tokens a través de la red blockchain de Ethereum.

1.1 OBJETIVOS

La plataforma posee diferentes objetivos ya que trata de solucionar varios problemas existentes dentro de las transacciones de criptomonedas entre particulares o empresas.

Tal y como lo vemos, existen 2 problemas principales a los cuales pretendemos dar solución con el desarrollo de esta plataforma descentralizada:

- 1- Actualmente, los individuos familiarizados con enviar o recibir tokens a través de la red de Ethereum, suelen poseer diferentes “wallets” desde las que ejecutan esas transacciones. Éstas “wallets” no están ubicadas en una misma plataforma, por lo que el usuario debe alternar de plataforma tantas veces requiera de usar una “wallet” distinta.
- 2- Cuando el usuario desea enviar tokens hacia una cuenta, éste necesita añadir una dirección pública completa de destino, la cual es larga y compleja. Con direcciones como ésta, el usuario puede cometer errores al enviar los tokens, ya que es ardua tarea comprobar que la dirección es la correcta. De igual modo, no existen opciones en la mayoría de los casos, de guardar estas direcciones como “direcciones a las que suele enviar tokens”, lo que implica poca eficiencia y pérdida de tiempo ya que aunque envíe tokens todos los días a la misma dirección, el usuario debe seguir proporcionando la dirección de destino.

Una vez comentados los dos puntos problemáticos, pasemos a las soluciones que E-ZUM propone.

- 1- A través de las semillas, el usuario puede tener almacenado cada una de sus “wallets” en un mismo lugar, permitiéndole realizar transacciones desde cualquiera de sus “wallets” sin necesidad de cambiar de plataforma. Además, al agregar una nueva “wallet” a la plataforma, el usuario le asigna un identificador para que al operar dentro de la misma, conozca desde que “wallet” está operando.
- 2- El usuario podrá agregar direcciones públicas y asignarles un nombre identificador. Las direcciones que va añadiendo se guardan en forma de lista de contactos o lista de confianza.
- 3- El usuario tiene la posibilidad de enviar tokens desde cualquiera de sus “wallets” a cualquier dirección pública que haya guardado de manera fácil y rápido. Solo tiene que seleccionar desde que “wallet” quiere enviar la transacción y a que dirección desea enviar los tokens. Cabe resaltar que el usuario elige la “wallet” y la dirección pública a la que enviará los tokens a través del identificador previamente asignado de ambas, por lo que mandar una transacción le será tan sencillo como seleccionar: “Mandar 0.01ETH desde: Wallet Metamask” a “Luis”, donde “0.01ETH” es la cantidad que desea enviar, “Wallet Metamask” es el identificador asignado a la wallet que añadió en la plataforma a través de las semillas y “Luis” es el identificador asignado a la dirección pública de su amigo o familiar “Luis”.
- 4- Además de las soluciones previamente comentadas, la plataforma consta de una sección con el historial de transacciones realizadas con anterioridad. Datos como desde que “wallet” se envió la transacción, la cuantía de la misma, el destinatario y la fecha y hora de la misma será la información que aparece reflejada en esta sección. Además, el usuario constará de un link para visualizar la transacción desde Etherscan.

2 Especificación de Casos de uso del Sistema

2.1 Actores del sistema

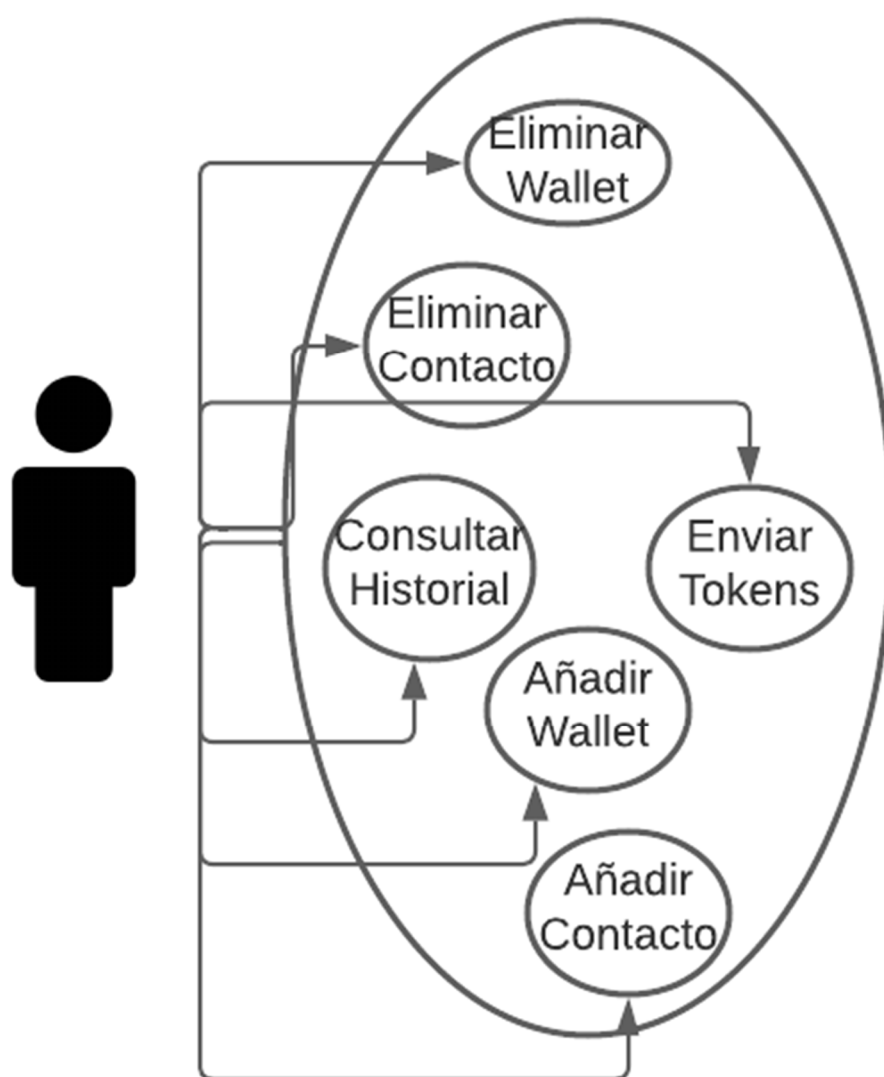
Definición de usuarios (o perfiles de usuario) que interactuarán en el sistema

1	<i>Usuario</i>
Descripción	El usuario podrá añadir sus “wallets” personales y guardar las direcciones públicas a las que suele enviar tokens, asignarles un identificador a ambas, y realizar transacciones usando los identificativos. Además, podrá consultar el historial de sus transacciones pasadas.
Comentarios	

2.2 Listado Casos de uso del Sistema

Código	Descripción	Prioridad	Dependencia
CU1	<i>Registro de usuario</i>		
CU2	Inicio de sesión		
CU3	<i>Añadir wallet</i>		
CU4	Añadir contacto (dirección pública)		
CU5	<i>Enviar Transacción</i>		
CU6	<i>Consultar Historial</i>		
CU7	<i>Eliminar wallet</i>		
CU8	<i>Eliminar contacto(dirección pública)</i>		


2.3 Diagrama UML



3 Interfaces y pantallas

¿Como funciona?

WhitePaper



¿Desarrollador?

Paso a Paso

Añade tus wallets:

Introduce las semillas de la wallet que deseas añadir:

Proporciona el nombre que deseas asignar a la wallet:

Añadir Wallet Personal

Mis wallets

Eliminar

aa

Enviar Ethers:

¿A quien quieres mandar ethers?:

juanAntonio

¿Desde que wallet deseas enviar ethers?:

aa

Cantidad de ethers que deseas enviar:

0.0202

Enviar Ethers

Guardar direcciones:

Introduce la dirección pública de la wallet que deseas guardar:

Nombre que deseas asignar a la wallet de destino:

Guardar wallet de destino

Wallets de destino

juanAntonio

Mostrar Historial de Transacciones

Historial:

Ver transacción en Etherscan

aa envió 0.0202ETH a juanAntonio, Hora: 20:17:14 Día: 15/3/2023

Ver transacción en Etherscan

aa envió 0.0202ETH a juanAntonio, Hora: 20:16:25 Día: 15/3/2023

Ésta es la pantalla principal, en la cual está disponible tanto el formulario para guardar las wallets, el formulario para guardar direcciones y el necesario para enviar la transacción. Además, un poco más abajo aparece la lista del historial de transacciones.

4 Detalle Casos de uso del Sistema

CU1	<i>Registro de Usuario</i>
Descripción	Registro de usuario
Pantalla	Pantalla formulario registro
Rol	usuario
Flujo	<i>Secuencia normal:</i> 1) Usuario accede a la plataforma 2) Completa el formulario de registro 3) Se le redirige a la página principal.
Flujo alternativo	1) Usuario accede a la plataforma 2) Completa el formulario de registro 3) El sistema detecta que el usuario que se está tratando de registrar ya está registrado. 4) Se muestra un error de que el usuario ya está registrado.

CU2	<i>Inicio de sesión</i>
Descripción	<i>Inicio de sesión</i>
Pantalla	Pantalla inicio de sesión
Rol	usuario
Flujo	<i>Secuencia normal:</i> 1) Usuario accede a la plataforma 2) Completa el formulario de inicio de sesión 3) Se le redirige a la página principal.

Flujo alternativo	1) Usuario accede a la plataforma 2) Completa el formulario de inicio de sesión 3) El sistema detecta que el usuario que está tratando de iniciar sesión no está registrado. 4) Se muestra un error de que el usuario no está registrado.
--------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CU3	<i>Inicio de sesión</i>
Descripción	<i>Inicio de sesión</i>
Pantalla	Pantalla inicio de sesión
Rol	usuario
Flujo	<i>Secuencia normal:</i> 1) Usuario accede a la plataforma 2) Completa el formulario de inicio de sesión 3) Se le redirige a la página principal.
Flujo alternativo	1) Usuario accede a la plataforma 2) Completa el formulario de inicio de sesión 3) El sistema detecta que el usuario que está tratando de iniciar sesión no está registrado. 4) Se muestra un error de que el usuario no está registrado.

CU3	Añadir wallet
Descripción	Añadir wallet a la lista de "MIS WALLETS"
Pantalla	principal
Rol	usuario
Flujo	<i>El usuario completa el formulario con las semillas correspondientes a su wallet y le asigna un identificador. Las semillas encriptadas y el identificador se almacenan en el local storage del usuario y se añade un nuevo wallet a la lista.</i>
Flujo alternativo	Al enviar el formulario retorna un error debido a que ya existe un wallet añadido con esas semilla, o las semillas no corresponden a ningún wallet, o cualquier otro error.

CU4	Añadir contacto (dirección pública)
Descripción	Añadir contacto a través de su dirección pública
Pantalla	principal
Rol	usuario
Flujo	<i>El usuario completa el formulario con la dirección pública correspondiente y le asigna un identificador. La dirección pública y el identificador se almacenan en el local storage del usuario y se añade un nuevo contacto a la lista.</i>
Flujo alternativo	Retorna un error al enviar el formulario ya que ya existe un contacto con esa dirección pública asignada o por otro error.

CU5	Enviar Transacción
Descripción	El usuario envía tokens desde una de sus wallets a un contacto guardado.
Pantalla	principal
Rol	usuario
Flujo	<i>El usuario rellena un formulario en el que elige desde que cuenta quiere enviar sus tokens, que cantidad quiere enviar y a que contacto. Al enviar los tokens, una vez la transacción esté confirmada, se añade al historial de transacciones el recibo de la transacción.</i>

Flujo alternativo	Si el usuario no dispone de los tokens + gasLimit que desea enviar en su wallet, retorna un error.
--------------------------	----------------------------------------------------------------------------------------------------

CU6	<i>Consultar Historial</i>
Descripción	El usuario revisa el historial de transacciones que ha realizado
Pantalla	principal
Rol	usuario
Flujo	<i>El usuario consulta el historial de transacciones e incluso puede acceder al link que le redirige a la transacción publicada en etherscan.</i>
Flujo alternativo	El usuario no ve ninguna transacción publicada ya que no ha realizado ninguna transacción.

CU7	<i>Eliminar wallet</i>
Descripción	El usuario elimina una de las wallets previamente añadidas
Pantalla	principal
Rol	usuario
Flujo	<i>El usuario pulsa el botón “eliminar” que se muestra junto al identificador del wallet añadido en la lista, éste automáticamente se borra del local storage y de la lista.</i>

CU8	<i>Eliminar contacto (dirección pública)</i>
Descripción	Eliminar contacto
Pantalla	principal
Rol	usuario
Flujo	<i>El usuario pulsa el botón “eliminar” que se muestra junto al identificador del contacto añadido en la lista, éste automáticamente se borra del local storage y de la lista.</i>

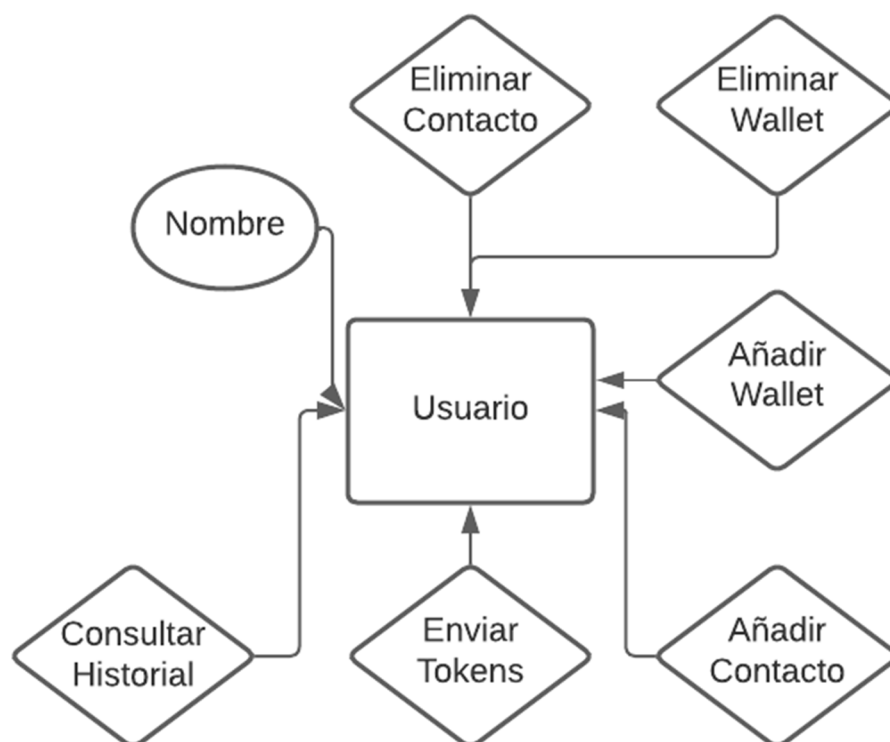
5 Entidades del sistema

Descripción entidades del sistema

1	<i>usuario</i>
Descripción	<i>Usuario que accede a la plataforma y se registra o identifica.</i>
Atributos	<i>nombre</i>
Acciones	<i>Añade wallets, Añade contactos, Elimina ambos, Envía transacciones y consulta el historial de transacciones.</i>

5.1 Diagrama Entidad Relación E/R

Descripción entidades del sistema



6. Arquitectura de la solución

La plataforma ha sido desarrollada utilizando HTML, JS, y CSS. Es un prototipo web-app, el cual hace uso de diferentes librerías como CriptoJS, Ethereum utils ... etc. En cuando a la red de prueba elegida sobre la que desarrollar la maqueta, he optado por “Goerli Tesnet” ya que es una red sobre la que ya he estado trabajando y me gusta como funciona. Además, existen bastantes faucets actualmente abiertos los cuales me aportaron mucha liquidez a la hora de realizar las pruebas.