1-3-2018

Tihomir Stoychev Stoychev

Flashback company

MINIBLOCKCHAIN

*Analisis,Diseño,Implementacion,Progreso*

# INDICE

Proyecto MINIBLOCKCHAIN

# PROGRAMACION – DISEÑO

Concepto:

Se desea realizar un proyecto de una aplicación en Java,para gestionar la creación de un intercambio de monedas, conocido también como **exchange**, el cual permetirá al usuario ingresar criptomonedas en una wallet y enviarlas a otro usuario, mediante esta wallet.

De las criptomonedas nos interesa saber su Nombre(Bitcoin,Ethereum, IOTA…), si es minable y su capitalización de mercado.

De la wallet nos interesará guardar una clave privada la cual será una cadena de 20 digitos la cual proporcionará al usuario acceso, y otra clave publica la cual tendrá un longitud de 10 digitos la cual será usada para realizar las transacciones entre usuarios.

De las transacciones nos interesa saber la fecha y el importe en dolares, aparte de las comisiones.

Muchas monedas diferentes se pueden enviar a un solo wallet y el usuario puede tener más de un monedero/wallet.

Del usuario nos interesa saber su nombre, dni, correo, documentacion, numeroTel, inversion.

Cada campo debe:

* En caso del nombre: Debe no contener ningun numero.
* En caso del dni: Deberá terminar el ultimo digito en un numero y tener una longitud igual a 9.
* En caso del email y el telefono simplemente una longitud minima de 6 y 9 , respectivamente.

Una vez el usuario tiene las monedas en su cartera o wallet, puede venderlas o comprar más entre usuarios(enviarselas mutuamente), puede enviar una criptomoneda a otro usuario, por ejemplo con 1 BTC (11.000$) puede enviarlo a otro usuario. De esta forma de su cuenta se restarian 11.000$ y del usuario destino se sumaria dicha cantidad.

La transaccion no se llevará a cabo si :

El destinatario es el mismo que el emisor.

El emisor no tiene fondos suficientes:

El destinatario no dispone de una clave publica.

El emisor no tiene la cantidad que se queire enviar de criptomonedas.

Link directo para el Diagrama de clases

<https://drive.google.com/file/d/1eDwgvOg0vmmiBxp-gt3TRpF-PSxT_7TU/view?usp=sharing>

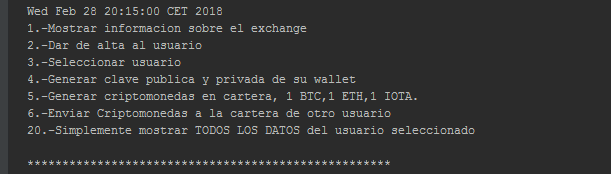
Funcionamiento MINIBLOCKCHAIN

# Descripcion detallada.

Funcionamiento:

Realmente es una aplicación sencillisima, se puede comparar con el funcionamiento de un banco pero con una pizca de trueque y encriptación, mucha encriptación.

Cuando ejecutemos el programa se nos abrirá un menu como el siguiente:



*(Es recomendable seguir el orden del menu y ejecutarlos de forma vertical, descendente, de todas formas , el programa esta preparado para el 95% de los casos posibles de error y informará al usuario si se produce algun accidente.)*

Opcion 1 :

Al elegir la opción 1, el programa mostrará al usuario los datos sobre el exchange el cual el Usuario realizará transacciones. Se mostrará patrimonio del exchange, nombre, domicilio y numero de usuarios registrados.

Opcion 2 :

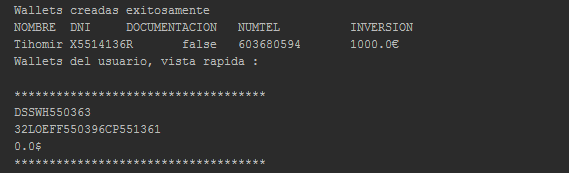
Al elegir la opcion 2, el programa nos volverá a formular una pregunta, 2 deseamos dar de alta el usuario automaticamente (con un constructor predefinido) o 1 preferimos introducir los datos y agregarlo manualmente. Si elegimos la opcion 1 ( introducir manualmente) el exchange , deberá verificar que los datos introducidos son veridicos y cumplen con un minimo de requisitos para autenticar su valideza. Una vez verificada su validez se proceden a añadir al ArrayList del exchange. De esta forma si volviesemos a llamar a la opcion 1 ahora tendriamos 1 usuario registrado en el exchange.

Opcion 3 :

Al elegir la opcion 3, para aumentar la flexibilidad del programa, el usuario podrá elegir diferentes -Usuarios- los cuales modificar, generar wallets o mostrar datos.

Opcion 4 :

Al elegir la opcion 4, antes que nada se comprueba si hay un usuario seleccionado, el cual será usado (su nombre,dni,telefono,correo) para generar claves publicas y privadas en funcion de los valores introducidos de las variables anteriormente mencionadas, junto a varios numeros aleatorios, de esta forma se asegura que no hay dos wallets iguales y no hay ambigüedad en ningun momento. Cuando se genera el wallet se añade y se muestran los datos, tal que asi:



Balance en dolares en cartera

Clave privada

Clave publica

***Opcion 5:***

Al elegir la opcion 5, se instanciaran objetos de la clase Criptomonedas, 3 de ejemplo que serán Bitcoin, Ether, y IOTA. Cada uno con sus atributos seran iniciados. El wallet del usuario pasa a ser el activo (hará falta mas en adelante) y las objetos de las Criptomoneda se añade al arrayList de Wallet, aparte de cambiar ciertos atributos y invocar al metodo calcularDolares, el cual hará un calculo de la equivalencia en dolares de cada moneda y el resultado de este será el nuevo balance del wallet. Una vez hecho todo lo anterior el programa mostrara informacion completa sobre el wallet:

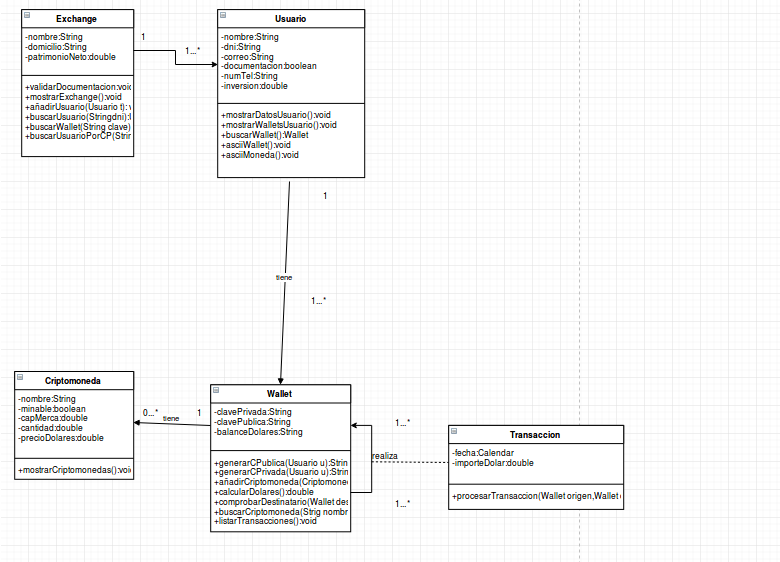


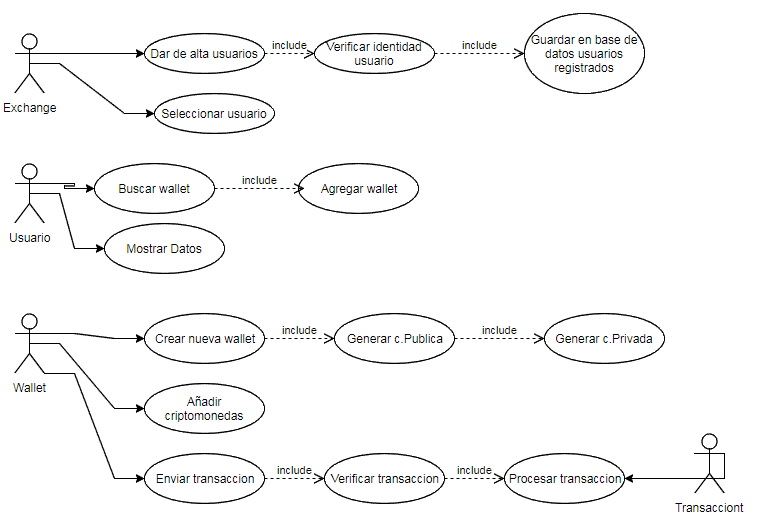
***Opcion 6:***

Al elegir la opcion 6 , se elige invocar a Satanas, el nucleo de este software gira entorno a esta opcion y la mayoria de comprobantes y metodos apuntan a esta opcion. Al seleccionar la opcion 6 se harán dos comprobaciones basicas antes de realizar la transaccion, se comprobarán los datos del destinatario y de la criptomoneda en cuestion. Del destinatario se comprobará que no sea el mismo que el emisor, se comprobará que no se inserte una cadena en blanco y tambien se comprobará si existe. Si el destinatario cumple estas 3 condiciones se seguirá con la ejecucion. Entonces se seleccionará el wallet destino y posteriormente el del usuario. Despues se introduciran datos para realizar una busqueda en el wallet del usuario sobre una criptomoneda en concreto, para la busqueda se aportará el nombre de la criptomoneda y la cantidad. Se realizará una pequeña comprobacion del a cantidad ( que sea mayor que O) y nos adentraremos al ultimo comprobante. Una vez teniendo el destinatario y la criptomoneda a enviar comprobados y estan en orden se procede a ejecutar la transaccion. La cual apuntaria las publicas del emisor y el destinatario la cantidad y realizaria operaciones de suma/resta para cada usuario, tambien calcularia otros datos como Fechas y al final se comprobaria si el usuario emisor tiene los fondos necesarios para emitir esa transaccion.

***Opcion 20***:

Se usa para mostrar los datos , todos, sobre un usuario , tal que asi:

DIAGRAMA DE CLASES

CASOS DE USO