

CriptoP2P

Se requiere construir MVP (Minimum Viable Product) para el uso Peer-to-Peer (p2p) de criptomonedas. El objetivo del producto es generar una comunidad de confianza para poder canjear criptomonedas por pesos argentinos. El sistema es similar a las billeteras que permiten realizar pagos p2p entre personas ya sea con dinero como Airtm, Skrill o Binance y el servicio P2P para comprar criptomonedas entre personas.

El servicio de P2P es muy simple, una persona pone a la venta X cantidad de determinado criptoactivo a determinado precio y otro usuario teniendo en cuenta su reputación anterior (cantidad de operación, y % de operación exitosas) selecciona el mejor vendedor y se realiza la transacción. La persona envía el dinero al vendedor y avisa por la plataforma, cuando la persona que vendió los crypto activos chequea que recibió el dinero, envía los crypto activos a la dirección correspondiente y da por finalizada la transacción. Para toda operación se requiere alguien que venda o compre y otra persona que acepte esa transacción. Las operaciones de movimientos de dinero o crypto activos quedan por fuera de la plataforma.

Al momento de informar la intención de venta o compra, el usuario indicará el precio del criptoactivo al que desea vender o comprar. Dicho precio puede tener un margen de variación de +/- 5% con respecto a la última cotización actualizada en el sistema. Cuando un usuario decide comprar o vender el criptoactivo para el que otro usuario manifestó previamente su intención, la operación se realizará tomando como precio la última cotización registrada en el sistema.

Si al momento de concretar una intención de compra, la cotización de sistema está por encima del precio manifestado por el usuario, la misma debe cancelarse. De igual manera, si al momento de concretar una intención de venta, la cotización de sistema está por debajo del precio manifestado por el usuario, la misma también debe cancelarse.

El sistema debe permitir:

1. Registrar usuarios

- Nombre, Obligatorio, Min:3, Max:30
- Apellido, Obligatorio, Min:3, Max:30
- Email, Obligatorio, Formato de email
- Dirección, Obligatorio, Min:10, Max:30
- Contraseña: al menos 1 minúscula, 1 mayúscula, 1 carac especial y min 6
- CVU MercadoPago, Obligatorio (22 dígitos)
- Dirección Billetera de CriptoActivos, Obligatorio (8 dígitos)

IMPORTANTE: Por una cuestión de simplificación tomamos la dirección como única de 8 dígitos

2. Mostrar la cotización de los siguientes crypto activos:

- ALICEUSDT
- MATICUSDT
- AXSUSDT
- AAVEUSDT
- ATOMUSDT
- NEOUSDT
- DOTUSDT
- ETHUSDT
- CAKEUSDT
- BTCUSDT
- BNBUSDT
- ADAUSDT
- TRXUSDT
- AUDIOUSDT

3. Mostrar las cotizaciones de las últimas 24hs para un cripto activo dado.

- Criptoactivo
- Día y Hora de cotización
- Cotización del Cripto Activo

4. Permitir que un usuario exprese su intención de compra/venta

- Criptoactivo
- Cantidad nominal del Cripto Activo
- Cotización del Cripto Activo
- Monto de la operación en pesos ARG
- Usuario (Nombre/Apellido)
- Operación: [COMPRA|VENTA]

5. Construir un listado donde se muestran las transacciones activas de compra/venta, es decir, las transacciones que expresan la intención de comprar o vender de un usuario.

- Día y hora del momento en que se creó la intención de compra/venta
- Criptoactivo
- Cantidad nominal del Cripto Activo
- Cotización del Cripto Activo expresa en la intención de compra/venta
- Monto de la operación en pesos ARG
- Usuario (Nombre/Apellido)
- Cantidad de operaciones (realizadas por el usuario)
- Reputación, se calcula como cantidad de puntos otorgados / cantidad de operaciones. Si la persona no ha operado se muestra "Sin operaciones".

6. Procesar la transacción informada por un usuario

- Criptoactivo
- Cantidad nominal del Cripto Activo
- Cotización del Cripto Activo

- Monto de la operación
- Usuario (Nombre/Apellido)
- Cantidad de operaciones (realizadas por el usuario)
- Reputación
- Dirección de envío:
 - Si la operación es venta, debe mostrar un CVU para que el usuario 2 haga la transferencia
 - Si la operación es compra, debe mostrar la dirección de la billetera de CriptoActivos
- ACCIÓN: “Realice la transferencia” (Si el usuario es comprador)
- ACCIÓN: “Confirmar recepción” (Si el usuario es vendedor)
- ACCIÓN: Cancelar

Si el usuario 1 o el 2 cancela la operación, no se le suma una operación realizada y se le descuenta 20 puntos de la reputación.

Si la operación es cancelada por el sistema debido a la diferencia de precios, no se le suma la operación a los usuarios pero tampoco reciben descuento de puntos por cancelación.

Si la operación se realiza dentro de los 30 minutos, suman 10 puntos y si supera los 30 minutos suman 5.

7. Dado un usuario, Informar el volumen operado de cripto activos entre dos fechas.

- Día y hora de solicitud
- Valor total operado en dólares
- Valor total operado en pesos ARG
- Activos:
 - Criptoactivo
 - Cantidad nominal del Cripto Activo
 - Cotización actual del Cripto Activo
 - Monto de la cotización en pesos ARG

La aplicación debe ser diseñada con APIs los cuales se podrían utilizar sin la necesidad de tener las pantallas.

Además se deben crear 2 servicios adicionales en la API:

- A. Listado de usuarios de la plataforma
 - a. Nombre, Apellido, Cantidad Operaciones, Reputación
- B. Listado de cotizaciones (precio cada 10 minutos)
 - a. Cripto Activo, precio, hora de actualización

El servicio B debe poder contestar en el orden de los milisegundos y puede contener valores no actualizados.

IMPORTANTE:

- TODAS las APIs deben estar correctamente documentadas con Swagger3
- La app debe ser stateless
- Las validaciones deben estar correctamente realizadas en el Backend. Se debe respetar los error code de http y manejar los errores.

APIS:

- <https://api1.binance.com/api/v3/ticker/price?symbol=BNBUSDT>
- <https://estadisticasbcra.com/api/documentacion>
- <https://www.dolarsi.com/api/api.php?type=valoresprincipales>

Requerimientos técnicos

- **Backend:**
 - Java o Kotlin / Spring Boot / Hibernate.
 - BD a elección.
 - Cache, algo manual, Redis o alguna otra.
- **Servicios:**
 - Restful.
 - Swagger
- **Infra:**
 - Github
 - Travis / github actions / circle CI
 - Docker
 - Heroku