



Tips para el trabajo final - módulo 3

Conceptos DeFi

DEX

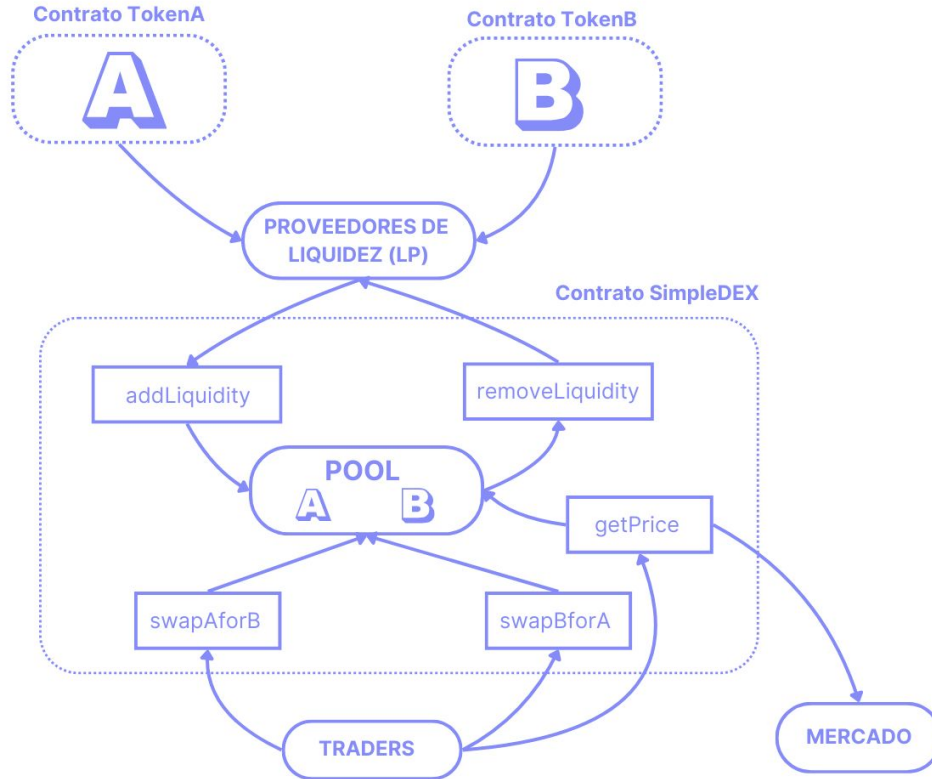
- Plataforma para intercambiar criptomonedas sin un intermediario central.
- Operaciones ejecutadas con contratos inteligentes.
- Ventajas: Mayor transparencia, seguridad y control.

POOL DE LIQUIDEZ

- Fondo compartido de dos tokens para facilitar intercambios.
- Ventajas: Intercambios rápidos y eficientes; proveedores de liquidez reciben comisiones.
- Ejemplo: Dos usuarios aportan TokenA y TokenB. Los intercambios se realizan directamente con el pool, no con una contraparte específica.



Diagrama de un DEX sencillo



Funciones a implementar en SimpleDEX

- `constructor(address _tokenA, address _tokenB)`
- `addLiquidity(uint256 amountA, uint256 amountB)`
- `swapAforB(uint256 amountAIn)`
- `swapBforA(uint256 amountBIn)`
- `removeLiquidity(uint256 amountA, uint256 amountB)`
- `getPrice(address _token)`



Consideraciones para añadir liquidez

- Cuando añado liquidez debo hacerlo en pares: una cantidad de tokens A y otra equivalente de tokens B.
- ¿Cómo transfiero tokens desde el proveedor de liquidez al contrato SimpleDEX? Revisar que función del estándar ERC20 debo utilizar.
- ¿Cómo ejecuto una función del contrato ERC20 del token desde el contrato SimpleDEX? Una pista: Al utilizar `IERC20(_tokenA)`, estamos creando una instancia que nos permite interactuar con el contrato del token ERC-20 ubicado en la dirección `_tokenA`.



Fórmula para el swap

Fórmula del Producto Constante: $(x + dx) * (y - dy) = x * y$.

Supongamos que hay 1000 unidades de TokenA (x) y 2000 unidades de TokenB (y) en el pool. Si un usuario quiere intercambiar 100 unidades de TokenA (dx):

1. La cantidad de TokenB (dy) que recibirá se puede calcular a partir de la fórmula del producto constante:
 $(x + dx) * (y - dy) = x * y$
2. Reemplazando los valores conocidos: $(1000 + 100) * (2000 - dy) = 1000 * 2000$
3. Resolviendo para dy:
 $1100 * (2000 - dy) = 2000000$
 $2000 - dy = 2000000 / 1100$
 $2000 - dy \approx 1818.18 \quad \Rightarrow \quad dy \approx 2000 - 1818.18 \approx 181.82$

Esto significa que el usuario recibirá aproximadamente 181.82 unidades de TokenB a cambio de las 100 unidades de TokenA que depositó en el pool.



Fórmula para determinar el precio actual (getPrice)

getPrice calcula el precio actual de un token en relación con el otro, basándose en la cantidad de tokens disponibles en el pool de liquidez.

Por ejemplo, si en el pool hay 500 unidades de TokenA y 1000 unidades de TokenB, el precio de 1 TokenA en términos de TokenB sería $1000 / 500 = 2$ TokenB por cada TokenA.

Primero, verifica que la dirección del token ingresado (token) sea válida, es decir, que corresponda a TokenA o TokenB. Si no es así, devuelve un mensaje de error.

Luego, si el token ingresado es TokenA, la función calcula el precio dividiendo la cantidad de TokenB en el pool entre la cantidad de TokenA. Esto nos da el precio de una unidad de TokenA en términos de TokenB.

No olvides multiplicar por 10^{18} para manejar los decimales y asegurar la precisión en los cálculos.



Ejemplos de eventos

- **LiquidityAdded:** Añadir liquidez.
- **TokensSwapped:** Intercambiar tokens.
- **LiquidityRemoved:** Retirar liquidez.