Proyecto CryptoCampo

Documentación Smart Contract

Autor: Eduardo Mannarino
Actualizado al 06/05/2022

Consideraciones generales

Se desarrollará un único contrato que será responsable de los NFT y de todas las tareas on-chain que sean necesarias (se detallarán más adelante).
Se utilizarán librerías de OpenZeppelin (https://openzeppelin.com/).

- **ERC721:** estándar para Non Fungible Token (NFT).
- **ERC721Enumerable:** Extensión para poder enumerar los tokens por propietario.
- **Ownable:** Módulo para control de acceso a funciones de administrador.
- **ReentrancyGuard:** Módulo para evitar reentradas en funciones.
- **Counters:** Utilitario para contadores.

Variables

buyFee: Porcentaje de comisión por compra de token. Usa 2 posiciones decimales (5% = 500).
canBuy: Indica si se pueden generar tokens.
canClaim: Indica si se pueden reclamar los tokens.
canTrade: Indica si se pueden intercambiar los tokens.
fundsToken: Address para indicar la moneda de cobro y pago. Tiene que ser ERC20.
fundsCollector: Address (wallet) para indicar el recolector de fondos.
feesCollector: Address (wallet) para indicar el recolector de fees.
listTokensOnSale: Array que almacena los ID de tokens en venta, para acceso secuencial.
maxBatchCount: Máxima cantidad de tokens a comprar o reclamar por lote. Este límite tiene como objetivo evitar superar la cantidad máxima de gas.
maxValueToRaise: Valor máximo del total de tokens. (Monto máximo de inversión a recibir). Usa 18 posiciones decimales (1 USD = 10 ^18).
profitToPay: porcentaje de ganancia sobre capital a devolver. Usa 2 posiciones decimales (20% = 2000).

tokenIdTracker: Manejador de IDs secuenciales para la identificación de tokens.
tokensOnSale: Mapping para almacenar los tokens en venta y su eventual precio.
totalValue: Indica el valor total de todos los tokens. Usa 18 posiciones decimales (1 USD = 10 ^18).
tradeFee: Porcentaje de comisión por intercambio de token. Usa 2 posiciones decimales (1% = 100).
validValues: Array para indicar los valores válidos de tokens. (Ej: 100, 500, 1000, etc). Usa 18 posiciones decimales (1 USD = 10 ^18).
values: Mapping para almacenar el valor de cada token.

Funciones

Buy

Descripción
Permite al usuario invertir en la plataforma mediante la generación de los NFTs correspondientes.

Parámetros
value: Valor del token (valor único para cada token). Usa 18 posiciones decimales (1 USD = 10 ^18).
amount: Cantidad de tokens a comprar.

Requerimientos
Que el usuario haya aprobado el contrato para utilizar su token de pago.
Que canBuy sea verdadero.
Que la cantidad a comprar sea menor a la cantidad máxima por lote.
Que el valor sea uno de los valores válidos.
Que el total a comprar no supere el valor total de tokens permitida.

Acción
Se suma el valor total de tokens a totalValue.
Se transfiere de la wallet del comprador a la wallet recolectora de fondos el total a adquirir. (Cantidad * Valor).
Se transfiere de la wallet del comprador a la wallet recolectora de fees el valor correspondiente a la comisión (Valor Total * buyFee / 100)
Se generan y transfieren a la wallet del comprador los NFT correspondientes.
Se emite el evento Buy.

PutOnSale

Descripción

Permite al usuario poner en venta su token a un determinado precio.

Parámetros

tokenId: ID del token a poner en venta.
price: precio fijado para la venta. Usa 18 posiciones decimales (1 USD = 10 ^18).

Requerimientos

Que canTrade sea verdadero.
Que el ID de token exista. (Generado y no quemado).
Que el token a poner en venta sea propiedad de quien ejecuta la función.

Acción

Se registra el token como disponible para la venta y su precio correspondiente.
Se emite el evento PutOnSale.

RemoveFromSale

Descripción

Permite al usuario quitar de la venta su token.

Parámetros

tokenId: ID del token a quitar de la venta.

Requerimientos

Que canTrade sea verdadero.
Que el ID de token exista. (Generado y no quemado).
Que el token a quitar de la venta sea propiedad de quien ejecuta la función.

Acción

Se quita el token como disponible para la venta.
Se emite el evento RemoveFromSale.

Trade

Descripción

Permite comprar un token a la venta.

Parámetros
tokenId: ID del token a poner a comprar

Requerimientos
Que el comprador haya aprobado el contrato para utilizar su token de pago.
Que canTrade sea verdadero.
Que el token exista. (Generado y no quemado).
Que el comprar no sea a su vez el vendedor.
Que el token se encuentre a la venta.

Acción
Se transfiere de la wallet del comprador a la wallet del vendedor el precio establecido.
Se transfiere de la wallet del comprador a la wallet recolectora de fees el monto de comisión (Valor nominal * tradeFee / 100).
Se transfiere el token del vendedor al comprador.
Se quita el token como disponible para la venta.
Se emite el evento Trade.

Claim

Descripción
Permite reclamar varios tokens, recuperando su valor más la ganancia correspondiente.

Parámetros
listTokenId: lista de ID de tokens a reclamar.

Requerimientos
Que canClaim sea verdadero.
Que la cantidad a reclamar sea menor a la cantidad máxima por lote.
Que los tokens existan. (Generados y no quemados).
Que todos los tokens sean propiedad de quien reclama.

Acción
Se resta el valor total de tokens de totalValue.
Se queman todos los tokens indicados.
Se transfiere de la wallet recolectora de fondos a la wallet del reclamador por el total de los mismos más la ganancia a pagar.
Se emite el evento Claim.