**CONTRATO DE SUBASTA:**

En el Trabajo Practico del modulo 2 se solicitaba realizar un Smart Contract de Una subasta cuyas características y consideraciones fueron indicadas en el Campus, motivo por el cual se expondrán los pasos realizados para concretar la tarea asignada.

1. Introducción

Este documento describe en detalle el funcionamiento de un contrato inteligente de subasta implementado en Solidity (versión 0.8.20) para la blockchain Ethereum. El contrato gestiona una subasta transparente con características avanzadas como extensiones de tiempo, incrementos mínimos de oferta y reembolsos automáticos.

2. Propósito

El contrato permite:

* Realizar subastas cronometradas con ETH como moneda
* Garantizar transparencia en el proceso de licitación
* Automatizar reembolsos para postores no ganadores
* Implementar mecanismos anti-snipe (extensión de tiempo)
* Gestionar comisiones automáticas del 2%

3. Estructuras de Datos Clave:

3.1 struct Bid { address bidder; ( Dirección del postor) uint amount;( Cantidad ofertada en wei)}

3.2. Mappings Principales

* bidHistory: Array histórico de todas las ofertas válidas
* accumulatedBids: Total acumulado por dirección
* userBidHistory: Historial de ofertas por usuario
* pendingWithdrawals: Fondos reembolsables por dirección

4. Variables de Estado

|  |  |  |
| --- | --- | --- |
| **Variable** | **Tipo** | **Descripción** |
| **owner** | address | Dueño del contrato |
| **deadLine** | uint | Fecha límite de la subasta |
| **commissionPercent** | uint(constante) | 2% de comisión |
| **timeExtension** | uint(constante) | 10 minutos de extensión |
| **ended** | bool | Estado de finalización |
| **highestBid** | Bid | Oferta más alta actual |

5. Funcionalidades Principales

5.1. Constructor

* Inicializa el contrato
* Establece al creador como owner
* Calcula el deadline basado en \_durationSeconds

5.2. placeBid()

Función payable para realizar ofertas:

1. Validaciones:

* Subasta activa (onlyBeforeEnd)
* Valor > 0
* Para ofertas no iniciales: debe ser 5% mayor que la actual
* El postor no puede ser el actual líder

1. Lógica:

* Primera oferta establece el mínimo
* Ofertas posteriores requieren incremento del 5%
* Extiende el tiempo si se ofrece cerca del deadline
* Emite evento NewBid

5.3. endAuction()

Finaliza la subasta:

1. Requisitos:

* Solo owner
* Deadline alcanzado
* Subasta no finalizada previamente
* Al menos una oferta válida

1. Acciones:

* Transfiere oferta ganadora (menos 2%) al owner
* Reembolsa automáticamente el 98% a perdedores
* Emite AuctionEnded

5.4. withdrawExcess()

Permite retirar fondos excedentes:

* Solo durante subasta activa
* Calcula exceso sobre el 105% del mínimo requerido
* No disponible para el postor líder actual

5.5. claimRefund()

Reclama reembolsos:

* Solo después de finalizada la subasta
* Transfiere el 98% de las ofertas acumuladas

6. Funciones de Visualización

|  |  |
| --- | --- |
| **Función** | **Descripción** |
| **getHighestBid()** | Devuelve postor y monto actual líder |
| **getBidHistory()** | Historial completo de ofertas |
| **timeRemaining()** | Tiempo restante en segundos |
| **bidsOf(address)** | Historial de ofertas por usuario |
| **totalBidOf(address)** | Suma total ofertada por usuario |

7. Eventos

|  |  |
| --- | --- |
| **Evento** | **Descripción** |
| **NewBid** | Nueva oferta registrada |
| **ExcessFundsAvailable** | Fondos excedentes detectados |
| **AuctionEnded** | Subasta finalizada con ganador |
| **PartialRefund** | Reembolso parcial ejecutado |
| **DepositWithdrawn** | Retiro completo de fondos |
| **EmergencyWithdraw** | Retiro de emergencia por owner |
| **DepositRefunded** | Reembolso automático a perdedores |

8. Mecanismos de Seguridad

8.1. Modificadores

* onlyBeforeEnd: Solo durante subasta activa
* onlyOwner: Restringe al dueño del contrato

8.2. Patrones Seguros

* Reembolsos automáticos para evitar "fondos atrapados"
* Withdraw pattern para retiros
* Mappings para seguimiento de fondos

9. Consideraciones Económicas

* Comisión del 2%: Retenida de la oferta ganadora
* Reembolsos del 98%: Para ofertas no ganadoras
* Incremento mínimo del 5%: Para nuevas ofertas

10. Flujo de Trabajo

1. Configuración:

* Owner despliega contrato con duración
* Subasta comienza inmediatamente

1. Fase de Ofertas:

* Postores envían ETH via placeBid()
* Cada oferta válida extiende el deadline si es cerca del final

1. Finalización:

* Owner ejecuta endAuction()
* Fondos distribuidos automáticamente

1. Post-Subasta:

* Postores pueden reclamar reembolsos
* Owner puede retirar fondos de emergencia

El diseño favorece la descentralización y minimiza la necesidad de intervención manual, automatizando procesos críticos como los reembolsos.