Contrato Inteligente de Subasta Segura - Subasta2.sol

Descripción General

Este contrato inteligente implementa una subasta segura y transparente en Ethereum.
La duración de la subasta es variable y la define el propietario (owner) al momento del despliegue, en minutos.
Si se realiza una oferta válida en los últimos 10 minutos, la subasta se extiende automáticamente 10 minutos, hasta un máximo igual a la duración original.

Todas las acciones importantes quedan registradas mediante eventos para mayor transparencia.
Solo el propietario puede finalizar la subasta y devolver los depósitos a los no ganadores, descontando una comisión del 2%.
El contrato está diseñado para uso académico y de pequeña escala.

Funcionalidades Principales

- **Duración variable de subasta:** El owner define la duración (en minutos) al desplegar el contrato. Si hay ofertas en los últimos 10 minutos, la subasta se extiende 10 minutos (hasta el máximo de la duración original).
- **Ofertar:** Solo se aceptan ofertas válidas (al menos 5% superiores a la oferta más alta actual).
- **Reembolso parcial:** Los participantes pueden retirar el exceso depositado sobre su última oferta válida durante la subasta.
- **Devolución de depósitos:** Al finalizar la subasta, solo el propietario puede devolver los depósitos a los no ganadores, descontando una comisión del 2%.
- **Retiro de emergencia:** El propietario puede recuperar todos los fondos del contrato solo si la subasta fue cancelada y no quedan depósitos pendientes.
- **Eventos:** Todas las acciones importantes emiten eventos para transparencia.

Variables Principales

- owner: Dirección del propietario del contrato.
- auctionEndTime: Momento en que finaliza la subasta.

- maxExtensionTime: Máxima extensión permitida (igual a la duración original).
- extendedTime: Tiempo total que se ha extendido la subasta.
- highestBidder: Dirección del mayor postor actual.
- highestBid: Monto de la oferta más alta actual.
- bidHistory: Array de todas las ofertas (dirección y monto).
- deposits: Depósitos de ETH por dirección.
- lastBid: Última oferta de cada dirección.
- bidIndex: Índice de cada postor en bidHistory.
- hasBid: Indica si una dirección ya ofertó.
- lastBidTime: Última vez que un usuario ofertó.
- ended: Verdadero si la subasta terminó.
- fundsWithdrawn: Verdadero si el owner retiró el monto ganador.
- cancelled: Verdadero si la subasta fue cancelada.

Funciones Principales

- **Constructor:** Inicializa la subasta con la duración definida por el owner (en minutos) y define al deployer como propietario.
  Parámetro: duración en minutos (ejemplo: 10080 para 7 días).
- **Ofertar:** Permite a los usuarios (excepto el owner) ofertar. La oferta debe ser al menos 5% mayor que la actual. Las ofertas en los últimos 10 minutos extienden la subasta 10 minutos (máximo igual a la duración original). Emite el evento NewBid.
  Parámetro: msg.value (ETH enviado con la oferta).
- **Reembolso Parcial:** Permite a los postores retirar el exceso depositado sobre su última oferta válida durante la subasta. Emite el evento PartialWithdrawal.
- **Devolver Depósitos a No Ganadores:** Solo el owner puede llamarla al finalizar la subasta. Devuelve los depósitos a los no ganadores, descontando una comisión del 2%. Emite los eventos DepositWithdrawn y FeeTransferred.
- **Finalizar Subasta:** Permite al owner finalizar la subasta manualmente antes del tiempo programado. Emite el evento AuctionEnded.

- **Retirar Oferta Ganadora:** Permite al owner retirar la oferta ganadora después de finalizar la subasta.
- **Cancelar Subasta:** Permite al owner cancelar la subasta si no hay ofertas. Emite el evento AuctionCancelled.
- **Retirar Depósito tras Cancelación:** Permite a los usuarios retirar su depósito si la subasta fue cancelada. Emite el evento DepositWithdrawnOnCancel.
- **Retiro de Emergencia:** Permite al owner recuperar todos los ETH del contrato solo si la subasta fue cancelada y no quedan depósitos. Emite el evento EmergencyWithdrawal.
- **Obtener Número de Ofertas:** Devuelve la cantidad de ofertas realizadas en la subasta.
- **Obtener Historial de Ofertas (Paginado):** Devuelve una página del historial de ofertas para paginación.
  Parámetros: offset (índice de inicio), limit (cantidad de ofertas a devolver).
- **Obtener Ganador:** Devuelve la dirección del mayor postor y el monto de la oferta ganadora.

Eventos

- NewBid: Se emite cuando se realiza una nueva oferta.
- AuctionEnded: Se emite cuando finaliza la subasta.
- PartialWithdrawal: Se emite cuando un usuario retira exceso de depósito.
- DepositWithdrawn: Se emite cuando un no-ganador recibe su reembolso.
- AuctionCancelled: Se emite cuando la subasta es cancelada.
- FeeTransferred: Se emite cuando el owner recibe la comisión del 2%.
- DepositWithdrawnOnCancel: Se emite cuando un usuario retira su depósito tras cancelación.
- EmergencyWithdrawal: Se emite cuando el owner recupera todos los fondos.

Seguridad y Buenas Prácticas

- Todas las funciones críticas usan modificadores para restringir acceso y asegurar el estado correcto de la subasta.
- Todas las transferencias de ETH usan call y verifican el éxito.
- Los cambios de estado se realizan antes de llamadas externas para evitar reentradas.
- Todas las acciones importantes emiten eventos para transparencia.
- El contrato utiliza Solidity 0.8.x, que incluye protección automática contra overflow/underflow.
- El código está completamente documentado en inglés y con comentarios NatSpec.
- Las longitudes de los arrays se almacenan en variables locales antes de los bucles para optimizar el uso de gas.

Limitaciones y Consideraciones de Seguridad

- **Límite de Gas:** La función withdrawDeposits podría alcanzar el límite de gas si hay demasiados postores. Para uso a gran escala, se recomienda implementar retiros individuales.
- **Retiro de Emergencia:** El owner solo puede recuperar todos los fondos si la subasta fue cancelada y no quedan depósitos pendientes.
- **Sin Modificador de Reentrancia:** El contrato sigue el patrón checks-effects-interactions, suficiente para este contexto.
- **Front-running:** El riesgo de front-running es inherente a las subastas públicas en blockchain y no se mitiga específicamente aquí.
- **Uso Previsto:** Este contrato está diseñado para uso académico y de pequeña escala. Para producción, se recomienda realizar pruebas exhaustivas y una auditoría de seguridad profesional.

Despliegue y Verificación

1. **Despliegue:**
    - Usa Remix IDE, Solidity 0.8.20, y despliega ingresando la duración en minutos como parámetro del constructor (por ejemplo, 10080 para 7 días).

2. **Verificación:**
    - En Etherscan, selecciona la versión de compilador y licencia correctas, y pega el código completo del contrato.
    - Ingresa el parámetro del constructor en formato ABI-encoded (puedes usar ABI Hashex para obtenerlo).

Conclusión

Este contrato inteligente implementa un sistema de subasta seguro y transparente, siguiendo los estándares modernos de Solidity y las mejores prácticas.
Está listo para su evaluación académica y despliegue en entornos de pequeña escala.
Para uso en producción, se recomienda realizar pruebas exhaustivas y una auditoría de seguridad profesional