

## Subasta Descentralizada Solidity – Documentación Completa

### Objetivo

Este contrato inteligente implementa una subasta descentralizada en Ethereum, permitiendo a los usuarios ofertar por un bien, gestionar reembolsos automáticos, aplicar una comisión del 2% a los retiros de no ganadores y extender el tiempo de subasta si se reciben ofertas en los últimos minutos. El diseño prioriza la seguridad, la transparencia y la facilidad de auditoría.

---

### Variables Principales

- **owner**: Dirección del creador/organizador de la subasta.
  - **highestBidder**: Dirección del mayor postor actual.
  - **highestBid**: Valor de la oferta más alta.
  - **auctionEndTime**: Timestamp de finalización de la subasta.
  - **initialEndTime**: Momento inicial de finalización (para cálculo de extensiones).
  - **maxEndTime**: Límite absoluto de extensión de la subasta.
  - **MIN\_BID\_INCREMENT\_PERCENT**: Incremento mínimo (5%) requerido para superar la mejor oferta.
  - **EXTENSION\_TIME**: Tiempo de extensión automática (10 minutos).
  - **COMMISSION\_PERCENT**: Comisión aplicada a retiros de no ganadores (2%).
  - **DURATION\_MINUTES**: Duración inicial de la subasta (7 días).
  - **MAX\_EXTENSION\_MINUTES**: Máxima extensión permitida (10 minutos).
  - **auctionEnded**: Indica si la subasta fue finalizada.
  - **pendingReturns**: Mapeo de depósitos reembolsables para cada usuario.
  - **bidHistory**: Historial de ofertas por usuario.
  - **bidders**: Lista de todos los oferentes únicos.
- 

### Funciones Principales

## **Constructor**

Inicializa la subasta con los parámetros necesarios:

- Define el owner.
- Establece la duración y los límites de extensión.

## **bid()**

Permite ofertar:

- Solo acepta ofertas mientras la subasta está activa.
- La oferta debe ser al menos 5% mayor que la actual.
- Si es la primera oferta del usuario, lo agrega a la lista de oferentes.
- Guarda la oferta en el historial.
- Acumula el monto de la oferta anterior para retiro del postor anterior.
- Si la oferta se realiza en los últimos 10 minutos, extiende la subasta hasta 10 minutos adicionales (sin superar el máximo permitido).
- **Evento:** NewBid.

## **endAuction()**

Finaliza la subasta:

- Solo el owner puede llamarla.
- Solo puede ejecutarse después de la fecha de finalización.
- Marca la subasta como finalizada.
- Transfiere el 100% de la oferta ganadora al owner.
- **Evento:** AuctionEnded.

## **withdraw()**

Permite a los no ganadores retirar sus depósitos:

- Solo después de finalizar la subasta.
- Aplica una comisión del 2% al retiro, que se transfiere al owner.
- El resto se transfiere al usuario.

- Protege contra reentrancia actualizando el estado antes de transferir fondos.

### **partialRefund()**

Permite a los participantes recuperar ofertas previas menores a la oferta más alta actual:

- Suma todas las ofertas previas menores a la actual y las marca como reembolsadas.
- El monto queda disponible para retiro en pendingReturns.

### **getWinner()**

Devuelve el ganador actual y el monto de su oferta.

### **getAllBiddersAndAmounts()**

Devuelve la lista de todos los oferentes y los montos acumulados para retiro.

### **getAllBids(address bidder)**

Devuelve el historial de ofertas de un usuario.

---

## **Eventos**

- **NewBid(address indexed bidder, uint amount):** Emitido cuando se realiza una nueva oferta.
- **AuctionEnded(address winner, uint amount):** Emitido al finalizar la subasta.

---

## **Seguridad y Robustez**

- **Validaciones estrictas:** Todas las funciones validan el estado de la subasta y los permisos del usuario.
- **Protección contra reentrancia:** El estado se actualiza antes de transferir fondos.
- **Gestión de errores:** Uso de require para manejar condiciones excepcionales.
- **Comisión automática:** El contrato calcula y transfiere la comisión al owner en cada retiro.
- **Extensión automática:** Si una oferta válida se realiza en los últimos 10 minutos, la subasta se extiende automáticamente, pero nunca más allá del máximo permitido.

- **Eventos:** Todos los cambios de estado relevantes emiten eventos para facilitar la auditoría y la interacción con interfaces de usuario.
- 

## Cumplimiento de Funcionalidades Requeridas

- **Constructor:** Inicializa todos los parámetros necesarios.
  - **Función para ofertar:** Solo acepta ofertas válidas, mayores en 5% y dentro del tiempo.
  - **Mostrar ganador:** `getWinner()` retorna el ganador y su oferta.
  - **Mostrar ofertas:** `getAllBiddersAndAmounts()` y `getAllBids()` permiten consultar todos los montos y el historial.
  - **Devolver depósitos:** `withdraw()` permite a los no ganadores retirar fondos descontando comisión.
  - **Manejo de depósitos:** Todas las ofertas quedan depositadas y asociadas a la dirección del oferente.
  - **Eventos requeridos:** `NewBid` y `AuctionEnded` informan los cambios clave.
  - **Reembolso parcial:** `partialRefund()` permite recuperar ofertas previas menores a la actual.
  - **Extensión automática:** Implementada en `bid()`.
  - **Seguridad:** Validaciones, protección contra reentrancia y manejo de errores en todas las funciones críticas.
  - **Documentación:** Este README explica variables, funciones y eventos de manera clara y completa.
- 

## Ejemplo de Flujo

1. El owner despliega el contrato.
2. Los usuarios ofertan usando `bid()`.
3. Si ofertan en los últimos 10 minutos, la subasta se extiende.
4. Al finalizar, el owner llama a `endAuction()`.

5. Los no ganadores retiran sus fondos con `withdraw()`, recibiendo el monto menos la comisión.
  6. Los usuarios pueden solicitar reembolsos parciales de ofertas previas con `partialRefund()`.
-