

Funcionalidades Requeridas del Contrato [Subasta2.sol](#)

1. Constructor

Inicializa la subasta con los parámetros necesarios para su funcionamiento, como el tiempo de duración (fijo en 7 días) y el propietario (deployer).

Referencia en el código:

```
constructor()
```

2. Función para ofertar

Permite a los participantes ofertar por el artículo.

- La oferta debe ser al menos un 5% mayor que la oferta más alta actual.
- Solo se puede ofertar mientras la subasta está activa.
- No permite que el propietario oferte ni que el mejor postor actual vuelva a ofertar.
- Controla el tiempo mínimo entre ofertas del mismo usuario (1 minuto).

Referencia en el código:

```
function bid() external payable onlyWhileActive
```

3. Mostrar ganador

Devuelve el oferente ganador y el valor de la oferta ganadora.

Referencia en el código:

```
function getWinner() external view returns (address, uint)
```

4. Mostrar ofertas

Devuelve la lista de oferentes y sus respectivos montos ofrecidos, con paginación.

Referencia en el código:

```
function getBidHistory(uint offset, uint limit) external view returns (Bid[] memory)
```

5. Devolver depósitos

Permite al owner devolver los depósitos a los oferentes no ganadores al finalizar la subasta, descontando una comisión del 2%.

Referencia en el código:

```
function withdrawDeposits() external onlyOwner onlyWhenEnded
```

6. Manejo de depósitos

Las ofertas se depositan en el contrato y se asocian a las direcciones de los oferentes.

Referencia en el código:

```
mapping(address => uint) public deposits;
```

7. Eventos requeridos

- Nueva Oferta: Se emite cuando se realiza una nueva oferta.
- Subasta Finalizada: Se emite cuando finaliza la subasta.
- Otros eventos: Para retiros parciales, retiros de depósitos, cancelación y transferencia de comisiones.

Referencia en el código:

```
event NewBid(address indexed bidder, uint amount);  
event AuctionEnded(address winner, uint winningAmount);  
event PartialWithdrawal(address indexed bidder, uint amount);  
event DepositWithdrawn(address indexed bidder, uint amount, uint fee);  
event AuctionCancelled();  
event FeeTransferred(address indexed to, uint amount);  
event DepositWithdrawnOnCancel(address indexed bidder, uint amount);
```

8. Reembolso parcial (Funcionalidad avanzada)

Durante la subasta, los participantes pueden retirar el importe por encima de su última oferta válida.

Referencia en el código:

```
function partialWithdraw() external onlyWhileActive
```

9. Consideraciones adicionales

- Uso de modificadores para controlar el acceso y el estado de la subasta.

- Extensión automática del tiempo de subasta si una oferta válida se realiza en los últimos 10 minutos, hasta un máximo de 7 días.
- Seguridad: manejo adecuado de errores y situaciones excepcionales.
- Uso de eventos para comunicar cambios de estado a los participantes.

Nota:

El contrato cumple con todas las funcionalidades requeridas y avanzadas especificadas en el enunciado, con la salvedad de que la devolución de depósitos a los no ganadores la realiza el owner, no cada usuario individualmente.
