**Explicación del Contrato** Subasta5.sol

**¿Qué realiza este contrato?**

El contrato Subasta5.sol implementa una **subasta abierta** en la blockchain de Ethereum. Permite a cualquier usuario ofertar enviando Ether, siempre que la subasta esté activa. El contrato gestiona las ofertas, depósitos, reembolsos parciales, retiro de fondos para no ganadores y para el propietario (owner), y protege contra ataques de reentrancia. Además, emite eventos para que cualquier persona pueda seguir el estado de la subasta.

---

**Variables del contrato y su propósito**

- **owner**:
  Dirección del propietario del contrato, quien tiene permisos especiales (por ejemplo, finalizar la subasta y retirar fondos).
- **auctionEndTime**:
  Momento (timestamp) en que termina la subasta. Se define al desplegar el contrato.
- **ended**:
  Booleano que indica si la subasta ha finalizado.
- **highestBid**:
  Valor de la oferta más alta realizada hasta el momento.
- **highestBidder**:
  Dirección del usuario que realizó la oferta más alta.
- **bidsByAddress**:
  Mapping que asocia cada dirección de usuario con un array de sus ofertas (en wei).
- **deposits**:
  Mapping que asocia cada dirección de usuario con el total de Ether depositado.
- **biddersList**:
  Array que almacena las direcciones de todos los usuarios que han ofertado al menos una vez.
- **locked**:
  Booleano privado usado como protección básica contra ataques de reentrancia.

---

**Eventos**

- **NewBid**:
  Se emite cada vez que un usuario realiza una nueva oferta.
- **PartialRefund**:
  Se emite cuando un usuario retira el excedente de sus depósitos.
- **AuctionEnded**:
  Se emite cuando la subasta finaliza.

---

**Modificadores**

- **onlyWhileActive**:
  Permite ejecutar funciones solo si la subasta está activa.
- **onlyOwner**:
  Permite ejecutar funciones solo al propietario del contrato.
- **noReentrancy**:
  Previene ataques de reentrancia en funciones críticas.

---

**Funciones del contrato**

- **constructor**:
  Inicializa el propietario y define el tiempo de finalización de la subasta (por defecto, 10 minutos desde el despliegue).
- **receive()**:
  Permite recibir Ether directamente al contrato y emite un evento de nueva oferta.
- **placeBid()**:
  Permite a cualquier usuario ofertar enviando Ether.
    - La oferta debe ser mayor a cero y al menos un 5% superior a la oferta más alta anterior.
    - Si el usuario es nuevo, se agrega a la lista de oferentes.
    - Si faltan menos de 10 minutos para el cierre, la subasta se extiende 10 minutos.

- **requestPartialRefund()**:
  Permite a un usuario retirar el excedente de sus depósitos, manteniendo su última oferta activa.
- **endAuction()**:
  Solo el owner puede finalizar la subasta cuando el tiempo ha terminado.
- **getWinner()**:
  Devuelve la dirección y el monto del ganador de la subasta.
- **getBiddersPaginated(uint start, uint count)**:
  Permite consultar una porción de la lista de oferentes y sus depósitos, útil para evitar problemas de gas con muchos usuarios.
- **withdrawIfNotWinner()**:
  Permite a los usuarios que no ganaron retirar su depósito, descontando una comisión del 2%.
- **ownerWithdraw()**:
  Permite al owner retirar los fondos restantes del contrato tras finalizar la subasta.

---

**Resumen de funcionamiento**

1. **Inicio:**
   El owner despliega el contrato. La subasta inicia automáticamente.
2. **Ofertas:**
   Los usuarios llaman a placeBid() enviando Ether. Cada oferta debe superar la anterior en al menos un 5%.
3. **Reembolsos parciales:**
   Los usuarios pueden retirar el excedente de sus depósitos con requestPartialRefund().
4. **Finalización:**
   Cuando el tiempo termina, el owner llama a endAuction().
5. **Retiros:**
   - Los no ganadores pueden retirar su depósito menos comisión con withdrawIfNotWinner().
   - El owner puede retirar los fondos restantes con ownerWithdraw().

---

**Seguridad**

- El contrato implementa protección básica contra reentrancia.
- Usa validaciones (require) para evitar errores y comportamientos inesperados.
- Emite eventos para transparencia.

---