[**Análisis general del modelo ER** 2](#_Toc208334397)

[**Entidades detalladas** 6](#_Toc208334398)

[**Dominio: Usuarios** 6](#_Toc208334399)

[**1) Usuario** 6](#_Toc208334400)

[**2) Billetera (Wallet)** 6](#_Toc208334401)

[**3) (Opcional) Preferencias de Notificación** 7](#_Toc208334402)

[**Dominio: NFTs/Obras** 7](#_Toc208334403)

[**1) NFT (núcleo)** 7](#_Toc208334404)

[**2) (Opcional) ValidaciónTécnicaNFT** 8](#_Toc208334405)

[**3) (Opcional) Categorías/Tags del NFT** 8](#_Toc208334406)

[**4) (Opcional) HistorialPropiedadNFT** 9](#_Toc208334407)

[**Dominio: Curaduría/Validación** 9](#_Toc208334408)

[**1) CurationReview (revisión de curaduría)** 9](#_Toc208334409)

[**2) (Opcional) ReviewAssignment (asignación/cola de revisión)** 10](#_Toc208334410)

[**Dominio: Subastas** 10](#_Toc208334411)

[**1) Entidad: Subasta** 10](#_Toc208334412)

[**2) Reglas de ofertas que impactan a Subasta (comportamiento esperado)** 11](#_Toc208334413)

[**3) Cierre de subasta (qué le pasa a la entidad Subasta al finalizar)** 11](#_Toc208334414)

[**4) Parámetros de subasta (configuración)** 11](#_Toc208334415)

[**Dominio: Ofertas** 12](#_Toc208334416)

[**1) Entidad: Oferta (Bid)** 12](#_Toc208334417)

[**2) Reglas de negocio (validación de una oferta)** 12](#_Toc208334418)

[**3) Entidad: ReservaDeFondos (FundsReservation)** 13](#_Toc208334419)

[**4) Oferta del mismo postor varias veces** 14](#_Toc208334420)

[**5) Concurrencia y desempate** 14](#_Toc208334421)

[**6) (Opcional) Entidad: IntentoDeOferta (BidAttempt / Log de validación)** 14](#_Toc208334422)

[**7) Reglas operativas y de integridad** 15](#_Toc208334423)

[**8) Relaciones con otros dominios** 15](#_Toc208334424)

[**9) Métricas y reportes que salen de Ofertas** 15](#_Toc208334425)

# **Análisis general del modelo ER**

**Dominio y entidades (qué tablas habrá y para qué)**

1. **Usuarios**
   * **User**: persona que usa la plataforma. Rol: *artista*, *curador*, *coleccionista* (y opcional *admin*). Necesario para todas las acciones (subir NFT, revisar, ofertar).
   * **Wallet**: “billetera” por usuario, con **saldo disponible** y **saldo reservado** (se reserva al hacer ofertas exitosas). El enunciado exige validar saldo antes de aceptar una oferta y reservar fondos al registrar la oferta.
2. **Obras / NFTs**
   * **NFT**: obra con metadatos (nombre, descripción, tipo de archivo), **hash único** para identidad/autenticidad, **estado** de ciclo de vida (*pendiente/aprobado/rechazado/finalizado*) y referencias a **artista** y **propietario actual**. Al subir, se generan hash y correo de confirmación; pasa a “pendiente” mientras se valida.
3. **Curaduría / Validación**
   * **CurationReview**: revisiones realizadas por **curadores** con decisión (*aprobar/rechazar*), comentarios y tiempos. Esto alimenta el reporte “Eficiencia de curadores” (volumen, tiempos, tasas).
4. **Subastas**
   * **Auction**: se **crea automáticamente** cuando un NFT pasa a aprobado; **duración estándar 72h**, **precio inicial** (sugerido por artista si existe o **precio base** de la plataforma), y estado (*activa/finalizada*). También se notifica por correo y aparece en la galería de subastas activas. Es configurable (admin puede cambiar duración y precio base).
   * **(Decisión de diseño clave)**: ¿1 NFT = 1 subasta o permitir múltiples subastas históricas? El enunciado habla de “crear una subasta” al aprobar; si quieres re-subastar no vendidas, podríamos pasar a **1:N (NFT→Subastas)** con un campo “motivo/oleada”. Si lo mantienes simple para la rúbrica, **1:1** funciona y cumple.
5. **Ofertas**
   * **Bid**: oferta con **monto**, **momento exacto**, **postor** y **subasta**. Debe pasar validaciones: saldo suficiente, subasta activa/no finalizada, y que la oferta sea **mayor** a la actual (el ejemplo narra también una regla de incremento mínimo del 5%, pero el requisito formal pide al menos que sea mayor; si quieres, ese incremento puede quedar como **parámetro** de configuración). Se debe **registrar historial** y enviar confirmación por correo.
6. **Reservas y contabilidad interna**
   * **FundsReservation**: registra **reservas** de fondos por oferta aceptada (activo/liberado/aplicado). Esto permite reembolsar a perdedores o aplicar al ganador al finalizar. Está implícito en “reservar temporalmente el saldo” y “reembolsar perdedores”.
   * **Ledger** (opcional pero recomendable): **asiento contable** por débito/crédito (compra del ganador; pago al artista). Facilita auditoría y reportes financieros.
7. **Notificaciones**
   * **EmailOutbox**: bandeja de salida para correos del sistema (recepción de NFT, activación de subasta, confirmaciones de oferta, finalización a ganador/artista/perdedores). La finalización requiere expresamente esos correos.
8. **Parámetros / Configuración**
   * **Settings**: claves como **duración estándar (72h)**, **precio base (p. ej. 0.1 ETH)** y (si decides) **% mínimo de incremento**. El enunciado pide explícitamente que duración y precio base sean configurables por el administrador.
9. **(Para reportes)**
   * Con lo anterior, salen 4 vistas/consultas:  
     **Eficiencia curadores**, **Actividad de coleccionistas**, **Valorización de artistas en el tiempo**, **Métricas por período**. Los campos (tiempos, conteos, montos, tasas) están descritos en el enunciado.

**Alcance**: todo en **capa BD** (T-SQL, transacciones, SPs, triggers), sin app.

**Relaciones (cardinalidades en “modo hablado”)**

* **User 1—1 Wallet** (una billetera por usuario).
* **User (artista) 1—N NFT** (un artista puede crear muchas obras).
* **User (propietario) 1—N NFT** (propiedad se transfiere al ganador al finalizar).
* **NFT 1—N CurationReview** (pueden existir varias revisiones; sólo la última decisión válida cambia el estado).
* **NFT 1—1 Auction** (simplificado) **o** **NFT 1—N Auction** (si permites re-subastas).
* **Auction 1—N Bid** (múltiples ofertas por subasta).
* **User (coleccionista) 1—N Bid** (un usuario hace muchas ofertas).
* **User 1—N FundsReservation** y **Auction 1—N FundsReservation** (reservas ligadas a postor y subasta).
* **Auction 1—N Ledger**, **User 1—N Ledger** (asientos de débito/crédito en finalización).
* **EmailOutbox** se relaciona lógicamente con eventos (subida, aprobación, oferta, finalización).

**Ciclos de vida y reglas (para estados y procesos)**

1. **Creación/validación de NFT**
   * *Subir* → **PENDIENTE** (hash + correo) → *Curador aprueba/rechaza* → **APROBADO** (si aprueba) **o** **RECHAZADO** (si no). Aprobado **dispara subasta**.
2. **Subasta automática**
   * Al pasar a **APROBADO**: crear subasta (72h, precio inicial sugerido o base), estado **ACTIVA**, notificar artista y coleccionistas, mostrar en galería.
3. **Ofertas**
   * Validar: **saldo suficiente**, **subasta activa y vigente**, **monto mayor** a precio actual (si decides, aplicar **incremento mínimo configurable**). Registrar **historial**, **reservar fondos** y **confirmar por correo**.
4. **Finalización (batch cada ~5 minutos)**
   * Detectar vencidas, para cada subasta: **ganador = mayor oferta** (primera en empate), **transferir NFT** al ganador, **pagar artista**, **reembolsar perdedores**, **marcar finalizada**, **enviar correos** (ganador/artista/perdedores).

**Claves, restricciones y trazabilidad (qué cuidar en el diseño)**

* **Identidad única de obra**: hash de contenido **único** para detectar duplicados.
* **Estados y fechas coherentes**: subasta end > start; una subasta sólo puede estar **ACTIVA** si el NFT está **APROBADO**.
* **Integridad financiera**: saldo ≥ 0, reservado ≥ 0; **oferta** sólo si hay **saldo disponible** suficiente; reservas pasan a **APLICADO** (ganador) o **LIBERADO** (perdedores).
* **Historial público de ofertas** y **auditoría** (reservas/ledger) para soportar reportes.
* **Configuración flexible**: **duración estándar** y **precio base** administrables (parámetros).
* **Alcance técnico**: todo se implementa con **T-SQL** (SPs, triggers, transacciones, funciones), **sin app**.

**Decisiones abiertas (valen la pena acordarlas antes del DDL)**

* **Una o varias subastas por NFT** (re-subastar no vendidas). Recomendación: empezar con **1:1** y dejar **1:N** como extensión.
* **Regla de incremento mínimo**: el ejemplo usa **5%**; el requisito formal exige “mayor que la actual”. Podemos dejar un **parámetro opcional de incremento mínimo** para alinearlo con la narrativa.
* **Comisiones de la plataforma**: no se mencionan; si decides agregar, impacta **Ledger** y **liquidación**.
* **Múltiples roles por usuario** (ej. un curador también colecciona): si lo permites, modela **UserRole** (N:M) en lugar de un único rol.

Si te parece bien este mapa conceptual, pasamos a dibujar el **ER a nivel lógico** (nombres de entidades/relaciones y cardinalidades) y luego a tu **diccionario de datos** (atributos, tipos, dominios y reglas), que además la rúbrica pide explícitamente.

# **Entidades detalladas**

## **Dominio: Usuarios**

### **1) Usuario**

Representa a cualquier persona que usa la plataforma: **artista, curador, coleccionista** (y, si quieres, admin). Estas categorías son necesarias para procesos y reportes (curadores; actividad de coleccionistas).

**Atributos clave**

* **IdUsuario**: identificador interno, único.
* **NombreCompleto**: para reportes y notificaciones.
* **Email**: único; contacto y login; aparece en reportes de coleccionistas.
* **FechaRegistro**: soporta métricas históricas (“antigüedad”).
* **EstadoCuenta** (*ACTIVO/BLOQUEADO*): útil para administración y consistencia.
* **Rol (modelo simple)**: *ARTISTA / CURADOR / COLECCIONISTA / ADMIN*. Suficiente si cada usuario tiene **un** rol principal.
  + **—o—**
* **Roles (modelo flexible N:M)**: si un usuario puede ser curador y también coleccionista, usa:
  + **Rol** (catálogo): CódigoRol, Nombre, Descripción.
  + **UsuarioRol** (puente): IdUsuario, CódigoRol, FechaAsignación (opcional).

**Reglas esenciales**

* Email **único**.
* Todo Usuario debe tener **al menos un rol** válido (según el modelo).
* Estados y fechas **no nulos** (especialmente FechaRegistro para reportes).

### **2) Billetera (Wallet)**

Necesaria porque **antes de aceptar una oferta** se valida saldo; además, al ofertar se **reserva temporalmente** y al finalizar se **descuenta/aplica** o **reembolsa**. El enunciado lo exige explícitamente.

**Atributos clave**

* **IdBilletera**: identificador interno.
* **IdUsuario**: relación 1:1 (una billetera por usuario).
* **SaldoDisponible (ETH)**: fondos líquidos.
* **SaldoReservado (ETH)**: suma de ofertas vigentes aceptadas; no se puede usar para otras ofertas.
* **Moneda**: “ETH” (puede ser catálogo si contemplas multicurrency a futuro).
* **ActualizadoEn**: última actualización, para auditoría y concurrencia.

**Reglas esenciales**

* **SaldoDisponible ≥ 0** y **SaldoReservado ≥ 0**.
* **DisponibleUsable = SaldoDisponible − SaldoReservado** (debe ser ≥ monto ofertado al ofertar).
* Relación **obligatoria** con Usuario (no hay billeteras huérfanas).

### **3) (Opcional) Preferencias de Notificación**

No es obligatorio por la rúbrica, pero ayuda porque el flujo incluye **emails** en varios puntos (recepción NFT, confirmación de oferta, cierre de subasta a ganador/artista/perdedores).

* **IdUsuario**
* **Canales** (email como mínimo)
* **Opt-in/Opt-out por evento** (aprobación NFT, subasta creada, oferta aceptada, subasta finalizada)

## **Dominio: NFTs/Obras**

### **1) NFT (núcleo)**

**Propósito:** representar la obra registrada por el artista y su ciclo de vida (subida → validación técnica → curaduría → subasta → finalización).

**Atributos sugeridos**

* **IdNFT**: identificador interno único.
* **Artista** (ref. a Usuario): autor que sube la obra.
* **PropietarioActual** (ref. a Usuario): cambia al **ganador** al finalizar la subasta.
* **Nombre** y **Descripción**: metadatos visibles. Se piden explícitamente al crear el NFT.
* **TipoContenido** (MIME, ej. image/png): parte de la verificación técnica.
* **TamañoArchivo** y **Dimensiones** (ancho/alto en px): validar límite de MB y mínimos de dimensión.
* **HashArchivo** (SHA-256 o similar): **identidad única** de la obra para evitar duplicados.
* **Ubicación/URI del archivo** (si se almacena fuera de la BD): para enlazar el recurso.
* **PrecioSugeridoETH** (opcional): el artista puede proponerlo; si se aprueba, puede usarse como base de subasta.
* **EstadoNFT**: {**Pendiente**, **Aprobado**, **Rechazado**, **EnSubasta**, **Finalizado**}.
  + Al subir: **Pendiente** + email de confirmación.
  + Al aprobar: se **crea subasta automáticamente** (72h, notifica).
  + Al finalizar: **transferencia de propiedad** al ganador.
* **Fechas clave**: **CreadoEn**, **AprobadoEn**, **ActualizadoEn** (soportan reportes y auditoría).

**Reglas clave**

* **Hash único** por NFT (impide copias).
* No pasar a **Aprobado** si la **validación técnica** no cumple (formato/tamaño/dimensiones/metadatos).
* Al pasar a **Aprobado** ⇒ **crear subasta** (72h, precio base o sugerido) y **notificar**.
* Al finalizar subasta ⇒ **cambiar PropietarioActual** al ganador y notificar (ganador, artista, perdedores).

### **2) (Opcional) ValidaciónTécnicaNFT**

**Propósito:** dejar trazabilidad de la verificación automática antes/además de la curaduría.

**Atributos sugeridos**

* **IdNFT**
* **FormatoVálido / TamañoVálido / DimensionesVálidas / MetadatosCompletos** (sí/no)
* **Observaciones** (mensaje de error si falla)
* **FechaValidación**

**Reglas**

* Si alguna validación falla, el NFT permanece **Pendiente/Rechazado**, se **notifica** y puede **corregirse** para reintentar.

### **3) (Opcional) Categorías/Tags del NFT**

**Propósito:** segmentar gustos y **notificar a coleccionistas interesados** (ej. “cyberpunk”).

* **Tag** (catálogo) y **NFTTag** (relación N:M).
* Útil para emails tipo “nueva subasta disponible en tu estilo”.

### **4) (Opcional) HistorialPropiedadNFT**

**Propósito:** auditoría y analítica (quién tuvo la obra y cuándo).

* **IdNFT**, **UsuarioAnterior**, **UsuarioNuevo**, **Motivo** (venta/subasta), **FechaCambio**.
* Se alimenta en la **finalización** de la subasta (venta) y otras transferencias.

## **Dominio: Curaduría/Validación**

### **1) CurationReview (revisión de curaduría)**

**Propósito:** registrar cada revisión humana que hace un **curador** a un NFT **pendiente**, con su decisión, criterios y tiempos. El enunciado pide que el equipo valide **originalidad** y **calidad técnica**, y **apruebe o rechace con comentarios**; además, debe **notificar por email** y, si aprueba, **cambiar estado y crear una subasta**.

**Atributos clave**

* **IdRevisión**: identificador interno.
* **IdNFT**: NFT revisado.
* **IdCurador**: usuario con rol de curador que realizó la revisión.
* **Decisión**: {**APROBAR**, **RECHAZAR**}.
* **Comentarios**: explicación breve (motivo de rechazo o notas de aprobación).
* **OriginalidadOK** (sí/no): verificación de que **no es copia**.
* **CalidadTécnicaOK** (sí/no): cumple estándares técnicos de la plataforma.
* **InicióRevisiónEn** / **CerróRevisiónEn**: marcas de tiempo para medir **tiempo de revisión** (insumo directo del reporte de “Eficiencia de curadores”).
* **NotificadoAlArtista** (sí/no) y **FechaNotificación**: respaldo de que se envió el resultado por email.

**Reglas y notas**

* Un **NFT** puede tener **múltiples revisiones** (rechazos y reintentos); el **estado actual** del NFT se deriva de la **última** revisión válida.
* Si **Decisión = APROBAR** ⇒ el sistema **cambia estado** del NFT y **crea subasta automática** (y notifica).
* Si **Decisión = RECHAZAR** ⇒ el sistema permite **corregir datos y reenviar** a validación (nuevo ciclo).

**Por qué estos campos:** alimentan directamente el reporte “**Eficiencia de curadores**”: volumen, **tiempos promedio**, tasa **aprobados vs. rechazados**, y **consistencia**.

### **2) (Opcional) ReviewAssignment (asignación/cola de revisión)**

**Propósito:** modelar la **asignación** de NFTs pendientes a curadores (útil si gestionas colas/SLA).

**Atributos sugeridos**

* **IdAsignación**, **IdNFT**, **IdCuradorAsignado**
* **AsignadoEn**, **AceptadoEn** (cuando el curador “toma” el caso), **ExpiraEn** (si hay SLA)
* **EstadoAsignación**: {PENDIENTE, ACEPTADA, REASIGNADA, VENCIDA}

**Valor agregado:** permite medir el **tiempo de espera** desde que el NFT queda pendiente hasta que un curador lo toma, no sólo el tiempo “de revisión”, dando una lectura más fina del flujo y cuellos de botella (útil para tu análisis de eficiencia).

## **Dominio: Subastas**

### **1) Entidad: Subasta**

**Propósito:** abrir la puja pública de un NFT aprobado, con reglas claras de tiempo, precio y estado del concurso. La subasta se **crea automáticamente** al aprobar un NFT, con **duración estándar (p. ej., 72 h)** y **precio inicial** (sugerido por el artista o un precio base configurable). Además, se notifica y aparece en la galería de subastas activas.

**Atributos recomendados**

* **IdSubasta**: identificador interno único.
* **IdNFT**: referencia al NFT subastado (creada al aprobarlo). **Cardinalidad**: simple para el curso → **1:1 (NFT—Subasta)**; si quieres re-subastar no vendidas, pasa a **1:N**.
* **Inicio** / **Fin**: marcas de tiempo. **Restricción:** Fin > Inicio. (El ejemplo narra un “job” que detecta subastas terminadas a los 5 min para cerrarlas).
* **DuraciónHoras** (opcional): para auditoría/configuración (p. ej., 72 h).
* **PrecioInicial**: tomado del **precio sugerido** del artista si existe; si no, del **precio base** de la plataforma (configurable por admin).
* **PrecioActual**: monto de la mejor oferta vigente.
* **LíderActual** (IdUsuario): postor que encabeza la subasta (se actualiza al recibir una oferta válida).
* **EstadoSubasta**: {**ACTIVA**, **FINALIZADA**, (opcional **CANCELADA**)}.
* **CreadaEn** / **ActualizadaEn**: trazabilidad operativa.

**Reglas de negocio (ligadas a subasta)**

* **Creación automática** al aprobar NFT, con **parámetros configurables** (duración y precio base).
* **Visibilidad/galería**: la subasta aprobada aparece en listados y puede disparar notificaciones a interesados.
* **Cierre automático**: un proceso (p. ej., cada 5 min) detecta subastas vencidas y ejecuta el cierre.

### **2) Reglas de ofertas que impactan a Subasta (comportamiento esperado)**

Aunque la **oferta** es otra entidad, estas reglas definen cómo **Subasta** evoluciona.

* **Validez temporal y de estado**: solo acepta pujas si la subasta está **ACTIVA** y dentro de **Inicio–Fin**.
* **Regla de incremento mínimo** (opcional pero usada en el ejemplo): la oferta debe ser **mayor** que el precio actual y al menos **+5 %** sobre el precio vigente. El sistema lo valida al instante.
* **Saldo suficiente**: antes de aceptar, la plataforma verifica **saldo disponible**; si falta, **rechaza** la oferta.
* **Historial y concurrencia**: registrar cada oferta y proteger la actualización de **PrecioActual/LíderActual** ante ofertas simultáneas. (La rúbrica pide manejo de concurrencia e historial).
* **Empates**: si dos ofertas tienen **mismo monto**, prevalece la que llegó **primero** (orden de llegada).

### **3) Cierre de subasta (qué le pasa a la entidad Subasta al finalizar)**

* **Determinar ganador**: el de **mayor oferta** (si hay empate, gana la oferta **más temprana**).
* **Transferir propiedad del NFT** al ganador y **pagar** al artista; **reembolsar** a perdedores; **marcar FINALIZADA**. (Todo automatizado por el proceso de cierre).
* **Sin ofertas**: la subasta se **finaliza sin transacciones** y se notifica al artista.
* **Notificaciones**: emails a ganador y artista (y reembolsos informados a perdedores).

### **4) Parámetros de subasta (configuración)**

Puedes modelarlos como **configuración administrativa** (fuera de Subasta) pero son parte esencial del dominio:

* **Duración estándar** (p. ej., 72 h).
* **Precio base** de arranque cuando no hay precio sugerido.
* **% Mínimo de incremento por oferta** (si decides aplicarlo).

La rúbrica exige **configuración flexible** y reglas de ofertas/concurrencia.

## **Dominio: Ofertas**

### **1) Entidad: Oferta (Bid)**

**Propósito:** registrar cada puja **aceptada** sobre una subasta y sostener el historial que determina al ganador.

**Atributos sugeridos**

* **IdOferta**: identificador interno.
* **IdSubasta**: referencia a la subasta.
* **IdPostor**: usuario que oferta (coleccionista).
* **Monto** *(ETH)*: valor ofertado.
* **ColocadaEn** *(UTC)*: sello de tiempo exacto.
* **EsLíderTrasOferta** *(sí/no)*: indica si esta oferta quedó líder en ese instante (útil para consultas rápidas).
* **Observaciones** *(opcional)*: mensajes operativos (“incremento aplicado”, “auto-puja rechazada”, etc.); puede quedarse vacío si sólo guardas ofertas válidas.

**Notas de modelado**

* **Oferta** sólo debe almacenar **pujas válidas** (que superaron todas las verificaciones).
* Los intentos **rechazados** (por saldo, tiempo, reglas) se pueden registrar en una tabla de **IntentosDeOferta** (ver §3) para auditoría sin “ensuciar” el historial oficial.

**Cardinalidades**

* **Subasta 1—N Oferta** (una subasta recibe muchas ofertas).
* **Usuario 1—N Oferta** (un usuario realiza muchas ofertas).

### **2) Reglas de negocio (validación de una oferta)**

Secuencia recomendada (todas deben cumplirse):

1. **Estado/tiempo de la subasta**
   * La subasta debe estar **ACTIVA** y la hora actual dentro de [Inicio, Fin).
2. **Elegibilidad del postor**
   * El postor debe tener **billetera** activa.
   * (Opcional) Prohibir que el **artista** oferte su propio NFT.
   * (Opcional) Limitar a usuarios **bloqueados** o con verificaciones pendientes.
3. **Monto mínimo requerido**
   * La oferta debe ser **mayor** al **PrecioActual**.
   * (Opcional, recomendable) **Incremento mínimo** configurable (p. ej., +5% sobre PrecioActual).
   * Desempates: si dos ofertas llegan con **igual monto**, gana la **más temprana** (ver §5).
4. **Saldo suficiente**
   * **DisponibleUsable = SaldoDisponible − SaldoReservado** del postor debe ser **≥ Monto** (o ≥ **Delta** si ya tenía una oferta previa en la misma subasta; ver §4).
5. **Aceptación atómica**
   * Registrar la oferta, **ajustar líder y precio** de la subasta, y **reservar fondos** (ver §3) en una misma operación atómica (transacción) para evitar carreras.
6. **Notificación**
   * Confirmar al postor (y opcionalmente al líder anterior si fue desplazado).

### **3) Entidad: ReservaDeFondos (FundsReservation)**

**Propósito:** inmovilizar fondos del postor mientras la subasta está activa; al finalizar, se **aplican** (ganador) o se **liberan** (perdedores).

**Atributos sugeridos**

* **IdReserva**
* **IdUsuario** (postor)
* **IdSubasta**
* **IdOferta** *(opcional pero útil para trazar de qué oferta nace la reserva)*
* **Monto** *(ETH)*
* **EstadoReserva**: **ACTIVA** (vigente), **LIBERADA** (devuelta), **APLICADA** (descontada al cierre)
* **CreadaEn / ActualizadaEn** *(UTC)*

**Reglas**

* Al aceptar una oferta ⇒ crear/actualizar **reserva ACTIVA**.
* Al **finalizar** la subasta:
  + Ganador: reservas ⇒ **APLICADA**; descontar de saldo y abonar al artista.
  + Perdedores: reservas ⇒ **LIBERADA**; devolver al saldo disponible.
* Mantener consistencia con **Wallet**: SaldoReservado refleja la suma de reservas **ACTIVAS** del usuario.

### **4) Oferta del mismo postor varias veces**

Para no sobre-reservar, manejar por **diferencial (Δ)**:

* Si el mismo usuario mejora su oferta en la **misma subasta**:
  + Reserva adicional = MontoNuevo − MontoAnteriorDelMismoPostor.
  + Actualizar **SaldoReservado** sólo por el **Δ**.
* Si pierde el liderazgo más tarde, **no** se liberan reservas hasta el cierre (siguen “comprometidas” en esa subasta).
  + (Opcional) Política alternativa: liberar la oferta anterior del mismo postor y reservar sólo la última; es más compleja y puede permitir “juegos” de liquidez. La opción **conservadora** es mantener reservas acumuladas por postor/subasta.

### **5) Concurrencia y desempate**

* Las actualizaciones de **PrecioActual / LíderActual** deben ser **exclusivas** mientras se valida e inserta la oferta (bloqueo/serialización a nivel de subasta).
* **Orden determinista** para decidir líder:
  1. **Monto** (mayor primero),
  2. **ColocadaEn** (más temprano primero),
  3. **IdOferta** (menor primero) como último desempate.
* Rechazar (o postergar) cualquier oferta que llegue mientras se está aplicando otra, si conlleva inconsistencia; luego reintentar validación con el **nuevo PrecioActual**.

### **6) (Opcional) Entidad: IntentoDeOferta (BidAttempt / Log de validación)**

**Propósito:** registrar **todos** los intentos (válidos o no) para auditoría, UX y soporte.

**Atributos sugeridos**

* **IdIntento**, **IdSubasta**, **IdUsuario**, **MontoPropuesto**, **IntentadaEn**
* **Resultado**: **ACEPTADA** / **RECHAZADA**
* **MotivoRechazo**: *SubastaNoActiva*, *FueraDeTiempo*, *SaldoInsuficiente*, *IncrementoInsuficiente*, *AutoPuja*, etc.
* **IdOferta** (si terminó aceptada)

**Valor agregado**

* Métricas de **intentos fallidos**, razones y horarios pico.
* Soporte al usuario (“tu oferta falló por saldo insuficiente”).
* No se mezcla con el historial oficial (**Oferta**).

### **7) Reglas operativas y de integridad**

* **Moneda y precisión**: ETH con precisión alta (sin redondeos que distorsionen).
* **UTC** para todas las marcas de tiempo.
* **No permitir** cancelar ofertas ya aceptadas (evita abuso).
* **Prohibir auto-pujas del artista** (si lo decides).
* **Visibilidad**: el historial de ofertas de una subasta debe ser consultable (monto y hora; identidad del postor puede mostrarse según política).
* **Privacidad**: cuidado con exponer correos/nombres; usa “alias” si el curso lo prefiere.

### **8) Relaciones con otros dominios**

* **Oferta → Subasta** (N:1) actualiza **PrecioActual** y **LíderActual**.
* **Oferta → ReservaDeFondos** (1:1 o 1:N si reservas diferenciales) mantiene la inmovilización de saldo.
* **ReservaDeFondos / Oferta → Wallet** ajusta **SaldoReservado** (al ofertar) y **SaldoDisponible** (al finalizar).
* **Cierre de subasta** consume todo: determina ganador por **Oferta** más alta (y más temprana en empate), aplica reservas, paga al **artista** y transfiere el **NFT**.

### **9) Métricas y reportes que salen de Ofertas**

* **Actividad de coleccionistas**: total ofertado, número de subastas donde participó, % subastas ganadas, montos máximos, horarios de actividad.
* **Dinámica de subasta**: número de ofertas por subasta, pujas por hora, “sniping” de último minuto (si decides medir).
* **Eficiencia de validación financiera**: tasa de intentos rechazados por **saldo** o **incremento**.

# **Diccionario de datos — Plataforma de subastas de NFT (ArteCrypto)**

Versión: v1 • Alcance: modelo lógico del ER actual • Notación de tipos: SQL Server sugerido • Tiempos: **UTC**

## **Tabla: USER**

**Propósito:** Identificar a cualquier persona en la plataforma (artistas, curadores, coleccionistas, admin).

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| UserID | bigint | NO | — | **PK** | Identificador único del usuario. |
| FullName | varchar(100) | NO | — |  | Nombre completo. |
| Email | varchar(100) | NO | — | **UK** | Correo único para contacto e inicio de sesión. |

**Índices sugeridos:** IX\_USER\_Email (Email) UNIQUE.

## **Tabla: ROLE**

**Propósito:** Catalogar roles del sistema.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción** |
| --- | --- | --- | --- | --- | --- |
| RoleID | bigint | NO | — | **PK** | Identificador del rol. |
| Name | varchar(100) | NO | — | **UK opcional** | Nombre legible (p. ej., ARTIST, CURATOR, COLLECTOR, ADMIN). |

## **Tabla: USER\_ROLE**

**Propósito:** Asignación **M:N** de roles a usuarios.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| UserID | bigint | NO | — | **FK→USER(UserID)** | Usuario. |
| RoleID | bigint | NO | — | **FK→ROLE(RoleID)** | Rol. |
| Asignacion | datetime | NO | SYSUTCDATETIME() |  | Fecha/hora de asignación. |

**Claves/Índices:** **PK compuesta (UserID, RoleID)** *o* UK(UserID,RoleID) para evitar duplicados.

## **Tabla: WALLET**

**Propósito:** Custodiar fondos; distinguir saldo disponible vs. reservado por ofertas vigentes.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| WalletID | bigint | NO | — | **PK** | Identificador de la billetera. |
| UserID | bigint | NO | — | **FK→USER(UserID), UK** | 1:1 con usuario. |
| Balance | decimal(38,18) | NO | 0 | **CK** >= 0 | Fondos líquidos del usuario (ETH). |
| Reserved | decimal(38,18) | NO | 0 | **CK** >= 0 | Fondos inmovilizados por subastas activas (ETH). |
| UpdatedAt | datetime | NO | SYSUTCDATETIME() |  | Última actualización. |

**Invariantes:** SaldoUsable = Balance − Reserved ≥ 0.

**Índices:** UQ\_WALLET\_UserID (UserID).

## **Tabla: NFT**

**Propósito:** Representar la obra (activo digital) y su ciclo de vida.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| NFTID | bigint | NO | — | **PK** | Identificador del NFT. |
| ArtistID | bigint | NO | — | **FK→USER(UserID)** | Autor/creador. |
| CurrentOwnerID | bigint | SÍ | — | **FK→USER(UserID)** | Propietario actual (se asigna al ganador al finalizar); puede inicializarse al artista. |
| Name | varchar(160) | NO | — |  | Nombre visible del NFT. |
| Descripcion | varchar(max) | SÍ | — |  | Descripción de la obra. |
| ContentType | varchar(100) | NO | — |  | MIME (ej. image/png). |
| HashCode | char(64) | NO | — | **UK** | Hash SHA‑256; identidad única del contenido. |
| FileSizeBytes | bigint | SÍ | — |  | Tamaño de archivo en bytes. |
| WidthPx | int | SÍ | — |  | Ancho (px). |
| HeightPx | int | SÍ | — |  | Alto (px). |
| SuggestedPrice | decimal(38,18) | SÍ | — |  | Precio sugerido (ETH). |
| Status | varchar(20) | NO | 'PENDING' | **CK** ∈ {PENDING, APPROVED, REJECTED, FINALIZED} | Estado del ciclo. |
| CreatedAt | datetime | NO | SYSUTCDATETIME() |  | Creación. |
| ApprovedAt | datetime | SÍ | — |  | Fecha de aprobación (si aplica). |

**Índices:** UQ\_NFT\_Hash (HashCode); IX\_NFT\_Artist (ArtistID, CreatedAt).

## **Tabla: CURATION\_REVIEW**

**Propósito:** Registro de revisiones humanas por curadores.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| ReviewID | bigint | NO | — | **PK** | Identificador de la revisión. |
| NFTID | bigint | NO | — | **FK→NFT(NFTID)** | NFT revisado. |
| CuratorID | bigint | NO | — | **FK→USER(UserID)** | Curador responsable. |
| Decision | varchar(10) | NO | — | **CK** ∈ {APPROVE, REJECT} | Decisión. |
| Comment | varchar(max) | SÍ | — |  | Comentario u observaciones. |
| StartedAt | datetime | NO | SYSUTCDATETIME() |  | Inicio de revisión. |
| ReviewedAt | datetime | SÍ | — | **CK opcional**: ReviewedAt ≥ StartedAt | Fin de revisión. |

**Índices:** IX\_CurationReview\_NFT (NFTID, ReviewedAt).

## **Tabla: AUCTION**

**Propósito:** Subasta pública de un NFT aprobado.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| AuctionID | bigint | NO | — | **PK** | Identificador de subasta. |
| NFTID | bigint | NO | — | **FK→NFT(NFTID), UK** | Relación 1:1 con NFT (modelo base). |
| StartAt | datetime | NO | — |  | Inicio de subasta (UTC). |
| EndAt | datetime | NO | — | **CK**: EndAt > StartAt | Fin de subasta (UTC). |
| StartingPrice | decimal(38,18) | NO | — | **CK** > 0 | Precio inicial (ETH). |
| CurrentPrice | decimal(38,18) | NO | — | **CK** ≥ StartingPrice | Precio líder actual (ETH). |
| CurrentLeaderID | bigint | SÍ | — | **FK→USER(UserID)** | Usuario líder actual. |
| Status | varchar(15) | NO | 'ACTIVE' | **CK** ∈ {ACTIVE, FINALIZED, CANCELED} | Estado de la subasta. |

**Índices:** UQ\_Auction\_NFT (NFTID); IX\_Auction\_Status\_End (Status, EndAt).

## **Tabla: BID**

**Propósito:** Historial oficial de ofertas aceptadas en una subasta.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| BidID | bigint | NO | — | **PK** | Identificador de oferta. |
| AuctionID | bigint | NO | — | **FK→AUCTION(AuctionID)** | Subasta referida. |
| UserID | bigint | NO | — | **FK→USER(UserID)** | Postor. |
| Amount | decimal(38,18) | NO | — | **CK** > 0 | Monto ofertado (ETH). |
| PlacedAt | datetime | NO | SYSUTCDATETIME() |  | Fecha/hora de la oferta (UTC). |

**Reglas de negocio (validación fuera de la tabla):** subasta activa; monto > CurrentPrice (y, si aplica, incremento mínimo %); saldo usable suficiente; desempate por PlacedAt (y BidID).

**Índices:** IX\_Bid\_Auction (AuctionID, Amount DESC, PlacedAt ASC); IX\_Bid\_User (UserID, PlacedAt DESC).

## **Tabla: FUNDS\_RESERVATION**

**Propósito:** Inmovilizar fondos del postor hasta el cierre de la subasta.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| ReservationID | bigint | NO | — | **PK** | Identificador de la reserva. |
| UserID | bigint | NO | — | **FK→USER(UserID)** | Usuario que reserva. |
| AuctionID | bigint | NO | — | **FK→AUCTION(AuctionID)** | Subasta asociada. |
| BidID | bigint | SÍ | — | **FK→BID(BidID)** | Oferta origen (opcional, útil para auditoría). |
| Amount | decimal(38,18) | NO | — | **CK** > 0 | Monto reservado (ETH). |
| State | varchar(10) | NO | 'ACTIVE' | **CK** ∈ {ACTIVE, RELEASED, APPLIED} | Estado de la reserva. |
| CreatedAt | datetime | NO | SYSUTCDATETIME() |  | Creación. |
| UpdatedAt | datetime | NO | SYSUTCDATETIME() |  | Última actualización. |

**Invariantes:** para cada usuario, Reserved == SUM(Amount WHERE State='ACTIVE').

**Índices:** IX\_Reservation\_Auction (AuctionID, State); IX\_Reservation\_User (UserID, State).

## **Tabla: LEDGER**

**Propósito:** Asientos contables de débitos/créditos para auditoría económica.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción / Reglas** |
| --- | --- | --- | --- | --- | --- |
| EntryID | bigint | NO | — | **PK** | Identificador del asiento. |
| UserID | bigint | NO | — | **FK→USER(UserID)** | Usuario afectado. |
| AuctionID | bigint | SÍ | — | **FK→AUCTION(AuctionID)** | Subasta relacionada (si aplica). |
| EntryType | varchar(10) | NO | — | **CK** ∈ {DEBIT, CREDIT} | Naturaleza del asiento. |
| Amount | decimal(38,18) | NO | — | **CK** > 0 | Monto del asiento (ETH). |
| Description | varchar(200) | SÍ | — |  | Descripción breve (Compra NFT, Venta NFT, etc.). |
| CreatedAt | datetime | NO | SYSUTCDATETIME() |  | Fecha/hora del asiento (UTC). |

**Índices:** IX\_Ledger\_UserTime (UserID, CreatedAt DESC).

## **(Opcional) Tabla: SETTINGS**

**Propósito:** Parámetros de negocio editables por administración.

| **Campo** | **Tipo** | **Nulable** | **Default** | **Clave / Restricción** | **Descripción** |
| --- | --- | --- | --- | --- | --- |
| SettingKey | sysname / varchar(128) | NO | — | **PK** | Clave del parámetro (ej., DefaultAuctionHours). |
| SettingValue | nvarchar(200) | NO | — |  | Valor como texto (convertible según uso). |
| UpdatedAt | datetime | NO | SYSUTCDATETIME() |  | Última modificación. |

**Ejemplos de parámetros:** DefaultAuctionHours=72, BasePriceETH=0.1, MinBidIncrementPct=5.

## **Reglas transversales y notas de implementación**

* **UTC** en todos los datetime.
* **Precisión financiera:** usar decimal(38,18) para ETH; evitar float.
* **Checks de negocio** que viven en SPs/Triggers (fuera del diccionario):
  + Oferta válida: subasta ACTIVA, PlacedAt ∈ [StartAt, EndAt), Amount > CurrentPrice (y si aplica, >= CurrentPrice \* (1 + pct/100)), SaldoUsable ≥ Monto.
  + Cierre de subasta: determina ganador (mayor monto; si empate, la más temprana), **aplica** reservas del ganador (**DEBIT**), **abona** al artista (**CREDIT**), **libera** reservas de perdedores, transfiere propiedad del NFT y marca AUCTION.Status='FINALIZED'.
* **Concurrencia:** serializar por AuctionID al aceptar ofertas y al finalizar.
* **Calidad de datos:** HashCode único por NFT; EndAt > StartAt en subastas; saldos no negativos.