

Anexo. Descripción funcional de tablas y esquemas

Descripción de tablas

1. ops.Status

Función. Catálogo maestro de estados parametrizados por dominio (por ejemplo: NFT, AUCTION, USER_EMAIL, FUNDS_RESERVATION, EMAIL_OUTBOX, CURATION_DECISION). Sirve de referencia para las columnas StatusCode mediante una FK compuesta (Domain, Code).

Por qué es importante. Centraliza la semántica de los estados y evita divergencias entre tablas. Facilita validaciones consistentes, control de calidad y elaboración de reportes por estado.

2. ops.Settings

Función. Tabla de parámetros de negocio (por ejemplo: porcentajes mínimos de incremento, duración por defecto de subastas). Guarda clave-valor y fecha de actualización.

Por qué es importante. Externaliza reglas de negocio del código de aplicación; permite reconfigurar comportamientos sin cambios estructurales ni despliegues.

3. core.User

Función. Maestro de usuarios del sistema. Contiene identidad básica y fecha de alta.

Por qué es importante. Actúa como entidad raíz para relaciones con roles, correos, billeteras, pujas, curadurías y asientos contables; es clave para integridad referencial.

4. core.Role

Función. Catálogo de roles (p. ej. ARTIST, CURATOR, COLLECTOR, ADMIN).

Por qué es importante. Establece perfiles funcionales y permite segmentar responsabilidades y consultas por tipo de usuario.

5. core.UserRole

Función. Relación M:N entre usuarios y roles, con marca temporal de asignación.

Por qué es importante. Permite que un usuario acumule varios roles y aporta trazabilidad sobre cuándo se otorgó cada rol.

6. core.UserEmail

Función. Correos electrónicos asociados a un usuario, con indicador IsPrimary y estado (ACTIVE/INACTIVE).

Por qué es importante. Soporta múltiples correos por usuario y garantiza **un** principal (vía índice único filtrado), requisito clave para notificaciones y contacto confiable.

7. core.Wallet

Función. Saldo de la billetera del usuario: disponible y reservado, con marca de última actualización.

Por qué es importante. Respaldar el proceso de subastas (bloqueos y liberaciones de fondos) y se concilia con reservas y libro mayor para consistencia financiera.

8. nft.NFT

Función. Representación de la obra/NFT: metadatos (tipo MIME, tamaño, dimensiones), **hash único** del contenido, autor, propietario actual y estado.

Por qué es importante. Es el activo subastado. El hash único garantiza unicidad técnica; además, enlaza con curaduría y subastas, articulando el flujo completo de publicación y venta.

9. admin.CurationReview

Función. Registro de revisiones curatoriales por NFT: curador, decisión (APPROVE/REJECT), comentarios y tiempos (inicio/fin).

Por qué es importante. Asegura control de calidad previo a la subasta y habilita indicadores (tiempos de revisión, tasas de aprobación por curador/periodo).

10. auction.Auction

Función. Configuración y estado de cada subasta: ventana temporal (inicio/fin), precio inicial, estado, y **cache** de precio/líder actuales.

Por qué es importante. Orquesta el proceso competitivo de venta y garantiza unicidad 1:1 con el NFT listado, evitando duplicidades de publicación.

11. auction.Bid

Función. Historial de ofertas: quién ofertó, cuánto y cuándo, por subasta.

Por qué es importante. Determina la evolución del precio, sustenta la selección del ganador y posibilita análisis de actividad y competitividad.

12. finance.FundsReservation

Función. Reservas de fondos por usuario y subasta (y, si aplica, por puja), con estados (ACTIVE, RELEASED, APPLIED) y trazabilidad temporal.

Por qué es importante. Garantiza que las ofertas estén respaldadas, evita sobregiros y vincula la fase de pujas con el cierre financiero.

13. finance.Ledger

Función. Asientos contables por usuario y subasta (DEBIT/CREDIT), con importe, fecha y descripción.

Por qué es importante. Proporciona una traza financiera auditable, base para balances, ingresos por artista y conciliación con billeteras.

14. audit.EmailOutbox

Función. Cola de correos transaccionales: destinatario (usuario y/o email), estado (PENDING/SENT/FAILED), fechas y clave de correlación.

Por qué es importante. Desacopla el envío respecto de los eventos de negocio, permite reintentos y medición de SLA; preserva el correo efectivo aun si cambia el email principal del usuario.

Descripción de esquemas

Esquema: ops

Función. Proveer servicios transversales del sistema: catálogo maestro de estados por dominio (Status) y parámetros de negocio configurables (Settings).

Por qué es importante. Centraliza reglas y códigos que usan las demás áreas; garantiza consistencia semántica (vía FKs a Status) y flexibilidad (cambiar configuraciones sin tocar el modelo ni el código).

Esquema: core

Función. Gestionar la identidad y el perfil de los usuarios: maestro de usuarios (User), roles y asignaciones (Role, UserRole), correos electrónicos (UserEmail) y billeteras (Wallet).

Por qué es importante. Es la base referencial del sistema: desde aquí se anclan autoría, postores, curadores, destinatarios de notificaciones y saldos. Sin core no hay trazabilidad de “quién” hizo “qué”.

Esquema: nft

Función. Modelar el activo digital a subastar: metadatos técnicos, hash único de contenido, autor y propietario actual (NFT).

Por qué es importante. Es el objeto de negocio central. Conecta la creación/curaduría con el proceso de subasta y la posterior transferencia de propiedad. El hash evita duplicidades y sustenta la unicidad de la obra.

Esquema: admin

Función. Encapsular la curaduría y validación previa de obras (CurationReview): quién revisa, decisión y tiempos de revisión.

Por qué es importante. Asegura control de calidad antes de publicar una obra, habilita indicadores (SLA, tasas de aprobación) y mantiene la trazabilidad de gobernanza del contenido.

Esquema: auction

Función. Orquestar el proceso de subastas: configuración y estado de cada subasta (Auction) y el historial de pujas (Bid).

Por qué es importante. Es el motor competitivo que fija el precio de mercado. Mantiene el estado actual (líder, precio) y registra la dinámica de ofertas para auditoría y reportes de desempeño.

Esquema: finance

Función. Garantizar la integridad financiera: reservas de fondos por subasta/usuario (FundsReservation) y libro mayor de asientos (Ledger) con cargos/abonos.

Por qué es importante. Evita sobregiros, asegura que las pujas estén respaldadas y deja una traza contable auditable de liquidaciones (ganador/autor/plataforma), base para conciliaciones y KPIs económicos.

Esquema: audit

Función. Gestionar notificaciones y trazas transversales; actualmente, la cola de correos transaccionales (EmailOutbox) con su ciclo PENDING/SENT/FAILED.

Por qué es importante. Desacopla los eventos del envío, permite reintentos y medición de SLA, y preserva el correo efectivo utilizado, aun si el email principal del usuario cambia.