Diagramas entidad-relación

Plataforma/Core · Mantenimiento · Calendario · Depósito

Fecha del relevamiento: 20/11/2026

La correspondencia con una base desplegada requiere contraste con su catálogo. Las relaciones entre servicios se distinguen de las claves foráneas locales.

Este documento acompaña el sitio navegable. Incluye las vistas generales, los DER por dominio, las notas de interpretación y el diccionario de datos. Las láminas extensas incorporan ampliaciones con solapamiento para conservar la lectura en papel.

Las páginas de diagramas están preparadas en A3 horizontal; las de texto y diccionario, en A4 horizontal. Para imprimir, seleccionar el tamaño de papel de cada página. La reducción automática a A4 puede disminuir la legibilidad de las láminas.

Dominio	Tablas	FK	Láminas
Plataforma / Core	25	41	4
Mantenimiento	81	215	17
Calendario	17	31	5
Depósito	30	40	6

La Rural · modelo de datos

Índice de contenido

Sección	Página
Guía de lectura	3
Vista general e integraciones	4
Matriz de integraciones	13
Plataforma / Core · DER y lectura técnica	15
Plataforma / Core · diccionario de datos	22
Mantenimiento · DER y lectura técnica	49
Mantenimiento · diccionario de datos	104
Calendario · DER y lectura técnica	201
Calendario · diccionario de datos	212
Depósito · DER y lectura técnica	234
Depósito · diccionario de datos	250

En la versión digital del PDF también se puede navegar mediante los marcadores del documento. Las ampliaciones siguen a la vista completa de cada diagrama.

Alcance y estado de verificación

El modelo abarca 153 tablas lógicas y 327 claves foráneas locales. Las particiones de auditoría y los controles de migraciones no se cuentan como entidades adicionales. Se reconstruyó a partir del código y las migraciones. No acredita por sí solo el catálogo de una base desplegada.

Notación

Círculo = cero; barra = uno; pata de cuervo = muchos. PK identifica la clave primaria; FK, la clave foránea. En los DER detallados una línea continua indica una relación identificadora y una discontinua, una no identificadora. Ambas son FK físicas. Las referencias entre servicios se muestran por separado, sin inventar cardinalidades de base. Gris identifica una tabla desarrollada en otra vista. El signo ? indica que una columna admite NULL.

La nulabilidad y la unicidad total determinan la cardinalidad física. Las restricciones CHECK, los estados y los triggers pueden añadir condiciones. Los índices parciales no convierten toda una relación histórica en 1:1.

Fuentes y límites

Verificado en código y migraciones: estructura declarada. Inferido: interpretación indicada expresamente, sin constraint física acreditada. Base desplegada no contrastada: el catálogo y las integraciones habilitadas requieren contraste con el ambiente. Los identificadores externos, snapshots y destinos polimórficos no se convierten automáticamente en FK.

La Rural · modelo de datos

Vista general e integraciones

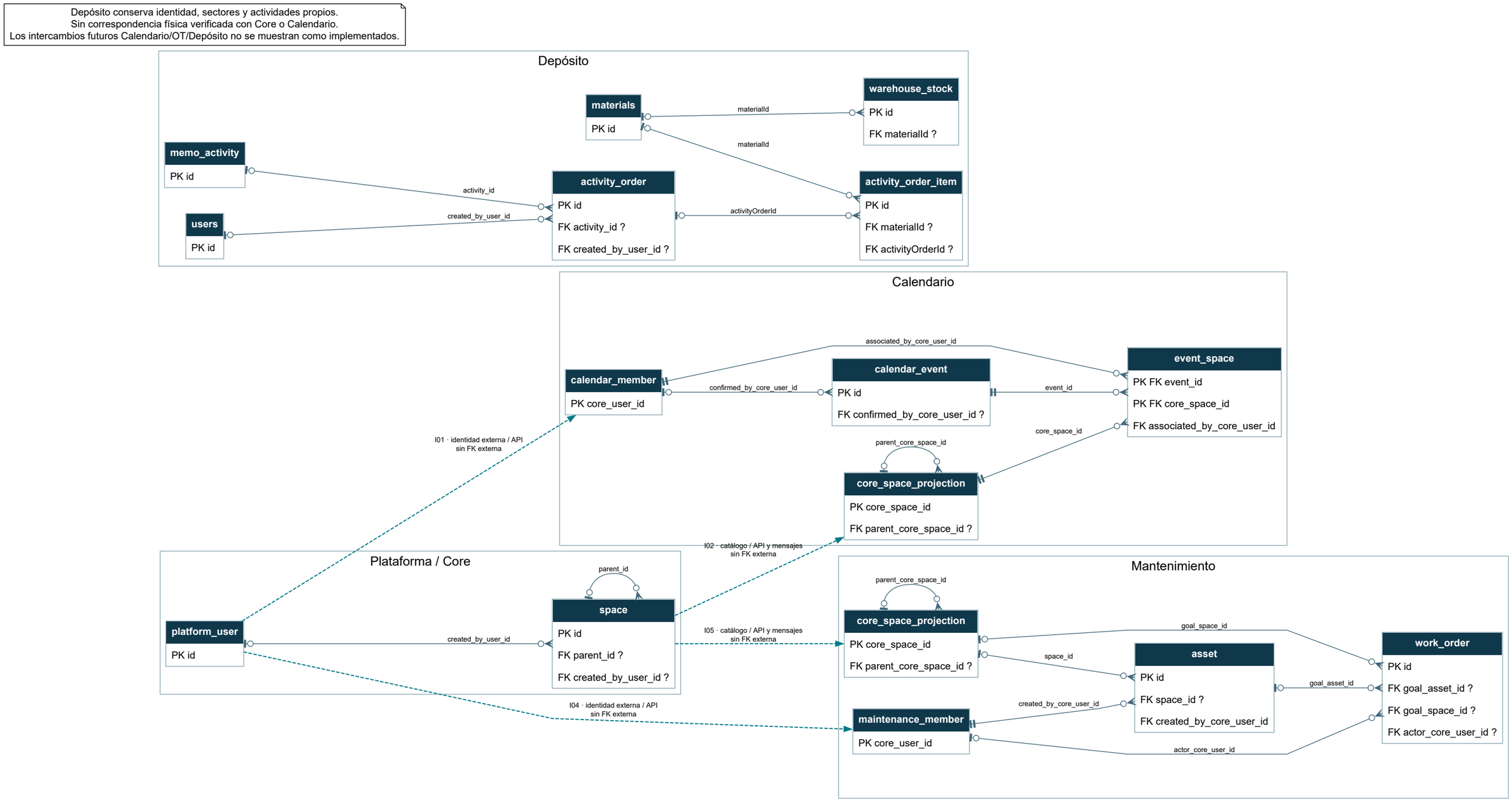
La vista general muestra las entidades principales y FK representativas. Los DER por dominio desarrollan el modelo completo. Las líneas azules entre servicios son referencias lógicas y no prolongan las FK de una proyección local hasta su propietario central.

La Rural · modelo de datos

Vista general del sistema

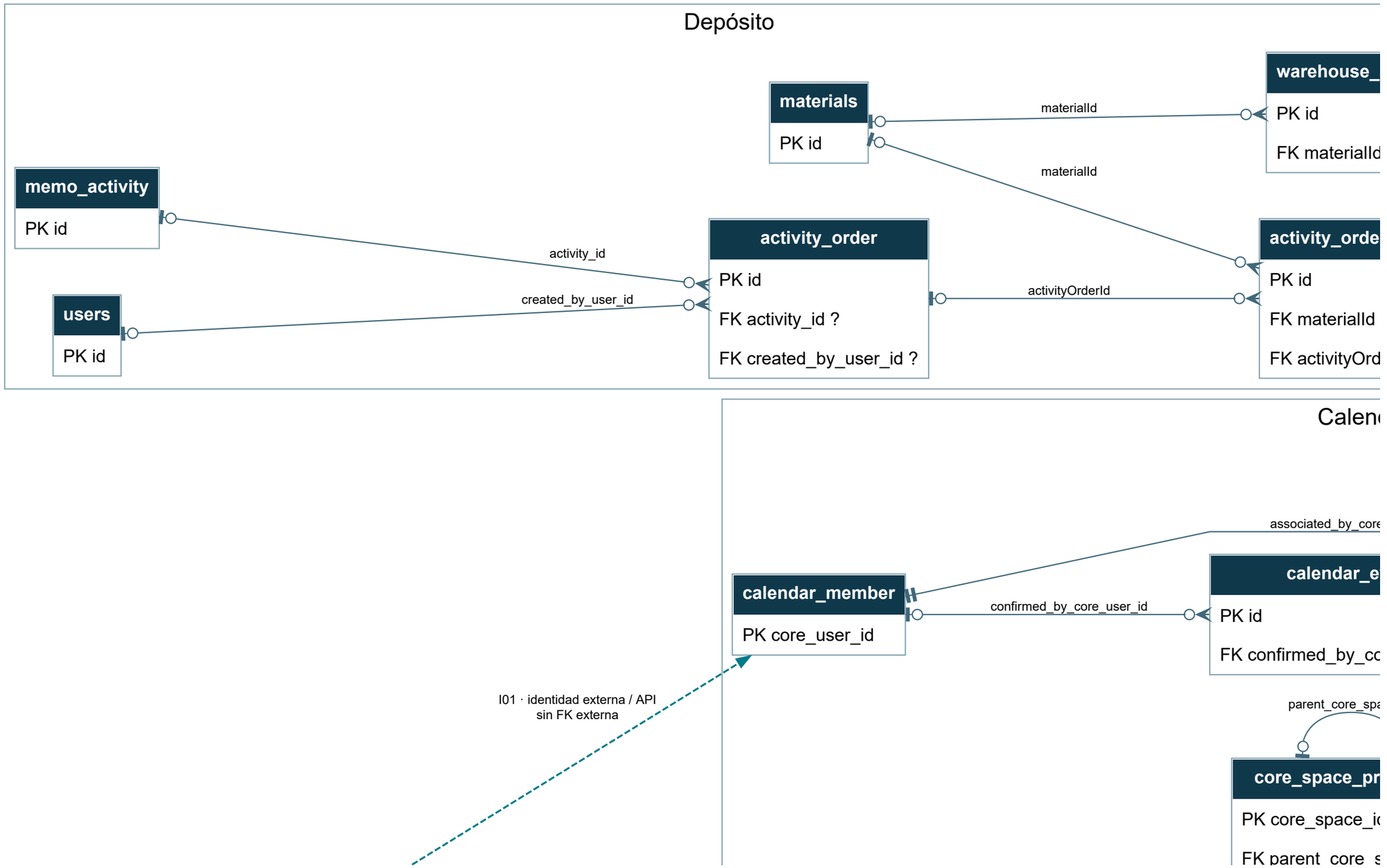
La Rural · vista general de datos · 20/11/2026
FK representativas con pata de cuervo; ver campos y relaciones completas por dominio.
Referencia entre servicios = línea azul sin cardinalidad. Base desplegada pendiente.

Depósito conserva identidad, sectores y actividades propios.
Sin correspondencia física verificada con Core o Calendario.
Los intercambios futuros Calendario/OT/Depósito no se muestran como implementados.

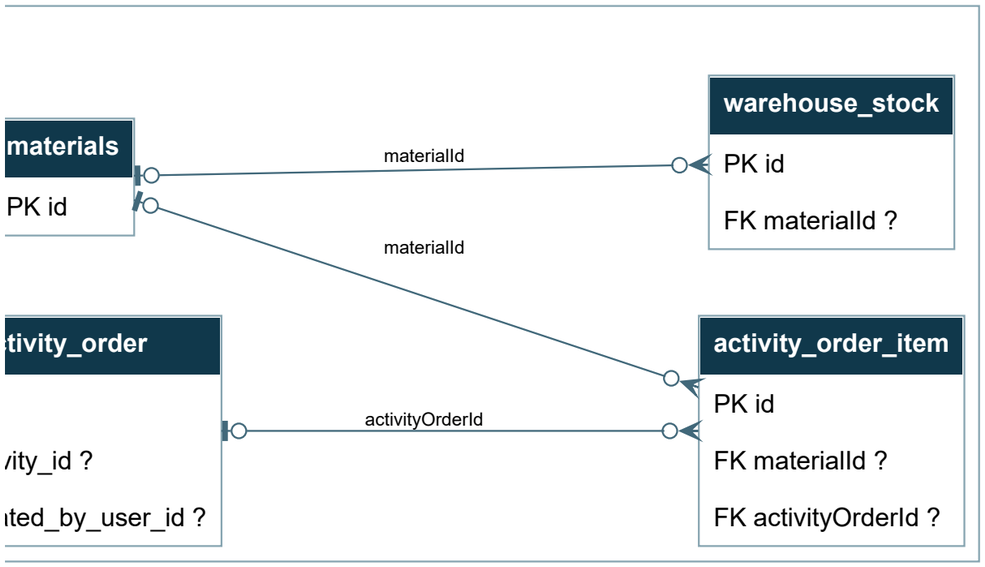


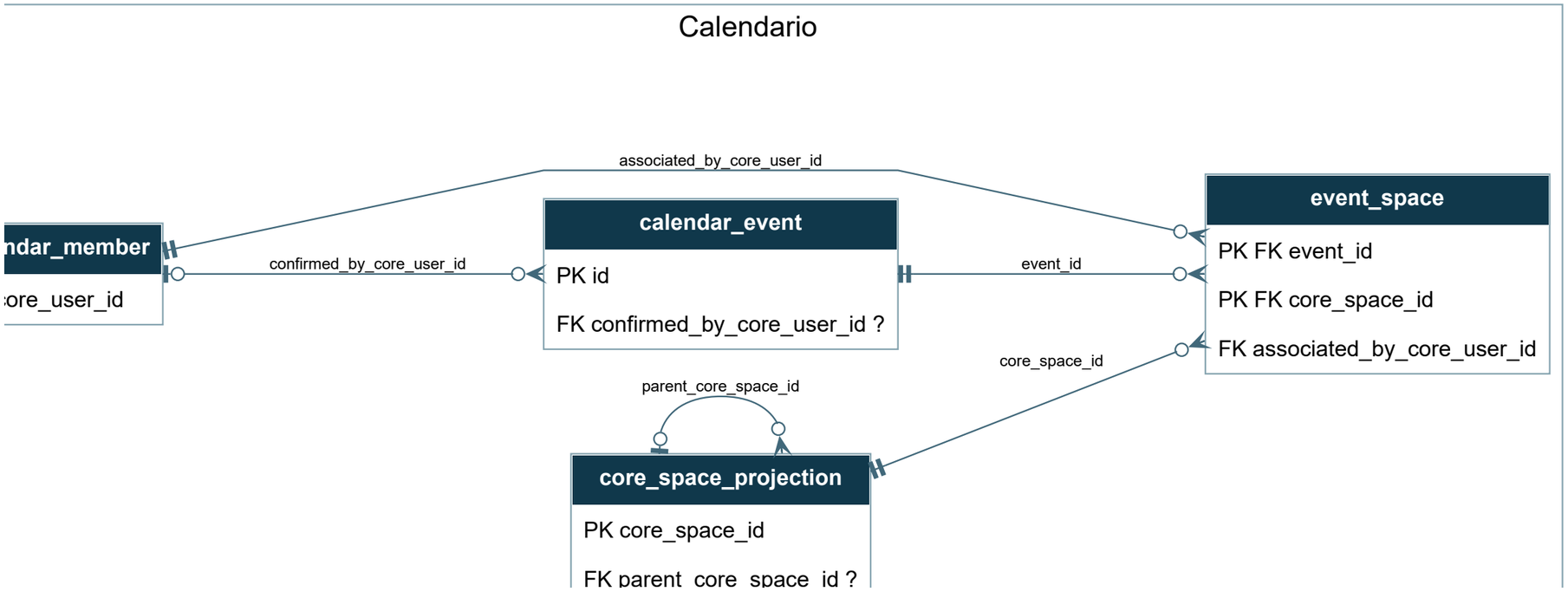
La Rural · vista general de datos · 20/11/2026
FK representativas con pata de cuervo; ver campos y relaciones completa
Referencia entre servicios = línea azul sin cardinalidad. Base desplegad

Depósito conserva identidad, sectores y actividades propios.
Sin correspondencia física verificada con Core o Calendario.
Los intercambios futuros Calendario/OT/Depósito no se muestran como implementados.



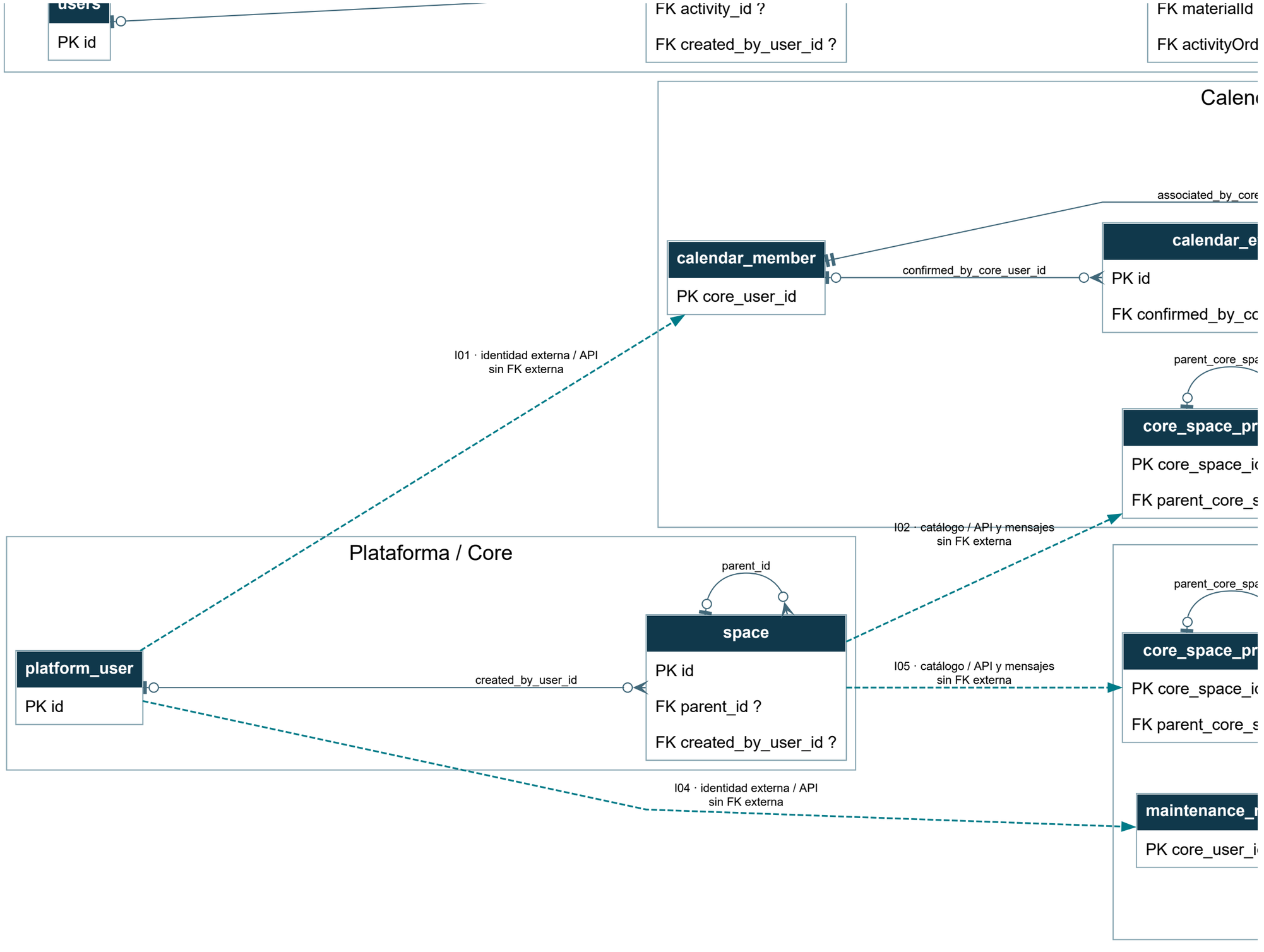
La Rural · vista general de datos · 20/11/2026
tivas con pata de cuervo; ver campos y relaciones completas por dominio.
ntre servicios = línea azul sin cardinalidad. Base desplegada pendiente.

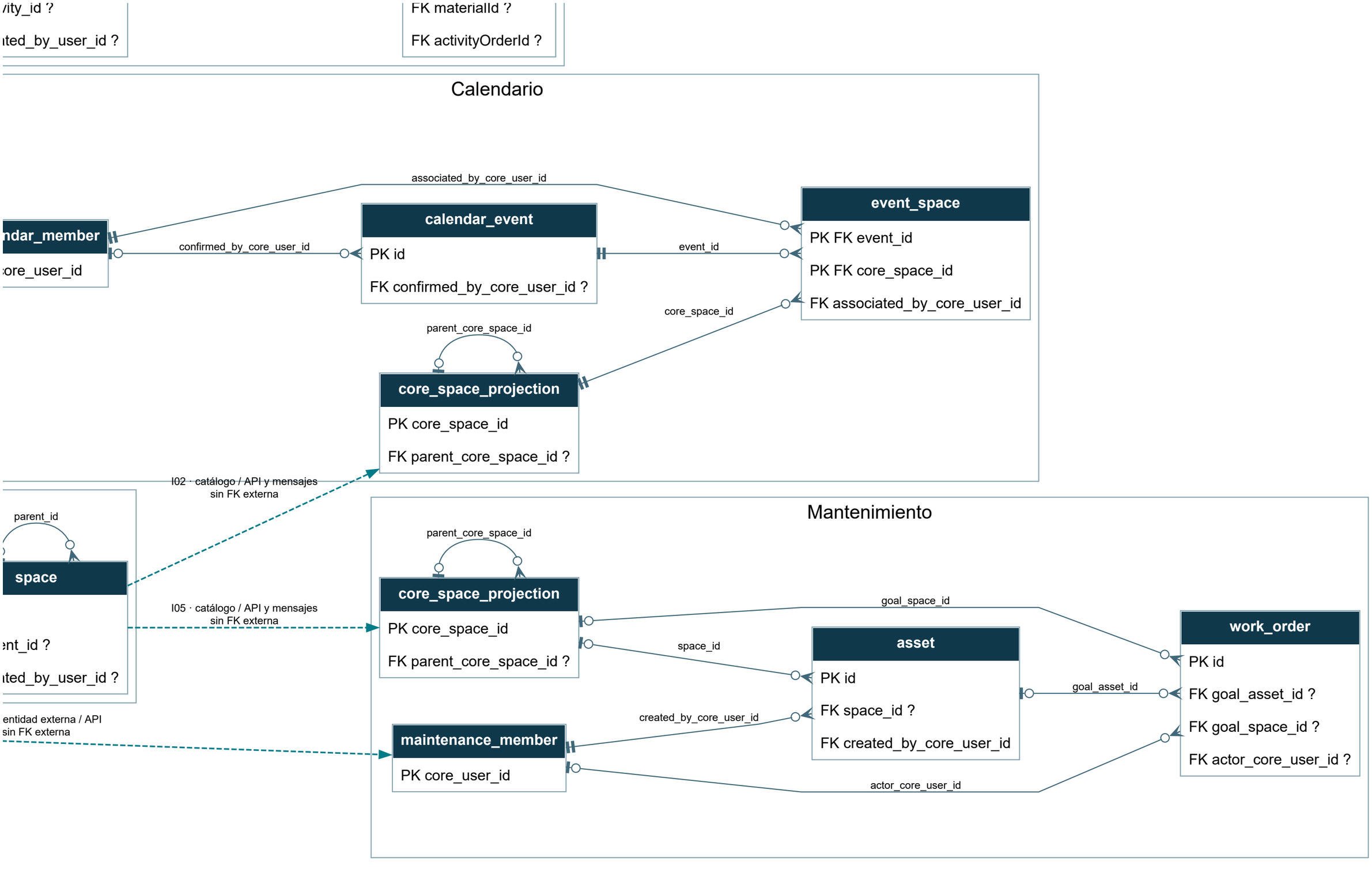




La Rural · modelo de datos

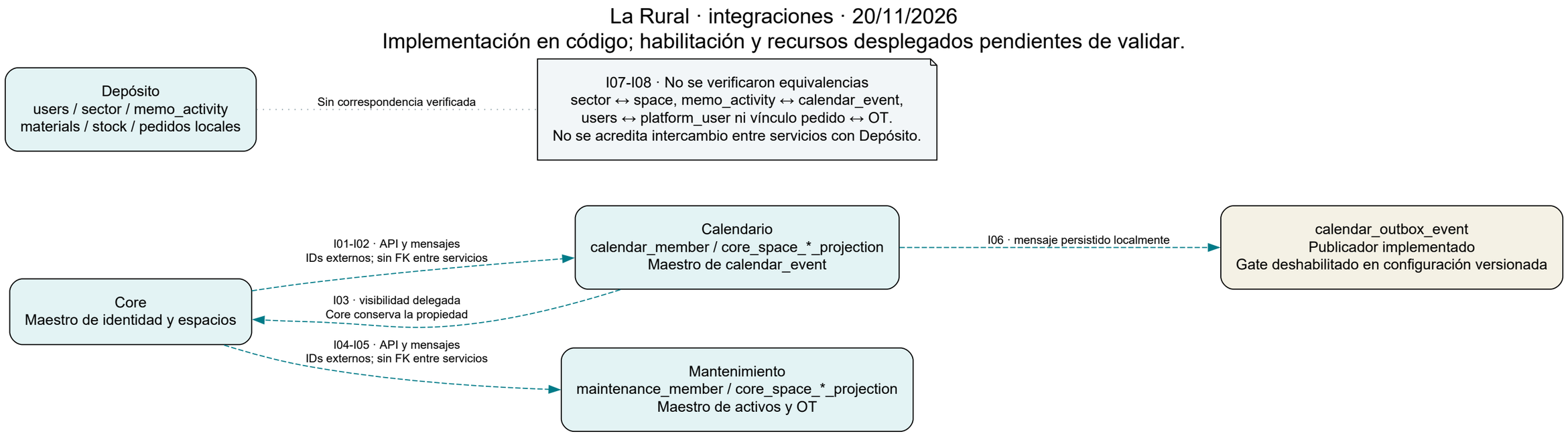
Vista general del sistema · ampliación 2/2, 1/2

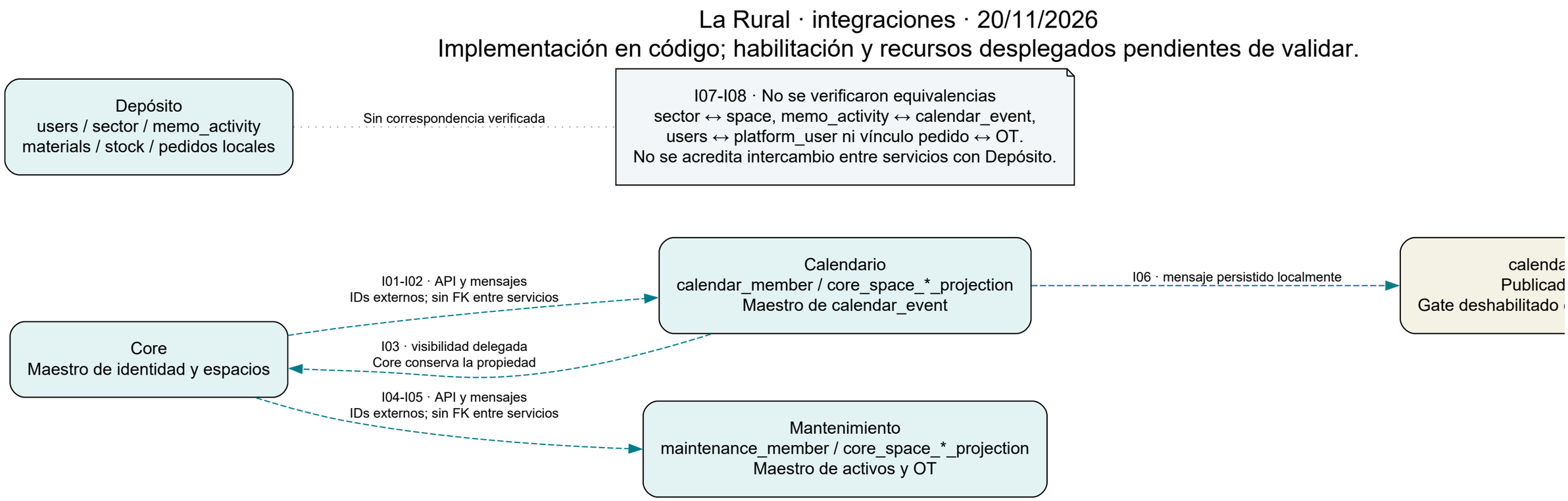


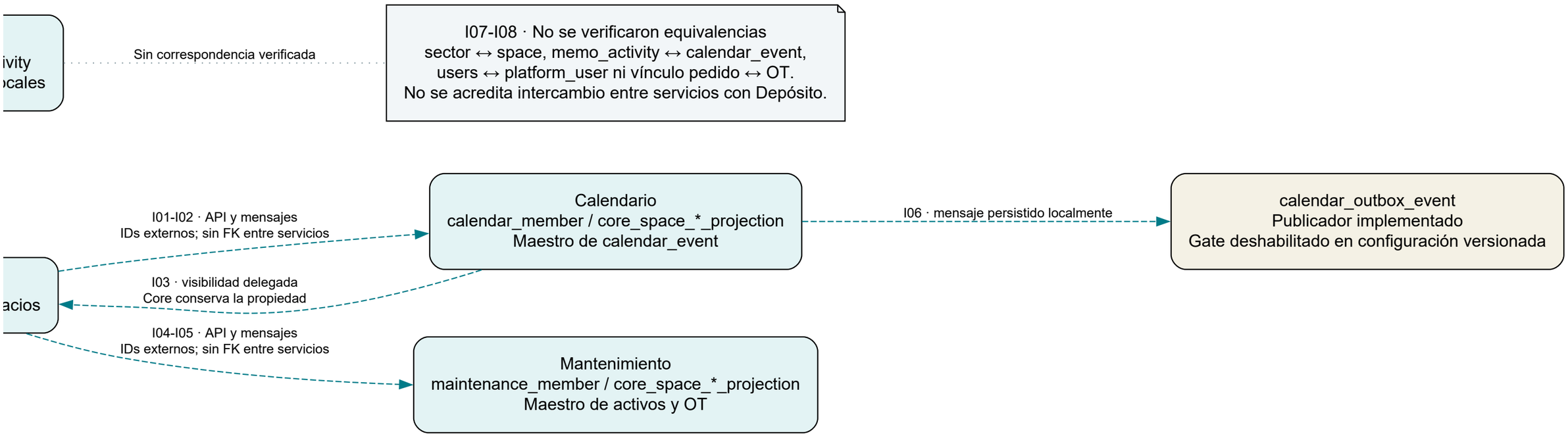

ity_id ?
ted_by_user_id ?
FK materialId ?
FK activityOrderId ?
Calendario
associated_by_core_user_id
ndar_member
core_user_id
confirmed_by_core_user_id
calendar_event
PK id
FK confirmed_by_core_user_id ?
event_id
event_space
PK FK event_id
PK FK core_space_id
FK associated_by_core_user_id
core_space_id
parent_core_space_id
core_space_projection
PK core_space_id
FK parent_core_space_id ?
I02 · catálogo / API y mensajes sin FK externa
parent_id
space
nt_id ?
ted_by_user_id ?
I05 · catálogo / API y mensajes sin FK externa
entidad externa / API sin FK externa
Mantenimiento
parent_core_space_id
core_space_projection
PK core_space_id
FK parent_core_space_id ?
goal_space_id
space_id
asset
PK id
FK space_id ?
FK created_by_core_user_id
goal_asset_id
work_order
PK id
FK goal_asset_id ?
FK goal_space_id ?
FK actor_core_user_id ?
created_by_core_user_id
maintenance_member
PK core_user_id
actor_core_user_id

La Rural · modelo de datos

Integraciones entre módulos




La Rural · integraciones · 20/11/2026
Implementación en código; habilitación y recursos desplegados pendientes de validar.
Depósito
users / sector / memo_activity
materials / stock / pedidos locales
Sin correspondencia verificada
I07-I08 · No se verificaron equivalencias
sector ↔ space, memo_activity ↔ calendar_event,
users ↔ platform_user ni vínculo pedido ↔ OT.
No se acredita intercambio entre servicios con Depósito.
Calendario
calendar_member / core_space_*_projection
Maestro de calendar_event
I06 · mensaje persistido localmente
calenda
Publicad
Gate deshabilitado
I01-I02 · API y mensajes
IDs externos; sin FK entre servicios
Core
Maestro de identidad y espacios
I03 · visibilidad delegada
Core conserva la propiedad
I04-I05 · API y mensajes
IDs externos; sin FK entre servicios
Mantenimiento
maintenance_member / core_space_*_projection
Maestro de activos y OT


La Rural · integraciones · 20/11/2026
Implementación en código; habilitación y recursos desplegados pendientes de validar.
ivity
ocales
Sin correspondencia verificada
I07-I08 · No se verificaron equivalencias
sector ↔ space, memo_activity ↔ calendar_event,
users ↔ platform_user ni vínculo pedido ↔ OT.
No se acredita intercambio entre servicios con Depósito.
I01-I02 · API y mensajes
IDs externos; sin FK entre servicios
Calendario
calendar_member / core_space_*_projection
Maestro de calendar_event
I06 · mensaje persistido localmente
calendar_outbox_event
Publicador implementado
Gate deshabilitado en configuración versionada
acios
I03 · visibilidad delegada
Core conserva la propiedad
I04-I05 · API y mensajes
IDs externos; sin FK entre servicios
Mantenimiento
maintenance_member / core_space_*_projection
Maestro de activos y OT

I01 · Core / Calendario

Dato: platform_user y user_module_access. Representación: calendar_member.core_user_id.

Vínculo: API privada de miembros; JWT de módulo y reconciliación. Estado técnico: Integración verificada en código; referencia UUID lógica, sin FK externa. La habilitación en un ambiente no queda acreditada por el modelo.

I02 · Core / Calendario

Dato: space, jerarquía, linaje y visibilidad por módulo. Representación: core_space_projection, core_space_relation_projection, core_space_lineage_projection.

Vínculo: API snapshot/cambios y mensajes Core → inbox; checkpoints. Estado técnico: Integración verificada en código. Las FK de event_space apuntan a la proyección local, no a Core.

I03 · Core / Calendario

Dato: space_module_setting. Representación: Comando de visibilidad delegado; luego proyección.

Vínculo: PUT interno con actor delegado, control de versión e idempotencia. Estado técnico: Integración verificada en código; la propiedad del dato sigue en Core. Sin FK entre servicios.

I04 · Core / Mantenimiento

Dato: platform_user y accesos al módulo. Representación: maintenance_member.core_user_id.

Vínculo: API privada, reconciliación e identidad firmada. Estado técnico: Integración verificada en código; UUID externo y FK locales hacia maintenance_member.

I05 · Core / Mantenimiento

Dato: space, jerarquía, linaje y visibilidad. Representación: core_space_*_projection, core_inbox_event, core_sync_checkpoint.

Vínculo: API privada y mensajes con copia local; visibilidad delegada. Estado técnico: Integración verificada en código; no FK al esquema central.

I06 · Calendario / Consumidor futuro

Dato: calendar_event y calendar_outbox_event. Representación: Mensaje con aggregate_id/version/payload.

Vínculo: Publicador Pub/Sub implementado; gate de configuración. Estado técnico: Código verificado. Default y configuración versionada lo deshabilitan; consumidor Mantenimiento/Depósito no verificado. No se acredita entrega efectiva a esos módulos.

I07 · Depósito / Propio módulo

Dato: users, sector, memo_activity. Representación: IDs enteros locales; sin correspondencia Core/Calendario declarada.

Vínculo: Identidad/autorización y persistencia propias. Estado técnico: Modelo local verificado. Sin equivalencias verificadas users↔platform_user, sector↔space o memo_activity↔calendar_event.

I08 · Depósito / Mantenimiento / OT (posible evolución)

Dato: materials, warehouse_stock, activity_order y counter_order. Representación: No se verificó campo/FK/consumidor que materialice el vínculo.

Vínculo: Sin implementación verificada en el código relevado. Estado técnico: Sin vínculo de integración verificado. Los pedidos locales no son OT. Sin cardinalidad entre servicios.

I09 · Cada módulo / Propio módulo

Dato: Auditoría, snapshots y actores. Representación: IDs locales, UUID externos, etiquetas y JSON históricos.

Vínculo: FK donde existe constraint; referencias lógicas en los demás casos. Estado técnico: Verificado por tabla. Un identificador histórico no acredita una FK.

25 tablas lógicas · 41 FK locales · 4 láminas de dominio.

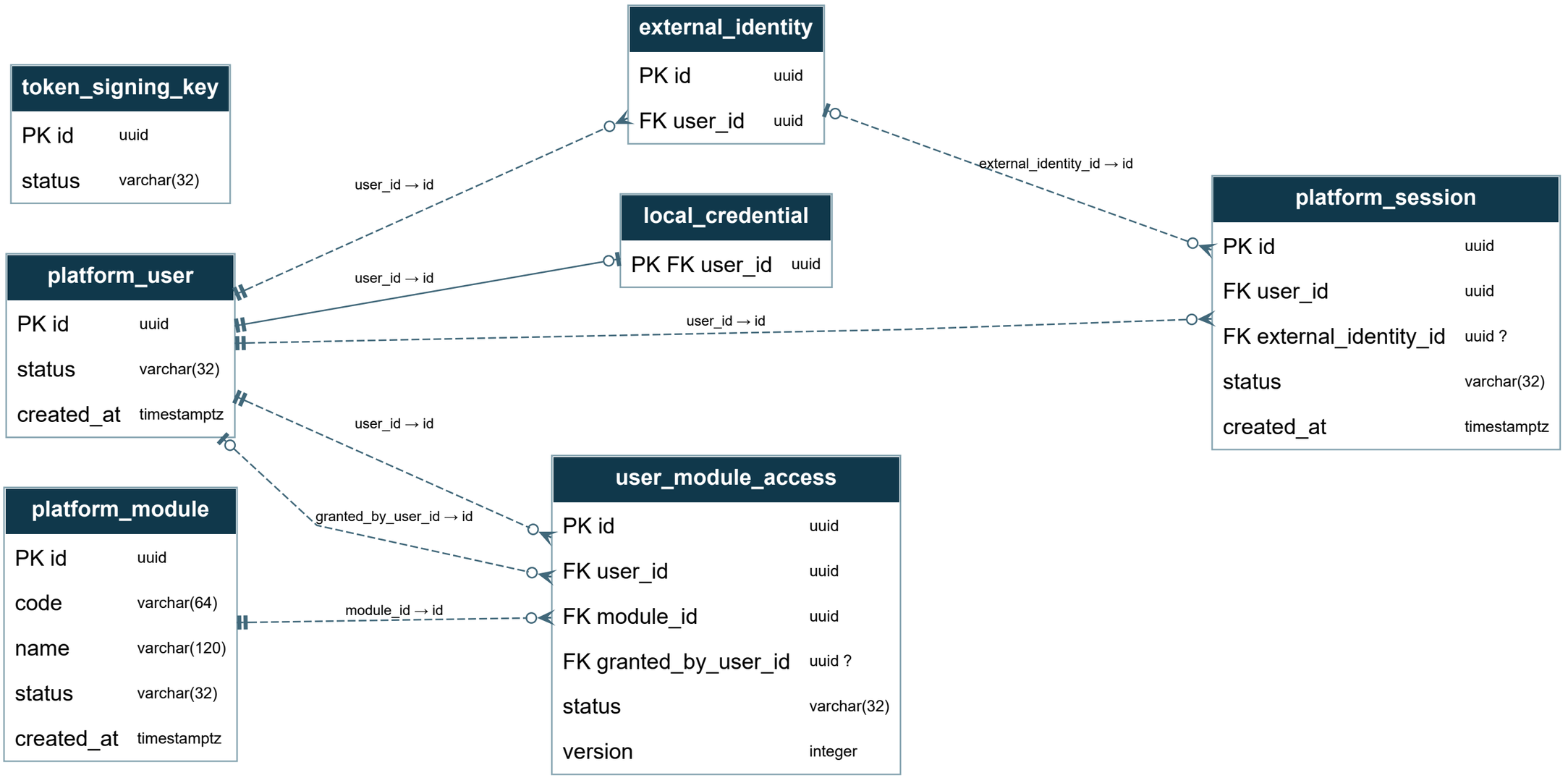
Propiedad central de los datos

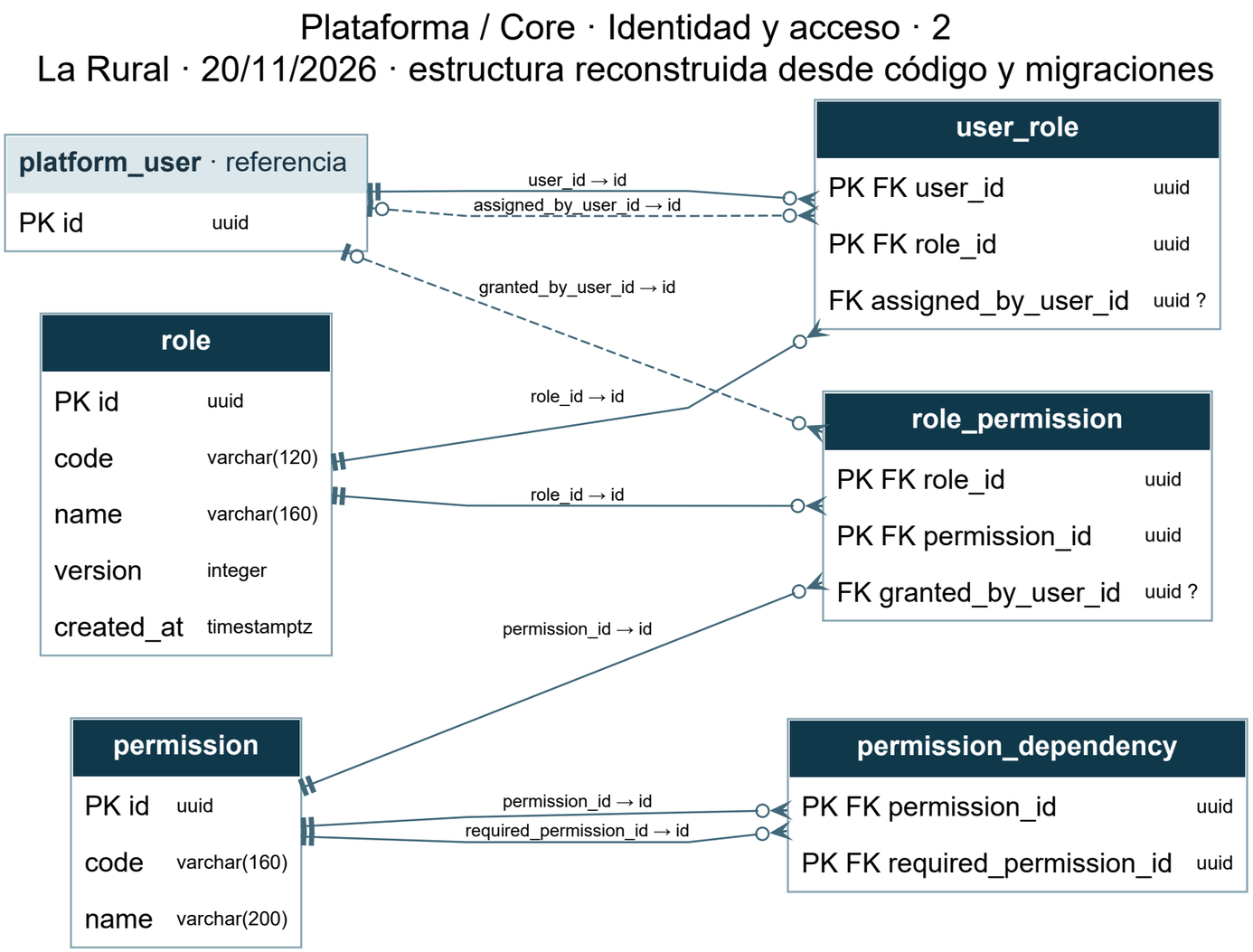
Plataforma y Core forman un único dominio central. La identidad, los accesos a módulos y el catálogo de espacios pertenecen a este dominio. Los módulos consumidores pueden conservar proyecciones locales; esto no crea una FK entre servicios.

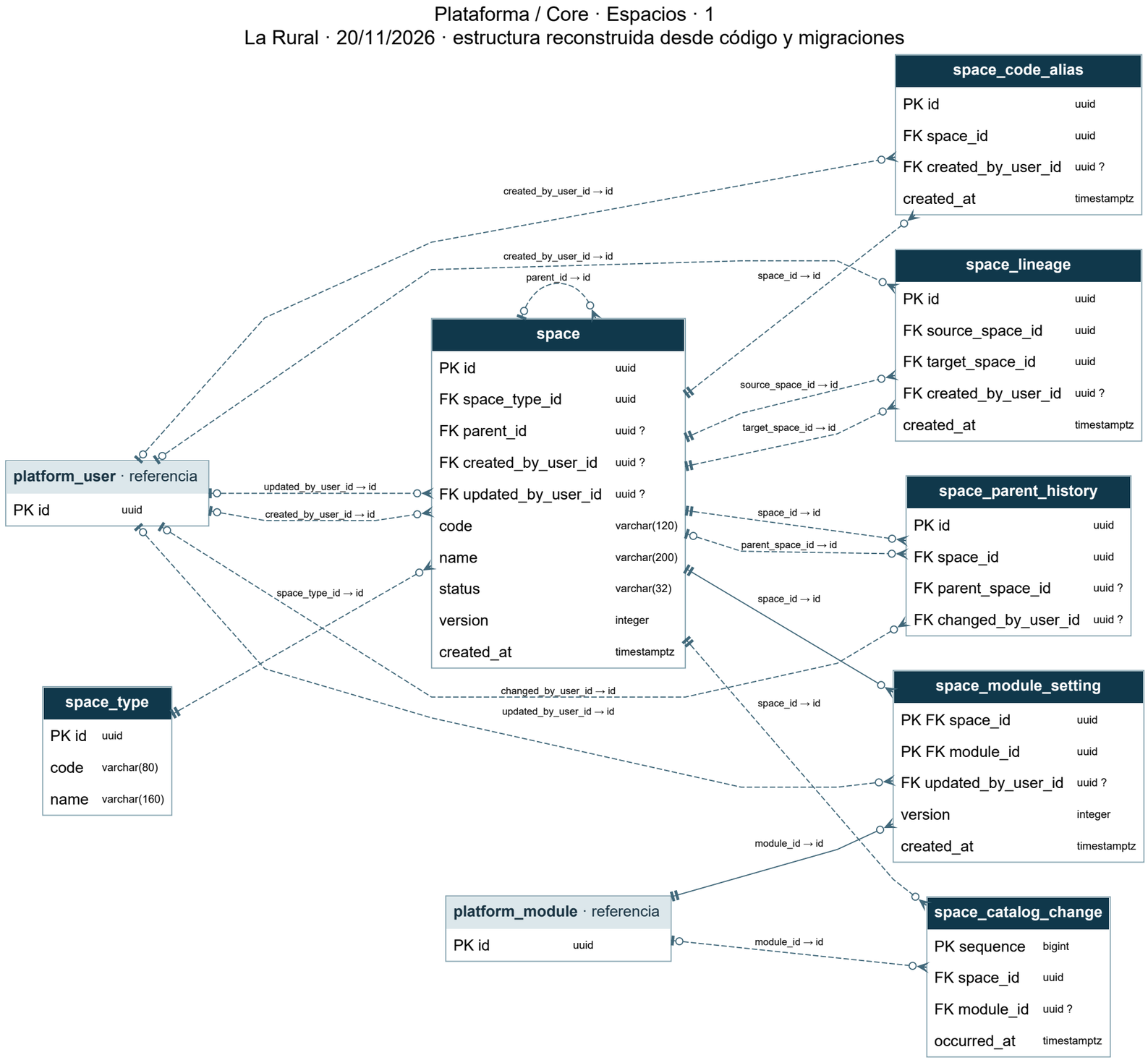
Historia y auditoría

Las tablas de linaje, parentesco y alias conservan información de la evolución espacial. audit_log es una cabecera particionada; las particiones no constituyen entidades de negocio adicionales. Los identificadores y datos históricos sin constraint explícita se interpretan como referencias lógicas.

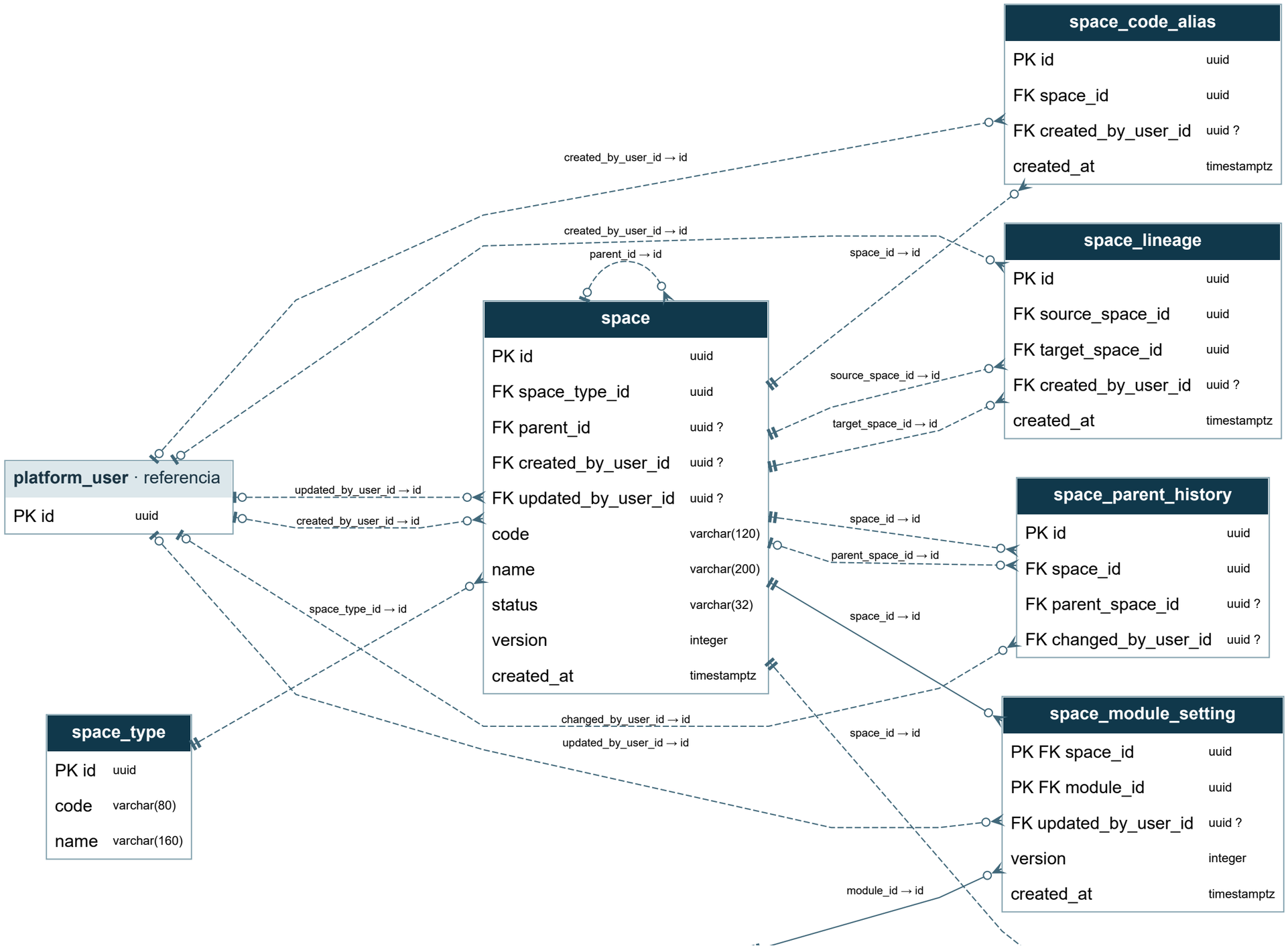
Plataforma / Core · Identidad y acceso · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

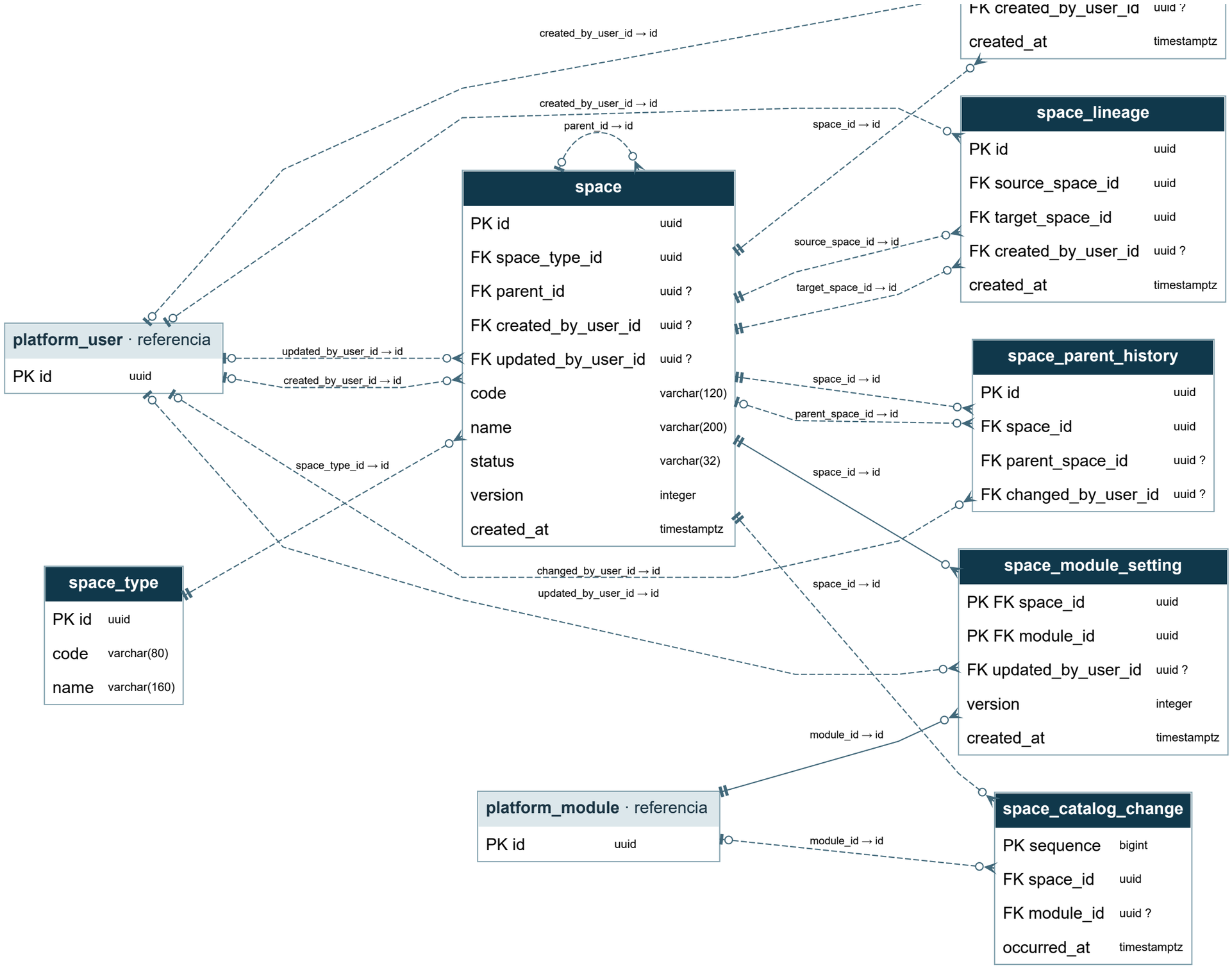



Plataforma / Core · Identidad y acceso · 2
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
platform_user · referencia
PK id uuid
user_role
PK FK user_id uuid
PK FK role_id uuid
FK assigned_by_user_id uuid ?
role
PK id uuid
code varchar(120)
name varchar(160)
version integer
created_at timestamptz
role_permission
PK FK role_id uuid
PK FK permission_id uuid
FK granted_by_user_id uuid ?
permission
PK id uuid
code varchar(160)
name varchar(200)
permission_dependency
PK FK permission_id uuid
PK FK required_permission_id uuid
user_id → id
assigned_by_user_id → id
granted_by_user_id → id
role_id → id
role_id → id
permission_id → id
permission_id → id
required_permission_id → id

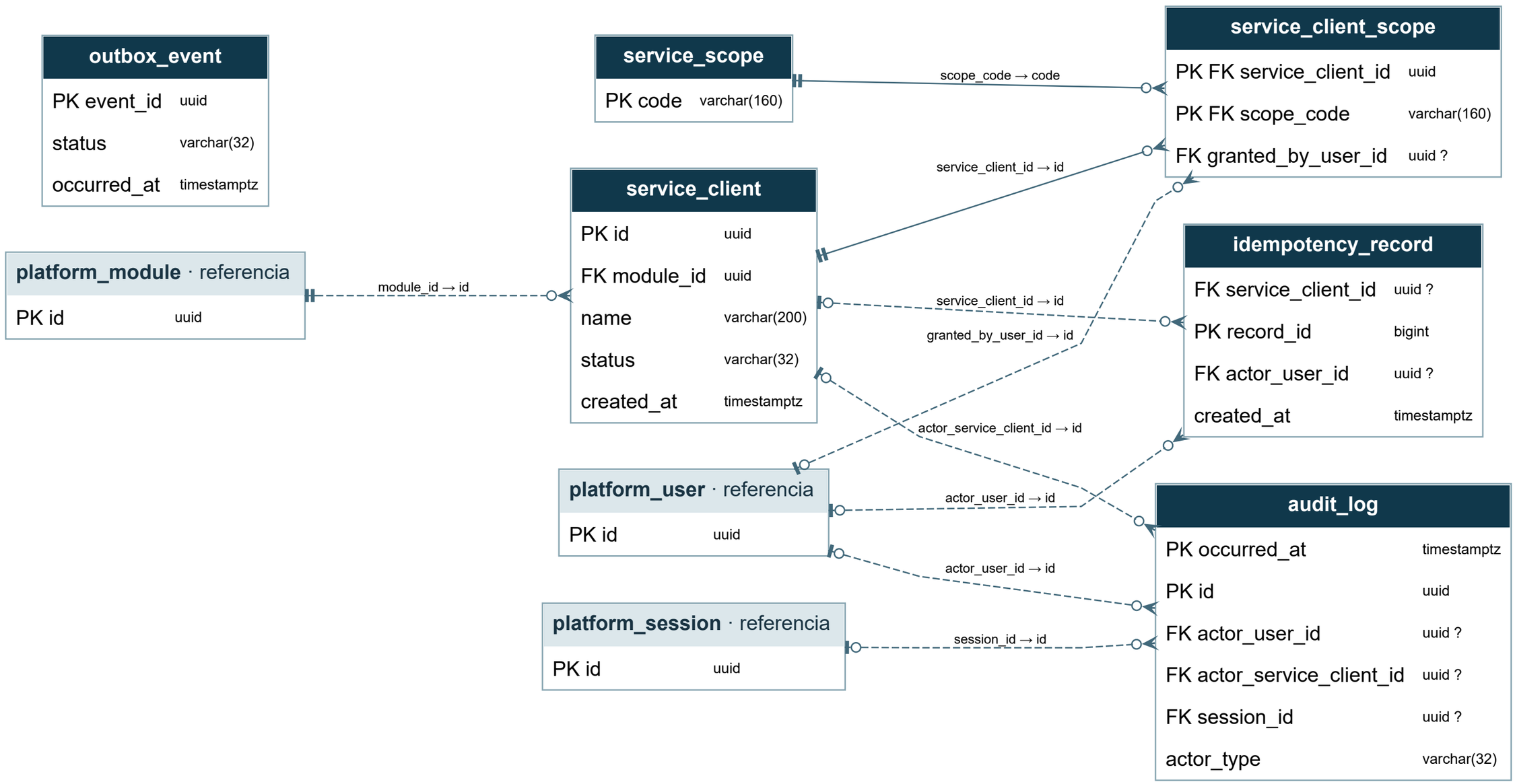

Plataforma / Core · Espacios · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
space_code_alias
PK id uuid
FK space_id uuid
FK created_by_user_id uuid ?
created_at timestamptz
created_by_user_id → id
created_by_user_id → id
parent_id → id
space_id → id
space
PK id uuid
FK space_type_id uuid
FK parent_id uuid ?
FK created_by_user_id uuid ?
FK updated_by_user_id uuid ?
code varchar(120)
name varchar(200)
status varchar(32)
version integer
created_at timestamptz
space_lineage
PK id uuid
FK source_space_id uuid
FK target_space_id uuid
FK created_by_user_id uuid ?
created_at timestamptz
source_space_id → id
target_space_id → id
platform_user · referencia
PK id uuid
updated_by_user_id → id
created_by_user_id → id
space_id → id
parent_space_id → id
space_parent_history
PK id uuid
FK space_id uuid
FK parent_space_id uuid ?
FK changed_by_user_id uuid ?
space_type_id → id
space_id → id
space_type
PK id uuid
code varchar(80)
name varchar(160)
changed_by_user_id → id
updated_by_user_id → id
space_id → id
space_module_setting
PK FK space_id uuid
PK FK module_id uuid
FK updated_by_user_id uuid ?
version integer
created_at timestamptz
module_id → id
platform_module · referencia
PK id uuid
module_id → id
space_catalog_change
PK sequence bigint
FK space_id uuid
FK module_id uuid ?
occurred_at timestamptz

Plataforma / Core · Espacios · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones




FK created_by_user_id uuid ?
created_at timestamptz
created_by_user_id → id
created_by_user_id → id
parent_id → id
space_id → id
space
PK id uuid
FK space_type_id uuid
FK parent_id uuid ?
FK created_by_user_id uuid ?
FK updated_by_user_id uuid ?
code varchar(120)
name varchar(200)
status varchar(32)
version integer
created_at timestamptz
space_lineage
PK id uuid
FK source_space_id uuid
FK target_space_id uuid
FK created_by_user_id uuid ?
created_at timestamptz
source_space_id → id
target_space_id → id
platform_user · referencia
PK id uuid
updated_by_user_id → id
created_by_user_id → id
space_id → id
parent_space_id → id
space_parent_history
PK id uuid
FK space_id uuid
FK parent_space_id uuid ?
FK changed_by_user_id uuid ?
space_type_id → id
space_id → id
space_type
PK id uuid
code varchar(80)
name varchar(160)
changed_by_user_id → id
updated_by_user_id → id
space_id → id
space_module_setting
PK FK space_id uuid
PK FK module_id uuid
FK updated_by_user_id uuid ?
version integer
created_at timestamptz
module_id → id
platform_module · referencia
PK id uuid
module_id → id
space_catalog_change
PK sequence bigint
FK space_id uuid
FK module_id uuid ?
occurred_at timestamptz

Plataforma / Core · Integración y auditoría · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



platform_user

Identidad central de una persona usuaria.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
email	varchar(320)	NO		
display_name	varchar(200)	NO		
status	varchar(32)	NO		
session_version	integer	NO		
last_login_at	timestamptz	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
profile_version	integer	NO		

CHECK: CONSTRAINT ck_platform_user_email_normalized CHECK (email = lower(btrim(email)) AND email <> '') · Migración 1786406400000 · línea 48

CHECK: CONSTRAINT ck_platform_user_display_name CHECK (btrim(display_name) <> '') · Migración 1786406400000 · línea 48

CHECK: CONSTRAINT ck_platform_user_status CHECK (status IN ('pending_access', 'active', 'suspended', 'blocked')) · Migración 1786406400000 · línea 48

CHECK: CONSTRAINT ck_platform_user_session_version CHECK (session_version >= 1) · Migración 1786406400000 · línea 48

CHECK: CONSTRAINT ck_platform_user_timestamps CHECK (updated_at >= created_at) · Migración 1786406400000 · línea 48

CHECK: CONSTRAINT ck_platform_user_profile_version CHECK (profile_version >= 1) · Migración 1789516800000 · línea 47

Índice: CREATE UNIQUE INDEX IF NOT EXISTS uq_platform_user_email_normalized ON esquema.platform_user (lower(email)) · Migración 1786406400000 · línea 70

Índice: CREATE INDEX IF NOT EXISTS ix_platform_user_status ON esquema.platform_user (status) · Migración 1786406400000 · línea 74

Fuentes técnicas: Migración 1786406400000 · línea 48; Migración 1789516800000 · línea 31.

external_identity

Vínculo de identidad con proveedor externo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (provider, tenant_id, subject_id); (user_id, provider).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
user_id	uuid	NO	FK	platform_user(id)
provider	varchar(32)	NO		
tenant_id	varchar(128)	NO		
subject_id	varchar(128)	NO		
email_snapshot	varchar(320)	NO		
linked_at	timestamptz	NO		
last_login_at	timestamptz	SÍ		

FK (user_id) → platform_user(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 79.

CHECK: CONSTRAINT ck_external_identity_provider CHECK (provider IN ('entra')) · Migración 1786406400000 · línea 79

CHECK: CONSTRAINT ck_external_identity_values CHECK (btrim(tenant_id) <> '' AND btrim(subject_id) <> '' AND email_snapshot = lower(btrim(email_snapshot)) AND email_snapshot <> '') · Migración 1786406400000 · línea 79

Índice: CREATE INDEX IF NOT EXISTS ix_external_identity_user ON esquema.external_identity (user_id) · Migración 1786406400000 · línea 101

Índice: CREATE UNIQUE INDEX IF NOT EXISTS uq_external_identity_user_provider ON esquema.external_identity (user_id, provider) · Migración 1786492800000 · línea 64

Fuentes técnicas: Migración 1786406400000 · línea 79.

local_credential

Credencial local y controles de contraseña.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: user_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
user_id	uuid	NO	PK, FK	platform_user(id)
password_hash	text	NO		
hash_algorithm	varchar(32)	NO		
failed_attempt_count	integer	NO		
locked_until	timestamptz	SÍ		
password_changed_at	timestamptz	NO		

FK (user_id) → platform_user(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1786406400000 · línea 106.

CHECK: CONSTRAINT ck_local_credential_failed_attempts CHECK (failed_attempt_count >= 0) · Migración 1786406400000 · línea 106

CHECK: CONSTRAINT ck_local_credential_hash CHECK (password_hash <> '' AND hash_algorithm = 'argon2id') · Migración 1786492800000 · línea 58

Índice: CREATE INDEX IF NOT EXISTS ix_local_credential_locked_until ON esquema.local_credential (locked_until) WHERE locked_until IS NOT NULL · Migración 1786406400000 · línea 125

Fuentes técnicas: Migración 1786406400000 · línea 106.

platform_session

Sesión central de acceso.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (token_hash).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
user_id	uuid	NO	FK	platform_user(id)
external_identity_id	uuid	SÍ	FK	external_identity(id)
auth_method	varchar(32)	NO		
status	varchar(32)	NO		
created_ip	inet	SÍ		
created_at	timestamptz	NO		
last_seen_at	timestamptz	NO		
expires_at	timestamptz	NO		
revoked_at	timestamptz	SÍ		
token_hash	text	NO		

FK (user_id) → platform_user(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 155.

FK (external_identity_id) → external_identity(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 155.

CHECK: CONSTRAINT ck_platform_session_auth_method CHECK (auth_method IN ('entra', 'local')) · Migración 1786406400000 · línea 155

CHECK: CONSTRAINT ck_platform_session_status CHECK (status IN ('active', 'revoked', 'expired')) · Migración 1786406400000 · línea 155

CHECK: CONSTRAINT ck_platform_session_identity CHECK ((auth_method = 'entra' AND external_identity_id IS NOT NULL) OR (auth_method = 'local' AND external_identity_id IS NULL)) · Migración 1786406400000 · línea 155

CHECK: CONSTRAINT ck_platform_session_timestamps CHECK (last_seen_at >= created_at AND expires_at > created_at AND (revoked_at IS NULL OR revoked_at >= created_at)) · Migración 1786406400000 · línea 155

Índice: CREATE INDEX IF NOT EXISTS ix_platform_session_user_status_expires ON esquema.platform_session (user_id, status, expires_at) · Migración 1786406400000 · línea 187

Índice: CREATE INDEX IF NOT EXISTS ix_platform_session_status_expires ON esquema.platform_session (status, expires_at) · Migración 1786406400000 · línea 191

Índice: CREATE UNIQUE INDEX IF NOT EXISTS uq_platform_session_token_hash ON esquema.platform_session (token_hash) · Migración 1786492800000 · línea 98

Fuentes técnicas: Migración 1786406400000 · línea 155; Migración 1786492800000 · línea 90.

token_signing_key

Ciclo de vida de claves de firma.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (kid).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
kid	varchar(128)	NO		
algorithm	varchar(64)	NO		
public_jwk	jsonb	NO		
private_key_secret_ref	varchar(512)	NO		
status	varchar(32)	NO		
not_before	timestamptz	NO		
expires_at	timestamptz	NO		
retired_at	timestamptz	SÍ		

CHECK: CONSTRAINT ck_token_signing_key_values CHECK (btrim(kid) <> '' AND btrim(algorithm) <> '' AND jsonb_typeof(public_jwk) = 'object' AND btrim(private_key_secret_ref) <> '') · Migración 1786406400000 · línea 196

CHECK: CONSTRAINT ck_token_signing_key_status CHECK (status IN ('pending', 'active', 'retiring', 'retired')) · Migración 1786406400000 · línea 196

CHECK: CONSTRAINT ck_token_signing_key_lifetime CHECK (expires_at > not_before AND (retired_at IS NULL OR retired_at >= not_before)) · Migración 1786406400000 · línea 196

Índice: CREATE INDEX IF NOT EXISTS ix_token_signing_key_status_validity ON esquema.token_signing_key (status, not_before, expires_at) · Migración 1786406400000 · línea 221

Fuentes técnicas: Migración 1786406400000 · línea 196.

platform_module

Módulo habilitable de la plataforma.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code); (base_path); (api_audience); (api_cookie_path).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(64)	NO		
name	varchar(120)	NO		
base_path	varchar(160)	NO		
api_audience	varchar(120)	NO		
api_cookie_path	varchar(160)	NO		
status	varchar(32)	NO		
sort_order	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

CHECK: CONSTRAINT ck_platform_module_code CHECK (code = lower(code) AND code ~ '^[a-z][a-z0-9_]*$') · Migración 1786406400000 · línea 226

CHECK: CONSTRAINT ck_platform_module_paths CHECK (base_path ~ '^/[a-z][a-z0-9_-]*$' AND api_cookie_path ~ '^/api/[a-z][a-z0-9_-]*$') · Migración 1786406400000 · línea 226

CHECK: CONSTRAINT ck_platform_module_status CHECK (status IN ('active', 'inactive')) · Migración 1786406400000 · línea 226

CHECK: CONSTRAINT ck_platform_module_timestamps CHECK (updated_at >= created_at) · Migración 1786406400000 · línea 226

Índice: CREATE INDEX IF NOT EXISTS ix_platform_module_status_sort ON esquema.platform_module (status, sort_order) · Migración 1786406400000 · línea 253

Fuentes técnicas: Migración 1786406400000 · línea 226.

user_module_access

Acceso de una persona a un módulo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (user_id, module_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
user_id	uuid	NO	FK	platform_user(id)
module_id	uuid	NO	FK	platform_module(id)
status	varchar(32)	NO		
reason	text	SÍ		
granted_by_user_id	uuid	SÍ	FK	platform_user(id)
version	integer	NO		
updated_at	timestamptz	NO		

FK (user_id) → platform_user(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 258.

FK (module_id) → platform_module(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 258.

FK (granted_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 258.

CHECK: CONSTRAINT ck_user_module_access_status CHECK (status IN ('pending', 'active', 'suspended', 'revoked')) · Migración 1786406400000 · línea 258

CHECK: CONSTRAINT ck_user_module_access_version CHECK (version >= 1) · Migración 1786406400000 · línea 258

Índice: CREATE INDEX IF NOT EXISTS ix_user_module_access_user_status ON esquema.user_module_access (user_id, status) · Migración 1786406400000 · línea 283

Índice: CREATE INDEX IF NOT EXISTS ix_user_module_access_module_status ON esquema.user_module_access (module_id, status) · Migración 1786406400000 · línea 287

Fuentes técnicas: Migración 1786406400000 · línea 258.

role

Rol de autorización del dominio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(120)	NO		
name	varchar(160)	NO		
description	text	NO		
is_active	boolean	NO		
is_system	boolean	NO		
is_protected	boolean	NO		
protection_level	integer	NO		
version	integer	NO		
created_at	timestamptz	NO		

CHECK: CONSTRAINT ck_role_code CHECK (code = lower(code) AND code ~ '^[a-z][a-z0-9_]*$') · Migración 1786406400000 · línea 292

CHECK: CONSTRAINT ck_role_version CHECK (version >= 1) · Migración 1786406400000 · línea 292

CHECK: CONSTRAINT ck_role_protection CHECK (protection_level IN (0, 50, 100) AND (is_protected OR protection_level = 0)) · Migración 1788307200000 · línea 375

Índice: CREATE INDEX IF NOT EXISTS ix_role_active_protection ON esquema.role (is_active, protection_level) · Migración 1786406400000 · línea 314

Fuentes técnicas: Migración 1786406400000 · línea 292.

permission

Permiso granular del dominio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(160)	NO		
name	varchar(200)	NO		
category	varchar(80)	NO		
description	text	NO		
is_reserved	boolean	NO		
required_protection_level	integer	NO		
is_active	boolean	NO		

CHECK: CONSTRAINT ck_permission_code CHECK (code = lower(code) AND code ~ '^[a-z][a-z0-9_.]*$') · Migración 1786406400000 · línea 319

CHECK: CONSTRAINT ck_permission_protection CHECK (required_protection_level IN (0, 50, 100) AND ((is_reserved AND required_protection_level > 0) OR (NOT is_reserved AND required_protection_level = 0))) · Migración 1788307200000 · línea 383

Índice: CREATE INDEX IF NOT EXISTS ix_permission_category_active ON esquema.permission (category, is_active) · Migración 1786406400000 · línea 339

Índice: CREATE INDEX IF NOT EXISTS ix_permission_reserved_level ON esquema.permission (is_reserved, required_protection_level) · Migración 1786406400000 · línea 343

Fuentes técnicas: Migración 1786406400000 · línea 319.

user_role

Asignación de roles a usuarios.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: user_id, role_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
user_id	uuid	NO	PK, FK	platform_user(id)
role_id	uuid	NO	PK, FK	role(id)
assigned_by_user_id	uuid	SÍ	FK	platform_user(id)
assigned_at	timestamptz	NO		
expires_at	timestamptz	SÍ		
is_active	boolean	NO		

FK (user_id) → platform_user(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 348.

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 348.

FK (assigned_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 348.

CHECK: CONSTRAINT ck_user_role_expiry CHECK (expires_at IS NULL OR expires_at > assigned_at) · Migración 1786406400000 · línea 348

Índice: CREATE INDEX IF NOT EXISTS ix_user_role_active_role ON esquema.user_role (role_id, user_id) WHERE is_active = true · Migración 1786406400000 · línea 370

Índice: CREATE INDEX IF NOT EXISTS ix_user_role_user_active ON esquema.user_role (user_id, is_active) · Migración 1786406400000 · línea 375

Fuentes técnicas: Migración 1786406400000 · línea 348.

role_permission

Tabla puente entre roles y permisos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: role_id, permission_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
role_id	uuid	NO	PK, FK	role(id)
permission_id	uuid	NO	PK, FK	permission(id)
granted_by_user_id	uuid	SÍ	FK	platform_user(id)
granted_at	timestamptz	NO		

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 380.

FK (permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 380.

FK (granted_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 380.

Índice: CREATE INDEX IF NOT EXISTS ix_role_permission_permission ON esquema.role_permission (permission_id) · Migración 1786406400000 · línea 398

Fuentes técnicas: Migración 1786406400000 · línea 380.

permission_dependency

Dependencias entre permisos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: permission_id, required_permission_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
permission_id	uuid	NO	PK, FK	permission(id)
required_permission_id	uuid	NO	PK, FK	permission(id)

FK (permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 403.

FK (required_permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 403.

CHECK: CONSTRAINT ck_permission_dependency_not_self CHECK (permission_id <> required_permission_id) · Migración 1786406400000 · línea 403

Índice: CREATE INDEX IF NOT EXISTS ix_permission_dependency_required ON esquema.permission_dependency (required_permission_id) · Migración 1786406400000 · línea 418

Fuentes técnicas: Migración 1786406400000 · línea 403.

space_type

Tipo de espacio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(80)	NO		
name	varchar(160)	NO		
description	text	NO		
is_color_anchor	boolean	NO		
can_have_children	boolean	NO		
is_active	boolean	NO		
sort_order	integer	NO		

CHECK: CONSTRAINT ck_space_type_code CHECK (code = lower(code) AND code ~ '^[a-z][a-z0-9_]*$') · Migración 1786406400000 · línea 435

CHECK: CONSTRAINT ck_space_type_name CHECK (btrim(name) <> '') · Migración 1786406400000 · línea 435

Índice: CREATE INDEX IF NOT EXISTS ix_space_type_active_sort ON esquema.space_type (is_active, sort_order) · Migración 1786406400000 · línea 451

Fuentes técnicas: Migración 1786406400000 · línea 435.

space

Espacio canónico del predio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(120)	NO		
name	varchar(200)	NO		
description	text	NO		
space_type_id	uuid	NO	FK	space_type(id)
parent_id	uuid	SÍ	FK	space(id)
status	varchar(32)	NO		
sort_order	integer	NO		
color_hex	varchar(7)	SÍ		
area_m2	numeric(12, 2)	SÍ		
dimensions	jsonb	SÍ		
notes	text	SÍ		
version	integer	NO		
published_at	timestamptz	SÍ		
created_by_user_id	uuid	SÍ	FK	platform_user(id)
updated_by_user_id	uuid	SÍ	FK	platform_user(id)
created_at	timestamptz	NO		

Campo	Tipo	NULL	Clave	Referencia física
updated_at	timestamptz	NO		

FK (space_type_id) → space_type(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 456.

FK (parent_id) → space(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 456.

FK (created_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 456.

FK (updated_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 456.

CHECK: CONSTRAINT ck_space_code CHECK (code = lower(code) AND code ~ '^[a-z0-9][a-z0-9_-]*$') · Migración 1786406400000 · línea 456

CHECK: CONSTRAINT ck_space_status CHECK (status IN ('draft', 'active', 'inactive', 'retired')) · Migración 1786406400000 · línea 456

CHECK: CONSTRAINT ck_space_not_self_parent CHECK (parent_id IS NULL OR parent_id <> id) · Migración 1786406400000 · línea 456

CHECK: CONSTRAINT ck_space_color CHECK (color_hex IS NULL OR color_hex ~ '^#[0-9A-Fa-f]{6}$') · Migración 1786406400000 · línea 456

CHECK: CONSTRAINT ck_space_dimensions CHECK (dimensions IS NULL OR jsonb_typeof(dimensions) = 'object') · Migración 1786406400000 · línea 456

CHECK: CONSTRAINT ck_space_version CHECK (version >= 1) · Migración 1786406400000 · línea 456

CHECK: CONSTRAINT ck_space_timestamps CHECK (updated_at >= created_at) · Migración 1786406400000 · línea 456

CHECK: CONSTRAINT ck_space_area CHECK (area_m2 IS NULL OR area_m2 > 0) · Migración 1788393600000 · línea 113

CHECK: CONSTRAINT ck_space_sort_order_nonnegative CHECK (sort_order >= 0) · Migración 1788393600000 · línea 113

CHECK: CONSTRAINT ck_space_publication CHECK ((status = 'draft' AND published_at IS NULL) OR (status <> 'draft' AND published_at IS NOT NULL)) · Migración 1788393600000 · línea 113

CHECK: CONSTRAINT ck_space_root_type CHECK ((space_type_id = '${PREDIO_SPACE_TYPE.id}'::uuid AND parent_id IS NULL) OR (space_type_id <> '${PREDIO_SPACE_TYPE.id}'::uuid AND parent_id IS NOT NULL)) · Migración 1788393600000 · línea 113

Índice: CREATE INDEX IF NOT EXISTS ix_space_parent_sort ON esquema.space (parent_id, sort_order) · Migración 1786406400000 · línea 504

Índice: CREATE INDEX IF NOT EXISTS ix_space_type_status ON esquema.space (space_type_id, status) · Migración 1786406400000 · línea 508

Índice: CREATE UNIQUE INDEX uq_space_single_root ON esquema.space ((true)) WHERE parent_id IS NULL · Migración 1788393600000 · línea 127

Índice: CREATE INDEX ix_space_status_tree_order ON esquema.space (status, sort_order, code, id) · Migración 1788393600000 · línea 131

Fuentes técnicas: Migración 1786406400000 · línea 456.

space_code_alias

Códigos alternativos/históricos de espacios.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (alias_code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
space_id	uuid	NO	FK	space(id)
alias_code	varchar(120)	NO		
reason	text	NO		
created_by_user_id	uuid	SÍ	FK	platform_user(id)
created_at	timestamptz	NO		

FK (space_id) → space(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 513.

FK (created_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 513.

CHECK: CONSTRAINT ck_space_code_alias_code CHECK (alias_code = lower(alias_code) AND alias_code ~ '^[a-z0-9][a-z0-9_-]*$') · Migración 1786406400000 · línea 513

CHECK: CONSTRAINT ck_space_code_alias_reason CHECK (btrim(reason) <> '') · Migración 1786406400000 · línea 513

Índice: CREATE INDEX IF NOT EXISTS ix_space_code_alias_space ON esquema.space_code_alias (space_id) · Migración 1786406400000 · línea 534

Fuentes técnicas: Migración 1786406400000 · línea 513.

space_parent_history

Historia de parentesco de espacios.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
space_id	uuid	NO	FK	space(id)
parent_space_id	uuid	SÍ	FK	space(id)
valid_from	timestamptz	NO		
valid_to	timestamptz	SÍ		
reason	text	NO		
changed_by_user_id	uuid	SÍ	FK	platform_user(id)
recorded_at	timestamptz	NO		

FK (space_id) → space(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 539.

FK (parent_space_id) → space(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 539.

FK (changed_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 539.

CHECK: CONSTRAINT ck_space_parent_history_not_self CHECK (parent_space_id IS NULL OR parent_space_id <> space_id) · Migración 1786406400000 · línea 539

CHECK: CONSTRAINT ck_space_parent_history_validity CHECK (valid_to IS NULL OR valid_to > valid_from) · Migración 1786406400000 · línea 539

CHECK: CONSTRAINT ck_space_parent_history_reason CHECK (btrim(reason) <> '') · Migración 1786406400000 · línea 539

Índice: CREATE UNIQUE INDEX IF NOT EXISTS uq_space_parent_history_current ON esquema.space_parent_history (space_id) WHERE valid_to IS NULL · Migración 1786406400000 · línea 565

Índice: CREATE INDEX IF NOT EXISTS ix_space_parent_history_parent_valid ON esquema.space_parent_history (parent_space_id, valid_from) · Migración 1786406400000 · línea 570

Índice: CREATE INDEX ix_space_parent_history_space_validity ON esquema.space_parent_history (space_id, valid_from DESC, valid_to) · Migración 1788393600000 · línea 135

Fuentes técnicas: Migración 1786406400000 · línea 539.

space_lineage

Cierre de jerarquía espacial.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (source_space_id, target_space_id, relation_type).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
source_space_id	uuid	NO	FK	space(id)
target_space_id	uuid	NO	FK	space(id)
relation_type	varchar(32)	NO		
effective_at	timestamptz	NO		
reason	text	NO		
created_by_user_id	uuid	SÍ	FK	platform_user(id)
created_at	timestamptz	NO		

FK (source_space_id) → space(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 575.

FK (target_space_id) → space(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 575.

FK (created_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 575.

CHECK: CONSTRAINT ck_space_lineage_not_self CHECK (source_space_id <> target_space_id) · Migración 1786406400000 · línea 575

CHECK: CONSTRAINT ck_space_lineage_relation CHECK (relation_type IN ('split_into', 'merged_into', 'replaced_by')) · Migración 1786406400000 · línea 575

CHECK: CONSTRAINT ck_space_lineage_reason CHECK (btrim(reason) <> '') · Migración 1786406400000 · línea 575

Índice: CREATE INDEX IF NOT EXISTS ix_space_lineage_target ON esquema.space_lineage (target_space_id, effective_at) · Migración 1786406400000 · línea 603

Índice: CREATE INDEX ix_space_lineage_source_effective ON esquema.space_lineage (source_space_id, effective_at) · Migración 1788393600000 · línea 140

Fuentes técnicas: Migración 1786406400000 · línea 575.

space_module_setting

Disponibilidad espacial por módulo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: space_id, module_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
space_id	uuid	NO	PK, FK	space(id)
module_id	uuid	NO	PK, FK	platform_module(id)
selection_state	varchar(32)	NO		
reason	text	NO		
version	integer	NO		
updated_by_user_id	uuid	SÍ	FK	platform_user(id)
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

FK (space_id) → space(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 608.

FK (module_id) → platform_module(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 608.

FK (updated_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 608.

CHECK: CONSTRAINT ck_space_module_setting_state CHECK (selection_state IN ('enabled', 'disabled')) · Migración 1786406400000 · línea 608

CHECK: CONSTRAINT ck_space_module_setting_reason CHECK (btrim(reason) <> '') · Migración 1786406400000 · línea 608

CHECK: CONSTRAINT ck_space_module_setting_version CHECK (version >= 1) · Migración 1786406400000 · línea 608

CHECK: CONSTRAINT ck_space_module_setting_timestamps CHECK (updated_at >= created_at) · Migración 1786406400000 · línea 608

Índice: CREATE INDEX IF NOT EXISTS ix_space_module_setting_module_state ON esquema.space_module_setting (module_id, selection_state) · Migración 1786406400000 · línea 635

Fuentes técnicas: Migración 1786406400000 · línea 608.

service_client

Identidad de un cliente de integración.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (client_code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
module_id	uuid	NO	FK	platform_module(id)
client_code	varchar(120)	NO		
name	varchar(200)	NO		
status	varchar(32)	NO		
auth_method	varchar(32)	NO		
credential_secret_ref	text	NO		
token_version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

FK (module_id) → platform_module(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 647.

CHECK: CONSTRAINT ck_service_client_code CHECK (client_code = lower(client_code) AND client_code ~ '^[a-z][a-z0-9_-]*$') · Migración 1786406400000 · línea 647

CHECK: CONSTRAINT ck_service_client_status CHECK (status IN ('active', 'suspended', 'revoked')) · Migración 1786406400000 · línea 647

CHECK: CONSTRAINT ck_service_client_auth_method CHECK (auth_method IN ('private_key_jwt', 'mtls')) · Migración 1786406400000 · línea 647

CHECK: CONSTRAINT ck_service_client_secret_ref CHECK (btrim(credential_secret_ref) <> '') · Migración 1786406400000 · línea 647

CHECK: CONSTRAINT ck_service_client_version CHECK (token_version >= 1) · Migración 1786406400000 · línea 647

CHECK: CONSTRAINT ck_service_client_timestamps CHECK (updated_at >= created_at) · Migración 1786406400000 · línea 647

Índice: CREATE INDEX IF NOT EXISTS ix_service_client_module_status ON esquema.service_client (module_id, status) · Migración 1786406400000 · línea 675

Fuentes técnicas: Migración 1786406400000 · línea 647.

service_scope

Alcance autorizado de un servicio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: code. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
code	varchar(160)	NO	PK	
description	text	NO		
is_active	boolean	NO		

CHECK: CONSTRAINT ck_service_scope_code CHECK (code = lower(code) AND code ~ '^[a-z][a-z0-9_.:]*$') · Migración 1786406400000 · línea 680

Fuentes técnicas: Migración 1786406400000 · línea 680.

service_client_scope

Tabla puente cliente/alcance.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: service_client_id, scope_code. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
service_client_id	uuid	NO	PK, FK	service_client(id)
scope_code	varchar(160)	NO	PK, FK	service_scope(code)
granted_by_user_id	uuid	SÍ	FK	platform_user(id)
granted_at	timestamptz	NO		

FK (service_client_id) → service_client(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 690.

FK (scope_code) → service_scope(code). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 690.

FK (granted_by_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 690.

Índice: CREATE INDEX IF NOT EXISTS ix_service_client_scope_scope ON esquema.service_client_scope (scope_code) · Migración 1786406400000 · línea 708

Fuentes técnicas: Migración 1786406400000 · línea 690.

idempotency_record

Control de repetición de solicitudes.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: record_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
service_client_id	uuid	SÍ	FK	service_client(id)
idempotency_key	varchar(200)	NO		
operation	varchar(160)	NO		
request_hash	varchar(128)	NO		
response_status	integer	NO		
response_body	jsonb	NO		
created_at	timestamptz	NO		
expires_at	timestamptz	NO		
record_id	bigint	NO	PK	
actor_user_id	uuid	SÍ	FK	platform_user(id)

FK (service_client_id) → service_client(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 713.

FK (actor_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788307200000 · línea 79.

CHECK: CONSTRAINT ck_idempotency_record_values CHECK (btrim(idempotency_key) <> '' AND btrim(operation) <> '' AND btrim(request_hash) <> '' AND response_status BETWEEN 100 AND 599) · Migración 1786406400000 · línea 713

CHECK: CONSTRAINT ck_idempotency_record_lifetime CHECK (expires_at > created_at) · Migración 1786406400000 · línea 713

CHECK: CONSTRAINT ck_idempotency_record_scope CHECK ((service_client_id IS NOT NULL AND actor_user_id IS NULL) OR (service_client_id IS NULL AND actor_user_id IS NOT NULL)) · Migración 1788307200000 · línea 79

Índice: CREATE INDEX IF NOT EXISTS ix_idempotency_record_expires ON esquema.idempotency_record (expires_at) · Migración 1786406400000 · línea 734

Índice: CREATE UNIQUE INDEX uq_idempotency_record_service_key ON esquema.idempotency_record (service_client_id, idempotency_key) WHERE service_client_id IS NOT NULL · Migración 1788307200000 · línea 93

Índice: CREATE UNIQUE INDEX uq_idempotency_record_actor_key ON esquema.idempotency_record (actor_user_id, idempotency_key) WHERE actor_user_id IS NOT NULL · Migración 1788307200000 · línea 98

Fuentes técnicas: Migración 1786406400000 · línea 713; Migración 1788307200000 · línea 75; Migración 1788307200000 · línea 79.

outbox_event

Mensaje saliente transaccional.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: event_id. Unicidad total: (sequence); (aggregate_type, aggregate_id, aggregate_version).

Campo	Tipo	NULL	Clave	Referencia física
event_id	uuid	NO	PK	
sequence	bigint	SÍ		
event_type	varchar(200)	NO		
schema_version	integer	NO		
aggregate_type	varchar(120)	NO		
aggregate_id	uuid	NO		
aggregate_version	integer	NO		
correlation_id	uuid	NO		
payload	jsonb	NO		

Campo	Tipo	NULL	Clave	Referencia física
status	varchar(32)	NO		
attempt_count	integer	NO		
occurred_at	timestamptz	NO		
available_at	timestamptz	NO		
next_attempt_at	timestamptz	SÍ		
published_at	timestamptz	SÍ		
last_error	text	SÍ		
lease_expires_at	timestamptz	SÍ		

CHECK: CONSTRAINT ck_outbox_event_values CHECK (btrim(event_type) <> '' AND schema_version >= 1 AND btrim(aggregate_type) <> '' AND aggregate_version >= 1 AND jsonb_typeof(payload) = 'object') · Migración 1786406400000 · línea 739

CHECK: CONSTRAINT ck_outbox_event_status CHECK (status IN ('pending', 'publishing', 'published', 'failed')) · Migración 1786406400000 · línea 739

CHECK: CONSTRAINT ck_outbox_event_attempts CHECK (attempt_count >= 0) · Migración 1786406400000 · línea 739

CHECK: CONSTRAINT ck_outbox_event_timestamps CHECK (available_at >= occurred_at AND (next_attempt_at IS NULL OR next_attempt_at >= occurred_at) AND (published_at IS NULL OR published_at >= occurred_at)) · Migración 1786406400000 · línea 739

Índice: CREATE INDEX IF NOT EXISTS ix_outbox_event_claim ON esquema.outbox_event (status, available_at, next_attempt_at, sequence) · Migración 1786406400000 · línea 775

Índice: CREATE INDEX IF NOT EXISTS ix_outbox_event_aggregate ON esquema.outbox_event (aggregate_type, aggregate_id, aggregate_version) · Migración 1786406400000 · línea 779

Índice: CREATE INDEX ix_outbox_event_lease_recovery ON esquema.outbox_event (lease_expires_at, sequence) WHERE status = 'publishing' · Migración 1789516800000 · línea 57

Fuentes técnicas: Migración 1786406400000 · línea 739; Migración 1789516800000 · línea 53.

space_catalog_change

Registro incremental de cambios del catálogo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: sequence. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
sequence	bigint	NO	PK	
space_id	uuid	NO	FK	space(id)
module_id	uuid	SÍ	FK	platform_module(id)
entity_version	integer	NO		
change_type	varchar(80)	NO		
projection_payload	jsonb	NO		
outbox_event_id	uuid	SÍ		
occurred_at	timestamptz	NO		

FK (space_id) → space(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1786406400000 · línea 784.

FK (module_id) → platform_module(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 784.

CHECK: CONSTRAINT ck_space_catalog_change_values CHECK (entity_version >= 1 AND btrim(change_type) <> '' AND jsonb_typeof(projection_payload) = 'object') · Migración 1786406400000 · línea 784

Índice: CREATE INDEX IF NOT EXISTS ix_space_catalog_change_space_sequence ON esquema.space_catalog_change (space_id, sequence) · Migración 1786406400000 · línea 806

Índice: CREATE INDEX IF NOT EXISTS ix_space_catalog_change_module_sequence ON esquema.space_catalog_change (module_id, sequence) · Migración 1786406400000 · línea 810

Índice: CREATE UNIQUE INDEX IF NOT EXISTS uq_space_catalog_change_outbox_event ON esquema.space_catalog_change (outbox_event_id) WHERE outbox_event_id IS NOT NULL · Migración 1786406400000 · línea 814

Índice: CREATE INDEX ix_space_catalog_change_module_space_sequence ON esquema.space_catalog_change (module_id, space_id, sequence DESC) · Migración 1788393600000 · línea 145

Fuentes técnicas: Migración 1786406400000 · línea 784.

audit_log

Registro canónico de auditoría particionado.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: occurred_at, id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
occurred_at	timestamptz	NO	PK	
id	uuid	NO	PK	
domain	varchar(32)	NO		
action	varchar(200)	NO		
actor_type	varchar(32)	NO		
actor_user_id	uuid	SÍ	FK	platform_user(id)
actor_service_client_id	uuid	SÍ	FK	service_client(id)
session_id	uuid	SÍ	FK	platform_session(id)
actor_email_snapshot	varchar(320)	SÍ		
actor_name_snapshot	varchar(200)	SÍ		
target_type	varchar(160)	NO		
target_id	varchar(320)	SÍ		
result	varchar(32)	NO		
reason	text	SÍ		
before_data	jsonb	SÍ		
after_data	jsonb	SÍ		
request_id	uuid	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
correlation_id	uuid	SÍ		
source_ip	inet	SÍ		
user_agent	text	SÍ		
required_protection_level	integer	NO		
error_code	varchar(160)	SÍ		

FK (actor_user_id) → platform_user(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 827.

FK (actor_service_client_id) → service_client(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 827.

FK (session_id) → platform_session(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1786406400000 · línea 827.

CHECK: CONSTRAINT ck_audit_log_action CHECK (btrim(action) <> '') · Migración 1786406400000 · línea 827

CHECK: CONSTRAINT ck_audit_log_actor_type CHECK (actor_type IN ('user', 'service', 'system')) · Migración 1786406400000 · línea 827

CHECK: CONSTRAINT ck_audit_log_email_snapshot CHECK (actor_email_snapshot IS NULL OR actor_email_snapshot = lower(btrim(actor_email_snapshot))) · Migración 1786406400000 · línea 827

CHECK: CONSTRAINT ck_audit_log_target_type CHECK (btrim(target_type) <> '') · Migración 1786406400000 · línea 827

CHECK: CONSTRAINT ck_audit_log_result CHECK (result IN ('success', 'denied', 'failure')) · Migración 1786406400000 · línea 827

CHECK: CONSTRAINT ck_audit_log_before_data CHECK (before_data IS NULL OR jsonb_typeof(before_data) = 'object') · Migración 1786406400000 · línea 827

CHECK: CONSTRAINT ck_audit_log_after_data CHECK (after_data IS NULL OR jsonb_typeof(after_data) = 'object') · Migración 1786406400000 · línea 827

CHECK: CONSTRAINT ck_audit_log_domain CHECK (domain IN ('auth', 'access', 'rbac', 'spaces', 'admin', 'audit')) · Migración 1788307200000 · línea 120

CHECK: CONSTRAINT ck_audit_log_protection_level CHECK (required_protection_level IN (0, 50, 100)) · Migración 1788307200000 · línea 120

CHECK: CONSTRAINT ck_audit_log_error_code CHECK (error_code IS NULL OR btrim(error_code) <> '') · Migración 1788307200000 · línea 120

CHECK: CONSTRAINT ck_audit_log_actor CHECK ((actor_type = 'user' AND actor_user_id IS NOT NULL AND actor_service_client_id IS NULL) OR (actor_type = 'service' AND actor_service_client_id IS NOT NULL) OR (actor_type = 'system' AND actor_user_id IS NULL AND actor_service_client_id IS NULL)) · Migración 1789516800000 · línea 63

Índice: CREATE INDEX IF NOT EXISTS ix_audit_log_actor_user_occurred ON esquema.audit_log (actor_user_id, occurred_at DESC) · Migración 1786406400000 · línea 903

Índice: CREATE INDEX IF NOT EXISTS ix_audit_log_actor_service_occurred ON esquema.audit_log (actor_service_client_id, occurred_at DESC) · Migración 1786406400000 · línea 907

Índice: CREATE INDEX IF NOT EXISTS ix_audit_log_domain_action_occurred ON esquema.audit_log (domain, action, occurred_at DESC) · Migración 1786406400000 · línea 911

Índice: CREATE INDEX IF NOT EXISTS ix_audit_log_target_occurred ON esquema.audit_log (target_type, target_id, occurred_at DESC) · Migración 1786406400000 · línea 915

Índice: CREATE INDEX IF NOT EXISTS ix_audit_log_request ON esquema.audit_log (request_id) WHERE request_id IS NOT NULL · Migración 1786406400000 · línea 919

Índice: CREATE INDEX IF NOT EXISTS ix_audit_log_correlation ON esquema.audit_log (correlation_id) WHERE correlation_id IS NOT NULL · Migración 1786406400000 · línea 924

Índice: CREATE INDEX ix_audit_log_visibility_occurred ON esquema.audit_log (required_protection_level, occurred_at DESC, id DESC) · Migración 1788307200000 · línea 130

Índice: CREATE INDEX ix_audit_log_id_occurred ON esquema.audit_log (id, occurred_at DESC) · Migración 1788307200000 · línea 134

Fuentes técnicas: Migración 1786406400000 · línea 827; Migración 1788307200000 · línea 115.

81 tablas lógicas · 215 FK locales · 17 láminas de dominio.

Proyecciones de Core

maintenance_member y las proyecciones espaciales mantienen identificadores externos de Core. Las FK de Mantenimiento apuntan a esas tablas locales. La coincidencia del UUID no representa una FK hacia el esquema central.

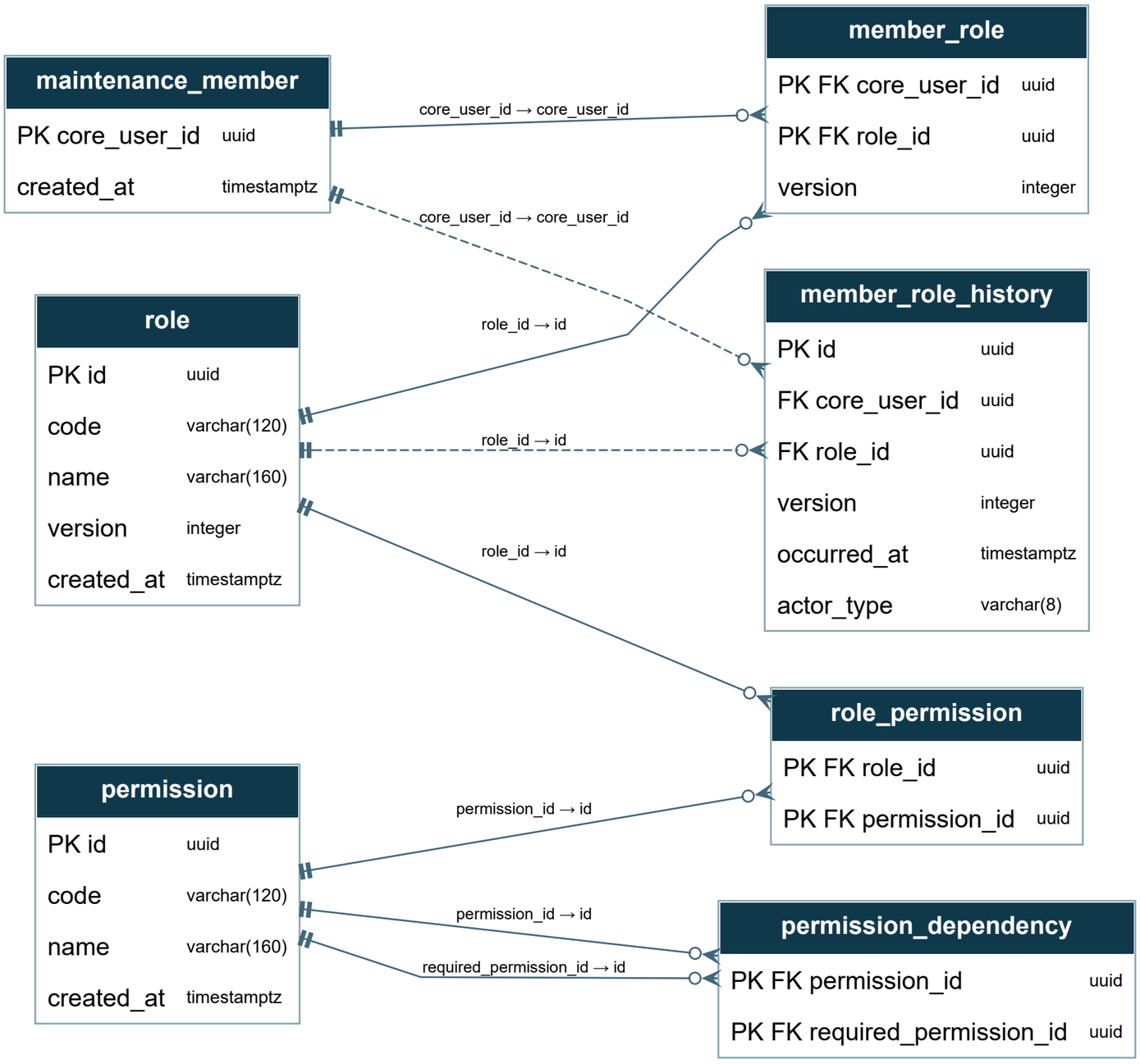
Objetivos, actores y documentos

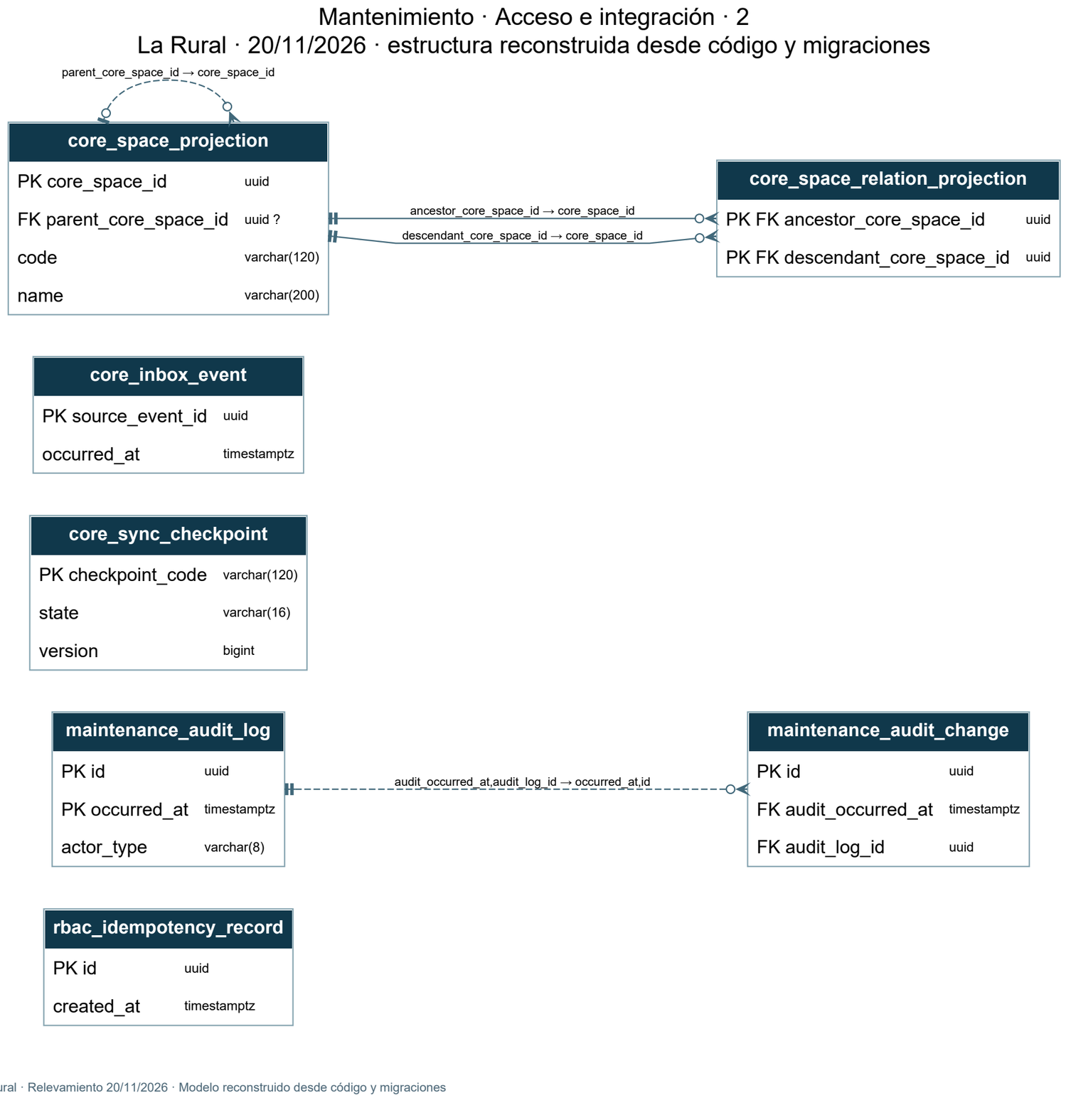
Los objetivos de las órdenes de trabajo, los actores y los propietarios documentales pueden tener alternativas condicionadas por discriminadores y restricciones CHECK. Las FK opcionales del DER no significan que todas las alternativas puedan estar ausentes o presentes simultáneamente. El diccionario conserva las condiciones declaradas.

Historia y versiones

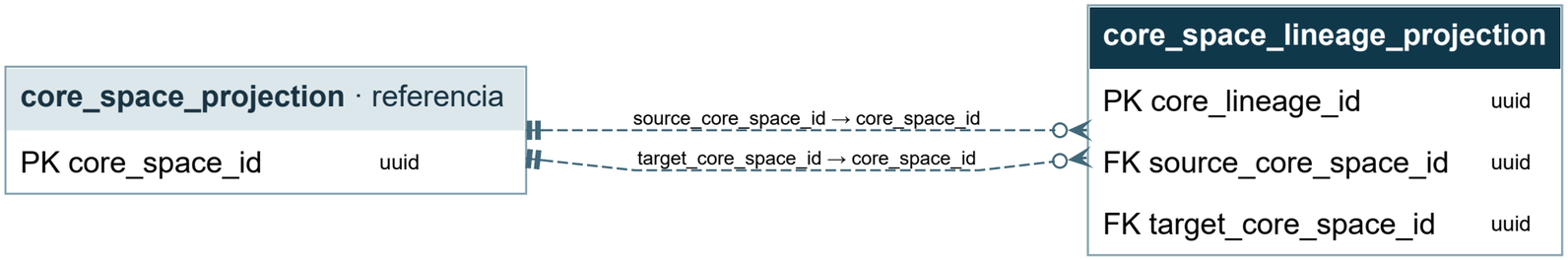
Las revisiones de documentos, intervenciones, hechos y decisiones preservan la evolución de la operación. Los snapshots JSON conservan contexto histórico y no se convierten en entidades ni FK sin una declaración explícita.

Mantenimiento · Acceso e integración · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



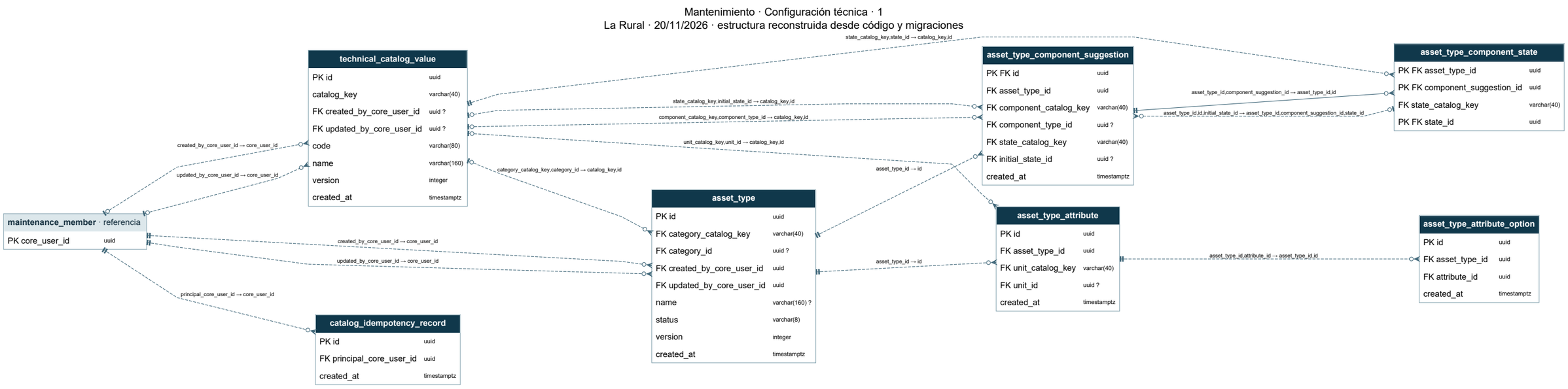

Mantenimiento · Acceso e integración · 2
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
parent_core_space_id → core_space_id
core_space_projection
PK core_space_id uuid
FK parent_core_space_id uuid ?
code varchar(120)
name varchar(200)
ancestor_core_space_id → core_space_id
descendant_core_space_id → core_space_id
core_space_relation_projection
PK FK ancestor_core_space_id uuid
PK FK descendant_core_space_id uuid
core_inbox_event
PK source_event_id uuid
occurred_at timestamptz
core_sync_checkpoint
PK checkpoint_code varchar(120)
state varchar(16)
version bigint
maintenance_audit_log
PK id uuid
PK occurred_at timestamptz
actor_type varchar(8)
audit_occurred_at,audit_log_id → occurred_at,id
maintenance_audit_change
PK id uuid
FK audit_occurred_at timestamptz
FK audit_log_id uuid
rbac_idempotency_record
PK id uuid
created_at timestamptz

Mantenimiento · Acceso e integración · 3
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

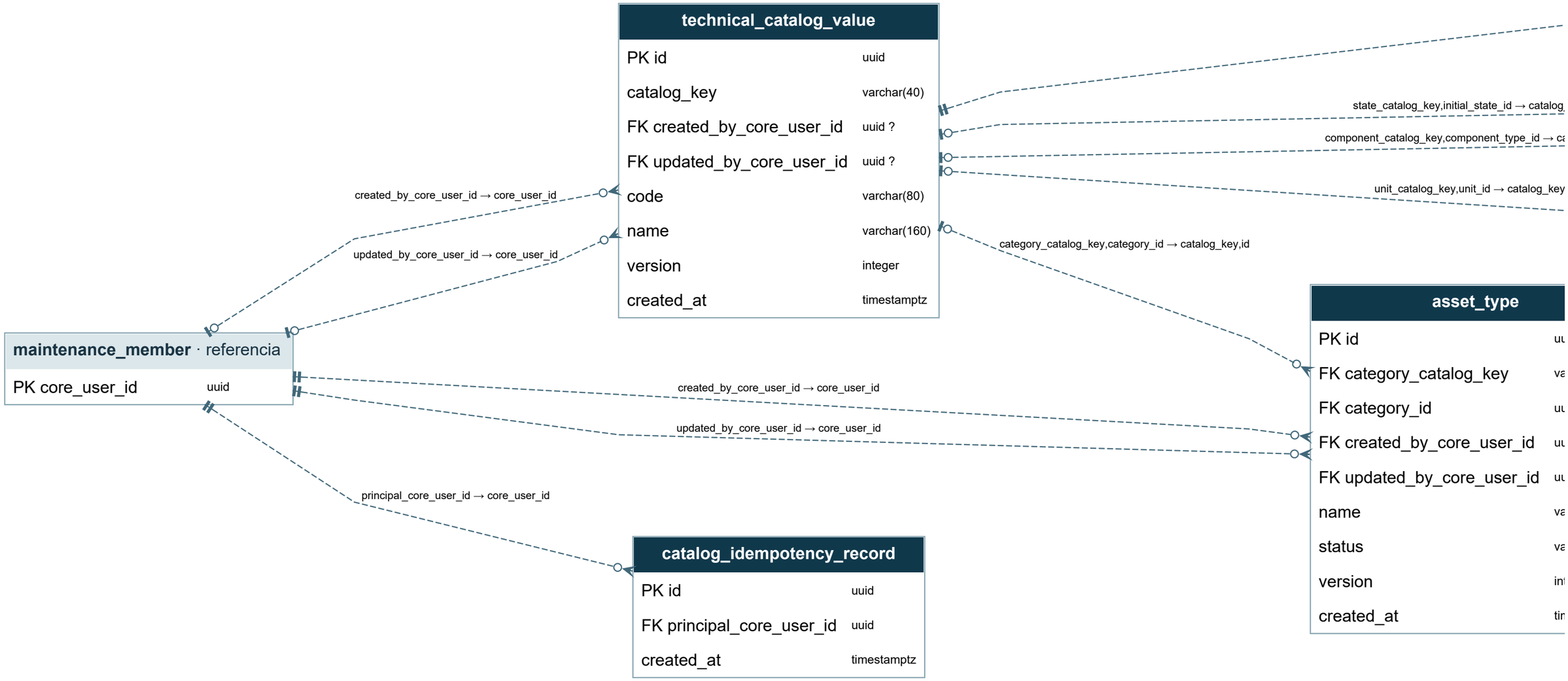


La Rural · modelo de datos

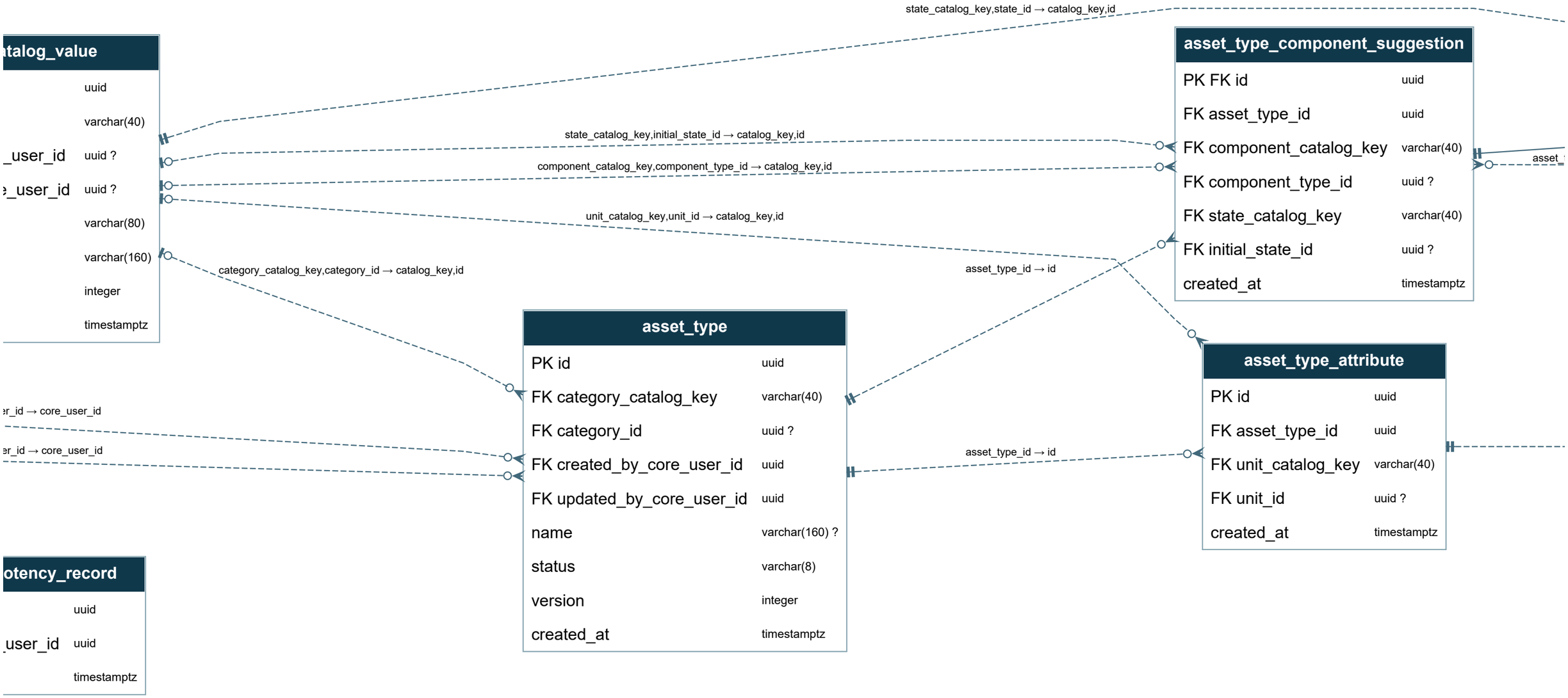
Mantenimiento · Configuración técnica · 1



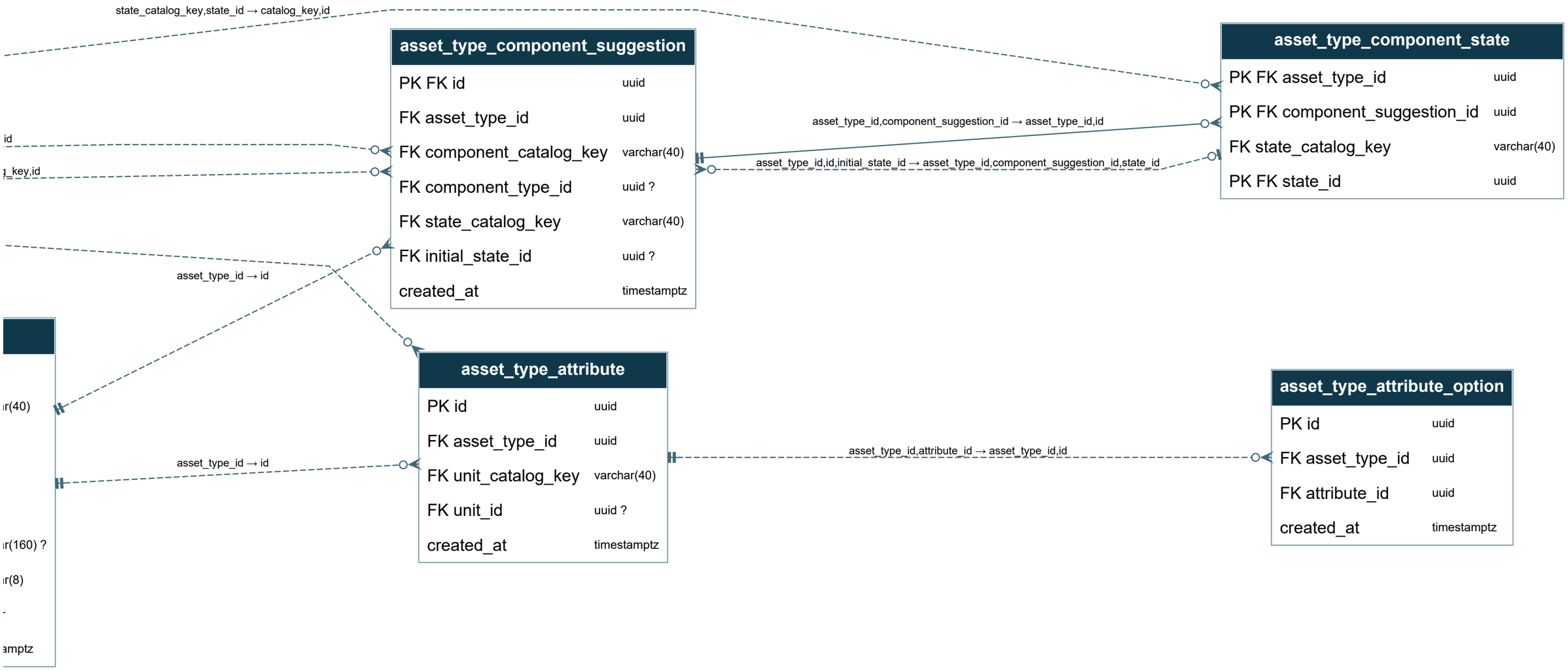
Mantenimiento · Co
La Rural · 20/11/2026 · estructura re



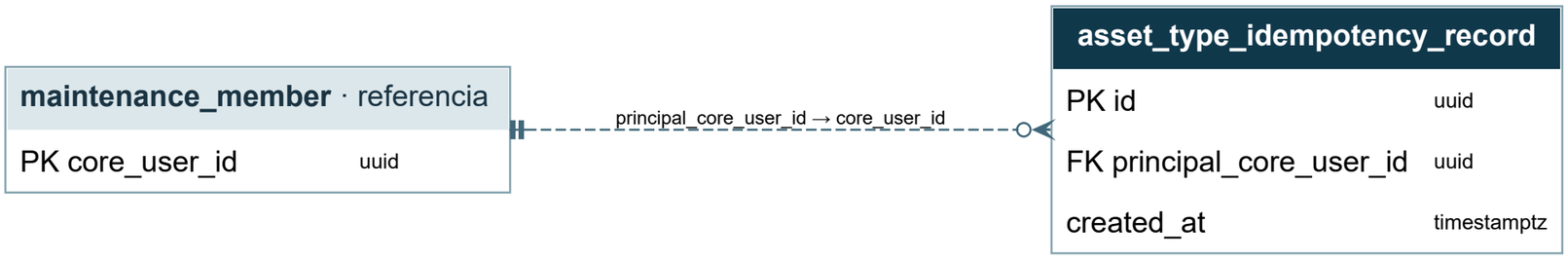
Mantenimiento · Configuración técnica · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

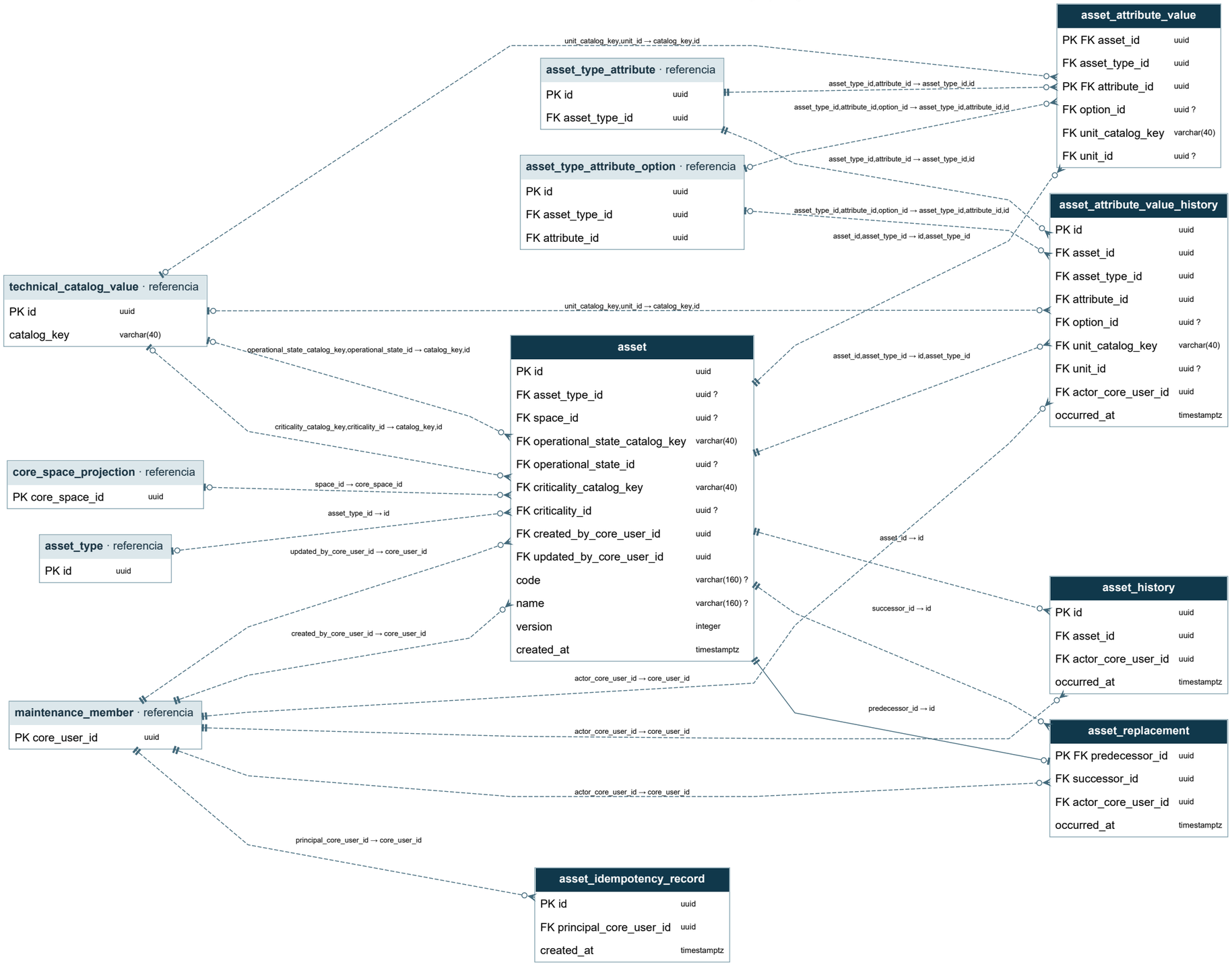


guración técnica · 1
nstruida desde código y migraciones

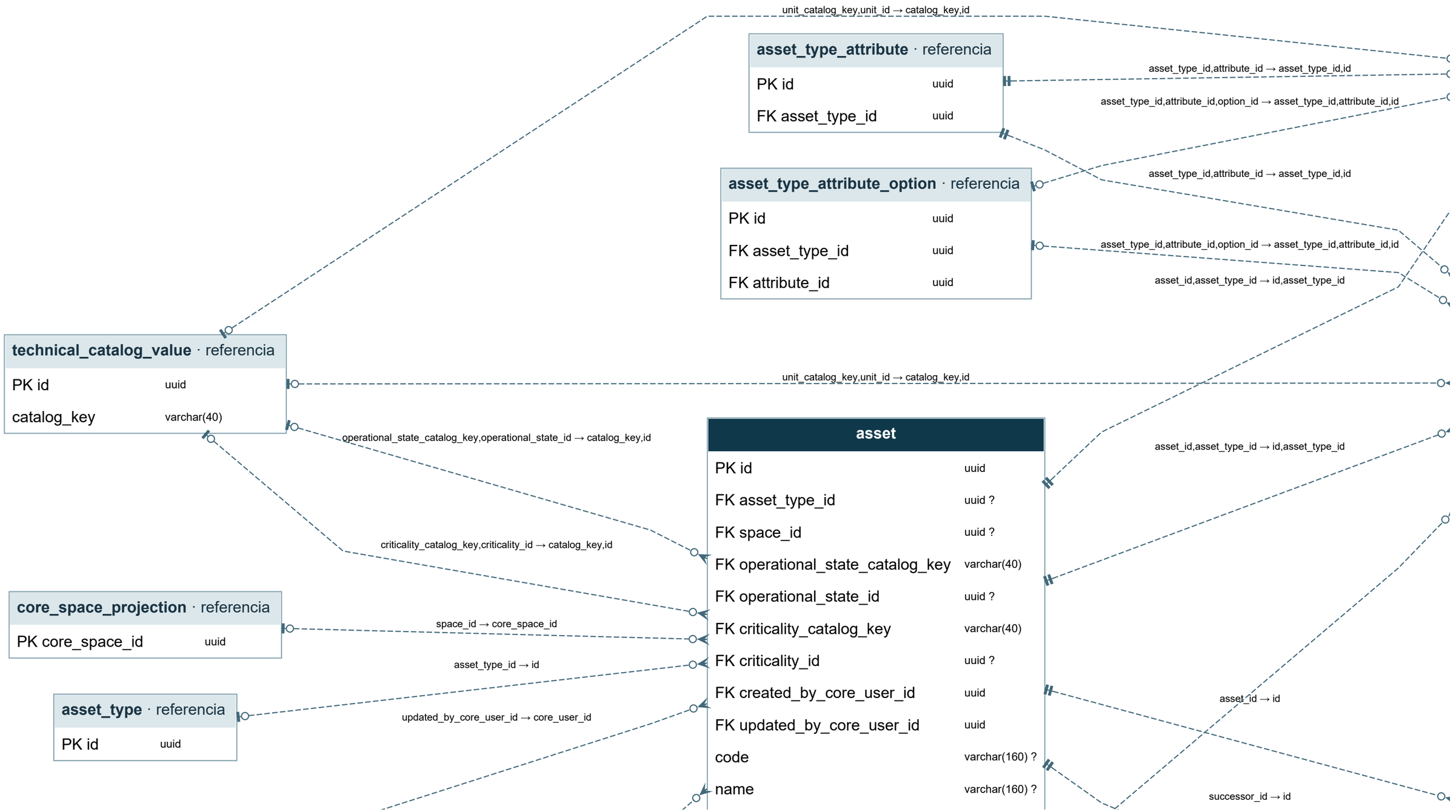


Mantenimiento · Configuración técnica · 2
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

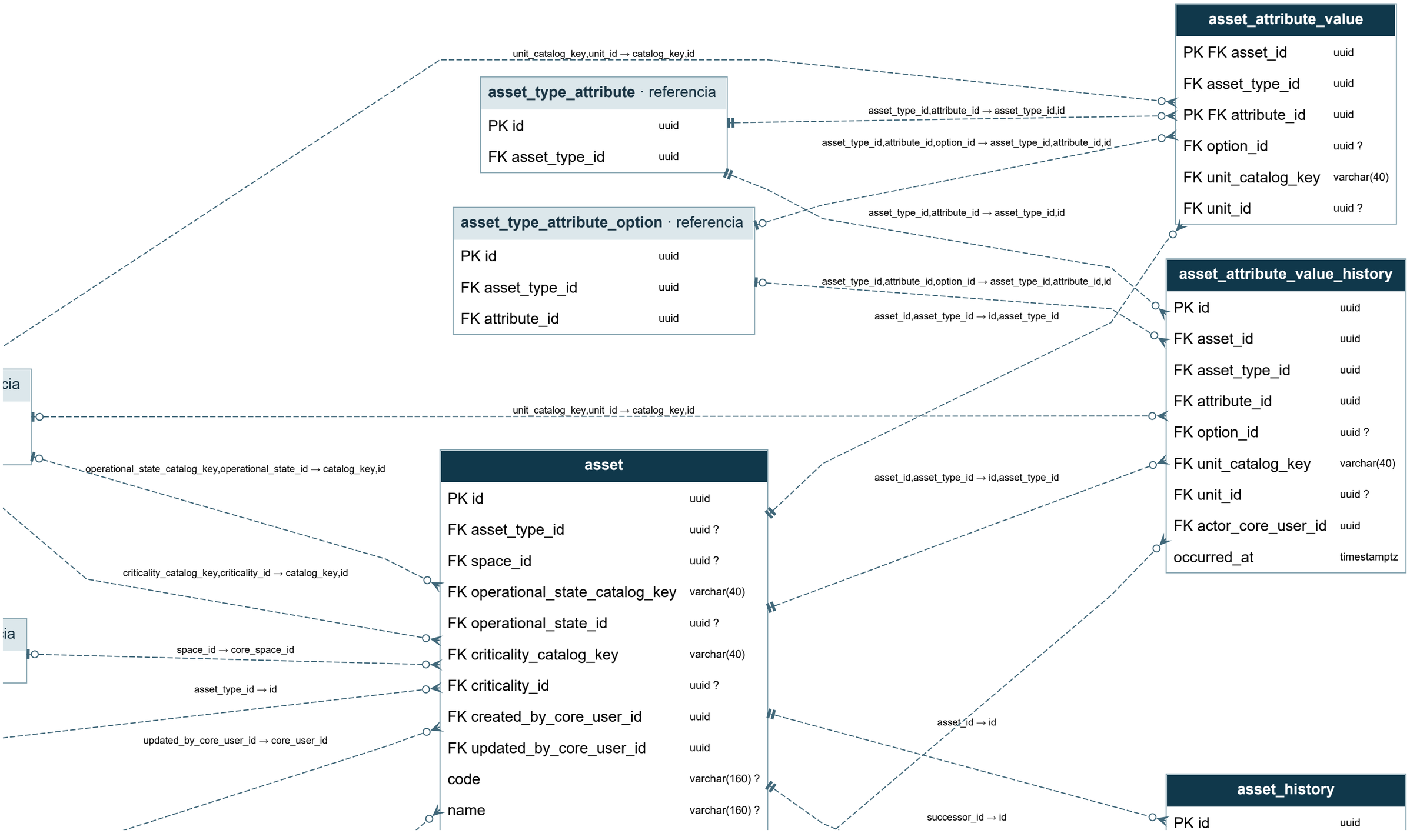


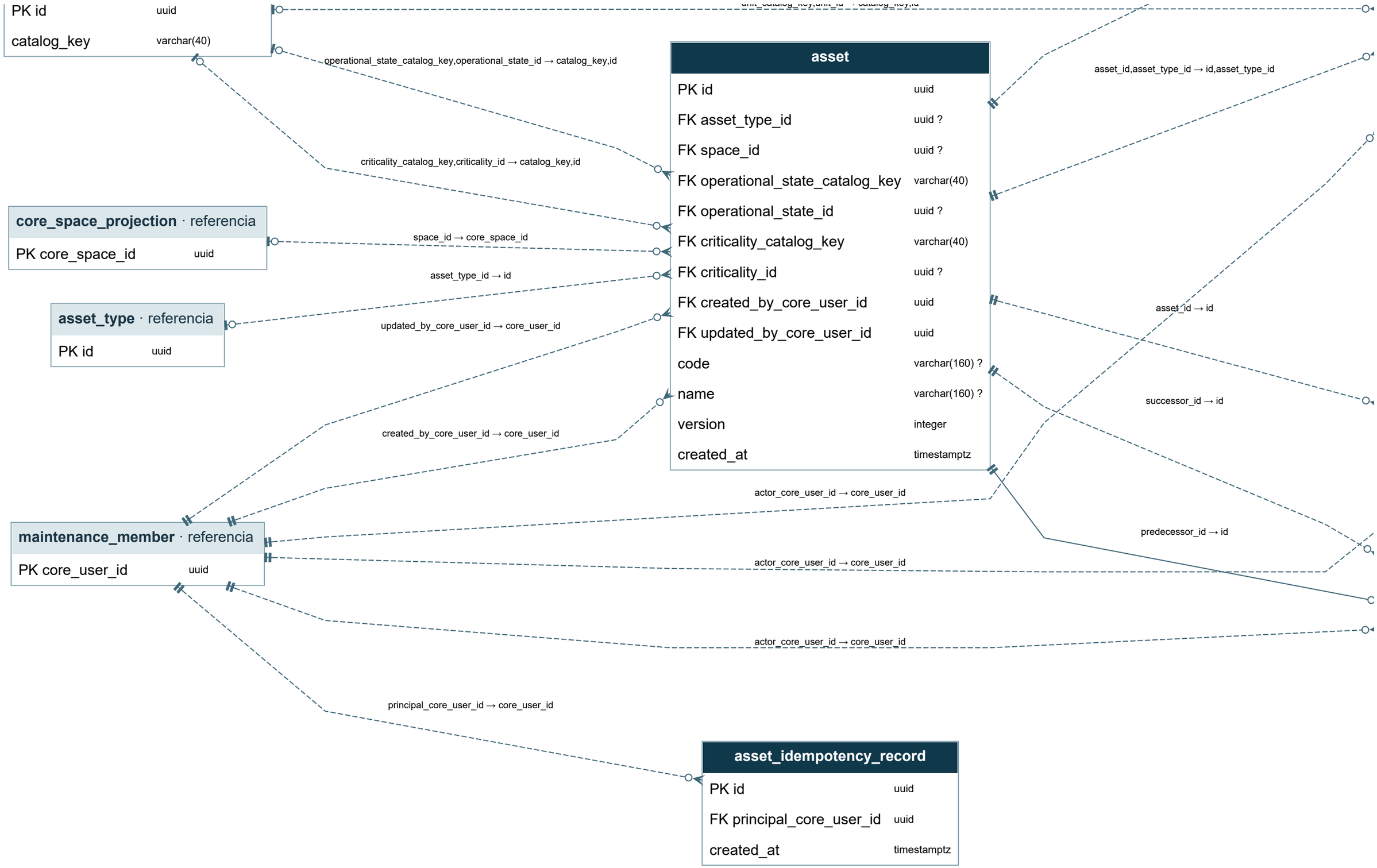

Mantenimiento · Activos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
asset_attribute_value
PK FK asset_id uuid
FK asset_type_id uuid
PK FK attribute_id uuid
FK option_id uuid ?
FK unit_catalog_key varchar(40)
FK unit_id uuid ?
asset_type_attribute · referencia
PK id uuid
FK asset_type_id uuid
asset_type_attribute_option · referencia
PK id uuid
FK asset_type_id uuid
FK attribute_id uuid
asset_attribute_value_history
PK id uuid
FK asset_id uuid
FK asset_type_id uuid
FK attribute_id uuid
FK option_id uuid ?
FK unit_catalog_key varchar(40)
FK unit_id uuid ?
FK actor_core_user_id uuid
occurred_at timestamptz
technical_catalog_value · referencia
PK id uuid
catalog_key varchar(40)
asset
PK id uuid
FK asset_type_id uuid ?
FK space_id uuid ?
FK operational_state_catalog_key varchar(40)
FK operational_state_id uuid ?
FK criticality_catalog_key varchar(40)
FK criticality_id uuid ?
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
code varchar(160) ?
name varchar(160) ?
version integer
created_at timestamptz
core_space_projection · referencia
PK core_space_id uuid
asset_type · referencia
PK id uuid
asset_history
PK id uuid
FK asset_id uuid
FK actor_core_user_id uuid
occurred_at timestamptz
asset_replacement
PK FK predecessor_id uuid
FK successor_id uuid
FK actor_core_user_id uuid
occurred_at timestamptz
maintenance_member · referencia
PK core_user_id uuid
asset_idempotency_record
PK id uuid
FK principal_core_user_id uuid
created_at timestamptz
unit_catalog_key,unit_id → catalog_key,id
asset_type_id,attribute_id → asset_type_id,id
asset_type_id,attribute_id,option_id → asset_type_id,attribute_id,id
asset_type_id,attribute_id → asset_type_id,id
asset_type_id,attribute_id,option_id → asset_type_id,attribute_id,id
asset_id,asset_type_id → id,asset_type_id
unit_catalog_key,unit_id → catalog_key,id
operational_state_catalog_key,operational_state_id → catalog_key,id
asset_id,asset_type_id → id,asset_type_id
criticality_catalog_key,criticality_id → catalog_key,id
space_id → core_space_id
asset_type_id → id
updated_by_core_user_id → core_user_id
asset_id → id
created_by_core_user_id → core_user_id
successor_id → id
actor_core_user_id → core_user_id
predecessor_id → id
actor_core_user_id → core_user_id
actor_core_user_id → core_user_id
principal_core_user_id → core_user_id

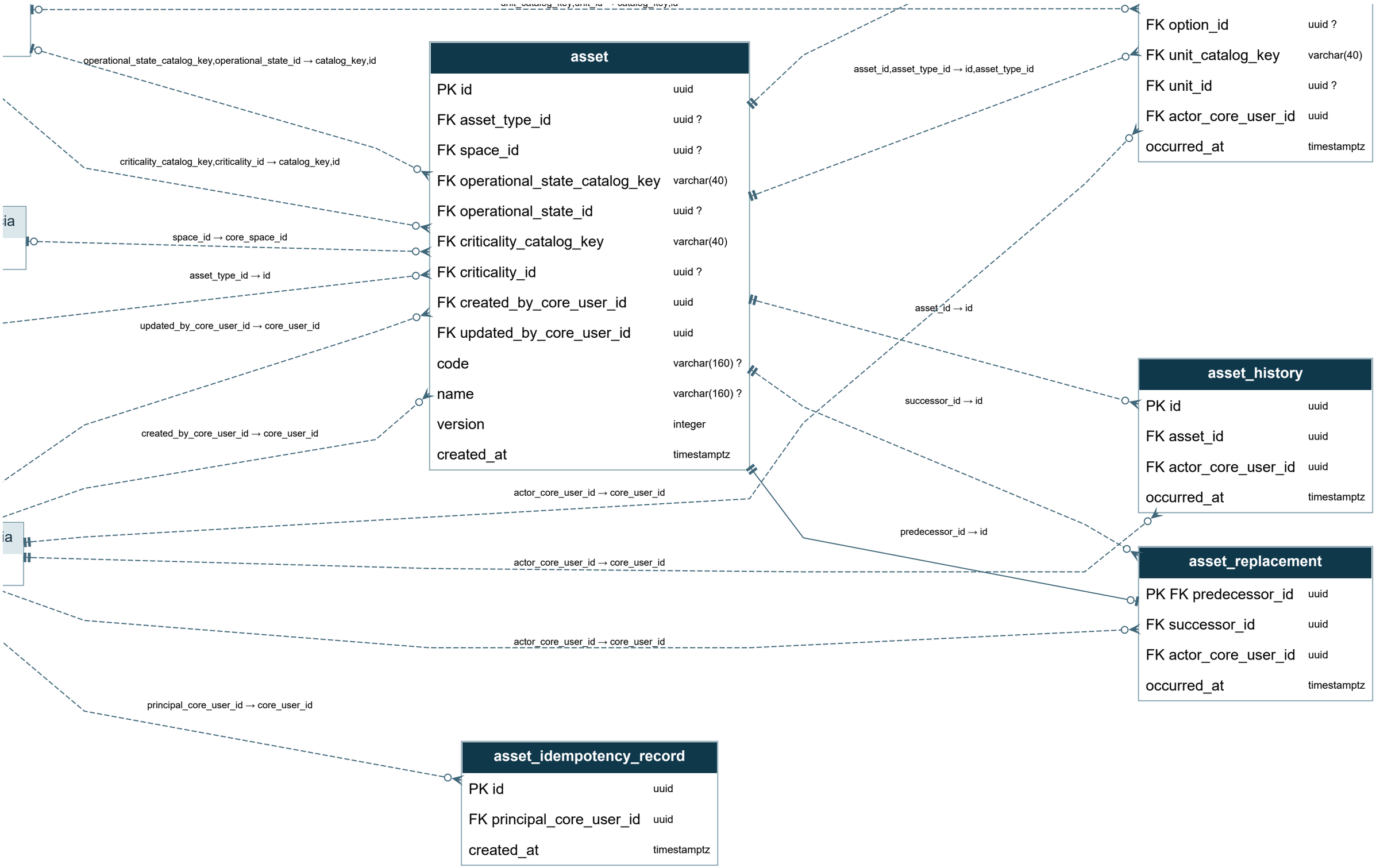
Mantenimiento · Activos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

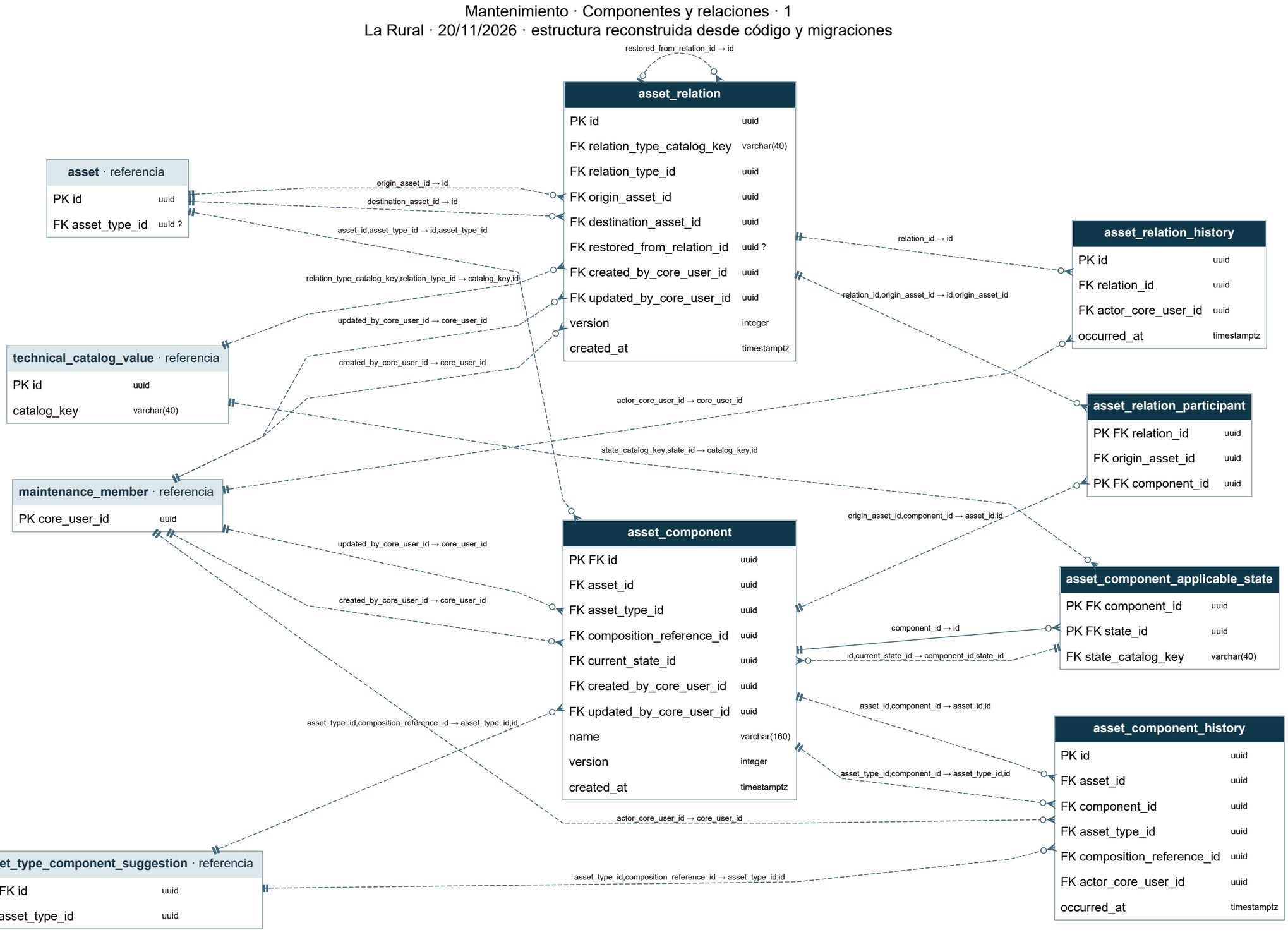


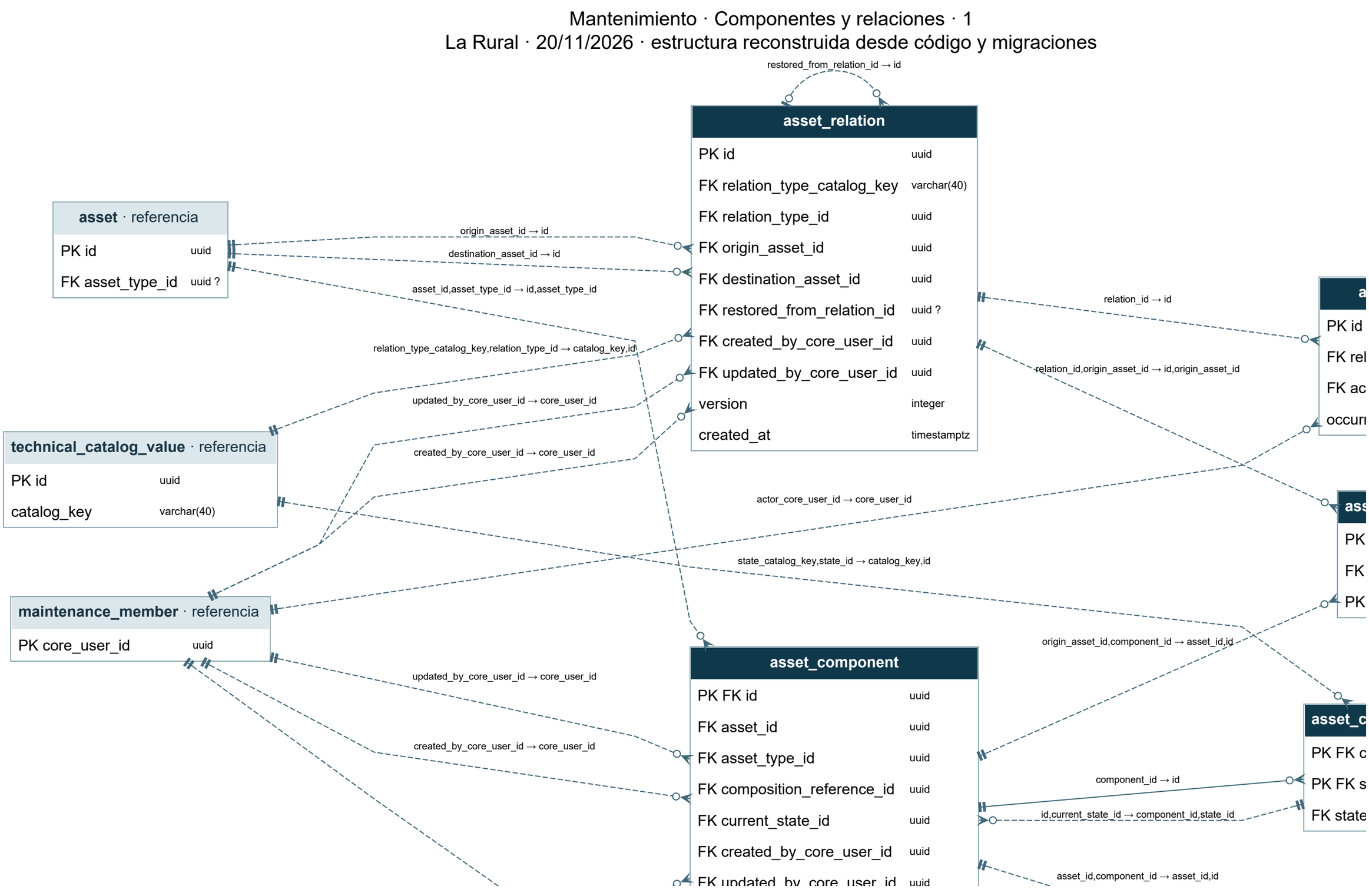
Mantenimiento · Activos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones




PK id uuid
catalog_key varchar(40)
operational_state_catalog_key,operational_state_id → catalog_key,id
criticality_catalog_key,criticality_id → catalog_key,id
asset
PK id uuid
FK asset_type_id uuid ?
FK space_id uuid ?
FK operational_state_catalog_key varchar(40)
FK operational_state_id uuid ?
FK criticality_catalog_key varchar(40)
FK criticality_id uuid ?
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
code varchar(160) ?
name varchar(160) ?
version integer
created_at timestamptz
asset_id,asset_type_id → id,asset_type_id
core_space_projection · referencia
PK core_space_id uuid
space_id → core_space_id
asset_type_id → id
asset_type · referencia
PK id uuid
updated_by_core_user_id → core_user_id
asset_id → id
successor_id → id
created_by_core_user_id → core_user_id
actor_core_user_id → core_user_id
predecessor_id → id
maintenance_member · referencia
PK core_user_id uuid
actor_core_user_id → core_user_id
actor_core_user_id → core_user_id
principal_core_user_id → core_user_id
asset_idempotency_record
PK id uuid
FK principal_core_user_id uuid
created_at timestamptz

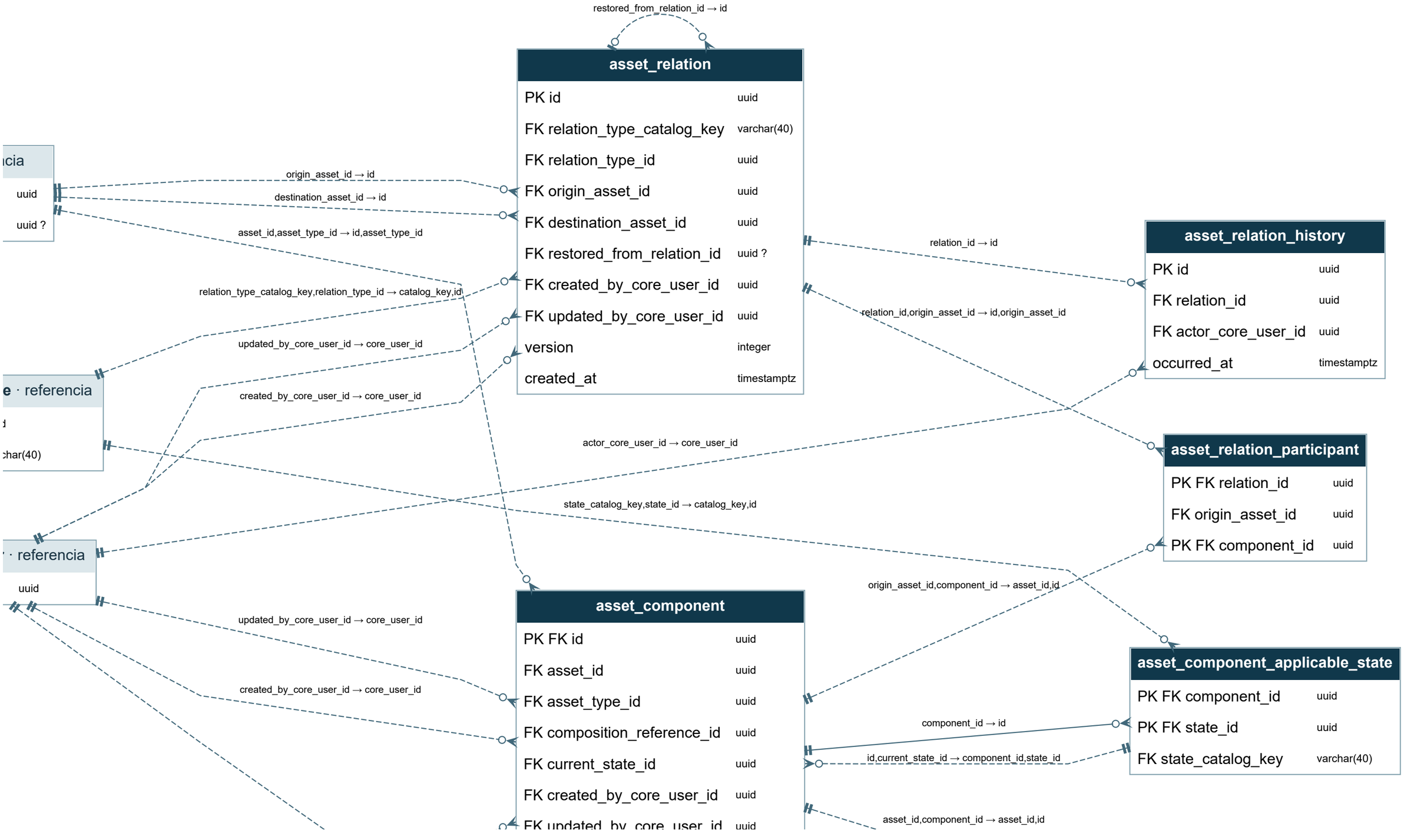

operational_state_catalog_key,operational_state_id → catalog_key,id
criticality_catalog_key,criticality_id → catalog_key,id
space_id → core_space_id
asset_type_id → id
updated_by_core_user_id → core_user_id
created_by_core_user_id → core_user_id
actor_core_user_id → core_user_id
actor_core_user_id → core_user_id
actor_core_user_id → core_user_id
principal_core_user_id → core_user_id
asset_id,asset_type_id → id,asset_type_id
asset_id → id
successor_id → id
predecessor_id → id
asset
PK id uuid
FK asset_type_id uuid ?
FK space_id uuid ?
FK operational_state_catalog_key varchar(40)
FK operational_state_id uuid ?
FK criticality_catalog_key varchar(40)
FK criticality_id uuid ?
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
code varchar(160) ?
name varchar(160) ?
version integer
created_at timestamptz
FK option_id uuid ?
FK unit_catalog_key varchar(40)
FK unit_id uuid ?
FK actor_core_user_id uuid
occurred_at timestamptz
asset_history
PK id uuid
FK asset_id uuid
FK actor_core_user_id uuid
occurred_at timestamptz
asset_replacement
PK FK predecessor_id uuid
FK successor_id uuid
FK actor_core_user_id uuid
occurred_at timestamptz
asset_idempotency_record
PK id uuid
FK principal_core_user_id uuid
created_at timestamptz

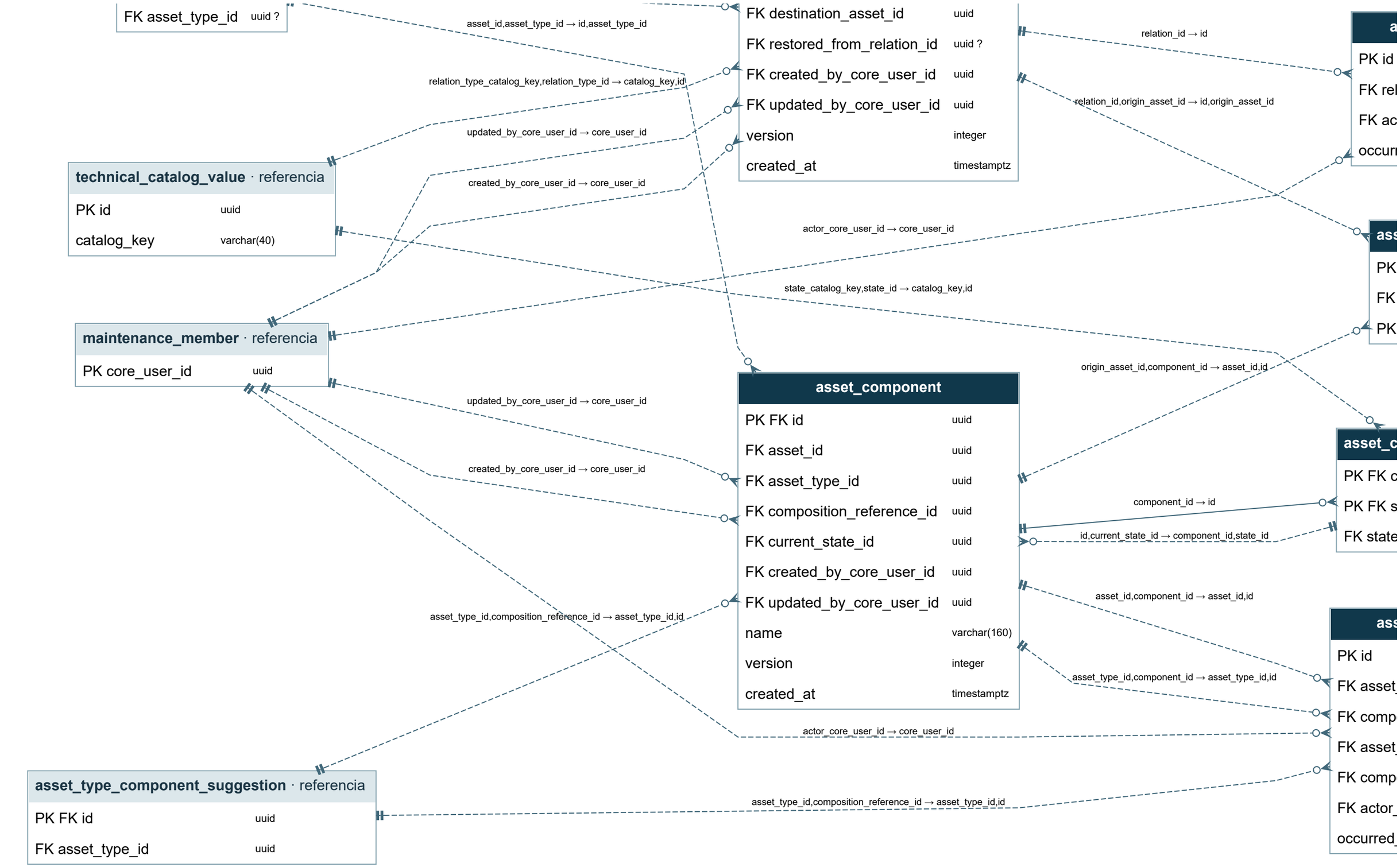

Mantenimiento · Componentes y relaciones · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
restored_from_relation_id → id
asset_relation
PK id uuid
FK relation_type_catalog_key varchar(40)
FK relation_type_id uuid
FK origin_asset_id uuid
FK destination_asset_id uuid
FK restored_from_relation_id uuid ?
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
version integer
created_at timestamptz
asset · referencia
PK id uuid
FK asset_type_id uuid ?
origin_asset_id → id
destination_asset_id → id
asset_id,asset_type_id → id,asset_type_id
relation_type_catalog_key,relation_type_id → catalog_key,id
updated_by_core_user_id → core_user_id
created_by_core_user_id → core_user_id
relation_id → id
relation_id,origin_asset_id → id,origin_asset_id
asset_relation_history
PK id uuid
FK relation_id uuid
FK actor_core_user_id uuid
occurred_at timestamptz
technical_catalog_value · referencia
PK id uuid
catalog_key varchar(40)
actor_core_user_id → core_user_id
state_catalog_key,state_id → catalog_key,id
asset_relation_participant
PK FK relation_id uuid
FK origin_asset_id uuid
PK FK component_id uuid
maintenance_member · referencia
PK core_user_id uuid
origin_asset_id,component_id → asset_id,id
asset_component
PK FK id uuid
FK asset_id uuid
FK asset_type_id uuid
FK composition_reference_id uuid
FK current_state_id uuid
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
name varchar(160)
version integer
created_at timestamptz
asset_component_applicable_state
PK FK component_id uuid
PK FK state_id uuid
FK state_catalog_key varchar(40)
component_id → id
id,current_state_id → component_id,state_id
asset_id,component_id → asset_id,id
asset_component_history
PK id uuid
FK asset_id uuid
FK component_id uuid
FK asset_type_id uuid
FK composition_reference_id uuid
FK actor_core_user_id uuid
occurred_at timestamptz
asset_type_id,composition_reference_id → asset_type_id,id
asset_type_component_suggestion · referencia
PK FK id uuid
FK asset_type_id uuid
asset_type_id,composition_reference_id → asset_type_id,id


Mantenimiento · Componentes y relaciones · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
restored_from_relation_id → id
asset_relation
PK id uuid
FK relation_type_catalog_key varchar(40)
FK relation_type_id uuid
FK origin_asset_id uuid
FK destination_asset_id uuid
FK restored_from_relation_id uuid ?
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
version integer
created_at timestamptz
asset · referencia
PK id uuid
FK asset_type_id uuid ?
technical_catalog_value · referencia
PK id uuid
catalog_key varchar(40)
maintenance_member · referencia
PK core_user_id uuid
asset_component
PK FK id uuid
FK asset_id uuid
FK asset_type_id uuid
FK composition_reference_id uuid
FK current_state_id uuid
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
origin_asset_id → id
destination_asset_id → id
asset_id,asset_type_id → id,asset_type_id
relation_type_catalog_key,relation_type_id → catalog_key,id
updated_by_core_user_id → core_user_id
created_by_core_user_id → core_user_id
relation_id → id
relation_id,origin_asset_id → id,origin_asset_id
actor_core_user_id → core_user_id
state_catalog_key,state_id → catalog_key,id
origin_asset_id,component_id → asset_id,id
component_id → id
id,current_state_id → component_id,state_id
asset_id,component_id → asset_id,id

Mantenimiento · Componentes y relaciones · 1

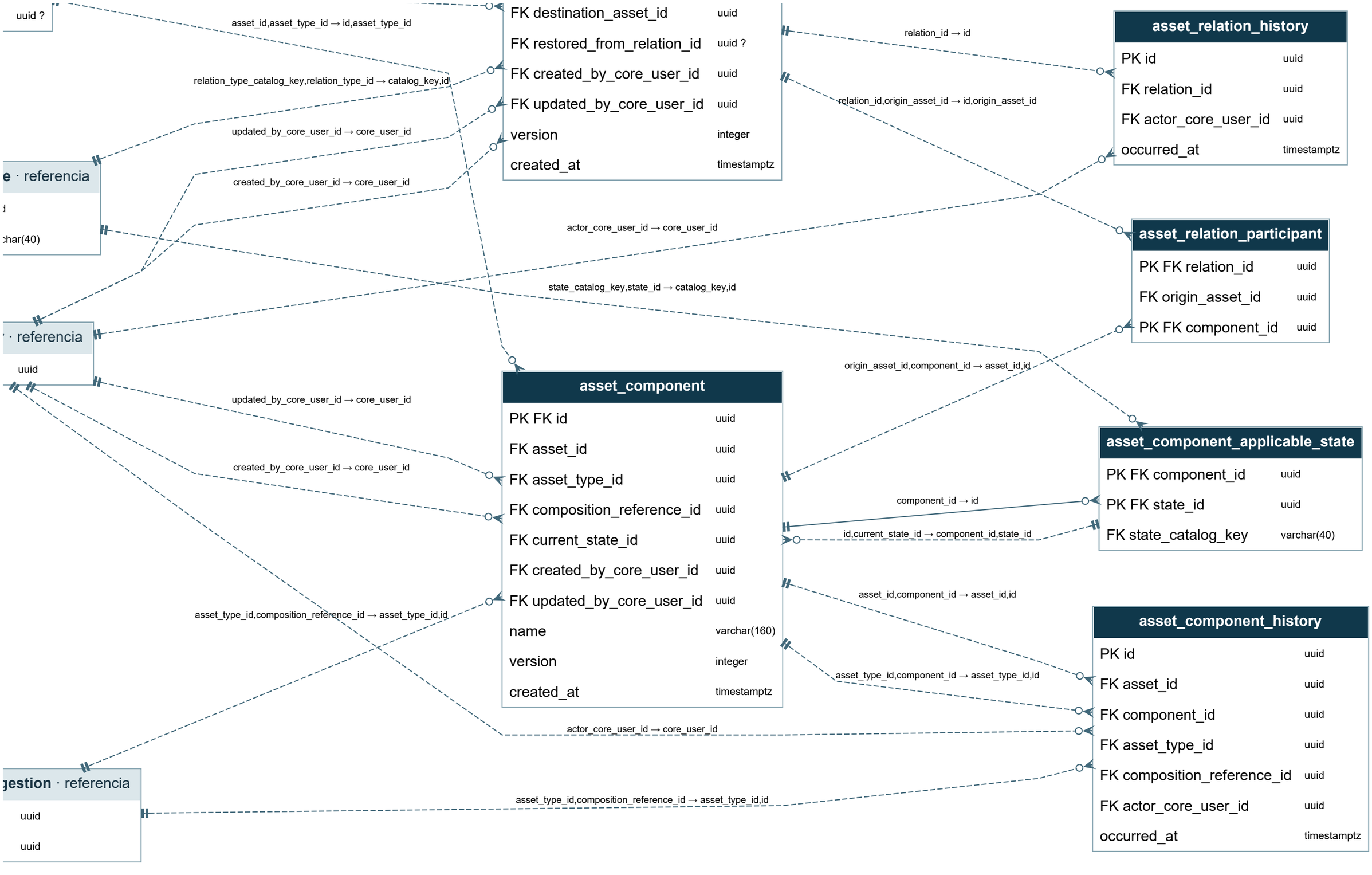
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



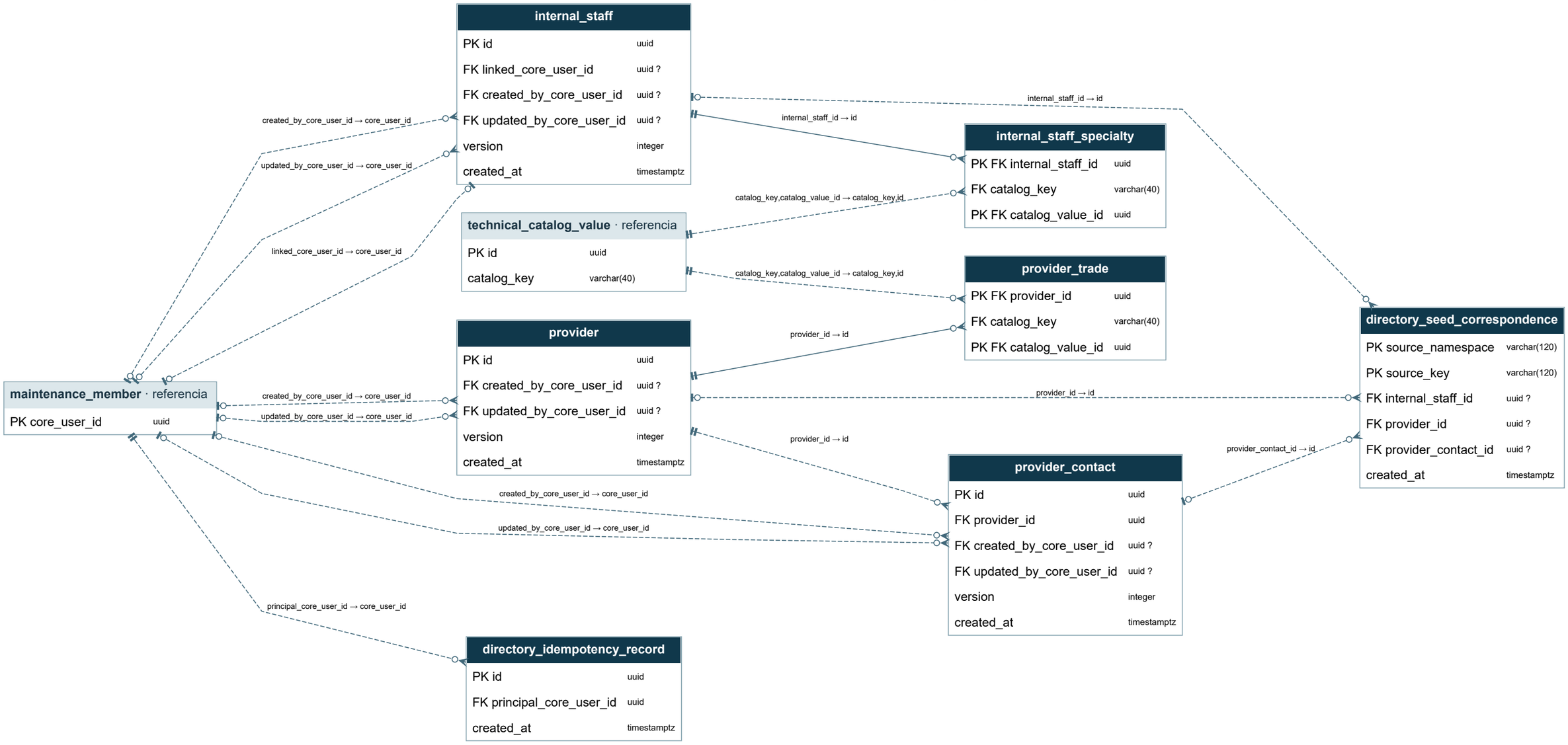

FK asset_type_id uuid ?
asset_id,asset_type_id → id,asset_type_id
relation_type_catalog_key,relation_type_id → catalog_key,id
updated_by_core_user_id → core_user_id
created_by_core_user_id → core_user_id
FK destination_asset_id uuid
FK restored_from_relation_id uuid ?
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
version integer
created_at timestamptz
relation_id → id
relation_id,origin_asset_id → id,origin_asset_id
technical_catalog_value · referencia
PK id uuid
catalog_key varchar(40)
actor_core_user_id → core_user_id
state_catalog_key,state_id → catalog_key,id
maintenance_member · referencia
PK core_user_id uuid
origin_asset_id,component_id → asset_id,id
asset_component
PK FK id uuid
FK asset_id uuid
FK asset_type_id uuid
FK composition_reference_id uuid
FK current_state_id uuid
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
name varchar(160)
version integer
created_at timestamptz
component_id → id
id,current_state_id → component_id,state_id
asset_id,component_id → asset_id,id
asset_type_id,component_id → asset_type_id,id
asset_type_id,composition_reference_id → asset_type_id,id
actor_core_user_id → core_user_id
asset_type_component_suggestion · referencia
PK FK id uuid
FK asset_type_id uuid
asset_type_id,composition_reference_id → asset_type_id,id

La Rural · modelo de datos

Mantenimiento · Componentes y relaciones · 1 · ampliación 2/2, 2/2

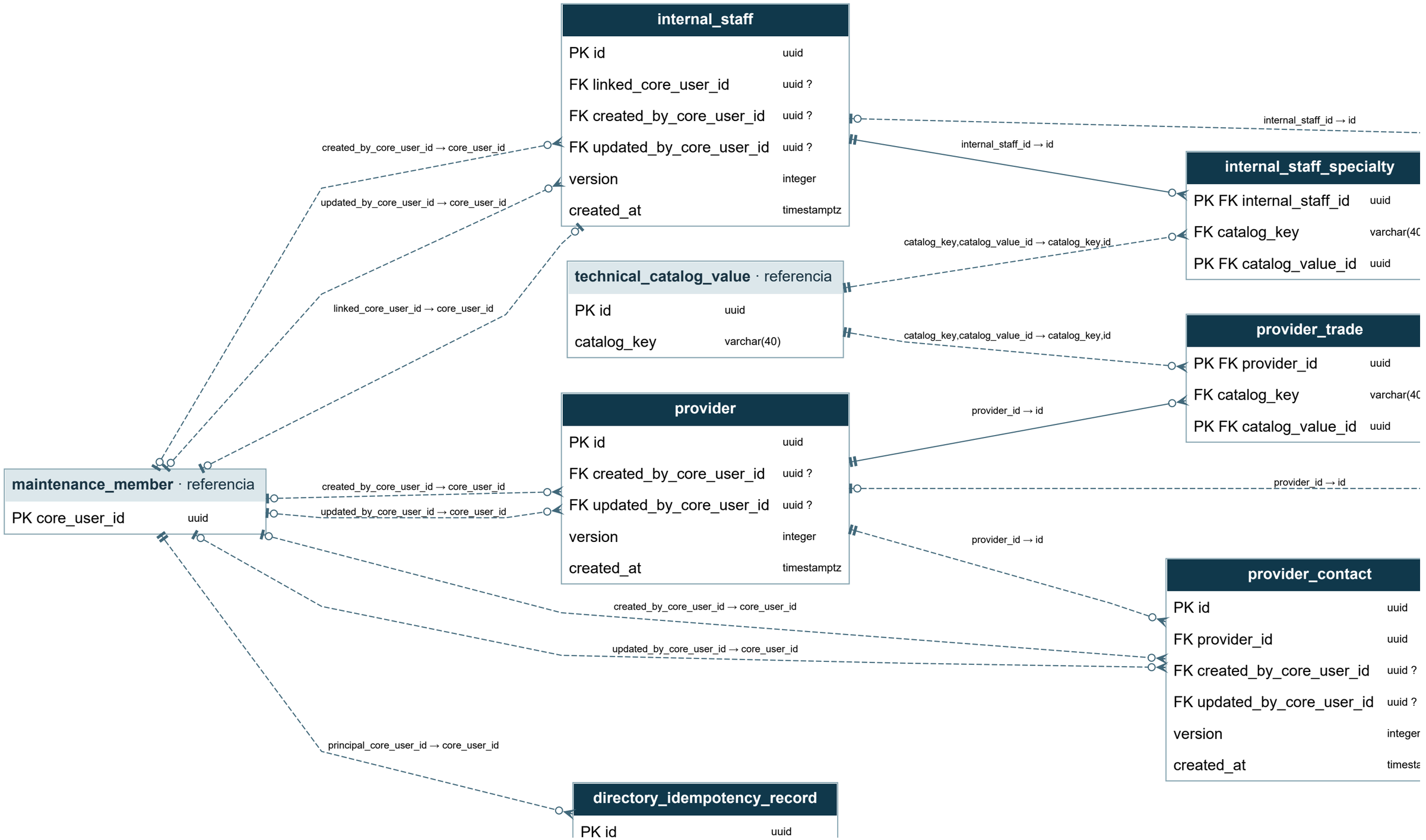


Mantenimiento · Directorio operativo · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

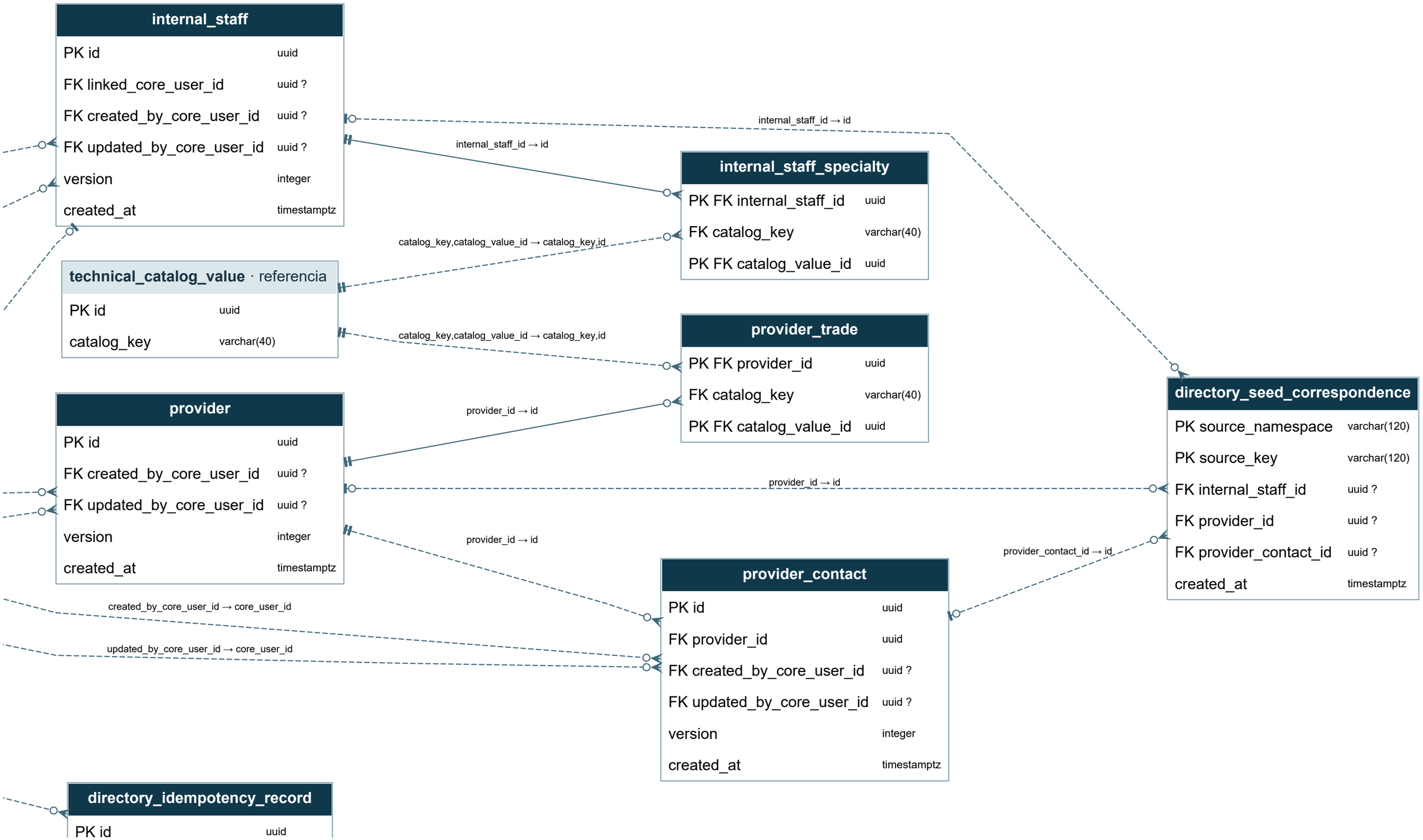


Mantenimiento · Directorio operativo · 1

La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

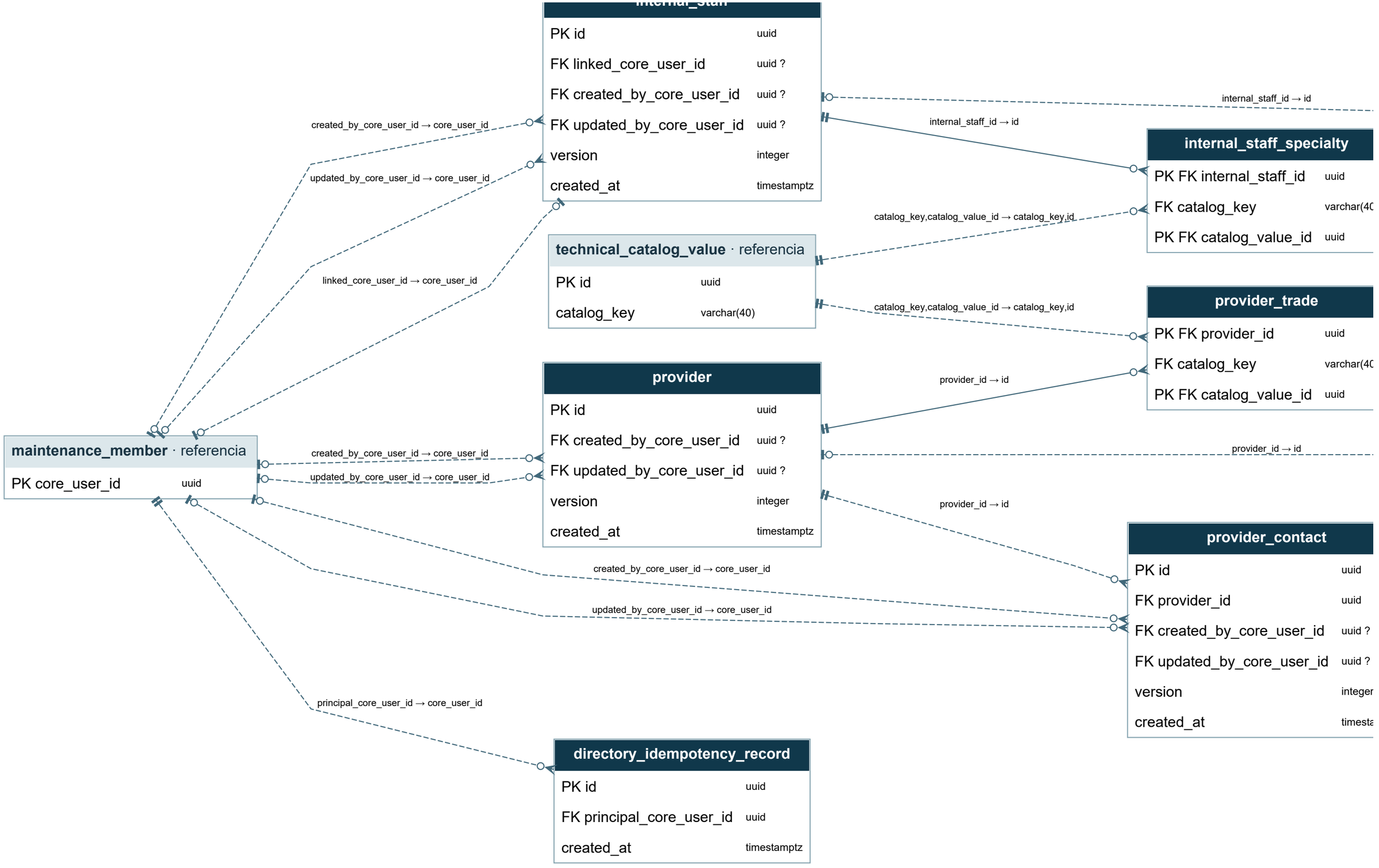


Mantenimiento · Directorio operativo · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



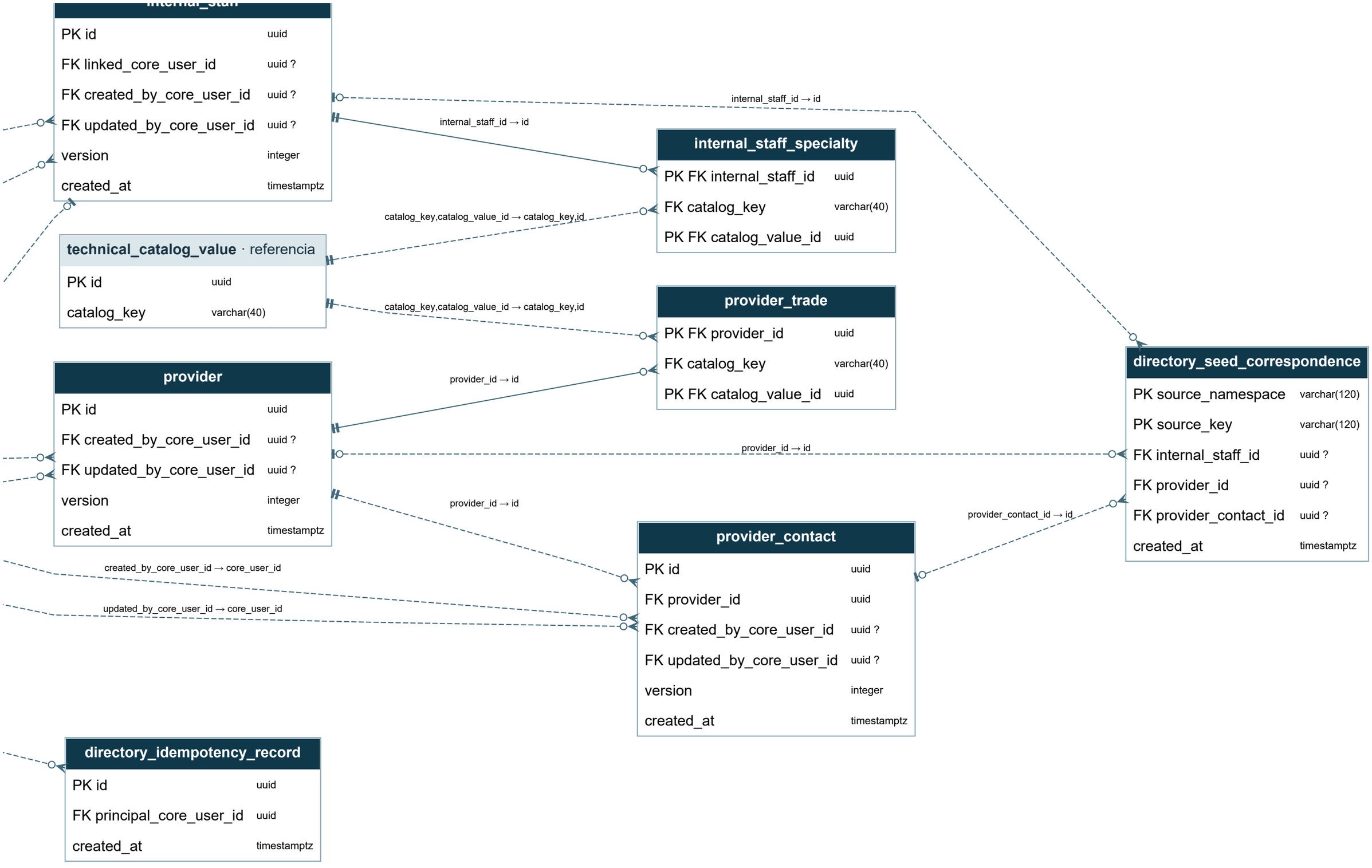
La Rural · modelo de datos

Mantenimiento · Directorio operativo · 1 · ampliación 2/2, 1/2



La Rural · modelo de datos

Mantenimiento · Directorio operativo · 1 · ampliación 2/2, 2/2

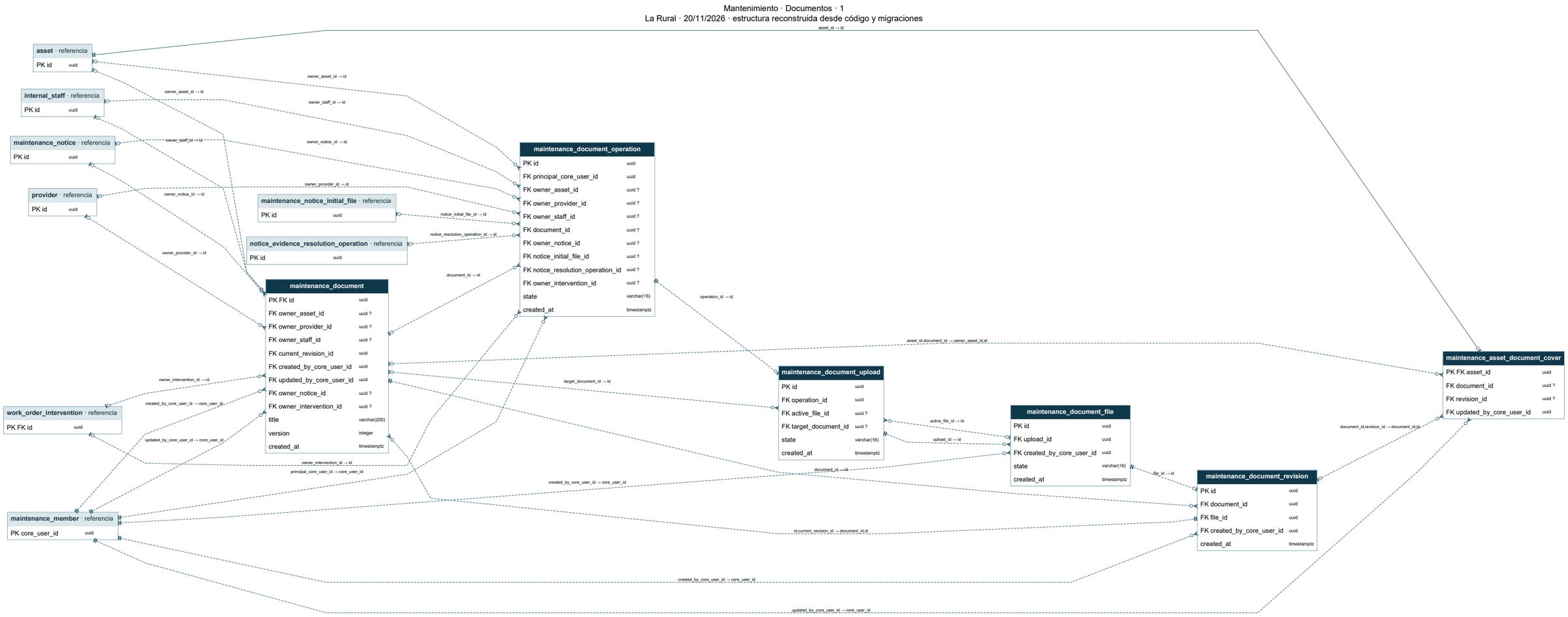


Mantenimiento · Directorio operativo · 2
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

directory_seed_application	
PK id	uuid
occurred_at	timestamptz

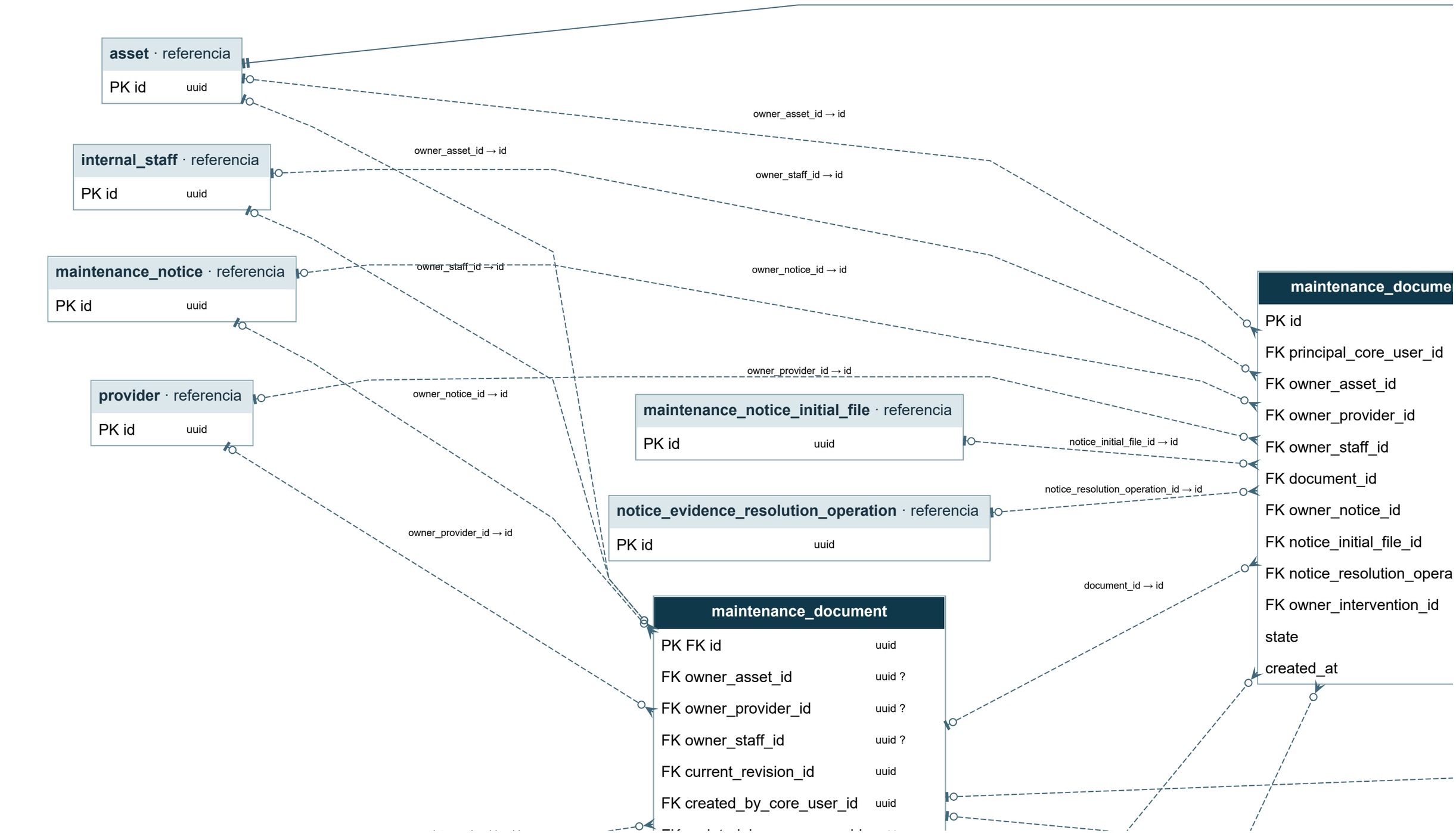
La Rural · modelo de datos

Mantenimiento · Documentos · 1



La Rural · modelo de datos

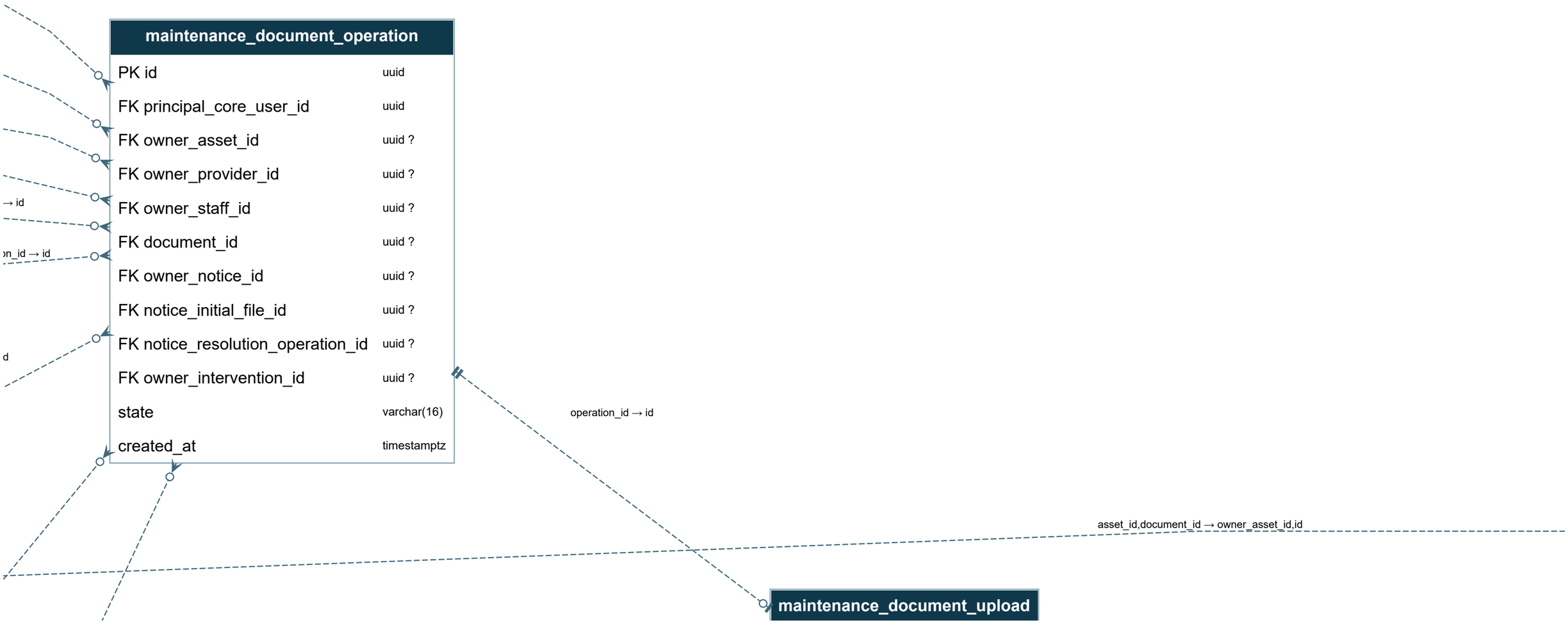
Mantenimiento · Documentos · 1 · ampliación 1/2, 1/3



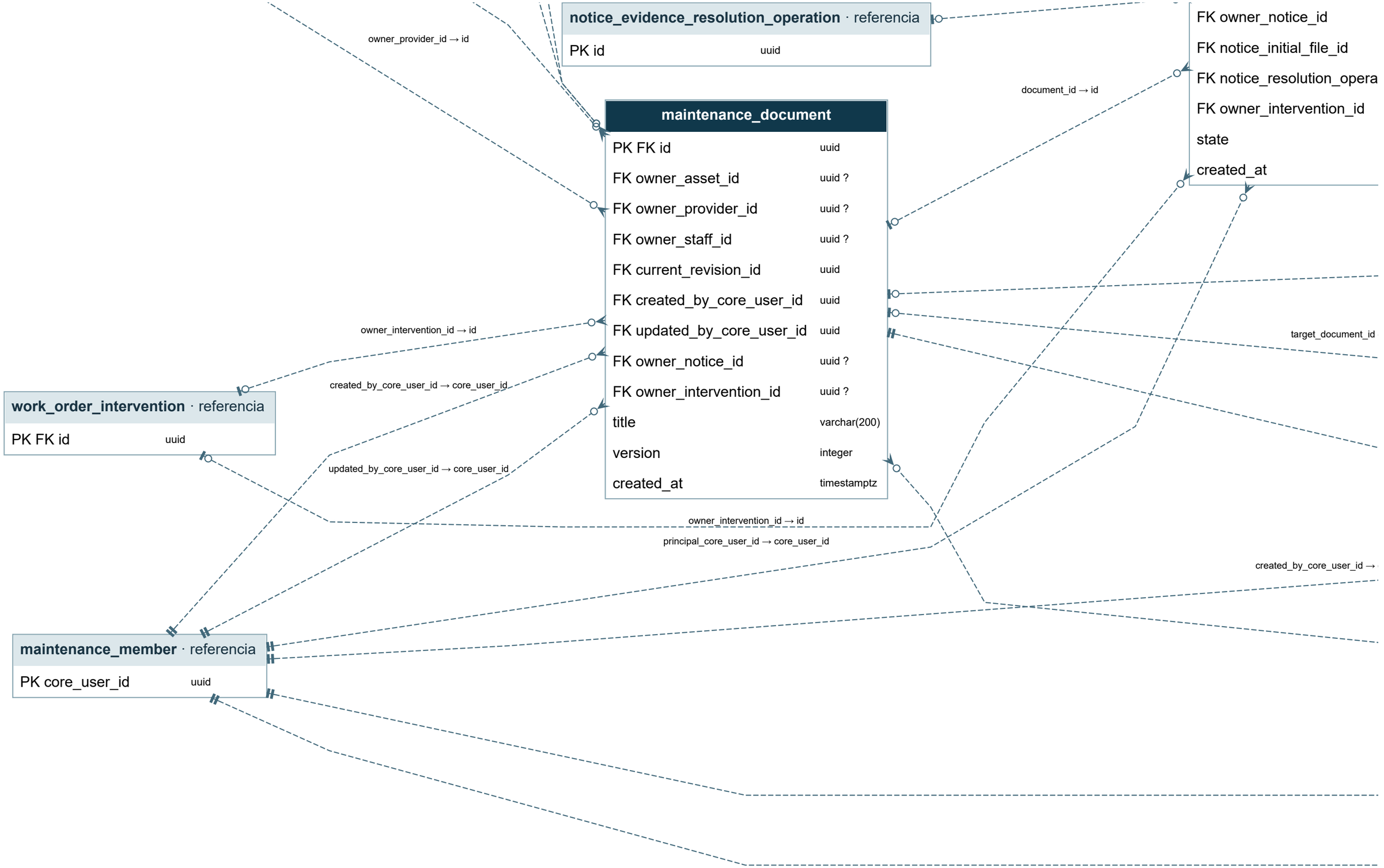
Mantenimiento · Documentos · 1

La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

asset_id → id

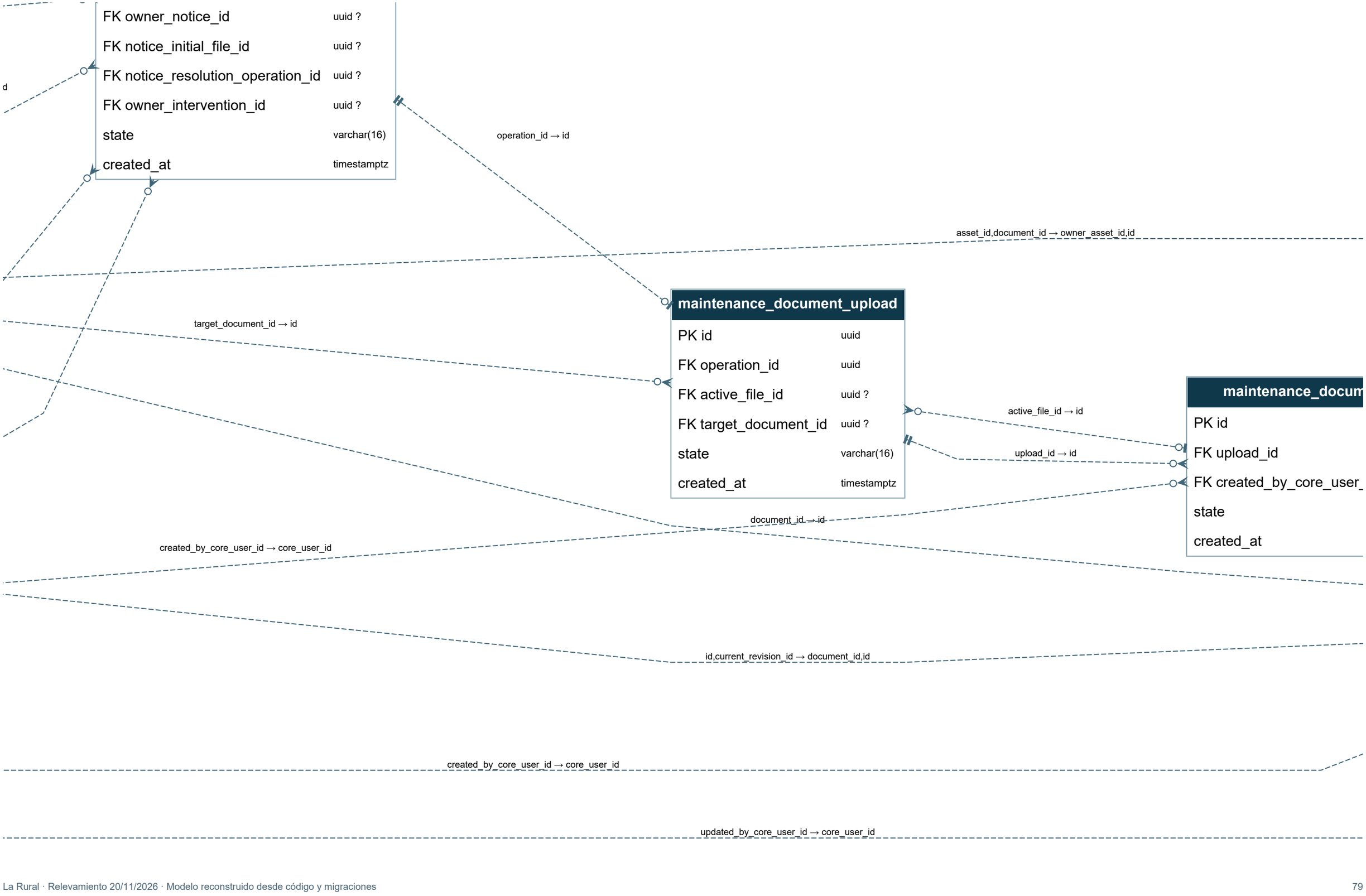


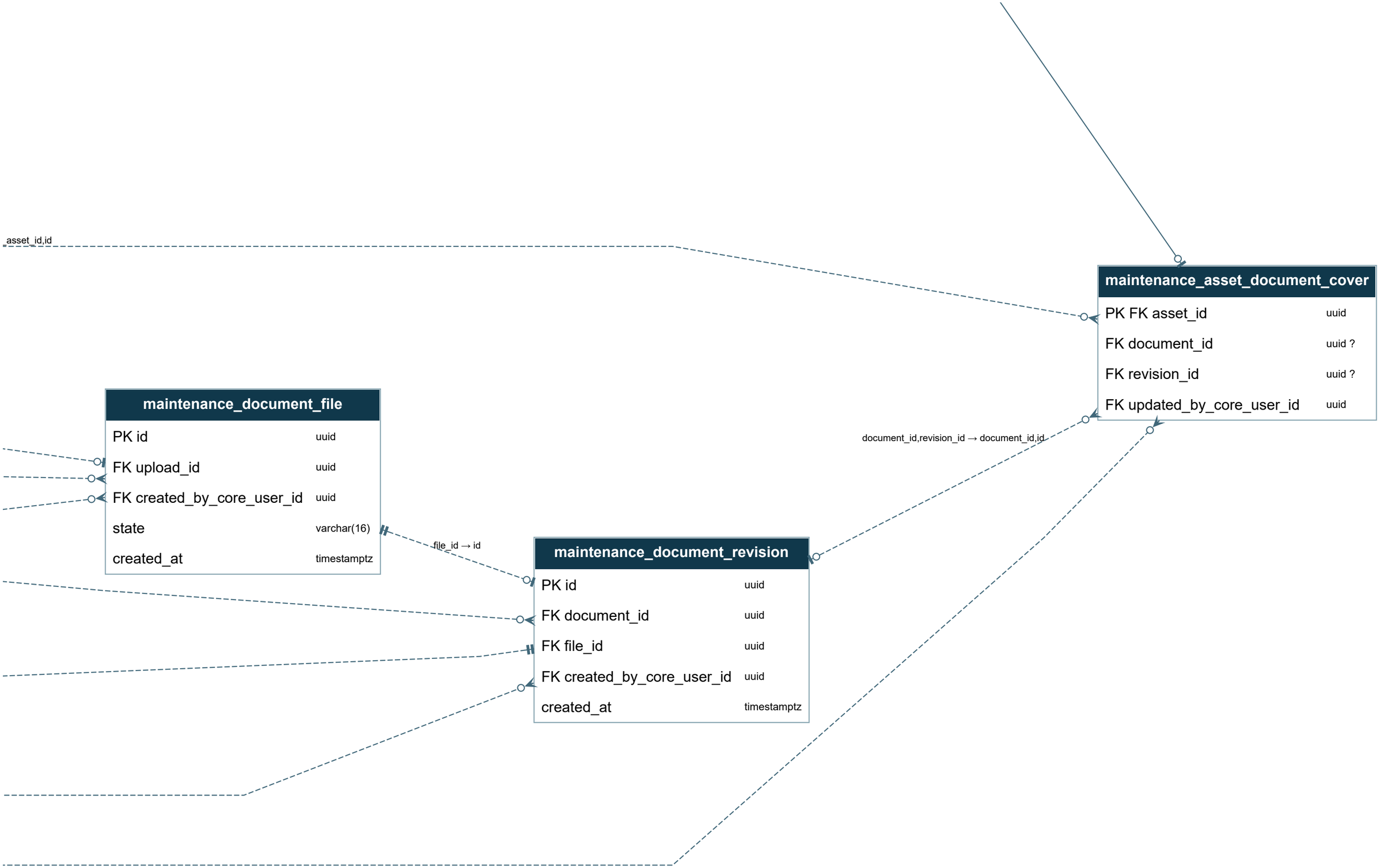

_asset_id,id
maintenance_asset_document_cover
PK FK asset_id
uuid


notice_evidence_resolution_operation · referencia
PK id uuid
owner_provider_id → id
document_id → id
FK owner_notice_id
FK notice_initial_file_id
FK notice_resolution_opera
FK owner_intervention_id
state
created_at
maintenance_document
PK FK id uuid
FK owner_asset_id uuid ?
FK owner_provider_id uuid ?
FK owner_staff_id uuid ?
FK current_revision_id uuid
FK created_by_core_user_id uuid
FK updated_by_core_user_id uuid
FK owner_notice_id uuid ?
FK owner_intervention_id uuid ?
title varchar(200)
version integer
created_at timestamptz
owner_intervention_id → id
created_by_core_user_id → core_user_id
target_document_id
work_order_intervention · referencia
PK FK id uuid
updated_by_core_user_id → core_user_id
owner_intervention_id → id
principal_core_user_id → core_user_id
created_by_core_user_id →
maintenance_member · referencia
PK core_user_id uuid

La Rural · modelo de datos

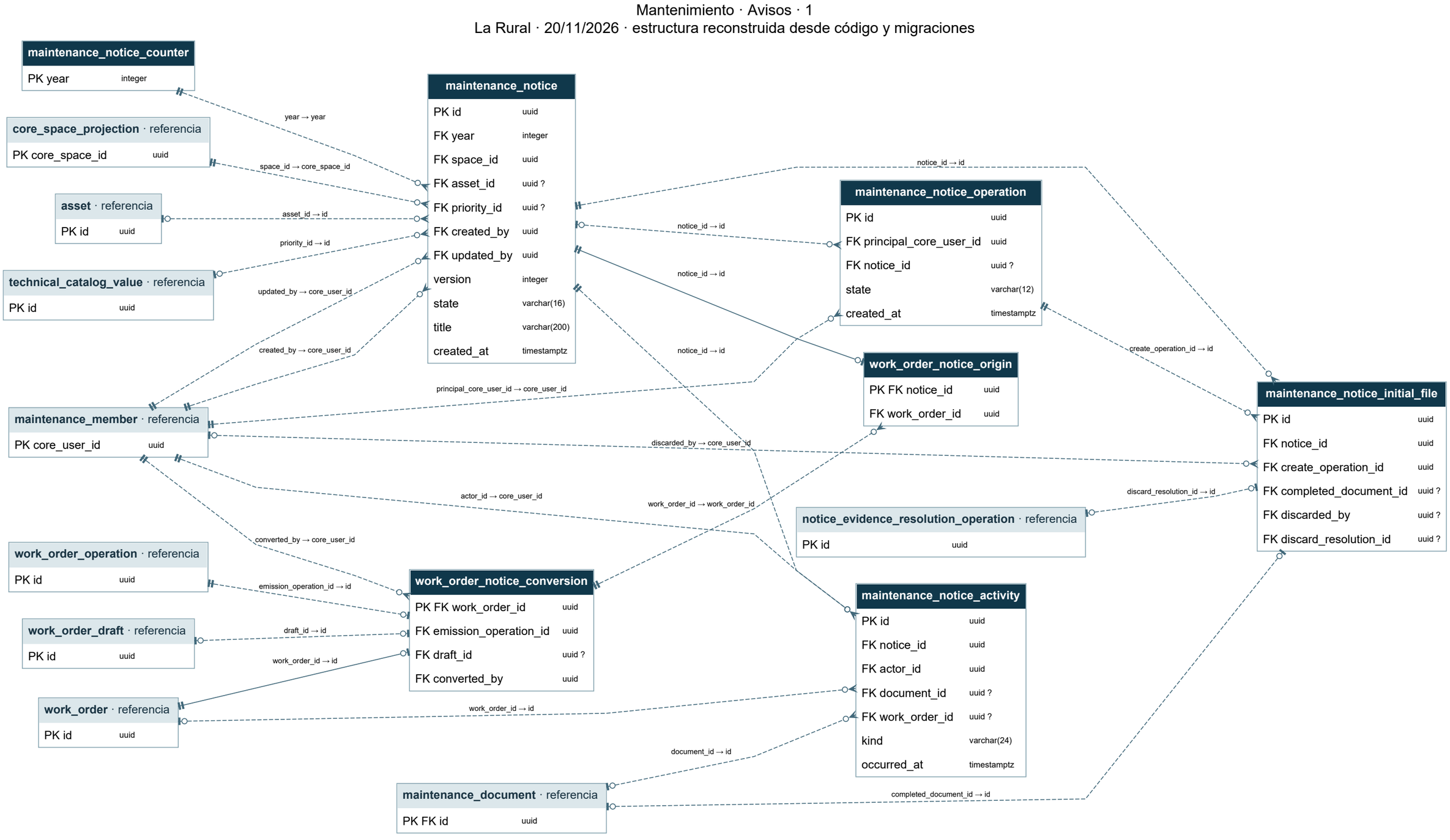
Mantenimiento · Documentos · 1 · ampliación 2/2, 2/3

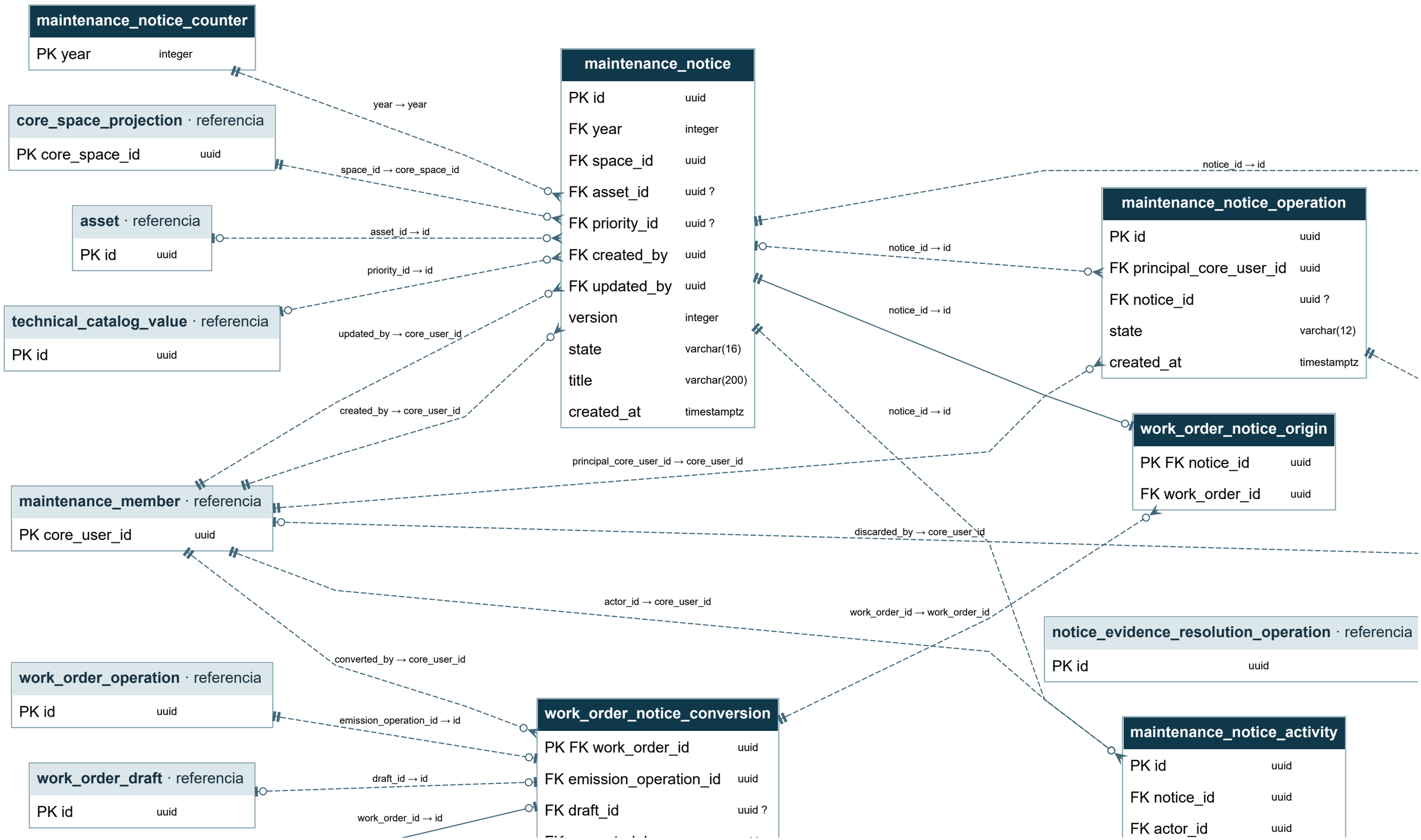



asset_id,id
maintenance_asset_document_cover
PK FK asset_id uuid
FK document_id uuid ?
FK revision_id uuid ?
FK updated_by_core_user_id uuid
maintenance_document_file
PK id uuid
FK upload_id uuid
FK created_by_core_user_id uuid
state varchar(16)
created_at timestamptz
file_id → id
maintenance_document_revision
PK id uuid
FK document_id uuid
FK file_id uuid
FK created_by_core_user_id uuid
created_at timestamptz
document_id,revision_id → document_id,id

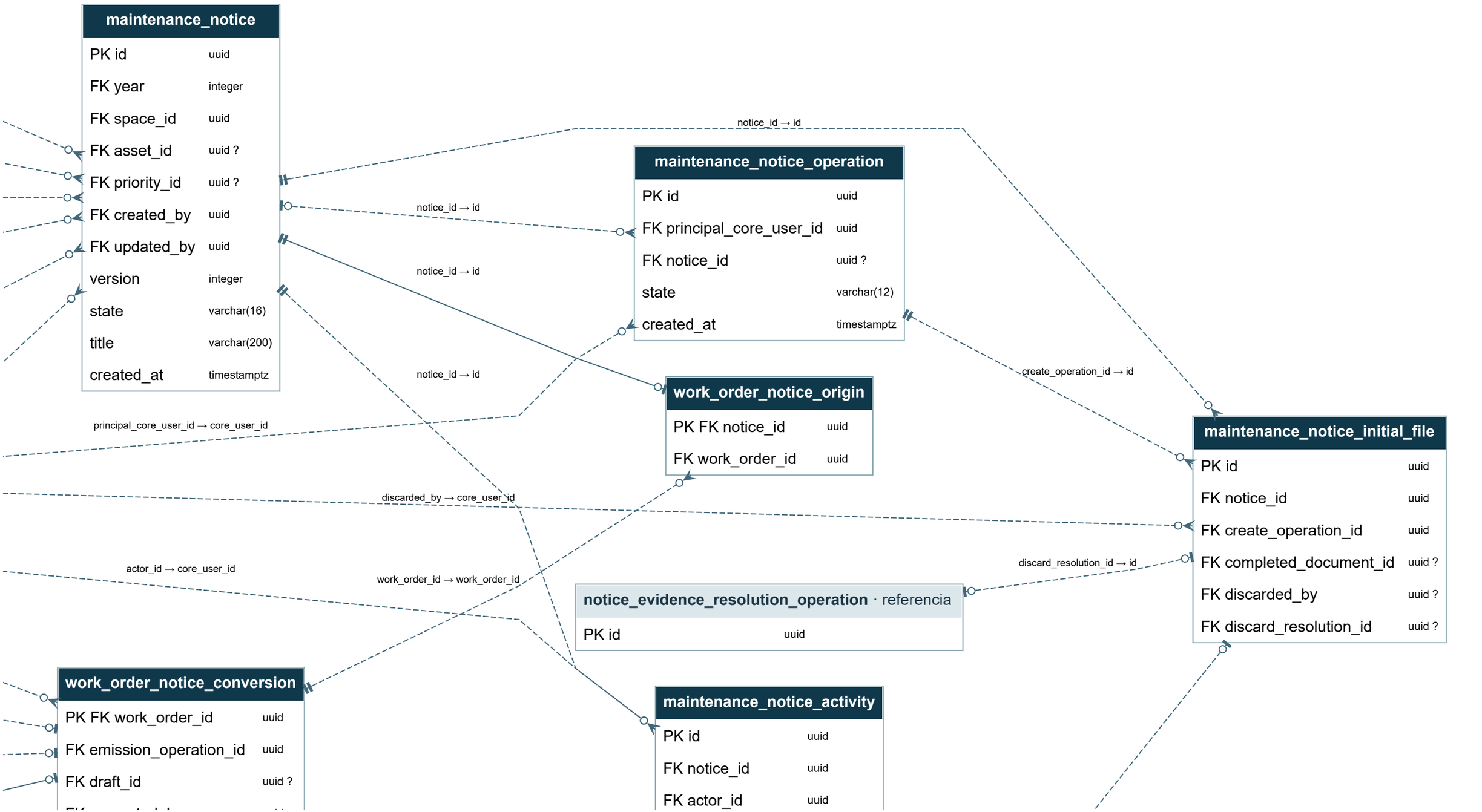
La Rural · modelo de datos

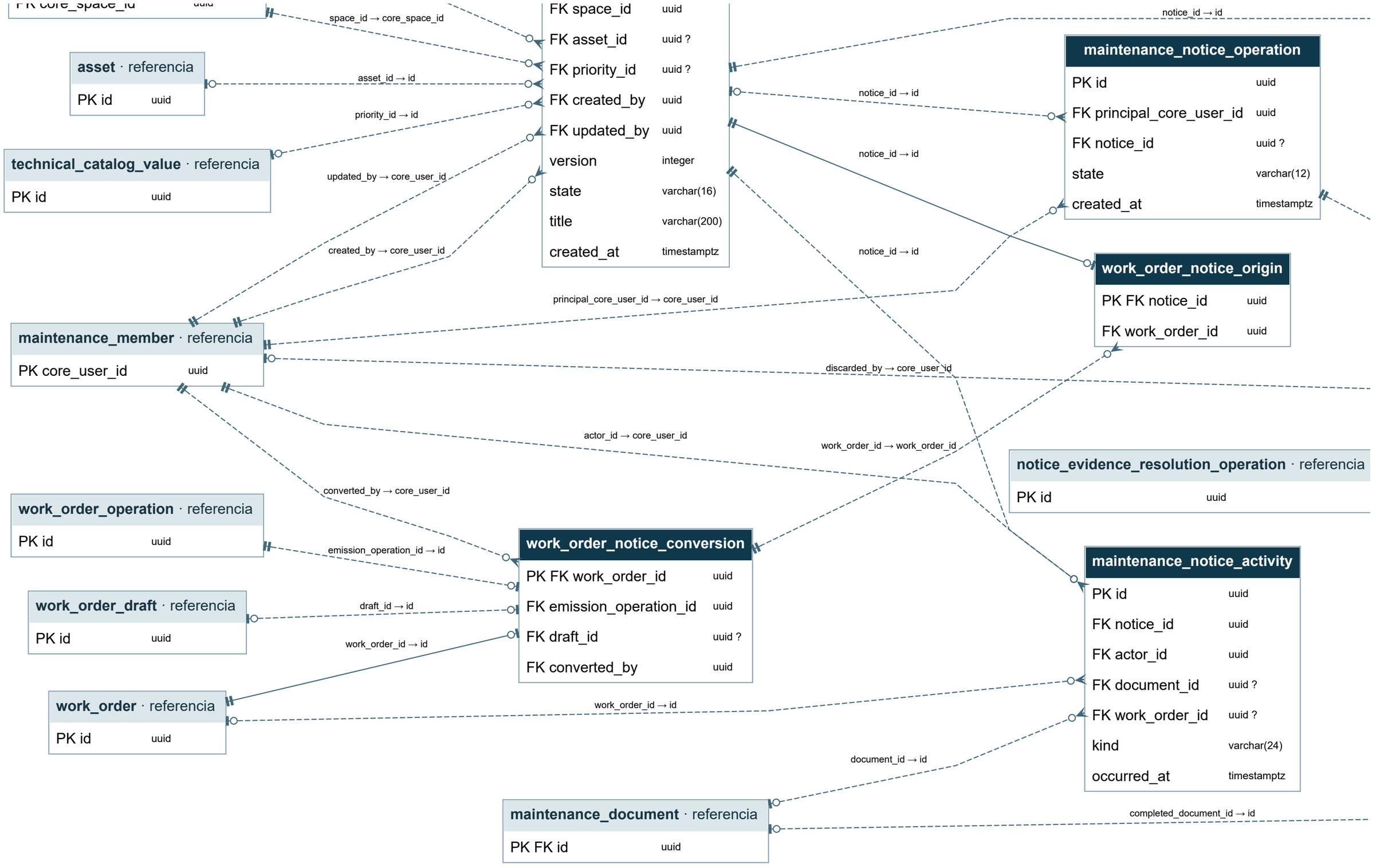
Mantenimiento · Avisos · 1




Mantenimiento · Avisos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
maintenance_notice_counter
PK year integer
core_space_projection · referencia
PK core_space_id uuid
asset · referencia
PK id uuid
technical_catalog_value · referencia
PK id uuid
maintenance_member · referencia
PK core_user_id uuid
work_order_operation · referencia
PK id uuid
work_order_draft · referencia
PK id uuid
maintenance_notice
PK id uuid
FK year integer
FK space_id uuid
FK asset_id uuid ?
FK priority_id uuid ?
FK created_by uuid
FK updated_by uuid
version integer
state varchar(16)
title varchar(200)
created_at timestamptz
maintenance_notice_operation
PK id uuid
FK principal_core_user_id uuid
FK notice_id uuid ?
state varchar(12)
created_at timestamptz
work_order_notice_origin
PK FK notice_id uuid
FK work_order_id uuid
notice_evidence_resolution_operation · referencia
PK id uuid
work_order_notice_conversion
PK FK work_order_id uuid
FK emission_operation_id uuid
FK draft_id uuid ?
maintenance_notice_activity
PK id uuid
FK notice_id uuid
FK actor_id uuid
year → year
space_id → core_space_id
asset_id → id
priority_id → id
updated_by → core_user_id
created_by → core_user_id
notice_id → id
notice_id → id
notice_id → id
principal_core_user_id → core_user_id
discarded_by → core_user_id
actor_id → core_user_id
work_order_id → work_order_id
converted_by → core_user_id
emission_operation_id → id
draft_id → id
work_order_id → id

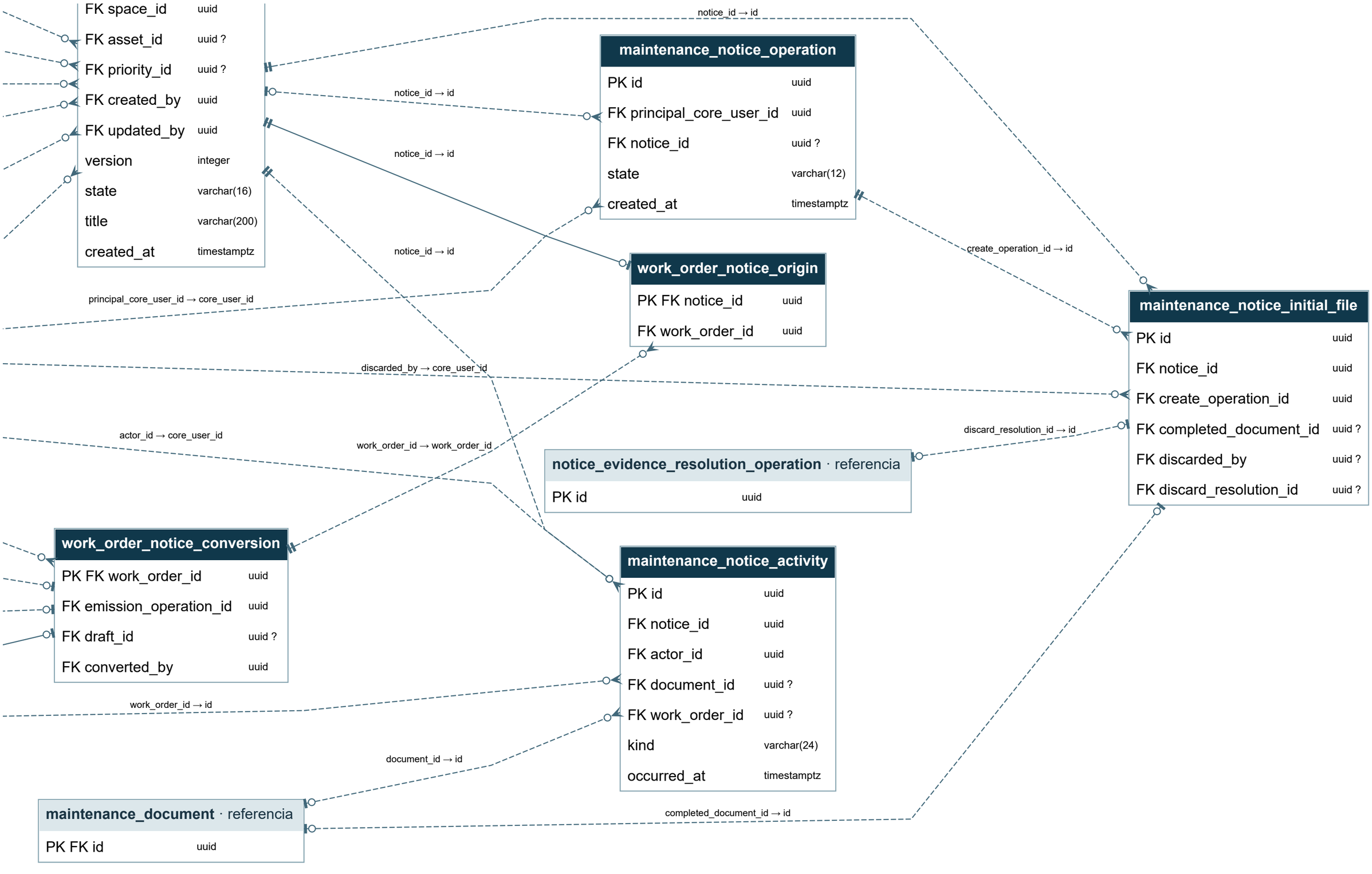
Mantenimiento · Avisos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones




PK core_space_id uuid
space_id → core_space_id
asset · referencia
PK id uuid
asset_id → id
technical_catalog_value · referencia
PK id uuid
priority_id → id
updated_by → core_user_id
created_by → core_user_id
FK space_id uuid
FK asset_id uuid ?
FK priority_id uuid ?
FK created_by uuid
FK updated_by uuid
version integer
state varchar(16)
title varchar(200)
created_at timestamptz
notice_id → id
maintenance_notice_operation
PK id uuid
FK principal_core_user_id uuid
FK notice_id uuid ?
state varchar(12)
created_at timestamptz
notice_id → id
notice_id → id
work_order_notice_origin
PK FK notice_id uuid
FK work_order_id uuid
principal_core_user_id → core_user_id
maintenance_member · referencia
PK core_user_id uuid
discarded_by → core_user_id
actor_id → core_user_id
work_order_id → work_order_id
notice_evidence_resolution_operation · referencia
PK id uuid
converted_by → core_user_id
work_order_operation · referencia
PK id uuid
emission_operation_id → id
work_order_notice_conversion
PK FK work_order_id uuid
FK emission_operation_id uuid
FK draft_id uuid ?
FK converted_by uuid
maintenance_notice_activity
PK id uuid
FK notice_id uuid
FK actor_id uuid
FK document_id uuid ?
FK work_order_id uuid ?
kind varchar(24)
occurred_at timestamptz
work_order_draft · referencia
PK id uuid
draft_id → id
work_order_id → id
work_order · referencia
PK id uuid
work_order_id → id
document_id → id
maintenance_document · referencia
PK FK id uuid
completed_document_id → id

La Rural · modelo de datos

Mantenimiento · Avisos · 1 · ampliación 2/2, 2/2



Mantenimiento · Avisos · 2

La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

maintenance_notice · referencia

PK id	uuid

maintenance_member · referencia

PK core_user_id	uuid

maintenance_notice_initial_file · referencia

PK id	uuid

maintenance_document_operation · referencia

PK id	uuid

maintenance_document · referencia

PK FK id	uuid

maintenance_document_revision · referencia

PK id	uuid

work_order_notice_origin · referencia

PK FK notice_id	uuid
FK work_order_id	uuid

notice_id → id

principal_core_user_id → core_user_id

initial_file_id → id

document_operation_id → id

active_document_operation_id → id

document_id → id

revision_id → id

work_order_id,notice_id → work_order_id,notice_id

notice_evidence_resolution_operation

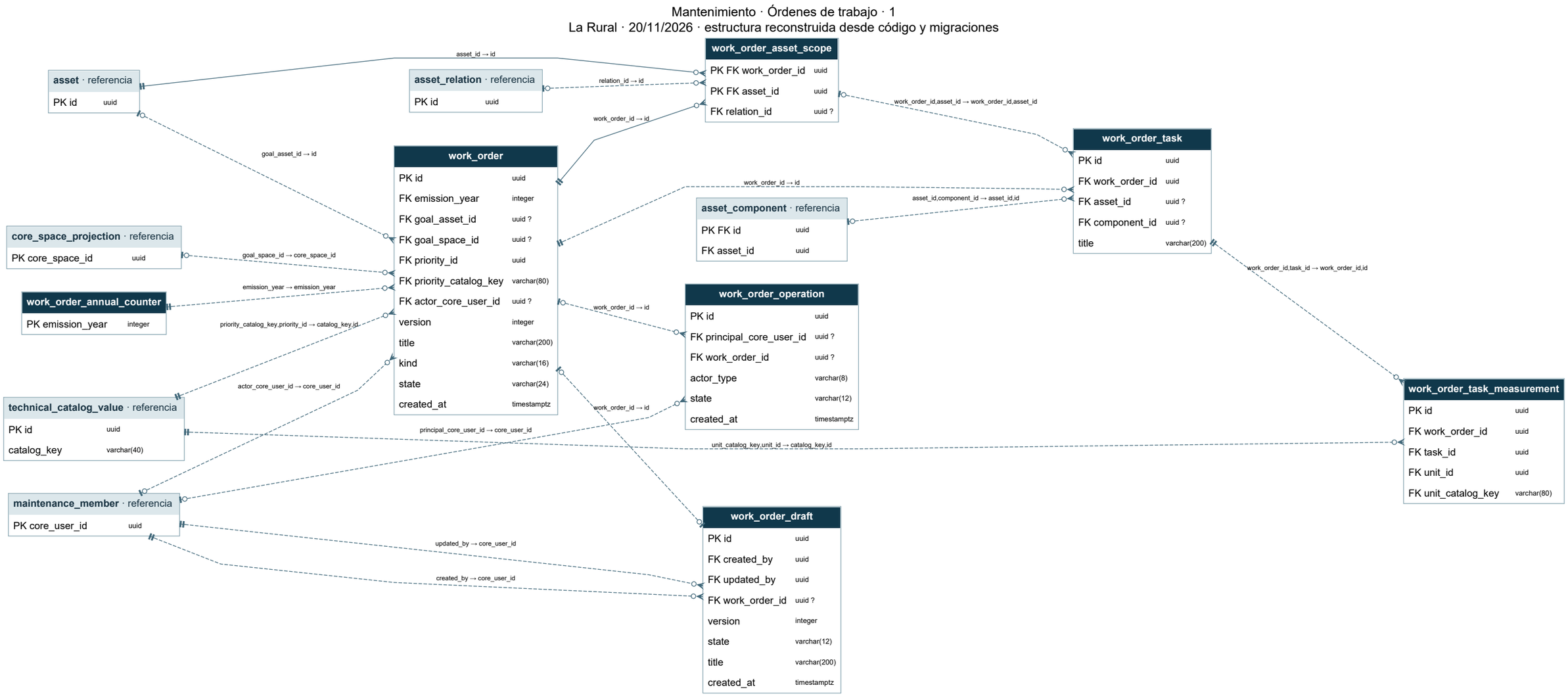
PK id	uuid
FK notice_id	uuid
FK principal_core_user_id	uuid
FK initial_file_id	uuid ?
FK document_operation_id	uuid ?
FK active_document_operation_id	uuid ?
state	varchar(12)
created_at	timestamptz

work_order_notice_evidence

PK FK work_order_id	uuid
FK notice_id	uuid
FK document_id	uuid
PK FK revision_id	uuid

La Rural · modelo de datos

Mantenimiento · Órdenes de trabajo · 1



Mantenimiento · Órdenes de trabajo · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y

asset · referencia

PK id	uuid

asset_relation · referencia

PK id	uuid

work_order_asset_scope

PK FK work_order_id	uuid
PK FK asset_id	uuid
FK relation_id	uuid ?

work_order

PK id	uuid
FK emission_year	integer
FK goal_asset_id	uuid ?
FK goal_space_id	uuid ?
FK priority_id	uuid
FK priority_catalog_key	varchar(80)
FK actor_core_user_id	uuid ?
version	integer
title	varchar(200)
kind	varchar(16)
state	varchar(24)
created_at	timestamptz

asset_component · referencia

PK FK id	uuid
FK asset_id	uuid

core_space_projection · referencia

PK core_space_id	uuid

work_order_annual_counter

PK emission_year	integer

work_order_operation

PK id	uuid
FK principal_core_user_id	uuid ?
FK work_order_id	uuid ?
actor_type	varchar(8)
state	varchar(12)
created_at	timestamptz

technical_catalog_value · referencia

PK id	uuid
catalog_key	varchar(40)

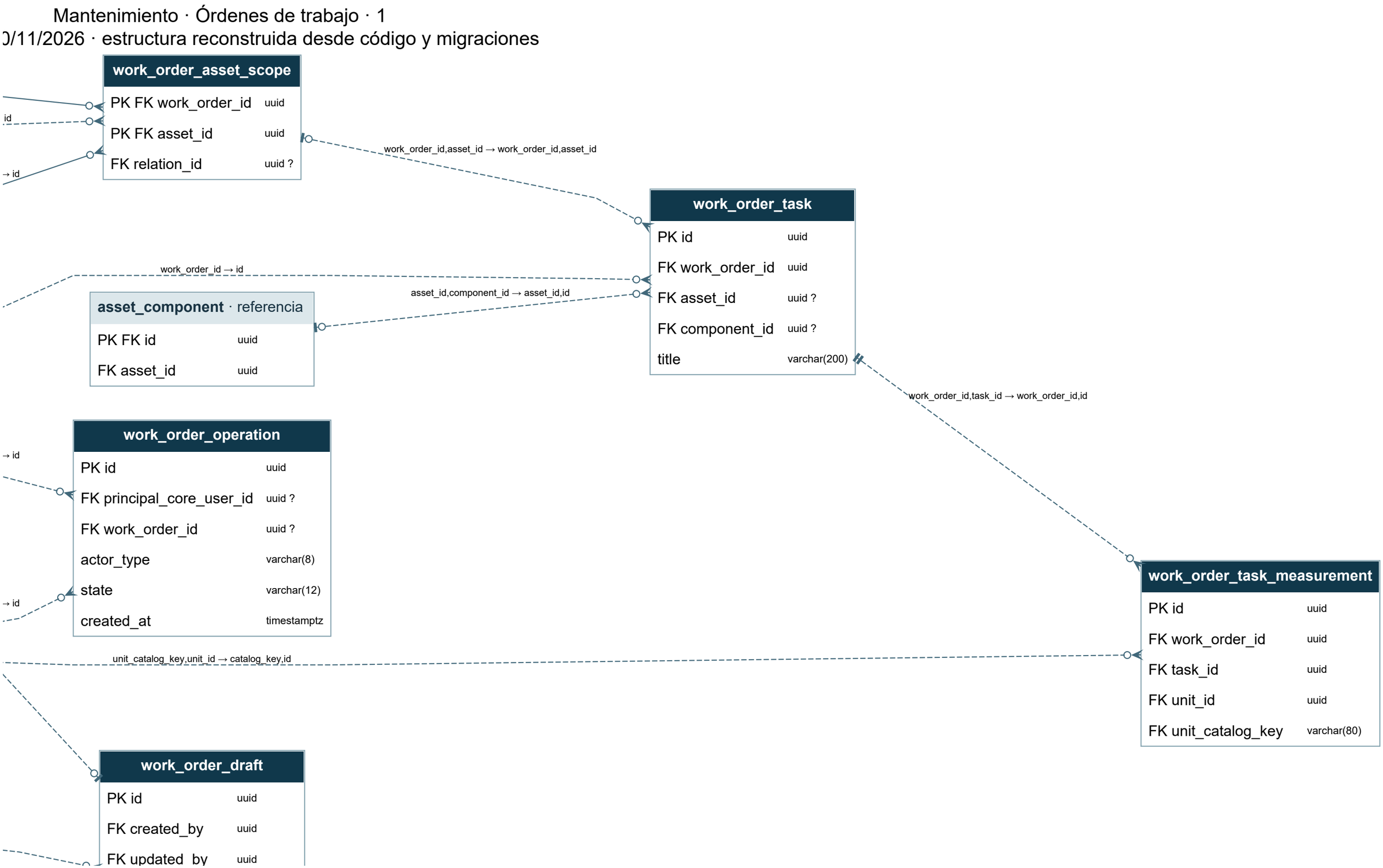
maintenance_member · referencia

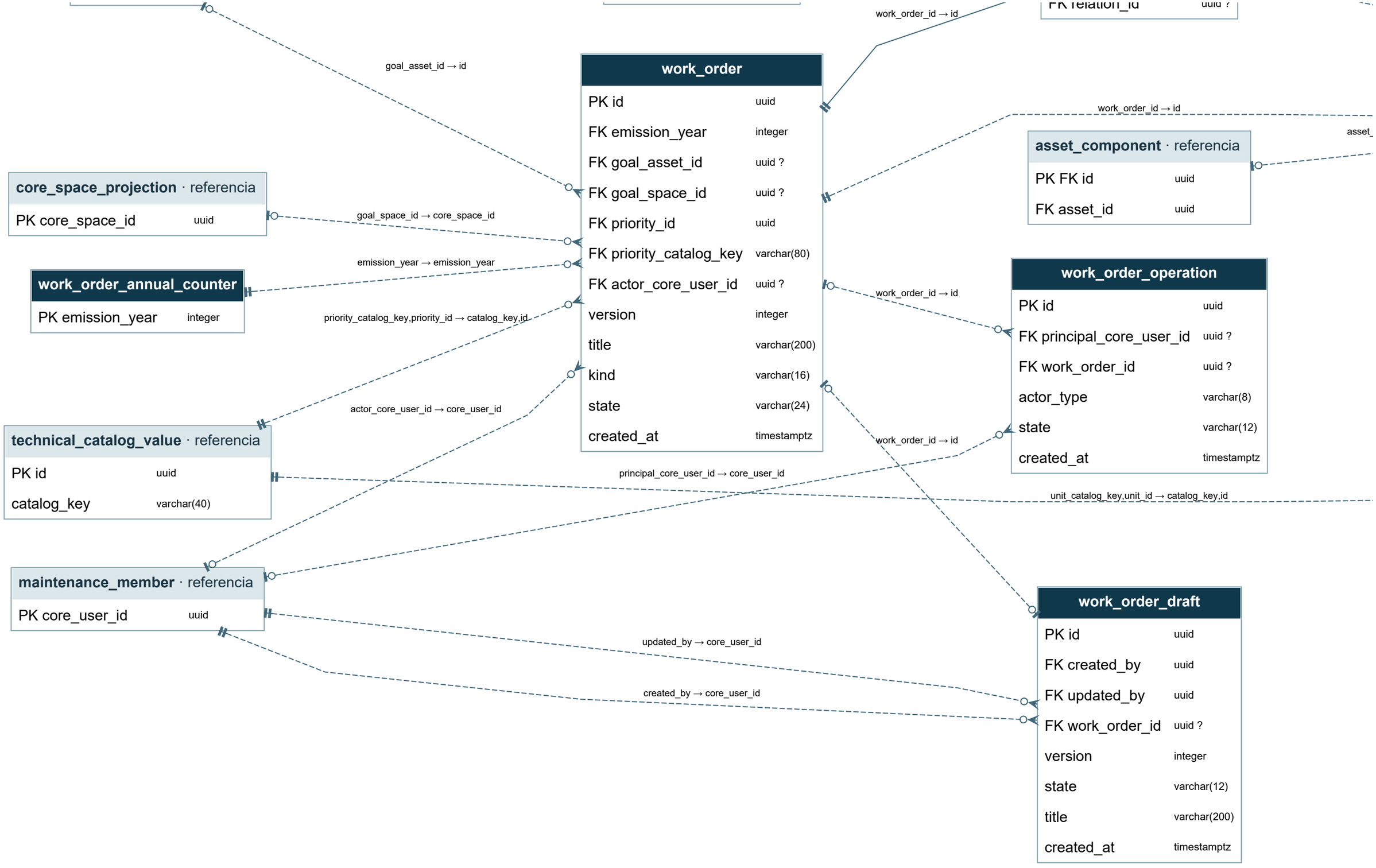
PK core_user_id	uuid

work_order_draft

PK id	uuid
FK created_by	uuid
FK updated_by	uuid

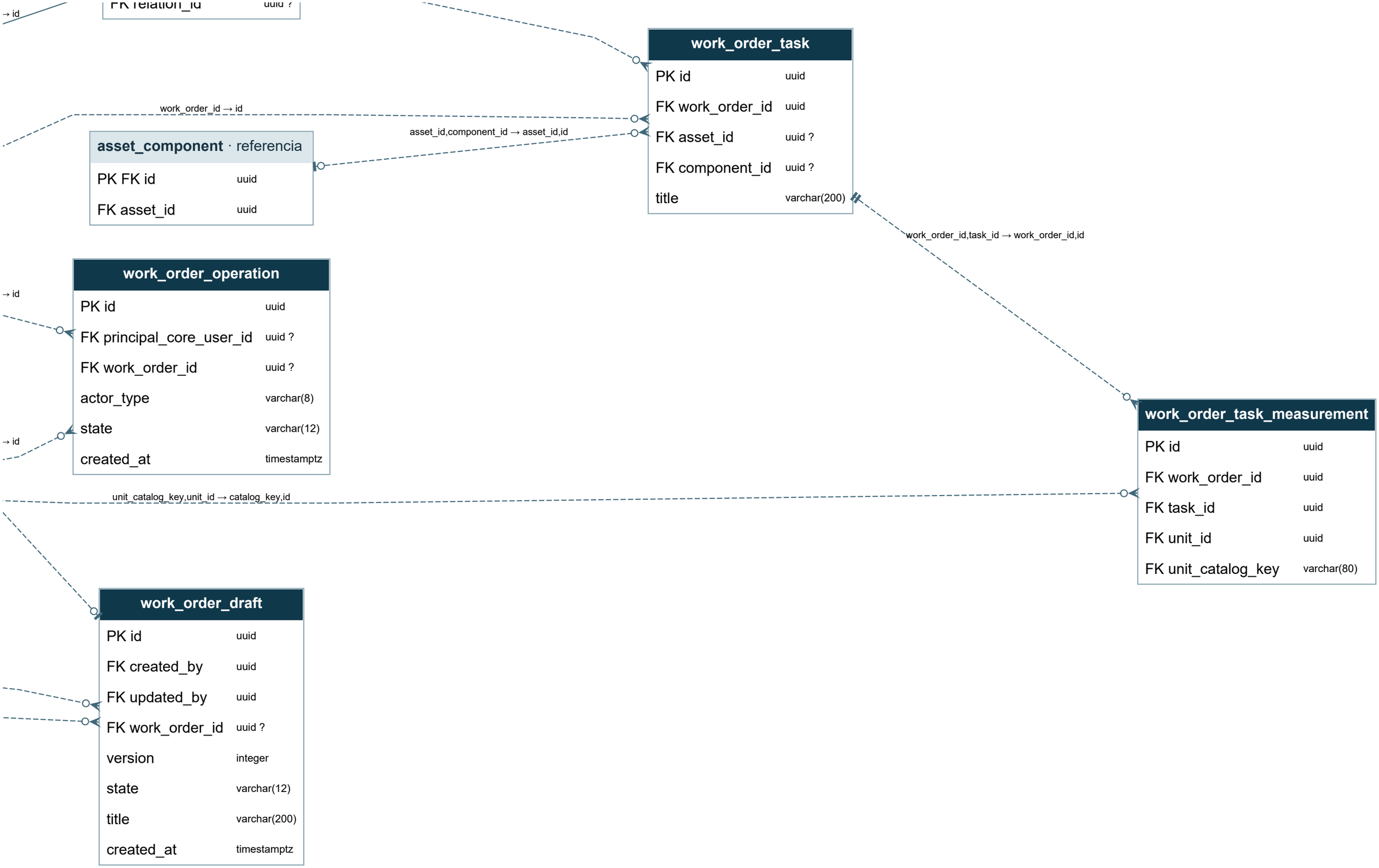
asset_id → id
relation_id → id
work_order_
work_order_id → id
goal_asset_id → id
work_order_id → id
asset_
goal_space_id → core_space_id
emission_year → emission_year
work_order_id → id
priority_catalog_key,priority_id → catalog_key,id
actor_core_user_id → core_user_id
work_order_id → id
principal_core_user_id → core_user_id
unit_catalog_key,unit_id → catalog_key,id
updated_by → core_user_id
created_by → core_user_id


Mantenimiento · Órdenes de trabajo · 1
0/11/2026 · estructura reconstruida desde código y migraciones
work_order_asset_scope
PK FK work_order_id uuid
PK FK asset_id uuid
FK relation_id uuid ?
id
→ id
work_order_id,asset_id → work_order_id,asset_id
work_order_task
PK id uuid
FK work_order_id uuid
FK asset_id uuid ?
FK component_id uuid ?
title varchar(200)
work_order_id → id
asset_id,component_id → asset_id,id
asset_component · referencia
PK FK id uuid
FK asset_id uuid
work_order_id,task_id → work_order_id,id
work_order_operation
PK id uuid
FK principal_core_user_id uuid ?
FK work_order_id uuid ?
actor_type varchar(8)
state varchar(12)
created_at timestamptz
→ id
→ id
work_order_task_measurement
PK id uuid
FK work_order_id uuid
FK task_id uuid
FK unit_id uuid
FK unit_catalog_key varchar(80)
unit_catalog_key,unit_id → catalog_key,id
work_order_draft
PK id uuid
FK created_by uuid
FK updated_by uuid


goal_asset_id → id
work_order_id → id
work_order_id → id
goal_space_id → core_space_id
emission_year → emission_year
priority_catalog_key,priority_id → catalog_key,id
actor_core_user_id → core_user_id
work_order_id → id
principal_core_user_id → core_user_id
unit_catalog_key,unit_id → catalog_key,id
updated_by → core_user_id
created_by → core_user_id
work_order
PK id uuid
FK emission_year integer
FK goal_asset_id uuid ?
FK goal_space_id uuid ?
FK priority_id uuid
FK priority_catalog_key varchar(80)
FK actor_core_user_id uuid ?
version integer
title varchar(200)
kind varchar(16)
state varchar(24)
created_at timestamptz
asset_component · referencia
PK FK id uuid
FK asset_id uuid
core_space_projection · referencia
PK core_space_id uuid
work_order_annual_counter
PK emission_year integer
work_order_operation
PK id uuid
FK principal_core_user_id uuid ?
FK work_order_id uuid ?
actor_type varchar(8)
state varchar(12)
created_at timestamptz
technical_catalog_value · referencia
PK id uuid
catalog_key varchar(40)
maintenance_member · referencia
PK core_user_id uuid
work_order_draft
PK id uuid
FK created_by uuid
FK updated_by uuid
FK work_order_id uuid ?
version integer
state varchar(12)
title varchar(200)
created_at timestamptz

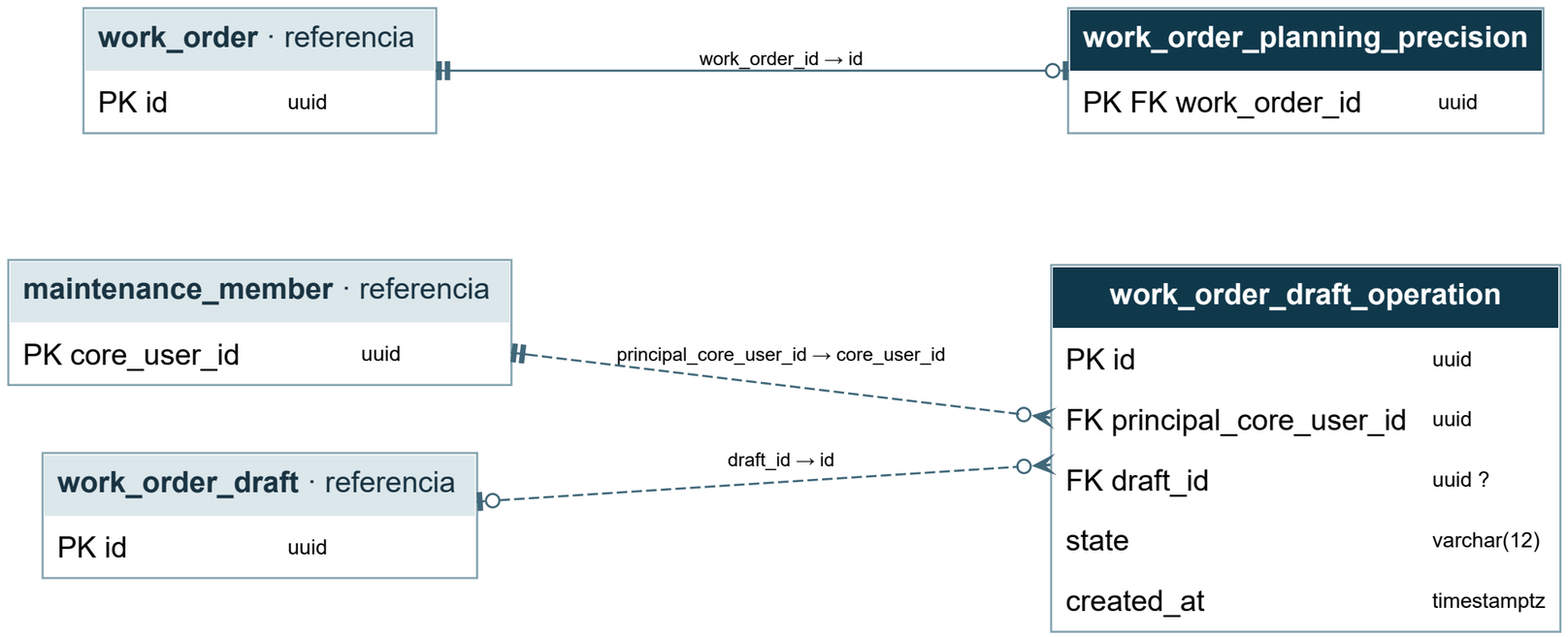
La Rural · modelo de datos

Mantenimiento · Órdenes de trabajo · 1 · ampliación 2/2, 2/2

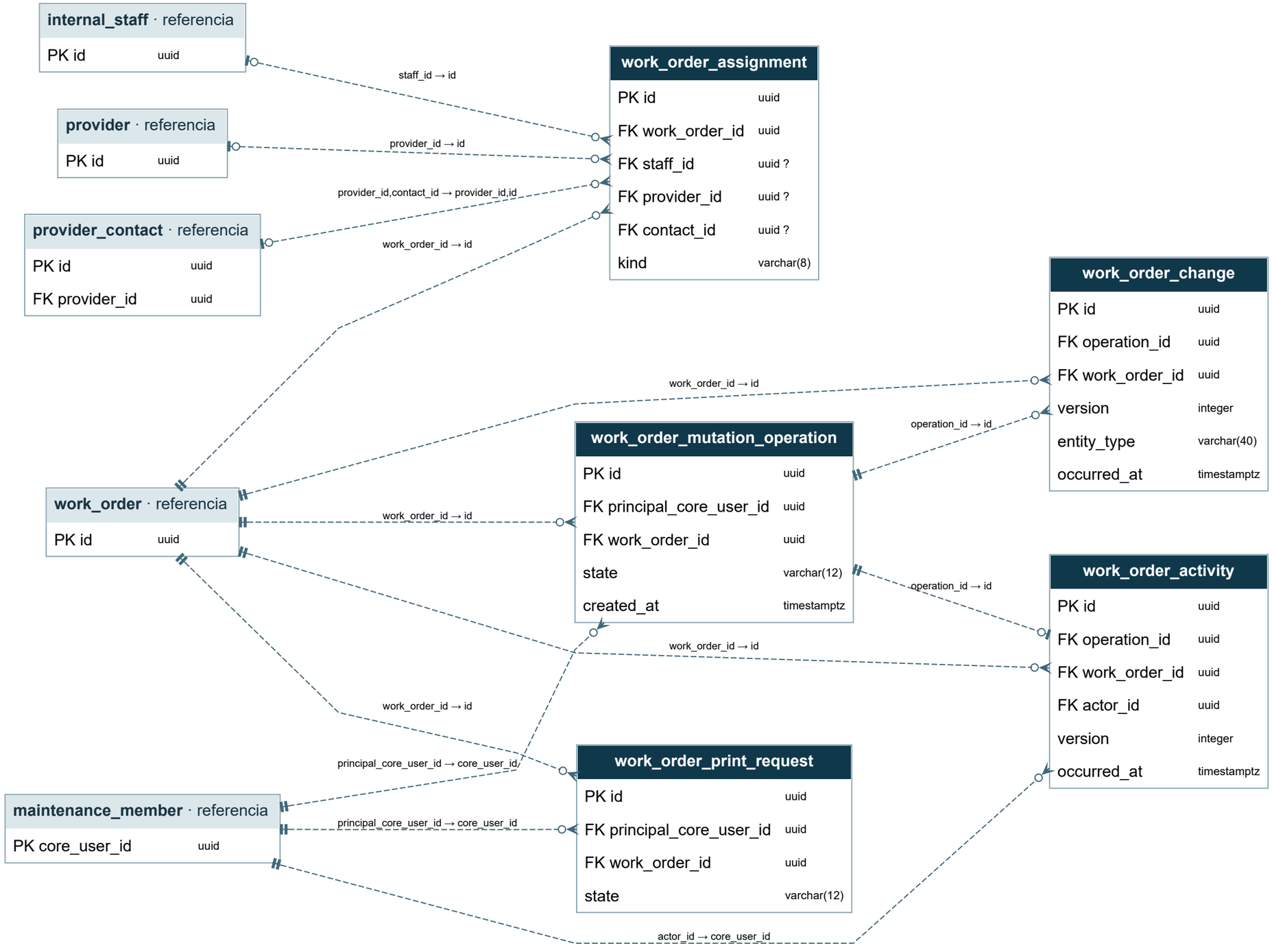


Mantenimiento · Órdenes de trabajo · 2

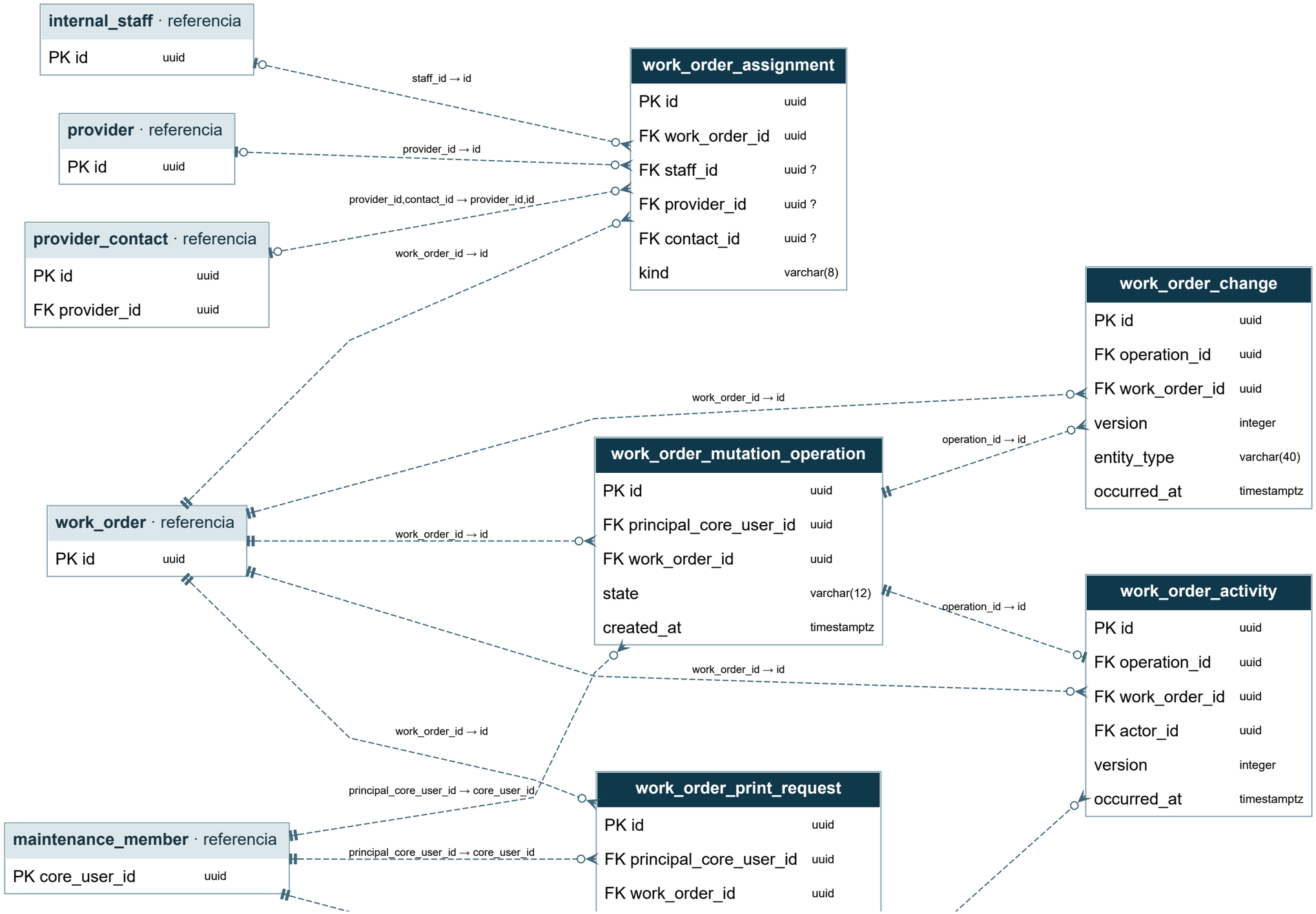
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Mantenimiento · Operación de órdenes de trabajo · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

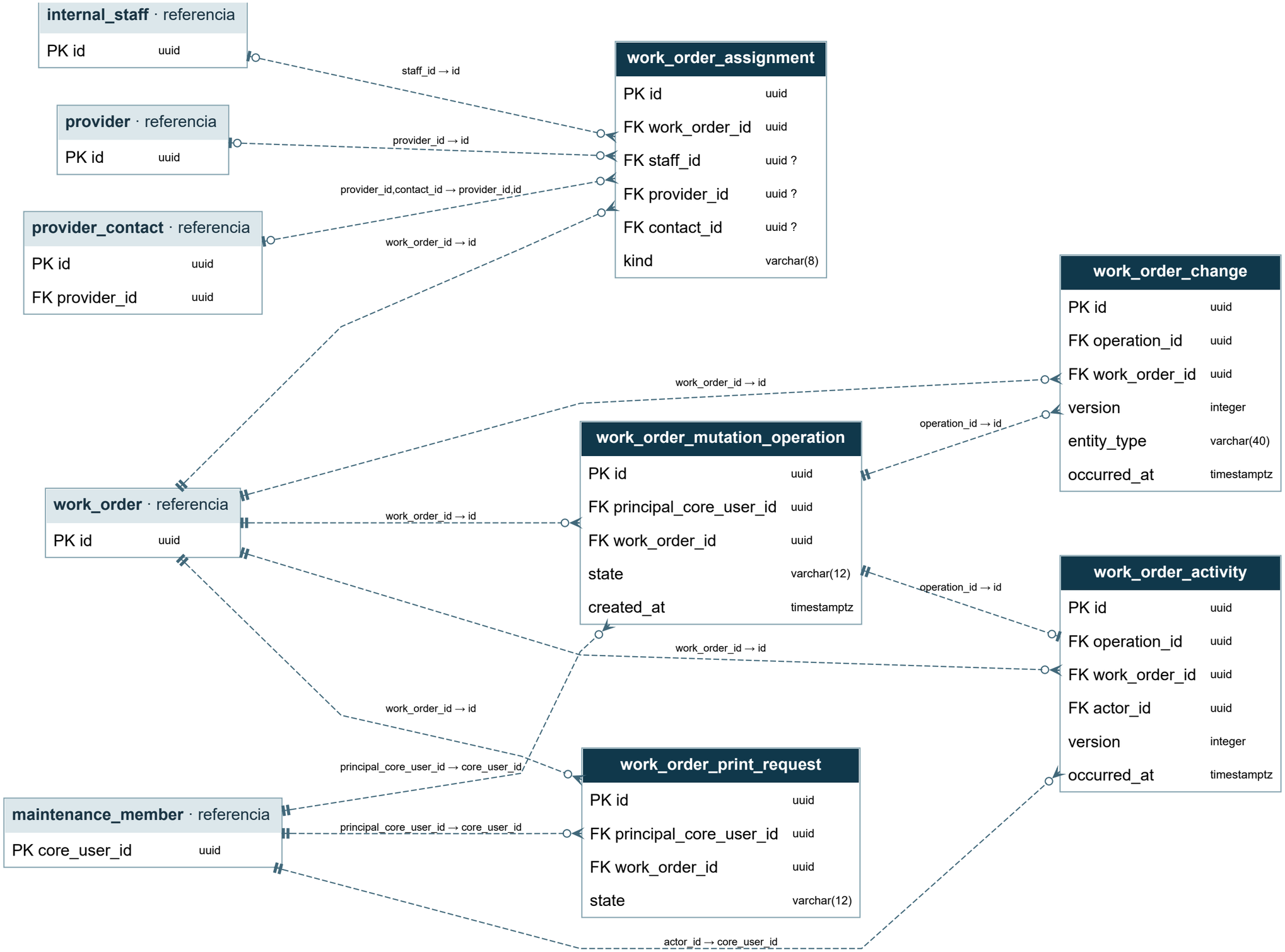


Mantenimiento · Operación de órdenes de trabajo · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



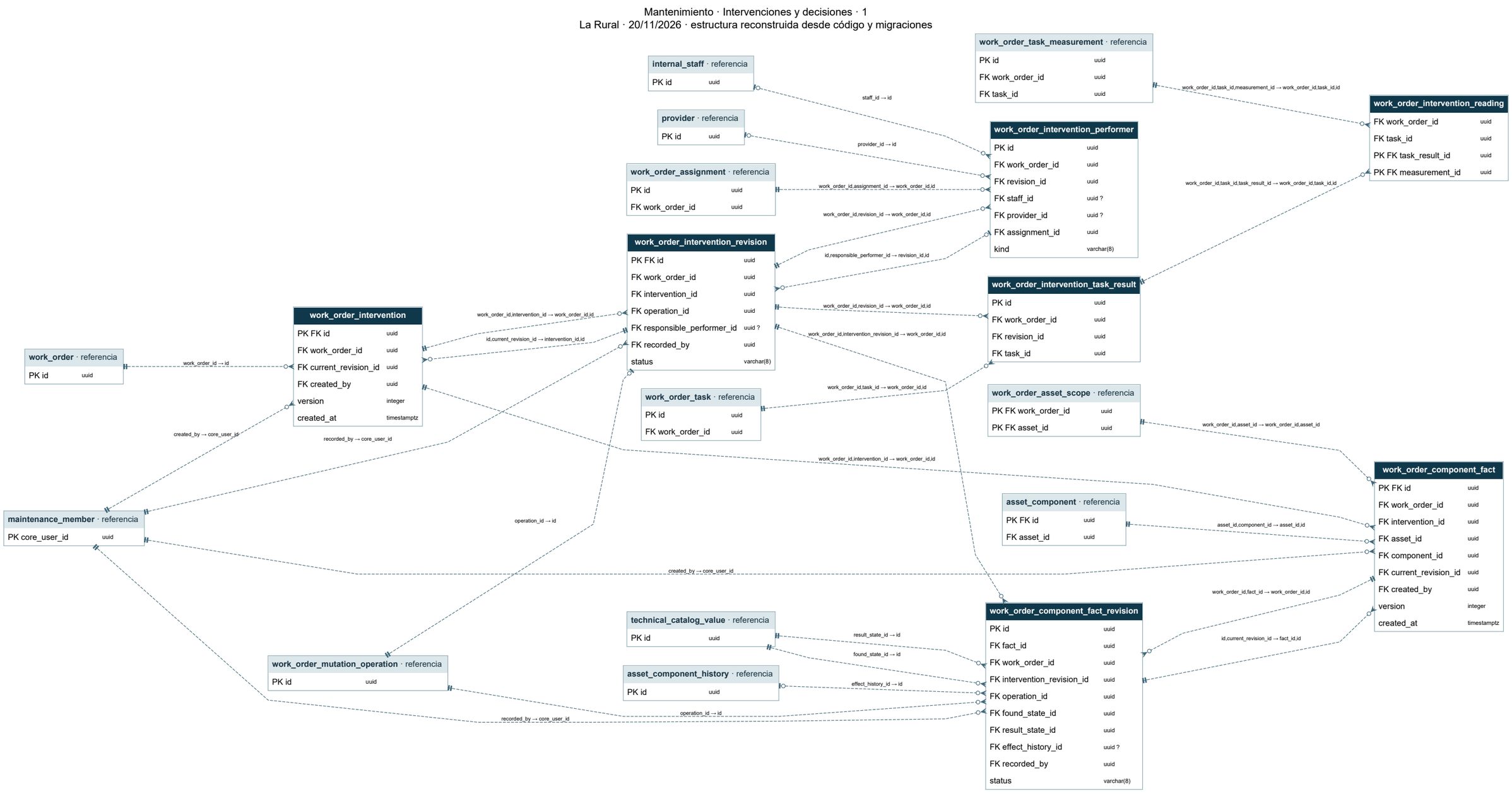
La Rural · modelo de datos

Mantenimiento · Operación de órdenes de trabajo · 1 · ampliación 2/2, 1/1

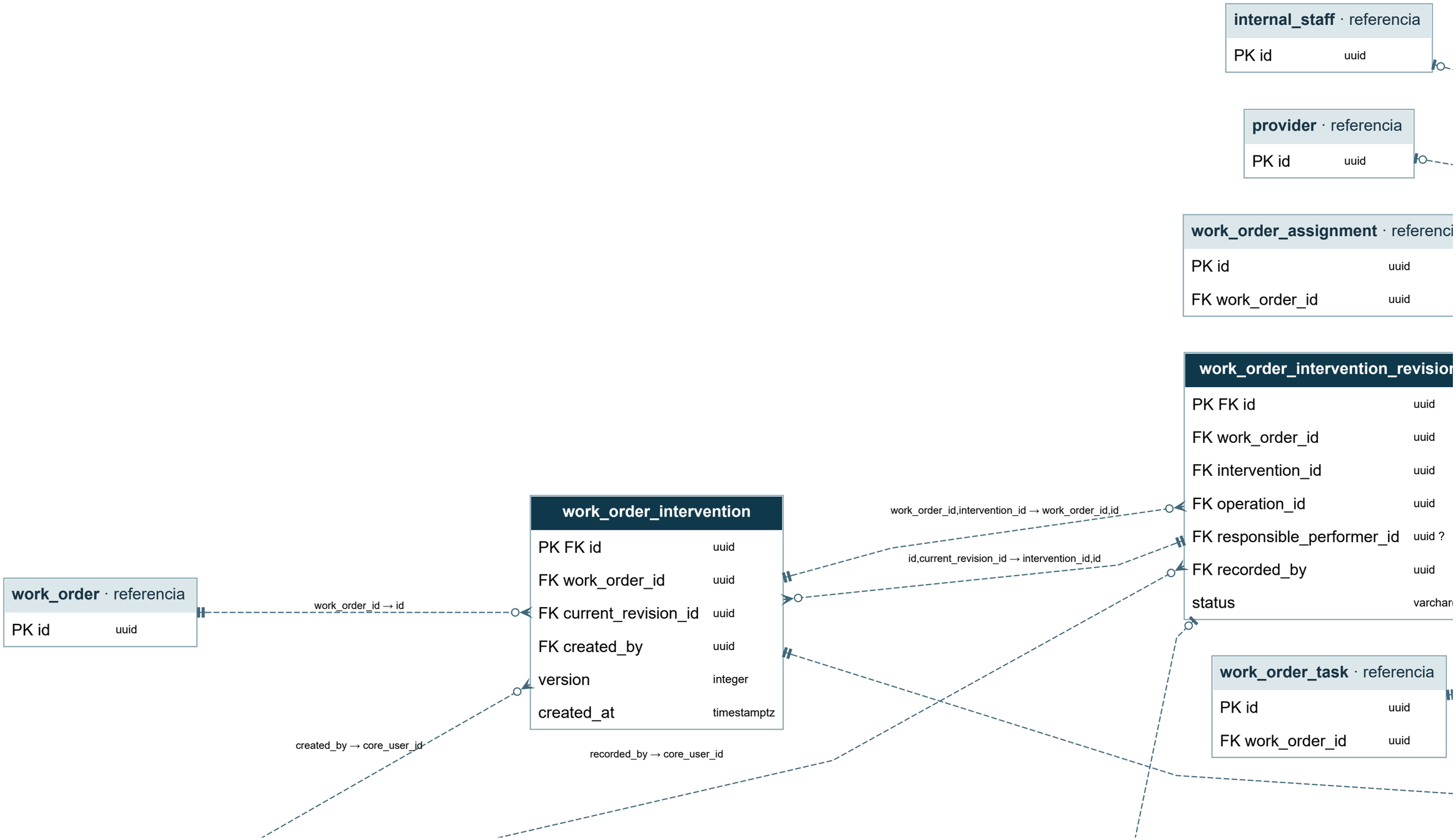


La Rural · modelo de datos

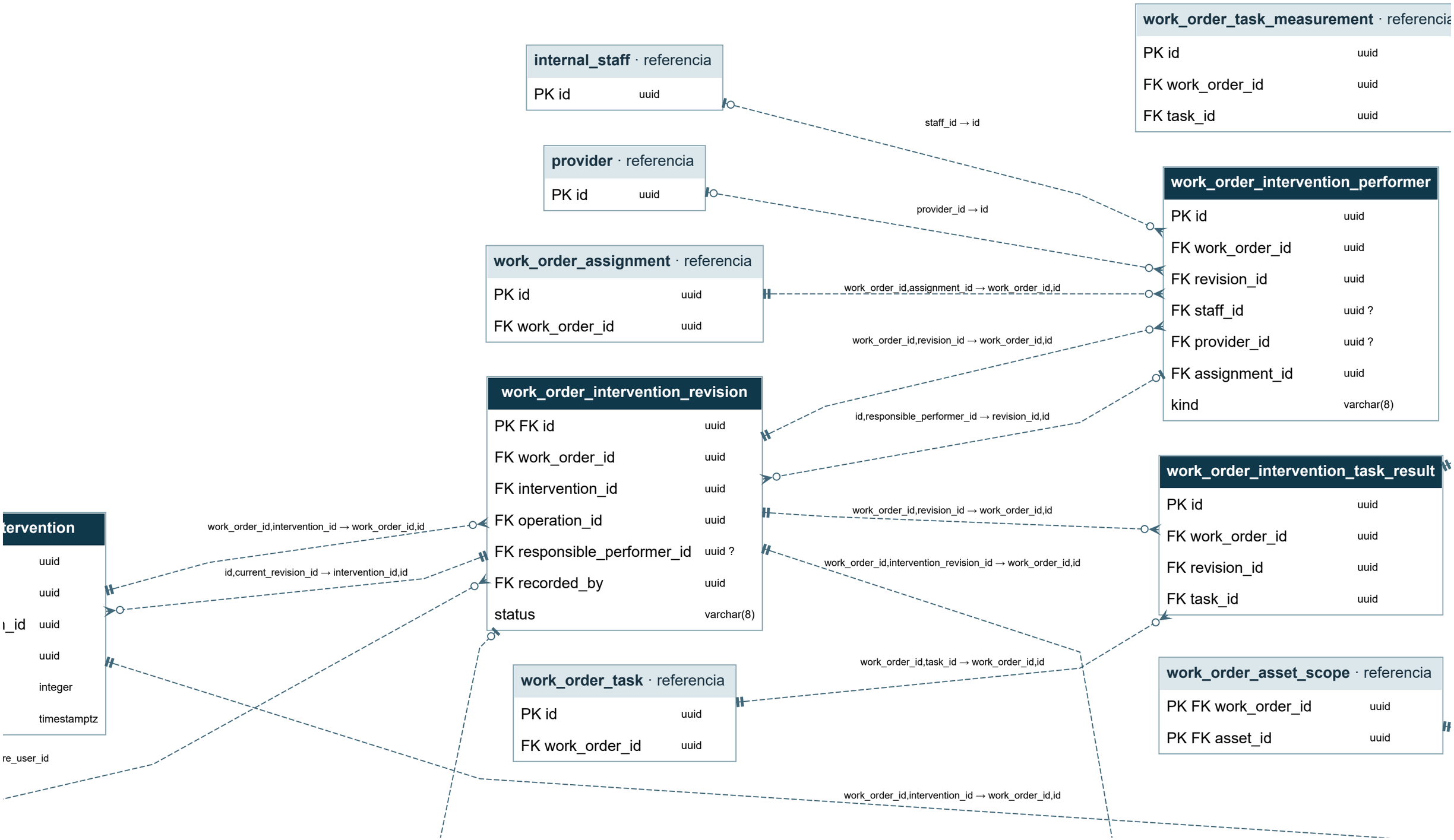
Mantenimiento · Intervenciones y decisiones · 1



Mantenimiento · Intervenc
La Rural · 20/11/2026 · estructura recor



Mantenimiento · Intervenciones y decisiones · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



renciones y decisiones · 1
econstruida desde código y migraciones

work_order_task_measurement · referencia

PK id	uuid
FK work_order_id	uuid
FK task_id	uuid

work_order_id,task_id,measurement_id → work_order_id,task_id,id

work_order_intervention_reading

FK work_order_id	uuid
FK task_id	uuid
PK FK task_result_id	uuid
PK FK measurement_id	uuid

staff_id → id

provider_id → id

work_order_intervention_performer

PK id	uuid
FK work_order_id	uuid
FK revision_id	uuid
FK staff_id	uuid ?
FK provider_id	uuid ?
FK assignment_id	uuid
kind	varchar(8)

rencia

work_order_id,assignment_id → work_order_id,id

work_order_id,revision_id → work_order_id,id

ision

uid
uid
uid
uid
uid ?
uid
archar(8)

id,responsible_performer_id → revision_id,id

work_order_id,task_id,task_result_id → work_order_id,task_id,id

work_order_intervention_task_result

PK id	uuid
FK work_order_id	uuid
FK revision_id	uuid
FK task_id	uuid

work_order_id,revision_id → work_order_id,id

work_order_id,intervention_revision_id → work_order_id,id

work_order_id,task_id → work_order_id,id

ia

work_order_asset_scope · referencia

PK FK work_order_id	uuid
PK FK asset_id	uuid

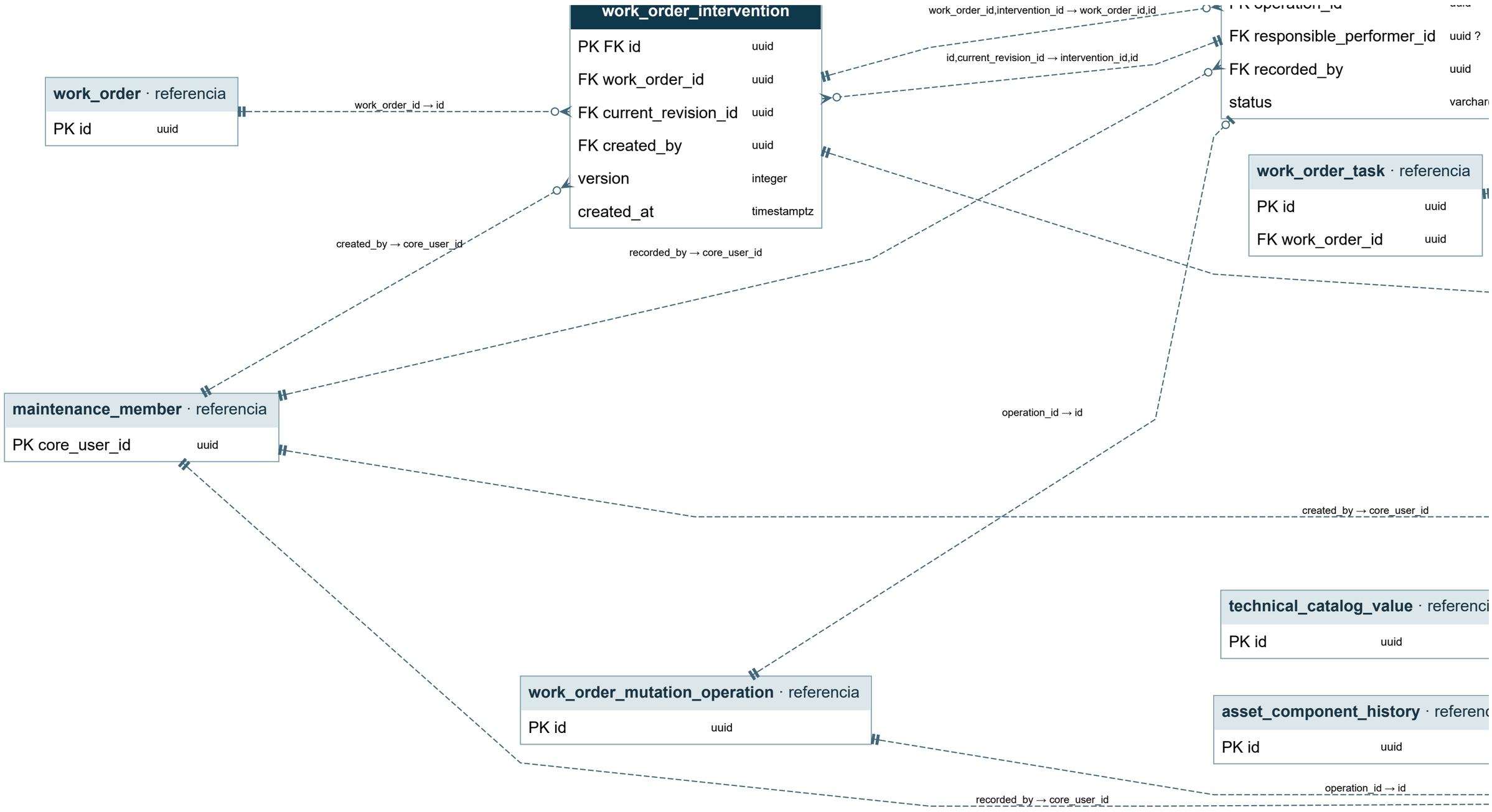
work_order_id,asset_id → work_order_id,asset_id

work_order_id,intervention_id → work_order_id,id

work_order_component_fact

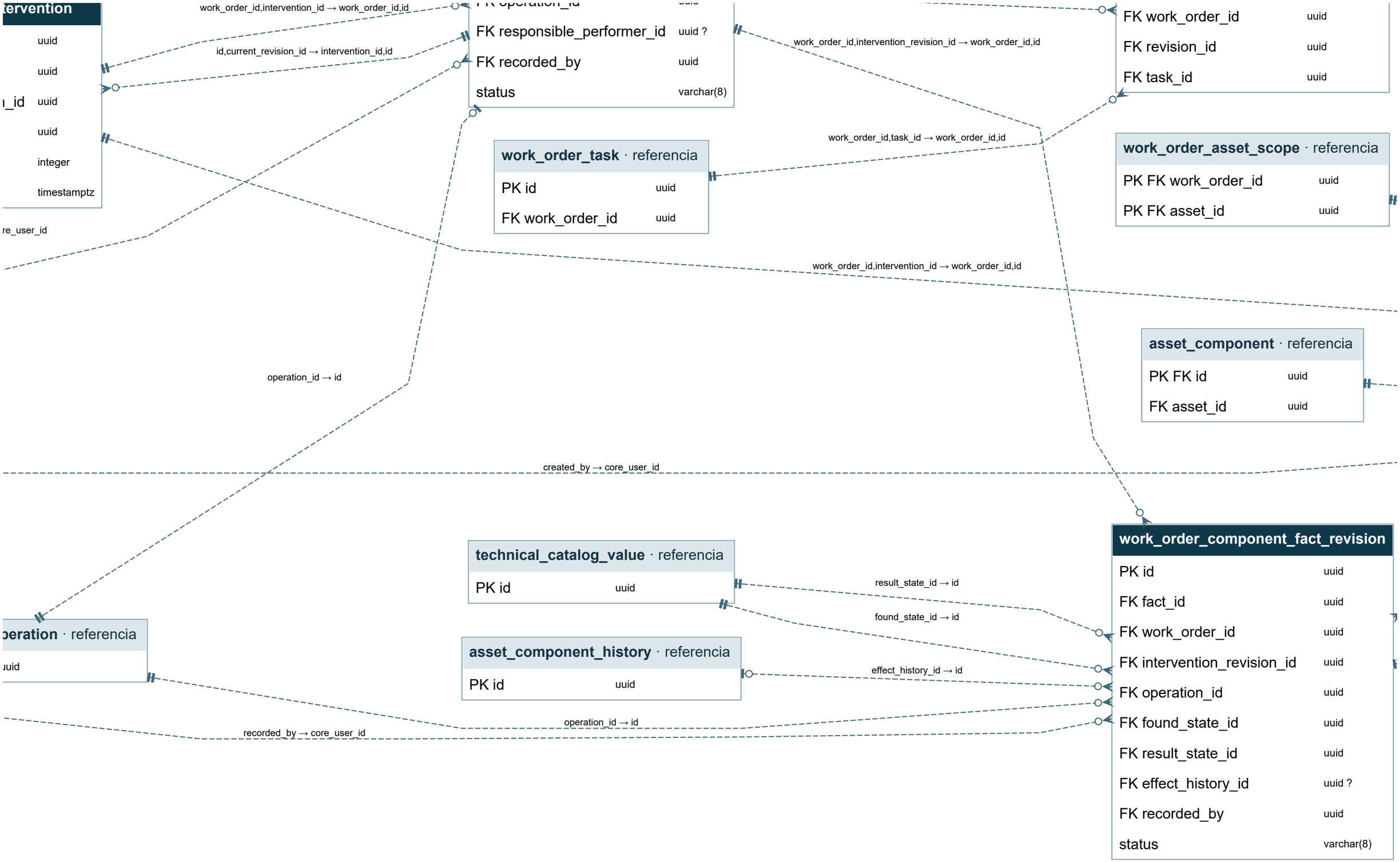
La Rural · modelo de datos

Mantenimiento · Intervenciones y decisiones · 1 · ampliación 2/2, 1/3



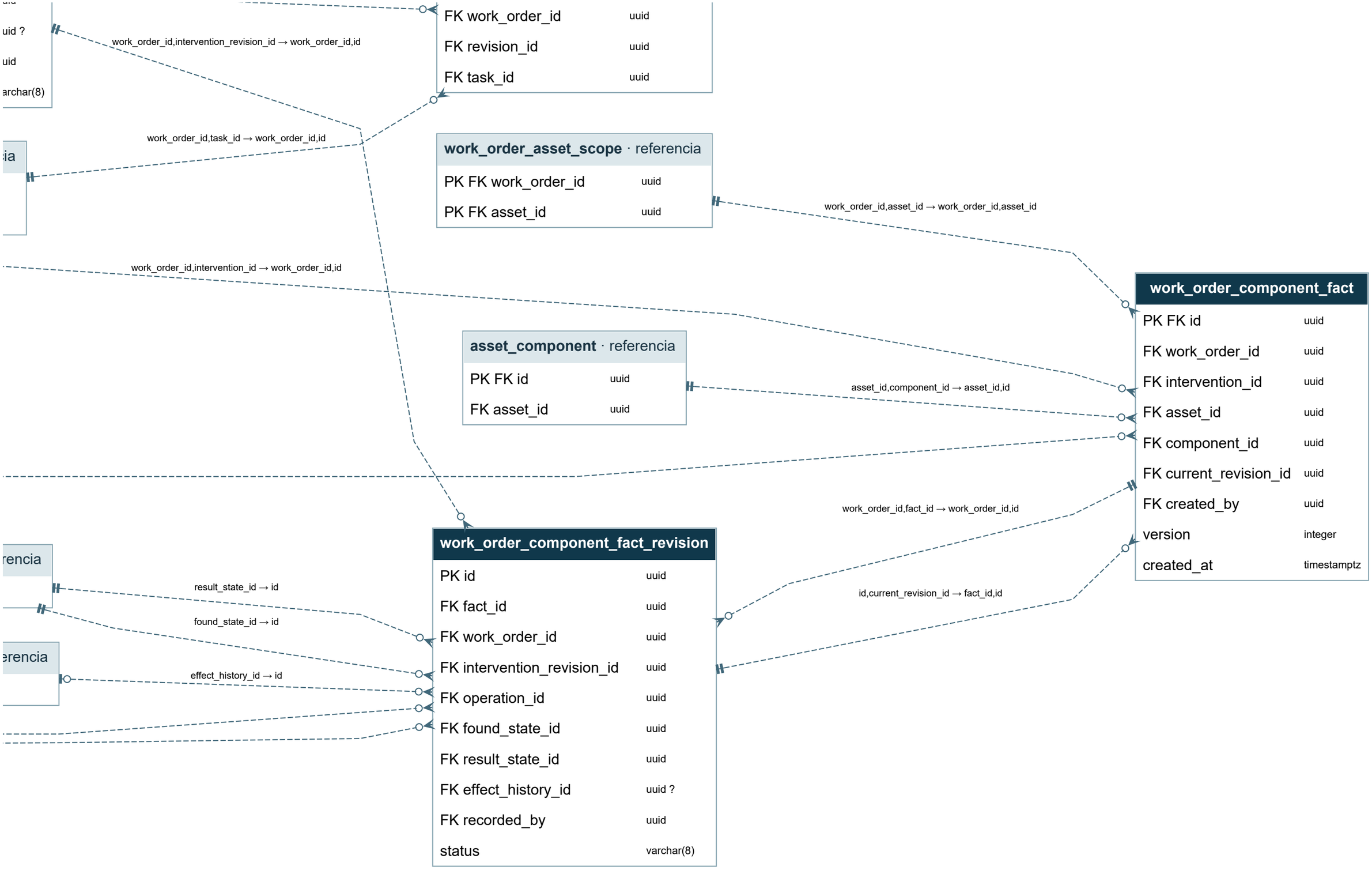
La Rural · modelo de datos

Mantenimiento · Intervenciones y decisiones · 1 · ampliación 2/2, 2/3

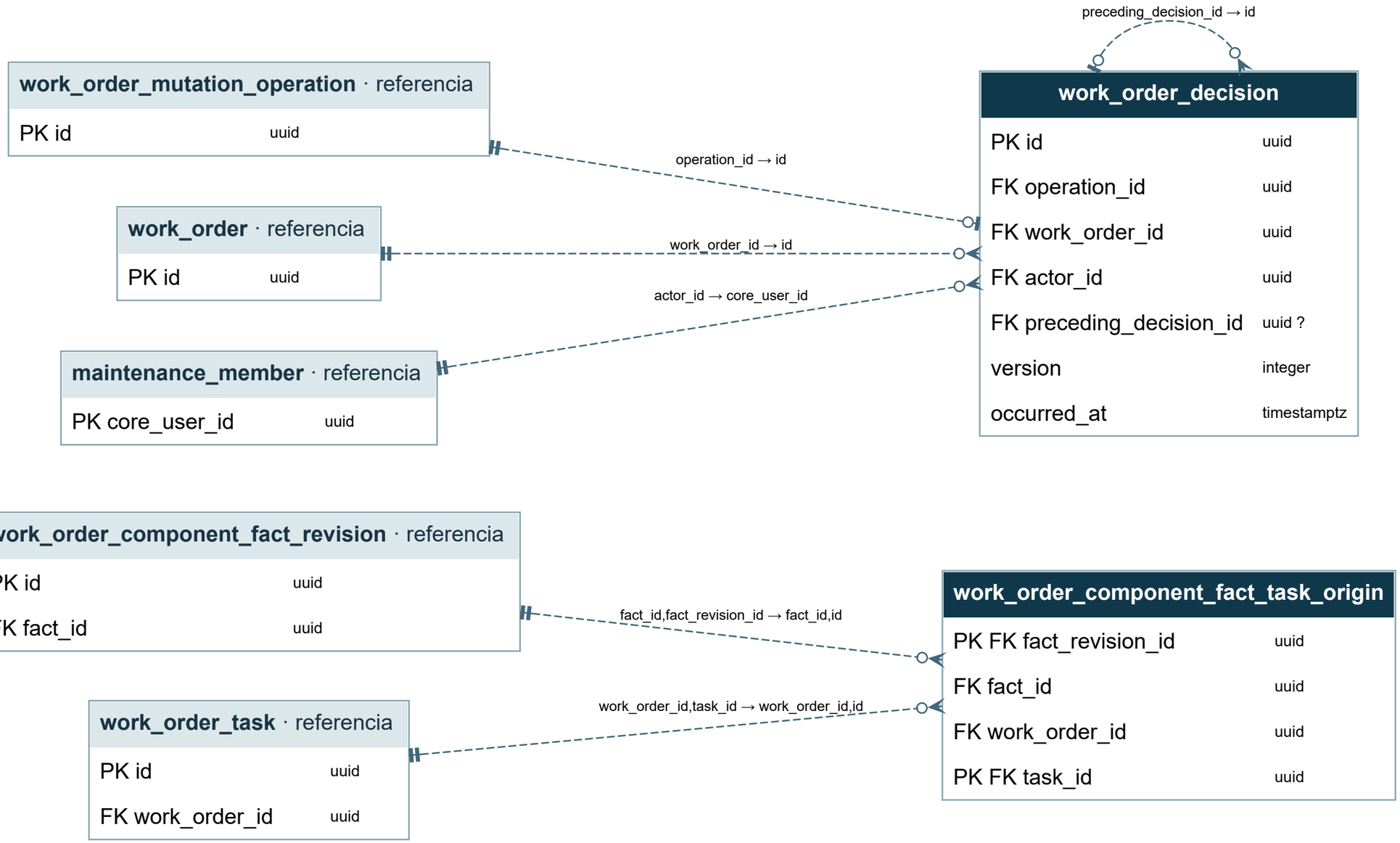


La Rural · modelo de datos

Mantenimiento · Intervenciones y decisiones · 1 · ampliación 2/2, 3/3



Mantenimiento · Intervenciones y decisiones · 2
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



maintenance_member

Proyección local de miembro procedente de Core.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_user_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
core_user_id	uuid	NO	PK	
email_snapshot	varchar(320)	SÍ		
display_name_snapshot	varchar(200)	SÍ		
module_access_status	varchar(16)	NO		
core_profile_version	bigint	SÍ		
core_access_version	bigint	SÍ		
last_synced_at	timestamptz	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
core_user_status	varchar(16)	SÍ		

CHECK: CHECK (email_snapshot IS NULL OR (email_snapshot <> '' AND email_snapshot=lower(btrim(email_snapshot)))) · Migración 1790112000000 · línea 151

CHECK: CHECK (display_name_snapshot IS NULL OR btrim(display_name_snapshot)<>'') · Migración 1790112000000 · línea 151

CHECK: CONSTRAINT maintenance_member_core_user_status_ck CHECK (core_user_status IS NULL OR core_user_status IN ('pending_access','active','suspended','blocked')) · Migración 1790208000000 · línea 65

Índice: CREATE INDEX maintenance_member_status_idx ON esquema.maintenance_member(module_access_status,core_user_id) · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151; Migración 1790208000000 · línea 65.

permission

Permiso granular del dominio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(120)	NO		
name	varchar(160)	NO		
category	varchar(120)	NO		
is_system	boolean	NO		
is_active	boolean	NO		
created_at	timestamptz	NO		

CHECK: CHECK (btrim(code)<>'' AND btrim(name)<>'') · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151.

role

Rol de autorización del dominio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(120)	NO		

Campo	Tipo	NULL	Clave	Referencia física
name	varchar(160)	NO		
description	varchar(4000)	SÍ		
is_system	boolean	NO		
is_protected	boolean	NO		
is_active	boolean	NO		
version	integer	NO		
created_by_core_user_id	uuid	SÍ		
updated_by_core_user_id	uuid	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

CHECK: CHECK (btrim(code)<>'' AND btrim(name)<>'') · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151.

role_permission

Tabla puente entre roles y permisos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: role_id, permission_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
role_id	uuid	NO	PK, FK	role(id)
permission_id	uuid	NO	PK, FK	permission(id)
granted_by_core_user_id	uuid	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
granted_at	timestamptz	NO		

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

FK (permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

Índice: CREATE INDEX maintenance_role_permission_reverse_idx ON esquema.role_permission(permission_id,role_id) · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151.

permission_dependency

Dependencias entre permisos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: permission_id, required_permission_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
permission_id	uuid	NO	PK, FK	permission(id)
required_permission_id	uuid	NO	PK, FK	permission(id)

FK (permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

FK (required_permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

CHECK: CHECK (permission_id<>required_permission_id) · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151.

member_role

Asignación de roles locales a miembros.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_user_id, role_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
core_user_id	uuid	NO	PK, FK	maintenance_member(core_user_id)
role_id	uuid	NO	PK, FK	role(id)
is_active	boolean	NO		
version	integer	NO		
assigned_by_core_user_id	uuid	SÍ		
assigned_at	timestamptz	NO		
revoked_by_core_user_id	uuid	SÍ		
revoked_at	timestamptz	SÍ		
revocation_reason	varchar(1000)	SÍ		
assigned_by_actor_type	varchar(8)	NO		
assigned_by_operation_code	varchar(120)	SÍ		
revoked_by_actor_type	varchar(8)	SÍ		
revoked_by_operation_code	varchar(120)	SÍ		

FK (core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

CHECK: CHECK ((is_active AND revoked_at IS NULL AND revoked_by_core_user_id IS NULL AND revocation_reason IS NULL) OR (NOT is_active AND revoked_at IS NOT NULL AND revocation_reason IS NOT NULL AND btrim(revocation_reason)<>')) · Migración 1790112000000 · línea 151

CHECK: CONSTRAINT maintenance_member_role_assigned_actor_ck CHECK ((assigned_by_actor_type='user' AND assigned_by_core_user_id IS NOT NULL AND assigned_by_operation_code IS NULL) OR (assigned_by_actor_type='system' AND assigned_by_core_user_id IS NULL AND assigned_by_operation_code IS NOT NULL AND btrim(assigned_by_operation_code)<>')) · Migración 1790208000000 · línea 65

CHECK: CONSTRAINT maintenance_member_role_revoked_actor_ck CHECK ((is_active AND revoked_by_actor_type IS NULL AND revoked_by_operation_code IS NULL) OR (NOT is_active AND ((revoked_by_actor_type='user' AND revoked_by_core_user_id IS NOT NULL AND revoked_by_operation_code IS NULL) OR (revoked_by_actor_type='system' AND revoked_by_core_user_id IS NULL AND revoked_by_operation_code IS NOT NULL AND btrim(revoked_by_operation_code)<>')))) · Migración 1790208000000 · línea 65

Índice: CREATE INDEX maintenance_member_role_active_idx ON esquema.member_role(core_user_id,role_id) WHERE is_active · Migración 1790112000000 · línea 151

Índice: CREATE INDEX maintenance_member_role_reverse_idx ON esquema.member_role(role_id,core_user_id) · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151; Migración 1790208000000 · línea 65.

member_role_history

Registro de member role history. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (core_user_id, role_id, version).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
role_id	uuid	NO	FK	role(id)
transition	varchar(16)	NO		
version	integer	NO		
actor_core_user_id	uuid	SÍ		
reason	varchar(1000)	SÍ		
occurred_at	timestamptz	NO		
actor_type	varchar(8)	NO		
actor_operation_code	varchar(120)	SÍ		

FK (core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 151.

CHECK: CONSTRAINT maintenance_role_history_actor_ck CHECK ((actor_type='user' AND actor_core_user_id IS NOT NULL AND actor_operation_code IS NULL) OR (actor_type='system' AND actor_core_user_id IS NULL AND actor_operation_code IS NOT NULL AND btrim(actor_operation_code)<>'')) · Migración 1790208000000 · línea 65

Índice: CREATE INDEX maintenance_member_role_history_member_idx ON esquema.member_role_history(core_user_id,occurred_at DESC) · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151; Migración 1790208000000 · línea 65.

rbac_idempotency_record

Registro de rbac idempotency record. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO		
operation	varchar(120)	NO		
idempotency_key	varchar(120)	NO		
request_hash	char(64)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
completed_at	timestamptz	SÍ		
created_at	timestamptz	NO		
expires_at	timestamptz	NO		

CHECK: CHECK (expires_at>created_at) · Migración 1790112000000 · línea 151

CHECK: CHECK ((completed_at IS NULL AND response_status IS NULL AND response_body IS NULL) OR (completed_at IS NOT NULL AND response_status IS NOT NULL AND response_body IS NOT NULL)) · Migración 1790112000000 · línea 151

Índice: CREATE INDEX maintenance_rbac_idempotency_expiry_idx ON esquema.rbac_idempotency_record(expires_at) · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151.

maintenance_audit_log

Registro de audit log. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: occurred_at, id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
occurred_at	timestamptz	NO	PK	
domain	varchar(120)	NO		
action	varchar(120)	NO		
target_type	varchar(120)	NO		
target_id	varchar(120)	NO		
actor_core_user_id	uuid	SÍ		
actor_email_snapshot	varchar(320)	SÍ		
actor_name_snapshot	varchar(200)	SÍ		
actor_roles_snapshot	jsonb	NO		
outcome	varchar(16)	NO		
reason	varchar(1000)	SÍ		
request_id	varchar(128)	SÍ		
correlation_id	varchar(128)	SÍ		
required_protection_level	integer	NO		

Campo	Tipo	NULL	Clave	Referencia física
actor_type	varchar(8)	NO		
actor_operation_code	varchar(120)	SÍ		

CHECK: CHECK (jsonb_typeof(actor_roles_snapshot)='array') · Migración 1790112000000 · línea 151

CHECK: CONSTRAINT maintenance_audit_actor_ck CHECK ((actor_type='user' AND actor_core_user_id IS NOT NULL AND actor_operation_code IS NULL) OR (actor_type='system' AND actor_core_user_id IS NULL AND actor_operation_code IS NOT NULL AND btrim(actor_operation_code)<>'')) · Migración 1790208000000 · línea 65

Índice: CREATE INDEX maintenance_audit_actor_idx ON esquema.maintenance_audit_log(actor_core_user_id,occurred_at DESC,id DESC) · Migración 1790112000000 · línea 151

Índice: CREATE INDEX maintenance_audit_target_idx ON esquema.maintenance_audit_log(domain,target_type,target_id,occurred_at DESC,id DESC) · Migración 1790112000000 · línea 151

Índice: CREATE INDEX maintenance_audit_correlation_idx ON esquema.maintenance_audit_log(correlation_id,occurred_at DESC,id DESC) · Migración 1790112000000 · línea 151

Índice: CREATE INDEX maintenance_audit_protection_idx ON esquema.maintenance_audit_log(required_protection_level,occurred_at DESC,id DESC) · Migración 1790112000000 · línea 151

Fuentes técnicas: Migración 1790112000000 · línea 151; Migración 1790208000000 · línea 65.

maintenance_audit_change

Registro de audit change. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
audit_occurred_at	timestamptz	NO	FK	maintenance_audit_log(occurred_at, id)
audit_log_id	uuid	NO	FK	maintenance_audit_log(occurred_at, id)
field_code	varchar(120)	NO		
field_label_snapshot	varchar(160)	NO		
old_value	jsonb	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
new_value	jsonb	SÍ		
sort_order	integer	NO		

FK (audit_occurred_at, audit_log_id) → maintenance_audit_log(occurred_at, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790112000000 · línea 269.

Índice: CREATE INDEX maintenance_audit_change_header_idx ON esquema.maintenance_audit_change(audit_occurred_at,audit_log_id,sort_order) · Migración 1790112000000 · línea 269

Fuentes técnicas: Migración 1790112000000 · línea 269.

core_sync_checkpoint

Avance de sincronización externa.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: checkpoint_code. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
checkpoint_code	varchar(120)	NO	PK	
state	varchar(16)	NO		
cursor	text	SÍ		
last_attempt_at	timestamptz	SÍ		
last_success_at	timestamptz	SÍ		
last_error_code	varchar(120)	SÍ		
version	bigint	NO		
updated_at	timestamptz	NO		
snapshot_version	varchar(2000)	SÍ		
projection_revision	bigint	NO		

CHECK: CONSTRAINT core_sync_checkpoint_state_ck CHECK (state IN ('uninitialized','syncing','healthy','degraded','failed')) · Migración 1790208000000 · línea 65

CHECK: CONSTRAINT core_sync_checkpoint_version_ck CHECK (version>=1) · Migración 1790208000000 · línea 65

CHECK: CONSTRAINT core_sync_checkpoint_projection_revision_ck CHECK (projection_revision>=0) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_sync_checkpoint_snapshot_version_ck CHECK (snapshot_version IS NULL OR length(snapshot_version)>0) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_sync_checkpoint_spatial_health_ck CHECK (checkpoint_code<>'core.space-catalog.mantenimiento.v1' OR state<>'healthy' OR last_success_at IS NOT NULL) · Migración 1790812800000 · línea 28

Fuentes técnicas: Migración 1790208000000 · línea 65; Migración 1790812800000 · línea 28.

core_inbox_event

Mensaje externo persistido antes de procesarse.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: source_event_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
source_event_id	uuid	NO	PK	
producer	varchar(16)	NO		
event_type	varchar(120)	NO		
schema_version	integer	NO		
aggregate_type	varchar(120)	NO		
aggregate_id	uuid	NO		
aggregate_version	bigint	NO		
source_cursor	varchar(2000)	SÍ		
correlation_id	varchar(120)	SÍ		
occurred_at	timestamptz	NO		
payload_hash	varchar(64)	NO		

Campo	Tipo	NULL	Clave	Referencia física
payload	jsonb	NO		
processing_status	varchar(16)	NO		
processing_attempts	integer	NO		
received_at	timestamptz	NO		
processed_at	timestamptz	SÍ		
next_attempt_at	timestamptz	SÍ		
lease_expires_at	timestamptz	SÍ		
last_error_code	varchar(120)	SÍ		

CHECK: CONSTRAINT core_inbox_producer_ck CHECK (producer='core') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_schema_version_ck CHECK (schema_version=1) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_event_type_ck CHECK (btrim(event_type)<>'') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_aggregate_type_ck CHECK (btrim(aggregate_type)<>'') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_aggregate_version_ck CHECK (aggregate_version>=1) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_payload_hash_ck CHECK (payload_hash ~ '^[0-9a-f]{64}$') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_payload_ck CHECK (jsonb_typeof(payload)='object') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_status_ck CHECK (processing_status IN ('pending','processing','processed','failed')) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_attempts_ck CHECK (processing_attempts>=0) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_processed_ck CHECK ((processing_status='processed')=(processed_at IS NOT NULL)) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_lease_ck CHECK ((processing_status='processing')=(lease_expires_at IS NOT NULL)) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_retry_ck CHECK ((processing_status='failed')=(next_attempt_at IS NOT NULL)) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_inbox_claimed_ck CHECK (processing_status='pending' OR processing_attempts>=1) · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_inbox_claim_idx ON esquema.core_inbox_event(next_attempt_at,received_at,source_event_id) WHERE processing_status IN ('pending','failed') · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_inbox_lease_recovery_idx ON esquema.core_inbox_event(lease_expires_at,received_at,source_event_id) WHERE processing_status='processing' · Migración 1790812800000 · línea 28

Fuentes técnicas: Migración 1790812800000 · línea 28.

core_space_projection

Proyección local de espacios canónicos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_space_id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
core_space_id	uuid	NO	PK	
code	varchar(120)	NO		
name	varchar(200)	NO		
type_code	varchar(120)	NO		
parent_core_space_id	uuid	SÍ	FK	core_space_projection(core_space_id)
sort_order	integer	NO		
path_text	varchar(4000)	NO		
path_ids	jsonb	NO		
source_status	varchar(16)	NO		
effective_status	varchar(16)	NO		
module_selection_state	varchar(8)	NO		
module_setting_is_explicit	boolean	NO		
module_setting_version	bigint	NO		
selectable	boolean	NO		
context_only	boolean	NO		
color_hex	varchar(7)	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
effective_color_hex	varchar(7)	SÍ		
core_version	bigint	NO		
catalog_sequence	bigint	NO		
is_tombstone	boolean	NO		
synced_at	timestamptz	NO		

FK (parent_core_space_id) → core_space_projection(core_space_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1790812800000 · línea 28.

CHECK: CONSTRAINT core_space_code_ck CHECK (btrim(code)<>'') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_name_ck CHECK (btrim(name)<>'') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_type_ck CHECK (btrim(type_code)<>'') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_sort_order_ck CHECK (sort_order>=0) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_path_ids_ck CHECK (jsonb_typeof(path_ids)='array') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_source_status_ck CHECK (source_status IN ('active','inactive','retired')) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_effective_status_ck CHECK (effective_status IN ('active','inactive','retired')) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_selection_state_ck CHECK (module_selection_state IN ('enabled','disabled')) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_setting_version_ck CHECK ((module_setting_is_explicit AND module_setting_version>=1) OR (NOT module_setting_is_explicit AND module_setting_version=0)) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_selectable_ck CHECK (NOT selectable OR (NOT context_only AND NOT is_tombstone AND parent_core_space_id IS NOT NULL AND effective_status='active' AND module_selection_state='enabled')) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_color_ck CHECK (color_hex ~ '^#[0-9A-Fa-f]{6}$') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_effective_color_ck CHECK (effective_color_hex ~ '^#[0-9A-Fa-f]{6}$') · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_version_ck CHECK (core_version>=1) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_sequence_ck CHECK (catalog_sequence>=0) · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_space_parent_idx ON esquema.core_space_projection(parent_core_space_id,sort_order,core_space_id) · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_space_selectable_idx ON esquema.core_space_projection(selectable,core_space_id) · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_space_status_idx ON esquema.core_space_projection(effective_status,core_space_id) · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_space_sequence_idx ON esquema.core_space_projection(catalog_sequence,core_space_id) · Migración 1790812800000 · línea 28

Fuentes técnicas: Migración 1790812800000 · línea 28.

core_space_relation_projection

Proyección de relaciones espaciales.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: ancestor_core_space_id, descendant_core_space_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
ancestor_core_space_id	uuid	NO	PK, FK	core_space_projection(core_space_id)
descendant_core_space_id	uuid	NO	PK, FK	core_space_projection(core_space_id)
depth	integer	NO		
catalog_sequence	bigint	NO		
synced_at	timestamptz	NO		

FK (ancestor_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790812800000 · línea 28.

FK (descendant_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790812800000 · línea 28.

CHECK: CONSTRAINT core_space_relation_depth_ck CHECK (depth>=0) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_relation_sequence_ck CHECK (catalog_sequence>=0) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_relation_reflexive_ck CHECK ((ancestor_core_space_id=descendant_core_space_id)=(depth=0)) · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_space_relation_reverse_idx ON esquema.core_space_relation_projection(descendant_core_space_id,ancestor_core_space_id) · Migración 1790812800000 · línea 28

Fuentes técnicas: Migración 1790812800000 · línea 28.

core_space_lineage_projection

Proyección de jerarquía espacial.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_lineage_id. Unicidad total: (source_core_space_id, target_core_space_id, relation_type).

Campo	Tipo	NULL	Clave	Referencia física
core_lineage_id	uuid	NO	PK	
source_core_space_id	uuid	NO	FK	core_space_projection(core_space_id)
target_core_space_id	uuid	NO	FK	core_space_projection(core_space_id)
relation_type	varchar(16)	NO		
effective_at	timestamptz	NO		
catalog_sequence	bigint	NO		
synced_at	timestamptz	NO		

FK (source_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790812800000 · línea 28.

FK (target_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1790812800000 · línea 28.

CHECK: CONSTRAINT core_space_lineage_type_ck CHECK (relation_type IN ('split_into','merged_into','replaced_by')) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_lineage_sequence_ck CHECK (catalog_sequence>=0) · Migración 1790812800000 · línea 28

CHECK: CONSTRAINT core_space_lineage_distinct_ck CHECK (source_core_space_id<>target_core_space_id) · Migración 1790812800000 · línea 28

Índice: CREATE INDEX core_space_lineage_target_idx ON esquema.core_space_lineage_projection(target_core_space_id,source_core_space_id) · Migración 1790812800000 · línea 28

Fuentes técnicas: Migración 1790812800000 · línea 28.

technical_catalog_value

Valor técnico normalizado de catálogo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (catalog_key, code); (catalog_key, name_normalized); (catalog_key, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
catalog_key	varchar(40)	NO		
code	varchar(80)	NO		
name	varchar(160)	NO		
name_normalized	text	NO		
description	varchar(4000)	SÍ		
is_active	boolean	NO		
sort_order	integer	NO		
non_operational	boolean	SÍ		
semantic_rank	integer	SÍ		
symbol	varchar(24)	SÍ		
direction	varchar(16)	SÍ		
color_hex	varchar(7)	SÍ		
version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_actor_type	varchar(8)	NO		

Campo	Tipo	NULL	Clave	Referencia física
created_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
created_operation_code	varchar(120)	SÍ		
updated_actor_type	varchar(8)	NO		
updated_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
updated_operation_code	varchar(120)	SÍ		

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791158400000 · línea 32.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791158400000 · línea 32.

CHECK: CONSTRAINT technical_catalog_key_ck CHECK (catalog_key IN ('asset_categories','component_types','asset_operational_states','component_states', 'criticalities','measurement_units','internal_staff_specialties','provider_trades', 'technical_relation_types','work_order_priorities')) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_code_ck CHECK (code ~ '^[a-z][a-z0-9_]{0,79}$') · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_name_ck CHECK (name_normalized <> '') · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_order_ck CHECK (sort_order >= 0) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_version_ck CHECK (version >= 1) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_dates_ck CHECK (updated_at >= created_at) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_non_operational_ck CHECK ((catalog_key='component_states' AND non_operational IS NOT NULL) OR (catalog_key<>'component_states' AND non_operational IS NULL)) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_rank_ck CHECK ((catalog_key IN ('criticalities','work_order_priorities') AND semantic_rank IS NOT NULL AND semantic_rank>0) OR (catalog_key NOT IN ('criticalities','work_order_priorities') AND semantic_rank IS NULL)) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_symbol_ck CHECK ((catalog_key='measurement_units' AND symbol IS NOT NULL AND btrim(symbol)<>'') OR (catalog_key<>'measurement_units' AND symbol IS NULL)) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_direction_ck CHECK ((catalog_key='technical_relation_types' AND direction IS NOT NULL AND direction IN ('directed','symmetric')) OR (catalog_key<>'technical_relation_types' AND direction IS NULL)) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_color_ck CHECK ((catalog_key='work_order_priorities' AND color_hex IS NOT NULL AND color_hex ~ '^#[0-9a-fA-F]{6}$') OR (catalog_key<>'work_order_priorities' AND color_hex IS NULL)) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_created_actor_ck CHECK ((created_actor_type='user' AND created_by_core_user_id IS NOT NULL AND created_operation_code IS NULL) OR (created_actor_type='system' AND created_by_core_user_id IS NULL AND created_operation_code IS NOT NULL AND created_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$')) · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT technical_catalog_updated_actor_ck CHECK ((updated_actor_type='user' AND updated_by_core_user_id IS NOT NULL AND updated_operation_code IS NULL) OR (updated_actor_type='system' AND updated_by_core_user_id IS NULL AND updated_operation_code IS NOT NULL AND updated_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$')) · Migración 1791158400000 · línea 32

Índice: CREATE INDEX technical_catalog_order_idx ON esquema.technical_catalog_value(catalog_key,sort_order,code,id) · Migración 1791158400000 · línea 32

Índice: CREATE INDEX technical_catalog_activation_idx ON esquema.technical_catalog_value(catalog_key,is_active,sort_order,code,id) · Migración 1791158400000 · línea 32

Fuentes técnicas: Migración 1791158400000 · línea 32.

catalog_idempotency_record

Registro de catalog idempotency record. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
operation	varchar(120)	NO		
idempotency_key	varchar(120)	NO		
request_hash	char(64)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
response_etag	varchar(180)	SÍ		
response_location	varchar(512)	SÍ		
completed_at	timestamptz	SÍ		
created_at	timestamptz	NO		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791158400000 · línea 32.

CHECK: CONSTRAINT catalog_idempotency_operation_ck CHECK (operation ~ '^[a-z][a-z0-9_.:-]{0,119}$') · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT catalog_idempotency_key_ck CHECK (idempotency_key ~ '^[A-Za-z0-9._:-]{1,120}$') · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT catalog_idempotency_hash_ck CHECK (request_hash ~ '^[0-9a-f]{64}$') · Migración 1791158400000 · línea 32

CHECK: CONSTRAINT catalog_idempotency_response_ck CHECK ((completed_at IS NULL AND response_status IS NULL AND response_body IS NULL AND response_etag IS NULL AND response_location IS NULL) OR (completed_at IS NOT NULL AND completed_at>=created_at AND response_status IS NOT NULL AND response_status IN (200,201) AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND response_etag IS NOT NULL)) · Migración 1791158400000 · línea 32

Fuentes técnicas: Migración 1791158400000 · línea 32.

asset_type

Definición de familia técnica de activos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
name	varchar(160)	SÍ		
name_normalized	text	SÍ		
category_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
category_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id)
description	varchar(4000)	SÍ		
status	varchar(8)	NO		
version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
updated_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (category_catalog_key, category_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

CHECK: CONSTRAINT asset_type_name_ck CHECK (name IS NULL OR name_normalized<>'') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_category_ck CHECK (category_catalog_key='asset_categories') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_status_ck CHECK (status IN ('draft','active','inactive')) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_activation_ck CHECK (status<>'active' OR (name IS NOT NULL AND category_id IS NOT NULL)) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_version_ck CHECK (version>=1) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_dates_ck CHECK (updated_at>=created_at) · Migración 1791331200000 · línea 39

Índice: CREATE UNIQUE INDEX asset_type_active_name_key ON esquema.asset_type(name_normalized) WHERE status='active' · Migración 1791331200000 · línea 39

Índice: CREATE INDEX asset_type_list_idx ON esquema.asset_type(status,updated_at DESC,id) · Migración 1791331200000 · línea 39

Índice: CREATE INDEX asset_type_category_list_idx ON esquema.asset_type(category_id,status,updated_at DESC,id) · Migración 1791331200000 · línea 39

Fuentes técnicas: Migración 1791331200000 · línea 39.

asset_type_attribute

Registro de asset type attribute. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (asset_type_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
asset_type_id	uuid	NO	FK	asset_type(id)
label	varchar(160)	SÍ		
data_type	varchar(24)	SÍ		
is_required	boolean	NO		

Campo	Tipo	NULL	Clave	Referencia física
help_text	varchar(4000)	SÍ		
sort_order	integer	NO		
is_active	boolean	NO		
numeric_kind	varchar(8)	SÍ		
minimum	numeric	SÍ		
maximum	numeric	SÍ		
unit_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
unit_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id)
created_at	timestamptz	NO		

FK (asset_type_id) → asset_type(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

FK (unit_catalog_key, unit_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

CHECK: CONSTRAINT asset_type_attribute_label_ck CHECK (label IS NULL OR btrim(label)<>'') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_attribute_data_type_ck CHECK (data_type IS NULL OR data_type IN ('short_text','long_text','number','number_with_unit','date','boolean','single_choice')) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_attribute_order_ck CHECK (sort_order>=0) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_attribute_numeric_ck CHECK ((data_type IS NOT NULL AND data_type IN ('number','number_with_unit') AND (numeric_kind IS NULL OR numeric_kind IN ('integer','decimal'))) OR (numeric_kind IS NULL AND minimum IS NULL AND maximum IS NULL)) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_attribute_limits_ck CHECK ((minimum IS NULL OR minimum::text NOT IN ('NaN','Infinity','-Infinity')) AND (maximum IS NULL OR maximum::text NOT IN ('NaN','Infinity','-Infinity')) AND (minimum IS NULL OR maximum IS NULL OR minimum<=maximum) AND (numeric_kind IS DISTINCT FROM 'integer' OR ((minimum IS NULL OR minimum=trunc(minimum)) AND (maximum IS NULL OR maximum=trunc(maximum))))) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_attribute_unit_ck CHECK (unit_catalog_key='measurement_units' AND (unit_id IS NULL OR (data_type IS NOT NULL AND data_type='number_with_unit'))) · Migración 1791331200000 · línea 39

Índice: CREATE INDEX asset_type_attribute_order_idx ON esquema.asset_type_attribute(asset_type_id,sort_order,id) · Migración 1791331200000 · línea 39

Fuentes técnicas: Migración 1791331200000 · línea 39.

asset_type_attribute_option

Registro de asset type attribute option. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (asset_type_id, attribute_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
asset_type_id	uuid	NO	FK	asset_type_attribute(asset_type_id, id)
attribute_id	uuid	NO	FK	asset_type_attribute(asset_type_id, id)
label	varchar(160)	SÍ		
sort_order	integer	NO		
is_active	boolean	NO		
created_at	timestamptz	NO		

FK (asset_type_id, attribute_id) → asset_type_attribute(asset_type_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

CHECK: CONSTRAINT asset_type_option_label_ck CHECK (label IS NULL OR btrim(label)<>'') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_option_order_ck CHECK (sort_order>=0) · Migración 1791331200000 · línea 39

Índice: CREATE INDEX asset_type_option_order_idx ON esquema.asset_type_attribute_option(asset_type_id,attribute_id,sort_order,id) · Migración 1791331200000 · línea 39

Fuentes técnicas: Migración 1791331200000 · línea 39.

asset_type_component_suggestion

Registro de asset type component suggestion. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (asset_type_id, id); (asset_type_id, reference_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK, FK	asset_type_component_state(asset_type_id, component_suggestion_id, state_id)
asset_type_id	uuid	NO	FK	asset_type(id); asset_type_component_state(asset_type_id, component_suggestion_id, state_id)
reference_key	varchar(80)	NO		
component_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
component_type_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id)
suggested_name	varchar(160)	SÍ		
suggested_quantity	integer	SÍ		
state_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
initial_state_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id); asset_type_component_state(asset_type_id, component_suggestion_id, state_id)
included_by_default	boolean	NO		
help_text	varchar(4000)	SÍ		
sort_order	integer	NO		
is_active	boolean	NO		
created_at	timestamptz	NO		

FK (asset_type_id) → asset_type(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

FK (component_catalog_key, component_type_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

FK (state_catalog_key, initial_state_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

FK (asset_type_id, id, initial_state_id) → asset_type_component_state(asset_type_id, component_suggestion_id, state_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

CHECK: CONSTRAINT asset_type_component_reference_ck CHECK (reference_key ~ '^[a-z][a-z0-9_]{0,79}$') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_component_catalog_ck CHECK (component_catalog_key='component_types') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_component_name_ck CHECK (suggested_name IS NULL OR btrim(suggested_name)<>'') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_component_quantity_ck CHECK (suggested_quantity IS NULL OR suggested_quantity>=1) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_component_order_ck CHECK (sort_order>=0) · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_component_state_ck CHECK (state_catalog_key='component_states') · Migración 1791331200000 · línea 39

Índice: CREATE INDEX asset_type_component_order_idx ON esquema.asset_type_component_suggestion(asset_type_id,sort_order,id) · Migración 1791331200000 · línea 39

Fuentes técnicas: Migración 1791331200000 · línea 39.

asset_type_component_state

Registro de asset type component state. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: asset_type_id, component_suggestion_id, state_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
asset_type_id	uuid	NO	PK, FK	asset_type_component_suggestion(asset_type_id, id)
component_suggestion_id	uuid	NO	PK, FK	asset_type_component_suggestion(asset_type_id, id)
state_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
state_id	uuid	NO	PK, FK	technical_catalog_value(catalog_key, id)

FK (asset_type_id, component_suggestion_id) → asset_type_component_suggestion(asset_type_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

FK (state_catalog_key, state_id) → technical_catalog_value(catalog_key, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

CHECK: CONSTRAINT asset_type_component_state_catalog_ck CHECK (state_catalog_key='component_states') · Migración 1791331200000 · línea 39

Índice: CREATE INDEX asset_type_component_state_catalog_idx ON esquema.asset_type_component_state(state_catalog_key,state_id) · Migración 1791331200000 · línea 39

Fuentes técnicas: Migración 1791331200000 · línea 39.

asset_type_idempotency_record

Registro de asset type idempotency record. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
operation	varchar(120)	NO		
idempotency_key	varchar(120)	NO		
request_hash	char(64)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
response_etag	varchar(180)	SÍ		
response_location	varchar(512)	SÍ		
completed_at	timestamptz	SÍ		
created_at	timestamptz	NO		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791331200000 · línea 39.

CHECK: CONSTRAINT asset_type_idempotency_operation_ck CHECK (operation ~ '^[a-z][a-z0-9_.:-]{0,119}$') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_idempotency_key_ck CHECK (idempotency_key ~ '^[A-Za-z0-9._:-]{1,120}$') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_idempotency_hash_ck CHECK (request_hash ~ '^[0-9a-f]{64}$') · Migración 1791331200000 · línea 39

CHECK: CONSTRAINT asset_type_idempotency_response_ck CHECK ((completed_at IS NULL AND response_status IS NULL AND response_body IS NULL AND response_etag IS NULL AND response_location IS NULL) OR (completed_at IS NOT NULL AND completed_at>=created_at AND response_status IS NOT NULL AND response_status IN (200,201) AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND response_etag IS NOT NULL)) · Migración 1791331200000 · línea 39

Fuentes técnicas: Migración 1791331200000 · línea 39.

asset

Activo mantenible.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (id, asset_type_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(160)	SÍ		
code_normalized	text	SÍ		
name	varchar(160)	SÍ		
asset_type_id	uuid	SÍ	FK	asset_type(id)
space_id	uuid	SÍ	FK	core_space_projection(core_space_id)
operational_state_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
operational_state_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id)
criticality_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
criticality_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id)
manufacturer	varchar(160)	SÍ		
model	varchar(160)	SÍ		
serial_number	varchar(160)	SÍ		
commissioned_on	date	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
notes	varchar(4000)	SÍ		
condition	varchar(10)	NO		
version	integer	NO		
draft_data	jsonb	SÍ		
registered_at	timestamptz	SÍ		
archived_at	timestamptz	SÍ		
archive_reason	varchar(1000)	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
updated_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (asset_type_id) → asset_type(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (space_id) → core_space_projection(core_space_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (operational_state_catalog_key, operational_state_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (criticality_catalog_key, criticality_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_state_catalog_ck CHECK (operational_state_catalog_key='asset_operational_states') · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_criticality_catalog_ck CHECK (criticality_catalog_key='criticalities') · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_condition_ck CHECK (condition IN ('draft','registered','archived')) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_code_ck CHECK (code IS NULL OR (code=btrim(code) AND code<>'')) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_name_ck CHECK (name IS NULL OR btrim(name)<>'') · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_version_ck CHECK (version>=1) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_dates_ck CHECK (updated_at>=created_at AND (registered_at IS NULL OR registered_at>=created_at)) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_registration_ck CHECK ((registered_at IS NULL AND condition<>'registered') OR (registered_at IS NOT NULL AND condition<>'draft' AND code IS NOT NULL AND name IS NOT NULL AND asset_type_id IS NOT NULL AND space_id IS NOT NULL AND operational_state_id IS NOT NULL AND criticality_id IS NOT NULL)) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_draft_data_ck CHECK ((registered_at IS NULL AND draft_data IS NOT NULL AND jsonb_typeof(draft_data)='object') OR (registered_at IS NOT NULL AND draft_data IS NULL)) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_archive_ck CHECK ((condition='archived' AND archived_at IS NOT NULL AND archive_reason IS NOT NULL AND btrim(archive_reason)<>'') OR (condition<>'archived' AND archived_at IS NULL AND archive_reason IS NULL)) · Migración 1791504000000 · línea 44

Índice: CREATE UNIQUE INDEX asset_code_key ON esquema.asset(code_normalized) WHERE code IS NOT NULL · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_list_idx ON esquema.asset(condition,created_at DESC,id DESC) · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_by_type_list_idx ON esquema.asset(asset_type_id,condition,created_at DESC,id DESC) · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_space_list_idx ON esquema.asset(space_id,condition,created_at DESC,id DESC) · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_status_list_idx ON esquema.asset(operational_state_id,criticality_id,condition) · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_code_prefix_idx ON esquema.asset(code_normalized text_pattern_ops,created_at DESC,id DESC) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_attribute_value

Registro de asset attribute value. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: asset_id, attribute_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
asset_id	uuid	NO	PK, FK	asset(id, asset_type_id)
asset_type_id	uuid	NO	FK	asset(id, asset_type_id); asset_type_attribute(asset_type_id, id); asset_type_attribute_option(asset_type_id, attribute_id, id)
attribute_id	uuid	NO	PK, FK	asset_type_attribute(asset_type_id, id); asset_type_attribute_option(asset_type_id, attribute_id, id)
data_type	varchar(24)	NO		

Campo	Tipo	NULL	Clave	Referencia física
value_text	varchar(4000)	SÍ		
value_number	numeric	SÍ		
value_date	date	SÍ		
value_boolean	boolean	SÍ		
option_id	uuid	SÍ	FK	asset_type_attribute_option(asset_type_id, attribute_id, id)
unit_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
unit_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id)
updated_asset_version	integer	NO		

FK (asset_id, asset_type_id) → asset(id, asset_type_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (asset_type_id, attribute_id) → asset_type_attribute(asset_type_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (asset_type_id, attribute_id, option_id) → asset_type_attribute_option(asset_type_id, attribute_id, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (unit_catalog_key, unit_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_attribute_value_version_ck CHECK (updated_asset_version>=1) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_attribute_value_shape_ck CHECK (data_type IN ('short_text','long_text','number','number_with_unit','date','boolean','single_choice') AND num_nonnulls(value_text,value_number,value_date,value_boolean,option_id)<=1 AND (value_text IS NULL OR (data_type IN ('short_text','long_text') AND btrim(value_text)<>'' AND (data_type<>'short_text' OR length(value_text)<=160))) AND (value_number IS NULL OR (data_type IN ('number','number_with_unit') AND value_number::text NOT IN ('NaN','Infinity','-Infinity'))) AND (value_date IS NULL OR data_type='date') AND (value_boolean IS NULL OR data_type='boolean') AND (option_id IS NULL OR data_type='single_choice') AND unit_catalog_key='measurement_units' AND (unit_id IS NULL OR data_type='number_with_unit') AND (data_type<>'number_with_unit' OR value_number IS NULL OR unit_id IS NOT NULL)) · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_attribute_value_usage_idx ON esquema.asset_attribute_value(asset_type_id,attribute_id,option_id) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_attribute_value_history

Registro de asset attribute value history. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (asset_id, attribute_id, asset_version).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
asset_id	uuid	NO	FK	asset(id, asset_type_id)
asset_type_id	uuid	NO	FK	asset(id, asset_type_id); asset_type_attribute(asset_type_id, id); asset_type_attribute_option(asset_type_id, attribute_id, id)
attribute_id	uuid	NO	FK	asset_type_attribute(asset_type_id, id); asset_type_attribute_option(asset_type_id, attribute_id, id)
asset_version	integer	NO		
data_type	varchar(24)	NO		
value_text	varchar(4000)	SÍ		
value_number	numeric	SÍ		
value_date	date	SÍ		
value_boolean	boolean	SÍ		
option_id	uuid	SÍ	FK	asset_type_attribute_option(asset_type_id, attribute_id, id)
unit_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
unit_id	uuid	SÍ	FK	technical_catalog_value(catalog_key, id)
occurred_at	timestamptz	NO		
actor_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (asset_id, asset_type_id) → asset(id, asset_type_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (asset_type_id, attribute_id) → asset_type_attribute(asset_type_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (asset_type_id, attribute_id, option_id) → asset_type_attribute_option(asset_type_id, attribute_id, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (unit_catalog_key, unit_id) → technical_catalog_value(catalog_key, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (actor_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_attribute_history_version_ck CHECK (asset_version>=1) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_attribute_history_shape_ck CHECK (data_type IN ('short_text','long_text','number','number_with_unit','date','boolean','single_choice') AND num_nonnulls(value_text,value_number,value_date,value_boolean,option_id)<=1 AND (value_text IS NULL OR (data_type IN ('short_text','long_text') AND btrim(value_text)<>'' AND (data_type<>'short_text' OR length(value_text)<=160))) AND (value_number IS NULL OR (data_type IN ('number','number_with_unit') AND value_number::text NOT IN ('NaN','Infinity','-Infinity'))) AND (value_date IS NULL OR data_type='date') AND (value_boolean IS NULL OR data_type='boolean') AND (option_id IS NULL OR data_type='single_choice') AND unit_catalog_key='measurement_units' AND (unit_id IS NULL OR data_type='number_with_unit') AND (data_type<>'number_with_unit' OR value_number IS NULL OR unit_id IS NOT NULL)) · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_attribute_history_usage_idx ON esquema.asset_attribute_value_history(asset_type_id,attribute_id,option_id) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_component

Componente concreto de un activo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (asset_id, id); (asset_type_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK, FK	asset_component_applicable_state(component_id, state_id)
asset_id	uuid	NO	FK	asset(id, asset_type_id)
asset_type_id	uuid	NO	FK	asset(id, asset_type_id); asset_type_component_suggestion(asset_type_id, id)
composition_reference_id	uuid	NO	FK	asset_type_component_suggestion(asset_type_id, id)
name	varchar(160)	NO		
quantity	integer	NO		
current_state_id	uuid	NO	FK	asset_component_applicable_state(component_id, state_id)
observations	varchar(4000)	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
version	integer	NO		
created_asset_version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
updated_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
condition	varchar(8)	NO		
retired_at	timestamptz	SÍ		
retirement_reason	varchar(1000)	SÍ		

FK (asset_id, asset_type_id) → asset(id, asset_type_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (asset_type_id, composition_reference_id) → asset_type_component_suggestion(asset_type_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (id, current_state_id) → asset_component_applicable_state(component_id, state_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_component_name_ck CHECK (btrim(name)<>'') · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_component_quantity_ck CHECK (quantity>=1) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_component_version_ck CHECK (version>=1 AND created_asset_version>=1) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_component_dates_ck CHECK (updated_at>=created_at) · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_component_condition_ck CHECK (condition IN ('current','retired')) · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_component_retirement_ck CHECK ((condition='current' AND retired_at IS NULL AND retirement_reason IS NULL) OR (condition='retired' AND retired_at IS NOT NULL AND retired_at>=created_at AND retirement_reason IS NOT NULL AND btrim(retirement_reason)<>'')) · Migración 1791676800000 · línea 42

Índice: CREATE INDEX asset_component_reference_idx ON esquema.asset_component(asset_type_id,composition_reference_id) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44; Migración 1791676800000 · línea 42.

asset_component_applicable_state

Registro de asset component applicable state. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: component_id, state_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
component_id	uuid	NO	PK, FK	asset_component(id)
state_id	uuid	NO	PK, FK	technical_catalog_value(catalog_key, id)
state_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)

FK (component_id) → asset_component(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (state_catalog_key, state_id) → technical_catalog_value(catalog_key, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_component_applicable_catalog_ck CHECK (state_catalog_key='component_states') · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_component_applicable_catalog_idx ON esquema.asset_component_applicable_state(state_id) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_component_history

Registro de asset component history. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (component_id, component_version).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
asset_id	uuid	NO	FK	asset_component(asset_id, id)
component_id	uuid	NO	FK	asset_component(asset_id, id); asset_component(asset_type_id, id)

Campo	Tipo	NULL	Clave	Referencia física
asset_type_id	uuid	NO	FK	asset_component(asset_type_id, id); asset_type_component_suggestion(asset_type_id, id)
composition_reference_id	uuid	NO	FK	asset_type_component_suggestion(asset_type_id, id)
component_version	integer	NO		
action	varchar(120)	NO		
snapshot	jsonb	NO		
occurred_at	timestamptz	NO		
actor_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (asset_id, component_id) → asset_component(asset_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (asset_type_id, component_id) → asset_component(asset_type_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (asset_type_id, composition_reference_id) → asset_type_component_suggestion(asset_type_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (actor_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_component_history_shape_ck CHECK (component_version>=1 AND action ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND jsonb_typeof(snapshot)='object') · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_component_history_usage_idx ON esquema.asset_component_history(asset_type_id,composition_reference_id) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_history

Registro de asset history. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (asset_id, asset_version, event_order).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
asset_id	uuid	NO	FK	asset(id)

Campo	Tipo	NULL	Clave	Referencia física
asset_version	integer	NO		
event_order	integer	NO		
action	varchar(120)	NO		
reason	varchar(1000)	SÍ		
changes	jsonb	NO		
occurred_at	timestamptz	NO		
actor_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
request_id	varchar(120)	NO		
correlation_id	varchar(120)	NO		

FK (asset_id) → asset(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (actor_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_history_shape_ck CHECK (asset_version>=1 AND event_order>=0 AND action ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND jsonb_typeof(changes)='array') · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_replacement

Registro de asset replacement. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: predecessor_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
predecessor_id	uuid	NO	PK, FK	asset(id)
successor_id	uuid	NO	FK	asset(id)
occurred_at	timestamptz	NO		

Campo	Tipo	NULL	Clave	Referencia física
actor_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
reason	varchar(1000)	SÍ		

FK (predecessor_id) → asset(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1791504000000 · línea 44.

FK (successor_id) → asset(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

FK (actor_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_replacement_distinct_ck CHECK (predecessor_id<>successor_id) · Migración 1791504000000 · línea 44

Índice: CREATE INDEX asset_replacement_successor_idx ON esquema.asset_replacement(successor_id) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_idempotency_record

Registro de asset idempotency record. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
operation	varchar(120)	NO		
idempotency_key	varchar(120)	NO		
request_hash	char(64)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
response_etag	varchar(180)	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
response_location	varchar(512)	SÍ		
completed_at	timestamptz	SÍ		
created_at	timestamptz	NO		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791504000000 · línea 44.

CHECK: CONSTRAINT asset_idempotency_operation_ck CHECK (operation ~ '^[a-z][a-z0-9_.:-]{0,119}$') · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_idempotency_key_ck CHECK (idempotency_key ~ '^[A-Za-z0-9._:-]{1,120}$') · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_idempotency_hash_ck CHECK (request_hash ~ '^[0-9a-f]{64}$') · Migración 1791504000000 · línea 44

CHECK: CONSTRAINT asset_idempotency_response_ck CHECK ((completed_at IS NULL AND response_status IS NULL AND response_body IS NULL AND response_etag IS NULL AND response_location IS NULL) OR (completed_at IS NOT NULL AND completed_at>=created_at AND response_status IS NOT NULL AND response_status IN (200,201) AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND response_etag IS NOT NULL)) · Migración 1791504000000 · línea 44

Fuentes técnicas: Migración 1791504000000 · línea 44.

asset_relation

Relación técnica entre activos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (id, origin_asset_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
relation_type_catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
relation_type_id	uuid	NO	FK	technical_catalog_value(catalog_key, id)
direction	varchar(16)	NO		
origin_asset_id	uuid	NO	FK	asset(id)
destination_asset_id	uuid	NO	FK	asset(id)

Campo	Tipo	NULL	Clave	Referencia física
observations	varchar(4000)	SÍ		
condition	varchar(8)	NO		
version	integer	NO		
finished_at	timestamptz	SÍ		
finalization_reason	varchar(1000)	SÍ		
restored_from_relation_id	uuid	SÍ	FK	asset_relation(id)
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
updated_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (relation_type_catalog_key, relation_type_id) → technical_catalog_value(catalog_key, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

FK (origin_asset_id) → asset(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

FK (destination_asset_id) → asset(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

FK (restored_from_relation_id) → asset_relation(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

CHECK: CONSTRAINT asset_relation_catalog_ck CHECK (relation_type_catalog_key='technical_relation_types') · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_relation_direction_ck CHECK (direction IN ('directed','symmetric')) · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_relation_endpoints_ck CHECK (origin_asset_id<>destination_asset_id AND (direction<>'symmetric' OR origin_asset_id<destination_asset_id)) · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_relation_condition_ck CHECK (condition IN ('active','finished')) · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_relation_version_ck CHECK (version>=1) · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_relation_dates_ck CHECK (updated_at>=created_at) · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_relation_finalization_ck CHECK ((condition='active' AND finished_at IS NULL AND finalization_reason IS NULL) OR (condition='finished' AND finished_at IS NOT NULL AND finished_at>=created_at AND finalization_reason IS NOT NULL AND btrim(finalization_reason)<>'')) · Migración 1791676800000 · línea 42

CHECK: CONSTRAINT asset_relation_restored_from_ck CHECK (restored_from_relation_id IS NULL OR restored_from_relation_id<>id) · Migración 1791676800000 · línea 42

Índice: CREATE UNIQUE INDEX asset_relation_active_key ON esquema.asset_relation(relation_type_id,origin_asset_id,destination_asset_id) WHERE condition='active' · Migración 1791676800000 · línea 42

Índice: CREATE UNIQUE INDEX asset_relation_restoration_key ON esquema.asset_relation(restored_from_relation_id) WHERE restored_from_relation_id IS NOT NULL · Migración 1791676800000 · línea 42

Índice: CREATE INDEX asset_relation_origin_list_idx ON esquema.asset_relation(origin_asset_id,condition,created_at DESC,id DESC) · Migración 1791676800000 · línea 42

Índice: CREATE INDEX asset_relation_destination_list_idx ON esquema.asset_relation(destination_asset_id,condition,created_at DESC,id DESC) · Migración 1791676800000 · línea 42

Fuentes técnicas: Migración 1791676800000 · línea 42.

asset_relation_participant

Participantes de una relación técnica.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: relation_id, component_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
relation_id	uuid	NO	PK, FK	asset_relation(id, origin_asset_id)
origin_asset_id	uuid	NO	FK	asset_relation(id, origin_asset_id); asset_component(asset_id, id)
component_id	uuid	NO	PK, FK	asset_component(asset_id, id)

FK (relation_id, origin_asset_id) → asset_relation(id, origin_asset_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

FK (origin_asset_id, component_id) → asset_component(asset_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

Índice: CREATE INDEX asset_relation_participant_component_idx ON esquema.asset_relation_participant(component_id,relation_id) · Migración 1791676800000 · línea 42

Fuentes técnicas: Migración 1791676800000 · línea 42.

asset_relation_history

Registro de asset relation history. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (relation_id, relation_version).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
relation_id	uuid	NO	FK	asset_relation(id)
relation_version	integer	NO		
action	varchar(120)	NO		
reason	varchar(1000)	SÍ		
snapshot	jsonb	NO		
actor_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
request_id	varchar(160)	SÍ		
correlation_id	varchar(160)	SÍ		
occurred_at	timestamptz	NO		

FK (relation_id) → asset_relation(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

FK (actor_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791676800000 · línea 42.

CHECK: CONSTRAINT asset_relation_history_shape_ck CHECK (relation_version>=1 AND action ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND jsonb_typeof(snapshot)='object') · Migración 1791676800000 · línea 42

Índice: CREATE INDEX asset_relation_history_page_idx ON esquema.asset_relation_history(relation_id,occurred_at DESC,id DESC) · Migración 1791676800000 · línea 42

Fuentes técnicas: Migración 1791676800000 · línea 42.

internal_staff

Persona del directorio operativo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (name_normalized); (linked_core_user_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
full_name	varchar(200)	NO		
name_normalized	text	NO		
phone	varchar(120)	SÍ		
contact_email	varchar(320)	SÍ		
notes	varchar(4000)	SÍ		
linked_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
is_active	boolean	NO		
version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_actor_type	varchar(8)	NO		
created_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
created_operation_code	varchar(120)	SÍ		
updated_actor_type	varchar(8)	NO		
updated_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
updated_operation_code	varchar(120)	SÍ		

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (linked_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..1. Migración 1791849600000 · línea 66.

CHECK: CONSTRAINT internal_staff_version_ck CHECK(version>=1) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT internal_staff_dates_ck CHECK(updated_at>=created_at) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT internal_staff_created_actor_ck CHECK((created_actor_type='user' AND created_by_core_user_id IS NOT NULL AND created_operation_code IS NULL) OR (created_actor_type='system' AND created_by_core_user_id IS NULL AND created_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND created_operation_code IS NOT NULL)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT internal_staff_updated_actor_ck CHECK((updated_actor_type='user' AND updated_by_core_user_id IS NOT NULL AND updated_operation_code IS NULL) OR (updated_actor_type='system' AND updated_by_core_user_id IS NULL AND updated_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND updated_operation_code IS NOT NULL)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT internal_staff_name_ck CHECK(name_normalized<>'') · Migración 1791849600000 · línea 66

Índice: CREATE INDEX internal_staff_list_idx ON esquema.internal_staff(is_active,name_normalized,id) · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

provider

Empresa proveedora.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (cuit).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
commercial_name	varchar(200)	NO		
name_normalized	text	NO		
legal_name	varchar(200)	SÍ		
cuit	varchar(11)	SÍ		
address	varchar(500)	SÍ		
general_phones	text[]	NO		

Campo	Tipo	NULL	Clave	Referencia física
contact_email	varchar(320)	SÍ		
notes	varchar(4000)	SÍ		
is_active	boolean	NO		
version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_actor_type	varchar(8)	NO		
created_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
created_operation_code	varchar(120)	SÍ		
updated_actor_type	varchar(8)	NO		
updated_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
updated_operation_code	varchar(120)	SÍ		

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

CHECK: CONSTRAINT provider_version_ck CHECK(version>=1) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_dates_ck CHECK(updated_at>=created_at) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_created_actor_ck CHECK((created_actor_type='user' AND created_by_core_user_id IS NOT NULL AND created_operation_code IS NULL) OR (created_actor_type='system' AND created_by_core_user_id IS NULL AND created_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND created_operation_code IS NOT NULL)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_updated_actor_ck CHECK((updated_actor_type='user' AND updated_by_core_user_id IS NOT NULL AND updated_operation_code IS NULL) OR (updated_actor_type='system' AND updated_by_core_user_id IS NULL AND updated_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND updated_operation_code IS NOT NULL)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_name_ck CHECK(name_normalized<>'') · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_cuit_ck CHECK(cuit IS NULL OR esquema.maintenance_directory_cuit_valid(cuit)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_phones_ck CHECK(array_position(general_phones,NULL) IS NULL AND array_position(general_phones,'') IS NULL) · Migración 1791849600000 · línea 66

Índice: CREATE INDEX provider_list_idx ON esquema.provider(is_active,name_normalized,id) · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

provider_contact

Contacto de empresa proveedora.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (provider_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
provider_id	uuid	NO	FK	provider(id)
identification	varchar(200)	SÍ		
description	varchar(4000)	SÍ		
phone	varchar(120)	SÍ		
email	varchar(320)	SÍ		
is_active	boolean	NO		
version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
created_actor_type	varchar(8)	NO		
created_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
created_operation_code	varchar(120)	SÍ		
updated_actor_type	varchar(8)	NO		
updated_by_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)

Campo	Tipo	NULL	Clave	Referencia física
updated_operation_code	varchar(120)	SÍ		

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (provider_id) → provider(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

CHECK: CONSTRAINT provider_contact_version_ck CHECK(version>=1) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_contact_dates_ck CHECK(updated_at>=created_at) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_contact_created_actor_ck CHECK((created_actor_type='user' AND created_by_core_user_id IS NOT NULL AND created_operation_code IS NULL) OR (created_actor_type='system' AND created_by_core_user_id IS NULL AND created_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND created_operation_code IS NOT NULL)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_contact_updated_actor_ck CHECK((updated_actor_type='user' AND updated_by_core_user_id IS NOT NULL AND updated_operation_code IS NULL) OR (updated_actor_type='system' AND updated_by_core_user_id IS NULL AND updated_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND updated_operation_code IS NOT NULL)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT provider_contact_channel_ck CHECK(NULLIF(btrim(phone),'') IS NOT NULL OR NULLIF(btrim(email),'') IS NOT NULL) · Migración 1791849600000 · línea 66

Índice: CREATE INDEX provider_contact_list_idx ON esquema.provider_contact(provider_id,is_active,id) · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

internal_staff_specialty

Registro de internal staff specialty. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: internal_staff_id, catalog_value_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
internal_staff_id	uuid	NO	PK, FK	internal_staff(id)
catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
catalog_value_id	uuid	NO	PK, FK	technical_catalog_value(catalog_key, id)

FK (internal_staff_id) → internal_staff(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (catalog_key, catalog_value_id) → technical_catalog_value(catalog_key, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

CHECK: CONSTRAINT internal_staff_specialty_catalog_ck CHECK(catalog_key='internal_staff_specialties') · Migración 1791849600000 · línea 66

Índice: CREATE INDEX internal_staff_specialty_value_idx ON esquema.internal_staff_specialty(catalog_value_id,internal_staff_id) · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

provider_trade

Registro de provider trade. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: provider_id, catalog_value_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
provider_id	uuid	NO	PK, FK	provider(id)
catalog_key	varchar(40)	NO	FK	technical_catalog_value(catalog_key, id)
catalog_value_id	uuid	NO	PK, FK	technical_catalog_value(catalog_key, id)

FK (provider_id) → provider(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (catalog_key, catalog_value_id) → technical_catalog_value(catalog_key, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

CHECK: CONSTRAINT provider_trade_catalog_ck CHECK(catalog_key='provider_trades') · Migración 1791849600000 · línea 66

Índice: CREATE INDEX provider_trade_value_idx ON esquema.provider_trade(catalog_value_id,provider_id) · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

directory_idempotency_record

Registro de directory idempotency record. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
operation	varchar(120)	NO		
idempotency_key	varchar(120)	NO		
request_hash	char(64)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
response_etag	varchar(180)	SÍ		
response_location	varchar(512)	SÍ		
completed_at	timestamptz	SÍ		
created_at	timestamptz	NO		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

CHECK: CONSTRAINT directory_idempotency_operation_ck CHECK(operation ~ '^[a-z][a-z0-9_.:-]{0,119}$') · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT directory_idempotency_key_ck CHECK(idempotency_key ~ '^[A-Za-z0-9._:-]{1,120}$') · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT directory_idempotency_hash_ck CHECK(request_hash ~ '^[0-9a-f]{64}$') · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT directory_idempotency_response_ck CHECK((completed_at IS NULL AND response_status IS NULL AND response_body IS NULL AND response_etag IS NULL AND response_location IS NULL) OR (completed_at IS NOT NULL AND completed_at>=created_at AND response_status IN (200,201) AND response_status IS NOT NULL AND jsonb_typeof(response_body)='object' AND response_body IS NOT NULL AND response_etag IS NOT NULL)) · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

directory_seed_correspondence

Registro de directory seed correspondence. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: source_namespace, source_key. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
source_namespace	varchar(120)	NO	PK	
source_key	varchar(120)	NO	PK	
entity_kind	varchar(24)	NO		
internal_staff_id	uuid	SÍ	FK	internal_staff(id)
provider_id	uuid	SÍ	FK	provider(id)
provider_contact_id	uuid	SÍ	FK	provider_contact(id)
initial_record_hash	char(64)	NO		
source_sha256	char(64)	NO		
created_at	timestamptz	NO		

FK (internal_staff_id) → internal_staff(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (provider_id) → provider(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

FK (provider_contact_id) → provider_contact(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1791849600000 · línea 66.

CHECK: CONSTRAINT directory_seed_correspondence_shape_ck CHECK((entity_kind='internal_staff' AND internal_staff_id IS NOT NULL AND provider_id IS NULL AND provider_contact_id IS NULL) OR (entity_kind='provider' AND internal_staff_id IS NULL AND provider_id IS NOT NULL AND provider_contact_id IS NULL) OR (entity_kind='provider_contact' AND internal_staff_id IS NULL AND provider_id IS NULL AND provider_contact_id IS NOT NULL)) · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT directory_seed_correspondence_keys_ck CHECK(source_namespace ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND source_key ~ '^[A-Za-z0-9._:-]{1,120}$') · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT directory_seed_correspondence_hash_ck CHECK(initial_record_hash ~ '^[0-9a-f]{64}$' AND source_sha256 ~ '^[0-9a-f]{64}$') · Migración 1791849600000 · línea 66

Índice: CREATE UNIQUE INDEX directory_seed_staff_key ON esquema.directory_seed_correspondence(source_namespace,internal_staff_id) WHERE internal_staff_id IS NOT NULL · Migración 1791849600000 · línea 66

Índice: CREATE UNIQUE INDEX directory_seed_provider_key ON esquema.directory_seed_correspondence(source_namespace,provider_id) WHERE provider_id IS NOT NULL · Migración 1791849600000 · línea 66

Índice: CREATE UNIQUE INDEX directory_seed_contact_key ON esquema.directory_seed_correspondence(source_namespace,provider_contact_id) WHERE provider_contact_id IS NOT NULL · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

directory_seed_application

Registro de directory seed application. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (source_namespace, manifest_hash).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
source_namespace	varchar(120)	NO		
manifest_hash	char(64)	NO		
dataset_hash	char(64)	NO		
approved_plan_hash	char(64)	NO		
operation_code	varchar(120)	NO		
inserted_staff	integer	NO		
inserted_providers	integer	NO		
inserted_contacts	integer	NO		
occurred_at	timestamptz	NO		

CHECK: CONSTRAINT directory_seed_application_hash_ck CHECK(manifest_hash ~ '^[0-9a-f]{64}$' AND dataset_hash ~ '^[0-9a-f]{64}$' AND approved_plan_hash ~ '^[0-9a-f]{64}$') · Migración 1791849600000 · línea 66

CHECK: CONSTRAINT directory_seed_application_shape_ck CHECK(source_namespace ~ '^[a-z][a-z0-9_.:-]{0,119}$' AND operation_code='maintenance-directory-seed-v1' AND inserted_staff>=0 AND inserted_providers>=0 AND inserted_contacts>=0) · Migración 1791849600000 · línea 66

Fuentes técnicas: Migración 1791849600000 · línea 66.

maintenance_document

Documento asociado a propietario operativo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (owner_asset_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK, FK	maintenance_document_revision(document_id, id)
owner_kind	varchar(16)	NO		
owner_asset_id	uuid	SÍ	FK	asset(id)
owner_provider_id	uuid	SÍ	FK	provider(id)
owner_staff_id	uuid	SÍ	FK	internal_staff(id)
title	varchar(200)	NO		
notes	varchar(4000)	SÍ		
version	integer	NO		
current_revision_id	uuid	NO	FK	maintenance_document_revision(document_id, id)
is_retired	boolean	NO		
retired_at	timestamptz	SÍ		
retire_reason	varchar(1000)	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

Campo	Tipo	NULL	Clave	Referencia física
created_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
updated_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
owner_notice_id	uuid	SÍ	FK	maintenance_notice(id)
owner_intervention_id	uuid	SÍ	FK	work_order_intervention(id)

FK (owner_asset_id) → asset(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (owner_provider_id) → provider(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (owner_staff_id) → internal_staff(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (id, current_revision_id) → maintenance_document_revision(document_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (owner_notice_id) → maintenance_notice(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (owner_intervention_id) → work_order_intervention(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1794441600000 · línea 378.

CHECK: CHECK(updated_at>=created_at) · Migración 1792022400000 · línea 48

CHECK: CHECK((is_retired AND retired_at IS NOT NULL AND retire_reason IS NOT NULL AND btrim(retire_reason)<>'') OR (NOT is_retired AND retired_at IS NULL AND retire_reason IS NULL)) · Migración 1792022400000 · línea 48

CHECK: CONSTRAINT maintenance_document_owner_ck CHECK((owner_kind='asset' AND owner_asset_id IS NOT NULL AND num_nonnulls(owner_provider_id,owner_staff_id,owner_notice_id,owner_intervention_id)=0) OR (owner_kind='provider' AND owner_provider_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_staff_id,owner_notice_id,owner_intervention_id)=0) OR (owner_kind='staff' AND owner_staff_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_provider_id,owner_notice_id,owner_intervention_id)=0) OR (owner_kind='notice' AND owner_notice_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_provider_id,owner_staff_id,owner_intervention_id)=0) OR (owner_kind='intervention' AND owner_intervention_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_provider_id,owner_staff_id,owner_notice_id)=0)) · Migración 1794441600000 · línea 382

Índice: CREATE INDEX maintenance_document_asset_list_idx ON esquema.maintenance_document(owner_asset_id,is_retired,created_at DESC,id DESC) WHERE owner_asset_id IS NOT NULL · Migración 1792022400000 · línea 48

Índice: CREATE INDEX maintenance_document_provider_list_idx ON esquema.maintenance_document(owner_provider_id,is_retired,created_at DESC,id DESC) WHERE owner_provider_id IS NOT NULL · Migración 1792022400000 · línea 48

Índice: CREATE INDEX maintenance_document_staff_list_idx ON esquema.maintenance_document(owner_staff_id,is_retired,created_at DESC,id DESC) WHERE owner_staff_id IS NOT NULL · Migración 1792022400000 · línea 48

Índice: CREATE INDEX maintenance_document_notice_list_idx ON esquema.maintenance_document(owner_notice_id,is_retired,created_at DESC,id DESC) WHERE owner_notice_id IS NOT NULL · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_document_intervention_list_idx ON esquema.maintenance_document(owner_intervention_id,is_retired,created_at DESC,id DESC) WHERE owner_intervention_id IS NOT NULL · Migración 1794441600000 · línea 390

Fuentes técnicas: Migración 1792022400000 · línea 48; Migración 1793145600000 · línea 14; Migración 1794441600000 · línea 378.

maintenance_document_file

Archivo físico y propiedades documentales.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
upload_id	uuid	NO	FK	maintenance_document_upload(id)
storage_instance_id	uuid	NO		
storage_key	uuid	NO		
original_name	varchar(255)	NO		
declared_mime	varchar(120)	SÍ		
expected_size	integer	NO		
expected_sha256	char(64)	NO		
actual_size	integer	SÍ		
sha256	char(64)	SÍ		
format	varchar(4)	SÍ		
mime	varchar(32)	SÍ		
state	varchar(16)	NO		

Campo	Tipo	NULL	Clave	Referencia física
created_at	timestamptz	NO		
last_activity_at	timestamptz	NO		
created_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (upload_id) → maintenance_document_upload(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

CHECK: CHECK(last_activity_at>=created_at) · Migración 1792022400000 · línea 48

CHECK: CHECK((actual_size IS NULL AND sha256 IS NULL AND format IS NULL AND mime IS NULL) OR (actual_size=expected_size AND sha256=expected_sha256 AND actual_size IS NOT NULL AND sha256 IS NOT NULL AND format IS NOT NULL AND mime IS NOT NULL AND ((format='jpeg' AND mime='image/jpeg') OR (format='png' AND mime='image/png') OR (format='webp' AND mime='image/webp') OR (format='pdf' AND mime='application/pdf')))) · Migración 1792022400000 · línea 48

CHECK: CHECK(state NOT IN ('ready','published','confirmed') OR (actual_size IS NOT NULL AND sha256 IS NOT NULL AND format IS NOT NULL AND mime IS NOT NULL)) · Migración 1792022400000 · línea 48

Índice: CREATE INDEX maintenance_document_file_cleanup_idx ON esquema.maintenance_document_file(state,last_activity_at,id) WHERE state<>'confirmed' · Migración 1792022400000 · línea 48

Índice: CREATE INDEX maintenance_document_file_upload_idx ON esquema.maintenance_document_file(upload_id,created_at,id) · Migración 1792022400000 · línea 48

Fuentes técnicas: Migración 1792022400000 · línea 48.

maintenance_document_revision

Versión documental.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (file_id); (document_id, id); (document_id, revision_number).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
document_id	uuid	NO	FK	maintenance_document(id)
revision_number	integer	NO		
file_id	uuid	NO	FK	maintenance_document_file(id)

Campo	Tipo	NULL	Clave	Referencia física
expires_on	date	SÍ		
created_at	timestamptz	NO		
created_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (document_id) → maintenance_document(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (file_id) → maintenance_document_file(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1792022400000 · línea 48.

FK (created_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

Índice: CREATE INDEX maintenance_document_revision_expiry_idx ON esquema.maintenance_document_revision(expires_on,document_id) WHERE expires_on IS NOT NULL · Migración 1792022400000 · línea 48

Fuentes técnicas: Migración 1792022400000 · línea 48.

maintenance_document_operation

Registro de document operation. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
operation	varchar(120)	NO		
idempotency_key	uuid	NO		
request_hash	char(64)	NO		
request_body	jsonb	NO		
owner_kind	varchar(16)	NO		
owner_asset_id	uuid	SÍ	FK	asset(id)

Campo	Tipo	NULL	Clave	Referencia física
owner_provider_id	uuid	SÍ	FK	provider(id)
owner_staff_id	uuid	SÍ	FK	internal_staff(id)
document_id	uuid	SÍ	FK	maintenance_document(id)
state	varchar(16)	NO		
phase	varchar(24)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
completed_at	timestamptz	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
owner_notice_id	uuid	SÍ	FK	maintenance_notice(id)
notice_initial_file_id	uuid	SÍ	FK	maintenance_notice_initial_file(id)
notice_resolution_operation_id	uuid	SÍ	FK	notice_evidence_resolution_operation(id)
owner_intervention_id	uuid	SÍ	FK	work_order_intervention(id)

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (owner_asset_id) → asset(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (owner_provider_id) → provider(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (owner_staff_id) → internal_staff(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (document_id) → maintenance_document(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (owner_notice_id) → maintenance_notice(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (notice_initial_file_id) → maintenance_notice_initial_file(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (notice_resolution_operation_id) → notice_evidence_resolution_operation(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (owner_intervention_id) → work_order_intervention(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1794441600000 · línea 378.

CHECK: CHECK(updated_at>=created_at) · Migración 1792022400000 · línea 48

CHECK: CHECK((state='pending' AND phase<>'terminal' AND completed_at IS NULL AND response_status IS NULL AND response_body IS NULL) OR (state IN ('applied','not_applied') AND phase='terminal' AND completed_at IS NOT NULL AND completed_at>=created_at AND response_status BETWEEN 200 AND 599 AND response_status IS NOT NULL AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object')) · Migración 1792022400000 · línea 48

CHECK: CONSTRAINT maintenance_document_operation_initial_owner_ck CHECK(notice_initial_file_id IS NULL OR (owner_kind='notice' AND operation='document.created')) · Migración 1793145600000 · línea 14

CHECK: CONSTRAINT maintenance_document_resolution_owner_ck CHECK(notice_resolution_operation_id IS NULL OR (owner_kind='notice' AND operation='document.created')) · Migración 1793232000000 · línea 13

CHECK: CONSTRAINT maintenance_document_operation_owner_ck CHECK((owner_kind='asset' AND owner_asset_id IS NOT NULL AND num_nonnulls(owner_provider_id,owner_staff_id,owner_notice_id,owner_intervention_id)=0) OR (owner_kind='provider' AND owner_provider_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_staff_id,owner_notice_id,owner_intervention_id)=0) OR (owner_kind='staff' AND owner_staff_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_provider_id,owner_notice_id,owner_intervention_id)=0) OR (owner_kind='notice' AND owner_notice_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_provider_id,owner_staff_id,owner_intervention_id)=0) OR (owner_kind='intervention' AND owner_intervention_id IS NOT NULL AND num_nonnulls(owner_asset_id,owner_provider_id,owner_staff_id,owner_notice_id)=0)) · Migración 1794441600000 · línea 382

Índice: CREATE INDEX maintenance_document_operation_owner_idx ON esquema.maintenance_document_operation(owner_kind,owner_asset_id,owner_provider_id,owner_staff_id,state,id) · Migración 1792022400000 · línea 48

Índice: CREATE INDEX maintenance_document_operation_notice_idx ON esquema.maintenance_document_operation(owner_notice_id,state,id) WHERE owner_notice_id IS NOT NULL · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_document_operation_intervention_idx ON esquema.maintenance_document_operation(owner_intervention_id,state,id) WHERE owner_intervention_id IS NOT NULL · Migración 1794441600000 · línea 390

Fuentes técnicas: Migración 1792022400000 · línea 48; Migración 1793145600000 · línea 14; Migración 1793232000000 · línea 13; Migración 1794441600000 · línea 378.

maintenance_document_upload

Operación de carga documental.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (operation_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
operation_id	uuid	NO	FK	maintenance_document_operation(id)

Campo	Tipo	NULL	Clave	Referencia física
active_file_id	uuid	SÍ	FK	maintenance_document_file(id)
target_document_id	uuid	SÍ	FK	maintenance_document(id)
planned_document_id	uuid	NO		
planned_revision_id	uuid	NO		
state	varchar(16)	NO		
created_at	timestamptz	NO		
last_activity_at	timestamptz	NO		

FK (operation_id) → maintenance_document_operation(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1792022400000 · línea 48.

FK (active_file_id) → maintenance_document_file(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (target_document_id) → maintenance_document(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

CHECK: CHECK(last_activity_at>=created_at) · Migración 1792022400000 · línea 48

CHECK: CHECK(state IN ('prepared','closed') OR active_file_id IS NOT NULL) · Migración 1792022400000 · línea 48

Índice: CREATE INDEX maintenance_document_upload_cleanup_idx ON esquema.maintenance_document_upload(state,last_activity_at,id) · Migración 1792022400000 · línea 48

Fuentes técnicas: Migración 1792022400000 · línea 48.

maintenance_asset_document_cover

Portada documental de activo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: asset_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
asset_id	uuid	NO	PK, FK	asset(id); maintenance_document(owner_asset_id, id)
document_id	uuid	SÍ	FK	maintenance_document(owner_asset_id, id); maintenance_document_revision(document_id, id)

Campo	Tipo	NULL	Clave	Referencia física
revision_id	uuid	SÍ	FK	maintenance_document_revision(document_id, id)
updated_at	timestamptz	NO		
updated_by_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

FK (asset_id) → asset(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1792022400000 · línea 48.

FK (updated_by_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (asset_id, document_id) → maintenance_document(owner_asset_id, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

FK (document_id, revision_id) → maintenance_document_revision(document_id, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792022400000 · línea 48.

CHECK: CHECK((document_id IS NULL)=(revision_id IS NULL)) · Migración 1792022400000 · línea 48

Fuentes técnicas: Migración 1792022400000 · línea 48.

work_order_annual_counter

Registro de work order annual counter. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: emission_year. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
emission_year	integer	NO	PK	
last_number	bigint	NO		

Fuentes técnicas: Migración 1792108800000 · línea 55.

work_order

Orden de trabajo emitida.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (number); (emission_year, annual_number).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
emission_year	integer	NO	FK	work_order_annual_counter(emission_year)
annual_number	bigint	NO		
number	text	SÍ		
version	integer	NO		
title	varchar(200)	NO		
description	varchar(4000)	SÍ		
kind	varchar(16)	NO		
origin_kind	varchar(32)	NO		
origin_context	jsonb	NO		
goal_asset_id	uuid	SÍ	FK	asset(id)
goal_space_id	uuid	SÍ	FK	core_space_projection(core_space_id)
space_snapshot	jsonb	SÍ		
goal_space_path_ids	jsonb	SÍ		
priority_id	uuid	NO	FK	technical_catalog_value(catalog_key, id)
priority_catalog_key	varchar(80)	NO	FK	technical_catalog_value(catalog_key, id)
priority_snapshot	jsonb	NO		

Campo	Tipo	NULL	Clave	Referencia física
scheduled_for	timestamptz	SÍ		
deadline_at	timestamptz	SÍ		
estimated_minutes	integer	SÍ		
planning_notes	varchar(4000)	SÍ		
state	varchar(24)	NO		
technical_dataset_key	varchar(120)	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
actor_type	varchar(8)	NO		
actor_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
actor_operation_code	varchar(120)	SÍ		
creation_transaction_id	xid8	NO		

FK (emission_year) → work_order_annual_counter(emission_year). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (goal_asset_id) → asset(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (goal_space_id) → core_space_projection(core_space_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (actor_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (priority_catalog_key, priority_id) → technical_catalog_value(catalog_key, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

CHECK: CONSTRAINT work_order_goal_ck CHECK (num_nonnulls(goal_asset_id,goal_space_id)=1) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_space_snapshot_ck CHECK ((goal_space_id IS NULL AND space_snapshot IS NULL) OR (goal_space_id IS NOT NULL AND esquema.maintenance_work_order_snapshot_valid('location',space_snapshot) AND space_snapshot->>'id'=goal_space_id::text)) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_priority_snapshot_ck CHECK (esquema.maintenance_work_order_snapshot_valid('priority',priority_snapshot) AND priority_snapshot->>'id'=priority_id::text) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_origin_ck CHECK (esquema.maintenance_work_order_snapshot_valid('origin.'||origin_kind,origin_context) AND (origin_kind='manual' OR (kind='preventive' AND origin_kind IN ('preventive_periodic','preventive_event')) OR (kind='corrective' AND origin_kind IN ('corrective_reports','corrective_finding')))) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_emission_year_ck CHECK (emission_year=extract(year FROM created_at AT TIME ZONE 'America/Argentina/Buenos_Aires')) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_dates_ck CHECK (updated_at>=created_at) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_actor_ck CHECK ((actor_type='user' AND actor_core_user_id IS NOT NULL AND actor_operation_code IS NULL) OR (actor_type='system' AND actor_core_user_id IS NULL AND actor_operation_code IS NOT NULL AND actor_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$')) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_dataset_ck CHECK (technical_dataset_key IS NULL OR (actor_type='system' AND technical_dataset_key ~ '^[a-z][a-z0-9_.:-]{0,119}$')) · Migración 1792108800000 · línea 55

Índice: CREATE INDEX work_order_created_idx ON esquema.work_order(created_at DESC,id DESC) · Migración 1792108800000 · línea 55

Índice: CREATE INDEX work_order_goal_asset_idx ON esquema.work_order(goal_asset_id) · Migración 1792108800000 · línea 55

Índice: CREATE INDEX work_order_space_path_idx ON esquema.work_order USING gin(goal_space_path_ids) · Migración 1792108800000 · línea 55

Fuentes técnicas: Migración 1792108800000 · línea 55; Migración 1793577600000 · línea 5.

work_order_asset_scope

Alcance de activos congelado para una OT.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: work_order_id, asset_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
work_order_id	uuid	NO	PK, FK	work_order(id)
asset_id	uuid	NO	PK, FK	asset(id)
role	varchar(12)	NO		
asset_snapshot	jsonb	NO		
location_path_ids	jsonb	SÍ		
relation_id	uuid	SÍ	FK	asset_relation(id)
relation_snapshot	jsonb	SÍ		
retired_at	timestamptz	SÍ		

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (asset_id) → asset(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (relation_id) → asset_relation(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

CHECK: CONSTRAINT work_order_scope_asset_snapshot_ck CHECK (esquema.maintenance_work_order_snapshot_valid('asset',asset_snapshot) AND asset_snapshot->>'id'=asset_id::text) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_scope_relation_ck CHECK ((role='principal' AND relation_id IS NULL AND relation_snapshot IS NULL) OR (role='secondary' AND relation_id IS NOT NULL AND esquema.maintenance_work_order_snapshot_valid('relation',relation_snapshot) AND relation_snapshot->>'id'=relation_id::text)) · Migración 1792108800000 · línea 55

Índice: CREATE INDEX work_order_scope_asset_idx ON esquema.work_order_asset_scope(asset_id,work_order_id) · Migración 1792108800000 · línea 55

Índice: CREATE INDEX work_order_scope_path_idx ON esquema.work_order_asset_scope USING gin(location_path_ids) · Migración 1792108800000 · línea 55

Fuentes técnicas: Migración 1792108800000 · línea 55; Migración 1793577600000 · línea 5.

work_order_task

Tarea perteneciente a una OT.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (work_order_id, position); (work_order_id, id); (work_order_id, active_position).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
work_order_id	uuid	NO	FK	work_order(id); work_order_asset_scope(work_order_id, asset_id)
position	integer	NO		
title	varchar(200)	NO		
description	varchar(4000)	SÍ		
asset_id	uuid	SÍ	FK	work_order_asset_scope(work_order_id, asset_id); asset_component(asset_id, id)
component_id	uuid	SÍ	FK	asset_component(asset_id, id)
component_snapshot	jsonb	SÍ		
space_actions	jsonb	NO		

Campo	Tipo	NULL	Clave	Referencia física
template_snapshot	jsonb	SÍ		
retired_at	timestamptz	SÍ		
active_position	integer	SÍ		

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (work_order_id, asset_id) → work_order_asset_scope(work_order_id, asset_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (asset_id, component_id) → asset_component(asset_id, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

CHECK: CONSTRAINT work_order_task_component_ck CHECK ((component_id IS NULL AND component_snapshot IS NULL) OR (asset_id IS NOT NULL AND component_id IS NOT NULL AND esquema.maintenance_work_order_snapshot_valid('component',component_snapshot) AND component_snapshot->>'id'=component_id::text AND component_snapshot->>'asset_id'=asset_id::text)) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_task_actions_ck CHECK (jsonb_typeof(space_actions)='array' AND esquema.maintenance_work_order_snapshot_valid('actions',jsonb_build_object('schema_version',1,'items',space_actions)) AND ((asset_id IS NOT NULL AND space_actions='[]') OR (asset_id IS NULL AND jsonb_array_length(space_actions)>0))) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_task_template_ck CHECK (template_snapshot IS NULL OR esquema.maintenance_work_order_snapshot_valid('template',template_snapshot)) · Migración 1792108800000 · línea 55

Fuentes técnicas: Migración 1792108800000 · línea 55; Migración 1793577600000 · línea 5.

work_order_task_measurement

Registro de work order task measurement. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (task_id, position); (task_id, active_position); (work_order_id, task_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
work_order_id	uuid	NO	FK	work_order_task(work_order_id, id)
task_id	uuid	NO	FK	work_order_task(work_order_id, id)
position	integer	NO		

Campo	Tipo	NULL	Clave	Referencia física
label	varchar(200)	NO		
unit_id	uuid	NO	FK	technical_catalog_value(catalog_key, id)
unit_catalog_key	varchar(80)	NO	FK	technical_catalog_value(catalog_key, id)
unit_snapshot	jsonb	NO		
is_required	boolean	NO		
suggested_decimals	integer	NO		
warning_min	numeric	SÍ		
warning_max	numeric	SÍ		
retired_at	timestamptz	SÍ		
active_position	integer	SÍ		

FK (work_order_id, task_id) → work_order_task(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (unit_catalog_key, unit_id) → technical_catalog_value(catalog_key, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

CHECK: CONSTRAINT work_order_measurement_unit_ck CHECK (esquema.maintenance_work_order_snapshot_valid('unit',unit_snapshot) AND unit_snapshot->>'id'=unit_id::text) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_measurement_bounds_ck CHECK ((warning_min IS NULL OR warning_min::text NOT IN ('NaN','Infinity','-Infinity')) AND (warning_max IS NULL OR warning_max::text NOT IN ('NaN','Infinity','-Infinity')) AND (warning_min IS NULL OR warning_max IS NULL OR warning_min<=warning_max)) · Migración 1792108800000 · línea 55

Fuentes técnicas: Migración 1792108800000 · línea 55; Migración 1793577600000 · línea 5.

work_order_assignment

Asignación operativa de OT.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (work_order_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
work_order_id	uuid	NO	FK	work_order(id)
kind	varchar(8)	NO		
staff_id	uuid	SÍ	FK	internal_staff(id)
provider_id	uuid	SÍ	FK	provider(id); provider_contact(provider_id, id)
contact_id	uuid	SÍ	FK	provider_contact(provider_id, id)
subject_snapshot	jsonb	NO		
contact_snapshot	jsonb	SÍ		
incorporated_at	timestamptz	NO		
removed_at	timestamptz	SÍ		

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (staff_id) → internal_staff(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (provider_id) → provider(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (provider_id, contact_id) → provider_contact(provider_id, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

CHECK: CONSTRAINT work_order_assignment_subject_ck CHECK ((kind='staff' AND staff_id IS NOT NULL AND provider_id IS NULL AND contact_id IS NULL) OR (kind='provider' AND staff_id IS NULL AND provider_id IS NOT NULL)) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_assignment_snapshot_ck CHECK (esquema.maintenance_work_order_snapshot_valid('assignment',subject_snapshot) AND subject_snapshot->>'id'=COALESCE(staff_id,provider_id)::text) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_assignment_contact_ck CHECK ((contact_id IS NULL AND contact_snapshot IS NULL) OR (contact_id IS NOT NULL AND esquema.maintenance_work_order_snapshot_valid('contact',contact_snapshot) AND contact_snapshot->>'id'=contact_id::text AND contact_snapshot->>'provider_id'=provider_id::text)) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_assignment_dates_ck CHECK (removed_at IS NULL OR removed_at>=incorporated_at) · Migración 1792108800000 · línea 55

Índice: CREATE UNIQUE INDEX work_order_assignment_staff_key ON esquema.work_order_assignment(work_order_id,staff_id) WHERE removed_at IS NULL AND staff_id IS NOT NULL · Migración 1792108800000 · línea 55

Índice: CREATE UNIQUE INDEX work_order_assignment_provider_key ON esquema.work_order_assignment(work_order_id,provider_id) WHERE removed_at IS NULL AND provider_id IS NOT NULL · Migración 1792108800000 · línea 55

Índice: CREATE INDEX work_order_assignment_subject_idx ON esquema.work_order_assignment(staff_id,provider_id,work_order_id) · Migración 1792108800000 · línea 55

Fuentes técnicas: Migración 1792108800000 · línea 55.

work_order_operation

Registro de work order operation. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
actor_type	varchar(8)	NO		
principal_core_user_id	uuid	SÍ	FK	maintenance_member(core_user_id)
actor_operation_code	varchar(120)	SÍ		
operation	varchar(120)	NO		
idempotency_key	uuid	NO		
request_hash	varchar(64)	NO		
state	varchar(12)	NO		
work_order_id	uuid	SÍ	FK	work_order(id)

Campo	Tipo	NULL	Clave	Referencia física
response_body	jsonb	SÍ		
created_at	timestamptz	NO		
completed_at	timestamptz	SÍ		
assignment_permission_required	boolean	NO		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

FK (work_order_id) → work_order(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1792108800000 · línea 55.

CHECK: CONSTRAINT work_order_operation_actor_ck CHECK ((actor_type='user' AND principal_core_user_id IS NOT NULL AND actor_operation_code IS NULL) OR (actor_type='system' AND principal_core_user_id IS NULL AND actor_operation_code IS NOT NULL AND actor_operation_code ~ '^[a-z][a-z0-9_.:-]{0,119}$')) · Migración 1792108800000 · línea 55

CHECK: CONSTRAINT work_order_operation_terminal_ck CHECK ((state='pending' AND completed_at IS NULL AND response_body IS NULL AND work_order_id IS NULL) OR (state='applied' AND completed_at IS NOT NULL AND completed_at>=created_at AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND work_order_id IS NOT NULL) OR (state='not_applied' AND completed_at IS NOT NULL AND completed_at>=created_at AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND work_order_id IS NULL)) · Migración 1792108800000 · línea 55

Índice: CREATE UNIQUE INDEX work_order_operation_user_key ON esquema.work_order_operation(principal_core_user_id,operation,idempotency_key) WHERE actor_type='user' · Migración 1792108800000 · línea 55

Índice: CREATE UNIQUE INDEX work_order_operation_system_key ON esquema.work_order_operation(actor_operation_code,operation,idempotency_key) WHERE actor_type='system' · Migración 1792108800000 · línea 55

Índice: CREATE UNIQUE INDEX work_order_operation_result_key ON esquema.work_order_operation(work_order_id) WHERE state='applied' · Migración 1792108800000 · línea 55

Fuentes técnicas: Migración 1792108800000 · línea 55; Migración 1793750400000 · línea 13.

work_order_draft

Preparación persistida previa a emisión.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (work_order_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	

Campo	Tipo	NULL	Clave	Referencia física
version	integer	NO		
state	varchar(12)	NO		
title	varchar(200)	NO		
payload	jsonb	NO		
created_by	uuid	NO	FK	maintenance_member(core_user_id)
created_by_snapshot	jsonb	NO		
updated_by	uuid	NO	FK	maintenance_member(core_user_id)
updated_by_snapshot	jsonb	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
completed_at	timestamptz	SÍ		
work_order_id	uuid	SÍ	FK	work_order(id)

FK (created_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793059200000 · línea 17.

FK (updated_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793059200000 · línea 17.

FK (work_order_id) → work_order(id). Padres por hija: 0..1; hijas por padre: 0..1. Migración 1793059200000 · línea 17.

CHECK: CONSTRAINT work_order_draft_actor_ck CHECK(COALESCE(jsonb_typeof(created_by_snapshot)='object' AND created_by_snapshot->>'id'=created_by::text AND jsonb_typeof(updated_by_snapshot)='object' AND updated_by_snapshot->>'id'=updated_by::text,false)) · Migración 1793059200000 · línea 17

CHECK: CONSTRAINT work_order_draft_dates_ck CHECK(updated_at>=created_at AND (completed_at IS NULL OR completed_at>=updated_at)) · Migración 1793059200000 · línea 17

CHECK: CONSTRAINT work_order_draft_terminal_ck CHECK((state='draft' AND completed_at IS NULL AND work_order_id IS NULL) OR (state='discarded' AND completed_at IS NOT NULL AND work_order_id IS NULL) OR (state='emitted' AND completed_at IS NOT NULL AND work_order_id IS NOT NULL)) · Migración 1793059200000 · línea 17

CHECK: CONSTRAINT work_order_draft_payload_ck CHECK(COALESCE(jsonb_typeof(payload)='object' AND payload->>'title'=title AND octet_length(payload::text)<=524288 AND (payload->'schema_version'='1'::jsonb OR (payload->'schema_version'='2'::jsonb AND payload->'origin'->>'kind'='notices' AND jsonb_typeof(payload->'origin'->'notices')='array' AND jsonb_array_length(payload->'origin'->'notices') BETWEEN 1 AND 100)),false)) · Migración 1793232000000 · línea 13

Índice: CREATE INDEX work_order_draft_pending_idx ON esquema.work_order_draft(updated_at DESC,id DESC) WHERE state='draft' · Migración 1793059200000 · línea 17

Índice: CREATE INDEX work_order_draft_author_idx ON esquema.work_order_draft(created_by,id) · Migración 1793059200000 · línea 17

Índice: CREATE INDEX work_order_draft_notices_idx ON esquema.work_order_draft USING gin((payload->'origin'->'notices') jsonb_path_ops) WHERE state='draft' AND payload->'schema_version'='2'::jsonb · Migración 1793232000000 · línea 13

Fuentes técnicas: Migración 1793059200000 · línea 17.

work_order_draft_operation

Registro de work order draft operation. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
operation	varchar(32)	NO		
idempotency_key	uuid	NO		
request_hash	varchar(64)	NO		
state	varchar(12)	NO		
draft_id	uuid	SÍ	FK	work_order_draft(id)
response_status	integer	SÍ		
response_body	jsonb	SÍ		
response_etag	varchar(180)	SÍ		
created_at	timestamptz	NO		
completed_at	timestamptz	SÍ		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793059200000 · línea 17.

FK (draft_id) → work_order_draft(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793059200000 · línea 17.

CHECK: CONSTRAINT work_order_draft_operation_terminal_ck CHECK((state='pending' AND completed_at IS NULL AND response_status IS NULL AND response_body IS NULL AND response_etag IS NULL AND draft_id IS NULL) OR (state IN ('applied','not_applied') AND completed_at IS NOT NULL AND completed_at>=created_at AND response_status IS NOT NULL AND response_status BETWEEN 200 AND 599 AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND (state<>'applied' OR draft_id IS NOT NULL))) · Migración 1793059200000 · línea 17

Índice: CREATE INDEX work_order_draft_operation_target_idx ON esquema.work_order_draft_operation(draft_id) · Migración 1793059200000 · línea 17

Fuentes técnicas: Migración 1793059200000 · línea 17.

work_order_planning_precision

Registro de work order planning precision. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: work_order_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
work_order_id	uuid	NO	PK, FK	work_order(id)
scheduled_precision	varchar(8)	SÍ		
deadline_precision	varchar(8)	SÍ		

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1793059200000 · línea 17.

Fuentes técnicas: Migración 1793059200000 · línea 17.

maintenance_notice_counter

Registro de notice counter. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: year. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
year	integer	NO	PK	
last_value	bigint	NO		

Fuentes técnicas: Migración 1793145600000 · línea 14.

maintenance_notice

Aviso operativo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (number); (year, annual_number).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
version	integer	NO		
number	varchar(40)	NO		
year	integer	NO	FK	maintenance_notice_counter(year)
annual_number	bigint	NO		
state	varchar(16)	NO		
title	varchar(200)	NO		
description	varchar(4000)	NO		
space_id	uuid	NO	FK	core_space_projection(core_space_id)
space_snapshot	jsonb	NO		
asset_id	uuid	SÍ	FK	asset(id)
asset_snapshot	jsonb	SÍ		
priority_id	uuid	SÍ	FK	technical_catalog_value(id)
priority_snapshot	jsonb	SÍ		
created_by	uuid	NO	FK	maintenance_member(core_user_id)
created_by_snapshot	jsonb	NO		

Campo	Tipo	NULL	Clave	Referencia física
created_at	timestamptz	NO		
updated_at	timestamptz	NO		
updated_by	uuid	NO	FK	maintenance_member(core_user_id)

FK (year) → maintenance_notice_counter(year). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (asset_id) → asset(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (priority_id) → technical_catalog_value(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (created_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (updated_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

CHECK: CONSTRAINT maintenance_notice_number_ck CHECK(number='AV-'||year::text||'-'||lpad(annual_number::text,GREATEST(4,length(annual_number::text)),'0')) · Migración 1793145600000 · línea 14

CHECK: CONSTRAINT maintenance_notice_moment_ck CHECK(updated_at>=created_at AND extract(year FROM created_at AT TIME ZONE 'America/Argentina/Buenos_Aires')::int=year) · Migración 1793145600000 · línea 14

CHECK: CONSTRAINT maintenance_notice_snapshots_ck CHECK(COALESCE(space_snapshot->'schema_version'='1'::jsonb AND space_snapshot->>'id'=space_id::text AND jsonb_typeof(space_snapshot->'path_ids')='array' AND jsonb_array_length(space_snapshot->'path_ids')>0 AND space_snapshot->'path_ids'->>-1=space_id::text AND created_by_snapshot->>'id'=created_by::text AND ((asset_id IS NULL AND asset_snapshot IS NULL) OR (asset_id IS NOT NULL AND asset_snapshot->'schema_version'='1'::jsonb AND asset_snapshot->>'id'=asset_id::text)) AND ((priority_id IS NULL AND priority_snapshot IS NULL) OR (priority_id IS NOT NULL AND priority_snapshot->'schema_version'='1'::jsonb AND priority_snapshot->>'id'=priority_id::text)),false)) · Migración 1793145600000 · línea 14

CHECK: CONSTRAINT maintenance_notice_state_check CHECK(state IN ('pending','dismissed','converted')) · Migración 1793232000000 · línea 13

Índice: CREATE INDEX maintenance_notice_created_idx ON esquema.maintenance_notice(created_at DESC,id DESC) · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_notice_space_path_idx ON esquema.maintenance_notice USING gin((space_snapshot->'path_ids')) · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_notice_asset_idx ON esquema.maintenance_notice(asset_id,created_at DESC,id DESC) WHERE asset_id IS NOT NULL · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_notice_author_idx ON esquema.maintenance_notice(created_by,created_at DESC,id DESC) · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_notice_state_idx ON esquema.maintenance_notice(state,created_at DESC,id DESC) · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_notice_priority_idx ON esquema.maintenance_notice(priority_id,created_at DESC,id DESC) · Migración 1793145600000 · línea 14

Fuentes técnicas: Migración 1793145600000 · línea 14.

maintenance_notice_operation

Registro de notice operation. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
operation	varchar(24)	NO		
idempotency_key	uuid	NO		
request_hash	char(64)	NO		
request_body	jsonb	NO		
notice_id	uuid	SÍ	FK	maintenance_notice(id)
state	varchar(12)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
response_etag	varchar(180)	SÍ		
created_at	timestamptz	NO		
completed_at	timestamptz	SÍ		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (notice_id) → maintenance_notice(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

CHECK: CONSTRAINT maintenance_notice_operation_terminal_ck CHECK((state='pending' AND completed_at IS NULL AND response_status IS NULL AND response_body IS NULL AND response_etag IS NULL) OR (state IN ('applied','not_applied') AND completed_at IS NOT NULL AND completed_at>=created_at AND response_status IS NOT NULL AND response_status BETWEEN 200 AND 599 AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND (state<>'applied' OR notice_id IS NOT NULL))) · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_notice_operation_target_idx ON esquema.maintenance_notice_operation(notice_id) · Migración 1793145600000 · línea 14

Fuentes técnicas: Migración 1793145600000 · línea 14.

maintenance_notice_initial_file

Registro de notice initial file. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (upload_key); (completed_document_id); (notice_id, position); (discard_resolution_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
notice_id	uuid	NO	FK	maintenance_notice(id)
position	smallint	NO		
upload_key	uuid	NO		
create_operation_id	uuid	NO	FK	maintenance_notice_operation(id)
descriptor	jsonb	NO		
completed_document_id	uuid	SÍ	FK	maintenance_document(id)
discarded_at	timestamptz	SÍ		
discarded_by	uuid	SÍ	FK	maintenance_member(core_user_id)
discard_resolution_id	uuid	SÍ	FK	notice_evidence_resolution_operation(id)

FK (notice_id) → maintenance_notice(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (create_operation_id) → maintenance_notice_operation(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (completed_document_id) → maintenance_document(id). Padres por hija: 0..1; hijas por padre: 0..1. Migración 1793145600000 · línea 14.

FK (discarded_by) → maintenance_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (discard_resolution_id) → notice_evidence_resolution_operation(id). Padres por hija: 0..1; hijas por padre: 0..1. Migración 1793232000000 · línea 13.

CHECK: CONSTRAINT maintenance_notice_initial_descriptor_ck CHECK(COALESCE(jsonb_typeof(descriptor)='object' AND jsonb_typeof(descriptor->'original_name')='string' AND length(descriptor->>'original_name') BETWEEN 1 AND 255 AND jsonb_typeof(descriptor->'size')='number' AND (descriptor->>'size')::numeric BETWEEN 1 AND 26214400 AND descriptor->>'sha256' ~ '^[0-9a-f]{64}$',false)) · Migración 1793145600000 · línea 14

CHECK: CONSTRAINT maintenance_notice_initial_discard_ck CHECK((discarded_at IS NULL AND discarded_by IS NULL AND discard_resolution_id IS NULL) OR (discarded_at IS NOT NULL AND discarded_by IS NOT NULL AND discard_resolution_id IS NOT NULL AND completed_document_id IS NULL)) · Migración 1793232000000 · línea 13

Fuentes técnicas: Migración 1793145600000 · línea 14; Migración 1793232000000 · línea 13.

maintenance_notice_activity

Registro de notice activity. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
notice_id	uuid	NO	FK	maintenance_notice(id)
kind	varchar(24)	NO		
text	varchar(4000)	NO		
actor_id	uuid	NO	FK	maintenance_member(core_user_id)
actor_snapshot	jsonb	NO		
occurred_at	timestamptz	NO		
notice_version	integer	SÍ		
document_id	uuid	SÍ	FK	maintenance_document(id)
work_order_id	uuid	SÍ	FK	work_order(id)

FK (notice_id) → maintenance_notice(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (actor_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (document_id) → maintenance_document(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793145600000 · línea 14.

FK (work_order_id) → work_order(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

CHECK: CONSTRAINT maintenance_notice_activity_kind_check CHECK(kind IN ('note','dismissed','reopened','evidence_retired','converted')) · Migración 1793232000000 · línea 13

CHECK: CONSTRAINT maintenance_notice_activity_kind_ck CHECK((kind='note' AND notice_version IS NULL AND document_id IS NULL AND work_order_id IS NULL) OR (kind IN ('dismissed','reopened') AND length(text)<=1000 AND notice_version>=2 AND document_id IS NULL AND work_order_id IS NULL) OR (kind='evidence_retired' AND length(text)<=1000 AND notice_version>=2 AND document_id IS NOT NULL AND work_order_id IS NULL) OR (kind='converted' AND length(text)<=1000 AND notice_version>=2 AND document_id IS NULL AND work_order_id IS NOT NULL)) · Migración 1793232000000 · línea 13

Índice: CREATE INDEX maintenance_notice_activity_list_idx ON esquema.maintenance_notice_activity(notice_id,occurred_at,id) · Migración 1793145600000 · línea 14

Índice: CREATE INDEX maintenance_notice_activity_document_idx ON esquema.maintenance_notice_activity(document_id) WHERE document_id IS NOT NULL · Migración 1793145600000 · línea 14

Índice: CREATE UNIQUE INDEX maintenance_notice_activity_conversion_idx ON esquema.maintenance_notice_activity(notice_id) WHERE kind='converted' · Migración 1793232000000 · línea 13

Fuentes técnicas: Migración 1793145600000 · línea 14; Migración 1793232000000 · línea 13.

work_order_notice_conversion

Operación de conversión de aviso en OT.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: work_order_id. Unicidad total: (emission_operation_id); (draft_id).

Campo	Tipo	NULL	Clave	Referencia física
work_order_id	uuid	NO	PK, FK	work_order(id)
emission_operation_id	uuid	NO	FK	work_order_operation(id)
requested_notice_ids	uuid[]	NO		
draft_id	uuid	SÍ	FK	work_order_draft(id)
converted_by	uuid	NO	FK	maintenance_member(core_user_id)
converted_at	timestamptz	NO		

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1793232000000 · línea 13.

FK (emission_operation_id) → work_order_operation(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1793232000000 · línea 13.

FK (draft_id) → work_order_draft(id). Padres por hija: 0..1; hijas por padre: 0..1. Migración 1793232000000 · línea 13.

FK (converted_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

Fuentes técnicas: Migración 1793232000000 · línea 13.

work_order_notice_origin

Vínculo de aviso de origen con OT.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: notice_id. Unicidad total: (work_order_id, notice_id); (work_order_id, position).

Campo	Tipo	NULL	Clave	Referencia física
notice_id	uuid	NO	PK, FK	maintenance_notice(id)
work_order_id	uuid	NO	FK	work_order_notice_conversion(work_order_id)
position	smallint	NO		
notice_version	integer	NO		
content_token	char(64)	NO		
selected_revision_ids	uuid[]	NO		
snapshot	jsonb	NO		

FK (notice_id) → maintenance_notice(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1793232000000 · línea 13.

FK (work_order_id) → work_order_notice_conversion(work_order_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

Fuentes técnicas: Migración 1793232000000 · línea 13.

work_order_notice_evidence

Evidencia de aviso conservada en OT.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: work_order_id, revision_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
work_order_id	uuid	NO	PK, FK	work_order_notice_origin(work_order_id, notice_id)
notice_id	uuid	NO	FK	work_order_notice_origin(work_order_id, notice_id)
document_id	uuid	NO	FK	maintenance_document(id)
revision_id	uuid	NO	PK, FK	maintenance_document_revision(id)

FK (document_id) → maintenance_document(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (revision_id) → maintenance_document_revision(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (work_order_id, notice_id) → work_order_notice_origin(work_order_id, notice_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

Índice: CREATE INDEX work_order_notice_evidence_origin_idx ON esquema.work_order_notice_evidence(notice_id,document_id) · Migración 1793232000000 · línea 13

Fuentes técnicas: Migración 1793232000000 · línea 13.

notice_evidence_resolution_operation

Registro de notice evidence resolution operation. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
notice_id	uuid	NO	FK	maintenance_notice(id)

Campo	Tipo	NULL	Clave	Referencia física
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
idempotency_key	uuid	NO		
decision	varchar(8)	NO		
initial_file_id	uuid	SÍ	FK	maintenance_notice_initial_file(id)
document_operation_id	uuid	SÍ	FK	maintenance_document_operation(id)
active_document_operation_id	uuid	SÍ	FK	maintenance_document_operation(id)
request_hash	char(64)	NO		
request_body	jsonb	NO		
state	varchar(12)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
created_at	timestamptz	NO		
completed_at	timestamptz	SÍ		

FK (notice_id) → maintenance_notice(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (initial_file_id) → maintenance_notice_initial_file(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (document_operation_id) → maintenance_document_operation(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

FK (active_document_operation_id) → maintenance_document_operation(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1793232000000 · línea 13.

CHECK: CONSTRAINT notice_resolution_subject_ck CHECK((initial_file_id IS NULL)<>(document_operation_id IS NULL)) · Migración 1793232000000 · línea 13

CHECK: CONSTRAINT notice_resolution_terminal_ck CHECK((state='pending' AND response_status IS NULL AND response_body IS NULL AND completed_at IS NULL) OR (state IN ('applied','not_applied') AND response_status IS NOT NULL AND response_status BETWEEN 200 AND 599 AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND completed_at IS NOT NULL AND completed_at>=created_at)) · Migración 1793232000000 · línea 13

Índice: CREATE INDEX notice_evidence_resolution_target_idx ON esquema.notice_evidence_resolution_operation(notice_id,state,id) · Migración 1793232000000 · línea 13

Fuentes técnicas: Migración 1793232000000 · línea 13.

work_order_mutation_operation

Registro de work order mutation operation. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)
work_order_id	uuid	NO	FK	work_order(id)
action	varchar(16)	NO		
idempotency_key	uuid	NO		
request_hash	char(64)	NO		
request_body	jsonb	NO		
expected_version	integer	NO		
transaction_id	xid8	NO		
state	varchar(12)	NO		
result_version	integer	SÍ		
response_status	integer	SÍ		
response_body	jsonb	SÍ		
response_etag	text	SÍ		
created_at	timestamptz	NO		
completed_at	timestamptz	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
component_permission_required	boolean	NO		
execution_permission_required	boolean	NO		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793577600000 · línea 5.

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793577600000 · línea 5.

CHECK: CONSTRAINT work_order_mutation_terminal_ck CHECK((state='pending' AND num_nonnulls(result_version,response_status,response_body,response_etag,completed_at)=0) OR (state='applied' AND result_version IS NOT NULL AND result_version::bigint=expected_version::bigint+1 AND response_status IS NOT NULL AND response_status=200 AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND response_etag IS NOT NULL AND completed_at IS NOT NULL AND completed_at>=created_at) OR (state='not_applied' AND result_version IS NULL AND response_status IS NOT NULL AND response_status BETWEEN 400 AND 599 AND response_body IS NOT NULL AND jsonb_typeof(response_body)='object' AND response_etag IS NULL AND completed_at IS NOT NULL AND completed_at>=created_at)) · Migración 1793577600000 · línea 5

CHECK: CONSTRAINT work_order_mutation_operation_action_check CHECK(action IN ('update','planning','postpone','cancel','assign','execute','correct','void','submit_review','close','reopen')) · Migración 1794873600000 · línea 35

Índice: CREATE UNIQUE INDEX work_order_mutation_version_key ON esquema.work_order_mutation_operation(work_order_id,result_version) WHERE state='applied' · Migración 1793577600000 · línea 5

Índice: CREATE INDEX work_order_mutation_order_idx ON esquema.work_order_mutation_operation(work_order_id,created_at,id) · Migración 1793577600000 · línea 5

Fuentes técnicas: Migración 1793577600000 · línea 5; Migración 1794787200000 · línea 27; Migración 1794873600000 · línea 35.

work_order_change

Registro de work order change. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (operation_id, entity_type, entity_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
operation_id	uuid	NO	FK	work_order_mutation_operation(id)
work_order_id	uuid	NO	FK	work_order(id)
version	integer	NO		

Campo	Tipo	NULL	Clave	Referencia física
entity_type	varchar(40)	NO		
entity_id	uuid	NO		
before_value	jsonb	SÍ		
after_value	jsonb	NO		
occurred_at	timestamptz	NO		

FK (operation_id) → work_order_mutation_operation(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793577600000 · línea 5.

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793577600000 · línea 5.

CHECK: CONSTRAINT work_order_change_entity_type_check CHECK(entity_type IN ('work_order','work_order_asset_scope','work_order_task','work_order_task_measurement','work_order_planning_precision','work_order_assignment','work_order_intervention')) · Migración 1794441600000 · línea 33

Índice: CREATE INDEX work_order_change_order_idx ON esquema.work_order_change(work_order_id,version,entity_type,entity_id) · Migración 1793577600000 · línea 5

Fuentes técnicas: Migración 1793577600000 · línea 5.

work_order_activity

Registro de work order activity. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (operation_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
operation_id	uuid	NO	FK	work_order_mutation_operation(id)
work_order_id	uuid	NO	FK	work_order(id)
version	integer	NO		
action	varchar(12)	NO		

Campo	Tipo	NULL	Clave	Referencia física
actor_id	uuid	NO	FK	maintenance_member(core_user_id)
actor_name	varchar(200)	SÍ		
occurred_at	timestamptz	NO		
scheduled_for	timestamptz	SÍ		
scheduled_precision	varchar(8)	SÍ		
note	varchar(1000)	SÍ		

FK (operation_id) → work_order_mutation_operation(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1793577600000 · línea 5.

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793577600000 · línea 5.

FK (actor_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1793577600000 · línea 5.

CHECK: CONSTRAINT work_order_activity_content_ck CHECK((action='postpone' AND ((scheduled_for IS NULL)=(scheduled_precision IS NULL))) OR (action='cancel' AND scheduled_for IS NULL AND scheduled_precision IS NULL AND NULLIF(btrim(note),'') IS NOT NULL)) · Migración 1793577600000 · línea 5

Índice: CREATE INDEX work_order_activity_order_idx ON esquema.work_order_activity(work_order_id,occurred_at DESC,id DESC) · Migración 1793577600000 · línea 5

Fuentes técnicas: Migración 1793577600000 · línea 5.

work_order_print_request

Solicitud controlada de impresión.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_core_user_id, idempotency_key); (request_order).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
request_order	bigint	NO		
principal_core_user_id	uuid	NO	FK	maintenance_member(core_user_id)

Campo	Tipo	NULL	Clave	Referencia física
idempotency_key	uuid	NO		
request_hash	varchar(64)	NO		
work_order_id	uuid	NO	FK	work_order(id)
expected_version	integer	NO		
preparation_id	uuid	NO		
prepared_at	timestamptz	SÍ		
document_sha256	varchar(64)	SÍ		
pdf_sha256	varchar(64)	SÍ		
state	varchar(12)	NO		
requested_at	timestamptz	SÍ		
response_status	integer	NO		
response_body	jsonb	NO		
completed_at	timestamptz	NO		

FK (principal_core_user_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794355200000 · línea 30.

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794355200000 · línea 30.

CHECK: CONSTRAINT work_order_print_request_hash_ck CHECK(request_hash ~ '^[0-9a-f]{64}$') · Migración 1794355200000 · línea 30

CHECK: CONSTRAINT work_order_print_request_version_ck CHECK(expected_version>0) · Migración 1794355200000 · línea 30

CHECK: CONSTRAINT work_order_print_request_terminal_ck CHECK((state='applied' AND response_status=200 AND requested_at IS NOT NULL AND prepared_at IS NOT NULL AND document_sha256 IS NOT NULL AND document_sha256 ~ '^[0-9a-f]{64}$' AND pdf_sha256 IS NOT NULL AND pdf_sha256 ~ '^[0-9a-f]{64}$') OR (state='not_applied' AND response_status BETWEEN 400 AND 599 AND requested_at IS NULL)) · Migración 1794355200000 · línea 30

CHECK: CONSTRAINT work_order_print_request_response_ck CHECK((jsonb_typeof(response_body)='object' AND octet_length(response_body::text)<=16384 AND response_body->>'state' IS NOT DISTINCT FROM state AND response_body->>'key' IS NOT DISTINCT FROM idempotency_key::text AND response_body->>'request_hash' IS NOT DISTINCT FROM request_hash AND CASE WHEN state='applied' THEN jsonb_typeof(response_body->'result')='object' AND response_body#>>'{result,id}'=work_order_id::text AND response_body#>>'{result,version}'=expected_version::text AND response_body#>>'{result,preparation_id}'=preparation_id::text AND (response_body#>>'{result,requested_at}')::timestamptz=requested_at AND length(response_body#>>'{result,number}')>0 ELSE jsonb_typeof(response_body->'error')='object' AND response_body#>>'{error,status}'=response_status::text AND length(response_body#>>'{error,code}')>0 AND length(response_body#>>'{error,message}')>0 END) IS TRUE) · Migración 1794355200000 · línea 30

Índice: CREATE INDEX work_order_print_request_latest_idx ON esquema.work_order_print_request(work_order_id,request_order DESC) WHERE state='applied' · Migración 1794355200000 · línea 30

Fuentes técnicas: Migración 1794355200000 · línea 30.

work_order_intervention

Intervención de trabajo.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (original_save_order); (work_order_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK, FK	work_order_intervention_revision(intervention_id, id)
work_order_id	uuid	NO	FK	work_order(id)
original_save_order	bigint	NO		
version	integer	NO		
current_revision_id	uuid	NO	FK	work_order_intervention_revision(intervention_id, id)
created_at	timestamptz	NO		
created_by	uuid	NO	FK	maintenance_member(core_user_id)
creator_name	varchar(200)	SÍ		
creator_email	varchar(320)	SÍ		
updated_at	timestamptz	NO		

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (created_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (id, current_revision_id) → work_order_intervention_revision(intervention_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

Índice: CREATE INDEX work_order_intervention_order_idx ON esquema.work_order_intervention(work_order_id,original_save_order DESC,id) · Migración 1794441600000 · línea 33

Fuentes técnicas: Migración 1794441600000 · línea 33.

work_order_intervention_revision

Revisión de intervención.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (operation_id); (intervention_id, revision_number); (intervention_id, id); (work_order_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK, FK	work_order_intervention_performer(revision_id, id)
work_order_id	uuid	NO	FK	work_order_intervention(work_order_id, id)
intervention_id	uuid	NO	FK	work_order_intervention(work_order_id, id)
revision_number	integer	NO		
operation_id	uuid	NO	FK	work_order_mutation_operation(id)
status	varchar(8)	NO		
performed_on	date	NO		
general_result	varchar(4000)	NO		
effective_minutes	integer	SÍ		
executor_notes	varchar(4000)	SÍ		
operator_notes	varchar(4000)	SÍ		
responsible_performer_id	uuid	SÍ	FK	work_order_intervention_performer(revision_id, id)
reason	varchar(1000)	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
recorded_at	timestamptz	NO		
recorded_by	uuid	NO	FK	maintenance_member(core_user_id)
actor_name	varchar(200)	SÍ		
actor_email	varchar(320)	SÍ		

FK (operation_id) → work_order_mutation_operation(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1794441600000 · línea 33.

FK (recorded_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (work_order_id, intervention_id) → work_order_intervention(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (id, responsible_performer_id) → work_order_intervention_performer(revision_id, id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

CHECK: CHECK((revision_number=1 AND status='valid' AND reason IS NULL) OR (revision_number>1 AND NULLIF(btrim(reason),'') IS NOT NULL)) · Migración 1794441600000 · línea 33

Índice: CREATE INDEX work_order_intervention_revision_date_idx ON esquema.work_order_intervention_revision(work_order_id,performed_on DESC,intervention_id,revision_number DESC) · Migración 1794441600000 · línea 33

Fuentes técnicas: Migración 1794441600000 · línea 33.

work_order_intervention_performer

Registro de work order intervention performer. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (revision_id, kind, subject_id); (revision_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
work_order_id	uuid	NO	FK	work_order_intervention_revision(work_order_id, id); work_order_assignment(work_order_id, id)
revision_id	uuid	NO	FK	work_order_intervention_revision(work_order_id, id)
kind	varchar(8)	NO		

Campo	Tipo	NULL	Clave	Referencia física
staff_id	uuid	SÍ	FK	internal_staff(id)
provider_id	uuid	SÍ	FK	provider(id)
subject_id	uuid	SÍ		
assignment_id	uuid	NO	FK	work_order_assignment(work_order_id, id)
subject_snapshot	jsonb	NO		
external_first_name	varchar(100)	SÍ		
external_last_name	varchar(100)	SÍ		

FK (staff_id) → internal_staff(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (provider_id) → provider(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (work_order_id, revision_id) → work_order_intervention_revision(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (work_order_id, assignment_id) → work_order_assignment(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

CHECK: CHECK((kind='staff' AND staff_id IS NOT NULL AND provider_id IS NULL AND external_first_name IS NULL AND external_last_name IS NULL) OR (kind='provider' AND staff_id IS NULL AND provider_id IS NOT NULL AND NULLIF(btrim(external_first_name),'') IS NOT NULL AND NULLIF(btrim(external_last_name),'') IS NOT NULL)) · Migración 1794441600000 · línea 33

CHECK: CHECK(esquema.maintenance_work_order_snapshot_valid('assignment',subject_snapshot) AND subject_snapshot->>'id'=COALESCE(staff_id,provider_id)::text) · Migración 1794441600000 · línea 33

Índice: CREATE INDEX work_order_intervention_performer_subject_idx ON esquema.work_order_intervention_performer(kind,subject_id,work_order_id,revision_id) · Migración 1794441600000 · línea 33

Fuentes técnicas: Migración 1794441600000 · línea 33.

work_order_intervention_task_result

Registro de work order intervention task result. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (revision_id, task_id); (work_order_id, task_id, id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	

Campo	Tipo	NULL	Clave	Referencia física
work_order_id	uuid	NO	FK	work_order_intervention_revision(work_order_id, id); work_order_task(work_order_id, id)
revision_id	uuid	NO	FK	work_order_intervention_revision(work_order_id, id)
task_id	uuid	NO	FK	work_order_task(work_order_id, id)
answer	varchar(8)	SÍ		
note	varchar(4000)	SÍ		
definition_snapshot	jsonb	NO		
definition_hash	char(64)	NO		

FK (work_order_id, revision_id) → work_order_intervention_revision(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (work_order_id, task_id) → work_order_task(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

CHECK: CHECK(answer IS DISTINCT FROM 'not_done' OR NULLIF(btrim(note),'') IS NOT NULL) · Migración 1794441600000 · línea 33

Índice: CREATE INDEX work_order_intervention_task_history_idx ON esquema.work_order_intervention_task_result(task_id,revision_id,id) · Migración 1794441600000 · línea 33

Índice: CREATE INDEX work_order_intervention_task_target_idx ON esquema.work_order_intervention_task_result(((definition_snapshot->>'asset_id')::uuid),((definition_snapshot->>'component_id')::uuid),revision_id) · Migración 1794441600000 · línea 33

Fuentes técnicas: Migración 1794441600000 · línea 33.

work_order_intervention_reading

Registro de work order intervention reading. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: task_result_id, measurement_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
work_order_id	uuid	NO	FK	work_order_intervention_task_result(work_order_id, task_id, id); work_order_task_measurement(work_order_id, task_id, id)

Campo	Tipo	NULL	Clave	Referencia física
task_id	uuid	NO	FK	work_order_intervention_task_result(work_order_id, task_id, id); work_order_task_measurement(work_order_id, task_id, id)
task_result_id	uuid	NO	PK, FK	work_order_intervention_task_result(work_order_id, task_id, id)
measurement_id	uuid	NO	PK, FK	work_order_task_measurement(work_order_id, task_id, id)
value	numeric	NO		

FK (work_order_id, task_id, task_result_id) → work_order_intervention_task_result(work_order_id, task_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

FK (work_order_id, task_id, measurement_id) → work_order_task_measurement(work_order_id, task_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794441600000 · línea 33.

Índice: CREATE INDEX work_order_intervention_reading_source_idx ON esquema.work_order_intervention_reading(measurement_id,task_result_id) · Migración 1794441600000 · línea 33

Fuentes técnicas: Migración 1794441600000 · línea 33.

work_order_component_fact

Hecho registrado sobre componente.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (intervention_id, component_id); (work_order_id, id); (id, intervention_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK, FK	work_order_component_fact_revision(fact_id, id)
work_order_id	uuid	NO	FK	work_order_intervention(work_order_id, id); work_order_asset_scope(work_order_id, asset_id)
intervention_id	uuid	NO	FK	work_order_intervention(work_order_id, id)
asset_id	uuid	NO	FK	work_order_asset_scope(work_order_id, asset_id); asset_component(asset_id, id)
component_id	uuid	NO	FK	asset_component(asset_id, id)
version	integer	NO		
current_revision_id	uuid	NO	FK	work_order_component_fact_revision(fact_id, id)

Campo	Tipo	NULL	Clave	Referencia física
created_at	timestamptz	NO		
created_by	uuid	NO	FK	maintenance_member(core_user_id)

FK (created_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (work_order_id, intervention_id) → work_order_intervention(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (work_order_id, asset_id) → work_order_asset_scope(work_order_id, asset_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (asset_id, component_id) → asset_component(asset_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (id, current_revision_id) → work_order_component_fact_revision(fact_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

Índice: CREATE INDEX work_order_component_fact_component_idx ON esquema.work_order_component_fact(component_id,created_at DESC,id) · Migración 1794787200000 · línea 27

Fuentes técnicas: Migración 1794787200000 · línea 27.

work_order_component_fact_revision

Registro de work order component fact revision. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (fact_id, revision_number); (fact_id, id); (operation_id, fact_id).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
fact_id	uuid	NO	FK	work_order_component_fact(work_order_id, id)
work_order_id	uuid	NO	FK	work_order_component_fact(work_order_id, id); work_order_intervention_revision(work_order_id, id)
intervention_revision_id	uuid	NO	FK	work_order_intervention_revision(work_order_id, id)
operation_id	uuid	NO	FK	work_order_mutation_operation(id)
revision_number	integer	NO		
status	varchar(8)	NO		

Campo	Tipo	NULL	Clave	Referencia física
performed_on	date	NO		
mode	varchar(20)	NO		
component_context	jsonb	NO		
installed_quantity	integer	NO		
found_state_id	uuid	NO	FK	technical_catalog_value(id)
result_state_id	uuid	NO	FK	technical_catalog_value(id)
found_state	jsonb	NO		
result_state	jsonb	NO		
resolution	varchar(16)	SÍ		
replaced	boolean	NO		
work_performed	boolean	NO		
has_pending	boolean	NO		
description	varchar(4000)	SÍ		
affected_quantity	integer	NO		
replaced_quantity	integer	NO		
effect_kind	varchar(16)	NO		
effect_history_id	uuid	SÍ	FK	asset_component_history(id)
reason	varchar(1000)	SÍ		
recorded_at	timestamptz	NO		
recorded_by	uuid	NO	FK	maintenance_member(core_user_id)
actor_name	varchar(200)	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
actor_email	varchar(320)	SÍ		

FK (operation_id) → work_order_mutation_operation(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (found_state_id) → technical_catalog_value(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (result_state_id) → technical_catalog_value(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (effect_history_id) → asset_component_history(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (recorded_by) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (work_order_id, fact_id) → work_order_component_fact(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (work_order_id, intervention_revision_id) → work_order_intervention_revision(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

CHECK: CHECK(found_state->>'id'=found_state_id::text AND result_state->>'id'=result_state_id::text AND jsonb_typeof(found_state->'is_non_operational')='boolean' AND jsonb_typeof(result_state->'is_non_operational')='boolean') · Migración 1794787200000 · línea 27

CHECK: CHECK((mode='reviewed_unchanged' AND found_state_id=result_state_id AND resolution IS NULL AND NOT replaced AND NOT work_performed AND affected_quantity=installed_quantity AND replaced_quantity=0 AND effect_kind IN ('unchanged','history_only','preserved','rectified')) OR (mode='intervention' AND resolution IS NOT NULL)) · Migración 1794787200000 · línea 27

CHECK: CHECK((replaced AND replaced_quantity>=1) OR (NOT replaced AND replaced_quantity=0)) · Migración 1794787200000 · línea 27

CHECK: CHECK(NOT (work_performed OR replaced OR resolution='no' OR (result_state->>'is_non_operational')::boolean OR has_pending) OR NULLIF(btrim(description),'') IS NOT NULL) · Migración 1794787200000 · línea 27

CHECK: CHECK((effect_kind IN ('updated','rectified'))=(effect_history_id IS NOT NULL)) · Migración 1794787200000 · línea 27

CHECK: CHECK((revision_number=1 AND status='valid' AND reason IS NULL) OR (revision_number>1 AND NULLIF(btrim(reason),'') IS NOT NULL)) · Migración 1794787200000 · línea 27

Índice: CREATE INDEX component_fact_revision_date_idx ON esquema.work_order_component_fact_revision(performed_on DESC,fact_id,revision_number DESC) · Migración 1794787200000 · línea 27

Índice: CREATE INDEX component_fact_revision_operation_idx ON esquema.work_order_component_fact_revision(operation_id,id) · Migración 1794787200000 · línea 27

Fuentes técnicas: Migración 1794787200000 · línea 27.

work_order_component_fact_task_origin

Registro de work order component fact task origin. Finalidad inferida a partir del nombre, los campos y las relaciones declaradas..

Finalidad inferida; estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: fact_revision_id, task_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
fact_revision_id	uuid	NO	PK, FK	work_order_component_fact_revision(fact_id, id)
fact_id	uuid	NO	FK	work_order_component_fact_revision(fact_id, id)
work_order_id	uuid	NO	FK	work_order_task(work_order_id, id)
task_id	uuid	NO	PK, FK	work_order_task(work_order_id, id)
definition_snapshot	jsonb	NO		
definition_hash	char(64)	NO		

FK (fact_id, fact_revision_id) → work_order_component_fact_revision(fact_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

FK (work_order_id, task_id) → work_order_task(work_order_id, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794787200000 · línea 27.

Índice: CREATE INDEX component_fact_task_origin_task_idx ON esquema.work_order_component_fact_task_origin(task_id,fact_id) · Migración 1794787200000 · línea 27

Fuentes técnicas: Migración 1794787200000 · línea 27.

work_order_decision

Decisión sobre el ciclo de una OT.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (operation_id); (work_order_id, version).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	

Campo	Tipo	NULL	Clave	Referencia física
operation_id	uuid	NO	FK	work_order_mutation_operation(id)
work_order_id	uuid	NO	FK	work_order(id)
version	integer	NO		
action	varchar(16)	NO		
source_state	varchar(20)	NO		
target_state	varchar(20)	NO		
actor_id	uuid	NO	FK	maintenance_member(core_user_id)
actor_name	varchar(200)	SÍ		
occurred_at	timestamptz	NO		
note	varchar(1000)	SÍ		
preceding_decision_id	uuid	SÍ	FK	work_order_decision(id)
historical_closed_origin	boolean	NO		
accepted_manifest	jsonb	NO		
accepted_exceptions	jsonb	NO		

FK (operation_id) → work_order_mutation_operation(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1794873600000 · línea 35.

FK (work_order_id) → work_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794873600000 · línea 35.

FK (actor_id) → maintenance_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1794873600000 · línea 35.

FK (preceding_decision_id) → work_order_decision(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1794873600000 · línea 35.

CHECK: CHECK((action='submit_review' AND source_state IN ('open','assigned','in_progress','reopened') AND target_state='pending_review') OR (action='close' AND source_state IN ('open','assigned','in_progress','reopened','pending_review') AND target_state='closed') OR (action='reopen' AND source_state IN ('pending_review','closed') AND target_state='reopened' AND NULLIF(btrim(note),'') IS NOT NULL)) · Migración 1794873600000 · línea 35

CHECK: CHECK(NOT historical_closed_origin OR (action='reopen' AND source_state='closed' AND preceding_decision_id IS NULL)) · Migración 1794873600000 · línea 35

Índice: CREATE INDEX work_order_decision_history_idx ON esquema.work_order_decision(work_order_id,version DESC,id) · Migración 1794873600000 · línea 35

Fuentes técnicas: Migración 1794873600000 · línea 35.

17 tablas lógicas · 31 FK locales · 5 láminas de dominio.

Identidad y permisos

calendar_member.core_user_id conserva el identificador externo de Core. No tiene una FK a platform_user: las relaciones de roles y actores son FK a calendar_member dentro del esquema de Calendario. role, permission y user_role son locales; no se comparte una tabla RBAC entre servicios.

Espacios y datos históricos

core_space_projection y las proyecciones de relaciones/linaje son copias locales del catálogo de Core. Sus FK entre proyecciones y hacia core_inbox_event son físicas locales. La igualdad de UUID con Core constituye una referencia de integración, no una FK entre bases. event_space conserva código, nombre, ruta y versión del espacio como snapshots históricos; conservan el contexto de la asociación.

Eventos y ocupación

calendar_event mantiene las fechas de ocupación y de apertura al público, zona horaria, estado y versión. event_space resuelve la relación N:M con espacios mediante PK (event_id,core_space_id). Cada fila puente requiere un evento y un espacio; la FK no obliga por sí sola a que todo evento tenga espacios. Los CHECK y la lógica de aplicación añaden reglas según estado: la opcionalidad física de un actor de confirmación no elimina esas reglas.

Auditoría y actores alternativos

calendar_audit_log es una tabla particionada con PK (occurred_at,id). calendar_audit_change tiene una FK compuesta a esa cabecera. Las particiones no son entidades de negocio adicionales. El registro distingue actor humano y actor de servicio: los campos y CHECK se describen en el diccionario; no se agrega una FK inexistente para el servicio. Triggers conservan historia y rechazan mutaciones de auditoría.

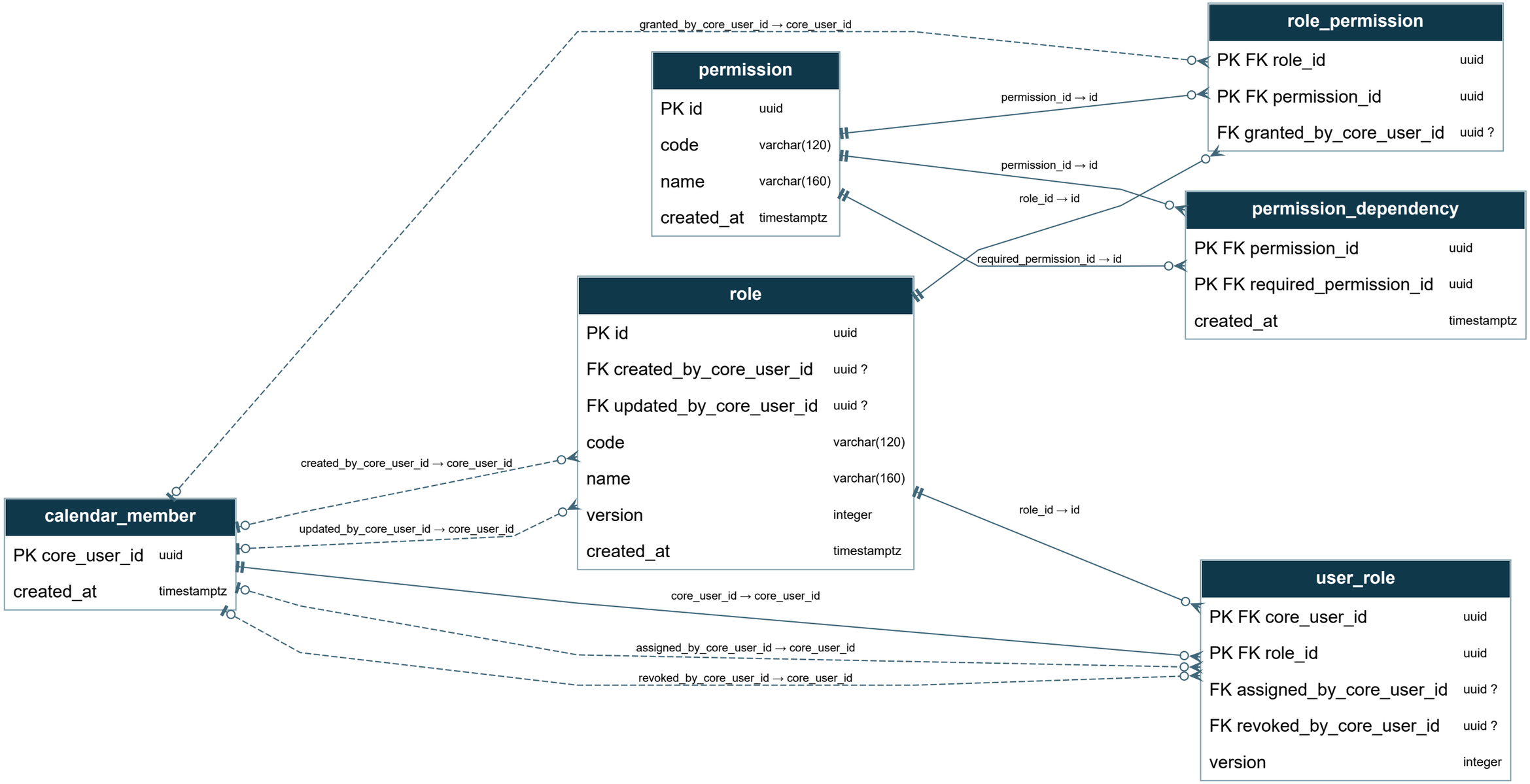
Mensajes y controles de repetición

core_inbox_event registra recepción y procesamiento; core_sync_checkpoint registra avance de cada stream. calendar_outbox_event pertenece a calendar_event mediante FK y usa unicidad (calendar_event_id,aggregate_version). Su campo sequence es una identidad generada: PostgreSQL implica NOT NULL aunque el SQL no lo escriba literalmente. idempotency_record conserva identificación de solicitud y respuesta, sin convertir cualquier UUID contenido en una FK.

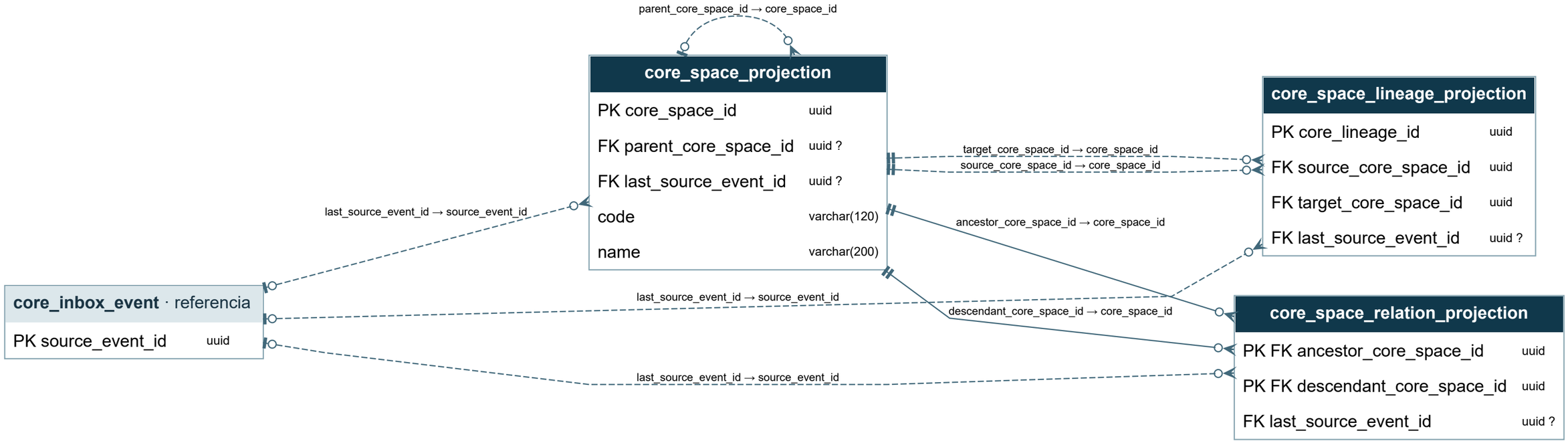
Publicación y despliegue

El publicador de eventos de Calendario existe en código. El default de CALENDAR_OUTBOX_PUBLISHER_ENABLED es false y la operación versionada de Infra exige mantenerlo deshabilitado. Esto no acredita qué corre en un ambiente. No se verificó un consumidor de esos eventos en Mantenimiento ni en Depósito; por eso no se muestra un flujo efectivo hacia ellos.

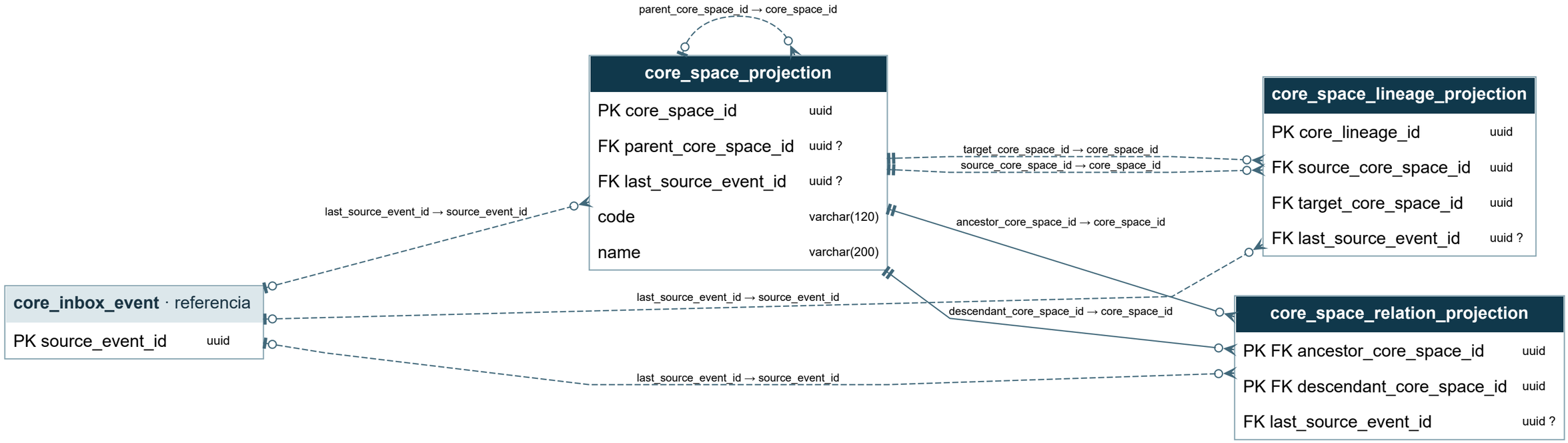
Calendario · Identidad y acceso · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

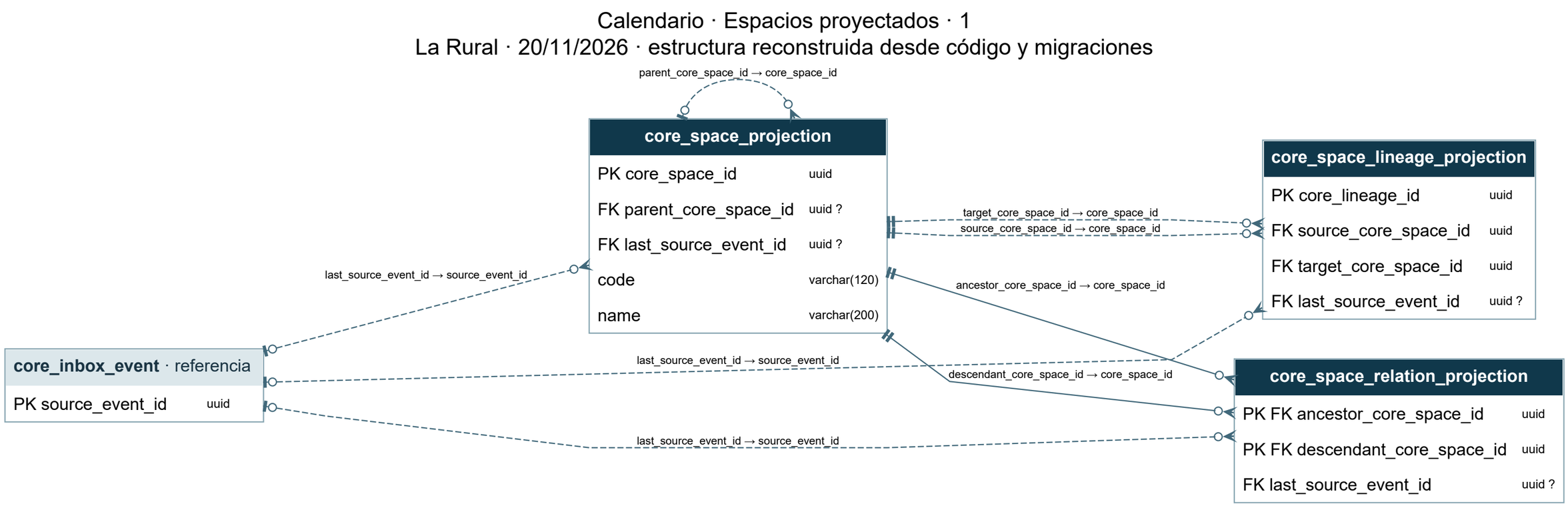


Calendario · Espacios proyectados · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

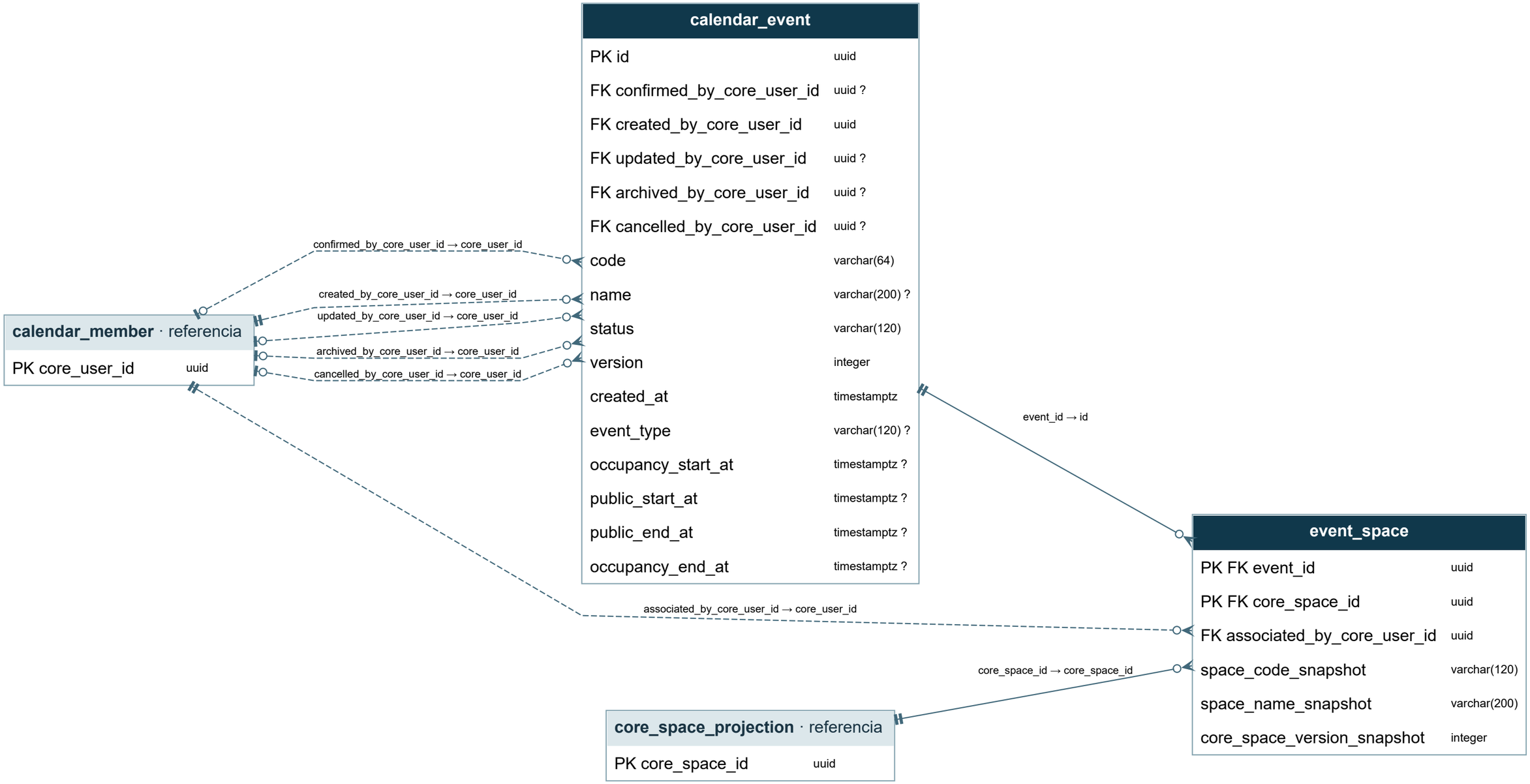


Calendario · Espacios proyectados · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

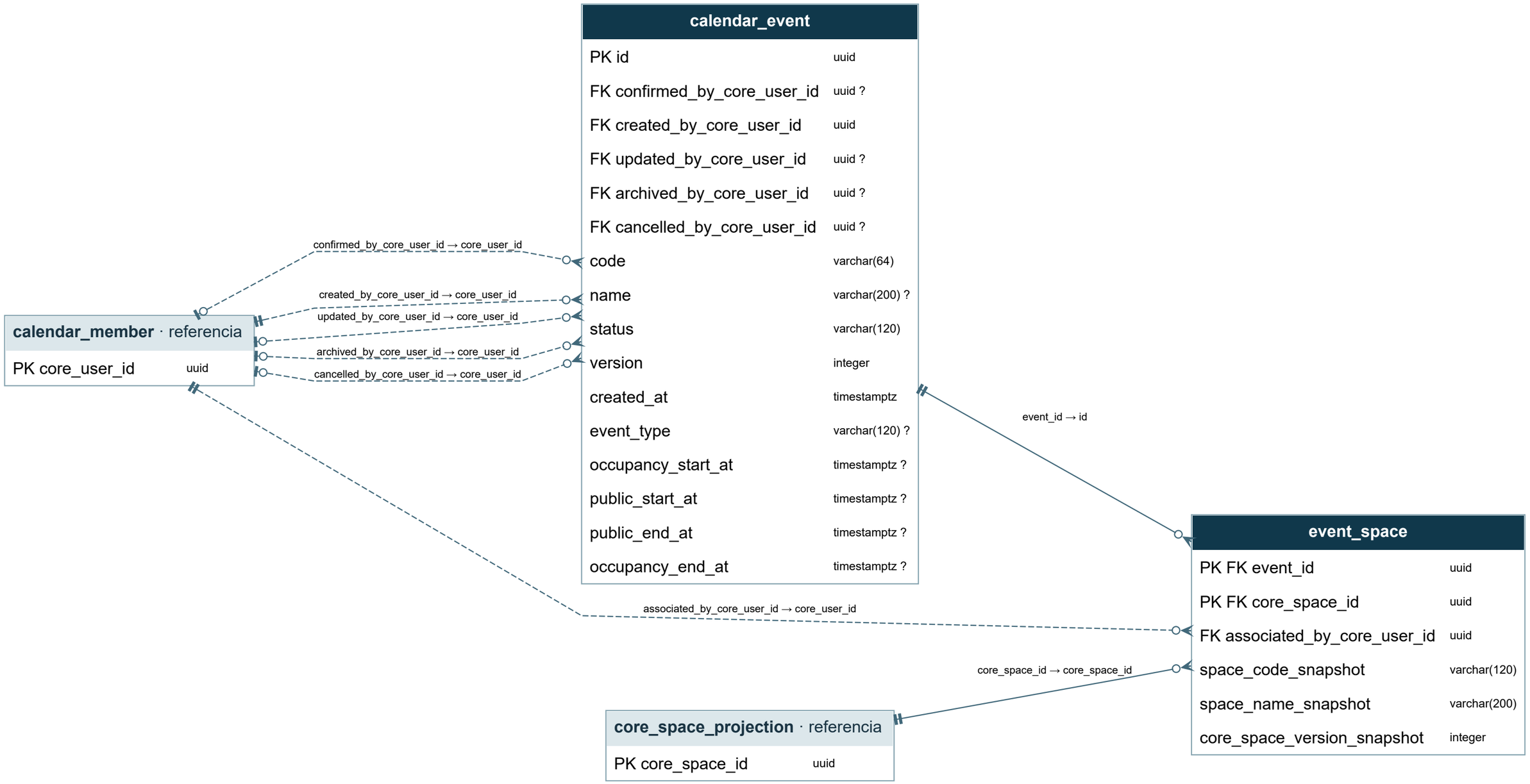



Calendario · Espacios proyectados · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones
parent_core_space_id → core_space_id
core_space_projection
PK core_space_id uuid
FK parent_core_space_id uuid ?
FK last_source_event_id uuid ?
code varchar(120)
name varchar(200)
core_space_lineage_projection
PK core_lineage_id uuid
FK source_core_space_id uuid
FK target_core_space_id uuid
FK last_source_event_id uuid ?
target_core_space_id → core_space_id
source_core_space_id → core_space_id
ancestor_core_space_id → core_space_id
descendant_core_space_id → core_space_id
last_source_event_id → source_event_id
core_inbox_event · referencia
PK source_event_id uuid
last_source_event_id → source_event_id
last_source_event_id → source_event_id
core_space_relation_projection
PK FK ancestor_core_space_id uuid
PK FK descendant_core_space_id uuid
FK last_source_event_id uuid ?

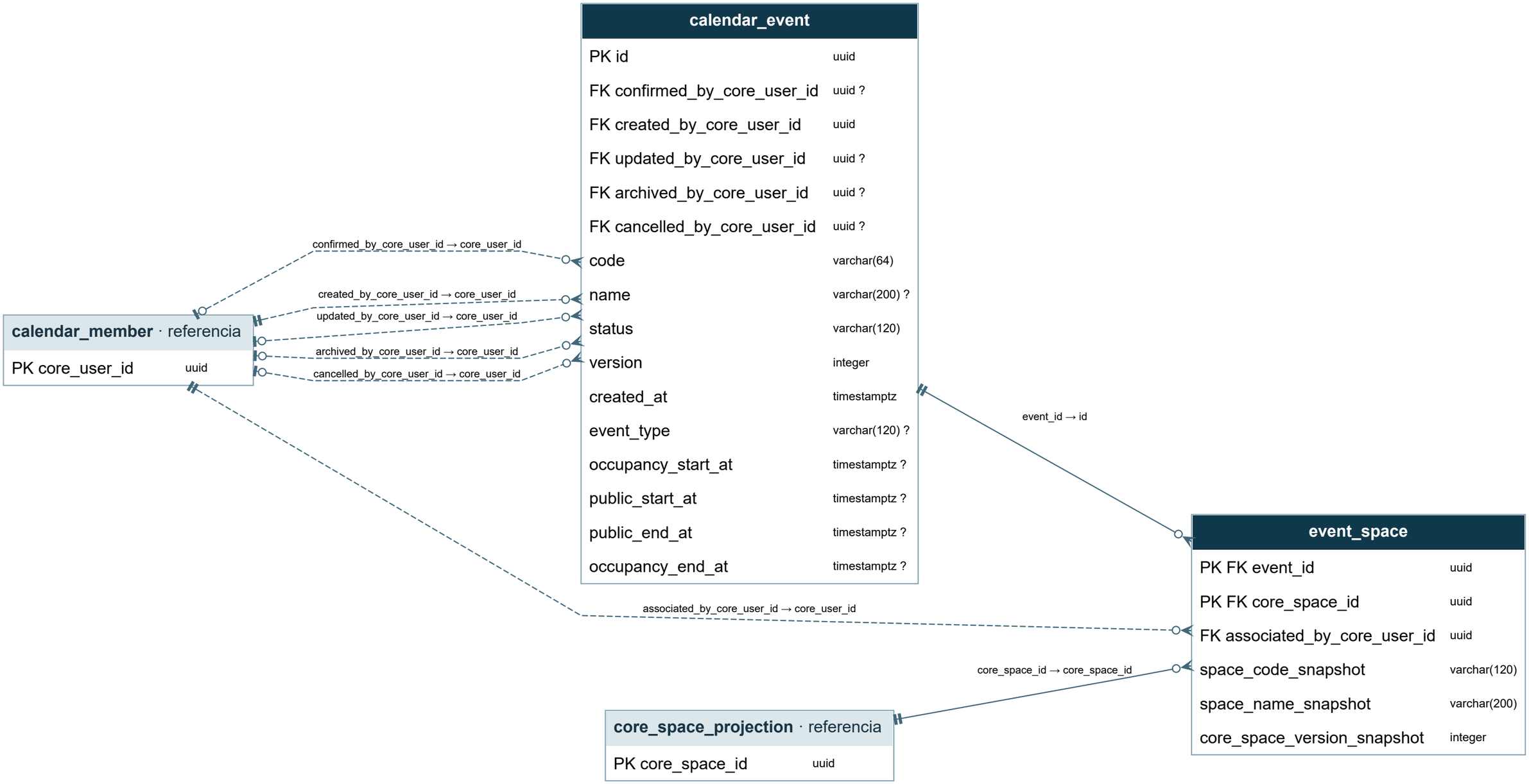
Calendario · Eventos y ocupación · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



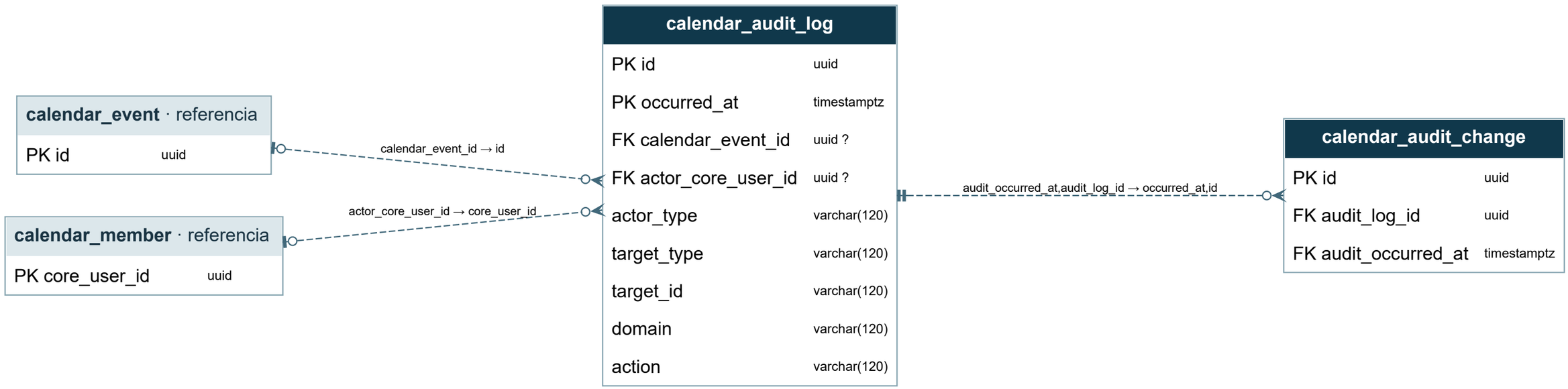
Calendario · Eventos y ocupación · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Calendario · Eventos y ocupación · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

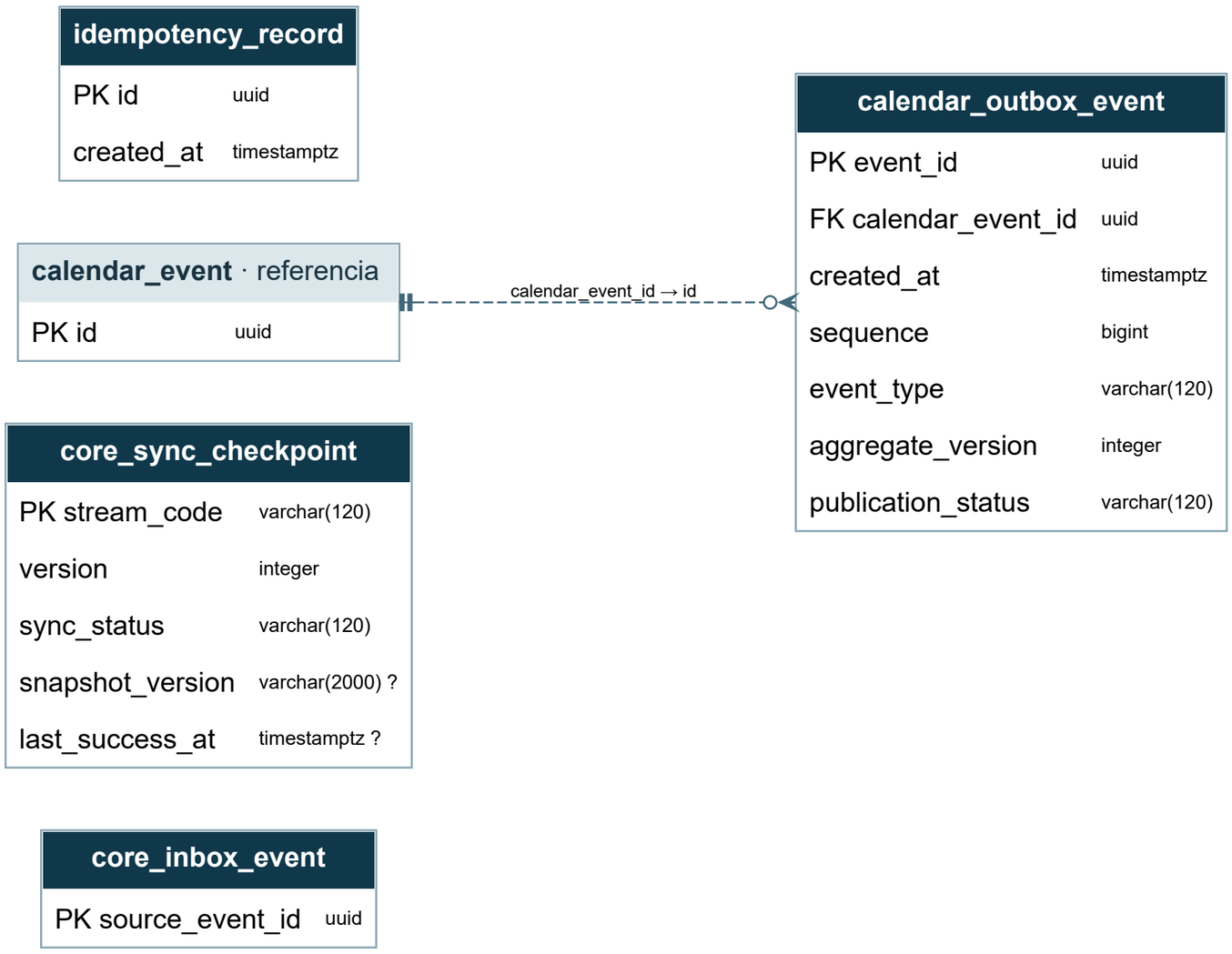


Calendario · Auditoría · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Calendario · Mensajería y control · 1

La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



calendar_member

Miembro local identificado por el UUID canónico de Core.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_user_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
core_user_id	uuid	NO	PK	
email	varchar(320)	NO		
display_name	varchar(200)	NO		
module_access_status	varchar(120)	NO		
core_profile_version	bigint	SÍ		
core_access_version	bigint	SÍ		
last_synced_at	timestamptz	SÍ		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

CHECK: email varchar(320) NOT NULL CHECK (email <> '' AND email = lower(btrim(email))) · Migración 1788739200000 · línea 77

CHECK: display_name varchar(200) NOT NULL CHECK (btrim(display_name) <> '') · Migración 1788739200000 · línea 77

CHECK: module_access_status varchar(120) NOT NULL CHECK (module_access_status IN ('pending','active','suspended','revoked')) · Migración 1788739200000 · línea 77

CHECK: core_profile_version bigint CHECK (core_profile_version >= 0) · Migración 1788739200000 · línea 77

CHECK: core_access_version bigint CHECK (core_access_version >= 0) · Migración 1788739200000 · línea 77

Índice: CREATE UNIQUE INDEX calendar_member_email_uq ON esquema.calendar_member (lower(email)) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_member_status_idx ON esquema.calendar_member (module_access_status, core_user_id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_member_name_idx ON esquema.calendar_member (lower(display_name), core_user_id) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

permission

Permiso granular del dominio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(120)	NO		
name	varchar(160)	NO		
description	varchar(4000)	SÍ		
category	varchar(120)	NO		
is_system	boolean	NO		
is_active	boolean	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

CHECK: code varchar(120) NOT NULL UNIQUE CHECK (btrim(code) <> '') · Migración 1788739200000 · línea 77

CHECK: name varchar(160) NOT NULL CHECK (btrim(name) <> '') · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

role

Rol de autorización del dominio.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(120)	NO		
name	varchar(160)	NO		
description	varchar(4000)	SÍ		
is_system	boolean	NO		
is_protected	boolean	NO		
is_active	boolean	NO		
version	integer	NO		
created_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
updated_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

FK (created_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (updated_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: code varchar(120) NOT NULL UNIQUE CHECK (btrim(code) <> '') · Migración 1788739200000 · línea 77

CHECK: name varchar(160) NOT NULL CHECK (btrim(name) <> '') · Migración 1788739200000 · línea 77

CHECK: is_protected boolean NOT NULL DEFAULT false CHECK (NOT is_protected OR is_system) · Migración 1788739200000 · línea 77

CHECK: version integer NOT NULL DEFAULT 1 CHECK (version >= 1) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

role_permission

Tabla puente entre roles y permisos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: role_id, permission_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
role_id	uuid	NO	PK, FK	role(id)
permission_id	uuid	NO	PK, FK	permission(id)
granted_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
granted_at	timestamptz	NO		

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (granted_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

Índice: CREATE INDEX role_permission_reverse_idx ON esquema.role_permission(permission_id, role_id) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

permission_dependency

Dependencias entre permisos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: permission_id, required_permission_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
permission_id	uuid	NO	PK, FK	permission(id)

Campo	Tipo	NULL	Clave	Referencia física
required_permission_id	uuid	NO	PK, FK	permission(id)
created_at	timestamptz	NO		

FK (permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (required_permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: CHECK (permission_id <> required_permission_id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX permission_dependency_reverse_idx ON esquema.permission_dependency(required_permission_id, permission_id) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

user_role

Asignación de roles a usuarios.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_user_id, role_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
core_user_id	uuid	NO	PK, FK	calendar_member(core_user_id)
role_id	uuid	NO	PK, FK	role(id)
is_active	boolean	NO		
assigned_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
assigned_at	timestamptz	NO		
revoked_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
revoked_at	timestamptz	SÍ		
revocation_reason	varchar(1000)	SÍ		
version	integer	NO		

FK (core_user_id) → calendar_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (assigned_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (revoked_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: CHECK ((is_active AND revoked_at IS NULL AND revoked_by_core_user_id IS NULL AND revocation_reason IS NULL) OR (NOT is_active AND revoked_at IS NOT NULL AND revocation_reason IS NOT NULL AND btrim(revocation_reason) <> '')) · Migración 1788739200000 · línea 77

CHECK: version integer NOT NULL DEFAULT 1 CHECK (version >= 1) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX user_role_reverse_idx ON esquema.user_role(role_id, core_user_id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX user_role_active_idx ON esquema.user_role(core_user_id, role_id) WHERE is_active · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

core_inbox_event

Mensaje externo persistido antes de procesarse.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: source_event_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
source_event_id	uuid	NO	PK	
event_type	varchar(120)	NO		
schema_version	varchar(120)	NO		
aggregate_type	varchar(120)	NO		
aggregate_id	uuid	NO		
aggregate_version	integer	NO		
source_cursor	varchar(2000)	SÍ		
payload_hash	varchar(64)	NO		

Campo	Tipo	NULL	Clave	Referencia física
payload	jsonb	NO		
processing_status	varchar(120)	NO		
processing_attempts	integer	NO		
received_at	timestamptz	NO		
processed_at	timestamptz	SÍ		
last_error	varchar(2000)	SÍ		
next_attempt_at	timestamptz	SÍ		
lease_expires_at	timestamptz	SÍ		

CHECK: CHECK ((processing_status = 'processed') = (processed_at IS NOT NULL)) · Migración 1788739200000 · línea 77

CHECK: aggregate_version integer NOT NULL CHECK (aggregate_version >= 1) · Migración 1788739200000 · línea 77

CHECK: payload_hash varchar(64) NOT NULL CHECK (payload_hash ~ '^[0-9a-f]{64}$') · Migración 1788739200000 · línea 77

CHECK: processing_status varchar(120) NOT NULL DEFAULT 'pending' CHECK (processing_status IN ('pending','processing','processed','failed')) · Migración 1788739200000 · línea 77

CHECK: processing_attempts integer NOT NULL DEFAULT 0 CHECK (processing_attempts >= 0) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_inbox_pending_idx ON esquema.core_inbox_event(received_at, source_event_id) WHERE processing_status = 'pending' · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_inbox_claim_idx ON esquema.core_inbox_event(next_attempt_at, received_at, source_event_id) WHERE processing_status IN ('pending','failed') · Migración 1789603200000 · línea 16

Índice: CREATE INDEX core_inbox_lease_recovery_idx ON esquema.core_inbox_event(lease_expires_at, received_at, source_event_id) WHERE processing_status = 'processing' · Migración 1789603200000 · línea 16

Fuentes técnicas: Migración 1788739200000 · línea 77; Migración 1789603200000 · línea 16.

core_sync_checkpoint

Avance de sincronización externa.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: stream_code. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
stream_code	varchar(120)	NO	PK	
last_cursor	varchar(2000)	SÍ		
snapshot_version	varchar(2000)	SÍ		
sync_status	varchar(120)	NO		
last_attempt_at	timestamptz	SÍ		
last_success_at	timestamptz	SÍ		
last_error	varchar(2000)	SÍ		
version	integer	NO		

CHECK: CHECK (sync_status <> 'healthy' OR last_success_at IS NOT NULL) · Migración 1788739200000 · línea 77

CHECK: stream_code varchar(120) PRIMARY KEY CHECK (btrim(stream_code) <> '') · Migración 1788739200000 · línea 77

CHECK: sync_status varchar(120) NOT NULL CHECK (sync_status IN ('uninitialized','syncing','healthy','degraded','failed')) · Migración 1788739200000 · línea 77

CHECK: version integer NOT NULL DEFAULT 1 CHECK (version >= 1) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

core_space_projection

Proyección local de espacios canónicos.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_space_id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
core_space_id	uuid	NO	PK	
code	varchar(120)	NO		
name	varchar(200)	NO		
parent_core_space_id	uuid	SÍ	FK	core_space_projection(core_space_id)
sort_order	integer	NO		
type_code	varchar(120)	NO		
source_status	varchar(120)	NO		
effective_status	varchar(120)	NO		
module_selection_state	varchar(120)	NO		
module_setting_is_explicit	boolean	NO		
module_setting_version	integer	NO		
selectable	boolean	NO		
context_only	boolean	NO		
color_hex	varchar(7)	SÍ		
effective_color_hex	varchar(7)	SÍ		
path_text	varchar(4000)	NO		
path_ids	jsonb	NO		

Campo	Tipo	NULL	Clave	Referencia física
core_version	integer	NO		
catalog_sequence	bigint	NO		
is_tombstone	boolean	NO		
last_source_event_id	uuid	SÍ	FK	core_inbox_event(source_event_id)
synced_at	timestamptz	NO		

FK (parent_core_space_id) → core_space_projection(core_space_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (last_source_event_id) → core_inbox_event(source_event_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: CHECK ((module_setting_is_explicit AND module_setting_version >= 1) OR (NOT module_setting_is_explicit AND module_setting_version = 0)) · Migración 1788739200000 · línea 77

CHECK: CHECK (NOT ((context_only OR is_tombstone) AND selectable)) · Migración 1788739200000 · línea 77

CHECK: code varchar(120) NOT NULL UNIQUE CHECK (btrim(code) <> '') · Migración 1788739200000 · línea 77

CHECK: name varchar(200) NOT NULL CHECK (btrim(name) <> '') · Migración 1788739200000 · línea 77

CHECK: sort_order integer NOT NULL CHECK (sort_order >= 0) · Migración 1788739200000 · línea 77

CHECK: source_status varchar(120) NOT NULL CHECK (source_status IN ('draft','active','inactive','retired')) · Migración 1788739200000 · línea 77

CHECK: effective_status varchar(120) NOT NULL CHECK (effective_status IN ('draft','active','inactive','retired')) · Migración 1788739200000 · línea 77

CHECK: module_selection_state varchar(120) NOT NULL CHECK (module_selection_state IN ('enabled','disabled')) · Migración 1788739200000 · línea 77

CHECK: module_setting_version integer NOT NULL CHECK (module_setting_version >= 0) · Migración 1788739200000 · línea 77

CHECK: color_hex varchar(7) CHECK (color_hex ~ '^#[0-9A-Fa-f]{6}$') · Migración 1788739200000 · línea 77

CHECK: effective_color_hex varchar(7) CHECK (effective_color_hex ~ '^#[0-9A-Fa-f]{6}$') · Migración 1788739200000 · línea 77

CHECK: path_ids jsonb NOT NULL CHECK (jsonb_typeof(path_ids) = 'array') · Migración 1788739200000 · línea 77

CHECK: core_version integer NOT NULL CHECK (core_version >= 1) · Migración 1788739200000 · línea 77

CHECK: catalog_sequence bigint NOT NULL CHECK (catalog_sequence >= 0) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_space_parent_idx ON esquema.core_space_projection(parent_core_space_id, sort_order, core_space_id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_space_selectable_idx ON esquema.core_space_projection(selectable, core_space_id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_space_status_idx ON esquema.core_space_projection(effective_status, core_space_id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_space_sequence_idx ON esquema.core_space_projection(catalog_sequence, core_space_id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_space_path_idx ON esquema.core_space_projection(md5(path_text)) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

core_space_relation_projection

Proyección de relaciones espaciales.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: ancestor_core_space_id, descendant_core_space_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
ancestor_core_space_id	uuid	NO	PK, FK	core_space_projection(core_space_id)
descendant_core_space_id	uuid	NO	PK, FK	core_space_projection(core_space_id)
depth	integer	NO		
catalog_sequence	bigint	NO		
last_source_event_id	uuid	SÍ	FK	core_inbox_event(source_event_id)
synced_at	timestamptz	NO		

FK (ancestor_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (descendant_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (last_source_event_id) → core_inbox_event(source_event_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: CHECK ((ancestor_core_space_id = descendant_core_space_id) = (depth = 0)) · Migración 1788739200000 · línea 77

CHECK: depth integer NOT NULL CHECK (depth >= 0) · Migración 1788739200000 · línea 77

CHECK: catalog_sequence bigint NOT NULL CHECK (catalog_sequence >= 0) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_space_relation_reverse_idx ON esquema.core_space_relation_projection(descendant_core_space_id, ancestor_core_space_id) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

core_space_lineage_projection

Proyección de jerarquía espacial.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: core_lineage_id. Unicidad total: (source_core_space_id, target_core_space_id, relation_type).

Campo	Tipo	NULL	Clave	Referencia física
core_lineage_id	uuid	NO	PK	
source_core_space_id	uuid	NO	FK	core_space_projection(core_space_id)
target_core_space_id	uuid	NO	FK	core_space_projection(core_space_id)
relation_type	varchar(120)	NO		
effective_at	timestamptz	NO		
catalog_sequence	bigint	NO		
last_source_event_id	uuid	SÍ	FK	core_inbox_event(source_event_id)
synced_at	timestamptz	NO		

FK (source_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (target_core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (last_source_event_id) → core_inbox_event(source_event_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: CHECK (source_core_space_id <> target_core_space_id) · Migración 1788739200000 · línea 77

CHECK: relation_type varchar(120) NOT NULL CHECK (relation_type IN ('split_into','merged_into','replaced_by')) · Migración 1788739200000 · línea 77

CHECK: catalog_sequence bigint NOT NULL CHECK (catalog_sequence >= 0) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX core_space_lineage_target_idx ON esquema.core_space_lineage_projection(target_core_space_id, source_core_space_id) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

calendar_event

Evento administrado por Calendario con estado y versión.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
code	varchar(64)	NO		
name	varchar(200)	SÍ		
description	varchar(4000)	SÍ		
event_type	varchar(120)	SÍ		
status	varchar(120)	NO		
occupancy_start_at	timestamptz	SÍ		
public_start_at	timestamptz	SÍ		
public_end_at	timestamptz	SÍ		
occupancy_end_at	timestamptz	SÍ		
business_timezone	varchar(120)	NO		
confirmed_at	timestamptz	SÍ		
confirmed_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
created_by_core_user_id	uuid	NO	FK	calendar_member(core_user_id)
created_by_email_snapshot	varchar(320)	NO		
created_by_name_snapshot	varchar(200)	NO		
updated_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)

Campo	Tipo	NULL	Clave	Referencia física
updated_by_email_snapshot	varchar(320)	SÍ		
updated_by_name_snapshot	varchar(200)	SÍ		
archived_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
archived_at	timestamptz	SÍ		
cancelled_by_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
cancelled_at	timestamptz	SÍ		
cancellation_reason	varchar(1000)	SÍ		
version	integer	NO		
created_at	timestamptz	NO		
updated_at	timestamptz	NO		

FK (confirmed_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (created_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (updated_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (archived_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (cancelled_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: CHECK (status = 'draft' OR (name IS NOT NULL AND btrim(name) <> '' AND description IS NOT NULL AND btrim(description) <> '' AND event_type IS NOT NULL AND occupancy_start_at IS NOT NULL AND public_start_at IS NOT NULL AND public_end_at IS NOT NULL AND occupancy_end_at IS NOT NULL)) · Migración 1788739200000 · línea 77

CHECK: CHECK (occupancy_start_at < public_start_at) · Migración 1788739200000 · línea 77

CHECK: CHECK (occupancy_start_at < public_end_at) · Migración 1788739200000 · línea 77

CHECK: CHECK (occupancy_start_at < occupancy_end_at) · Migración 1788739200000 · línea 77

CHECK: CHECK (public_start_at < public_end_at) · Migración 1788739200000 · línea 77

CHECK: CHECK (public_start_at < occupancy_end_at) · Migración 1788739200000 · línea 77

CHECK: CHECK (public_end_at < occupancy_end_at) · Migración 1788739200000 · línea 77

CHECK: CHECK ((archived_at IS NULL AND archived_by_core_user_id IS NULL) OR (status = 'draft' AND archived_at IS NOT NULL AND archived_by_core_user_id IS NOT NULL)) · Migración 1788739200000 · línea 77

CHECK: CHECK ((confirmed_at IS NULL) = (confirmed_by_core_user_id IS NULL)) · Migración 1788739200000 · línea 77

CHECK: CHECK (status <> 'confirmed' OR confirmed_at IS NOT NULL) · Migración 1788739200000 · línea 77

CHECK: CHECK (status <> 'draft' OR confirmed_at IS NULL) · Migración 1788739200000 · línea 77

CHECK: CHECK ((status = 'cancelled' AND cancelled_at IS NOT NULL AND cancelled_by_core_user_id IS NOT NULL AND cancellation_reason IS NOT NULL AND btrim(cancellation_reason) <> '') OR (status <> 'cancelled' AND cancelled_at IS NULL AND cancelled_by_core_user_id IS NULL AND cancellation_reason IS NULL)) · Migración 1788739200000 · línea 77

CHECK: CONSTRAINT calendar_event_business_timezone_ck CHECK (business_timezone = 'America/Argentina/Buenos_Aires') · Migración 1788825600000 · línea 57

CHECK: code varchar(64) NOT NULL UNIQUE CHECK (btrim(code) <> '') · Migración 1788739200000 · línea 77

CHECK: event_type varchar(120) CHECK (event_type IN ('own','third_party')) · Migración 1788739200000 · línea 77

CHECK: status varchar(120) NOT NULL DEFAULT 'draft' CHECK (status IN ('draft','confirmed','cancelled')) · Migración 1788739200000 · línea 77

CHECK: version integer NOT NULL DEFAULT 1 CHECK (version >= 1) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_status_idx ON esquema.calendar_event(status, id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_type_idx ON esquema.calendar_event(event_type, id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_public_start_idx ON esquema.calendar_event(public_start_at, id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_occupancy_start_idx ON esquema.calendar_event(occupancy_start_at, id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_occupancy_end_idx ON esquema.calendar_event(occupancy_end_at, id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_name_idx ON esquema.calendar_event(lower(name), id) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_draft_idx ON esquema.calendar_event(updated_at, id) WHERE status = 'draft' AND archived_at IS NULL · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_event_occupancy_gist_idx ON esquema.calendar_event USING gist(tstzrange(occupancy_start_at, occupancy_end_at, '[)')) WHERE status = 'confirmed' · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

event_space

Asociación de un evento con un espacio proyectado de Core.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: event_id, core_space_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
event_id	uuid	NO	PK, FK	calendar_event(id)
core_space_id	uuid	NO	PK, FK	core_space_projection(core_space_id)
space_code_snapshot	varchar(120)	NO		
space_name_snapshot	varchar(200)	NO		
space_path_snapshot	varchar(4000)	NO		
core_space_version_snapshot	integer	NO		
sort_order	integer	NO		
associated_by_core_user_id	uuid	NO	FK	calendar_member(core_user_id)
associated_at	timestamptz	NO		

FK (event_id) → calendar_event(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (core_space_id) → core_space_projection(core_space_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (associated_by_core_user_id) → calendar_member(core_user_id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: core_space_version_snapshot integer NOT NULL CHECK (core_space_version_snapshot >= 1) · Migración 1788739200000 · línea 77

CHECK: sort_order integer NOT NULL CHECK (sort_order >= 0) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX event_space_reverse_idx ON esquema.event_space(core_space_id, event_id) · Migración 1788739200000 · línea 77

Fuentes técnicas: Migración 1788739200000 · línea 77.

calendar_audit_log

Cabecera particionada de auditoría de Calendario.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: occurred_at, id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
occurred_at	timestamptz	NO	PK	
calendar_event_id	uuid	SÍ	FK	calendar_event(id)
domain	varchar(120)	NO		
action	varchar(120)	NO		
target_type	varchar(120)	NO		
target_id	varchar(120)	NO		
actor_type	varchar(120)	NO		
actor_core_user_id	uuid	SÍ	FK	calendar_member(core_user_id)
actor_service_id	varchar(120)	SÍ		
actor_email_snapshot	varchar(320)	SÍ		
actor_name_snapshot	varchar(200)	SÍ		
actor_roles_snapshot	jsonb	NO		
outcome	varchar(120)	NO		
reason	varchar(1000)	SÍ		
source_session_id	uuid	SÍ		
request_id	varchar(120)	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
correlation_id	varchar(120)	SÍ		
source_ip	inet	SÍ		
user_agent	varchar(2000)	SÍ		
required_protection_level	integer	NO		

FK (calendar_event_id) → calendar_event(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

FK (actor_core_user_id) → calendar_member(core_user_id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1788739200000 · línea 77.

CHECK: CHECK ((actor_type = 'user' AND actor_core_user_id IS NOT NULL AND actor_service_id IS NULL) OR (actor_type = 'service' AND actor_core_user_id IS NULL AND actor_service_id IS NOT NULL AND btrim(actor_service_id) <> '') OR (actor_type = 'system' AND actor_core_user_id IS NULL AND actor_service_id IS NULL)) · Migración 1788739200000 · línea 77

CHECK: CONSTRAINT calendar_audit_required_protection_level_ck CHECK (required_protection_level IN (0,50,100)) · Migración 1788912000000 · línea 93

CHECK: actor_type varchar(120) NOT NULL CHECK (actor_type IN ('user','service','system')) · Migración 1788739200000 · línea 77

CHECK: actor_roles_snapshot jsonb NOT NULL DEFAULT '[]'::jsonb CHECK (jsonb_typeof(actor_roles_snapshot) = 'array') · Migración 1788739200000 · línea 77

CHECK: outcome varchar(120) NOT NULL CHECK (outcome IN ('succeeded','denied','failed')) · Migración 1788739200000 · línea 77

Índice: CREATE INDEX calendar_audit_event_idx ON esquema.calendar_audit_log(calendar_event_id, occurred_at, id) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_audit_actor_idx ON esquema.calendar_audit_log(actor_core_user_id, occurred_at, id) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_audit_domain_idx ON esquema.calendar_audit_log(domain, occurred_at, id) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_audit_action_idx ON esquema.calendar_audit_log(action, occurred_at, id) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_audit_correlation_idx ON esquema.calendar_audit_log(correlation_id, occurred_at, id) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_audit_protection_idx ON esquema.calendar_audit_log(required_protection_level,occurred_at DESC,id DESC) · Migración 1788912000000 · línea 93

Índice: CREATE INDEX calendar_audit_event_protection_idx ON esquema.calendar_audit_log(calendar_event_id,required_protection_level,occurred_at DESC,id DESC) WHERE calendar_event_id IS NOT NULL · Migración 1788912000000 · línea 93

Fuentes técnicas: Migración 1788739200000 · línea 77; Migración 1788912000000 · línea 55.

calendar_audit_change

Cambio de campo asociado a una cabecera de auditoría.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
audit_log_id	uuid	NO	FK	calendar_audit_log(occurred_at, id)
audit_occurred_at	timestamptz	NO	FK	calendar_audit_log(occurred_at, id)
field_code	varchar(120)	NO		
field_label_snapshot	varchar(160)	NO		
old_value	jsonb	SÍ		
new_value	jsonb	SÍ		
sort_order	integer	NO		

FK (audit_occurred_at, audit_log_id) → calendar_audit_log(occurred_at, id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 281.

CHECK: sort_order integer NOT NULL CHECK (sort_order >= 0) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_audit_change_header_idx ON esquema.calendar_audit_change(audit_occurred_at, audit_log_id, sort_order) · Migración 1788739200000 · línea 281

Fuentes técnicas: Migración 1788739200000 · línea 281.

calendar_outbox_event

Mensaje transaccional de un evento de Calendario pendiente de publicación.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: event_id. Unicidad total: (sequence); (calendar_event_id, aggregate_version).

Campo	Tipo	NULL	Clave	Referencia física
event_id	uuid	NO	PK	
sequence	bigint	NO		
calendar_event_id	uuid	NO	FK	calendar_event(id)
event_type	varchar(120)	NO		
schema_version	varchar(120)	NO		
aggregate_version	integer	NO		
correlation_id	varchar(120)	SÍ		
causation_id	varchar(120)	SÍ		
payload	jsonb	NO		
publication_status	varchar(120)	NO		
publication_attempts	integer	NO		
available_at	timestamptz	NO		
next_attempt_at	timestamptz	SÍ		
published_at	timestamptz	SÍ		
last_error	varchar(2000)	SÍ		
created_at	timestamptz	NO		
lease_expires_at	timestamptz	SÍ		

FK (calendar_event_id) → calendar_event(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1788739200000 · línea 281.

CHECK: CHECK ((publication_status = 'published') = (published_at IS NOT NULL)) · Migración 1788739200000 · línea 281

CHECK: aggregate_version integer NOT NULL CHECK (aggregate_version >= 1) · Migración 1788739200000 · línea 281

CHECK: publication_status varchar(120) NOT NULL DEFAULT 'pending' CHECK (publication_status IN ('pending','publishing','published','failed')) · Migración 1788739200000 · línea 281

CHECK: publication_attempts integer NOT NULL DEFAULT 0 CHECK (publication_attempts >= 0) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_outbox_eligible_idx ON esquema.calendar_outbox_event(publication_status, available_at, next_attempt_at, sequence) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_outbox_aggregate_idx ON esquema.calendar_outbox_event(calendar_event_id, aggregate_version, sequence) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX calendar_outbox_lease_recovery_idx ON esquema.calendar_outbox_event(lease_expires_at, sequence) WHERE publication_status = 'publishing' · Migración 1789603200000 · línea 16

Fuentes técnicas: Migración 1788739200000 · línea 281; Migración 1789603200000 · línea 16.

idempotency_record

Control de repetición de solicitudes.

Estructura verificada en código y migraciones. Esquema configurable del módulo.

PK: id. Unicidad total: (principal_type, principal_id, operation, idempotency_key).

Campo	Tipo	NULL	Clave	Referencia física
id	uuid	NO	PK	
principal_type	varchar(120)	NO		
principal_id	varchar(120)	NO		
operation	varchar(120)	NO		
idempotency_key	varchar(120)	NO		
request_hash	varchar(64)	NO		
processing_status	varchar(120)	NO		
response_status	integer	SÍ		
response_body	jsonb	SÍ		

Campo	Tipo	NULL	Clave	Referencia física
resource_type	varchar(120)	SÍ		
resource_id	uuid	SÍ		
created_at	timestamptz	NO		
completed_at	timestamptz	SÍ		
expires_at	timestamptz	NO		

CHECK: CHECK (expires_at > created_at) · Migración 1788739200000 · línea 281

CHECK: CHECK ((processing_status = 'in_progress' AND completed_at IS NULL AND response_status IS NULL AND response_body IS NULL) OR (processing_status IN ('completed','failed') AND completed_at IS NOT NULL AND response_status IS NOT NULL)) · Migración 1788739200000 · línea 281

CHECK: principal_type varchar(120) NOT NULL CHECK (principal_type IN ('user','service')) · Migración 1788739200000 · línea 281

CHECK: principal_id varchar(120) NOT NULL CHECK (btrim(principal_id) <> '') · Migración 1788739200000 · línea 281

CHECK: operation varchar(120) NOT NULL CHECK (btrim(operation) <> '') · Migración 1788739200000 · línea 281

CHECK: idempotency_key varchar(120) NOT NULL CHECK (btrim(idempotency_key) <> '') · Migración 1788739200000 · línea 281

CHECK: request_hash varchar(64) NOT NULL CHECK (request_hash ~ '^[0-9a-f]{64}$') · Migración 1788739200000 · línea 281

CHECK: processing_status varchar(120) NOT NULL DEFAULT 'in_progress' CHECK (processing_status IN ('in_progress','completed','failed')) · Migración 1788739200000 · línea 281

CHECK: response_status integer CHECK (response_status BETWEEN 100 AND 599) · Migración 1788739200000 · línea 281

Índice: CREATE INDEX idempotency_expiry_idx ON esquema.idempotency_record(expires_at) · Migración 1788739200000 · línea 281

Fuentes técnicas: Migración 1788739200000 · línea 281.

30 tablas lógicas · 40 FK locales · 6 láminas de dominio.

Inventario físico y ORM

Se reconstruyeron 30 tablas a partir de ocho migraciones. Hay 29 clases @Entity más activity_order_destination_sector, tabla puente declarada con @JoinTable y creada por el baseline. La PK compuesta impide repetir el mismo sector de destino dentro del mismo pedido. La relación muchos a muchos se materializa en esta tabla.

Identidad y áreas propias

users utiliza IDs enteros locales. La autenticación admite identidad Azure y emisión de token del módulo; no se verificó una correspondencia persistida a platform_user de Core. Las áreas operativas delimitan permisos y recursos; no equivalen automáticamente a espacios canónicos. La migración multirrol elimina UNIQUE(user_id) y agrega UNIQUE(user_id,role_id): un usuario puede tener varios roles distintos.

Materiales y existencias

materials contiene stock total numeric(11,2); warehouse_stock y varias cantidades operativas usan integer. Esta diferencia de tipos está verificada y no acredita por sí sola una política uniforme de cantidades fraccionarias. Las migraciones no declaran una unicidad (materialId,materialLocationInSectorId) en warehouse_stock: el DER no impone una sola fila por par. material_packing_profile.material_id es FK NOT NULL y UNIQUE, por lo que un material tiene cero o un perfil y cada perfil pertenece a un material.

Actividades locales y pedidos

memo_activity y sector_activity pertenecen al modelo local de Depósito. activity_order se relaciona con memo_activity mediante activity_id, y con sectores de destino mediante la tabla puente. No se verificó calendar_event_id ni una equivalencia con eventos de Calendario. counter_order.destinationSector es texto; no se dibuja una FK a sector para ese campo. Las ubicaciones de origen de sus renglones sí tienen FK verificadas.

Historia y referencias sin FK

activity_order_history.materialId y fromSectorId son campos escalares, sin FK hacia materials/sector en las migraciones ni relación ORM explícita. Se interpretan como referencias históricas por sus nombres y contexto (inferido), sin cardinalidad de base acreditada. responsible y etiquetas históricas tampoco se convierten en FK a users. Los historiales, devoluciones y pérdidas mantienen las FK que sí declara el baseline.

Auditoría con destino polimórfico

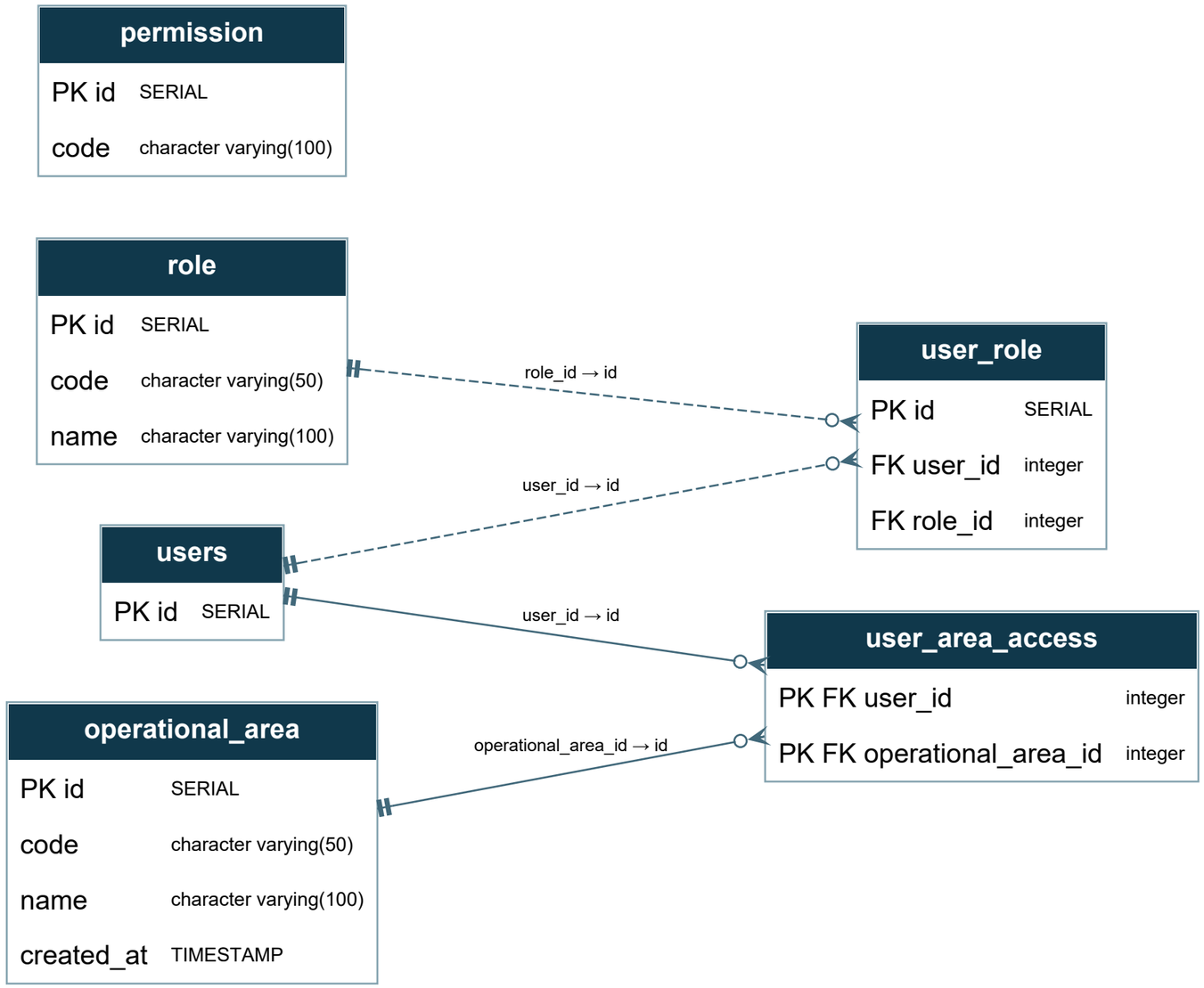
rbac_audit_log.actor_user_id no tiene FK a users. target_type y target_id identifican un destino variable (role o user en el tipo de aplicación); no existe una FK polimórfica ni un CHECK SQL que transforme ese varchar en enum. actor_email y target_label preservan contexto histórico. Se documenta la referencia lógica sin dibujar claves inexistentes.

Esquema y enumeraciones

El proveedor no fija schema y las tablas del baseline se crean sin calificación: public es una hipótesis de ubicación, cuya ubicación efectiva depende de search_path. Los tipos enum sí se califican como public. La migración de sectores retira Reparación mediante un tipo temporal y lo renombra a sector_sector_enum; ese nombre final es el que usa el diccionario. La migración posterior de actividades agrega Convención.

Depósito · Identidad, áreas y acceso · 1

La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Depósito · Inventario · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

operational_area · referencia

PK id	SERIAL

sector

PK id	SERIAL
FK management_area_id	integer ?
name	character varying(255)

warehouse_stock

PK id	SERIAL
FK materialId	integer ?
FK materialLocationInSectorId	integer ?
storaged_stock	integer

materials_category

PK id	SERIAL
FK operational_area_id	integer

materials

PK id	SERIAL
FK categoryId	integer ?
FK operational_area_id	integer
name	character varying(80) ?
code	character varying(150)
actual_stock	numeric(11,2)
minimum_stock	integer
reorder_point	integer ?
critical_stock	integer ?
is_one_use	boolean

stockchange

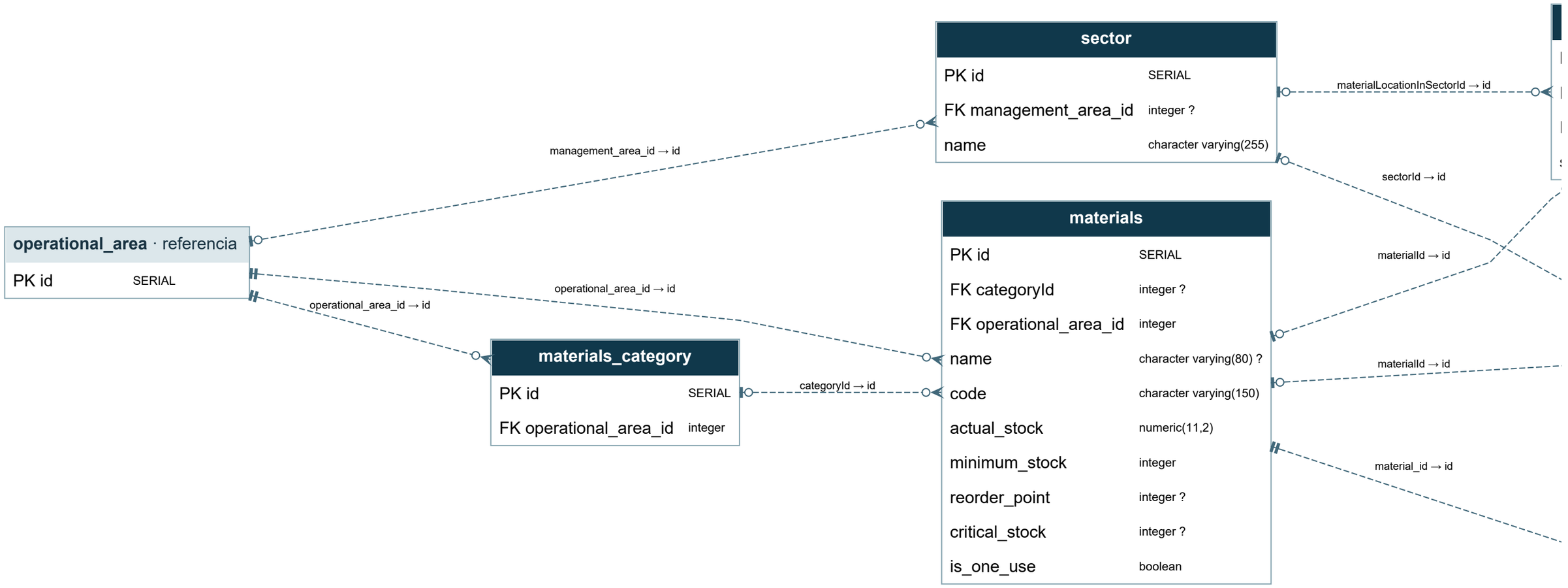
PK id	SERIAL
FK materialId	integer ?
FK sectorId	integer ?
changeAmount	integer
previousStock	integer
actualStock	integer
changeDate	TIMESTAMP

material_packing_profile

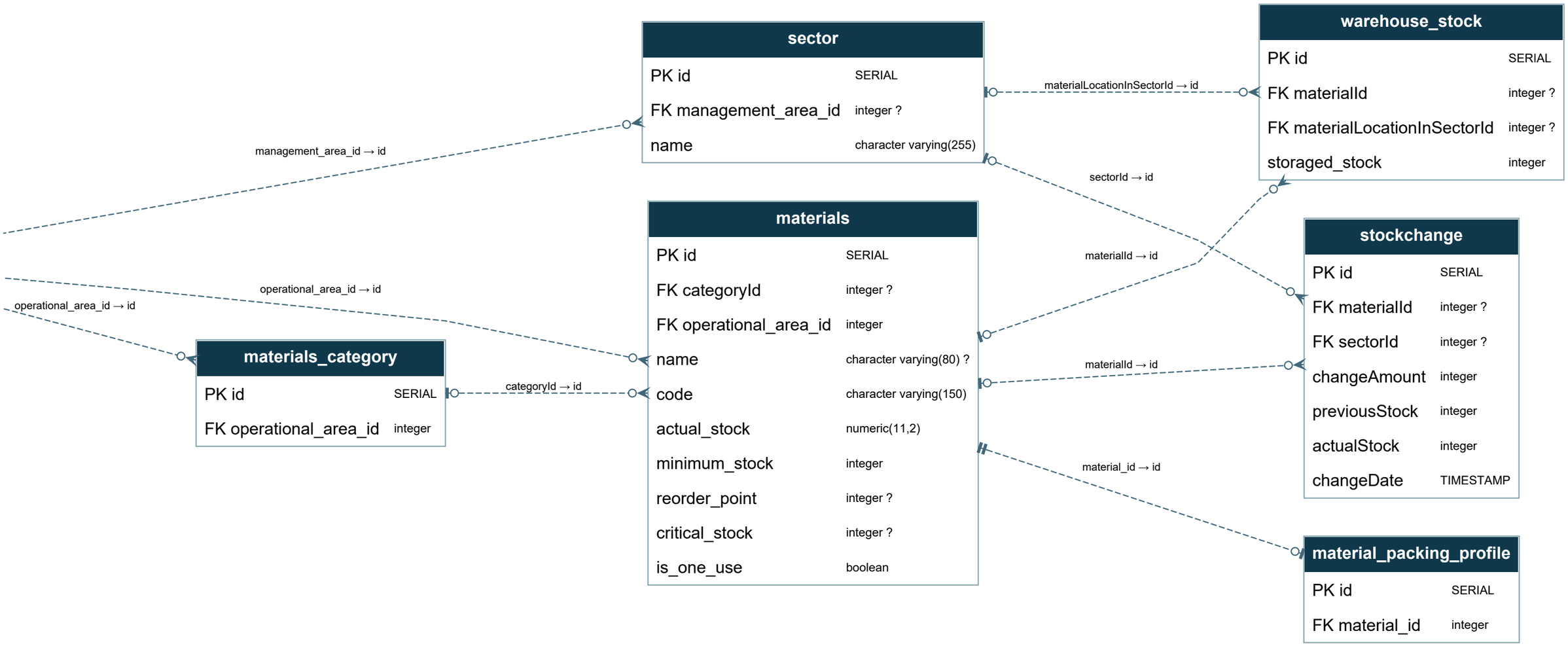
PK id	SERIAL
FK material_id	integer

management_area_id → id
operational_area_id → id
operational_area_id → id
categoryId → id
materialLocationInSectorId → id
sectorId → id
materialId → id
materialId → id
material_id → id

Depósito · Inventario · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Depósito · Inventario · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Depósito · Actividades y pedidos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

memo_activity

PK id	SERIAL
state	public.memo_activity_state_enum
activity_name	character varying(50)
initial_date	date
end_date	date
type_activity	public.memo_activity_type_activity_enum

sector_activity

PK id	SERIAL
FK sector_id	integer ?
FK memo_activity_id	integer ?
is_partially_rented	boolean
square_meters_rented	numeric(10,2) ?

sector · referencia

PK id	SERIAL

activity_order_destination_sector

PK FK activity_order_id	integer
PK FK sector_id	integer

users · referencia

PK id	SERIAL

operational_area · referencia

PK id	SERIAL

activity_order

PK id	SERIAL
FK activity_id	integer ?
FK operational_area_id	integer
FK created_by_user_id	integer ?
status	public.activity_order_status_enum
responsible	character varying(255)

activity_order_history

PK id	SERIAL
FK activity_order_id	integer ?
materialId	integer
fromSectorId	integer ?
previousAmount	integer
newAmount	integer
date	TIMESTAMP

materials · referencia

PK id	SERIAL

activity_order_item

PK id	SERIAL
FK materialId	integer ?
FK materialLocationInSectorId	integer ?
FK activityOrderId	integer ?
status	public.activity_order_item_status_enum
amount	integer
is_discounted	boolean

memo_activity_id → id
sector_id → id
activity_id → id
sector_id → id
activity_order_id → id
materialLocationInSectorId → id
created_by_user_id → id
operational_area_id → id
activity_order_id → id
activityOrderId → id
materialId → id

Depósito · Actividades y pedidos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

memo_activity

PK id	SERIAL
state	public.memo_activity_state_enum
activity_name	character varying(50)
initial_date	date
end_date	date
type_activity	public.memo_activity_type_activity_enum

sector_activity

PK id	SE
FK sector_id	inte
FK memo_activity_id	inte
is_partially_rented	boo
square_meters_rented	nu

sector · referencia

PK id	SERIAL

activity_order_destinatio

PK FK activity_order_id	
PK FK sector_id	

activity_order

PK id	SERIAL
FK activity_id	integer ?
FK operational_area_id	integer
FK created_by_user_id	integer ?
status	public.activity_order_status_enum
responsible	character varying(255)

activity_order_hist

PK id	SEI
FK activity_order_id	inte
materialId	inte
fromSectorId	inte
previousAmount	inte
newAmount	inte
date	TIM

users · referencia

PK id	SERIAL

operational_area · referencia

PK id	SERIAL

activity_order_ite

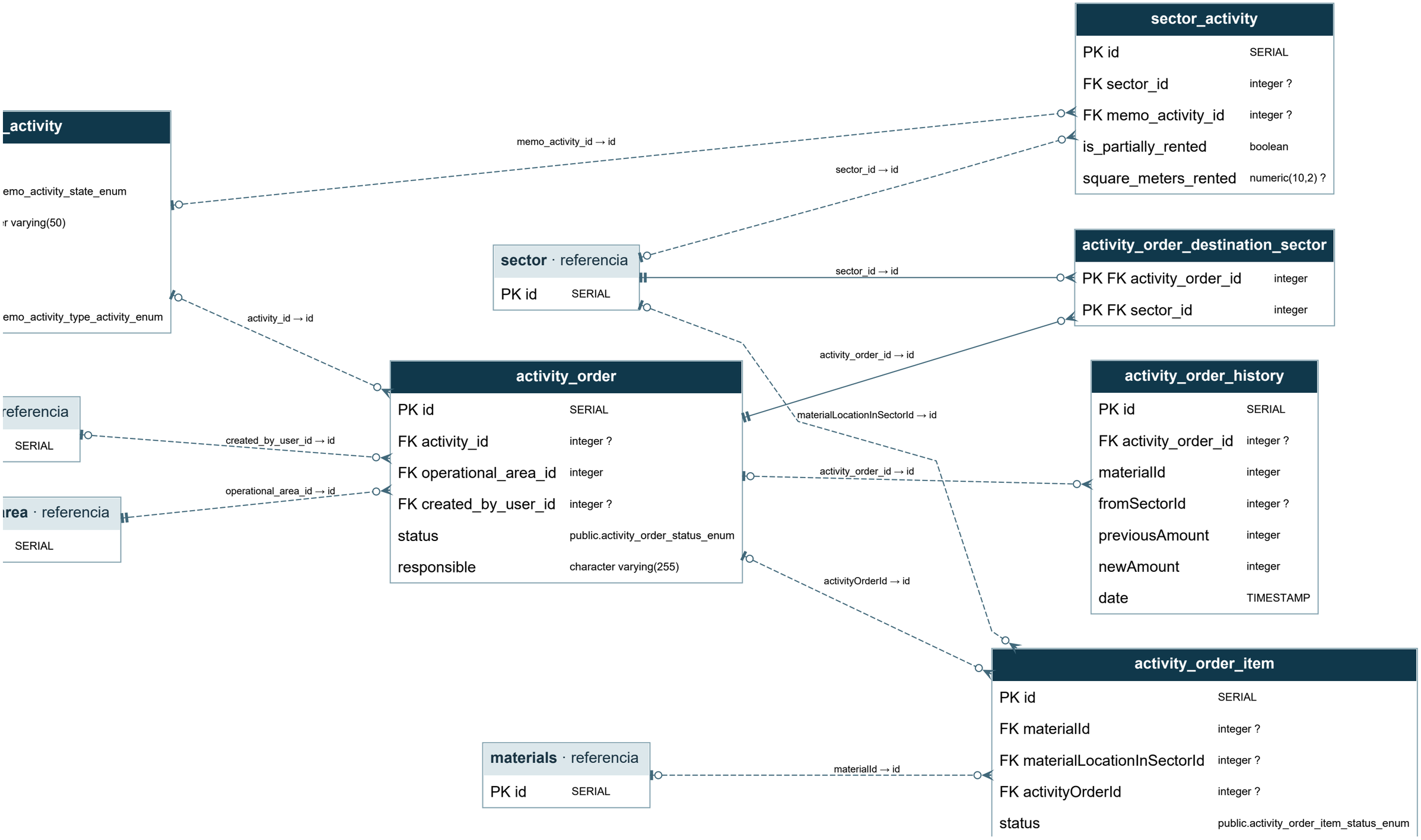
PK id	SERIAL
FK materialId	integer ?
FK materialLocationInSectorId	integer ?
FK activityOrderId	integer ?
status	public.act

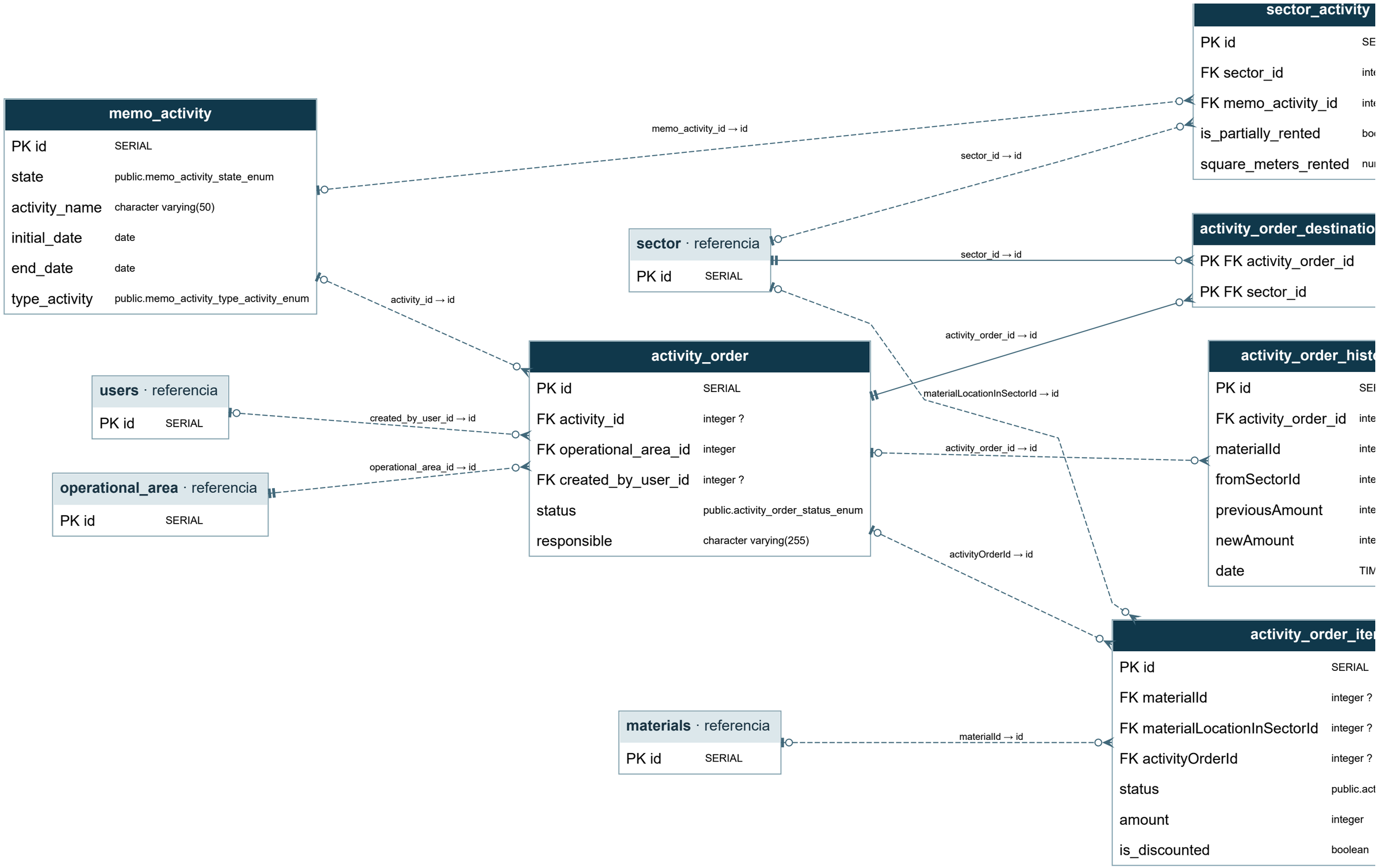
materials · referencia

PK id	SERIAL

memo_activity_id → id
sector_id → id
activity_id → id
sector_id → id
activity_order_id → id
materialLocationInSectorId → id
created_by_user_id → id
operational_area_id → id
activity_order_id → id
activityOrderId → id
materialId → id

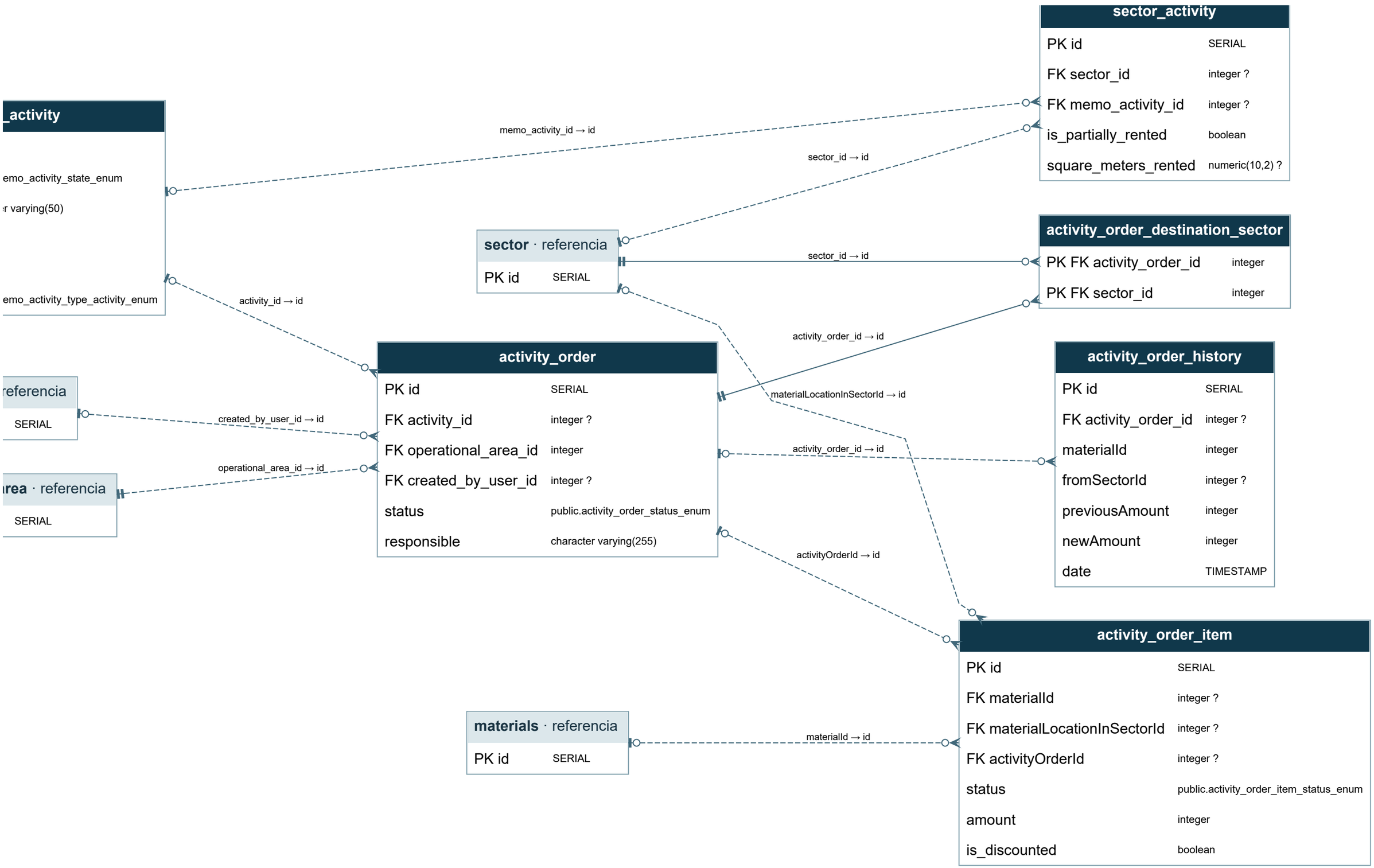
Depósito · Actividades y pedidos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



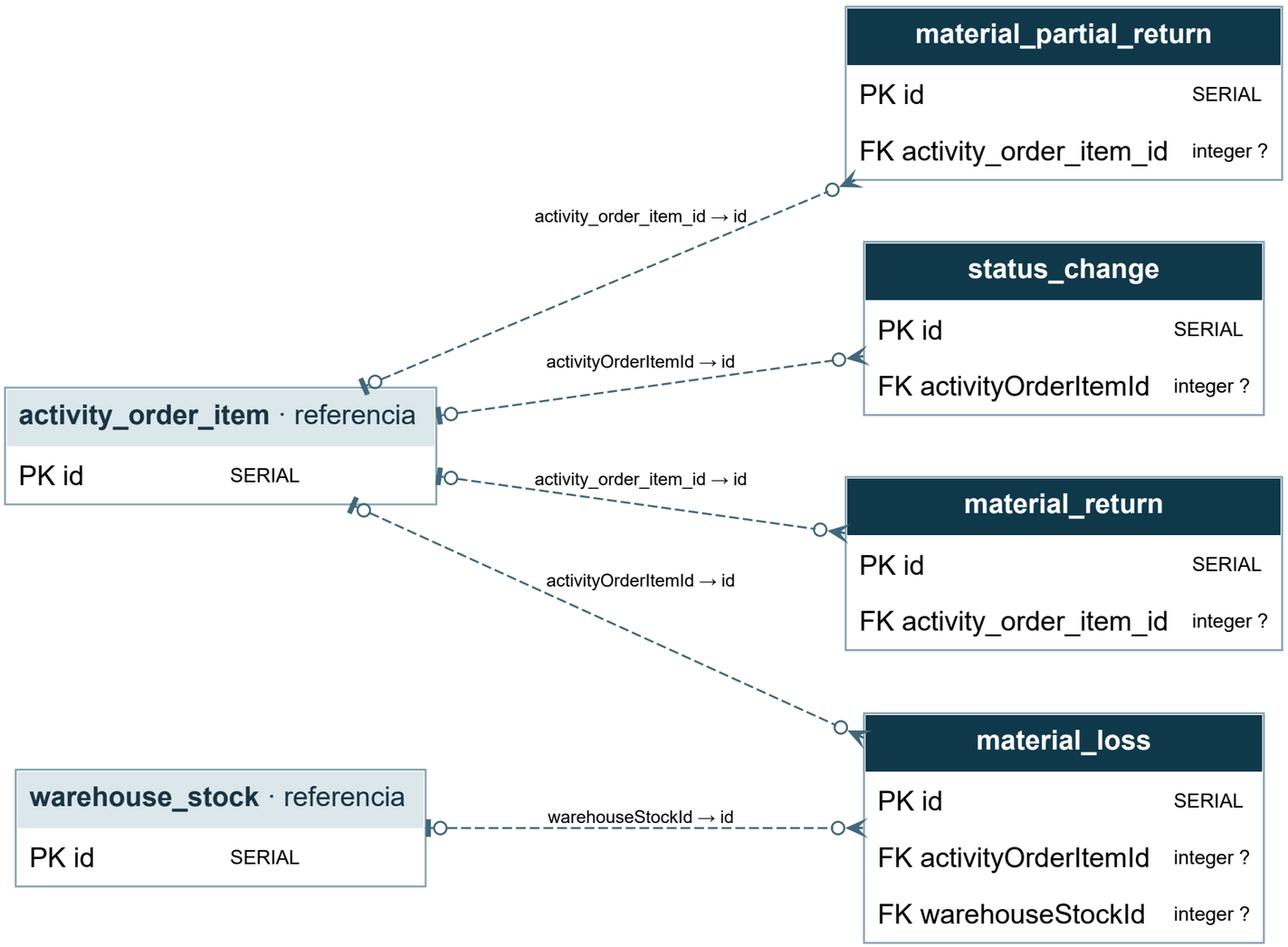

memo_activity
PK id SERIAL
state public.memo_activity_state_enum
activity_name character varying(50)
initial_date date
end_date date
type_activity public.memo_activity_type_activity_enum
sector_activity
PK id SE
FK sector_id inte
FK memo_activity_id inte
is_partially_rented boo
square_meters_rented nu
memo_activity_id → id
sector_id → id
sector · referencia
PK id SERIAL
activity_id → id
activity_order_destinatio
PK FK activity_order_id
PK FK sector_id
sector_id → id
activity_order_id → id
users · referencia
PK id SERIAL
created_by_user_id → id
activity_order
PK id SERIAL
FK activity_id integer ?
FK operational_area_id integer
FK created_by_user_id integer ?
status public.activity_order_status_enum
responsible character varying(255)
materialLocationInSectorId → id
activity_order_hist
PK id SEI
FK activity_order_id inte
materialId inte
fromSectorId inte
previousAmount inte
newAmount inte
date TIM
activity_order_id → id
operational_area_id → id
operational_area · referencia
PK id SERIAL
activityOrderId → id
activity_order_ite
PK id SERIAL
FK materialId integer ?
FK materialLocationInSectorId integer ?
FK activityOrderId integer ?
status public.act
amount integer
is_discounted boolean
materials · referencia
PK id SERIAL
materialId → id

La Rural · modelo de datos

Depósito · Actividades y pedidos · 1 · ampliación 2/2, 2/2



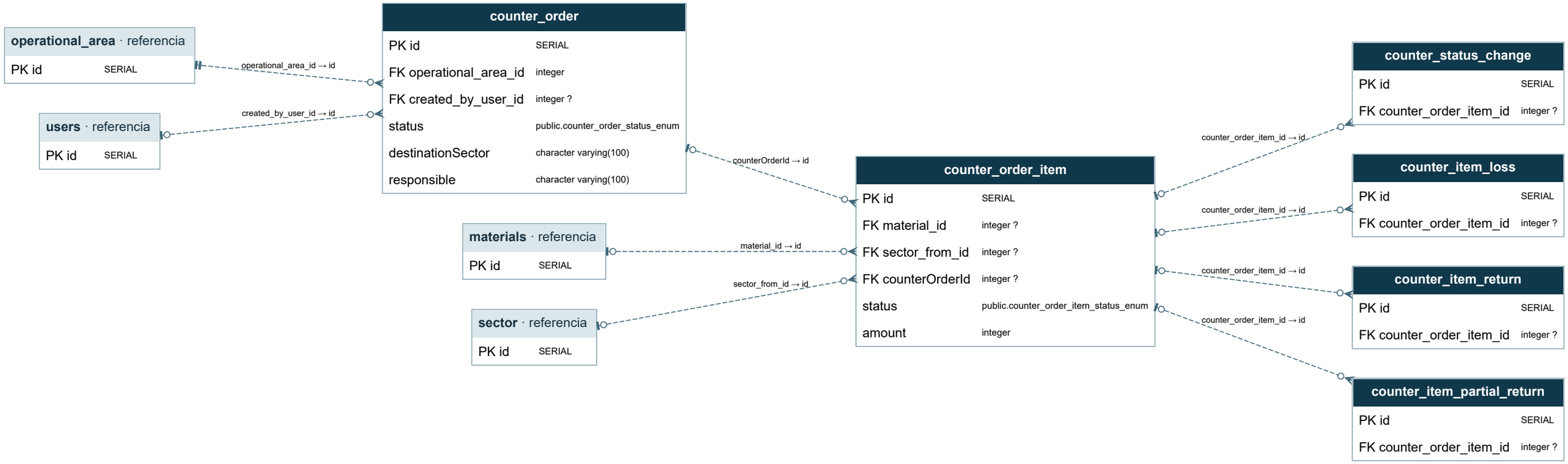
Depósito · Trazabilidad de pedidos · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



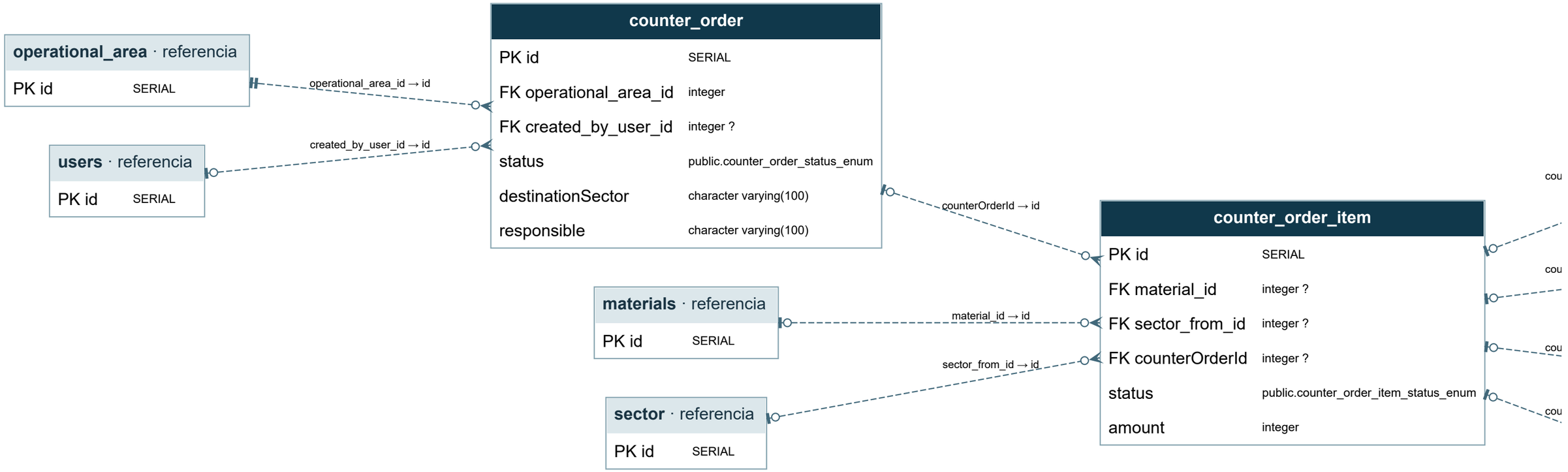
La Rural · modelo de datos

Depósito · Mostrador · 1

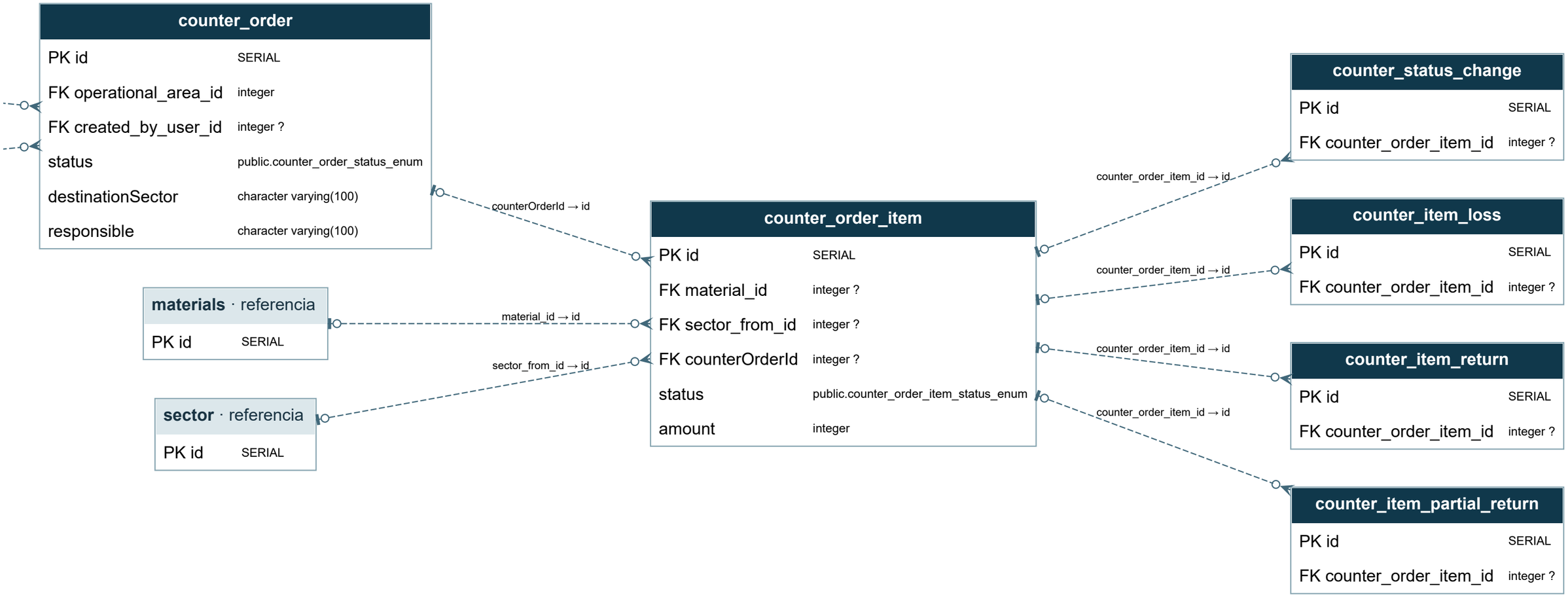
Depósito · Mostrador · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Depósito · Mostrador · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones

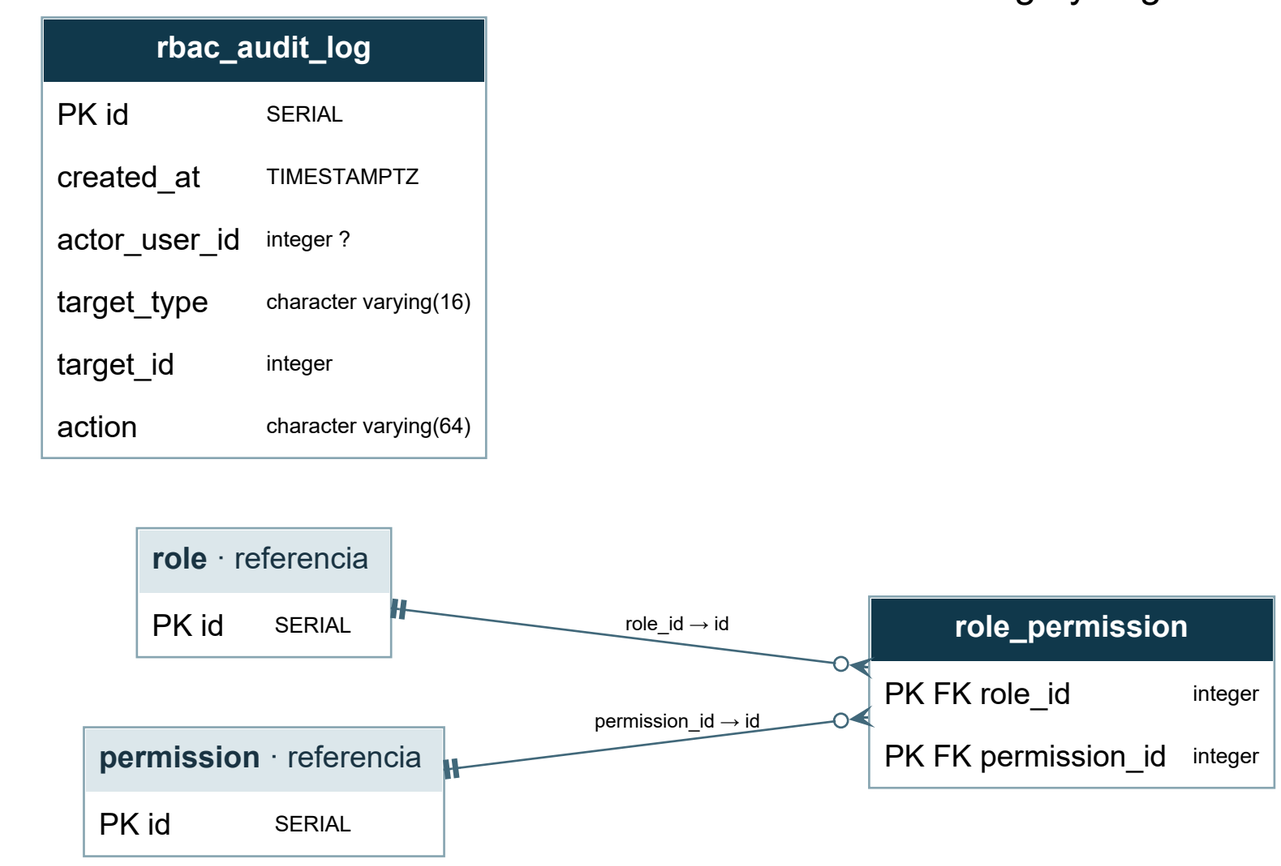


Depósito · Mostrador · 1
La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



Depósito · Auditoría de acceso · 1

La Rural · 20/11/2026 · estructura reconstruida desde código y migraciones



users

Identidad y credenciales propias de Depósito.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: (username).

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
username	character varying(255)	SÍ		
email	character varying(255)	NO		
createdAt	TIMESTAMP	NO		
updatedAt	TIMESTAMP	NO		
password	character varying(255)	SÍ		
phone_number	integer	SÍ		
is_active	boolean	NO		

Fuentes técnicas: Migración 1776366865007 · línea 10.

stockchange

Movimiento histórico de variación de stock.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
responsible	character varying(255)	NO		

Campo	Tipo	NULL	Clave	Referencia física
changeAmount	integer	NO		
previousStock	integer	NO		
actualStock	integer	NO		
changeDate	TIMESTAMP	NO		
actionType	character varying(255)	NO		
message	character varying(255)	SÍ		
invoiceNumber	character varying(255)	SÍ		
materialId	integer	SÍ	FK	materials(id)
sectorId	integer	SÍ	FK	sector(id)

FK (materialId) → materials(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 133.

FK (sectorId) → sector(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 136.

Fuentes técnicas: Migración 1776366865007 · línea 13.

operational_area

Área operativa que delimita autorizaciones y recursos de Depósito.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
code	character varying(50)	NO		
name	character varying(100)	NO		

Campo	Tipo	NULL	Clave	Referencia física
is_active	boolean	NO		
created_at	TIMESTAMP	NO		
updated_at	TIMESTAMP	NO		

Fuentes técnicas: Migración 1776366865007 · línea 16.

sector

Sector local utilizado como ubicación o destino operativo.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
name	character varying(255)	NO		
square_meters	numeric(10,0)	NO		
number_of_bathrooms	smallint	SÍ		
description	character varying(255)	SÍ		
sector	public.sector_sector_enum	NO		
is_active	boolean	NO		
color	character(1)	SÍ		
management_area_id	integer	SÍ	FK	operational_area(id)

FK (management_area_id) → operational_area(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 139.

Fuentes técnicas: Migración 1776366865007 · línea 22.

sector_activity

Asociación entre actividad local y sector ocupado.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: (sector_id, memo_activity_id).

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
is_partially_rented	boolean	NO		
square_meters_rented	numeric(10,2)	SÍ		
sector_id	integer	SÍ	FK	sector(id)
memo_activity_id	integer	SÍ	FK	memo_activity(id)

FK (sector_id) → sector(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 142.

FK (memo_activity_id) → memo_activity(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 145.

Índice: CREATE UNIQUE INDEX IDX_1090befa11bc95144edba9ef57 ON esquema.sector_activity (sector_id, memo_activity_id) · Migración 1776366865007 · línea 28

Fuentes técnicas: Migración 1776366865007 · línea 25.

memo_activity

Actividad local de Depósito con fechas y datos de cliente.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
createdAt	TIMESTAMP	NO		

Campo	Tipo	NULL	Clave	Referencia física
updatedAt	TIMESTAMP	NO		
activity_name	character varying(50)	NO		
initial_date	date	NO		
initial_time	TIME	NO		
end_date	date	NO		
end_time	TIME	NO		
opening_date	date	NO		
opening_time	TIME	NO		
closing_date	date	NO		
closing_time	TIME	NO		
type_activity	public.memo_activity_type_activity_enum	NO		
state	public.memo_activity_state_enum	NO		
is_active	boolean	NO		
client_name	character varying(20)	SÍ		
client_phone	character varying(20)	SÍ		
client_email	character varying(50)	NO		

Fuentes técnicas: Migración 1776366865007 · línea 37.

activity_order_history

Historial de cantidades y origen registrado para un pedido.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
responsible	character varying(255)	NO		
date	TIMESTAMP	NO		
materialId	integer	NO		
fromSectorId	integer	SÍ		
previousAmount	integer	NO		
newAmount	integer	NO		
activity_order_id	integer	SÍ	FK	activity_order(id)

FK (activity_order_id) → activity_order(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 148.

Fuentes técnicas: Migración 1776366865007 · línea 40.

activity_order

Pedido de materiales destinado a una actividad local.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	

Campo	Tipo	NULL	Clave	Referencia física
responsible	character varying(255)	NO		
holder	character varying(255)	SÍ		
createdAt	TIMESTAMP	NO		
status	public.activity_order_status_enum	NO		
is_active	boolean	NO		
activity_id	integer	SÍ	FK	memo_activity(id)
operational_area_id	integer	NO	FK	operational_area(id)
created_by_user_id	integer	SÍ	FK	users(id)

FK (activity_id) → memo_activity(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 151.

FK (operational_area_id) → operational_area(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 154.

FK (created_by_user_id) → users(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 157.

Fuentes técnicas: Migración 1776366865007 · línea 46.

status_change

Historial de estados de un renglón de pedido de actividad.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
newStatus	public.status_change_newstatus_enum	NO		

Campo	Tipo	NULL	Clave	Referencia física
updatedAt	TIMESTAMP	NO		
responsible	character varying(255)	NO		
activityOrderItemId	integer	SÍ	FK	activity_order_item(id)

FK (activityOrderItemId) → activity_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 160.

Fuentes técnicas: Migración 1776366865007 · línea 52.

material_loss

Pérdida de material vinculable a un renglón y a una existencia.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
reason	character varying(255)	NO		
responsible	character varying(255)	NO		
date	TIMESTAMP	NO		
additionalInfo	text	SÍ		
imageUrls	text	SÍ		
isReturned	boolean	NO		
activityOrderItemId	integer	SÍ	FK	activity_order_item(id)
warehouseStockId	integer	SÍ	FK	warehouse_stock(id)

FK (activityOrderItemId) → activity_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 163.

FK (warehouseStockId) → warehouse_stock(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 166.

Fuentes técnicas: Migración 1776366865007 · línea 55.

material_return

Reingreso asociado a un renglón de pedido de actividad.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
observations	character varying(255)	NO		
responsible	character varying(255)	NO		
date	TIMESTAMP	NO		
activity_order_item_id	integer	SÍ	FK	activity_order_item(id)

FK (activity_order_item_id) → activity_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 169.

Fuentes técnicas: Migración 1776366865007 · línea 58.

material_partial_return

Reingreso parcial asociado a un renglón de pedido de actividad.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
responsible	character varying(255)	NO		
date	TIMESTAMP	NO		
activity_order_item_id	integer	SÍ	FK	activity_order_item(id)

FK (activity_order_item_id) → activity_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 172.

Fuentes técnicas: Migración 1776366865007 · línea 61.

activity_order_item

Renglón de material y ubicación de origen de un pedido de actividad.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
status	public.activity_order_item_status_enum	NO		

Campo	Tipo	NULL	Clave	Referencia física
is_discounted	boolean	NO		
materialId	integer	SÍ	FK	materials(id)
materialLocationInSectorId	integer	SÍ	FK	sector(id)
activityOrderId	integer	SÍ	FK	activity_order(id)

FK (materialId) → materials(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 175.

FK (materialLocationInSectorId) → sector(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 178.

FK (activityOrderId) → activity_order(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 181.

Fuentes técnicas: Migración 1776366865007 · línea 67.

materials_category

Categoría de material perteneciente a un área operativa.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
category_name	character varying(25)	NO		
operational_area_id	integer	NO	FK	operational_area(id)

FK (operational_area_id) → operational_area(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 184.

Fuentes técnicas: Migración 1776366865007 · línea 70.

material_packing_profile

Perfil opcional de embalaje de un material.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: (material_id).

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
liters	numeric(10,4)	SÍ		
meters	numeric(10,4)	SÍ		
units_per_package	integer	SÍ		
material_id	integer	NO	FK	materials(id)

FK (material_id) → materials(id). Padres por hija: 1; hijas por padre: 0..1. Migración 1776366865007 · línea 187.

Fuentes técnicas: Migración 1776366865007 · línea 73.

materials

Material de inventario, con stock total y parámetros de reposición.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
name	character varying(80)	SÍ		
description	character varying(150)	SÍ		
code	character varying(150)	NO		

Campo	Tipo	NULL	Clave	Referencia física
color	character varying(30)	NO		
image_url	character varying(255)	SÍ		
weight	character varying(255)	NO		
width	character varying(255)	NO		
height	character varying(255)	NO		
depth	character varying(255)	NO		
actual_stock	numeric(11,2)	NO		
last_stock_update	TIMESTAMP	NO		
observations	character varying(250)	NO		
price	bigint	NO		
is_active	boolean	NO		
is_one_use	boolean	NO		
minimum_stock	integer	NO		
reorder_point	integer	SÍ		
critical_stock	integer	SÍ		
categoryId	integer	SÍ	FK	materials_category(id)
operational_area_id	integer	NO	FK	operational_area(id)

FK (categoryId) → materials_category(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 190.

FK (operational_area_id) → operational_area(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 193.

Fuentes técnicas: Migración 1776366865007 · línea 76.

warehouse_stock

Existencia de un material en un sector.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
storaged_stock	integer	NO		
materialId	integer	SÍ	FK	materials(id)
materialLocationInSectorId	integer	SÍ	FK	sector(id)

FK (materialId) → materials(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 196.

FK (materialLocationInSectorId) → sector(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 199.

Fuentes técnicas: Migración 1776366865007 · línea 79.

counter_status_change

Historial de estados de un renglón de mostrador.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
newStatus	public.counter_status_change_newstatus_enum	NO		
updatedAt	TIMESTAMP	NO		
responsible	character varying(255)	NO		

Campo	Tipo	NULL	Clave	Referencia física
counter_order_item_id	integer	SÍ	FK	counter_order_item(id)

FK (counter_order_item_id) → counter_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 202.

Fuentes técnicas: Migración 1776366865007 · línea 85.

counter_item_loss

Pérdida de material asociada a un renglón de mostrador.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
reason	character varying(255)	NO		
responsible	character varying(255)	NO		
date	TIMESTAMP	NO		
additionalInfo	text	SÍ		
imageUrls	text	SÍ		
isReturned	boolean	NO		
counter_order_item_id	integer	SÍ	FK	counter_order_item(id)

FK (counter_order_item_id) → counter_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 205.

Fuentes técnicas: Migración 1776366865007 · línea 88.

counter_item_return

Reingreso asociado a un renglón de mostrador.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
observations	character varying(255)	NO		
responsible	character varying(255)	NO		
date	TIMESTAMP	NO		
counter_order_item_id	integer	SÍ	FK	counter_order_item(id)

FK (counter_order_item_id) → counter_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 208.

Fuentes técnicas: Migración 1776366865007 · línea 91.

counter_item_partial_return

Reingreso parcial asociado a un renglón de mostrador.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
responsible	character varying(255)	NO		

Campo	Tipo	NULL	Clave	Referencia física
date	TIMESTAMP	NO		
counter_order_item_id	integer	SÍ	FK	counter_order_item(id)

FK (counter_order_item_id) → counter_order_item(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 211.

Fuentes técnicas: Migración 1776366865007 · línea 94.

counter_order_item

Renglón de material de un pedido de mostrador.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
amount	integer	NO		
status	public.counter_order_item_status_enum	NO		
material_id	integer	SÍ	FK	materials(id)
sector_from_id	integer	SÍ	FK	sector(id)
counterOrderId	integer	SÍ	FK	counter_order(id)

FK (material_id) → materials(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 214.

FK (sector_from_id) → sector(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 217.

FK (counterOrderId) → counter_order(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 220.

Fuentes técnicas: Migración 1776366865007 · línea 100.

counter_order

Pedido de mostrador con área y usuario creador.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
responsible	character varying(100)	NO		
holder	character varying(100)	NO		
destinationSector	character varying(100)	NO		
status	public.counter_order_status_enum	NO		
createdAt	TIMESTAMP	NO		
operational_area_id	integer	NO	FK	operational_area(id)
created_by_user_id	integer	SÍ	FK	users(id)

FK (operational_area_id) → operational_area(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 223.

FK (created_by_user_id) → users(id). Padres por hija: 0..1; hijas por padre: 0..N. Migración 1776366865007 · línea 226.

Fuentes técnicas: Migración 1776366865007 · línea 106.

role

Rol de autorización del dominio.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
code	character varying(50)	NO		
name	character varying(100)	NO		
description	character varying(255)	SÍ		
is_active	boolean	NO		
is_system	boolean	NO		

Fuentes técnicas: Migración 1776366865007 · línea 109; Migración 1782485000000 · línea 17.

user_role

Asignación de roles a usuarios.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: (user_id, role_id).

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
user_id	integer	NO	FK	users(id)
role_id	integer	NO	FK	role(id)

FK (user_id) → users(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 229.

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 232.

Fuentes técnicas: Migración 1776366865007 · línea 112.

user_area_access

Acceso de usuario a un área operativa.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: user_id, operational_area_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
user_id	integer	NO	PK, FK	users(id)
operational_area_id	integer	NO	PK, FK	operational_area(id)
is_primary	boolean	NO		

FK (user_id) → users(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 235.

FK (operational_area_id) → operational_area(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 238.

Fuentes técnicas: Migración 1776366865007 · línea 115.

permission

Permiso granular del dominio.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: (code).

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
code	character varying(100)	NO		
resource	character varying(50)	NO		

Campo	Tipo	NULL	Clave	Referencia física
action	character varying(50)	NO		
scope	character varying(50)	NO		
description	character varying(255)	SÍ		
is_active	boolean	NO		

Fuentes técnicas: Migración 1776366865007 · línea 118.

role_permission

Tabla puente entre roles y permisos.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: role_id, permission_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
role_id	integer	NO	PK, FK	role(id)
permission_id	integer	NO	PK, FK	permission(id)

FK (role_id) → role(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 241.

FK (permission_id) → permission(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 244.

Fuentes técnicas: Migración 1776366865007 · línea 121.

activity_order_destination_sector

Tabla puente de sectores de destino de un pedido de actividad.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: activity_order_id, sector_id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
activity_order_id	integer	NO	PK, FK	activity_order(id)
sector_id	integer	NO	PK, FK	sector(id)

FK (activity_order_id) → activity_order(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 247.

FK (sector_id) → sector(id). Padres por hija: 1; hijas por padre: 0..N. Migración 1776366865007 · línea 250.

Índice: CREATE INDEX IDX_7461f4367ed818a0a22ddd5f4e ON esquema.activity_order_destination_sector (activity_order_id) · Migración 1776366865007 · línea 127

Índice: CREATE INDEX IDX_49228b66925f0abaa1aaa48682 ON esquema.activity_order_destination_sector (sector_id) · Migración 1776366865007 · línea 130

Fuentes técnicas: Migración 1776366865007 · línea 124.

rbac_audit_log

Registro histórico de acciones de administración de accesos.

Estructura verificada en código y migraciones. Esquema de tablas no fijado en la configuración; public es una ubicación inferida por el contexto SQL.

PK: id. Unicidad total: sin declaración identificada.

Campo	Tipo	NULL	Clave	Referencia física
id	SERIAL	NO	PK	
action	character varying(64)	NO		
actor_user_id	integer	SÍ		
actor_email	character varying(255)	NO		

Campo	Tipo	NULL	Clave	Referencia física
target_type	character varying(16)	NO		
target_id	integer	NO		
target_label	character varying(255)	NO		
details	jsonb	SÍ		
created_at	TIMESTAMPTZ	NO		

Índice: CREATE INDEX IDX_rbac_audit_log_created_at ON esquema.rbac_audit_log (created_at DESC) · Migración 1782485000000 · línea 44

Índice: CREATE INDEX IDX_rbac_audit_log_target ON esquema.rbac_audit_log (target_type, target_id) · Migración 1782485000000 · línea 49

Índice: CREATE INDEX IDX_rbac_audit_log_action ON esquema.rbac_audit_log (action) · Migración 1782485000000 · línea 54

Fuentes técnicas: Migración 1782485000000 · línea 29.