

Rol: Diseñador físico

Misión dentro del equipo de trabajo

Implementar el modelo lógico en el sistema gestor de bases de datos elegido mediante DDL ejecutable y controlado, asegurando integridad, rendimiento, seguridad y operabilidad. Traduce definiciones lógicas a tipos nativos, claves, índices y restricciones reales, con despliegues versionados y reversibles.

Actividades que ejecuta

- Construcción de tablas, elección de tipos nativos, nulabilidad, valores por defecto, columnas calculadas y columnas autoincrementales cuando aplique.
- Creación de llaves principales y llaves foráneas, restricciones de unicidad, restricciones semánticas (intervalos, catálogos, formatos).
- Definición de índices simples o compuestos.
- Planificación de cargas iniciales y dependencias

Productos sobre los cuales es responsable

- Scripts para la creación de las tablas versionado: CREATE/ALTER de tablas, PK/FK, UNIQUE, CHECK, índices, secuencias/identidades.
- Esquema físico documentado: catálogo de tablas/columnas (tipo, colación, nulabilidad, default), catálogo de índices (propósito, columnas, selectividad), mapa de particiones y parámetros de almacenamiento.
- Evidencias de pruebas que demuestra que el esquema funciona como se espera.

Buenas prácticas para considerar

- Elección precisa del tipo y la longitud mínimos necesarios.
Crea índices solo para patrones reales de acceso; elimina duplicados y evita la sobre-indexación. Ordena columnas por selectividad/uso y evalúa el impacto en escrituras (INSERT/UPDATE/DELETE).
- Restricciones antes que lógica de aplicación. Garantiza la calidad en el motor con NOT NULL, CHECK, UNIQUE y FK efectivas (habilitadas y validadas), antes de delegar validaciones a la aplicación.
- Aplica una convención de nombres consistente para objetos y restricciones, e incluye comentarios/propiedades extendidas en el DDL cuando el motor lo permita.

Habilidades requeridas

- Convertir el diseño en algo real. Crear las tablas y ajustes concretos que usará la base de datos, eligiendo bien los tipos de datos, los nombres y, cuando convenga, cómo separar los datos (por ejemplo, por periodos) para que sea más fácil de manejar.
- Aplicar las reglas del negocio dentro de la base. Dejar claro qué campos son obligatorios, cuáles traen un valor por defecto, cómo se relacionan las tablas, qué debe pasar si se borra o cambia un dato relacionado, y usar formatos correctos (fechas, textos, números) y políticas de retención.
- Explicar el diseño de forma clara. Presentar qué tablas existen y para qué sirven, cómo se facilita la búsqueda, cómo están organizados los datos.
- Cuidar los detalles y dejar rastro. Elegir tamaños adecuados (sin exagerar), usar nombres consistentes, dejar notas útiles en los scripts y documentar cualquier excepción para que todos entiendan el “por qué” de las decisiones. Mantener el vínculo con el diseño lógico.
- Probar y desplegar con el equipo. Ensayar con datos parecidos a los reales.

Errores comunes

- Dejar toda la validación a la aplicación. No marcar campos obligatorios, valores por defecto o rangos permitidos en la base de datos termina permitiendo datos malos o incompletos.
- Poner índices “por si acaso”. Crear demasiados índices o sin un uso real (o con las columnas en mal orden) casi no ayuda a leer más rápido y sí hace más lentas las escrituras.
- Relaciones sin control (efecto dominó). Borrar o cambiar un dato y que arrastre otros sin querer, o desactivar las relaciones entre tablas y permitir combinaciones incoherentes.
- Scripts frágiles. Cambios que dependen del “estado” del ambiente, fallan si algo ya existe, no se pueden repetir con seguridad y no tienen forma clara de deshacer.
- Probar con datos de juguete. No usar volúmenes reales ni verificar si las consultas aprovechan los índices hace que, en producción, todo sea más lento de lo esperado.

Criterios de desempeño

El rol se considera cumplido de manera adecuada cuando:

- Identificadores y reglas listas. Cada tabla tiene su identificador único y están activas las reglas que evitan duplicados, vacíos, valores fuera de rango y faltantes; todo fue comprobado.

- Las consultas más importantes entregan resultados dentro de los tiempos comprometidos y con el rendimiento esperado.
- Datos coherentes. Las validaciones no muestran errores.
- Cambios bajo control. Los scripts están en control de versiones, se pueden aplicar sin sorpresas y también deshacer; además, se registra claramente qué cambió y cuándo.