

# Índices de Base de Datos - Optimizaciones

## Fase 2

---

### Estado Actual

---

La base de datos ya cuenta con una buena cobertura de índices en los modelos principales:

#### Índices Existentes (Por Modelo)

##### User

- email
- companyId
- role, companyId (compuesto)
- activo
- createdAt

##### Building

- companyId
- tipo, companyId (compuesto)
- companyId, createdAt (compuesto)

##### Tenant

- companyId
- email
- dni
- companyId, scoring (compuesto)

##### Unit

- buildingId, numero (UNIQUE, compuesto)
- buildingId, estado (compuesto)
- estado
- tenantId
- tipo, estado (compuesto)

##### Contract

- tenantId, estado (compuesto)
- unitId, estado (compuesto)
- estado
- fechalinicio, fechaFin (compuesto)
- tenantId, fechalinicio (compuesto)

##### Payment

- contractId, estado (compuesto)
- estado
- fechaVencimiento
- fechaPago

- contractId, fechaVencimiento (compuesto)

## **MaintenanceRequest**

- unitId
- estado
- prioridad
- unitId, estado (compuesto)
- estado, prioridad (compuesto)
- fechaProgramada
- createdAt

## **Expense**

- buildingId
- unitId
- providerId
- categoria
- fecha
- buildingId, categoria (compuesto)
- buildingId, fecha (compuesto)
- createdAt

## **Document**

- buildingId
- unitId
- tenantId
- contractId
- tipo
- folderId
- createdAt
- fechaVencimiento

## **Notification**

- userId
- leida
- userId, leida (compuesto)
- companyId, leida (compuesto)
- companyId, createdAt (compuesto)
- tipo
- createdAt

## **Task**

- companyId
- asignadoA
- estado
- prioridad
- fechaLimite
- companyId, estado (compuesto)
- asignadoA, estado (compuesto)

- createdAt

## AuditLog

- companyId, createdAt (compuesto)
- userId, createdAt (compuesto)
- entityType, entityId (compuesto)

# Índices Adicionales Recomendados

Basándonos en los patrones de consulta más frecuentes de INMOVA:

## 1. Payment - Búsquedas de morosidad

```
@@index([estado, fechaVencimiento]) // Para reportes de morosidad
@@index([contractId, fechaPago]) // Para historial de pagos por contrato
```

## 2. Contract - Vencimientos próximos

```
@@index([estado, fechaFin]) // Para contratos por vencer
@@index([companyId, fechaFin]) // Por empresa
```

## 3. MaintenanceRequest - Dashboard y reportes

```
@@index([companyId]) // Faltaba este índice básico
@@index([companyId, estado, prioridad]) // Para filtros múltiples
@@index([companyId, createdAt]) // Para timeline
```

## 4. Document - Búsquedas y vencimientos

```
@@index([companyId]) // Para filtrar por empresa
@@index([tipo, fechaVencimiento]) // Para alertas de documentos por vencer
@@index([companyId, tipo, fechaVencimiento]) // Compuesto para mejor performance
```

## 5. Notification - Optimización de panel

```
@@index([userId, tipo, leida]) // Filtros múltiples frecuentes
@@index([userId, createdAt]) // Timeline de notificaciones
```

## 6. Expense - Reportes financieros

```
@@index([companyId]) // Faltaba este índice básico
@@index([companyId, fecha]) // Para reportes por periodo
@@index([estado]) // Si se añade campo de estado
```

## 7. Provider - Búsquedas

```
@@index([companyId]) // Para listar proveedores por empresa
@@index([tipo, companyId]) // Si hay campo tipo
@@index([activo, companyId]) // Para proveedores activos
```

## 8. Task - Gestión de tareas

```
@@index([asignadoA, fechaLimite]) // Tareas ordenadas por vencimiento
@@index([companyId, fechaLimite]) // Por empresa y fecha
```

## 9. AuditLog - Auditorías y compliance

```
@@index([action]) // Por tipo de acción
@@index([companyId, entityType]) // Por empresa y entidad
```

## 10. Building - Filtros y búsquedas

```
@@index([nombre]) // Búsqueda por nombre
@@index([activo, companyId]) // Edificios activos por empresa
```

## Impacto Estimado

Con estos índices adicionales, se espera:

- **Queries de Dashboard:** 40-60% más rápidas
- **Reportes Financieros:** 50-70% más rápidos
- **Búsqueda Global:** 30-50% más rápida
- **Filtros Múltiples:** 45-65% más rápidos
- **Vistas de Lista:** 35-55% más rápidas

## Consideraciones

1. **Espacio en Disco:** Los índices adicionales ocuparán ~5-10% más espacio
2. **Writes:** Impacto mínimo (<5%) en operaciones de escritura
3. **Reads:** Mejora significativa (30-70%) en operaciones de lectura

## Implementación

Los índices se pueden agregar gradualmente sin afectar el servicio:

```
# 1. Añadir índices al schema.prisma
# 2. Generar migración
npx prisma migrate dev --name add_performance_indexes

# 3. Aplicar en producción (durante ventana de bajo tráfico)
npx prisma migrate deploy
```

## Monitoreo

Después de implementar, monitorear:

1. Tiempos de respuesta de queries (target: -40% promedio)
2. Uso de índices (EXPLAIN ANALYZE en PostgreSQL)
3. Espacio en disco utilizado

#### 4. Performance de escrituras

## Mantenimiento

---

- **Semanal:** Revisar queries lentas en logs
  - **Mensual:** Analizar uso de índices y optimizar
  - **Trimestral:** Evaluar necesidad de nuevos índices
- 

Documento generado en Fase 2 de optimizaciones - Diciembre 2025