

# Arreglar Error de TypeScript en Approvals API

## Problema

```
Type error: Type '{ userId: string; tipo: "alerta_sistema"; titulo: string; mensaje: string; leida: false; }' is not assignable to type 'NotificationCreateInput'
```

## Causa

Falta el campo `companyId` que es requerido en el schema de Prisma.

## Archivos Afectados

1. `app/api/approvals/route.ts`
2. `app/api/approvals/[id]/route.ts`

## Solución 1: app/api/approvals/route.ts

### Buscar el código similar a:

```
await prisma.notification.create({
  data: {
    userId: approval.createdById,
    tipo: 'alerta_sistema',
    titulo: 'Solicitud de aprobación',
    mensaje: `...`,
    leida: false,
  }
});
```

### Cambiar a:

```
await prisma.notification.create({
  data: {
    companyId: approval.companyId, // ✓ AGREGAR ESTA LÍNEA
    userId: approval.createdById,
    tipo: 'alerta_sistema',
    titulo: 'Solicitud de aprobación',
    mensaje: `...`,
    leida: false,
    prioridad: 'medio', // ✓ Opcional pero recomendado
  }
});
```

## Solución 2: app/api/approvals/[id]/route.ts

Buscar código similar y aplicar el mismo cambio:

```
// Todas las notificaciones deben incluir companyId
await prisma.notification.create({
  data: {
    companyId: approval.companyId, // ✓ AGREGAR
    userId: targetUserId,
    tipo: 'alerta_sistema',
    titulo: '....',
    mensaje: '....',
    leida: false,
    prioridad: 'medio',
  }
});
```

## Verificación

### 1. Verificar que companyId exista en el objeto approval

```
// Antes de crear la notificación, verificar:
console.log('Approval:', approval);
console.log('CompanyId:', approval.companyId);
```

### 2. Si approval no tiene companyId directamente

```
// Opción A: Obtener de la relación
const approval = await prisma.approval.findUnique({
  where: { id },
  include: {
    property: { select: { companyId: true } } // Si approval está relacionado con
    property
  }
});

// Luego usar:
companyId: approval.property.companyId,

// Opción B: Obtener del usuario actual
const session = await getServerSession(authOptions);
companyId: session.user.companyId,

// Opción C: Obtener de donde sea que esté disponible
```

## Comandos de Verificación

```
# 1. Buscar todos los lugares donde se crean notificaciones sin companyId
cd /home/ubuntu/homming_vidaro/nextjs_space
grep -r "tipo: 'alerta_sistema'" app/api/ -A 5 -B 5

# 2. Verificar el schema de Prisma
grep -A 20 "model Notification" prisma/schema.prisma

# 3. Después de arreglar, verificar tipos
yarn tsc --noEmit

# 4. Build completo
yarn build
```

## Patrón Correcto para Todas las Notificaciones

```
// ✅ PATRÓN CORRECTO
await prisma.notification.create({
  data: {
    // CAMPOS REQUERIDOS
    companyId: string, // ⚡ SIEMPRE REQUERIDO
    tipo: NotificationType, // enum
    titulo: string,
    mensaje: string,

    // CAMPOS OPCIONALES
    userId?: string, // Para notificaciones de usuario específico
    leida: boolean, // Default: false
    prioridad: RiskLevel, // Default: bajo (bajo, medio, alto, critico)
    fechaLimite?: Date, // Para notificaciones con deadline
    entityId?: string, // ID de la entidad relacionada
    entityType?: string, // Tipo de entidad (property, maintenance, etc.)
  }
});
```

## Testing

Después de aplicar los cambios:

```
# 1. Limpiar build anterior
rm -rf .next

# 2. Generar Prisma client
yarn prisma generate

# 3. Build
yarn build

# 4. Si build es exitoso, probar en dev
yarn dev

# 5. Probar la funcionalidad de approvals
# - Crear una aprobación
# - Verificar que se crea la notificación correctamente
# - Verificar que no hay errores en console
```

## Comandos Rápidos

```
# Ver todas las notificaciones en la DB para debugging
cd /home/ubuntu/homming_vidaro/nextjs_space
npx prisma studio
# Navegar a: Notification model

# O con query directa:
echo "SELECT id, companyId, userId, tipo, titulo FROM Notification LIMIT 10;" | \
psql $DATABASE_URL
```

## Notas Importantes

1. **Cada notificación DEBE tener un companyId** - Es parte del modelo de multi-tenancy
2. **No es opcional** - El schema de Prisma lo marca como requerido
3. **Buscar otros lugares** - Puede haber más creaciones de notificaciones con el mismo problema
4. **Migration** - Si cambias el schema, necesitas hacer migration

## Si el Problema Persiste

### Verificar el Schema de Prisma

```
model Notification {
    id      String      @id @default(cuid())
    companyId String     // ← Este campo es REQUERIDO
    tipo    NotificationType
    titulo  String
    mensaje String      @db.Text
    leida   Boolean     @default(false)
    prioridad RiskLevel @default(bajo)
    fechaLimite DateTime?
    entityId String?
    entityType String?
    userId   String?
    createdAt DateTime   @default(now())
    updatedAt DateTime   @updatedAt

    company Company @relation(fields: [companyId], references: [id], onDelete: Cascade)
    user    User?    @relation(fields: [userId], references: [id], onDelete: Cascade)

    @@index([companyId])
    @@index([userId])
    @@index([leida])
    @@index([tipo])
}
```

Si `companyId` no tiene `?` (opcional), entonces **ES REQUERIDO** en todas las creaciones.