

Errores de TypeScript en el Proyecto



Estado Actual

Se han identificado **15 errores de TypeScript** en el archivo `lib/str-housekeeping-service.ts` del módulo STR (Short-Term Rental).

⚠ Importante: Estos errores NO fueron causados por las optimizaciones implementadas (code splitting, memoria, skipLibCheck). Son errores pre-existentes en el código.

Errores Identificados

1. Propiedad 'cantidadActual' no existe (Línea 523)

```
Type '{ cantidadActual: string; }' is not assignable to type 'STRHousekeepingInventoryOrderByWithRelationInput'
```

Causa: La propiedad `cantidadActual` no está definida en el tipo de Prisma para ordering.

Solución:

```
// Antes
orderBy: { cantidadActual: 'desc' }

// Después (verificar schema de Prisma)
orderBy: { cantidad: 'desc' } // o el nombre correcto del campo
```

2. Propiedad 'tipoTurnover' no existe (Línea 540)

```
Object literal may only specify known properties, and 'tipoTurnover' does not exist
```

Causa: El campo se llama diferente en el schema de Prisma.

Solución:

```
// Verificar en schema.prisma el nombre correcto
// Probablemente es 'turnoverType' o 'tipo'
```

3. Campo 'deep_clean' no válido (Línea 559)

```
Object literal may only specify known properties, and 'deep_clean' does not exist
```

Causa: El enum `TurnoverType` usa mayúsculas o nombres diferentes.

Solución:

```
// Antes
const stats = {
  check_out: 0,
  check_in: 0,
  deep_clean: 0, // ✗ Incorrecto
  mantenimiento: 0,
  inspeccion: 0,
}

// Después (verificar enum en schema.prisma)
const stats: Record<TurnoverType, number> = {
  CHECK_OUT: 0,
  CHECK_IN: 0,
  DEEP_CLEAN: 0, // ✓ Correcto si es mayúsculas
  MANTENIMIENTO: 0,
  INSPECCION: 0,
}
```

4. Enum incorrecto para BookingStatus (Línea 580)

```
Type '"confirmada"' is not assignable to type 'BookingStatus'. Did you mean '"CONFIR-
ADA"'?
Type '"pendiente"' is not assignable to type 'BookingStatus'. Did you mean '"PENDI-
ENTE"'?
```

Causa: Los enums de Prisma usan mayúsculas.

Solución:

```
// Antes
where: {
  estado: { in: ["confirmada", "pendiente"] } // ✗ Incorrecto
}

// Después
import { BookingStatus } from '@prisma/client';

where: {
  estado: { in: [BookingStatus.CONFIRMADA, BookingStatus.PENDIENTE] } // ✓ Correcto
}
```

5. Propiedades 'bookingCheckOutId' y 'bookingCheckInId' no existen (Líneas 598-599)

```
Object literal may only specify known properties, and 'bookingCheckOutId' does not ex-
ist
```

Causa: Los nombres de las relaciones son diferentes en el schema.

Solución:

```
// Verificar en schema.prisma cómo se llaman estas relaciones
// Probablemente son 'checkOutBookingId' o 'bookingId' con un filtro adicional
```

6. Propiedad 'fechaCheckIn' no existe (Líneas 611-613)

```
Property 'fechaCheckIn' does not exist on type Booking
```

Causa: El campo tiene un nombre diferente en el modelo.

Solución:

```
// Antes
booking.fechaCheckIn // ✗ Incorrecto

// Después (verificar schema.prisma)
booking.checkInDate // ✓ Correcto si es el nombre en el schema
// o
booking.fecha_check_in // si usa snake_case
```

7. Propiedad 'fechaCheckOut' no existe (Líneas 625-627)

```
Property 'fechaCheckOut' does not exist on type Booking
```

Causa: Similar al anterior.

Solución:

```
// Antes
booking.fechaCheckOut // ✗ Incorrecto

// Después
booking.checkOutDate // ✓ Correcto
```



Cómo Corregir

Paso 1: Revisar Schema de Prisma

```
cd /home/ubuntu/homming_vidaro/nextjs_space
cat prisma/schema.prisma | grep -A 20 "model STRHousekeepingInventory"
cat prisma/schema.prisma | grep -A 20 "model STRBooking"
cat prisma/schema.prisma | grep -A 10 "enum TurnoverType"
cat prisma/schema.prisma | grep -A 10 "enum BookingStatus"
```

Paso 2: Comparar Nombres

Crear un mapeo de nombres usados vs nombres correctos:

Usado en Código	Nombre Correcto en Schema
fechaCheckIn	? (verificar)
fechaCheckOut	? (verificar)
cantidadActual	? (verificar)
tipoTurnover	? (verificar)
bookingCheckOutId	? (verificar)

Paso 3: Actualizar Código

Una vez identificados los nombres correctos, actualizar `lib/str-housekeeping-service.ts` con los nombres correctos.

Paso 4: Regenerar Cliente de Prisma

Si hubo cambios en el schema:

```
cd /home/ubuntu/homming_vidaro/nextjs_space
yarn prisma generate
```

Paso 5: Verificar

```
NODE_OPTIONS="--max-old-space-size=6144" yarn tsc --noEmit
```



Por Qué Ocurrieron Estos Errores

Posibles causas:

- Migración de Schema:** Se cambiaron nombres de campos en Prisma pero no se actualizó el código
- Feature Branches:** Código de diferentes branches con schemas diferentes
- Internacionalización:** Cambio de nombres en español a inglés o viceversa
- Refactoring Incompleto:** Se empezó un refactor pero no se completó



Impacto

En Desarrollo

- TypeScript no compila con `tsc --noEmit`
- Pero Next.js puede funcionar si `typescript.ignoreBuildErrors: true`

En Producción

Estado Actual del Config:

```
// next.config.js
typescript: {
  ignoreBuildErrors: false, // Actualmente NO ignora errores
}
```

Esto significa que **el build de producción fallará** si no se corrijen estos errores.

Funcionalidad Afectada

Solo el **módulo STR Housekeeping** está afectado:

- Gestión de inventario de limpieza
- Asignación de tareas de housekeeping
- Checklists de limpieza
- Estadísticas de turnover

El resto de la aplicación (finanzas, analytics, BI, etc.) no tiene errores.



Plan de Acción

Prioridad Alta

1. [] Revisar `prisma/schema.prisma` para nombres correctos
2. [] Crear mapeo de nombres incorrectos → correctos
3. [] Actualizar `lib/str-housekeeping-service.ts`
4. [] Verificar con `tsc --noEmit`
5. [] Probar funcionalidad STR en desarrollo

Prioridad Media

1. [] Agregar tests para prevenir regresiones
2. [] Documentar convenciones de nombres
3. [] Crear linter rules para validar nombres de Prisma

Prioridad Baja

1. [] Considerar usar Prisma's generated types más estrictamente
2. [] Revisar otros servicios por errores similares



Responsables

Estos errores deben ser corregidos por:

- **Equipo STR Module** - Conocen la lógica del negocio
- **Database Team** - Si hay cambios necesarios en schema
- **QA Team** - Para verificar que las correcciones no rompan funcionalidad



Siguiente Paso

Para continuar, necesitas:

1. **Decisión:** ¿Corregir errores ahora o documentar para después?
2. **Si corregir ahora:**

`bash`

```
# Dame acceso al schema de Prisma  
cat /home/ubuntu/homming_vidaro/nextjs_space/prisma/schema.prisma
```

3. Si documentar para después:

- Ya está documentado en este archivo
- Asignar tarea al equipo STR
- Continuar con deployment ignorando este módulo temporalmente