## 🗄 Mini Guía Técnica

**Rol: Backend – Base de Datos**

---

### 🎯 Objetivo del rol

Diseñar y mantener la **estructura de base de datos** que permita:

- almacenar órdenes creadas desde el frontend,
- preservar fielmente el JSON recibido,
- permitir recuperación eficiente,
- soportar futuras extensiones **sin migraciones destructivas**.

La base de datos **no interpreta la lógica del formulario**.

---

### 📌 Contexto del proyecto

El sistema recibe **órdenes dinámicas**, no formularios fijos.

Por tanto:

- no existe una tabla por campo
- no existen columnas fijas para cada tipo de campo
- la estructura debe ser flexible

📌 Se utiliza **PostgreSQL**.

---

### □ Responsabilidades técnicas

#### 1□ Definir modelo de datos principal

Entidad central: **Order**

Tabla mínima recomendada:

orders

---

id UUID PK

title VARCHAR

schema JSONB

created_at TIMESTAMP

updated_at TIMESTAMP

---

#### 2□ Uso de JSONB (obligatorio)

El campo schema debe almacenar:

```
{
  "fields": [
    {
      "id": "uuid",
      "type": "select",
      "order": 1,
      "props": { }
    }
  ]
}
```

📌 No descomponer el JSON en tablas relacionales.

---

### 3️⃣ Qué NO modelar como tablas

❌ Campos
❌ Opciones del select
❌ Props individuales
❌ Orden de campos

Todo eso **vive dentro del JSON**.

---

### 4️⃣ Índices recomendados

```
CREATE INDEX idx_orders_created_at ON orders(created_at);
```

Opcional:

```
CREATE INDEX idx_orders_schema ON orders USING GIN (schema);
```

---

### 5️⃣ Integridad de datos

La base de datos:

- no valida estructura de schema
- no impone constraints sobre el JSON
- confía en la API para validar

📌 Esto evita acoplamiento entre capas.

---

### 6️⃣ Escalabilidad futura

Este diseño permite:

- versionar schemas
- agregar metadatos
- soportar nuevos tipos de campo
- generar plantillas

Sin cambiar estructura base.

---

### 🗂️ Archivos que puede modificar

✔ Scripts SQL
✔ Migraciones
✔ Modelos ORM
✔ Configuración de conexión

---

### 🚫 Archivos que NO debe modificar

✖ Frontend
✖ API Controllers
✖ Validaciones de negocio
✖ Contrato JSON
✖ UI

---

### ☑ Checklist de validación

- Existe tabla orders
- schema es JSONB
- No hay tablas innecesarias
- No hay lógica de negocio en DB
- Se pueden guardar órdenes dinámicas
- No se rompe si cambian los campos

---

### 🏗️ Frase guía para IA (copiar y pegar)

*Estoy diseñando la base de datos de un sistema que almacena órdenes dinámicas creadas desde un frontend. Uso PostgreSQL y JSONB. No debo normalizar campos ni interpretar la estructura del formulario.*

---

**□ Regla de oro**

**Si mañana aparece un nuevo tipo de campo y no hay que tocar la base de datos, el diseño es correcto.**