Reto Técnico - RBAC (Role Based Access Control)

Fecha de entrega: Sábado 3:00 PM Objetivo: Implementar un sistema de control de acceso basado en roles (RBAC) utilizando PostgreSQL como base de datos principal. Requerimientos: - Base de datos: PostgreSQL (crear gratis en Neon). - Lenguajes/Frameworks: - Anthony: Node.js con Express, aplicando Clean Architecture y ORM Drizzle. - Iván: NestJS con TypeORM. - Entregable: - Repositorio en GitHub con la implementación. - Colección de Postman documentada para probar los endpoints de la API.

Modelo de Datos (mínimo requerido)

```sql
-- Empresas
CREATE TABLE company (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name TEXT NOT NULL
);

-- Usuarios
CREATE TABLE "user" (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    company_id UUID NOT NULL REFERENCES company(id) ON DELETE CASCADE,
    email TEXT UNIQUE NOT NULL,
    name TEXT NOT NULL,
    password_hash TEXT NOT NULL
);

-- Roles
CREATE TABLE role (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    company_id UUID NOT NULL REFERENCES company(id) ON DELETE CASCADE,
    name TEXT NOT NULL,
    description TEXT
);

-- Relación Usuario ↔ Rol
CREATE TABLE user_role (
    user_id UUID NOT NULL REFERENCES "user"(id) ON DELETE CASCADE,
    role_id UUID NOT NULL REFERENCES role(id) ON DELETE CASCADE,
    PRIMARY KEY (user_id, role_id)
);

-- Módulos (agrupadores de permisos)
CREATE TABLE module (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    company_id UUID NOT NULL REFERENCES company(id) ON DELETE CASCADE,
    name TEXT NOT NULL, -- ej: "Compras"
    module_id UUID REFERENCES module (id) ON DELETE CASCADE,
    key TEXT NOT NULL, -- ej: "purchases"
    status TEXT CHECK (status IN ('active', 'inactive'))
);

-- Permisos (acciones dentro de un módulo)
CREATE TABLE permission (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    module_id UUID NOT NULL REFERENCES module(id) ON DELETE CASCADE,
    name TEXT NOT NULL, -- ej: "approve_order"
    description TEXT
);

-- Relación Rol ↔ Permiso
CREATE TABLE role_permission (
    role_id UUID NOT NULL REFERENCES role(id) ON DELETE CASCADE,
    permission_id UUID NOT NULL REFERENCES permission(id) ON DELETE CASCADE,
    PRIMARY KEY (role_id, permission_id)
);
```