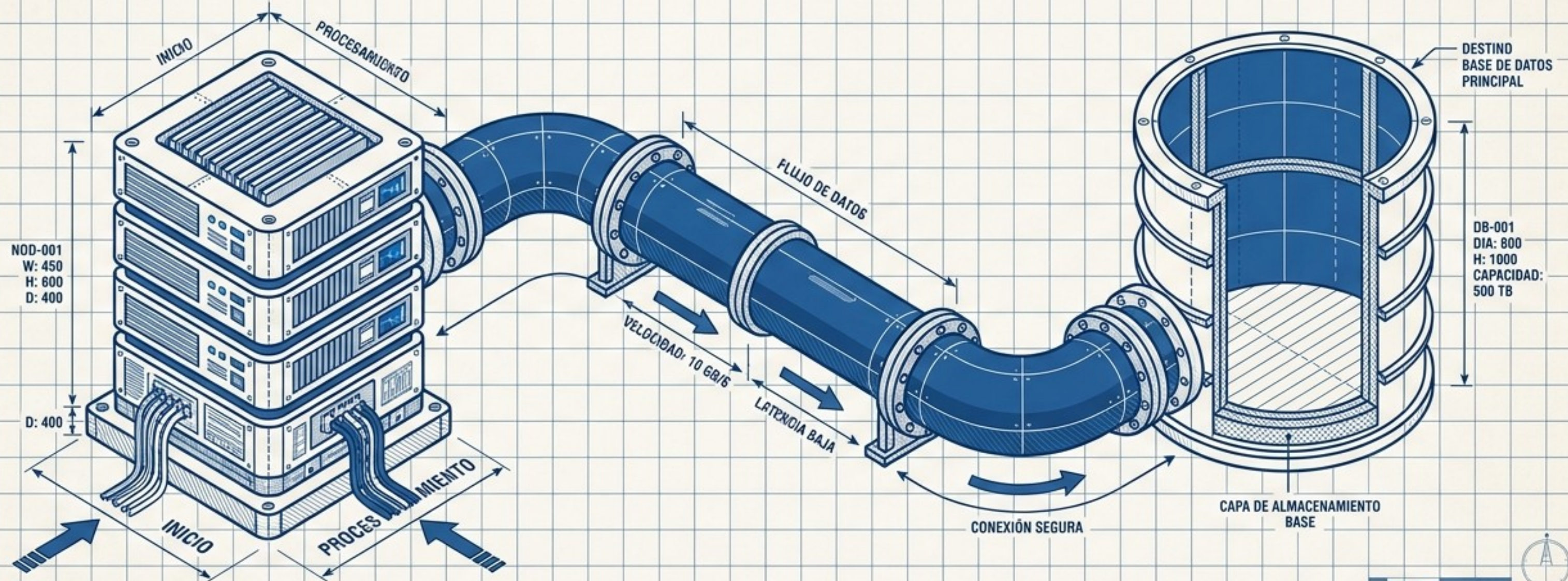


ISS-02: Infraestructura de Base de Datos

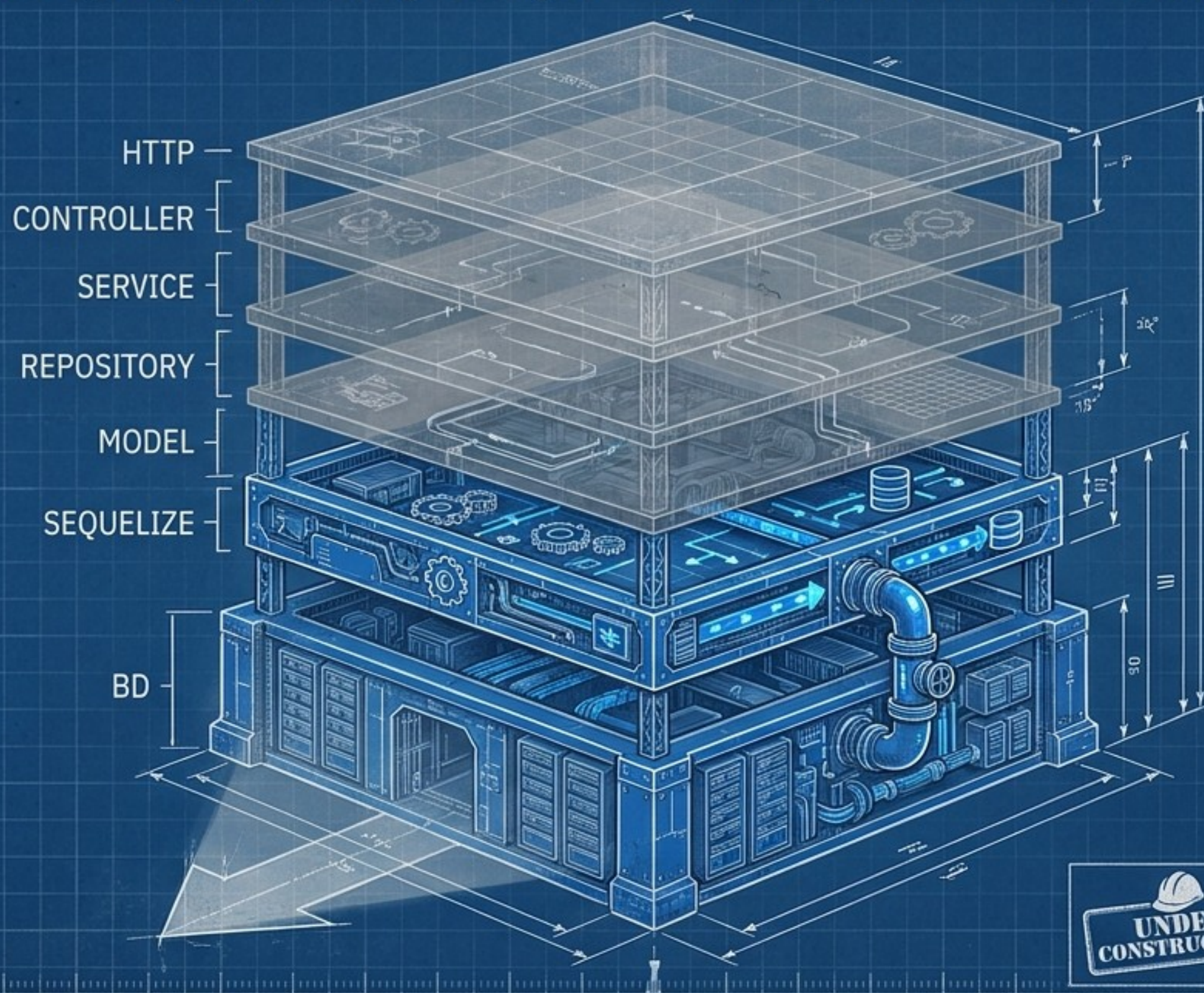
Construyendo la tubería principal del backend



PROYECTO: ISS-02
FECHA: 2024-05-23
ESCALA: 1:100
DIBUJADO POR: EQUIPO DE INGENIERÍA

EL OBJETIVO: CONSTRUIR EL CANAL, NO EL NEGOCIO

En esta etapa no hay modelos, no hay endpoints, ni reglas de negocio. Solo se construye la infraestructura de conexión estable, escalable y segura.

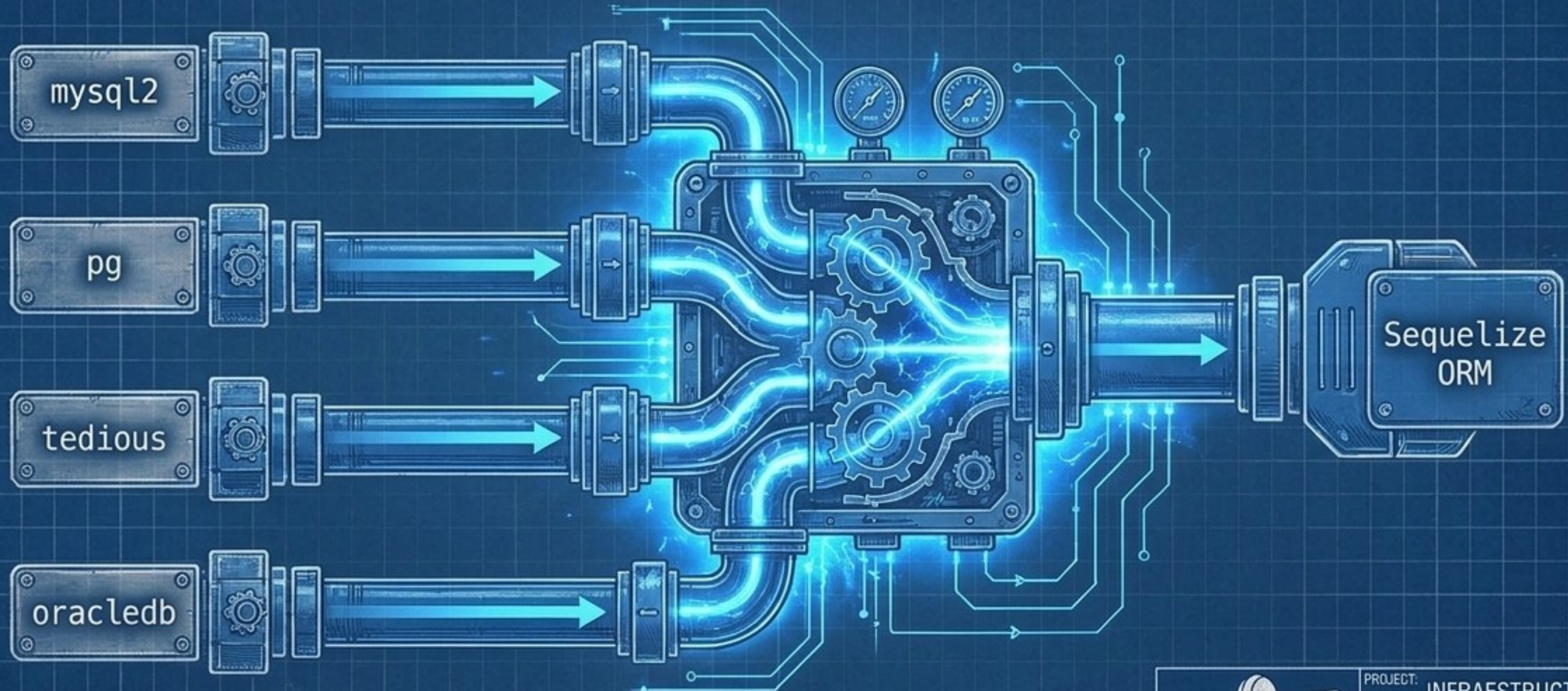


PROJECT: INFRAESTRUCTURA CANAL
DRAWN BY: SYSTEM ARCHITECT
DATE: OCT 26, 2024
REV: A.1

LA MATRIZ DE ADAPTABILIDAD: PORTABILIDAD MULTI-MOTOR

¿Por qué instalar 4 drivers si solo usamos uno?

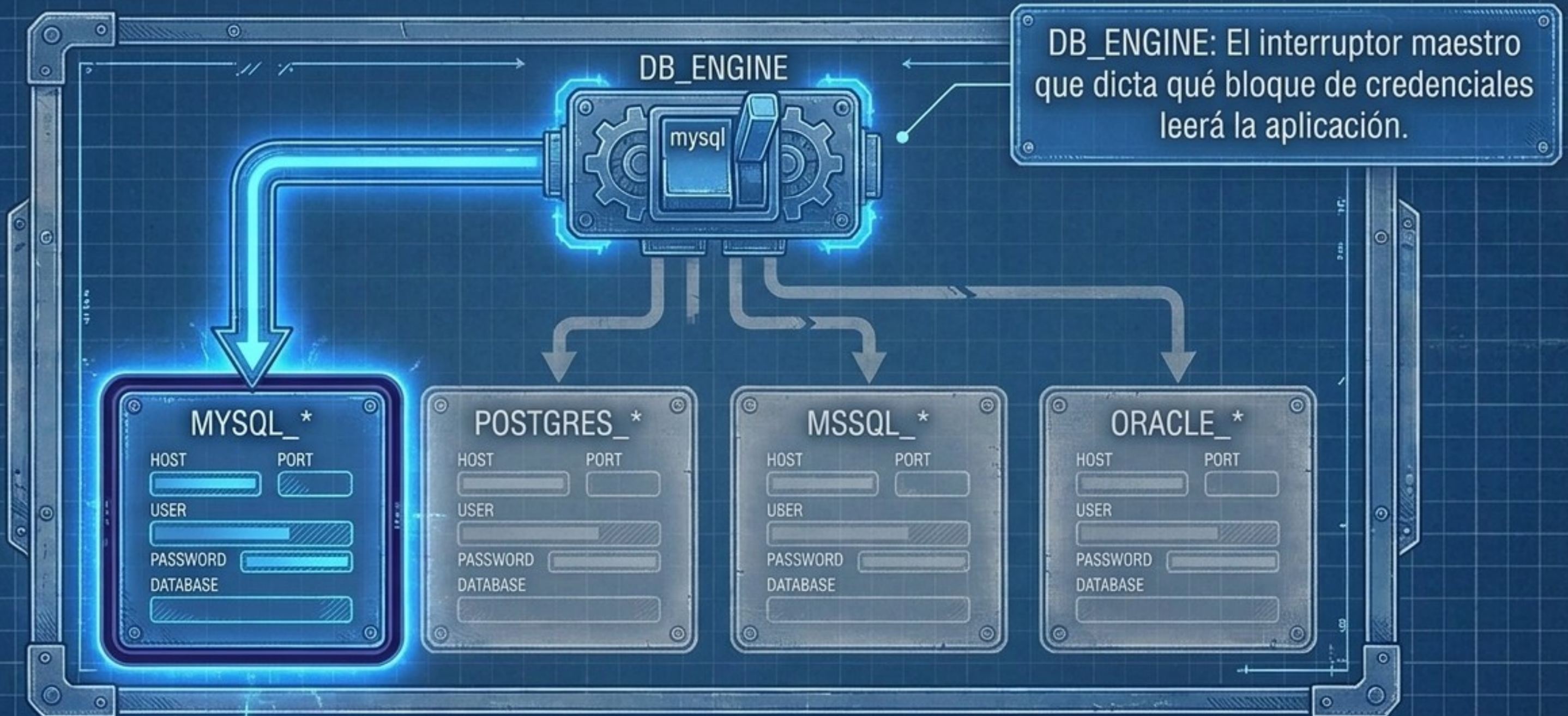
Write once, run anywhere: Permite cambiar de motor de base de datos editando una sola variable de entorno, sin tocar una sola línea de código ni reinstalar dependencias.



PROJECT: INFRAESTRUCTURA CANAL
DRAWN BY: SYSTEM ARCHITECT
DATE: OCT 26, 2024
REV: A.2

EL PANEL DE CONTROL: EL ARCHIVO .ENV

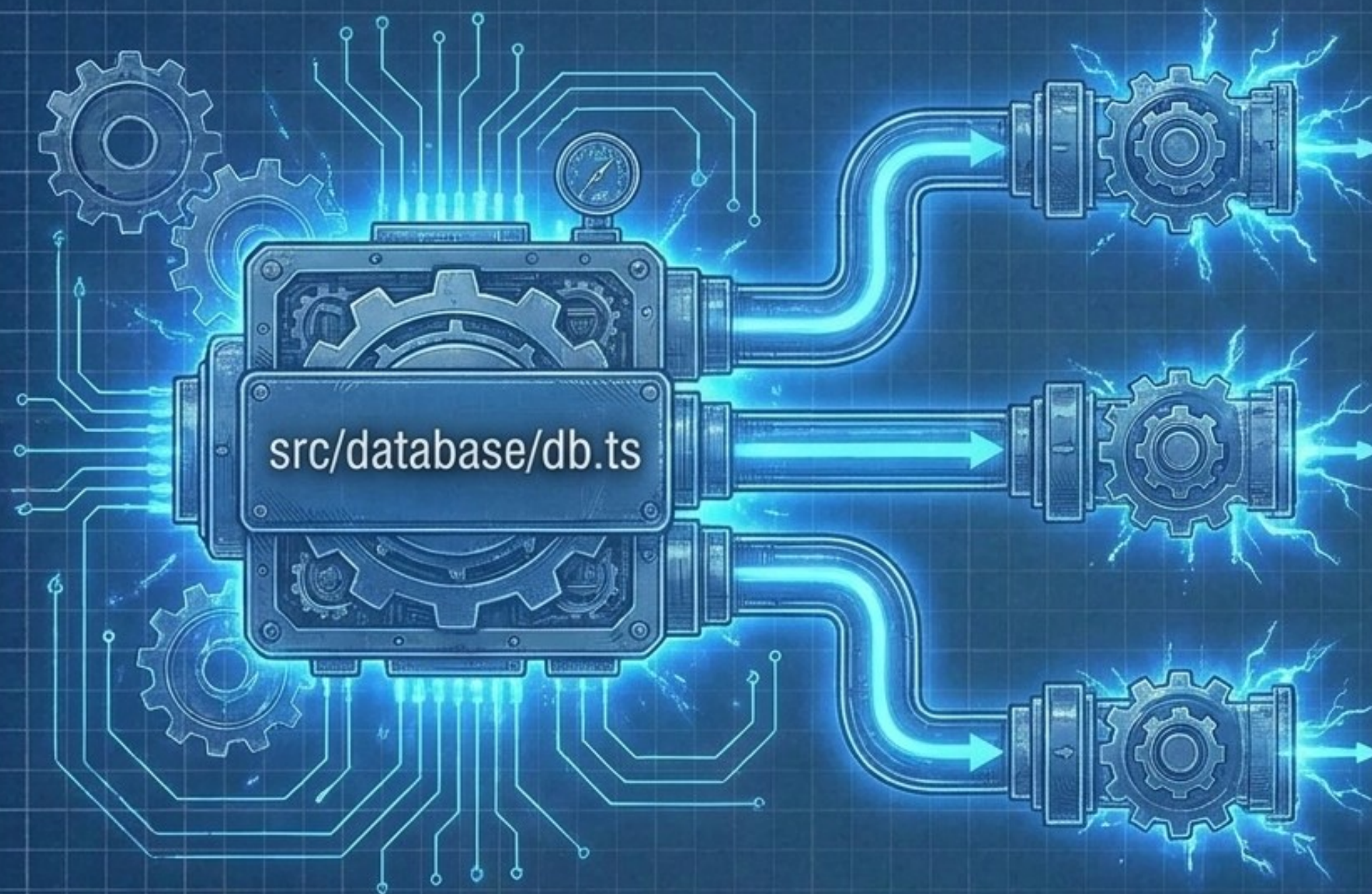
Centraliza la configuración fuera del código fuente. Contiene credenciales específicas y define el flujo de la aplicación.



PROJECT: INFRAESTRUCTURA CANAL
DRAWN BY: SYSTEM ARCHITECT
DATE: OCT 26, 2024
REV: A.3

EL ENRUTADOR CENTRAL: ANATOMÍA DE db.ts

El único punto del proyecto que construye la conexión y expone herramientas para el sistema.



1. sequelize

La instancia activa con su Pool de conexiones (max: 5, min: 0)

2. getDatabaseInfo

Devuelve metadatos de telemetría (motor, config, connectionString)

3. testConnection

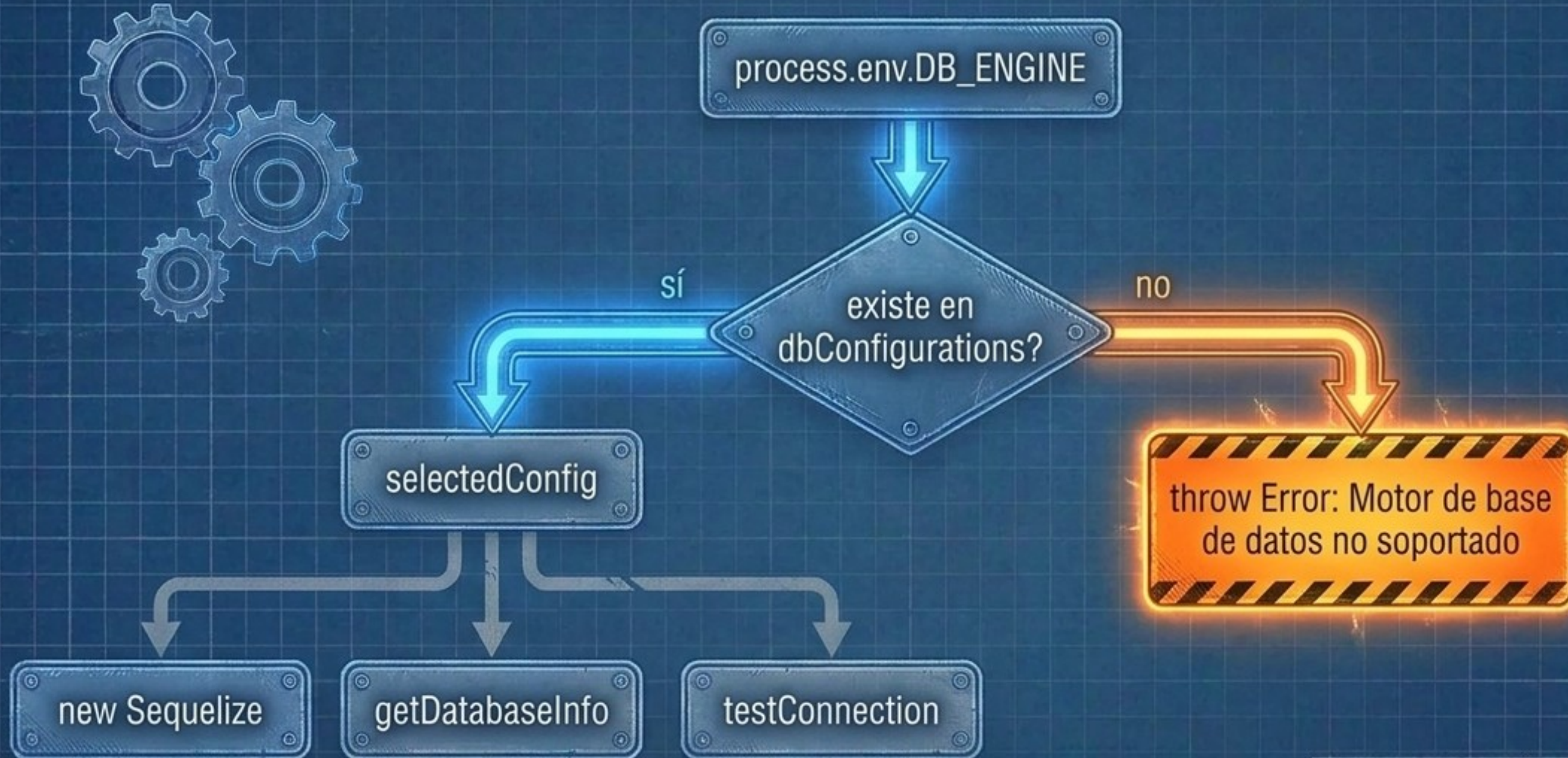
Verifica la conectividad mediante un diagnóstico controlado



PROJECT:	INFRAESTRUCTURA CANAL	
DRAWN BY:	SYSTEM ARCHITECT	REV:
DATE:	OCT 26, 2024	A.4

LÓGICA DE INICIALIZACIÓN: EL MECANISMO FAIL-FAST

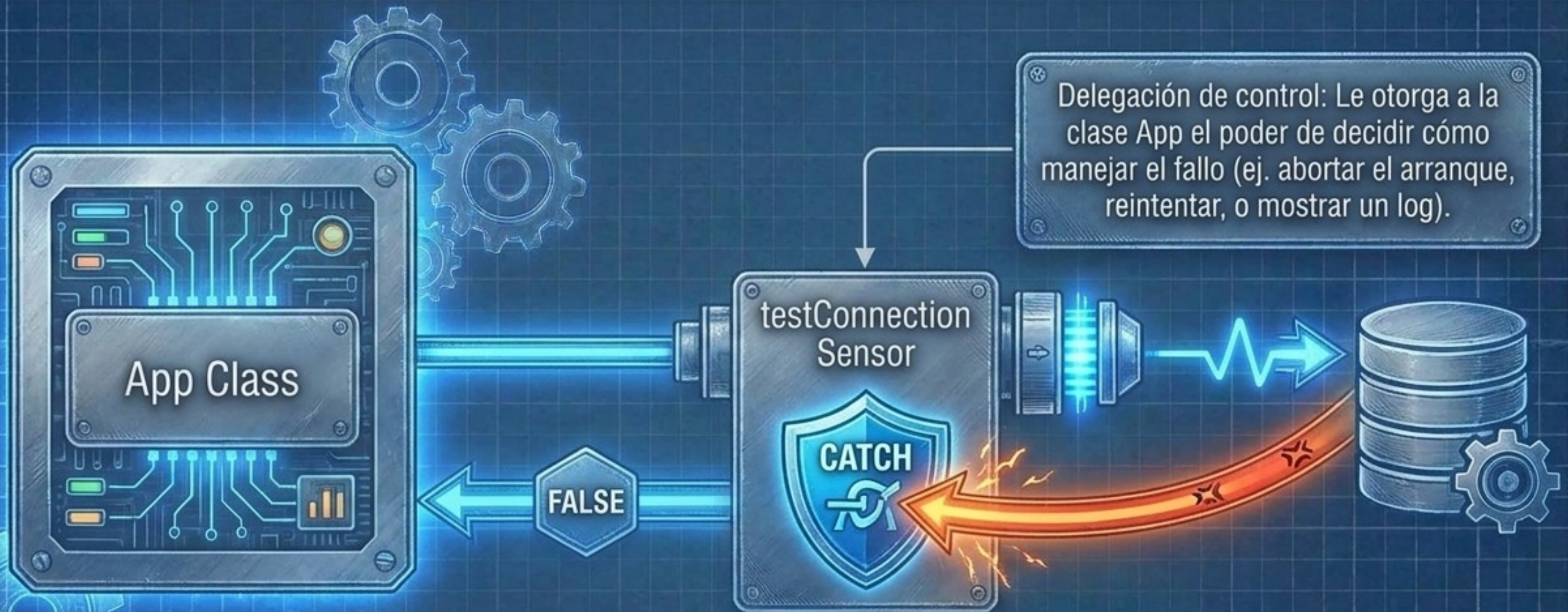
El código valida el motor seleccionado contra el mapa dbConfigurations. Fallar rápido y explícitamente es mejor que intentar conectar a un motor equivocado o no soportado.



PROJECT: INFRAESTRUCTURA CANAL
DRAWN BY: SYSTEM ARCHITECT
DATE: OCT 26, 2024
NEM: A.5

DIAGNÓSTICO CONTROLADO: EL DISEÑO DE `testConnection`

En lugar de lanzar una excepción letal, devuelve `true` o `false` tras ejecutar `sequelize.authenticate()`.



PROJECT: INFRAESTRUCTURA CANAL
DRAWN BY: SYSTEM ARCHITECT
DATE: OCT 26, 2024
NEM: A.6

Reserva espacial: Separación de responsabilidades

Se crea el directorio `src/database/seaders/` sin lógica (hasta el ISS-04).

¿Por qué? Separar la infraestructura pura de conexión de los scripts de generación de datos falsos. Los seeders correrán en paralelo a la capa HTTP.

`src/database/`

`seaders/`

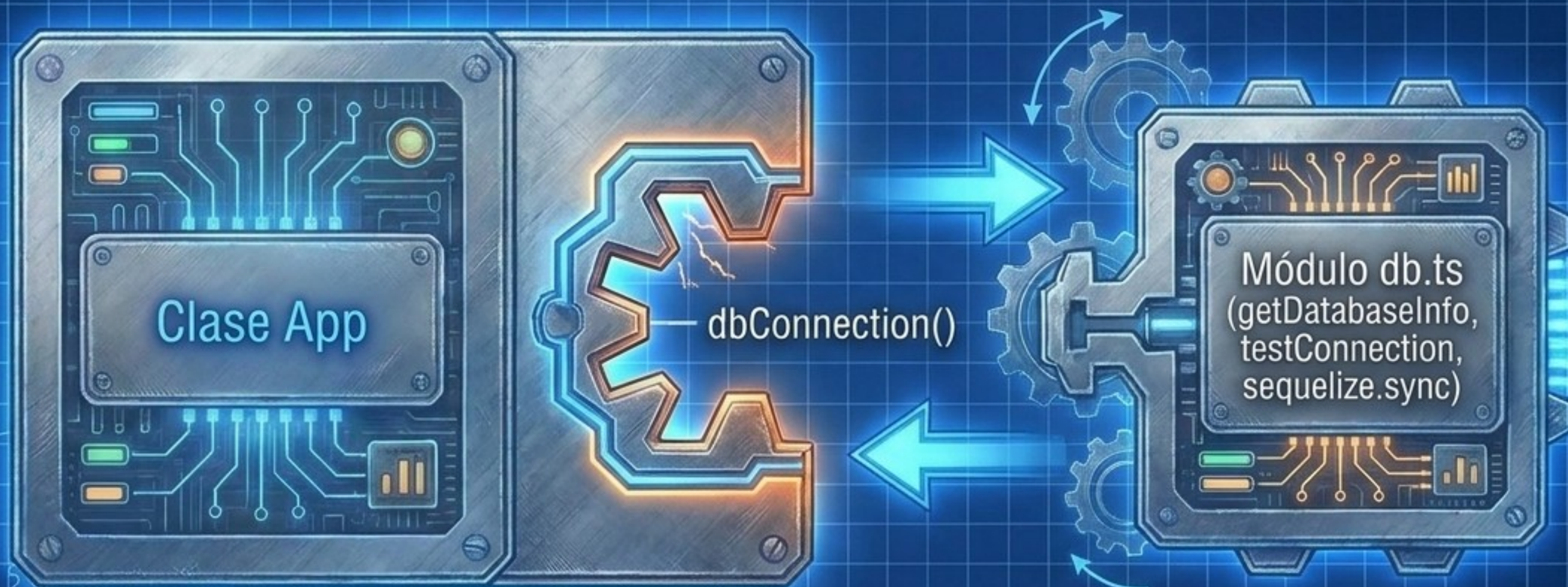
Zona reservada para el
orquestador externo

**UNDER
CONSTRUCTION**

PROJECT: INFRAESTRUCTURA CANAL
DRAWN BY: SYSTEM ARCHITECT
DATE: OCT 26, 2024 NEM: A.7

Acoplamiento estructural: Integración en config/index.ts

La infraestructura de BD se inyecta en la clase principal **App** a través del método privado **dbConnection()**.



**UNDER
CONSTRUCTION**

PROJECT: INFRAESTRUCTURA CANAL
DRAWN BY: SYSTEM ARCHITECT
DATE: OCT 26, 2024
NEM: A.8

La regla de oro del arranque: Orden de encendido

En el método `listen()`, `await this.dbConnection()` debe ocurrir ANTES de `this.app.listen()`.

Primero la **infraestructura**, después atender peticiones.

INICIO

1. Inicializar `dbConnection()`

getDatabaseInfo

testConnection

sync
force/alter

2. Abrir puerto HTTP (`listen`)

Bandera:
Tablas Listas



PROJECT:	INFRAESTRUCTURA CANAL	
DRAWN BY:	SYSTEM ARCHITECT	
DATE:	OCT 26, 2024	NEM: A.9

Colisión estructural: El costo de un arranque desordenado

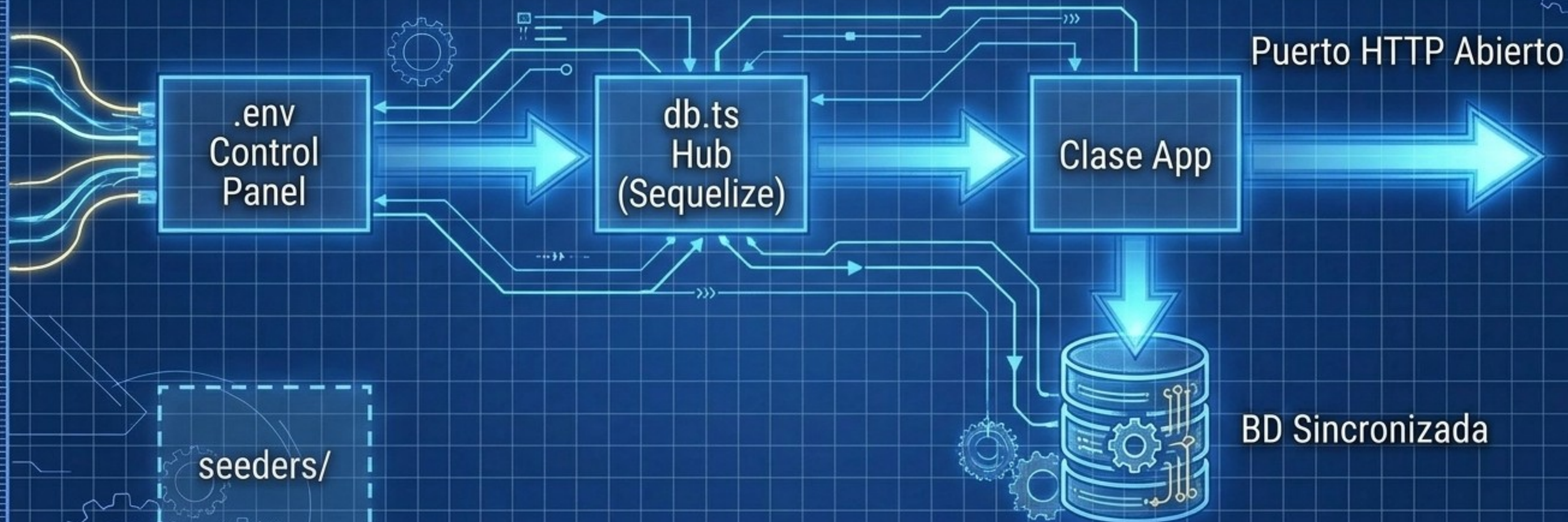
Si el puerto HTTP se abre antes de que termine `sync({ alter: true })`, las peticiones entrantes chocarán con las sentencias DDL (ej. ALTER TABLE, DROP/ADD FOREIGN KEY).

El resultado: **Deadlocks estructurales y errores intermitentes de claves foráneas.**



Síntesis arquitectónica: El canal está abierto

La tubería está lista para recibir los modelos y la lógica de negocio (ISS-03).



**UNDER
CONSTRUCTION**

PROYECTO:	INFRAESTRUCTURA CANAL
ANALIZADO POR:	SYSTEM ARCHITECT
FECHA:	OCT 26, 2024
NÚMERO:	A.11

Criterios GATE: Verificación de la infraestructura

Para dar por finalizado el ISS-02 y avanzar, la infraestructura debe superar la inspección técnica.



Los 3 artefactos de infraestructura existen (**db.ts**, **.env**, **seeders/**)



npx tsc --noEmit: Compilación de tipos limpia (0 errores)



npm run dev: El servidor arranca en runtime sin errores



La conexión y la sincronización con el motor seleccionado son exitosas

Detener el proceso con **Ctrl+C**.

La infraestructura está lista para la lógica de negocio.



PROJEC:	INFRAESTRUCTURA CANAL
DESIGN BY:	SYSTEM ARCHITECT
DATE:	OCT 26, 2024
NEM:	A.10