

MANUAL TÉCNICO

Repositorio

<https://github.com/danielsalguero/PRUEBA-A3.git>

Arquitectura

Contenedor Backend (Node.js + Express)

- El contenedor de **Backend** expone una API RESTful utilizando Node.js y Express. Esta API será la encargada de recibir peticiones desde el **Frontend** y acceder a la **Base de Datos**.
- Además, utilizas **Swagger** para la documentación de la API, lo que permite visualizar las rutas disponibles y los detalles de los endpoints.

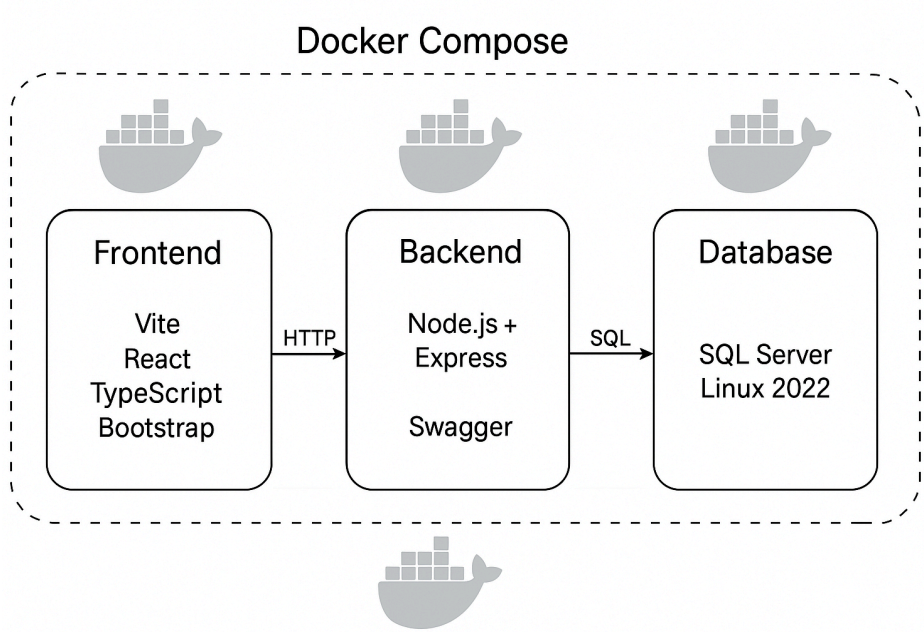
Contenedor Frontend (React, TypeScript, Bootstrap)

- El contenedor de **Frontend** es una aplicación construida con **Vite**, **React**, **TypeScript** y **Bootstrap**.
- Esta aplicación interactúa con la API expuesta por el backend mediante **peticiones HTTP**, presentando los datos al usuario en una interfaz moderna y responsiva.

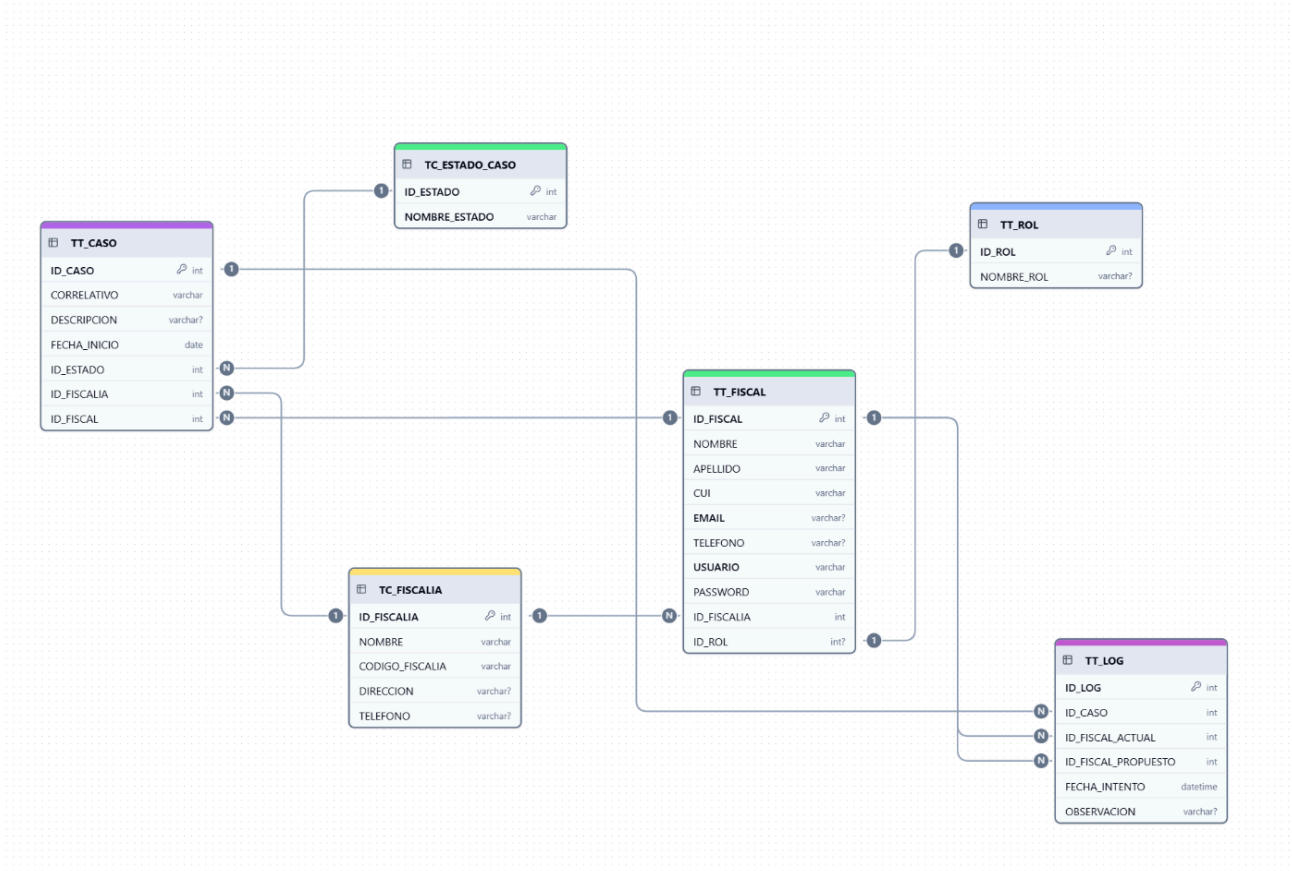
Contenedor Base de Datos (SQL Server Linux 2022)

- El contenedor de **Base de Datos** usa una **imagen de SQL Server en Linux 2022**.
- El backend realiza consultas y operaciones CRUD (Create, Read, Update, Delete) sobre la base de datos, que contiene las tablas y la información necesaria para la aplicación.

Diagrama de Arquitectura



Entidad relación



Base de Datos

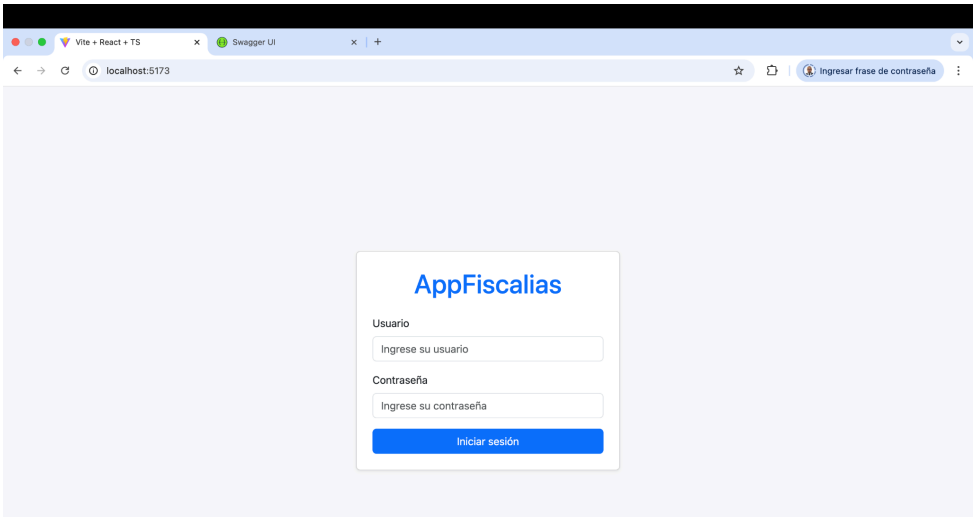
Tabla	Funcion
TC_ROL	Catalogo de Roles
TC_ESTADO_CASO	Catalogo de Estados de Caso
TC_FISCALIA	Catalogo de Fiscalias
TT_CASO	Tabla Transaccional, almacena los casos
TT_FISCAL	Tabla transaccional, almacena los fiscales
TT_LOG	Tabla Transaccional, almacena los logs de asignaciondn de casos.

Procedimiento Almacenado	Funcion
SP_ASIGNAR_FISCAL_A_CASO	Asigna a un fiscal a un caso, este debe de estar en estado pendiente y el nuevo fiscal debe de pertenecer a la misma fiscalia
SP_CAMBIAR_ESTADO_CASO	Cambia el estado de un caso, pendiente, judicializado o cerrado
SP_ELIMINAR_LOG	Elimina un registro de log
SP_LOGIN_FISCAL	Autentica a un usuario para ingresar al sistema
SP_OBTENER_CASOS_POR_FISCALIA	Devuelve todos los casos por fiscalia
SP_OBTENER_FISCALES_POR_FISCALIA	Devuelve todos los fiscales por fiscalia
SP_OBTENER_LOG_ASIGNACIONES	Devuelve todas las asignaciones de casos
SP_OBTENER_TODAS_LAS_FISCALIAS	Devuelve el catalogode todas las fiscalias
SP_OBTENER_TODOS_LOS_CASOS	Devuelve todos los casos
SP_OBTENER_TODOS_LOS_ESTADOS_CASO	Devuelve el catalogo de casos
SP_OBTENER_TODOS_LOS_FISCALES	Devuelve todos los fiscales

Pantallas

Administrador	Igomez	abc123
Fiscal	mramirez	abc123

Login



Gestión de Casos

Fiscalía App

Casos

Logs

Listado de Casos

Asignar Fiscal a Caso

Fiscalía

Seleccione Fiscalía

Fiscal

Seleccione Fiscal

Caso

Seleccione Caso

Asignar Fiscal al Caso

Actualizar Estado de Caso

Caso

Seleccione Caso

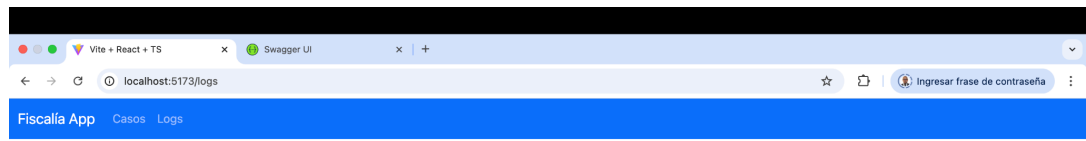
Estado

Seleccione Estado

Actualizar Estado

ID	Correlativo	Fecha Inicio	Fiscalía	Fiscal	Estado
1	MP-001-2025	14/1/2025	Fiscalía Central	Luis Gómez	PENDIENTE
2	MP-002-2025	9/2/2025	Fiscalía Norte	Fernando Martínez	JUDICIALIZADO
3	MP-003-2025	4/3/2025	Fiscalía Sur	Carlos Pérez	PENDIENTE

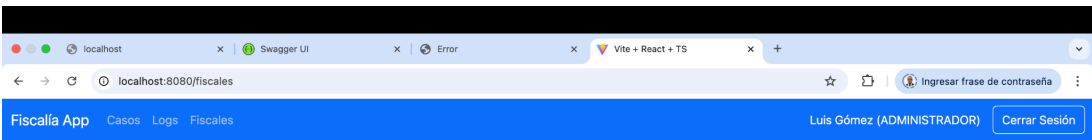
Logs de Asignaciones



Logs de Reasignación

ID LOG	ID CASO	FISCAL ACTUAL	FISCAL PROPUESTO	FECHA	OBSERVACIÓN	ACCIONES
4	5	2	3	1/6/2025, 7:50:12 p.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
5	1	1	2	1/6/2025, 8:34:05 p.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
6	5	2	3	1/6/2025, 10:30:04 p.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
7	5	2	3	1/6/2025, 10:30:14 p.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
13	1	4	2	2/6/2025, 12:48:21 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
14	1	4	2	2/6/2025, 12:51:34 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
15	1	4	2	2/6/2025, 12:51:36 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
16	1	4	2	2/6/2025, 12:52:51 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
17	2	5	3	2/6/2025, 12:54:21 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
18	2	5	3	2/6/2025, 12:55:03 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
19	2	5	3	2/6/2025, 12:55:04 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
20	2	5	1	2/6/2025, 12:55:12 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar
21	2	5	1	2/6/2025, 12:55:15 a.m.	ERROR: Fiscal nuevo no pertenece a la misma fiscalía que el actual.	Eliminar

Fiscales por Fiscalías



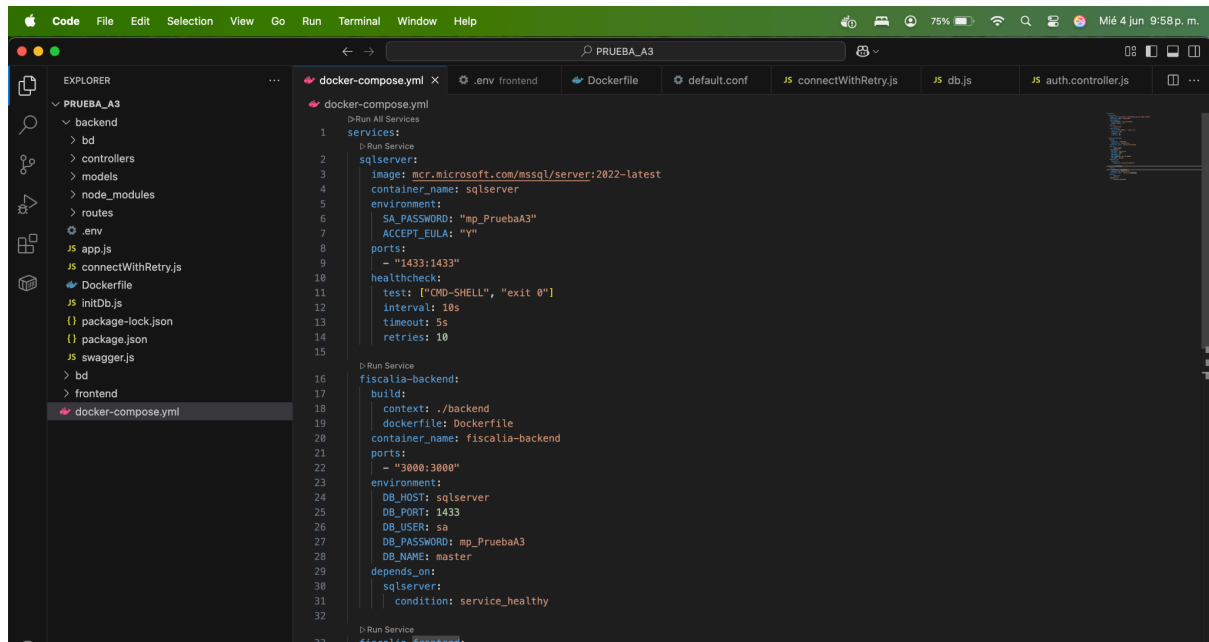
Listado de Fiscales por Fiscalía

Fiscalía
Fiscalía Norte

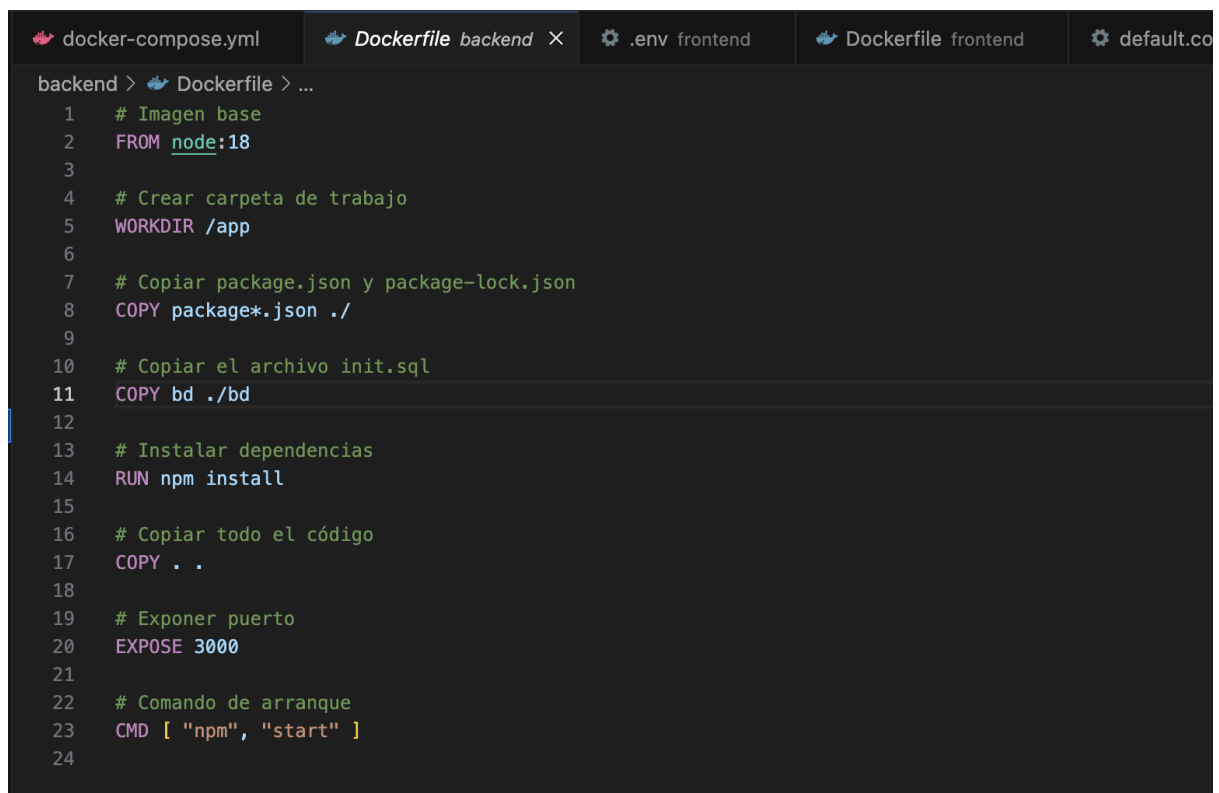
Nombre	CUI	Email	Teléfono
María Ramírez	9876543210987	mramirez@fiscalia.gt	5555-2222
María Lopez	1234567890125	maria.lopez@example.com	555-0002
Fernando Martínez	1234567890126	fernando.martinez@example.com	555-0003

Docker

Docker Compose



Dockerfile Backend



Dockerfile Frontend

```
frontend > Dockerfile > ...
1  # Etapa de build
2  FROM node:18 AS build
3
4  WORKDIR /app
5
6  COPY package*.json ./
7
8  RUN npm install
9
10 COPY . .
11
12 RUN npm run build
13
14 # Etapa de producción con NGINX
15 FROM nginx:alpine
16
17 # Copiar el build al NGINX
18 COPY --from=build /app/dist /usr/share/nginx/html
19 COPY nginx/default.conf /etc/nginx/conf.d/default.conf
20
21 # Exponer puerto
22 EXPOSE 80
23
24 CMD ["nginx", "-g", "daemon off;"]
25
```

Frontend

App.tsx

```
frontend > src > App.tsx > App
1 import { BrowserRouter as Router, Routes, Route, useLocation } from 'react-router-dom';
2 import Navbar from './components/Navbar';
3 import CasosPage from './pages/CasosPage';
4 import LogPage from './pages/LogPage';
5 import LoginPage from './pages/LoginPage';
6 import FiscalesPage from './pages/FiscalesPage';
7
8 function App() {
9   return (
10     <Router>
11       <MainLayout />
12     </Router>
13   );
14 }
15
16 function MainLayout() {
17   const location = useLocation();
18
19   // Determinar si estamos en login ("/") o en otra página
20   const hideNavbar = location.pathname === '/' || location.pathname === '/login';
21
22   return (
23     <>
24       {!hideNavbar && <Navbar />}
25       <Routes>
26         <Route path="/" element={<LoginPage />} />
27         <Route path="/login" element={<LoginPage />} />
28         <Route path="/casos" element={<CasosPage />} />
29         <Route path="/fiscales" element={<FiscalesPage />} />
30         <Route path="/logs" element={<LogPage />} />
31       </Routes>
32     </>
33   );
34 }
```

CasosPage.tsx

```
frontend > src > pages > CasosPage.tsx > ...
1 import { useEffect, useState } from 'react';
2 import { fetchCasos } from '../api/casos';
3 import { fetchFiscalias } from '../api/fiscalias';
4 import { fetchFiscalesByFiscalia } from '../api/fiscales';
5 import { fetchEstados } from '../api/estados';
6 import { asignarFiscalACaso } from '../api/asignarFiscal';
7 import { actualizarEstadoCaso } from '../api/actualizarEstado';
8 import { parseDateString } from '../utils/dateUtils';
9
10 import type { Caso } from '../api/casos';
11 import type { Fiscalia } from '../api/fiscalias';
12 import type { Fiscal } from '../api/fiscales';
13 import type { Estado } from '../api/estados';
14
15 export default function CasosPage() {
16   const [casos, setCasos] = useState<Caso[]>([]);
17   const [fiscalias, setFiscalias] = useState<Fiscalia[]>([]);
18   const [fiscales, setFiscales] = useState<Fiscal[]>([]);
19   const [estados, setEstados] = useState<Estado[]>([]);
20
21   const [selectedFiscalia, setSelectedFiscalia] = useState<number | ''>('');
22   const [selectedFiscal, setSelectedFiscal] = useState<number | ''>('');
23   const [selectedCaso, setSelectedCaso] = useState<number | ''>('');
24
25   const [selectedCasoEstado, setSelectedCasoEstado] = useState<number | ''>('');
26   const [selectedEstado, setSelectedEstado] = useState<number | ''>('');
27 }
```


Backend

App.js

```
backend > JS app.js > ...
1  const express = require('express');
2  const cors = require('cors');
3  const dotenv = require('dotenv');
4  const fiscalesRoutes = require('./routes/fiscales.routes');
5  const casosRoutes = require('./routes/casos.routes');
6  const authRoutes = require('./routes/auth.routes');
7  const fiscaliasRoutes = require('./routes/fiscalias.routes');
8  const logsRoutes = require('./routes/logs.routes');
9  const { swaggerUi, swaggerSpec } = require('./swagger');
10 const connectWithRetry = require('./connectWithRetry');
11
12 dotenv.config();
13
14 const app = express();
15
16 app.use(cors());
17 app.use(express.json());
18
19 app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerSpec));
20 app.use('/api', fiscalesRoutes);
21 app.use('/api', casosRoutes);
22 app.use('/api', authRoutes);
23 app.use('/api', fiscaliasRoutes);
24 app.use('/api', logsRoutes);
25
26 const PORT = process.env.PORT || 3000;
27
28 // Inicio del servidor después del retry de la BD
29 connectWithRetry().then(() => {
30   app.listen(PORT, () => {
31     console.log(`Servidor corriendo en puerto ${PORT}`);
32   });
33 });
```

auth.controller.js

```
backend > controllers > JS auth.controller.js > login > pool
1
2  const { getPool, sql } = require('../models/db');
3
4  const login = async (req, res) => {
5    const { usuario, password } = req.body;
6
7    try{
8
9      const pool = await getPool();
10     const result = await pool.request()
11       .input('USUARIO', usuario)
12       .input('PASSWORD', password)
13       .execute('SP_LOGIN_FISCAL');
14
15     const user = result.recordset[0];
16
17     if (user) {
18       res.json({
19         success: true,
20         fiscal: {
21           ID_FISCAL: user.ID_FISCAL,
22           NOMBRE: user.NOMBRE,
23           APELLIDO: user.APELLIDO,
24           ID_ROL: user.ID_ROL,
25           NOMBRE_ROL: user.NOMBRE_ROL
26         }
27       });
28     } else {
29       res.status(401).json({ success: false, message: 'Usuario o contraseña incorrectos' });
30     }
31   }
32 }
```