

# SentinelPro

Solución completa para la gestión de impresoras y registro documental



Autenticación Segura



Multi-Tenant



Análisis de Consumo

# Introducción

- **Problema de trazabilidad documental en PyMEs.**

- Falta de control centralizado.
- Falta de registros permanentes.
- Mayor consumo de recursos
- Riesgos legales y operativos



- **Solución propuesta:** Sentinel Pro es una plataforma web multi-tenant que centraliza la gestión, rastrea trabajos y genera reportes en tiempo real.



# Situación Actual

- **Deficiencias en el registro.**

- Registros incompletos.
- Falta de historial.
- Análisis limitado.



- **Ausencia de datos claves.**



- **Riesgos legales y operativos.**

- Riesgos legales.
- Riesgos operativos.
- Riesgos de seguridad.



# Alternativa de Solución

Las soluciones actuales para el control de impresiones en PyMEs presentan varias limitaciones

## Gestores Locales



## Software de Escritorio



## Soluciones Propietarias



### Limitaciones:

- ✖ No se pueden acceder desde otros dispositivos
- ✖ Problemas de mantenimiento frecuentes
- ✖ No se puede escalar a múltiples impresoras

### Limitaciones:

- ✖ Acceso restringido a la impresora
- ✖ No se puede gestionar desde cualquier lugar
- ✖ Complejidad para administrarlo a distancia

### Limitaciones:

- ✖ Alto costo de implementación y mantenimiento
- ✖ Dependencia de un único proveedor
- ✖ Modelos de licenciamiento restrictivos

# Solución a Implementar

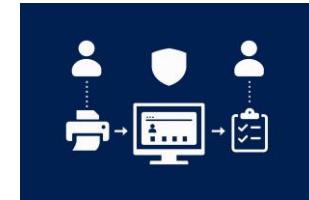
- Autenticación segura (JWT + bcrypt).



- 4 niveles de roles jerárquicos.



- Gestión multiempresa con aislamiento.



- Tracking completo de trabajos.



- Analytics en tiempo real.



- Reportes por periodo.

# Descripción del Sistema de Información

- Sentinel Pro está diseñado bajo una arquitectura cliente-servidor desacoplada, con enfoque cloud-native multi-tenant, orientada a la escalabilidad horizontal.
- **Frontend:** SPA desarrollada en React 18 + TypeScript, con TailwindCSS y Shadcn/UI para componentes reutilizables.
- **Backend:** API RESTful en Node.js + Express + TypeScript, utilizando Drizzle ORM para abstracción de acceso a datos.
- **Base de Datos:** PostgreSQL Neon serverless, con esquema relacional normalizado en BCNF.
- **Autenticación:** Implementación de JWT con bcrypt para hashing seguro, y validación Zod en cliente y servidor.
- **Hosting:** Render.com con CI/CD automático desde GitHub.

# Modelo Entidad Relación

- **Entidades:**

- Company
- User
- Printer
- PrintJob

- **Relaciones:**

- Company 1:N User
- Company 1:N Printer
- User 1:N PrintJob
- Printer 1:N PrintJob



## Requerimientos Funcionales

ID	NOMBRE REQUERIMIENTO
RF-001	Autenticación de usuarios
RF-002	Gestión de usuarios
RF-003	Gestión de roles
RF-004	CRUD impresoras
RF-005	CRUD trabajos de impresión
RF-006	Dashboard estadísticas
RF-007	Reportes por periodo
RF-008	Filtrado de datos
RF-009	Carga de archivos
RF-010	Multi-tenant

## Requerimientos No Funcionales

ID	NOMBRE REQUERIMIENTO
RNF-001	Seguridad: JWT + bcrypt + HTTPS
RNF-002	Performance: respuesta <200ms
RNF-003	Escalabilidad: 1000+ usuarios
RNF-004	Disponibilidad: 99.5% uptime
RNF-005	Confidencialidad: datos encriptados
RNF-006	Integridad: validación cliente/servidor
RNF-007	Usabilidad: interfaz intuitiva
RNF-008	Auditoría: registro de cambios

# Modelo No Relacional

- **¿Qué es?**

- Es un tipo de base de datos que no utiliza tablas ni relaciones tradicionales.

- Almacena la información en documentos tipo JSON, lo que permite mayor flexibilidad y escalabilidad.

- **Como por ejemplo:**

```
{
```

```
  "usuario": "Carlos",
```

```
  "impresora": "PR-001",
```

```
  "documento": "Informe.pdf",
```

```
  "paginas": 29,
```

```
  "estado": "Completado",
```

```
  "fecha": "2025-11-30"
```

- }

# Aspectos de Implementación

- **Tecnologías Utilizadas:**

Frontend: React 18 + TypeScript + TailwindCSS + Shadcn/UI

Backend: Node.js + Express + TypeScript + Drizzle ORM

Base de Datos: PostgreSQL Neon (serverless, escalable)

Autenticación: JWT + bcrypt

Validación: Zod (cliente y servidor)

ORM: Drizzle con consultas parametrizadas (prevención de SQL injection)

Hosting: Render.com con auto-deploy desde GitHub

- **Seguridad:**

Cifrado de contraseñas con bcrypt

Tokens JWT con expiración y firma segura

HTTPS obligatorio en toda la plataforma

Validación de entrada en frontend y backend

Aislamiento multiempresa garantizado por claves foráneas y middleware de contexto

# Aspectos de Implementación

- **DevOps y Deploy:**

CI/CD automático con GitHub → Render

Entorno serverless con escalabilidad horizontal

Uptime garantizado: 99.5%

Logs de auditoría y monitoreo de errores en tiempo real

- **Estructura del Proyecto:**

Separación por capas: presentación, lógica de negocio, persistencia

Modularización por dominio: auth, users, printers, jobs, analytics

Reutilización de componentes UI y validadores

# Conclusiones

- **Sentinel Pro** fue desarrollado y desplegado exitosamente en entorno productivo.
- El sistema cumple con los requerimientos funcionales y no funcionales definidos en el análisis.
- Se logró una arquitectura multi-tenant segura, con aislamiento completo entre empresas.
- Se implementaron módulos clave: autenticación, gestión de usuarios, impresoras, trabajos y dashboard de analytics.
- La solución está lista para ser utilizada por PyMEs reales, con escalabilidad garantizada.
- Se validó el uso de tecnologías modernas: React, Node.js, PostgreSQL Neon, JWT, bcrypt, Zod, Render.
- El modelo relacional aplicado permite integridad, trazabilidad y auditoría completa.

- **Acceso a la plataforma**

**URL:** <https://sentinelpro.onrender.com>

**Usuario demo:** sentinelpro **Contraseña:** 123456

# Tema

Seminario de Título/Proyecto de Título/Proyecto de Titulación

NOMBRE ESTUDIANTE:  
CARRERA:  
SEDE: CED  
PROFESOR DEL MÓDULO:



**4** INSTITUCION  
**ACREDITADA** |  
NIVEL AVANZADO  
AÑOS Hasta octubre 2025  
GESTIÓN INSTITUCIONAL Y DOCENCIA DE PREGRADO

