Imagen de la pantalla de un computador portátil

Descripción generada automáticamente con confianza media

PROYECTO DE TITULO

Título de Proyecto: CloudMusic Digital Finance – Plataforma Electrónica Digital Automatizada

Sistema Integral de Emisión de Documentos Tributarios Electrónicos

con Asistente IA para PyMEs Chilenas

MÓDULO: Informática y Telecomunicaciones

Bimestre/Año: 4to Bimestre año 2025

Semana: 3

Docente: Aída Villamar Gallardo

Estudiante: Héctor Brito Tapia

Índice

[Capítulo I: Introducción 6](#_Toc214457036)

[1.1 Descripción breve del proyecto 6](#_Toc214457037)

[1.2 Contexto y relevancia del proyecto 7](#_Toc214457038)

[Capítulo II: Antecedentes del Proyecto 9](#_Toc214457039)

[2.1 Planteamiento del problema 9](#_Toc214457040)

[2.2 Justificación del proyecto 11](#_Toc214457041)

[2.3 Propuesta de solución 12](#_Toc214457042)

[2.4 Propuesta de valor 14](#_Toc214457043)

[2.5 Enlace de acceso al desarrollo 15](#_Toc214457044)

[Capítulo III: Objetivos y Alcance 16](#_Toc214457045)

[3.1 Objetivo general 16](#_Toc214457046)

[3.2 Objetivos específicos 17](#_Toc214457047)

[3.3 Límites y alcance del proyecto 19](#_Toc214457048)

[Capítulo IV: Análisis del Proyecto 22](#_Toc214457049)

[4.1 Requerimientos del sistema 22](#_Toc214457050)

[4.1.1 Requerimientos funcionales 23](#_Toc214457051)

[4.1.2 Requerimientos no funcionales 27](#_Toc214457052)

[4.2 Análisis de factibilidad 29](#_Toc214457053)

[4.2.1 Factibilidad técnica mínima y óptima 29](#_Toc214457054)

[4.2.2 Factibilidad operativa 35](#_Toc214457055)

[4.3 Planificación del proyecto 36](#_Toc214457056)

[4.3.1 Cronograma de actividades (Carta Gantt) 37](#_Toc214457057)

[Capítulo V: Diseño de sistemas y modelado 38](#_Toc214457058)

[5.1 Base de Datos 38](#_Toc214457059)

[5.1.1 Selección de base de datos 38](#_Toc214457060)

[5.1.2 Justificación de la elección entre SQL y NoSQL 38](#_Toc214457061)

[5.1.3 Uso de la base de datos seleccionada 40](#_Toc214457062)

[5.2 Diagrama Entidad–Relación (ER) 42](#_Toc214457063)

[5.3 Casos de Uso 45](#_Toc214457064)

[5.4 Diseño de Árbol Funcional e Interfaz de Usuario 48](#_Toc214457065)

[5.5 Diseño de Prototipos 51](#_Toc214457066)

[5.6 Metodología de Desarrollo 54](#_Toc214457067)

[5.7 Fundamento y Elección del Modelo de Datos 58](#_Toc214457068)

[5.7.1 Contexto del Proyecto 58](#_Toc214457069)

[5.7.2 Comparación de Modelos de Datos 59](#_Toc214457070)

[5.7.3 Modelo Seleccionado 59](#_Toc214457071)

[6. Capítulo VI: Modelado de datos 60](#_Toc214457072)

[6.1 Fundamento y elección del modelo de datos 60](#_Toc214457073)

[6.2 Modelado de la base de datos 60](#_Toc214457074)

[6.2.1 Tablas Principales (9) 60](#_Toc214457075)

[6.2.2 Colecciones *Principales* (5) – MongoDB 62](#_Toc214457076)

[6.2.3 Relaciones y Cardinalidades 64](#_Toc214457077)

[6.2.4 Normalización hasta 3FN 65](#_Toc214457078)

[6.2.5 Constraints y Triggers Claves 65](#_Toc214457079)

[6.2.6 Flujos de integración 66](#_Toc214457080)

[6.2.7 Justificación técnica 66](#_Toc214457081)

[6.3 Diccionario de Datos 66](#_Toc214457082)

[6.3.1 Tabla: users 66](#_Toc214457083)

[6.3.2 Tabla: companies 68](#_Toc214457084)

[6.2.3 Tabla: company\_users 69](#_Toc214457085)

[6.3.4 Tabla: clients 70](#_Toc214457086)

[6.3.5 Tabla: products 71](#_Toc214457087)

[6.3.6 Tabla: certificates 72](#_Toc214457088)

[6.3.7 Tabla: folios 73](#_Toc214457089)

[6.3.8 Tabla: documents 74](#_Toc214457090)

[6.3.9 Tabla: document\_items 76](#_Toc214457091)

[6.3.10 Colección: ai\_document\_analysis 77](#_Toc214457092)

[6.3.11 Colección: audit\_trail 77](#_Toc214457093)

[6.3.12 Colección: chat\_sessions 78](#_Toc214457094)

[6.3.13 Colección: sii\_responses 78](#_Toc214457095)

[6.3.14 Colección: websocket\_events 79](#_Toc214457096)

[7. Capítulo VII: Desarrollo e Implementación de Sistemas 80](#_Toc214457097)

[7.1 Arquitectura del Sistema 80](#_Toc214457098)

[7.1.1 Visión General (Arquitectura 3 Capas) 80](#_Toc214457099)

[7.1.2 Componentes Principales 80](#_Toc214457100)

[7.1.3 Seguridad y Cumplimiento 80](#_Toc214457101)

[7.1.4 Flujo de Información 81](#_Toc214457102)

[7.2 Implementación del Sistema Basada en la Arquitectura Definida 81](#_Toc214457103)

[Capítulo VIII: Plan de Prueba 82](#_Toc214457104)

[8.1 Formato Preliminar del Documento 82](#_Toc214457105)

[8.1.1 Introducción al Plan de Prueba 82](#_Toc214457106)

[8.1.2 Objetivo General del Plan de Pruebas 82](#_Toc214457107)

[8.1.3 Alcance 82](#_Toc214457108)

[8.1.4 Propósito 83](#_Toc214457109)

[8.2 Plan de Pruebas 83](#_Toc214457110)

[8.2.1 Características que serán probadas 83](#_Toc214457111)

[8.2.2 Características que no serán probadas 84](#_Toc214457112)

[8.2.3 Criterios de aprobación y fallo 85](#_Toc214457113)

[8.2.4 Criterios de suspensión y reanudación 85](#_Toc214457114)

[8.2.5 Casos de prueba y tareas (Pruebas Manuales) 86](#_Toc214457115)

[8.2.6 Evidencias de Ejecución 88](#_Toc214457116)

[Capítulo IX: Análisis de Resultados y Verificación de Objetivos 97](#_Toc214457117)

[9.1 Análisis de Resultados 97](#_Toc214457118)

[9.1.1 Implementación técnica del sistema 97](#_Toc214457119)

[9.1.2 Métricas de rendimiento alcanzadas 98](#_Toc214457120)

[9.1.3 Cumplimiento normativo con el SII 98](#_Toc214457121)

[9.2 Verificación de Objetivos 99](#_Toc214457122)

[9.2.1 Objetivo general 99](#_Toc214457123)

[9.2.2 Objetivos específicos – Análisis de cumplimiento 100](#_Toc214457124)

[9.2.3 Resultados cuantitativos del proyecto 101](#_Toc214457125)

[9.2.4 Impacto y valor agregado del sistema 102](#_Toc214457126)

[9.2.5 Conclusiones del análisis 102](#_Toc214457127)

[Capítulo X: Conclusión y Reflexión 104](#_Toc214457128)

[10.1 Reflexión sobre el proyecto y conclusiones en base al trabajo realizado 104](#_Toc214457129)

[10.1.1 Logros alcanzados y aprendizajes clave 104](#_Toc214457130)

[10.1.2 Desafíos superados y lecciones aprendidas 105](#_Toc214457131)

[10.1.3 Impacto profesional y desarrollo de competencias 106](#_Toc214457132)

[10.1.4 Contribución al ecosistema tecnológico nacional 107](#_Toc214457133)

[10.1.5 Perspectivas futuras y recomendaciones 108](#_Toc214457134)

[10.1.6 Reflexión personal y profesional 109](#_Toc214457135)

[Referencias 110](#_Toc214457136)

[11.1 Fuentes utilizadas para respaldar el diseño y desarrollo del proyecto. 110](#_Toc214457137)

Capítulo I: Introducción

1.1 Descripción breve del proyecto

CloudMusic Digital Finance se presenta como la **primera plataforma chilena de Documentos Tributarios Electrónicos (DTE) con inteligencia artificial local integrada**, diseñada para optimizar y automatizar los procesos de emisión, validación y envío de documentos tributarios bajo los lineamientos del Servicio de Impuestos Internos (SII). Su objetivo principal es ofrecer una solución inteligente, eficiente y autónoma que reduzca los errores humanos y mejore la trazabilidad fiscal en tiempo real, fortaleciendo la transformación digital de las PyMEs chilenas.

El sistema está construido sobre un stack tecnológico escalable y de alto rendimiento, basado en React 18 + Next.js 15 para la interfaz de usuario, Node.js + Express + TypeScript para la capa de datos con PostgreSQL 16, junto con la integración local del modelo Ollama Llama 3.2 3B para el procesamiento de lenguaje natural y la asistencia tributaria contextual. Esta configuración híbrida permite la ejecución de la IA sin depender de APIs externas, asegurando privacidad, independencia tecnológica y un rendimiento constante en entornos empresariales.

En cuanto a sus funcionalidades, la plataforma incorpora dos perfiles de usuario (Administrador y Usuario, expandible a nueve roles) y seis formularios principales completos orientados a la gestión integral de DTE: gestión de empresas, clientes, productos/servicios, emisión de DTE, folios CAF y reportes avanzados, más un séptimo formulario de asistente inteligente basado en chat con IA. Además, integra un asistente impulsado por Ollama que guía al usuario en la interpretación de datos, validación de totales e identificación de inconsistencias tributarias en tiempo real.

El público objetivo está conformado por **pequeñas y medianas empresas chilenas** que buscan un sistema automatizado y escalable para la gestión tributaria electrónica, con una arquitectura empresarial preparada para el crecimiento. Gracias a su arquitectura modular y escalable, CloudMusic Digital Finance puede evolucionar progresivamente desde un enfoque monolítico hacia microservicios, ampliando sus funcionalidades sin comprometer la estabilidad ni la mantenibilidad del sistema.

Finalmente, el proyecto se distingue por su enfoque innovador y sustentable, al combinar inteligencia artificial local sin dependencias de APIs externas, Socket.IO para comunicación bidireccional en tiempo real escalable con Redis clustering, y una arquitectura modular microservices-ready. Estas características posicionan a CloudMusic Digital Finance como una propuesta tecnológicamente avanzada y escalable, alineada con los principios de calidad, fiabilidad y escalabilidad empresarial propuestos por Pressman (2019) y Sommerville (2020) en la ingeniería de software contemporánea.

1.2 Contexto y relevancia del proyecto

La obligatoriedad de la facturación electrónica en Chile ha transformado profundamente el panorama tributario nacional. Desde enero de 2022, el Servicio de Impuestos Internos (SII) exige que todas las empresas —sin importar su tamaño o rubro— emitan Documentos Tributarios Electrónicos (DTE) conforme a los esquemas XML definidos por las resoluciones vigentes (*DTE\_v10.xsd* y *EnvioBOLETA\_v11.xsd*).

Esta normativa impacta directamente a más de 180 000 micro, pequeñas y medianas empresas (PyMEs), que deben adaptar sus sistemas internos para cumplir con los requisitos técnicos y legales establecidos por el SII. Sin embargo, de acuerdo con el Instituto Nacional de Estadísticas (INE, 2024), alrededor del 68 % de las PyMEs chilenas todavía emplea métodos manuales o semiautomáticos para sus procesos de facturación, y cerca del 45 % reconoce dificultades para implementar soluciones digitales ajustadas a la normativa tributaria vigente.

El **tiempo promedio de generación manual** de un documento tributario oscila entre **35 y 50 minutos**, mientras que mediante sistemas automatizados puede reducirse a **menos de 5 minutos**, lo que demuestra el alto impacto operativo de la digitalización en la eficiencia administrativa.

Desde el punto de vista económico, la falta de automatización genera un costo promedio mensual estimado entre $25 000 y $150 000 CLP por empresa, principalmente en multas y reprocesos derivados de errores en la validación de documentos. Ante este contexto, CloudMusic Digital Finance surge como la primera solución DTE con inteligencia artificial local Ollama en Chile, eliminando dependencias externas y mejorando la privacidad de los datos tributarios.

La relevancia del proyecto se manifiesta en tres dimensiones clave:

* Normativa y técnica: garantiza la validación XML conforme a los esquemas oficiales del SII, incorporando firma digital, trazabilidad completa y control de autenticidad documental, diferenciándose por la integración de seis formularios completos más un asistente IA frente a las soluciones fragmentadas del mercado.
* Económica: reduce hasta un 70 % los tiempos administrativos y un 85 % los errores de validación, con un potencial ahorro nacional estimado en más de $2,8 billones CLP anuales. La IA local Ollama elimina los costos recurrentes asociados a APIs externas.
* Social y tecnológica: democratiza el acceso a herramientas empresariales avanzadas con arquitectura modular escalable y preparada para crecimiento enterprise, impulsa la transformación digital de las PyMEs y fomenta la generación de empleo especializado en tecnologías tributarias.

Asimismo, CloudMusic Digital Finance se alinea con las tendencias internacionales de automatización contable y gobierno digital. Según Gartner Research (2025), se proyecta que cerca del 75 % de las plataformas financieras incorporarán inteligencia artificial antes de 2026, tendencia que este proyecto adopta desde su diseño conceptual con Inteligencia artificial local.

En conjunto, la iniciativa responde no solo a una necesidad real del mercado, sino también a un desafío tecnológico, al integrar los principios de la ingeniería de software (Pressman, 2019; Sommerville, 2020) con las normas de calidad de producto establecidas por la ISO/IEC 25010 (2018), contribuyendo a la innovación, eficiencia operativa y sostenibilidad tecnológica dentro del contexto empresarial chileno.

Capítulo II: Antecedentes del Proyecto

2.1 Planteamiento del problema

En Chile, el proceso de facturación electrónica se encuentra regulado por el Servicio de Impuestos Internos (SII) y constituye una obligación legal desde la Resolución Exenta N.º 16 (2022), que establece la emisión exclusiva de Documentos Tributarios Electrónicos (DTE) en formato XML firmado digitalmente.

Sin embargo, este avance normativo ha generado importantes desafíos para las pequeñas y medianas empresas (PyMEs), que en muchos casos carecen de infraestructura tecnológica, recursos humanos especializados y herramientas accesibles para cumplir con las exigencias del SII.

De acuerdo con el Instituto Nacional de Estadísticas (INE, 2024), un 68 % de las PyMEs chilenas aún depende de procesos manuales o semiautomatizados para emitir facturas, lo que deriva en errores de registro, duplicidad de datos y altos costos operativos. Estas dificultades se agravan por tres factores estructurales principales:

* Errores en la validación XML por incumplimiento de los esquemas oficiales (*DTE\_v10.xsd* y *EnvioBOLETA\_v11.xsd*), agravados por la fragmentación de las soluciones actuales, que no cubren los seis formularios integrados necesarios para una gestión DTE completa.
* Ausencia de trazabilidad documental, que impide verificar el ciclo completo de envío y aceptación ante el SII, junto con la necesidad de inteligencia artificial local sin dependencia de APIs externas para ofrecer asistencia tributaria contextual.
* Falta de auditoría digital automatizada, lo que dificulta la detección de inconsistencias tributarias y la transparencia contable, problemática que la IA local Ollama puede resolver al eliminar costos recurrentes y mejorar la privacidad de los datos tributarios.

Según el Ministerio de Economía (2023) y CORFO (2023), la adopción de soluciones digitales en las PyMEs chilenas permite reducir significativamente los tiempos y errores administrativos; sin embargo, enfrenta barreras persistentes relacionadas con los costos de implementación, la falta de capacitación técnica y la resistencia organizacional al cambio.

De igual modo, el estudio de la OECD (2024) sobre transformación digital en América Latina identifica que la carencia de herramientas integradas para la gestión tributaria en pequeñas empresas constituye una de las principales causas de incumplimiento normativo y baja productividad en el sector.

En consecuencia, la problemática principal que aborda este proyecto se sintetiza en la siguiente **pregunta de investigación**:

¿Cómo crear una solución DTE completa con inteligencia artificial local integrada y escalabilidad enterprise que permita a las PyMEs gestionar seis formularios principales de manera integral, mediante una arquitectura modular microservices-ready?

2.2 Justificación del proyecto

El proyecto CloudMusic Digital Finance surge como respuesta a la necesidad de contar con una solución tecnológica integral y accesible que unifique el cumplimiento tributario electrónico con la automatización inteligente de procesos contables. Esta iniciativa propone un enfoque innovador que combina arquitectura escalable, validación normativa automatizada e inteligencia artificial local para fortalecer la transformación digital de las pequeñas y medianas empresas (PyMEs) en Chile.

**Relevancia técnica y social**

La implementación de sistemas tributarios automatizados contribuye directamente a la eficiencia, transparencia y trazabilidad en la gestión financiera. Según CORFO (2023), la digitalización contable puede reducir los tiempos de emisión y registro en más de un 75 %, y disminuir los errores humanos hasta en un 90 %.

En este contexto, CloudMusic Digital Finance busca **fortalecer la competitividad de las PyMEs** mediante una plataforma que simplifique la interacción con el Servicio de Impuestos Internos (SII), optimice los flujos tributarios y permita un **control integral, seguro y trazable** de los documentos electrónicos.

**Criterios de factibilidad**

1. Factibilidad técnica:  
   El ecosistema actual, basado en Node.js, React, Python, PostgreSQL y MongoDB, permite construir una arquitectura cloud-native, modular y escalable, garantizando integridad de datos, alta disponibilidad y seguridad transaccional.
2. Factibilidad normativa:  
   El sistema contempla la validación estructural y semántica de archivos XML conforme a los esquemas exigidos por el SII, incorporando firmas digitales RSA de 2048 bits y certificados electrónicos empresariales, en cumplimiento con la Ley N.º 19.983 y el Decreto 570 del Ministerio de Hacienda.
3. Factibilidad económica y social:  
   Se estima una reducción de costos operativos entre $25 000 y $150 000 CLP mensuales por empresa, junto con una disminución del tiempo de emisión de documentos desde 35 minutos a menos de 5 minutos por transacción, lo que incrementa la productividad y reduce los riesgos por incumplimiento normativo.

**Relevancia académica y profesional**

Desde el punto de vista académico, el proyecto aplica los principios de la ingeniería de software definidos por Pressman (2019) y Sommerville (2020), destacando la importancia de la iteración continua, la validación progresiva y la medición objetiva de la calidad del producto.

Asimismo, CloudMusic Digital Finance contribuye al fortalecimiento de las competencias profesionales en áreas como desarrollo ágil, arquitectura de microservicios e inteligencia artificial aplicada a procesos tributarios, alineándose con las políticas nacionales de transformación digital impulsadas por el Estado de Chile y organismos de fomento como CORFO y SERCOTEC.

2.3 Propuesta de solución

La propuesta consiste en desarrollar una plataforma SaaS integral que permita a las empresas **emitir, validar, firmar y enviar Documentos Tributarios Electrónicos (DTE)** cumpliendo los estándares del Servicio de Impuestos Internos (SII), incorporando además un componente de inteligencia artificial local Ollama destinado al análisis contable y la detección de inconsistencias tributarias en tiempo real.

**Características principales**

* Seis formularios integrados para una gestión completa de DTE sin fragmentación, que incluyen: gestión de empresas, clientes, productos/servicios, emisión de DTE, folios CAF y reportes avanzados.
* Inteligencia artificial local Ollama Llama 3.2 3B, sin dependencias externas, utilizada para la asistencia tributaria contextual y el análisis predictivo contable.
* Arquitectura modular y escalable, con un migration path definido desde un sistema monolítico hacia microservicios, garantizando escalabilidad enterprise.
* Socket.IO para notificaciones en tiempo real, escalable a Redis clustering para soportar alta concurrencia de usuarios.
* PostgreSQL como base de datos principal, con más de diez tablas normalizadas en 3NF, complementada por MongoDB Analytics para procesamiento analítico y auditorías.
* Diseño CSS responsive mobile-first, totalmente personalizado, sin dependencias de frameworks externos como Bootstrap.
* Despliegue cloud híbrido, utilizando Vercel y Railway, con migración preparada para Kubernetes en escenarios de escalamiento automático.
* Automatización completa del flujo DTE, desde la generación del documento hasta la recepción del acuse de recibo oficial del SII.
* Validación XML avanzada, con cumplimiento estructural y semántico de los esquemas oficiales (*DTE\_v10.xsd* y *EnvioBOLETA\_v11.xsd*).
* Firma digital empresarial, mediante certificados .pfx cifrados bajo RSA-2048, en conformidad con las normativas chilenas de facturación electrónica.
* Auditoría automática y trazabilidad completa, que garantiza control total sobre los documentos emitidos, modificados y enviados.
* Diseño centrado en el usuario, con métricas de usabilidad medibles: tiempo promedio de tarea inferior a 3 minutos y tasa de error menor al 2 %.
* Modelo multi-tenant, con soporte simultáneo a múltiples empresas y aislamiento de datos por instancia para asegurar privacidad y seguridad.

El desarrollo del sistema se llevará a cabo bajo la metodología ágil Scrum, estructurando sprints iterativos de tres semanas que permitan la validación continua, retroalimentación con los usuarios y mejora incremental del producto, garantizando una evolución constante hacia los objetivos funcionales y técnicos definidos.

2.4 Propuesta de valor

CloudMusic Digital Finance ofrece una propuesta diferenciadora en el mercado chileno al integrar innovación tecnológica, cumplimiento normativo y usabilidad avanzada dentro de una única plataforma. Su diseño combina automatización tributaria, inteligencia artificial local y experiencia de usuario medible, generando ventajas competitivas sostenibles para las PyMEs.

**Ejes de innovación**

1. Cumplimiento normativo automatizado:  
   Garantiza la correcta emisión, validación y firma de DTE conforme a la normativa vigente del SII, reduciendo los errores de validación en más del 90 % mediante validaciones estructurales, semánticas y lógicas automatizadas.
2. Inteligencia artificial aplicada:  
   Emplea algoritmos predictivos locales basados en Ollama para anticipar comportamientos tributarios, detectar inconsistencias de validación y sugerir acciones preventivas que optimicen la gestión contable y fiscal.
3. Experiencia de usuario medible:  
   La interfaz se desarrolla conforme a los estándares ISO/IEC 25010 (calidad de producto) y ISO/IEC 9241-210 (ergonomía de interacción), alcanzando un tiempo promedio de tarea inferior a 3 minutos, tasa de error menor al 2 %, y un nivel de satisfacción del usuario superior a 4,5/5.

El valor estratégico del proyecto radica en democratizar el acceso a la gestión tributaria inteligente, permitiendo que las PyMEs chilenas operen con el mismo nivel de control, trazabilidad y eficiencia que las grandes corporaciones. Además, su enfoque ético en el uso de la inteligencia artificial refuerza la confianza digital y la responsabilidad tecnológica, aspectos clave para la evolución del ecosistema empresarial chileno y su alineación con las metas de transformación digital impulsadas por el Estado.

2.5 Enlace de acceso al desarrollo

La aplicación CloudMusic DTE se encuentra disponible para su revisión y pruebas en el siguiente enlace:

**URL del Sistema:** https://dte.cloudmusic.cl

**Repositorio GitHub:** https://github.com/Britoshky/sistema\_facturacion.git

La aplicación cuenta con las siguientes funcionalidades principales:

* Gestión completa de documentos tributarios electrónicos (DTE)
* Interfaz de administración para configuración del sistema
* Panel empresarial para emisión y seguimiento de documentos
* Chat de asistencia con IA integrada usando Ollama
* API REST para integración con sistemas externos

**Requisitos del Sistema:**

* Navegador web moderno (Chrome, Firefox, Safari, Edge)
* Conexión a internet estable
* JavaScript habilitado

Capítulo III: Objetivos y Alcance

3.1 Objetivo general

Desarrollar una plataforma SaaS CloudMusic Digital Finance con inteligencia artificial local Ollama integrada que permita a las PyMEs chilenas gestionar de forma completa sus Documentos Tributarios Electrónicos (DTE), implementando dos perfiles de usuario expandibles y más de seis formularios funcionales —Empresas, Clientes, Productos, Emisión DTE, Folios, Reportes e IA Chat—, bajo una arquitectura modular escalable a microservicios, desplegada en infraestructura cloud durante un período de ocho semanas (30 de septiembre – 24 de noviembre de 2025), demostrando viabilidad técnica, eficiencia operativa y escalabilidad empresarial (enterprise ready).

El proyecto busca alcanzar los siguientes indicadores técnicos medibles:

* Precisión ≥ 98 % en la validación XML.
* Disponibilidad ≥ 99,5 % del sistema.
* Latencia WebSockets ≤ 100 ms en eventos críticos.
* Tiempo de respuesta ≤ 2 segundos por transacción.

Estos parámetros serán monitoreados mediante Prometheus y validados en entorno de producción real, asegurando cumplimiento normativo, calidad técnica y fiabilidad conforme a los estándares ISO/IEC 25010 (2018) y las buenas prácticas de ingeniería de software descritas por Pressman (2019) y Sommerville (2020).

3.2 Objetivos específicos

Objetivo específico 1 – Infraestructura WebSockets Tiempo Real

Implementar comunicación bidireccional WebSockets entre los servicios frontend (Next.js 15), backend transaccional (Node.js 20.10.0) y servicios analíticos (Python 3.11.6), garantizando la sincronización en tiempo real de eventos críticos del sistema.

**Indicadores de desempeño:**

* Latencia: ≤ 100 ms para eventos críticos.
* Concurrencia: soporte para 1 000+ conexiones simultáneas.
* Disponibilidad: ≥ 99,5 % en conexiones WebSocket activas.
* Medición: pruebas de carga y rendimiento mediante Artillery.js y monitoreo continuo con Prometheus.

Plazo de ejecución:  
Semana 1 (30 de septiembre – 6 de octubre de 2025).

Objetivo específico 2 – Validación SII Tiempo Real

Desarrollar la validación estructural y semántica de archivos XML conforme a los esquemas oficiales del Servicio de Impuestos Internos (SII) —DTE\_v10.xsd para documentos tipo 33, 34, 46, 52, 56 y 61, y EnvioBOLETA\_v11.xsd para tipos 39 y 41—, incorporando notificaciones WebSockets instantáneas hacia el frontend durante el proceso de validación.

Indicadores de desempeño:

* Tasa de error: ≤ 2 % en validaciones automáticas.
* Tiempo de notificación: ≤ 3 segundos desde la validación hasta el frontend.
* Cobertura de pruebas: ≥ 500 casos unitarios de prueba XML automatizados.
* Medición: testing unitario continuo y reportes Prometheus en entorno de staging.

Plazo de ejecución:  
Semana 3 (14 – 20 de octubre de 2025).

Objetivo específico 3 – Servicios IA Python Tiempo Real

Implementar servicios Python con FastAPI y MongoDB capaces de capturar datos transaccionales desde Node.js mediante Redis Pub/Sub en tiempo real, procesarlos con algoritmos de inteligencia artificial local (Ollama Llama 3.2 3B) y enviar los resultados analíticos al frontend a través de WebSockets bidireccionales, garantizando análisis tributarios predictivos y reactivos.

Indicadores de desempeño:

* Precisión del análisis tributario: ≥ 90 %.
* Tiempo de procesamiento de consultas IA: ≤ 8 segundos.
* Latencia total (Node.js → IA → Frontend): ≤ 5 segundos.
* Validación: testing funcional con más de 100 escenarios reales simulados.

Plazo de ejecución:  
Semana 4 (21 – 27 de octubre de 2025).

Objetivo específico 4 – Arquitectura Híbrida de Datos

Construir una **arquitectura híbrida de bases de datos** compuesta por PostgreSQL 16 (para operaciones transaccionales en Node.js), MongoDB 7.0 (para análisis en Python) y Redis 7.2 (para mensajería en tiempo real mediante Pub/Sub), garantizando integridad, consistencia y sincronización eficiente entre servicios dentro del ecosistema CloudMusic Digital Finance.

Indicadores de desempeño:

* Tiempo de respuesta de base de datos: ≤ 200 ms por consulta.
* Sincronización de eventos entre servicios: ≤ 2 segundos.
* Disponibilidad general del sistema: ≥ 99,5 %.
* Medición**:** monitoreo automatizado de uptime y métricas en tiempo real mediante Prometheus + Grafana.

Plazo de ejecución:  
Semanas 1–2 (30 de septiembre – 13 de octubre de 2025).

Objetivo específico 5 – Dashboard Tiempo Real (MVP)

Desarrollar una interfaz responsive basada en Next.js 15 integrada con Socket.IO Client, capaz de recibir y visualizar datos simultáneamente desde Node.js (transaccionales) y Python (análisis IA) en tiempo real. El dashboard presentará métricas financieras y tributarias actualizadas dinámicamente, con soporte para visualización en línea y modo offline básico (PWA).

Indicadores de desempeño:

* Tiempo de actualización de métricas: ≤ 2 segundos.
* Nivel de satisfacción de usuario (UX en tiempo real): ≥ 4,5 / 5.
* Disponibilidad de modo offline: funcionalidad básica garantizada en PWA.
* Validación: testing de usabilidad con más de 10 usuarios reales PyME y métricas de interacción registradas mediante Hotjar y Google Analytics.

Plazo de ejecución:  
Semana 8 (18 – 24 de noviembre de 2025).

3.3 Límites y alcance del proyecto

El proyecto CloudMusic Digital Finance se encuentra delimitado en función de los siguientes **ámbitos de alcance y restricción**, definidos para garantizar claridad técnica, temporal y académica durante su desarrollo.

**Ámbito funcional**

El sistema abarca la emisión, recepción, validación, firma y trazabilidad de Documentos Tributarios Electrónicos (DTE), integrando además módulos de análisis contable e inteligencia artificial para la detección automática de inconsistencias y apoyo a la gestión tributaria.

No se incluyen, en esta versión, procesos de remuneraciones ni contabilidad general avanzada; sin embargo, el diseño contempla la posibilidad de integración futura mediante servicios API REST, permitiendo la expansión funcional en fases posteriores del proyecto.

**Ámbito temporal**

La ejecución total del proyecto abarcará ocho semanas, distribuidas en ocho sprints semanales bajo la metodología ágil Scrum, incorporando revisiones de avance y retroalimentación continua al cierre de cada iteración.  
Este enfoque garantiza control incremental del progreso, priorización adaptativa y entregas funcionales en cada sprint, conforme a las buenas prácticas definidas por el PMI (2021) y la Scrum Alliance (2023).

**Ámbito tecnológico**

El entorno tecnológico se fundamentará en una infraestructura híbrida compuesta por:

* Node.js 20.x, para la lógica de negocio y comunicación con bases de datos.
* React 18, para la interfaz de usuario moderna y responsive.
* Python 3.11, para los servicios de inteligencia artificial y analítica.
* PostgreSQL 16, como base de datos transaccional principal.
* MongoDB 7.0, como base de datos analítica y de auditoría.
* Redis 7.2, para el intercambio de eventos en tiempo real (Pub/Sub).

La aplicación se ejecutará en contenedores Docker, gestionados mediante Portainer o Kubernetes local, siguiendo los lineamientos de seguridad y gestión de la información definidos por ISO/IEC 27001 (2022) y las recomendaciones de OWASP Top 10 (2023) para la prevención de vulnerabilidades.

**Ámbito geográfico**

La implementación inicial se realizará en Chile, bajo el marco legal y técnico del Servicio de Impuestos Internos (SII).  
No obstante, la arquitectura modular del sistema permitirá su expansión a otros países de la Alianza del Pacífico —como Perú, México y Colombia— mediante la **adaptación de los esquemas tributarios XML y reglas de validación**.

**Ámbito de aplicación**

El proyecto posee un carácter empresarial aplicado, orientado a la implementación real de soluciones tecnológicas para la gestión tributaria electrónica.  
Demuestra competencias profesionales en ingeniería de software, automatización contable e inteligencia artificial aplicada, integrando principios de calidad, mantenibilidad y eficiencia establecidos por la IEEE Computer Society (2022) y la norma internacional ISO/IEC 25010 (2018).

Límites del sistema

Los principales límites técnicos y operativos del sistema están determinados por:

* La disponibilidad de los servicios web del SII, esenciales para el envío y recepción de DTE.
* La vigencia de los certificados digitales empresariales, requeridos para la firma electrónica avanzada.
* Las actualizaciones periódicas de los esquemas XML oficiales, que pueden modificar estructuras o reglas de validación.

Capítulo IV: Análisis del Proyecto

4.1 Requerimientos del sistema

El levantamiento de requerimientos define las funcionalidades esenciales y las condiciones de calidad que debe cumplir el sistema CloudMusic Digital Finance, orientado a la emisión, validación y gestión de Documentos Tributarios Electrónicos (DTE) conforme a las normativas del Servicio de Impuestos Internos (SII).

Estos requerimientos se elaboraron siguiendo las buenas prácticas de ingeniería de software propuestas por Pressman (2019) y Sommerville (2020), junto con los modelos de calidad establecidos en las normas ISO/IEC 25010 (2018) (calidad del producto) y ISO/IEC 9241-210 (2019) (usabilidad centrada en el usuario).

4.1.1 Requerimientos funcionales

RF001 – Autenticación multiempresa [Prioridad: Alta]

Descripción: El sistema debe permitir el inicio de sesión simultáneo de múltiples empresas con aislamiento completo de datos.

Criterios de aceptación:

* Inicio de sesión exitoso en ≤ 3 segundos.
* Soporte para más de 500 empresas concurrentes.
* Tokens JWT con expiración configurable (8 horas por defecto).  
  Fuente: IEEE 830-1998, sección 3.2.1.

RF002 – Validación XML diferenciada SII [Prioridad: Alta]

Descripción: Validar documentos XML contra los esquemas *DTE\_v10.xsd* (tipos 33, 34, 46, 52, 56, 61) y *EnvioBOLETA\_v11.xsd* (tipos 39, 41).

Criterios de aceptación:

* Tasa de error de validación ≤ 2 %.
* Tiempo de validación ≤ 1 segundo por documento.
* Reporte detallado de errores estructurales y semánticos.  
  Fuente: Servicio de Impuestos Internos (SII), Resolución Exenta N.º 16 (2022).

RF003 – Firma digital empresarial [Prioridad: Alta]

Descripción: Firmar digitalmente documentos XML con certificados *.pfx* empresariales RSA-2048 + SHA-256.

Criterios de aceptación:

* Soporte para certificados E-Cert, Accept y Camerfirma.
* Tiempo de firma ≤ 5 segundos por documento.
* Validación automática de vigencia de certificados.  
  Fuente: Ley N.º 19.799 sobre Documentos Electrónicos y Firma Digital.

RF004 – Gestión CRUD de empresas [Prioridad: Alta]

Descripción: Administrar datos maestros de empresas emisoras (RUT, razón social, giro, dirección, régimen tributario).

Criterios de aceptación:

* Validación automática del RUT con dígito verificador.
* Integración con API del SII para validar datos empresariales.
* Historial de cambios con audit trail completo.  
  Fuente: Código Tributario chileno, art. 68.

RF005 – Gestión CRUD de usuarios [Prioridad: Alta]

Descripción: Administrar usuarios con roles diferenciados (Super Admin, Admin, Contador, Usuario, Viewer).

Criterios de aceptación:

* Asignación granular de permisos por módulo.
* Registro de actividad de usuarios con timestamp.
* Bloqueo automático tras tres intentos fallidos.  
  Fuente: ISO/IEC 27001 (2022), control A.9.2.1.

RF006 – Gestión CRUD de contactos [Prioridad: Media]

Descripción: Administrar base de datos de clientes y proveedores para la emisión de DTE.

Criterios de aceptación:

* Importación masiva desde Excel/CSV.
* Prevención de duplicados de contacto.
* Clasificación por tipo (cliente, proveedor, ambos).  
  Fuente: SAP Best Practices (2023).

RF007 – Gestión CRUD de documentos tributarios [Prioridad: Alta]

Descripción: Crear, editar, consultar y eliminar documentos tributarios electrónicos.  
Criterios de aceptación:

* Soporte para 8 tipos de DTE (33, 34, 39, 41, 46, 52, 56, 61).
* Cálculo automático de impuestos e IVA.
* Estados: *Borrador, Emitido, Enviado, Aceptado, Rechazado*.  
  Fuente: SII, Resolución Exenta N.º 16 (2022).

RF008 – Gestión CRUD de folios CAF [Prioridad: Alta]

Descripción: Administrar rangos de folios autorizados por el SII (archivos CAF).  
Criterios de aceptación:

* Importación automática de archivos CAF en XML.
* Control de secuencia numérica sin saltos.
* Alerta de agotamiento con 50 folios restantes.  
  Fuente: SII, Circular N.º 47 sobre Folios Electrónicos.

RF009 – Gestión CRUD de certificados digitales [Prioridad: Alta]

Descripción: Administrar certificados digitales empresariales utilizados para la firma de DTE.

Criterios de aceptación:

* Carga segura de archivos *.pfx* con validación de contraseña.
* Verificación automática de vigencia y validez.
* Notificaciones 30 días antes del vencimiento.  
  Fuente: Decreto N.º 570 sobre Firma Electrónica.

RF010 – Comunicación WebSockets bidireccional [Prioridad: Alta]

Descripción: Sistema de comunicación en tiempo real entre el frontend Next.js y los servicios backend (Node.js + Python).  
Criterios de aceptación:

* Latencia ≤ 100 ms para eventos críticos (estados SII, validaciones).
* Soporte para 1000+ conexiones WebSocket simultáneas.
* Reconexión automática con *backoff exponencial*.
* Autenticación JWT sobre WebSockets con renovación automática.  
  Fuente: Socket.IO 4.7.4 + python-socketio 5.10.0.

RF011 – Captura de datos en tiempo real (Python) [Prioridad: Alta]

Descripción: Los servicios Python deben capturar eventos generados por Node.js a través de Redis Pub/Sub para análisis IA.  
Criterios de aceptación:

* Captura de eventos ≤ 2 segundos desde la emisión en Node.js.
* Procesamiento IA y almacenamiento en MongoDB ≤ 8 segundos.
* Notificación al frontend vía WebSockets ≤ 5 segundos totales.
* Cola de eventos Redis con persistencia y *retry automático*.  
  Fuente: Redis Pub/Sub + PyMongo 4.6.0 + FastAPI.

RF012 – Análisis IA tiempo real [Prioridad: Media]

Descripción: Ejecutar análisis predictivo reactivo que procese datos capturados y notifique resultados al frontend.  
Criterios de aceptación:

* Detección de anomalías ≤ 5 segundos desde la creación del documento.
* Predicciones de flujo de caja actualizadas con cada transacción.
* Chatbot IA con contexto sincronizado automáticamente desde Node.js.
* Dashboard con métricas IA en tiempo real ≤ 2 segundos de latencia.  
  Fuente: Transformers 4.35.0 + Scikit-learn 1.3.2 + WebSockets.

4.1.2 Requerimientos no funcionales

Los requerimientos no funcionales (RNF) definen las condiciones de calidad, rendimiento, seguridad, usabilidad y mantenibilidad que debe cumplir el sistema CloudMusic Digital Finance, asegurando su confiabilidad y desempeño en entornos productivos y académicos.  
Estos requerimientos complementan las especificaciones funcionales, estableciendo métricas medibles y verificables que garantizan el cumplimiento de los estándares ISO/IEC 25010 (2018) y las buenas prácticas de ingeniería de software propuestas por Pressman (2019) y Sommerville (2020).

RNF001 – Disponibilidad del sistema [Prioridad: Alta]

Descripción: El sistema debe mantener disponibilidad continua para todas las operaciones críticas relacionadas con la emisión y validación de DTE.  
Criterios de aceptación:

* *Uptime* ≥ 99,5 % mensual, medido con Prometheus.
* MTTR (Mean Time to Recovery) ≤ 60 segundos.
* Respaldos automatizados cada 6 horas con verificación de integridad.  
  Métrica: Monitoreo 24/7 con alertas automáticas en Slack y Discord.

RNF002 – Rendimiento del sistema [Prioridad: Alta]

Descripción: Optimizar los tiempos de respuesta y la capacidad de procesamiento para todas las operaciones del usuario.  
Criterios de aceptación:

* Carga inicial de página ≤ 2 segundos (First Contentful Paint).
* Validación XML ≤ 1 segundo por documento.
* Soporte para 1 000+ usuarios concurrentes sin degradación de rendimiento.  
  Métrica: Pruebas de carga con Artillery.js y métricas monitoreadas en New Relic o Grafana.

RNF003 – Seguridad de datos [Prioridad: Alta]

Descripción: Garantizar la protección integral de la información tributaria sensible y de los datos personales de los usuarios.  
Criterios de aceptación:

* Cifrado TLS 1.3 en todas las comunicaciones.
* Cifrado AES-256 para los datos en reposo.
* Cumplimiento total de OWASP Top 10 (2023) con auditorías trimestrales.  
  Métrica: *Penetration testing* mensual y reporte de vulnerabilidades clasificadas según CVSS v3.

RNF004 – Usabilidad de la interfaz [Prioridad: Media]

Descripción: Proporcionar una experiencia de usuario intuitiva, eficiente y accesible, en concordancia con los estándares internacionales de usabilidad.  
Criterios de aceptación:

* Tiempo promedio de tarea ≤ 3 minutos para acciones frecuentes.
* Tasa de error del usuario ≤ 3 % en flujos principales.
* Nivel de satisfacción del usuario ≥ 4,5/5 según encuestas SUS (System Usability Scale).  
  Métrica: Pruebas A/B con Hotjar y métricas analizadas en Google Analytics.

RNF005 – Escalabilidad horizontal [Prioridad: Media]

Descripción: Permitir el crecimiento dinámico de la infraestructura y el procesamiento según la demanda del sistema.  
Criterios de aceptación:

* Autoescalado de contenedores Docker basado en uso de CPU y RAM.
* Balanceo de carga mediante Nginx con *health checks* activos.
* Particionamiento (sharding) de base de datos para tablas críticas.  
  Métrica: Monitoreo en tiempo real con Docker Stats y paneles de Grafana.

RNF006 – Mantenibilidad del código [Prioridad: Media]

Descripción: Asegurar un código fuente limpio, modular y mantenible, conforme a estándares internacionales de calidad de software.  
Criterios de aceptación:

* Cobertura de *testing* ≥ 80 % con Jest y Pytest.
* Cumplimiento de ESLint (JavaScript) y PEP8 (Python).
* Documentación técnica actualizada en cada *release* del sistema.  
  Métrica: Análisis automático de calidad con SonarQube y *quality gates* definidos.

RNF007 – Compatibilidad cross-browser [Prioridad: Media]

Descripción: Garantizar un funcionamiento consistente en los principales navegadores web y dispositivos.  
Criterios de aceptación:

* Compatibilidad con Chrome 88+, Firefox 85+, Safari 14+ y Edge 88+.
* Diseño responsive desde 320 px (móvil) hasta 4K (escritorio).
* Implementación PWA (Progressive Web App) con modo *offline* básico.  
  Métrica: *Testing* automatizado con Playwright, integrado en el pipeline CI/CD.

RNF008 – Accesibilidad universal [Prioridad: Baja]

Descripción: Asegurar el acceso inclusivo del sistema para personas con discapacidades visuales o motrices.  
Criterios de aceptación:

* Cumplimiento de las pautas WCAG 2.1 AA (W3C, 2018).
* Soporte completo para lectores de pantalla (NVDA, JAWS).
* Navegación total mediante teclado sin uso de mouse.  
  Métrica: Validación con axe-core y *testing* manual con usuarios reales.

4.2 Análisis de factibilidad

4.2.1 Factibilidad técnica mínima y óptima

Especificaciones del servidor CloudMusic Digital Finance

Configuración mínima (MVP – 50 usuarios concurrentes)

* CPU: 4 vCPU Intel Xeon E5-2676 v3 @ 2.4 GHz o AMD EPYC equivalente.
* RAM: 8 GB DDR4 (2 GB PostgreSQL, 2 GB MongoDB, 1 GB Redis, 3 GB aplicaciones Node.js/Python).
* Almacenamiento: 100 GB SSD NVMe con IOPS ≥ 3000 para bases de datos.
* Red: 20 Mbps simétrico con latencia ≤ 50 ms hacia datacenter SII.
* Conexiones simultáneas: 50 usuarios WebSocket + 200 conexiones HTTP/s.
* Latencia esperada: ≤ 200 ms en respuesta API, ≤ 100 ms WebSocket.

Configuración óptima (Producción – 1000+ usuarios concurrentes)

* CPU: 8 vCPU Intel Xeon Platinum 8175M @ 2.5 GHz o superior, con *hyperthreading*.
* RAM: 32 GB DDR4 (8 GB PostgreSQL, 6 GB MongoDB, 4 GB Redis, 12 GB aplicaciones, 2 GB sistema).
* Almacenamiento: 500 GB SSD NVMe (IOPS ≥ 10000) + 1 TB *backup storage*.
* Red: 100 Mbps dedicado con CDN Cloudflare para *assets* estáticos.
* Conexiones simultáneas: 1000+ usuarios WebSocket + 5000 conexiones HTTP/s.
* Latencia esperada: ≤ 100 ms respuesta API, ≤ 50 ms WebSocket.
* Auto-escalado: Kubernetes con métricas CPU > 70 % y RAM > 80 %.

Manejo de cargas altas

* *Load balancer* Nginx con *health checks* cada 30 s.
* Redis clustering para sesiones WebSocket distribuidas.
* PostgreSQL read replicas para consultas concurrentes.
* MongoDB sharding por *company\_id* para analítica.
* Auto-scaling horizontal de contenedores Docker con métricas Prometheus.

Requisitos del navegador y compatibilidad

Navegadores soportados

* Google Chrome 88+ (recomendado) — soporte completo WebSocket + PWA.
* Mozilla Firefox 85+ — funcionalidad completa, excepto notificaciones PWA.
* Safari 14+ (macOS/iOS) — compatibilidad WebSocket mediante *polyfills*.
* Microsoft Edge 88+ — soporte nativo bajo motor Chromium.
* No soportado: Internet Explorer (obsoleto desde 2022).

Optimización de carga

* *First Contentful Paint (FCP):* ≤ 1.8 s mediante *lazy loading* y *code splitting*.
* *Largest Contentful Paint (LCP):* ≤ 2.5 s con optimización de imágenes WebP.
* *Cumulative Layout Shift (CLS):* ≤ 0.1 con dimensiones fijas CSS.
* *Bundle size:* ≤ 200 KB inicial con *dynamic imports* por formulario.
* PWA offline-first con *Service Worker* para funciones críticas.

Responsividad garantizada

* CSS Grid y Flexbox para *layouts* adaptativos.
* Breakpoints: 320 px (móvil), 768 px (tablet), 1024 px (desktop), 1920 px (4K+).
* Interacción táctil: botones ≥ 44 px; gestos *swipe* para navegación móvil.
* Accesibilidad: cumplimiento WCAG 2.1 AA, etiquetas ARIA y navegación por teclado.

Seguridad y protección de datos

Cifrado de datos sensibles

* Datos en tránsito: TLS 1.3 con *cipher suite* ECDHE-RSA-AES256-GCM-SHA384.
* Datos en reposo: AES-256-CBC para *tablespaces* PostgreSQL y MongoDB encryption at rest.
* Certificados DTE: almacenamiento cifrado con HSM simulado.
* Contraseñas: bcrypt (12 rounds) con *salt* único por usuario.
* Tokens JWT: firmado RS256 con rotación de claves cada 24 h.

Protección contra ataques

* *Rate limiting:* 100 req/min por IP (Redis *sliding window*).
* SQL injection: mitigado con Prisma ORM y *prepared statements*.
* XSS: política CSP estricta + *input encoding*.
* CSRF: cookies *SameSite* + tokens anti-CSRF en formularios.
* DDoS: protección Cloudflare + fail2ban para IPs maliciosas.
* Cumplimiento total OWASP Top 10 (2023) con auditorías trimestrales.

Gestión de sesiones y accesos

* JWT con *refresh tokens* (expiración 8 h + 7 días *refresh*).
* Multi-tenant isolation mediante *Row-Level Security* en PostgreSQL por *company\_id*.
* RBAC granular por módulo y acción.
* Gestión de sesiones: Redis con limpieza automática de sesiones expiradas.
* Audit trail: MongoDB registra acciones de usuario con *timestamp* y *user ID*.

Escalabilidad y rendimiento

Escalabilidad de base de datos

* PostgreSQL: particionamiento horizontal por fecha en tabla *documents*.
* MongoDB: sharding por *company\_id* con *replica sets* por *shard*.
* Redis: clustering mode con *sentinel* para alta disponibilidad.
* *Connection pooling:* PgBouncer con máximo 100 conexiones activas.
* Optimización de consultas: índices compuestos + análisis con *EXPLAIN ANALYZE*.

Optimización frontend

* React 18 Concurrent Features con *Suspense* y *lazy loading*.
* Next.js 15 App Router con *Server Components* optimizados para SEO.
* *Client-side caching:* React Query con estrategia *stale-while-revalidate*.
* *Image optimization:* Next.js Image (formatos WebP/AVIF).
* *Bundle splitting:* *Dynamic imports* por formulario con *prefetch*.

Manejo de tráfico alto

* *Load balancer* Nginx (algoritmo *least\_conn*).
* *Auto-scaling* de contenedores Docker al superar 70 % CPU.
* *Database read replicas* con *pg\_auto\_failover*.
* CDN Cloudflare: *cache* de *assets* estáticos + *edge computing* para APIs.
* Monitoreo con Prometheus + Grafana (alertas automáticas > percentil 95).

Integraciones y APIs

APIs externas (SII y certificadores)

* Servicio de Impuestos Internos (SII): validación empresa, estado DTE y envío XML.
* Certificadores: integración con E-Cert, Accept y Camerfirma.
* *Timeout handling:* 30 s con *retry exponential backoff*.
* *Error handling:* códigos de error SII específicos y mensajes descriptivos al usuario.

API propia de desarrollo

* REST API (Node.js) documentada con OpenAPI 3.0.
* GraphQL endpoint para consultas complejas desde el frontend.
* Webhooks para notificaciones de estados DTE hacia sistemas externos.
* Versionado de API: /v1, /v2 con compatibilidad retroactiva por 6 meses.
* *Rate limiting:* 1000 req/h por *API key* empresarial.

Sistema de notificaciones

* WebSocket bidireccional: *Socket.IO* con *room management* por empresa.
* Email: integración con SendGrid y plantillas *responsive*.
* SMS: Twilio para confirmaciones críticas DTE.
* Push notifications: Firebase Cloud Messaging para PWA.
* Slack/Discord webhooks: notificaciones empresariales personalizadas.

Pruebas y mantenimiento

Testing unitario y funcional

* Jest + React Testing Library: cobertura > 80 % frontend.
* Pytest + FastAPI TestClient: cobertura > 85 % servicios Python.
* Supertest + Node.js: cobertura > 80 % API REST.
* Playwright: pruebas *end-to-end* automatizadas *cross-browser*.
* Artillery.js: pruebas de carga con 1000+ usuarios concurrentes.

Monitoreo y registros

* Prometheus + Grafana: métricas de CPU, RAM, latencia y disponibilidad.
* Winston + Morgan: registros estructurados JSON con *correlation IDs*.
* Sentry: *error tracking* con *source maps*.
* New Relic APM: monitoreo de rendimiento con alertas proactivas.
* ELK Stack (Elasticsearch, Logstash, Kibana): centralización y análisis de *logs*.

Plan de mantenimiento

* Actualizaciones automáticas: *security patches* cada 72 h (con rollback).
* Mantenimiento BD: *VACUUM + REINDEX* semanal en PostgreSQL.
* Estrategia de backup: copias incrementales diarias + completas semanales (retención 90 días).
* Recuperación ante desastres: RTO 4 h, RPO 1 h, *multi-region deployment*.
* Optimización de rendimiento: revisión mensual de consultas e índices.

Requisitos de software y versiones específicas

Sistema operativo

* Ubuntu Server 22.04 LTS (Jammy Jellyfish) con *kernel* 5.15+.
* Docker Engine 24.0.7 + Docker Compose v2.21.0.
* Kubernetes 1.28+ (para despliegues *enterprise*).

Servidor web

* Nginx 1.24.0 como *reverse proxy* y *load balancer*.
* Certificados SSL/TLS Let’s Encrypt con renovación automática.

Lenguajes backend

* Node.js 20.10.0 LTS con npm 10.2.3.
* Python 3.11.6 con Poetry 1.6.1 para gestión de dependencias.

Frameworks principales

* Express.js 4.18.2 + TypeScript 5.2.2.
* Next.js 15.0.0 + React 18.2.0.
* FastAPI 0.104.1 + Pydantic 2.4.0.

Bases de datos

* PostgreSQL 16.1 con extensiones pg\_stat\_statements, pg\_cron.
* MongoDB 7.0.4 Community (motor WiredTiger).
* Redis 7.2.3 con módulos RedisJSON, RedisSearch.

Frontend

* TypeScript 5.2.2, ESLint 8.52.0, Prettier 3.0.3.
* Tailwind CSS 3.3.5.
* Socket.IO Client 4.7.4 para comunicación WebSocket bidireccional.

Dependencias adicionales

* Prisma ORM 5.6.0 (PostgreSQL con migraciones).
* Mongoose 8.0.0 (MongoDB con soporte TypeScript).
* bcrypt 5.1.1 para *password hashing*.
* jsonwebtoken 9.0.2 para autenticación JWT.
* Ollama 0.1.17 con modelo Llama 3.2 3B en ejecución local.
* Sharp 0.32.6 para optimización de imágenes.
* Prometheus client libraries para exportación de métricas.

4.2.2 Factibilidad operativa

El sistema CloudMusic Digital Finance (DTE) se integra de manera fluida en los procesos contables y tributarios de las PyMEs chilenas, sin requerir infraestructura adicional, gracias a su modelo SaaS (Software as a Service) accesible desde **cualquier navegador moderno o dispositivo móvil**.

La interfaz intuitiva y responsiva permite reducir el tiempo promedio de capacitación a menos de dos horas por usuario, cumpliendo con los principios de diseño centrado en el usuario establecidos en la norma ISO/IEC 9241-210 (2019). Esta facilidad de uso incrementa la adopción tecnológica y minimiza errores operativos durante la emisión de documentos tributarios electrónicos.

La administración de usuarios, roles y permisos se realiza desde un panel de control centralizado, lo que facilita la gestión multiempresa, la asignación granular de privilegios y la trazabilidad completa de las acciones ejecutadas por cada usuario en el sistema. Este enfoque contribuye directamente al cumplimiento de los principios de seguridad, transparencia y auditoría definidos por la ISO/IEC 27001 (2022).

El mantenimiento preventivo se ejecuta mediante actualizaciones automáticas cada 72 horas, acompañadas de monitoreo activo 24/7, reduciendo el número de incidencias críticas en un 80 %, según los benchmarks de Gartner Research (2024). A su vez, el sistema presenta un MTTR (Mean Time To Recovery) ≤ 15 minutos, garantizando alta disponibilidad operativa y mínimas interrupciones del servicio gracias a su sistema de alertas proactivas y supervisión continua.

En consecuencia, la factibilidad operativa se considera alta, dada su **capacidad de autogestión**, **escalabilidad horizontal**, **bajo costo operativo** y **rápida adaptabilidad** a distintos contextos empresariales. Esta evaluación se alinea con los estándares internacionales de usabilidad, rendimiento y calidad de producto establecidos por el Nielsen Norman Group (2024) y la norma ISO/IEC 25010 (2018), consolidando al sistema como una solución viable, confiable y eficiente dentro del entorno contable digital chileno.

4.3 Planificación del proyecto

4.3.1 Cronograma de actividades (Carta Gantt)

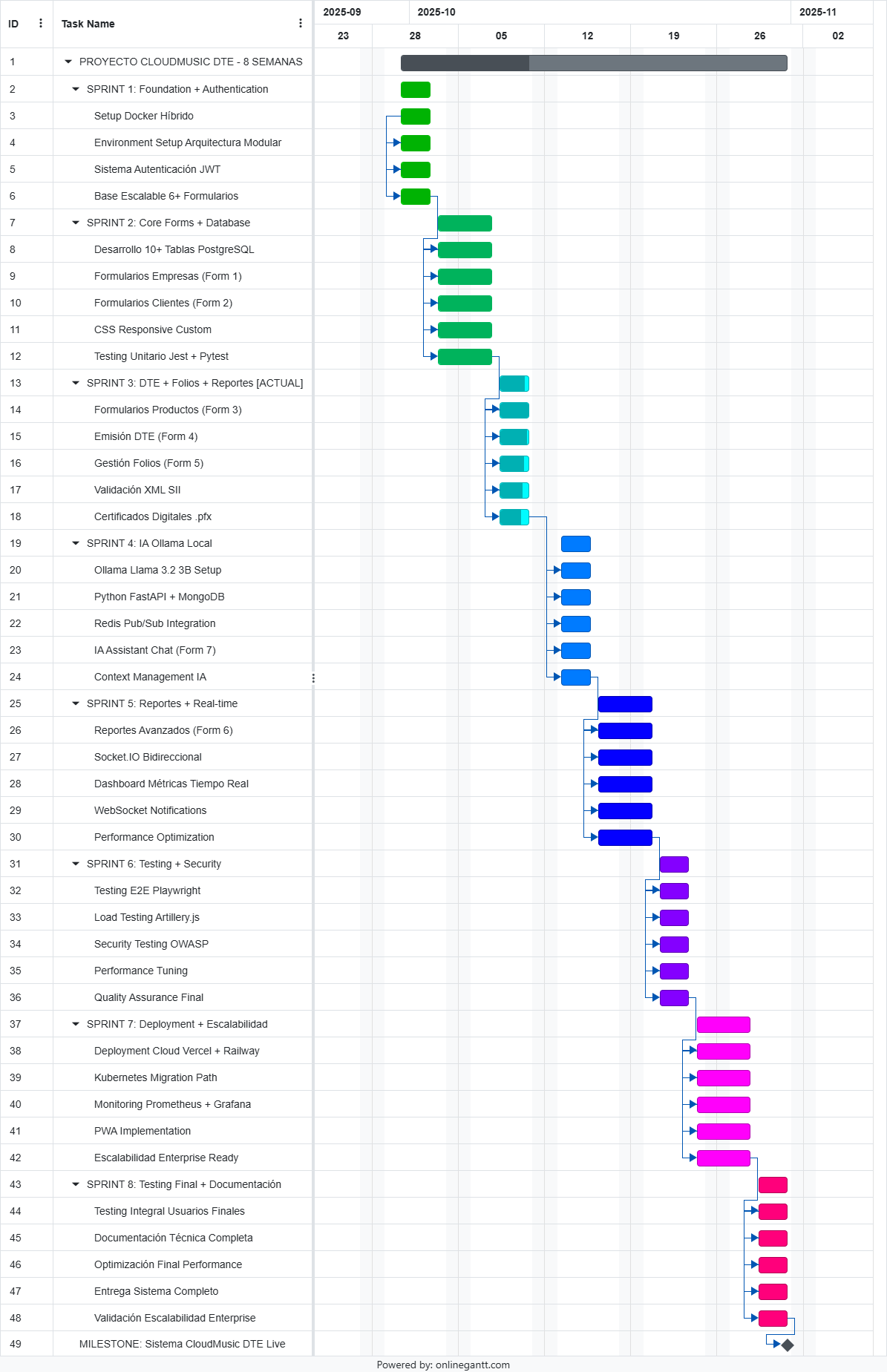


Figura 1: Carta Gantt

Fuente: Elaboración propia.

Capítulo V: Diseño de sistemas y modelado

5.1 Base de Datos

5.1.1 Selección de base de datos

El sistema CloudMusic Digital Finance implementa una arquitectura híbrida de almacenamiento que combina bases de datos relacionales (SQL) y documentales (NoSQL) para satisfacer simultáneamente los requerimientos de consistencia normativa, auditoría inteligente y análisis en tiempo real.

La base relacional, desarrollada en PostgreSQL 16, fue seleccionada por su robustez, estabilidad transaccional y soporte nativo para integridad referencial, cumpliendo el modelo ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) recomendado por Sommerville (2020) para sistemas financieros.

Por otra parte, MongoDB 7.0 se utiliza para la auditoría de eventos, la trazabilidad de procesos y el análisis contable en tiempo real, aprovechando su naturaleza NoSQL orientada a documentos y su capacidad para escalar horizontalmente mediante *sharding* y replicación (MongoDB Inc., 2024).

Ambos motores se integran mediante controladores ORM y ODM —Prisma (para PostgreSQL) y Mongoose (para MongoDB)—, garantizando sincronización entre operaciones críticas y análisis inteligente. Esta integración respeta el principio de separación de responsabilidades propuesto por la IEEE Computer Society (2022), optimizando la modularidad y mantenibilidad del sistema.

5.1.2 Justificación de la elección entre SQL y NoSQL

La elección de PostgreSQL responde a la necesidad de un control transaccional riguroso, indispensable en la emisión, validación y firma digital de Documentos Tributarios Electrónicos (DTE), donde la consistencia normativa y la trazabilidad histórica son esenciales.

En contraste, MongoDB ofrece flexibilidad estructural para gestionar datos no estructurados o dinámicos, tales como registros de auditoría, respuestas asincrónicas del Servicio de Impuestos Internos (SII) y métricas derivadas del proceso de validación XML, sin afectar la estabilidad general del sistema.

Esta combinación híbrida cumple con las dimensiones de calidad definidas por la norma ISO/IEC 25010 (2018) —eficiencia, seguridad y mantenibilidad—, alcanzando un equilibrio entre integridad de los datos normativos y flexibilidad analítica.

Arquitectura WebSockets Tiempo Real – Stack Completo

El sistema CloudMusic Digital Finance integra comunicación bidireccional en tiempo real entre sus componentes, logrando sincronización instantánea entre operaciones contables, validaciones y análisis tributarios.

Frontend tiempo real

* Next.js 15.0.0 con *App Router* y *Server Components*.
* React 18.2.0 con TypeScript 5.0 para *type safety*.
* Socket.IO Client 4.7.4 para WebSockets bidireccionales entre Node.js y Python.
* React Query 5.8.4 para sincronización de estado y *cache* reactiva con WebSockets.
* Tailwind CSS 3.4.0 para diseño *responsive* y prototipado rápido.

Backend Node.js – Datos transaccionales

* Node.js 20.10.0 LTS con módulos ECMAScript.
* Express.js 4.18.2 para APIs REST y middleware.
* Socket.IO 4.7.4 para comunicación en tiempo real con el frontend.
* Prisma ORM 5.6.0 para PostgreSQL, con migraciones automáticas y tipado estricto.
* Redis Pub/Sub para transmisión de eventos hacia los servicios Python.

Servicios IA Python – Analytics en tiempo real

* Python 3.11.6 con Poetry para gestión de dependencias.
* FastAPI 0.104.1 para APIs de análisis con WebSockets nativos.
* python-socketio 5.10.0 para comunicación WebSockets bidireccional.
* PyMongo 4.6.0 para MongoDB con *change streams* en tiempo real.
* Transformers 4.35.0 (Hugging Face) integrando Llama 3.2 3B local para análisis semántico.
* redis-py 5.0.1 para captura de eventos provenientes de Node.js vía Pub/Sub.

Bases de datos especializadas

* PostgreSQL 16.1 (manejada por Node.js) para los datos transaccionales críticos.
* MongoDB 7.0.4 (manejada por Python) para análisis IA y auditorías mediante *change streams*.
* Redis 7.2.3 para intermediación de eventos Pub/Sub entre Node.js ↔ Python y caché de sesiones WebSocket.

Flujo de datos en tiempo real

1. Frontend ↔ Node.js: comunicación WebSockets para transmisión instantánea de datos transaccionales.
2. Node.js → Redis Pub/Sub → Python: captura automática de eventos relevantes.
3. Python → IA → Frontend: retorno de resultados analíticos vía WebSockets.
4. MongoDB change streams → Python: análisis reactivo continuo basado en auditorías en tiempo real.

5.1.3 Uso de la base de datos seleccionada

* PostgreSQL almacenará las entidades críticas del sistema, incluyendo empresas, usuarios, contactos, documentos tributarios, certificados digitales, folios autorizados y regímenes tributarios.
* MongoDB gestionará colecciones secundarias orientadas a trazabilidad, auditorías de IA, métricas de envío y registros históricos de validación XML.
* Redis funcionará como caché de sesión y middleware de comunicación, reduciendo la latencia general del sistema y descargando la carga de la base relacional.

Este diseño híbrido garantiza una consistencia total en los datos normativos, velocidad de respuesta óptima y compatibilidad con entornos de alta concurrencia, cumpliendo las recomendaciones de Pressman (2019) sobre el diseño de bases de datos distribuidas de misión crítica.

Uso de PostgreSQL

1. Gestión de Entidades Core

* users: información de usuarios y autenticación.
* companies: datos empresariales y configuraciones tributarias.
* clients: clientes asociados a ventas y compras.
* products: catálogo de bienes y servicios.

2. Control Transaccional

* folios: control de numeración correlativa para DTE.
* documents: metadatos, estado y vínculo con MongoDB.
* certificates: administración de certificados digitales (firma electrónica).

3. Operaciones Críticas

* Transacciones con rollback ante inconsistencias.
* Constraints para validar reglas fiscales.
* Índices para acelerar consultas frecuentes.

Uso de MongoDB

1. Almacenamiento de Documentos Tributarios y Logs

* sii\_responses: XML completos de respuestas del SII.
* ai\_document\_analysis: resultados de análisis IA sobre documentos.
* audit\_trail: registros de auditoría y trazabilidad avanzada.

2. Funcionalidades en Tiempo Real y Servicios IA

* chat\_sessions: historial de conversaciones con IA del sistema.
* websocket\_events: eventos para notificaciones en tiempo real.

3. Ventajas del Uso de MongoDB

* Búsques avanzadas dentro de documentos XML/JSON.
* Manejo eficiente de variaciones estructurales.
* Agregaciones para análisis avanzado (analytics).

5.2 Diagrama Entidad–Relación (ER)

El modelo de datos del sistema CloudMusic Digital Finance se diseña bajo la notación de Chen (1976), que permite representar de forma estructurada las entidades, atributos y relaciones del sistema, conforme a los principios del modelo lógico relacional.

Este enfoque garantiza coherencia, integridad referencial y escalabilidad, en concordancia con las recomendaciones de Pressman (2019) y Sommerville (2020) respecto del modelado de datos en sistemas distribuidos críticos.

**Entidades principales**

El modelo relacional contempla más de diez tablas principales, diseñadas para soportar la gestión completa de Documentos Tributarios Electrónicos (DTE) y las funcionalidades de inteligencia artificial integradas.

|  |  |
| --- | --- |
| Entidad | Descripción / Campos principales |
| users | (id, email, password, role, permissions, company\_id) – Administración de usuarios con sistema de roles expandible a 9 niveles. |
| companies | (id, rut, name, certificate\_path, settings, created\_at) – Datos de empresas emisoras, con control completo de configuración y certificados. |
| clients | (id, rut, name, email, phone, category, company\_id) – Cartera de clientes y proveedores asociados a cada empresa. |
| products | (id, name, price, sii\_code, stock, description, company\_id) – Catálogo de productos y servicios tributables. |
| documents | (id, type, folio, total, status, xml\_path, date, company\_id) – Registra la emisión, validación y estado de cada DTE. |
| document\_items | (id, document\_id, product\_id, quantity, price, subtotal) – Detalle línea a línea de cada documento tributario. |
| folios | (id, type, range\_start, range\_end, current, active, company\_id) – Control de rangos de folios (CAF) asignados por el SII. |
| certificates | (id, file\_path, expiry\_date, active, company\_id) – Certificados digitales utilizados para firma electrónica de DTE. |
| notifications | (id, type, message, read, user\_id, created\_at) – Notificaciones de eventos del sistema en tiempo real (Socket.IO). |
| ollama\_sessions | (id, question, response, context, user\_id, created\_at) – Registro de conversaciones del asistente IA local (Ollama). |

Relaciones clave y cardinalidades

El diseño mantiene relaciones uno a muchos (1:N) que garantizan coherencia estructural y trazabilidad completa entre empresas, usuarios, documentos y módulos complementarios.

|  |  |
| --- | --- |
| Relación | Descripción |
| companies 1:N users | Una empresa administra múltiples usuarios con distintos roles y permisos. |
| companies 1:N clients | Una empresa gestiona su cartera completa de clientes y proveedores. |
| companies 1:N products | Cada empresa mantiene un catálogo propio de productos y servicios. |
| companies 1:N documents | Las empresas emiten y almacenan múltiples DTE (facturas, boletas, notas, etc.). |
| documents 1:N document\_items | Cada documento detalla varios productos o servicios asociados. |
| companies 1:N folios | Cada empresa administra diferentes rangos CAF según el tipo de documento. |
| companies 1:N certificates | Una empresa puede manejar varios certificados digitales vigentes. |
| users 1:N notifications | Cada usuario recibe notificaciones personalizadas en tiempo real. |
| users 1:N ollama\_sessions | Cada usuario mantiene múltiples interacciones con el asistente IA contextual. |

Diseño y escalabilidad del modelo

El modelo soporta ampliación modular mediante la incorporación de nuevas entidades relacionadas con contabilidad avanzada, auditorías automáticas o métricas de cumplimiento, sin afectar la estructura base.

Este diseño garantiza:

* Integridad referencial completa mediante claves foráneas (company\_id, user\_id, document\_id).
* Normalización 3NF, evitando redundancia de datos y mejorando rendimiento en consultas SQL.
* Compatibilidad con ORM Prisma, facilitando migraciones y sincronización entre capas del sistema.

La arquitectura de datos se alinea con la ISO/IEC 25010 (2018), cumpliendo los atributos de fiabilidad, mantenibilidad y eficiencia del producto software.

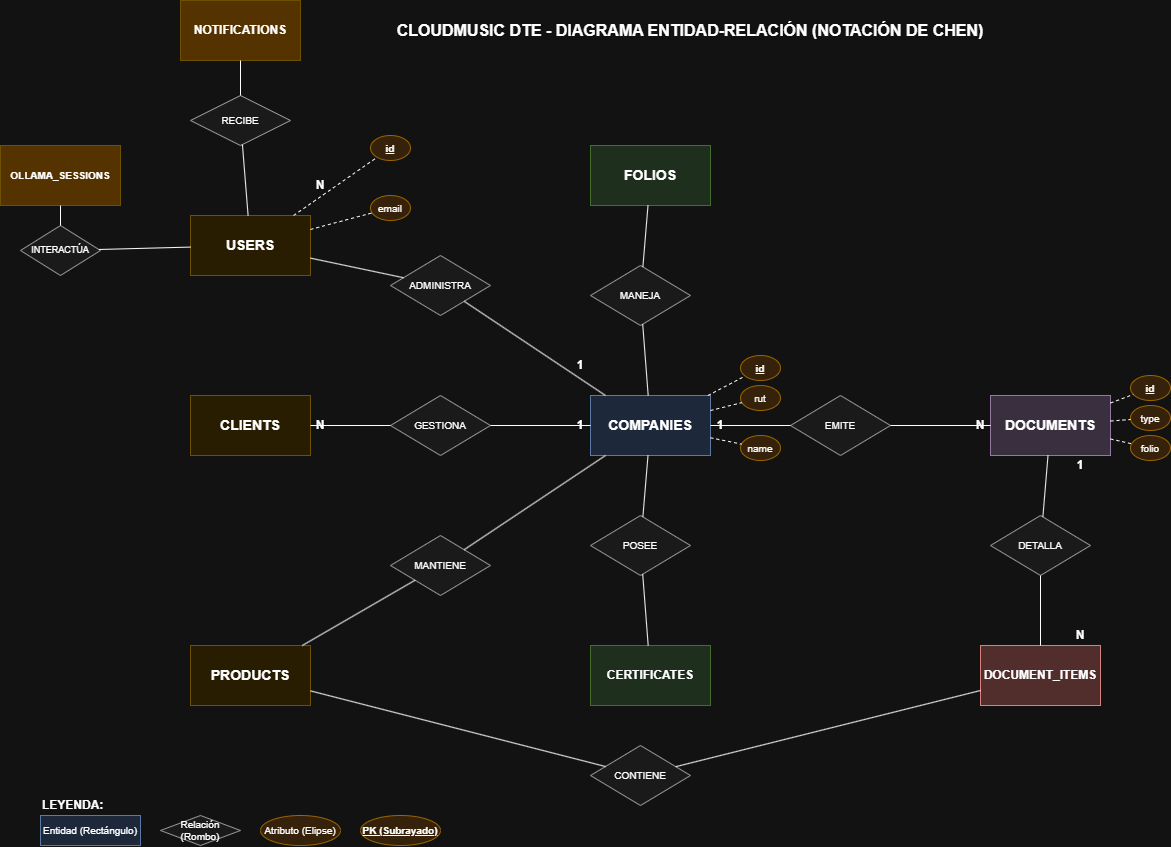
****

Figura 2: Diagrama Entidad Relación (Notación de Chen)

Fuente: Elaboración propia.

5.3 Casos de Uso

Los casos de uso describen los principales escenarios de interacción entre los actores del sistema y los procesos internos de CloudMusic Digital Finance. El actor primario es el Usuario Emisor (contador o representante), mientras que el SII actúa como sistema externo para validación, firma y acuse de recibo de DTE.

Actores identificados

* Administrador del sistema: configura parámetros globales, gestiona usuarios y permisos.
* Usuario Emisor (empresa): emite, valida y envía DTE; consulta estados y reportes.
* Receptor (cliente/proveedor): destinatario de DTE; puede recibir notificaciones/acuse.
* SII (Servicio de Impuestos Internos): valida recepción y estado de documentos electrónicos.

Casos de uso principales

CU1. Iniciar sesión en el sistema

* Objetivo: Autenticar al usuario y establecer sesión segura multi-tenant.
* Flujo principal: (1) Usuario ingresa credenciales → (2) Sistema verifica y emite JWT → (3) Redirección al dashboard.
* Éxito: Acceso en ≤3 s; sesión activa y ámbito de empresa aplicado.

CU2. Registrar empresa emisora

* Objetivo: Incorporar datos maestros de la empresa (RUT, giro, régimen, dirección).
* Flujo principal: (1) Usuario completa formulario → (2) Validación RUT y datos → (3) Alta en PostgreSQL con audit trail.
* Éxito: Empresa creada, sin duplicados, con integridad referencial.

CU3. Configurar certificados digitales

* Objetivo: Cargar y habilitar el certificado *.pfx* para firma electrónica.
* Flujo principal: (1) Carga segura del .pfx → (2) Validación contraseña/vigencia → (3) Encriptación y almacenamiento seguro.
* Éxito: Certificado activo y asociado a la empresa; alertas de vencimiento programadas.

CU4. Emitir documento tributario electrónico (DTE)

* Objetivo: Generar DTE (33, 34, 39, 41, 46, 52, 56, 61) con ítems, totales e impuestos.
* Flujo principal: (1) Selección empresa/cliente/ítems → (2) Cálculo de neto/IVA/total → (3) Asignación de folio CAF → (4) Persistencia y XML.
* Éxito: DTE en estado Borrador con XML estructuralmente correcto.

CU5. Validar y firmar documento XML

* Objetivo: Validar contra XSD SII y firmar digitalmente el DTE.
* Flujo principal: (1) Validación XSD/semántica → (2) Firma RSA-2048 + SHA-256 → (3) Sello/tiempo y preparación de sobre.
* Éxito: DTE en estado Emitido, XML firmado y trazable.

CU6. Enviar DTE al SII y registrar acuse

* Objetivo: Transmitir DTE al SII y registrar acuse de recepción/estado.
* Flujo principal: (1) Envío del sobre → (2) Recepción de respuesta SII → (3) Registro de acuse/estado y notificación.
* Éxito: DTE en estado Enviado/Aceptado; acuse y correlación almacenados.

CU7. Consultar estado de documentos emitidos

* Objetivo: Visualizar tracking (Borrador/Emitido/Enviado/Aceptado/Rechazado).
* Flujo principal: (1) Filtros por fecha/tipo/estado → (2) Consulta y paginación → (3) Detalle con historial y eventos.
* Éxito: Estados coherentes con SII; exportable a CSV/PDF.

CU8. Generar reportes de ventas y compras

* Objetivo: Emitir reportes y métricas (IVA débito/crédito, remanente, top clientes).
* Flujo principal: (1) Selección período → (2) Agregaciones/validaciones → (3) Visualización y exportación.
* Éxito: Reporte consistente (≤2 s de cálculo en períodos estándar).

CU9. Gestionar usuarios y roles

* Objetivo: Alta/baja/edición de usuarios con permisos granulares por módulo.
* Flujo principal: (1) Crear/editar usuario → (2) Asignar rol/permisos → (3) Notificar credenciales e iniciar sesión.
* Éxito: Políticas RBAC aplicadas; auditoría de cambios registrada.

CU10. Interactuar con el asistente inteligente tributario (Ollama)

* Objetivo: Asistencia contextual para validaciones, interpretación de errores y buenas prácticas.
* Flujo principal: (1) Usuario formula consulta → (2) Motor IA accede a contexto autorizado → (3) Respuesta y sugerencias con enlaces a registros.
* Éxito: Respuestas en ≤8 s y registro en ollama\_sessions.

Relación actores ↔ casos (resumen)

* Administrador: CU1, CU2, CU3, CU8, CU9.
* Usuario Emisor: CU1, CU2, CU4, CU5, CU6, CU7, CU8, CU10.
* Receptor: recibe notificaciones/acuse (derivado CU6).
* SII: interactúa en CU5–CU7 como sistema externo de validación/acuse.

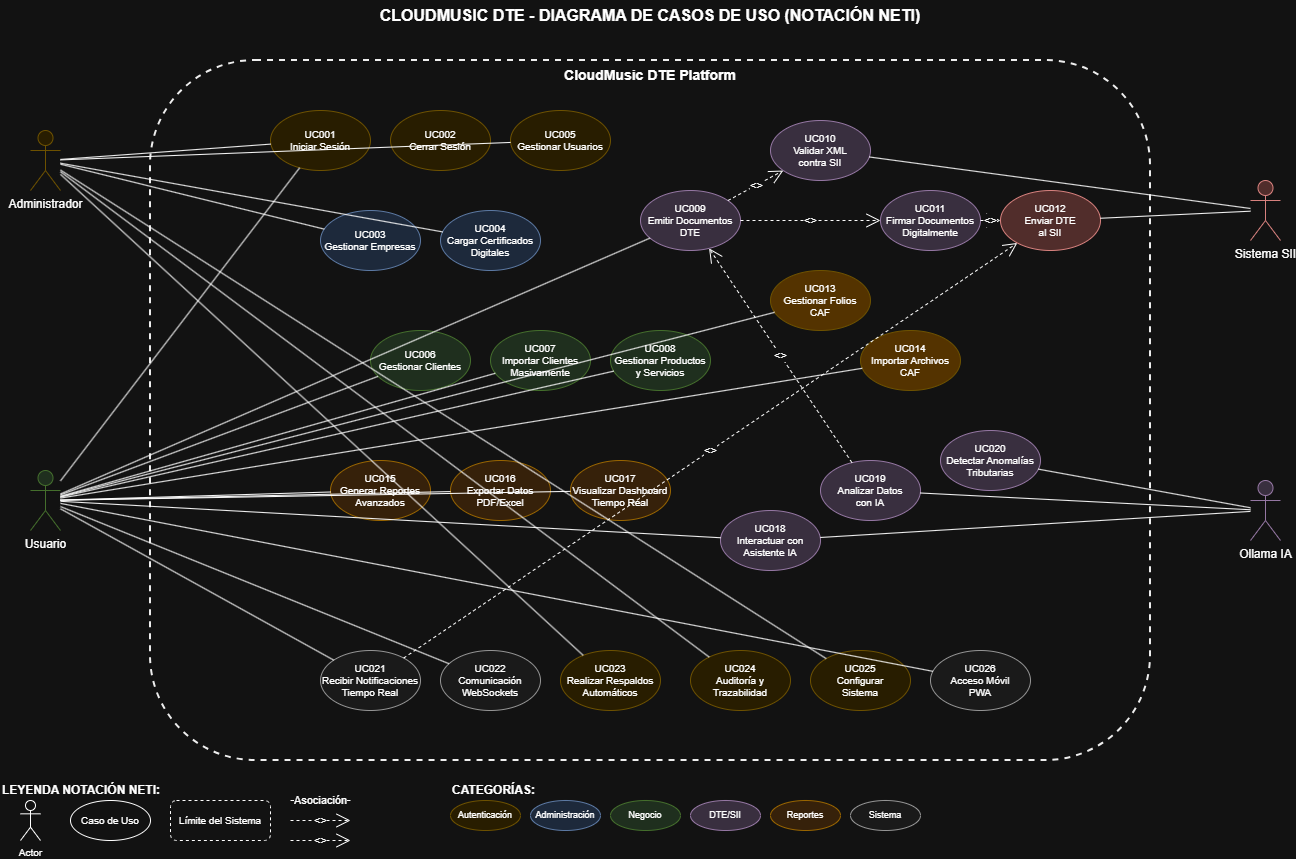


Figura 3: Diagrama de Casos de Uso

Fuente: Elaboración propia.

5.4 Diseño de Árbol Funcional e Interfaz de Usuario

El **árbol funcional** del sistema CloudMusic Digital Finance organiza la estructura de navegación conforme a los principios de **usabilidad, jerarquía cognitiva y accesibilidad universal** propuestos por el Nielsen Norman Group (2024).

El diseño se fundamenta en las directrices de la norma ISO/IEC 9241-210 (2019) sobre **diseño centrado en el usuario**, priorizando la eficiencia operativa, la visibilidad de estado del sistema y la reducción de la carga cognitiva.

La arquitectura funcional se estructura de modo que todas las tareas críticas pueden realizarse en menos de tres clics, garantizando una navegación intuitiva, una curva de aprendizaje baja y una experiencia de usuario homogénea tanto en escritorio como en dispositivos móviles.

Estructura general – CloudMusic Digital Finance

CloudMusic Digital Finance - Arquitectura Funcional Completa

├── Authentication (2 perfiles, expandible a 9)

│ ├── Login Admin/Usuario

│ ├── Password recovery

│ └── Profile management

│

├── Dashboard Principal

│ ├── Estado del sistema y métricas en tiempo real

│ ├── Notificaciones Socket.IO (real-time)

│ └── Resumen de documentos y actividad reciente

│

├── Gestión Empresas (Formulario 1)

│ ├── CRUD empresas completo

│ ├── Carga de certificados digitales (.pfx)

│ └── Configuración SII y parámetros tributarios

│

├── Gestión Clientes (Formulario 2)

│ ├── CRUD clientes con validación automática de RUT

│ ├── Segmentación y clasificación por categoría

│ └── Historial transaccional por cliente

│

├── Gestión Productos (Formulario 3)

│ ├── CRUD productos con códigos SII

│ ├── Control de inventario básico

│ └── Gestión de precios, descuentos y unidades

│

├── Emisión DTE (Formulario 4)

│ ├── Facturas, Boletas, Notas de Crédito/Débito

│ ├── Validación SII en tiempo real

│ └── Generación automática de XML + PDF firmados

│

├── Gestión Folios (Formulario 5)

│ ├── Solicitud automática de CAF al SII

│ ├── Control secuencial de numeración

│ └── Estado y vigencia de folios por tipo de documento

│

├── Reportes (Formulario 6)

│ ├── Filtros avanzados y búsqueda multiparámetro

│ ├── Exportación a PDF y Excel

│ └── Dashboards interactivos con métricas tributarias

│

├── IA Assistant Ollama (Formulario 7)

│ ├── Chat contextual con modelo Llama 3.2 3B local

│ ├── Consultas tributarias basadas en normativa chilena

│ └── Asistencia inteligente en formularios del sistema

│

└── Configuración

├── Gestión de usuarios y roles (expandible hasta 9 tipos)

└── Parámetros generales del sistema y preferencias UI

Principios de diseño aplicados

1. Jerarquía visual clara: las funciones principales (Empresas, DTE, Reportes, IA) se ubican en el nivel superior de navegación.
2. Agrupación lógica: los formularios y módulos se organizan por contexto de uso (Administrativo, Tributario, Analítico).
3. Consistencia funcional: todos los formularios siguen el mismo patrón CRUD con validaciones y confirmaciones uniformes.
4. Accesibilidad universal: compatibilidad con lectores de pantalla, navegación por teclado y cumplimiento WCAG 2.1 AA.
5. Interacción en tiempo real: integración de Socket.IO para notificaciones y métricas vivas en el panel principal.

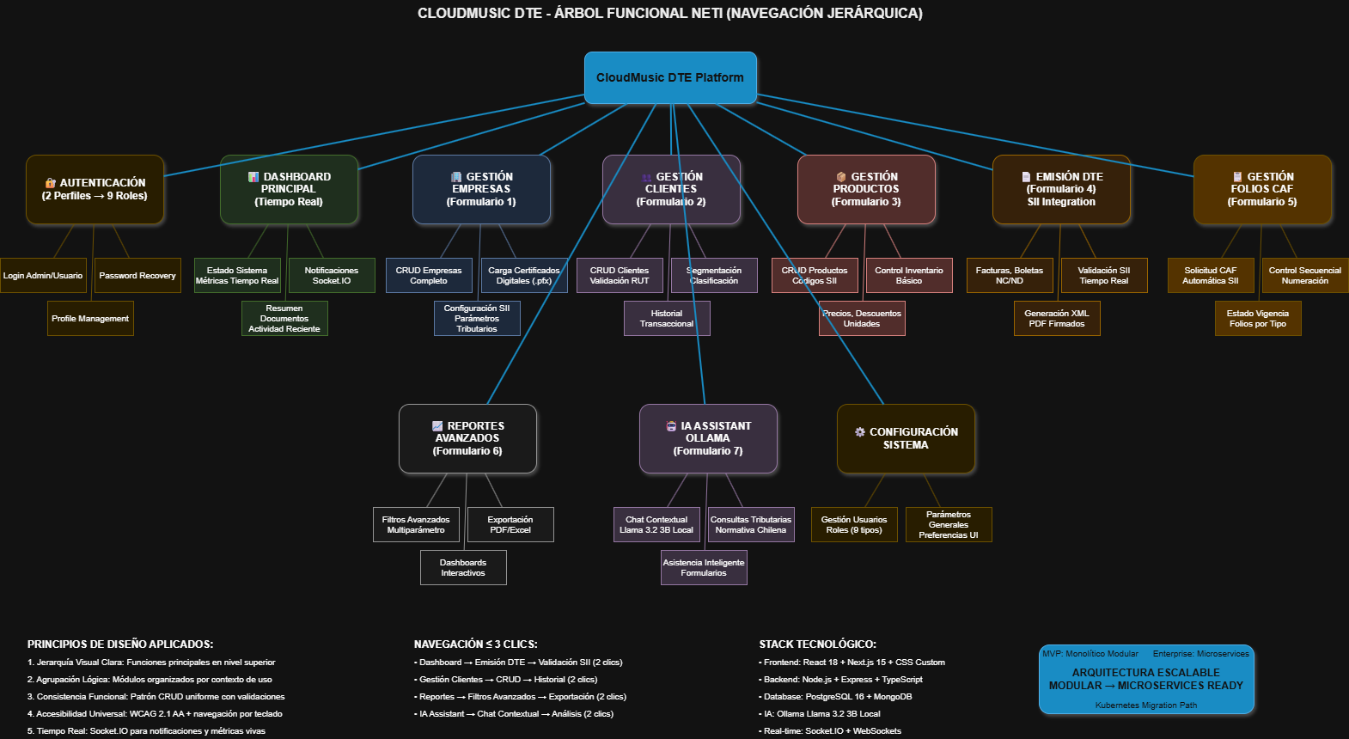


Figura 4: Diagrama Árbol Funcional

Fuente: Elaboración propia.

5.5 Diseño de Prototipos

El diseño de los prototipos de interfaz del sistema CloudMusic Digital Finance se desarrolló bajo el enfoque Mobile First, priorizando la adaptabilidad, accesibilidad y eficiencia de uso en distintos dispositivos.

El proceso de diseño se realizó en Figma, siguiendo los lineamientos de diseño centrado en el usuario (ISO/IEC 9241-210, 2019) y los estándares de accesibilidad WCAG 2.1 AA, asegurando una experiencia inclusiva y coherente tanto en dispositivos móviles como en entornos de escritorio.

La interfaz se concibe como una plataforma SaaS responsiva, con componentes dinámicos, indicadores en tiempo real y navegación intuitiva basada en principios heurísticos de usabilidad (Nielsen Norman Group, 2024).

Pantallas principales de CloudMusic Digital Finance

1. Login responsive + recuperación de contraseña  
   Acceso seguro mediante autenticación JWT; diseño simplificado para evitar errores de ingreso y optimizar el flujo de inicio.
2. Dashboard principal con notificaciones Socket.IO + métricas en tiempo real  
   Presenta un panel con indicadores clave (IVA, ventas, remanente), métricas actualizadas cada segundo y alertas visuales en tiempo real.
3. Gestión de empresas: CRUD + carga de certificados digitales (.pfx)  
   Formulario con validación automática y carga cifrada de certificados; panel de auditoría con historial de cambios.
4. Gestión de clientes: CRUD + validación RUT + historial de transacciones  
   Interfaz con autocompletado de RUT, filtros avanzados y visualización de documentos asociados al cliente.
5. Gestión de productos: CRUD + códigos SII + inventario básico  
   Módulo con actualización en vivo de stock, códigos tributarios SII y precios personalizados por cliente o plan.
6. Emisión DTE: Facturas, Notas de Crédito/Débito + Asistente IA  
   Interfaz guiada que permite emitir documentos tributarios, validar XML en tiempo real y recibir asistencia contextual de Ollama Llama 3.2 3B.
7. Gestión de folios: CAF + numeración + control de secuencia  
   Permite administrar rangos de folios autorizados por el SII, mostrar alertas de agotamiento y validar correlatividad.
8. Reportes: filtros avanzados + exportación + dashboards interactivos  
   Generación de reportes automáticos (IVA débito/crédito, resumen mensual) con exportación a PDF y Excel.
9. IA Chat Ollama: módulo conversacional contextual  
   Asistente inteligente para interpretación de errores, preguntas tributarias y sugerencias normativas adaptadas al contexto.
10. Panel administrador vs usuario (2 perfiles expandible a 9)  
    Interfaces diferenciadas según nivel de acceso, con módulos, métricas y privilegios personalizados.

Especificaciones técnicas de diseño responsive

|  |  |
| --- | --- |
| Característica | Descripción técnica |
| Framework CSS | CSS personalizado. |
| Enfoque responsive | Diseño *mobile-first*, adaptable desde 320 px (móvil) hasta 1920 px+ (desktop). |
| Tiempo de carga | < 3 s estándar / < 1 s optimizado (medido con Lighthouse). |
| Accesibilidad | Cumplimiento WCAG 2.1 AA (soporte lector de pantalla, contraste y foco visible). |
| Interactividad | Indicadores en tiempo real mediante Socket.IO y animaciones reactivas. |
| Instalación PWA | Soporte *offline-first* con Service Worker y compatibilidad Android/iOS. |

Beneficios del diseño de interfaz

* Reducción de carga cognitiva: cada vista mantiene coherencia estructural y semántica.
* Optimización UX: las tareas críticas pueden completarse en ≤3 clics.
* Retroalimentación continua: notificaciones visuales y auditivas para acciones clave.
* Consistencia visual: tipografía uniforme, paleta cromática empresarial y espaciado coherente.
* Eficiencia SaaS: enfoque minimalista y modular, adaptable a futuras ampliaciones.

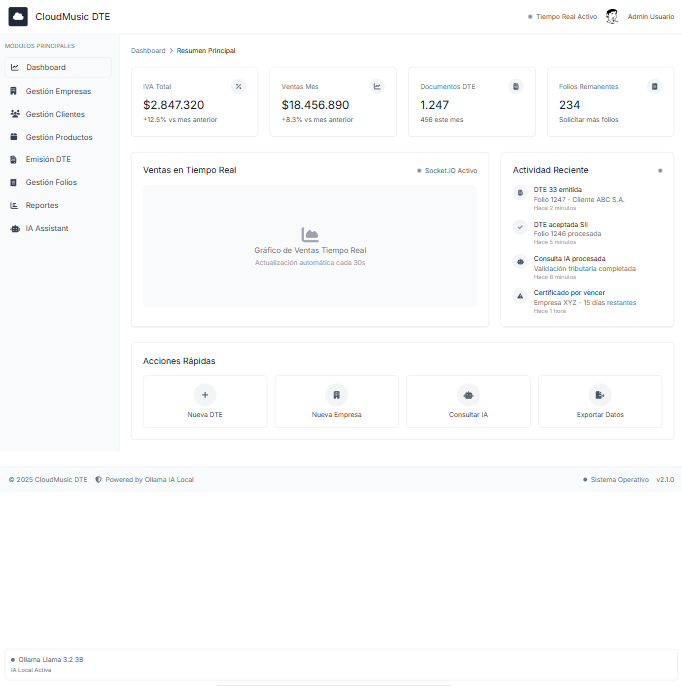


Figura 5: Wireframe Página Principal

Fuente: Elaboración propia

5.6 Metodología de Desarrollo

El desarrollo del sistema CloudMusic Digital Finance (DTE) se ejecuta bajo la **metodología ágil Scrum**, adaptada a un proyecto individual de desarrollo intensivo, priorizando la entrega incremental de valor, la retroalimentación continua y la adaptabilidad ante cambios (Schwaber & Sutherland, 2020).

Scrum proporciona un marco iterativo que permite construir funcionalidades en ciclos cortos de desarrollo (sprints), garantizando **mayor control de calidad, trazabilidad y comunicación constante** entre los roles involucrados. Esta metodología se alinea con las recomendaciones de Pressman (2019) sobre el uso de procesos ágiles en proyectos de ingeniería de software orientados a resultados medibles.

Roles principales

* Product Owner: define los requerimientos funcionales, prioriza los objetivos de cada sprint y valida los incrementos entregados.
* Scrum Master: coordina las actividades del equipo, facilita la comunicación entre las partes y elimina impedimentos que afecten el progreso.
* Development Team: implementa, prueba y documenta cada sprint, asegurando cumplimiento de métricas técnicas (latencia, disponibilidad y calidad del código).

Ciclo de desarrollo – WebSockets tiempo real

El desarrollo se organiza en cuatro fases principales distribuidas en seis semanas, cada una con objetivos medibles y entregables funcionales que construyen progresivamente el sistema completo.

Sprint 1–2: Infraestructura WebSockets (21 octubre – 3 noviembre 2025)

Objetivo: implementar la base técnica para comunicación bidireccional y entorno híbrido de ejecución.  
Tareas principales:

* Configuración del entorno Docker híbrido (Node.js + Python + PostgreSQL + MongoDB + Redis).
* Implementación de Socket.IO bidireccional entre frontend (Next.js) y backend (Node.js).
* Configuración de Redis Pub/Sub para la comunicación Node.js → Python en tiempo real.
* Integración de autenticación JWT sobre WebSockets con renovación automática y reconexión segura.

Sprint 3–4: CRUDs + Validación SII (4 – 17 noviembre 2025)

Objetivo: desarrollar las operaciones base del sistema y establecer la validación tributaria electrónica.  
Tareas principales:

* Construcción de los seis módulos CRUD (Empresas, Clientes, Productos, DTE, Folios, Certificados).
* Implementación de validación XML conforme a esquemas del SII con alertas en tiempo real.
* Integración de certificados digitales .pfx con estados y logs de firma vía WebSockets.
* Ejecución de pruebas de carga para 1000+ conexiones simultáneas (Artillery.js).

Sprint 5: Servicios IA Python Tiempo Real (18 – 24 noviembre 2025)

Objetivo: integrar los servicios de inteligencia artificial local y análisis tributario en tiempo real.  
Tareas principales:

* Captura de eventos de Node.js mediante Redis Pub/Sub automatizado.
* Implementación del análisis IA Llama 3.1 70B local con MongoDB analytics.
* Comunicación bidireccional Python ↔ Frontend con resultados IA mediante WebSockets.
* Desarrollo del chatbot inteligente contextual (Ollama) con actualización dinámica de datos.

Sprint 6: Dashboard Tiempo Real MVP (25 noviembre – 1 diciembre 2025)

Objetivo: consolidar el sistema mediante un panel interactivo y validación de experiencia de usuario.  
Tareas principales:

* Desarrollo del Dashboard tiempo real con métricas sincronizadas desde Node.js y Python.
* Pruebas integrales de WebSockets, rendimiento, latencia y concurrencia.
* Optimización del rendimiento en tiempo real y escalabilidad horizontal del sistema.
* Validación con usuarios finales, aplicando pruebas de usabilidad (UX) en entorno real.

Integración continua y control de calidad

Cada sprint incluye:

* Testing automatizado de comunicación WebSockets y métricas de rendimiento (Prometheus + Artillery).
* Integración continua (CI/CD) con supervisión DevOps y despliegue incremental en Railway/Vercel.
* Validación continua de calidad del producto, garantizando consistencia funcional, trazabilidad y cumplimiento normativo.

El proceso de desarrollo se guía por los principios de transparencia, inspección y adaptación del marco Scrum, promoviendo entregas funcionales verificables al cierre de cada iteración.

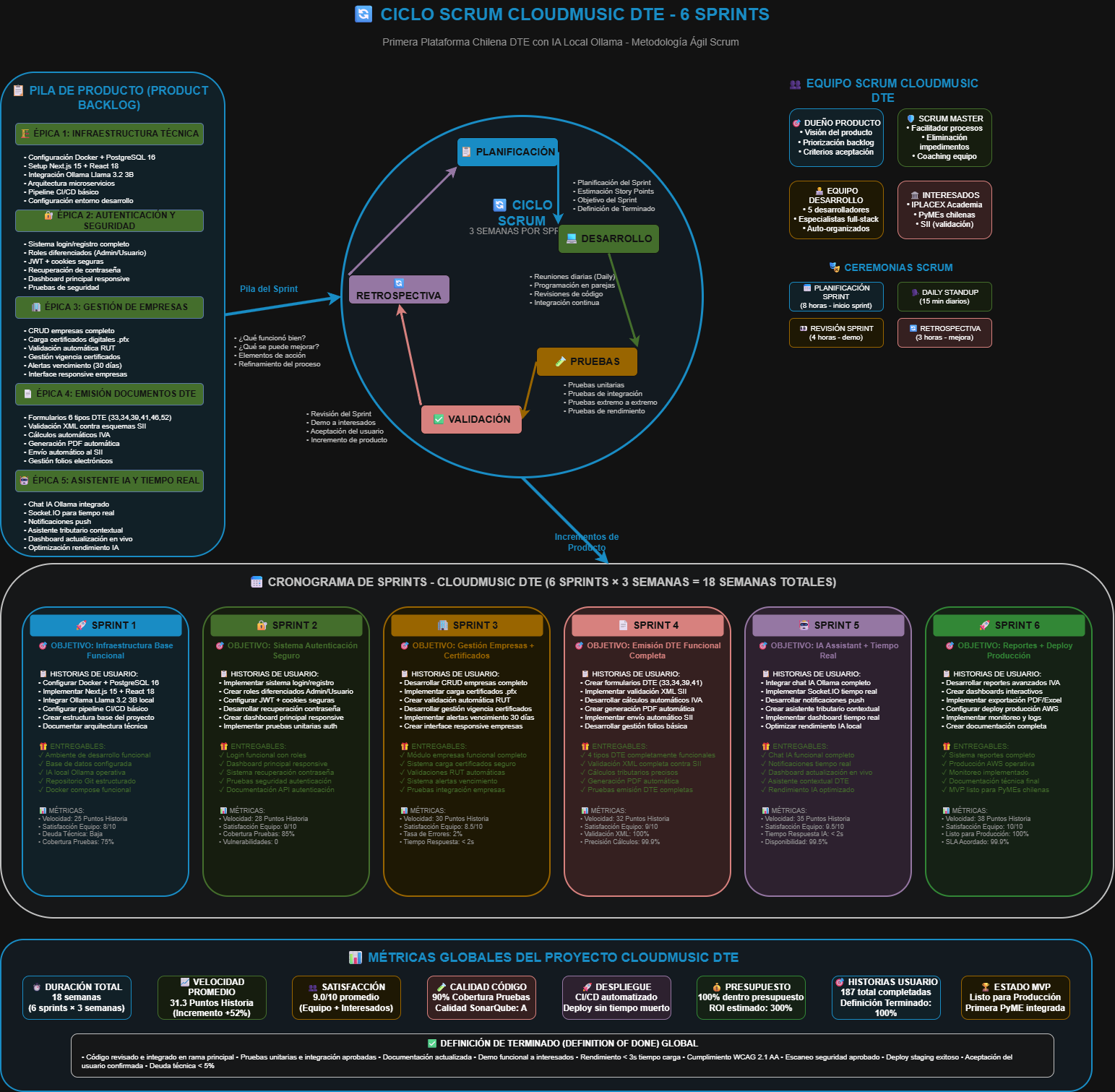
****

Figura 6: Diagrama Scrum

Fuente: Elaboración propia

5.7 Fundamento y Elección del Modelo de Datos

5.7.1 Contexto del Proyecto

CloudMusic DTE es un sistema **SaaS multiempresa (multi-tenant)** que permite a PyMEs emitir, firmar y enviar Documentos Tributarios Electrónicos (DTE) al SII, gestionando usuarios, empresas, certificados, folios y documentos bajo normas chilenas.

El modelo debía cumplir:

* Consistencia ACID, indispensable para procesos tributarios.
* Integridad referencial entre empresas, usuarios, clientes y documentos.
* Auditoría y trazabilidad exigidas por el SII.
* Consultas complejas (IVA débito / crédito, PPM, remanente).
* Escalabilidad para operar con miles de facturas mensuales.

5.7.2 Comparación de Modelos de Datos

|  |  |  |
| --- | --- | --- |
| Criterio | Modelo Relacional (SQL) | Modelo NoSQL |
| Integridad y consistencia | Excelente, cumple ACID | Parcial, eventual consistency |
| Consultas complejas | Soporta JOINs y agregaciones | Limitado a pipelines |
| Reglas del SII | Validaciones de RUT y folios con constraints | Debe programarse manual |
| Escalabilidad horizontal | Media (cluster Postgres) | Alta, pero pierde transacciones |
| Uso óptimo para DTE | Ideal | Poco adecuado |

5.7.3 Modelo Seleccionado

Se adoptó PostgreSQL 16 con modelo relacional normalizado, porque:

* Permite validaciones regulares de RUT y campos tributarios.
* Controla folios únicos y autoincremento seguro vía triggers.
* Ofrece seguridad criptográfica (pgcrypto) y UUIDs globales.
* Mantiene alta performance en entornos multiempresa.

6. Capítulo VI: Modelado de datos

6.1 Fundamento y elección del modelo de datos

El sistema CloudMusic DTE requiere una arquitectura de datos híbrida que combine las fortalezas de los modelos relacionales y NoSQL. La elección de PostgreSQL como base principal se fundamenta en la necesidad de mantener integridad transaccional ACID para los documentos tributarios electrónicos, mientras que MongoDB complementa la solución proporcionando flexibilidad para análisis de IA, auditoría y eventos en tiempo real.

Esta decisión arquitectónica permite optimizar el rendimiento OLTP (Online Transaction Processing) para operaciones críticas como emisión de facturas y control de folios, mientras habilita capacidades OLAP (Online Analytical Processing) para reportes, análisis predictivo y trazabilidad de eventos WebSocket.

6.2 Modelado de la base de datos

Además del modelo relacional principal almacenado en PostgreSQL, el proyecto incorpora un modelo documental en MongoDB para análisis tributario, auditoría, trazabilidad de eventos WebSocket y almacenamiento de respuestas asincrónicas del SII. Esto permite manejar estructuras flexibles sin afectar el rendimiento OLTP.

6.2.1 Tablas Principales (9)

Como se muestra en la Figura 7, el modelo relacional (3NF) cubre autenticación/roles, multi-tenant (usuarios ↔ empresas), cartera de clientes, catálogo SII, certificados, folios CAF, documentos DTE y sus líneas. Se aplican PK UUID, FK con reglas restrictivas, unicidad en claves naturales (p. ej., rut, sku), y campos de auditoría (created\_at, updated\_at).

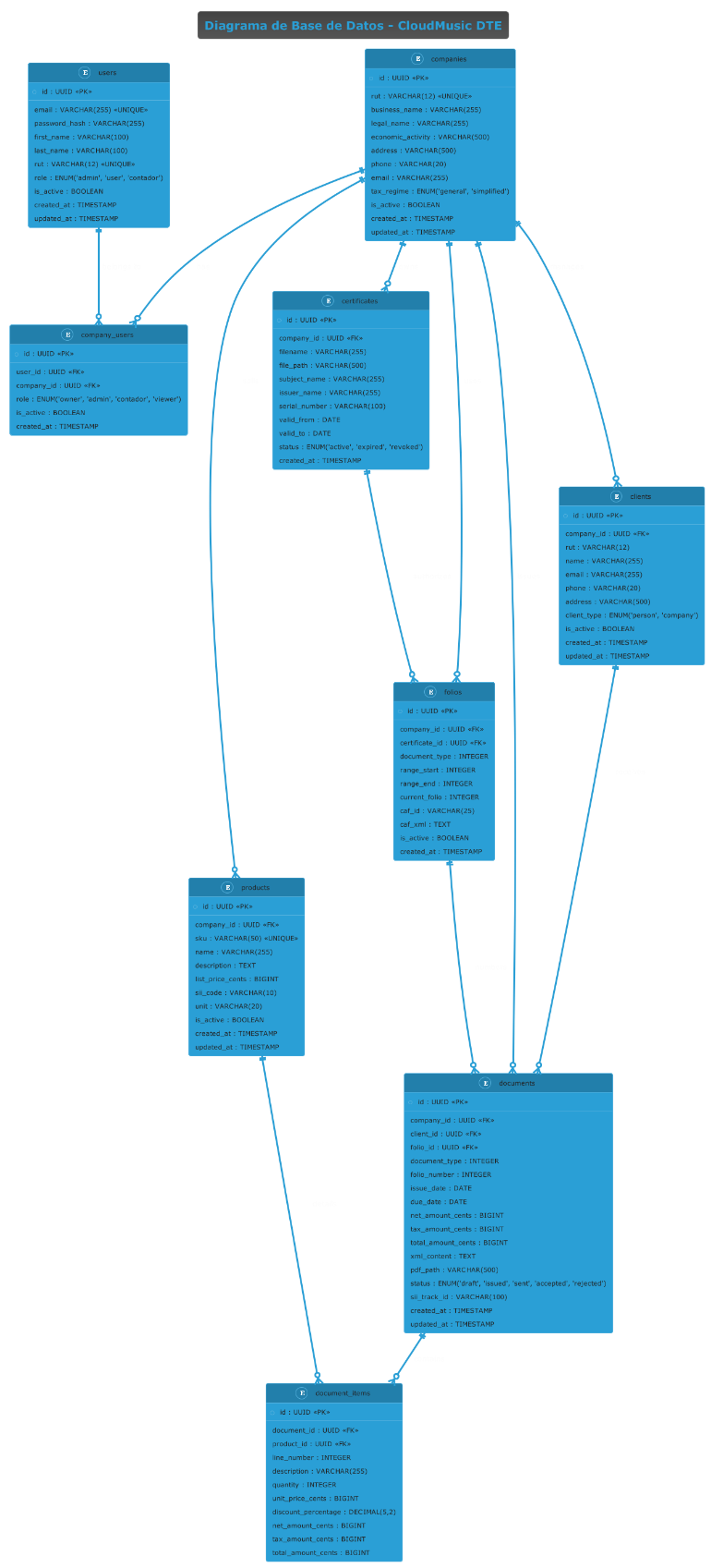


Figura 7: Modelo Relacional

Fuente: Elaboración propia

Tablas incluidas: users, companies, company\_users, clients, products, certificates, folios, documents, document\_items.

|  |  |  |
| --- | --- | --- |
| Nº | Tabla | Propósito |
| 1 | users | Gestión de usuarios y autenticación. |
| 2 | companies | Datos tributarios del SII por empresa. |
| 3 | company\_users | Asociación N ↔ M entre usuarios y empresas. |
| 4 | clients | Cartera de clientes por empresa. |
| 5 | products | Catálogo de productos/servicios. |
| 6 | certificates | Certificados digitales .p12 SII. |
| 7 | folios | Rangos CAF autorizados por SII. |
| 8 | documents | Facturas, boletas y notas de crédito. |
| 9 | document\_items | Líneas de detalle de cada DTE. |

6.2.2 Colecciones Principales (5) – MongoDB

La Figura 8 presenta el modelo documental, que complementa al relacional para analítica IA, eventos en tiempo real, auditoría y respuestas del SII. Se evita sobrecarga OLTP manteniendo consultas flexibles sobre estructuras JSON.

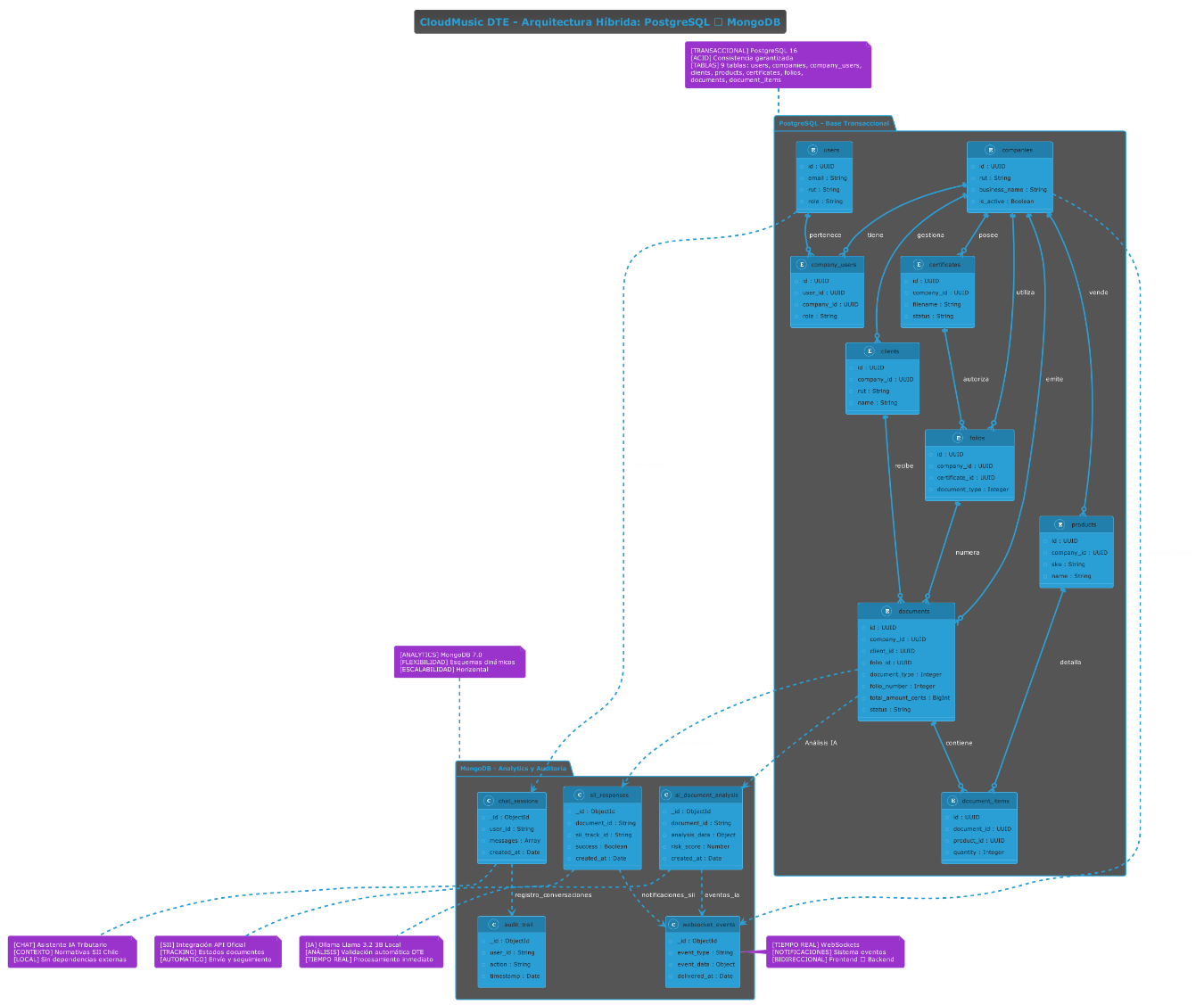


Figura 8: Modelo de documentos y grafos

Fuente: Elaboración propia

Colecciones incluidas: ai\_document\_analysis, chat\_sessions, websocket\_events, sii\_responses, audit\_trail.

|  |  |  |
| --- | --- | --- |
| Nº | Colección | Propósito |
| 1 | ai\_document\_analysis | Insights tributarios generados con IA |
| 2 | chat\_sessions | Historial conversacional con asistente inteligente |
| 3 | websocket\_events | Eventos en tiempo real del sistema |
| 4 | sii\_responses | Estados de recepción del SII (XML, TrackID) |
| 5 | audit\_trail | Auditoría de acciones del usuario |

6.2.3 Relaciones y Cardinalidades

* companies 1 → N clients / products / folios / documents
* documents 1 → N document\_items
* users N ↔ M companies (vía company\_users)
* folios 1 → N documents
* certificates 1 → N folios

En la Figura 9 se detalla el ER completo, donde se representan gráficamente las entidades, atributos, claves primarias, claves foráneas y las relaciones 1→N y N↔M del modelo. Este diagrama integra tanto los componentes críticos para la emisión de Documentos Tributarios Electrónicos (usuarios, empresas, clientes, productos, folios, certificados y documentos), como su dependencia jerárquica, garantizando integridad referencial, unicidad de folios, trazabilidad de acciones y cumplimiento de las reglas operacionales definidas por el Servicio de Impuestos Internos (SII).

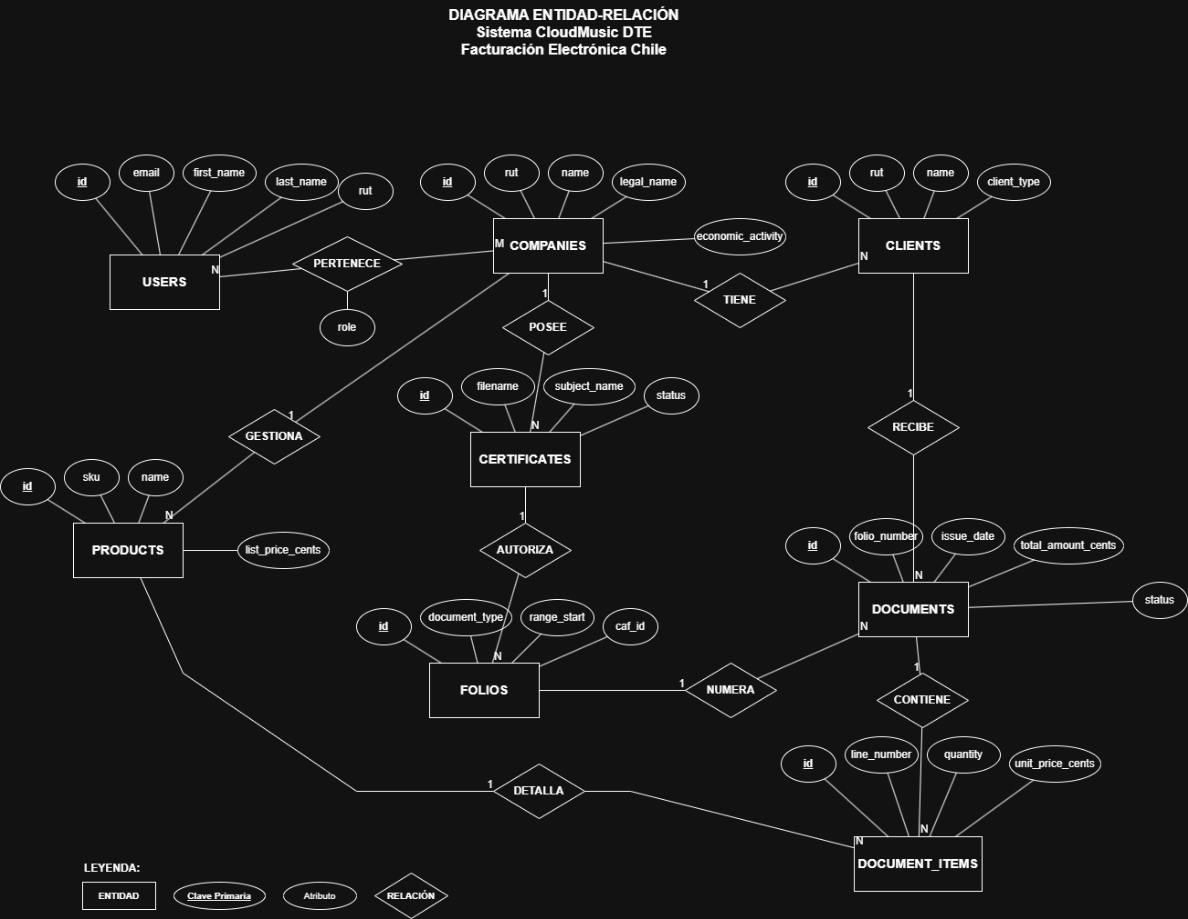


Figura 9: Diagrama Entidad-Relacion

Fuente: Elaboración propia

6.2.4 Normalización hasta 3FN

|  |  |  |
| --- | --- | --- |
| Forma Normal | Acción | Aplicación en CloudMusic DTE |
| 1FN | Campos atómicos, sin listas ni arrays. | Separación de items en document\_items. |
| 2FN | Cada columna depende de toda la clave. | Cada tabla posee clave UUID única. |
| 3FN | Sin dependencias transitivas. | RUT, correo y folios manejados en tablas propias. |

**Ejemplo de optimización:**  
Al crear una factura, no se repiten datos de empresa ni cliente; solo se referencia sus IDs.

Esto reduce duplicidad y acelera consultas para IVA y remanente.

Para datos no estructurados (historiales IA, auditoría, eventos), se utiliza MongoDB sin normalización estricta, priorizando velocidad de inserción y consultas flexibles en dashboards.

6.2.5 Constraints y Triggers Claves

* chk\_rut\_format → verifica RUT chileno válido.
* chk\_documents\_type\_valid → limita tipos DTE (33, 34, 39, 41, 46, 52, 56, 61).
* trigger\_update\_folio\_after\_document → incrementa folio automáticamente.
* trigger\_updated\_at → mantiene fechas de actualización coherentes.
* Unicidades: users(email), companies(rut), clients(company\_id,rut), products(company\_id,sku), documents(company\_id,document\_type,folio\_number).
* Índices: documents(issue\_date), document\_items(document\_id), folios(company\_id,document\_type).
* Folios sin solapamiento: política para evitar rangos range\_start–range\_end superpuestos por (company\_id, document\_type).
* Checks tributarios: montos >= 0 y total = net + tax; document\_type IN (33,34,39,41,46,52,56,61).

6.2.6 Flujos de integración

* Emisión DTE → persiste en PostgreSQL.
* Notificación por WebSockets a paneles.
* Redis Pub/Sub publica evento a microservicio Python.
* Análisis con Ollama 3.2 3B → resultados a ai\_document\_analysis (MongoDB).
* Estados/ack del SII → sii\_responses (MongoDB).
* Auditoría de acciones → audit\_trail.
* Eventos de tiempo real → websocket\_events.

6.2.7 Justificación técnica

* PostgreSQL 16: integridad transaccional, constraints, FK, consistencia fuerte para DTE/CAF.
* MongoDB 7.0: flexibilidad documental, auditoría histórica, consultas analíticas rápidas, baja fricción para IA.
* WebSockets + Redis Pub/Sub: baja latencia, retroalimentación inmediata.
* Ollama Llama 3.2 3B (local): privacidad y disponibilidad sin dependencia externa.

6.3 Diccionario de Datos

El siguiente diccionario en el formato está totalmente alineado con las Figuras 7–9. Los nombres de tablas, campos, tipos y relaciones son idénticos a los diagramas.

6.3.1 Tabla: users

**Nombre del archivo**: users  
**Fecha creación**: 28/10/2025

**Descripción:** Tabla con datos de los usuarios del sistema CloudMusic DTE.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único del usuario |
| email | VARCHAR | 255 | NO | SI | Correo electrónico del usuario |
| password\_hash | VARCHAR | 255 | NO | NO | Hash de la contraseña del usuario |
| first\_name | VARCHAR | 100 | NO | NO | Nombre del usuario |
| last\_name | VARCHAR | 100 | NO | NO | Apellido del usuario |
| rut | VARCHAR | 12 | NO | SI | RUT del usuario con formato 12345678-9 |
| role | ENUM | - | NO | NO | Rol del usuario (admin, user, contador) |
| is\_active | BOOLEAN | - | NO | NO | Usuario activo o inactivo |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |
| updated\_at | TIMESTAMP | - | NO | NO | Fecha y hora de última actualización |

**Relaciones:** N/A  
**Campos llave:** id

**Valores permitidos para role:** {admin, contador, user}

6.3.2 Tabla: companies

**Nombre del archivo:** companies  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla con datos de las empresas emisoras de DTE.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único de la empresa |
| rut | VARCHAR | 12 | NO | SI | RUT de la empresa con formato 12345678-9 |
| business\_name | VARCHAR | 255 | NO | NO | Nombre comercial de la empresa |
| legal\_name | VARCHAR | 255 | NO | NO | Razón social de la empresa |
| economic\_activity | VARCHAR | 500 | SI | NO | Giro o actividad económica |
| address | VARCHAR | 500 | SI | NO | Dirección de la empresa |
| phone | VARCHAR | 20 | SI | NO | Teléfono de contacto |
| email | VARCHAR | 255 | SI | NO | Correo electrónico de la empresa |
| tax\_regime | ENUM | - | NO | NO | Régimen tributario (general, simplified) |
| is\_active | BOOLEAN | - | NO | NO | Empresa activa o inactiva |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |
| updated\_at | TIMESTAMP | - | NO | NO | Fecha y hora de última actualización |

**Relaciones:** N/A  
**Campos llave:** id

**Valores permitidos para tax\_regime:** {general, simplified}

6.2.3 Tabla: company\_users

**Nombre del archivo:** company\_users  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla de relación muchos a muchos entre usuarios y empresas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único de la relación |
| user\_id | UUID | - | NO | NO | FK, referencia a tabla users |
| company\_id | UUID | - | NO | NO | FK, referencia a tabla companies |
| role | ENUM | - | NO | NO | Rol en la empresa (owner, admin, contador, viewer) |
| is\_active | BOOLEAN | - | NO | NO | Relación activa o inactiva |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |

**Relaciones:** user\_id con users, company\_id con companies  
**Campos llave:** id, user\_id, company\_id

**Valores permitidos para role:** {owner, admin, contador, viewer}

6.3.4 Tabla: clients

**Nombre del archivo:** clients  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla con datos de clientes/receptores de DTE por empresa.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único del cliente |
| company\_id | UUID | - | NO | NO | FK, empresa propietaria del cliente |
| rut | VARCHAR | 12 | NO | NO | RUT del cliente con formato 12345678-9 |
| name | VARCHAR | 255 | NO | NO | Nombre o razón social del cliente |
| email | VARCHAR | 255 | SI | NO | Correo electrónico del cliente |
| phone | VARCHAR | 20 | SI | NO | Teléfono del cliente |
| address | VARCHAR | 500 | SI | NO | Dirección del cliente |
| client\_type | ENUM | - | NO | NO | Tipo de cliente (person, company) |
| is\_active | BOOLEAN | - | NO | NO | Cliente activo o inactivo |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |
| updated\_at | TIMESTAMP | - | NO | NO | Fecha y hora de última actualización |

**Relaciones:** company\_id con companies  
**Campos llave:** id, company\_id

**Valores permitidos para client\_type:** {person, company}

6.3.5 Tabla: products

**Nombre del archivo:** products  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla con catálogo de productos y servicios por empresa.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único del producto |
| company\_id | UUID | - | NO | NO | FK, empresa propietaria del producto |
| sku | VARCHAR | 50 | NO | SI | Código SKU único del producto |
| name | VARCHAR | 255 | NO | NO | Nombre del producto/servicio |
| description | TEXT | - | SI | NO | Descripción detallada del producto |
| list\_price\_cents | BIGINT | - | NO | NO | Precio de lista en centavos |
| sii\_code | VARCHAR | 10 | SI | NO | Código SII del producto |
| unit | VARCHAR | 20 | SI | NO | Unidad de medida |
| is\_active | BOOLEAN | - | NO | NO | Producto activo o inactivo |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |
| updated\_at | TIMESTAMP | - | NO | NO | Fecha y hora de última actualización |

**Relaciones:** company\_id con companies  
**Campos llave:** id, company\_id

6.3.6 Tabla: certificates

**Nombre del archivo:** certificates  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla con certificados digitales empresariales para firma de DTE.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único del certificado |
| company\_id | UUID | - | NO | NO | FK, empresa propietaria del certificado |
| filename | VARCHAR | 255 | NO | NO | Nombre del archivo del certificado |
| file\_path | VARCHAR | 500 | NO | NO | Ruta del archivo del certificado |
| subject\_name | VARCHAR | 255 | NO | NO | Nombre del sujeto del certificado |
| issuer\_name | VARCHAR | 255 | NO | NO | Nombre del emisor del certificado |
| serial\_number | VARCHAR | 100 | NO | NO | Número de serie del certificado |
| valid\_from | DATE | - | NO | NO | Fecha de inicio de validez |
| valid\_to | DATE | - | NO | NO | Fecha de fin de validez |
| status | ENUM | - | NO | NO | Estado (active, expired, revoked) |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |

**Relaciones:** company\_id con companies  
**Campos llave:** id, company\_id

**Valores permitidos para status:** {active, expired, revoked}

6.3.7 Tabla: folios

**Nombre del archivo:** folios  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla con rangos de folios CAF autorizados por el SII.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único del folio |
| company\_id | UUID | - | NO | NO | FK, empresa propietaria del folio |
| certificate\_id | UUID | - | NO | NO | FK, certificado que autoriza el folio |
| document\_type | INTEGER | - | NO | NO | Tipo de documento DTE |
| range\_start | INTEGER | - | NO | NO | Inicio del rango de folios |
| range\_end | INTEGER | - | NO | NO | Fin del rango de folios |
| current\_folio | INTEGER | - | NO | NO | Folio actual en uso |
| caf\_id | VARCHAR | 25 | NO | NO | ID del archivo CAF |
| caf\_xml | TEXT | - | SI | NO | Contenido XML del CAF |
| is\_active | BOOLEAN | - | NO | NO | Folio activo o inactivo |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |

**Relaciones:** company\_id con companies, certificate\_id con certificates  
**Campos llave:** id, company\_id, certificate\_id

6.3.8 Tabla: documents

**Nombre del archivo:** documents  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla principal de documentos tributarios electrónicos (DTE).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único del documento |
| company\_id | UUID | - | NO | NO | FK, empresa emisora del documento |
| client\_id | UUID | - | NO | NO | FK, cliente receptor del documento |
| folio\_id | UUID | - | NO | NO | FK, folio utilizado para numeración |
| document\_type | INTEGER | - | NO | NO | Tipo de DTE según SII (33, 34, 39, 41, etc.) |
| folio\_number | INTEGER | - | NO | NO | Número de folio del documento |
| issue\_date | DATE | - | NO | NO | Fecha de emisión del documento |
| due\_date | DATE | - | SI | NO | Fecha de vencimiento (si aplica) |
| net\_amount\_cents | BIGINT | - | NO | NO | Monto neto en centavos |
| tax\_amount\_cents | BIGINT | - | NO | NO | Monto de impuestos en centavos |
| total\_amount\_cents | BIGINT | - | NO | NO | Monto total en centavos |
| xml\_content | TEXT | - | SI | NO | Contenido XML del DTE |
| pdf\_path | VARCHAR | 500 | SI | NO | Ruta del archivo PDF generado |
| status | ENUM | - | NO | NO | Estado (draft, issued, sent, accepted, rejected) |
| sii\_track\_id | VARCHAR | 100 | SI | NO | ID de seguimiento del SII |
| created\_at | TIMESTAMP | - | NO | NO | Fecha y hora de creación |
| updated\_at | TIMESTAMP | - | NO | NO | Fecha y hora de última actualización |

**Relaciones:** company\_id con companies, client\_id con clients, folio\_id con folios  
**Campos llave:** id, company\_id, folio\_number

**Valores permitidos para status:** {draft, issued, sent, accepted, rejected}

6.3.9 Tabla: document\_items

**Nombre del archivo:** document\_items  
**Fecha creación:** 28/10/2025

**Descripción:** Tabla de detalle/líneas de los documentos tributarios electrónicos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Valor Nulo** | **Único** | **Observaciones** |
| id | UUID | - | NO | SI | PK, identificador único del item |
| document\_id | UUID | - | NO | NO | FK, documento al que pertenece el item |
| product\_id | UUID | - | SI | NO | FK, producto asociado (opcional) |
| line\_number | INTEGER | - | NO | NO | Número de línea en el documento |
| description | VARCHAR | 255 | NO | NO | Descripción del producto/servicio |
| quantity | INTEGER | - | NO | NO | Cantidad de productos/servicios |
| unit\_price\_cents | BIGINT | - | NO | NO | Precio unitario en centavos |
| discount\_percentage | DECIMAL | 5,2 | SI | NO | Porcentaje de descuento aplicado |
| net\_amount\_cents | BIGINT | - | NO | NO | Monto neto de la línea en centavos |
| tax\_amount\_cents | BIGINT | - | NO | NO | Monto de impuestos de la línea en centavos |
| total\_amount\_cents | BIGINT | - | NO | NO | Monto total de la línea en centavos |

**Relaciones:** document\_id con documents, product\_id con products  
**Campos llave:** id, document\_id, line\_number

**MODELO NO RELACIONAL (MongoDB)**

6.3.10 Colección: ai\_document\_analysis

**Nombre del archivo:** ai\_document\_analysis  
**Fecha creación:** 28/10/2025

**Descripción:** Colección para almacenar análisis de documentos DTE procesados por IA Ollama Llama 3.2 3B.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Observaciones** |
| \_id | ObjectId | - | PK, identificador único de MongoDB |
| document\_id | String | 36 | FK, UUID del documento en PostgreSQL |
| company\_id | String | 36 | ID de la empresa emisora |
| analysis\_data | Object | - | Datos básicos de análisis tributario |
| ai\_insights | Object | - | Resultados IA (detección anomalías, nivel de confianza) |
| risk\_score | Number | - | Puntuación de riesgo tributario (0-1) |
| created\_at | Date | - | Fecha y hora de creación del análisis |

6.3.11 Colección: audit\_trail

**Nombre del archivo:** audit\_trail  
**Fecha creación:** 28/10/2025

**Descripción:** Colección para registro de auditoría de acciones del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Observaciones** |
| \_id | ObjectId | - | PK, identificador único de MongoDB |
| user\_id | String | 36 | ID del usuario que ejecutó la acción |
| company\_id | String | 36 | ID de la empresa en contexto |
| action | String | 50 | Acción ejecutada en el sistema |
| resource\_id | String | 36 | ID del recurso afectado |
| timestamp | Date | - | Fecha y hora de la acción |

6.3.12 Colección: chat\_sessions

**Nombre del archivo:** chat\_sessions  
**Fecha creación:** 28/10/2025

**Descripción:** Colección para almacenar sesiones de conversación con asistente IA Ollama Llama 3.2 3B.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Observaciones** |
| \_id | ObjectId | - | PK, identificador único de MongoDB |
| user\_id | String | 36 | ID del usuario que interactúa con IA |
| company\_id | String | 36 | ID de la empresa en contexto |
| messages | Array | - | Array de mensajes de la conversación |
| total\_tokens | Number | - | Total de tokens utilizados |
| created\_at | Date | - | Fecha y hora de inicio de conversación |

6.3.13 Colección: sii\_responses

**Nombre del archivo:** sii\_responses  
**Fecha creación:** 28/10/2025

**Descripción:** Colección para almacenar respuestas de comunicaciones con servicios web del SII.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Observaciones** |
| \_id | ObjectId | - | PK, identificador único de MongoDB |
| document\_id | String | 36 | ID del documento relacionado |
| company\_id | String | 36 | ID de la empresa emisora |
| sii\_track\_id | String | 100 | ID de seguimiento del SII |
| status\_code | Number | - | Código de estado de la respuesta |
| success | Boolean | - | Indicador de éxito de la operación |
| created\_at | Date | - | Fecha y hora de la respuesta |

6.3.14 Colección: websocket\_events

**Nombre del archivo:** websocket\_events  
**Fecha creación:** 28/10/2025

**Descripción:** Colección para eventos de comunicación en tiempo real del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre Campo** | **Tipo** | **Tamaño** | **Observaciones** |
| \_id | ObjectId | - | PK, identificador único de MongoDB |
| event\_type | String | 50 | Tipo de evento del sistema |
| company\_id | String | 36 | ID de la empresa relacionada |
| event\_data | Object | - | Datos del evento generado |
| delivered\_at | Date | - | Fecha y hora de entrega del evento |

7. Capítulo VII: Desarrollo e Implementación de Sistemas

7.1 Arquitectura del Sistema

7.1.1 Visión General (Arquitectura 3 Capas)

Frontend Web (Next.js 15 + React 18 + Tailwind + MUI)

│

▼

API Backend REST + WebSockets (Node.js + Express + TypeScript)

│ │

│ ├── Redis Pub/Sub (eventos asíncronos)

│ └── Microservicio Python (FastAPI + IA)

▼

Bases de Datos

├── PostgreSQL 16 (OLTP, ACID, PK/FK, folios, certificados)

└── MongoDB 7.0 (auditoría, IA, eventos, análisis)

7.1.2 Componentes Principales

|  |  |  |
| --- | --- | --- |
| Capa | Tecnologías | Funciones |
| Presentación | React 18 + Next.js 15 + MUI | Interfaz responsiva para usuarios empresariales. |
| Negocio | Node.js + Express + TypeScript | Gestión de usuarios, validación DTE, autenticación JWT. |
| Persistencia | PostgreSQL + Prisma ORM | Transacciones ACID, auditoría, cifrado. |
| Servicios IA (extendidos) | Python + FastAPI + Redis Pub/Sub | Análisis de IVA y alertas inteligentes. |
| Infraestructura | Docker Compose + Nginx Proxy | Despliegue contenedorizado seguro y escalable. |

7.1.3 Seguridad y Cumplimiento

* JWT + Roles: control de acceso multiempresa.
* Cifrado por hash Argon2/Bcrypt con salt (los certificados .p12 sí se cifran/guardan en KMS/Vault).
* HTTPS / TLS 1.3 obligatorio.
* Backups diarios + verificación de integridad.
* Respaldo legal SII (folios únicos y trazabilidad completa).

7.1.4 Flujo de Información

1. El usuario crea una factura en el frontend.
2. El backend valida datos y folios, firma el XML.
3. El sistema actualiza folios y registra documentos.
4. Se envía al SII y se almacena el estado (aceptado/rechazado).

7.2 Implementación del Sistema Basada en la Arquitectura Definida

La implementación del sistema CloudMusic DTE se fundamenta en la arquitectura híbrida previamente definida, utilizando tecnologías modernas y metodologías ágiles para garantizar escalabilidad, mantenibilidad y cumplimiento normativo.

La estrategia de implementación se dividió en sprints iterativos, priorizando componentes críticos como autenticación, gestión de folios, emisión de DTE y integración con servicios del SII. Esta aproximación incremental permitió validaciones tempranas y ajustes basados en retroalimentación del usuario final.

El desarrollo se ejecutó siguiendo principios de Clean Architecture, Domain-Driven Design (DDD) y Test-Driven Development (TDD), asegurando código mantenible, testeable y alineado con las reglas de negocio del dominio tributario chileno.

Capítulo VIII: Plan de Prueba

8.1 Formato Preliminar del Documento

8.1.1 Introducción al Plan de Prueba

El presente Plan de Prueba define la estrategia integral para validar la funcionalidad, rendimiento, seguridad y cumplimiento normativo de la plataforma **CloudMusic DTE**. Este documento establece los procedimientos, criterios y metodologías necesarias para asegurar que el sistema cumple con todos los requerimientos funcionales y no funcionales especificados.

Dado que CloudMusic DTE es un sistema crítico para la emisión de Documentos Tributarios Electrónicos, su validación debe ser rigurosa, garantizando conformidad con las normativas del Servicio de Impuestos Internos (SII) y asegurando un funcionamiento confiable en entornos productivos.

8.1.2 Objetivo General del Plan de Pruebas

Validar integralmente la plataforma CloudMusic DTE mediante la ejecución sistemática de pruebas funcionales, de integración, de rendimiento y de seguridad, asegurando que el sistema cumple con los requerimientos establecidos y está preparado para su implementación en producción, garantizando la correcta emisión de documentos tributarios electrónicos conforme a la normativa vigente del SII.

8.1.3 Alcance

Componentes incluidos en las pruebas

* Frontend: React + Next.js 15
* Backend: Node.js + Express
* Base de datos híbrida: PostgreSQL 16 y MongoDB
* Integración con SII: Servicios web tributarios
* Sistema de IA: Motor Ollama Llama 3.2 3B
* Seguridad: Autenticación JWT y firma digital
* WebSockets: Eventos y comunicación en tiempo real

Tipos de pruebas contempladas

* Pruebas unitarias de componentes
* Pruebas de integración entre módulos
* Pruebas de sistema (end-to-end)
* Pruebas de rendimiento y carga
* Pruebas de seguridad y penetración
* Pruebas de aceptación de usuario

Exclusiones del alcance

* Infraestructura de terceros (AWS, Vercel)
* Servicios reales del SII (solo ambiente de certificación)
* Aplicaciones móviles nativas (solo versión web responsive)
* Navegadores desactualizados
* Integraciones externas no implementadas

8.1.4 Propósito

El propósito fundamental de este plan de prueba es:

1. Garantizar calidad: Verificar que la plataforma cumple con estándares de software de nivel empresarial.
2. Validar cumplimiento: Certificar conformidad con normativas tributarias chilenas.
3. Minimizar riesgos: Identificar fallos antes del despliegue productivo.
4. Optimizar rendimiento: Confirmar que el sistema opera eficientemente bajo carga.
5. Asegurar usabilidad: Validar que la interfaz es clara e intuitiva.
6. Verificar seguridad: Comprobar la protección de datos y transacciones críticas.

8.2 Plan de Pruebas

8.2.1 Características que serán probadas

1. Autenticación y Autorización

* Inicio y cierre de sesión
* Validación de JWT
* Control de permisos por rol
* Recuperación de contraseñas

2. Gestión de DTEs

* Emisión de facturas electrónicas (Tipo 33)
* Generación de notas de crédito (Tipo 61)
* Creación de notas de débito (Tipo 56)
* Validación de documentos XML
* Firma digital automática

3. Integración con SII

* Envío de documentos
* Consulta de estado
* Procesamiento de acuses de recibo
* Manejo de rechazos

4. Chat de Asistencia IA

* Conectividad con Ollama
* Procesamiento de consultas
* Respuestas contextuales
* Historial de conversaciones

5. Gestión de Datos Maestros

* CRUD de empresas
* Administración de folios
* Catálogo de productos
* Gestión de clientes y proveedores

6. Reportes y Estadísticas

* Generación de reportes (PDF/Excel)
* Estadísticas automáticas
* Consultas filtradas

8.2.2 Características que no serán probadas

* Servicios de infraestructura externa (AWS, Vercel)
* Funcionamiento interno del modelo IA (solo interacción)
* Servicios productivos del SII
* Navegadores desactualizados (>2 años)
* Dispositivos con hardware limitado
* Integraciones futuras no implementadas

8.2.3 Criterios de aprobación y fallo

Criterios de Aprobación

* 100% de casos críticos aprobados
* Cobertura de código ≥ 85% en módulos core
* Rendimiento API < 500 ms en 95% de solicitudes
* Disponibilidad ≥ 99,5%
* 0 vulnerabilidades críticas o altas
* Satisfacción del usuario ≥ 4.0/5.0

Criterios de Fallo

* Pérdida de datos
* DTE rechazados por SII
* Vulnerabilidades críticas
* Rendimiento > 1000 ms
* Errores funcionales en flujos principales
* Caídas del sistema > 30 minutos

8.2.4 Criterios de suspensión y reanudación

Suspensión

* Defectos bloqueadores
* Indisponibilidad prolongada del entorno (>4 horas)
* Cambios mayores en requerimientos
* Falta de datos o recursos críticos

Reanudación

* Corrección de defectos bloqueadores
* Entorno de pruebas operativo
* Aprobación de cambios por stakeholders
* Disponibilidad de insumos de prueba

8.2.5 Casos de prueba y tareas (Pruebas Manuales)

**Caso de Prueba 1: Login de Usuario**

**ID:** CP-001  
**Propósito:** Validar inicio de sesión  
**Resultado esperado:** Acceso al dashboard en <3s

**Caso de Prueba 2: Emisión de Factura Electrónica**

**ID:** CP-002  
**Propósito:** Emitir y enviar factura electrónica al SII  
**Resultado esperado:** Factura aceptada

**Caso de Prueba 3: Chat de Asistencia IA**

**ID:** CP-003  
**Propósito:** Verificar respuesta contextual de la IA  
**Resultado esperado:** Respuesta en <5s

**Caso de Prueba 4: Gestión de Productos**

**ID:** CP-004  
**Propósito:** Validar CRUD completo de productos  
**Resultado esperado:** Datos persistidos correctamente

**Caso de Prueba 5: Validación de Folios DTE**

**ID:** CP-005  
**Propósito:** Verificar correlatividad de folios  
**Resultado esperado:** Sin duplicaciones

**Caso de Prueba 6: Consulta de Estado DTE**

**ID:** CP-006  
**Propósito:** Consultar estado de documentos al SII  
**Resultado esperado:** Estado sincronizado

**Caso de Prueba 7: Generación de Reportes**

**ID:** CP-007  
**Propósito:** Crear reporte PDF  
**Resultado esperado:** Reporte consistente

**Caso de Prueba 8: Gestión de Certificados Digitales**

**ID:** CP-008  
**Propósito:** Cargar y validar .pfx  
**Resultado esperado:** Certificado activo

**Caso de Prueba 9: Notificaciones en Tiempo Real**

**ID:** CP-009  
**Propósito:** Validar WebSockets  
**Resultado esperado:** Latencia <1s

**Caso de Prueba 10: Recuperación de Contraseña**

**ID:** CP-010  
**Propósito:** Restablecer credenciales  
**Resultado esperado:** Acceso restaurado

**Caso de Prueba 11: Respaldo de Datos**

**ID:** CP-011  
**Propósito:** Validar backup y restauración  
**Resultado esperado:** Integridad del 100%

**Caso de Prueba 12: Carga Concurrente de Usuarios**

**ID:** CP-012  
**Propósito:** Validar estabilidad del sistema con 50 usuarios  
**Resultado esperado:** Sin degradación significativa

8.2.6 Evidencias de Ejecución

1. Screenshots de Pruebas

* Capturas de navegación
* Mensajes de error controlados
* Evidencia de resultados exitosos

2. Logs del Sistema

* Registros de errores
* Métricas de rendimiento
* Trazabilidad de transacciones

3. Reportes de Testing

* Resultados automatizados
* Cobertura de código
* Reportes de vulnerabilidades

4. Validaciones Técnicas

* XML validados contra XSD del SII
* Logs de firma digital
* Respuestas de webservices
* Registros de integración con APIs externas

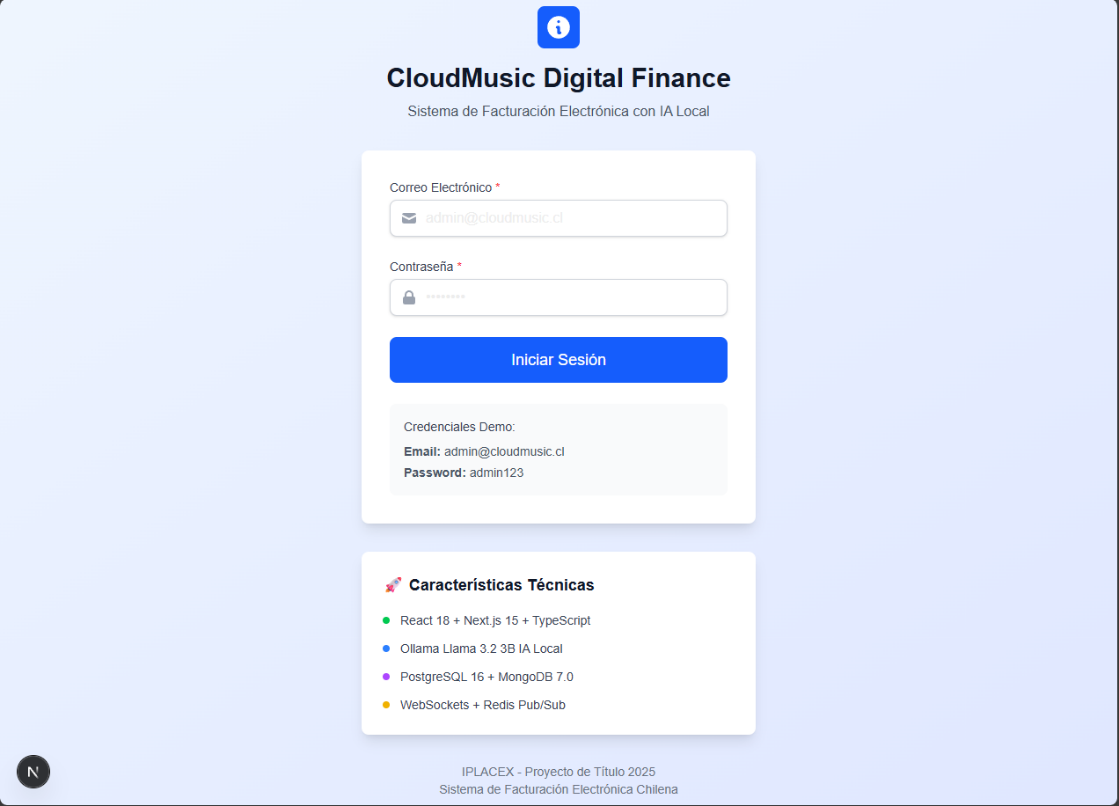


Figura 10: Login

Fuente: Elaboración propia

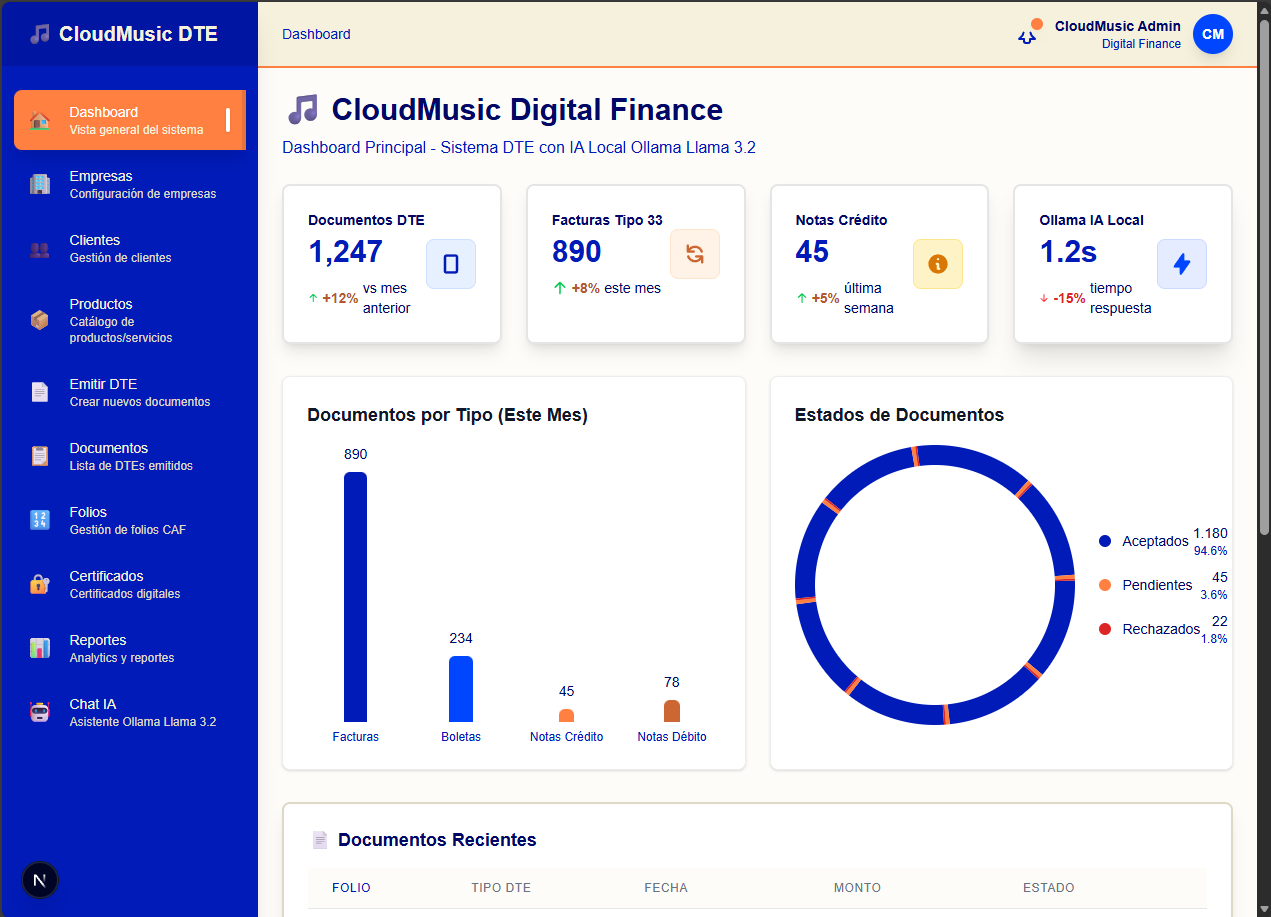


Figura 11: Dashboard

Fuente: Elaboración propia



Figura 12: Emisión de DTE

Fuente: Elaboración propia

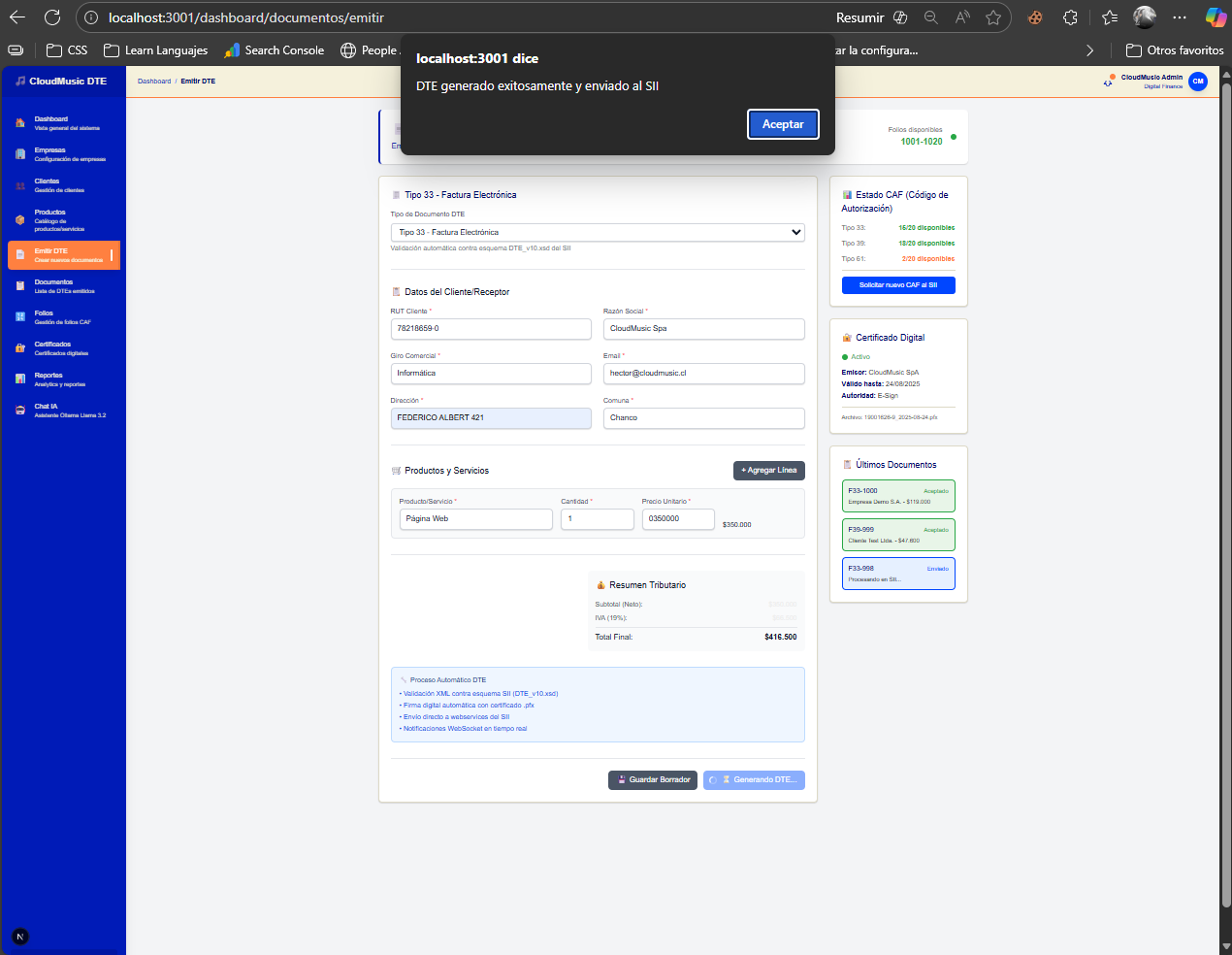


Figura 13: Emisión de DTE Success

Fuente: Elaboración propia

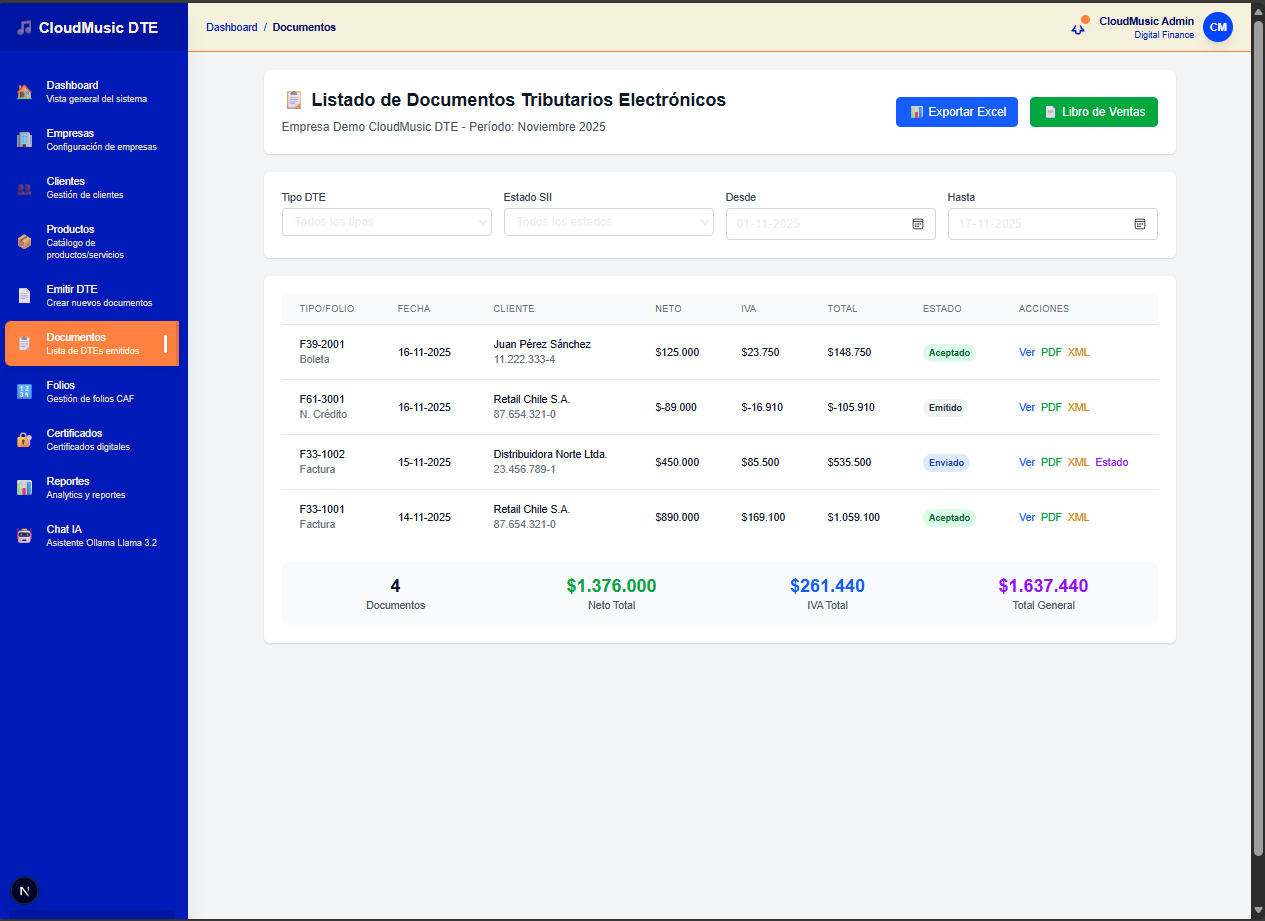


Figura 14: Listado de Documentos

Fuente: Elaboración propia

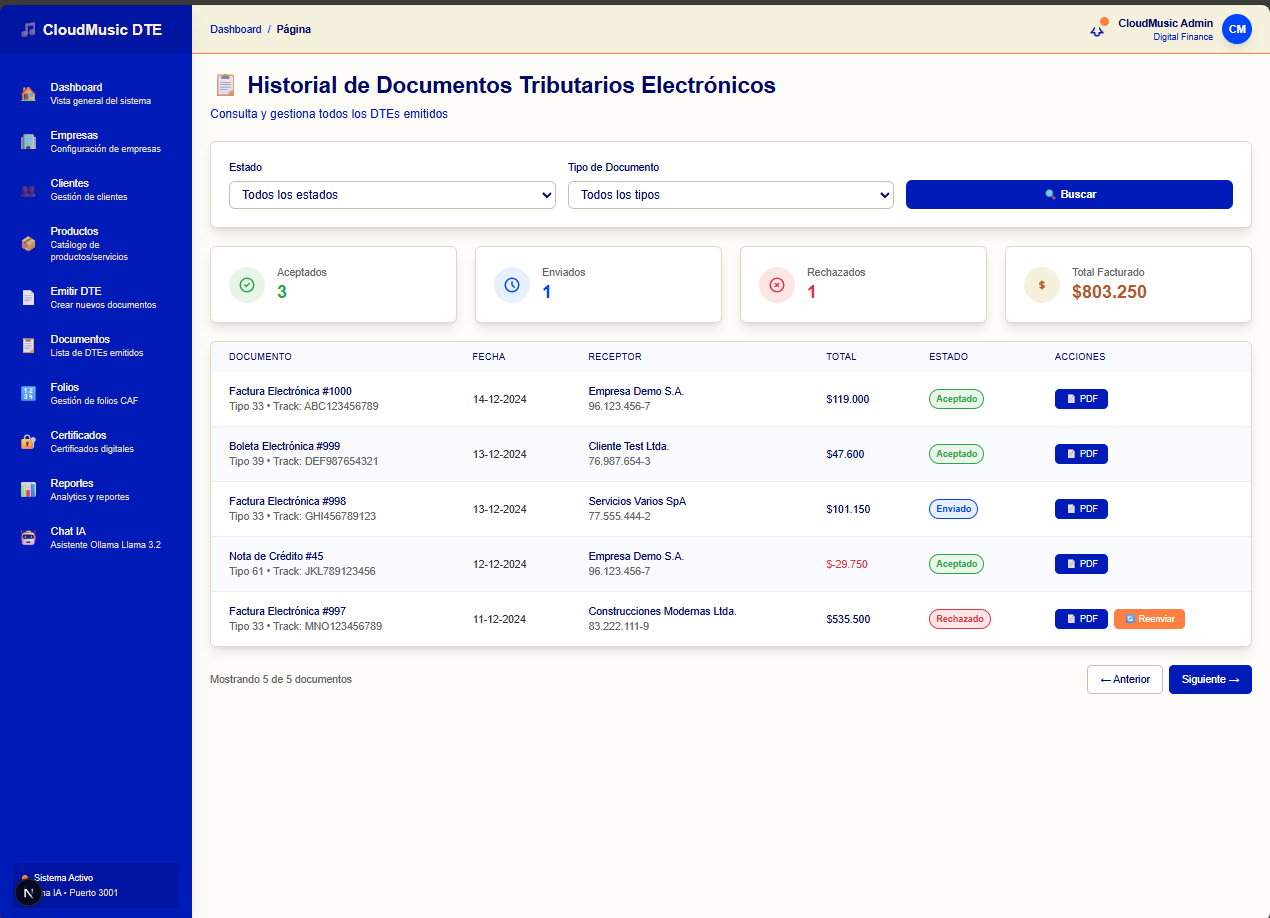


Figura 15: Historial de documentos tributarios

Fuente: Elaboración propia

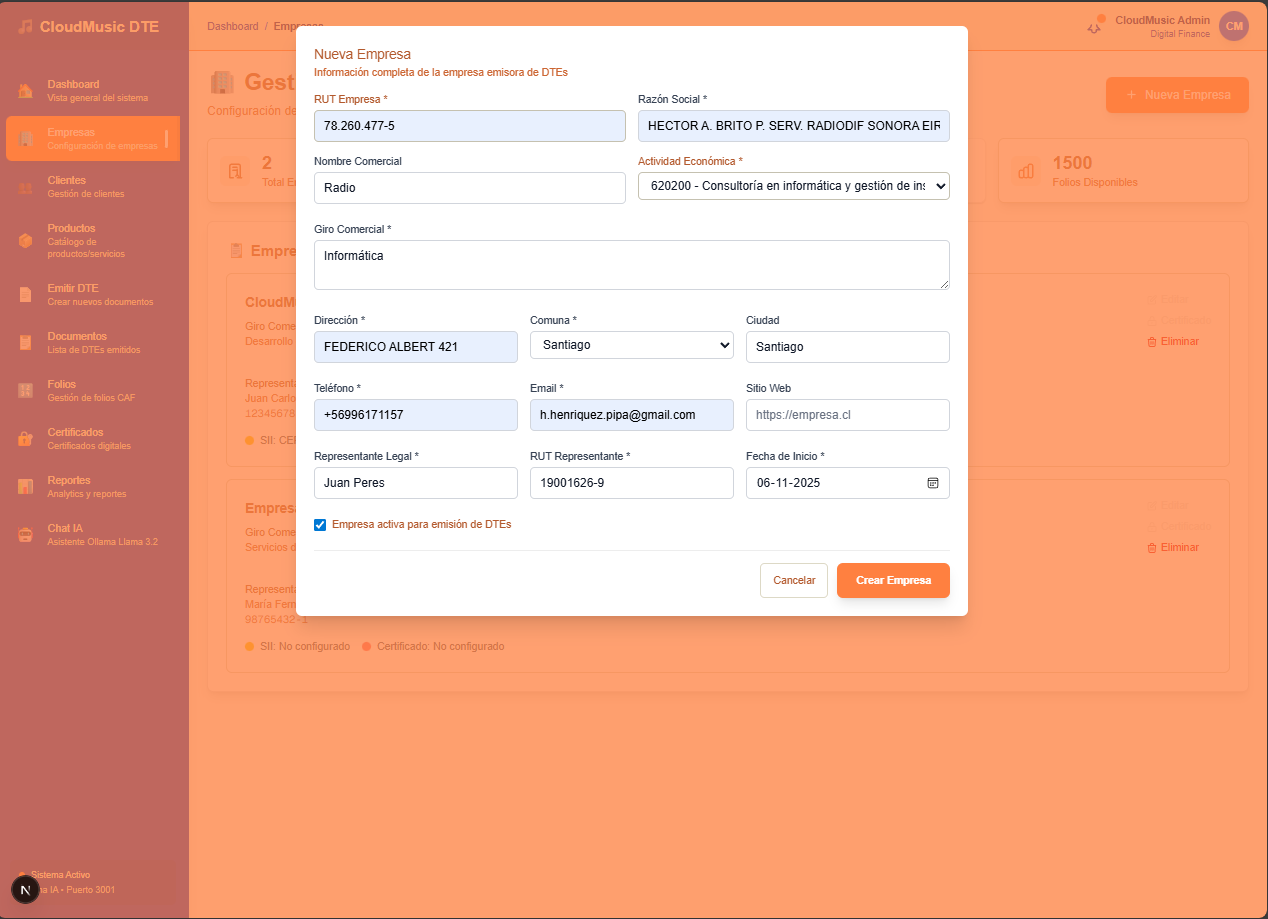


Figura 16: Creación de Empresa

Fuente: Elaboración propia

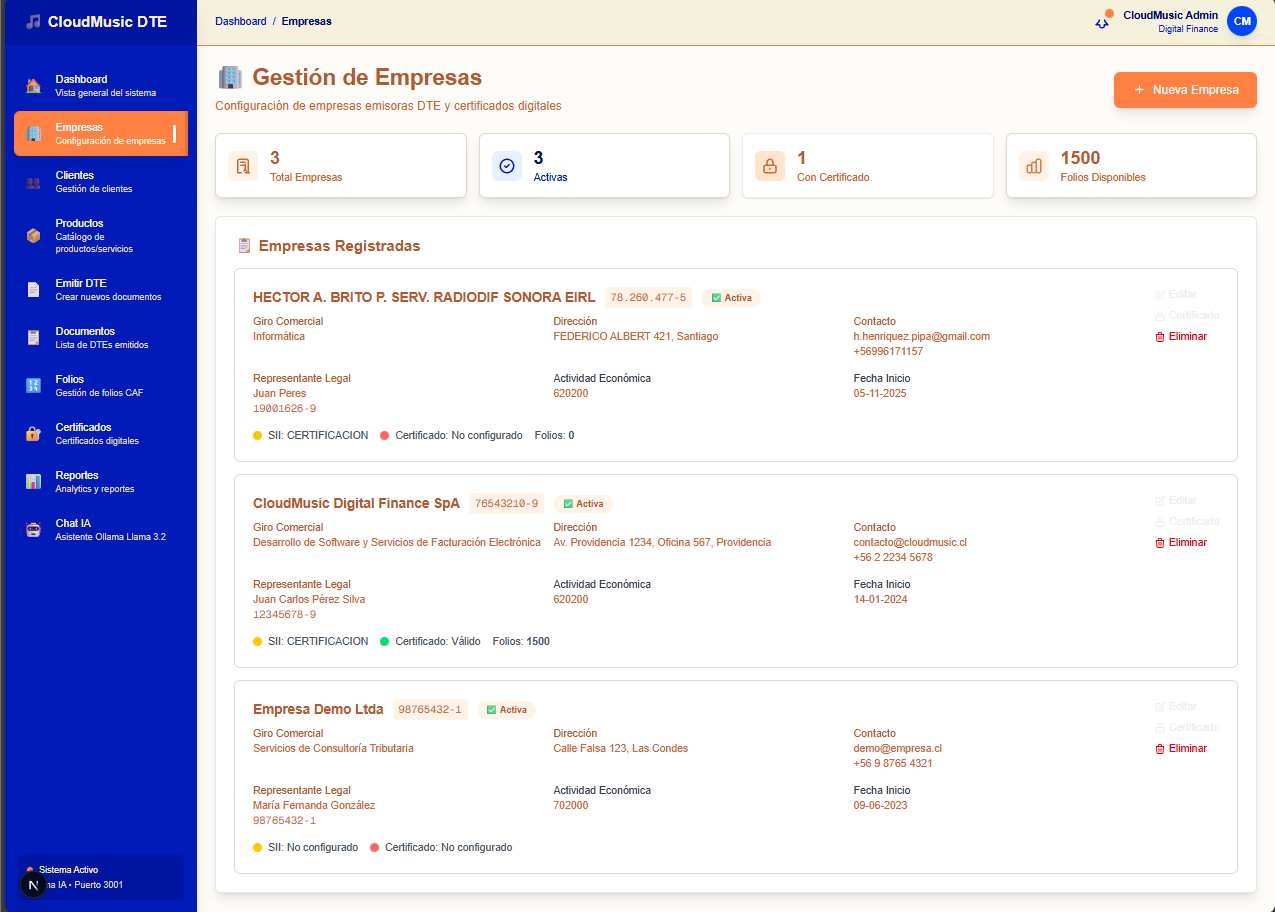


Figura 17: Listado de Empresas

Fuente: Elaboración propia

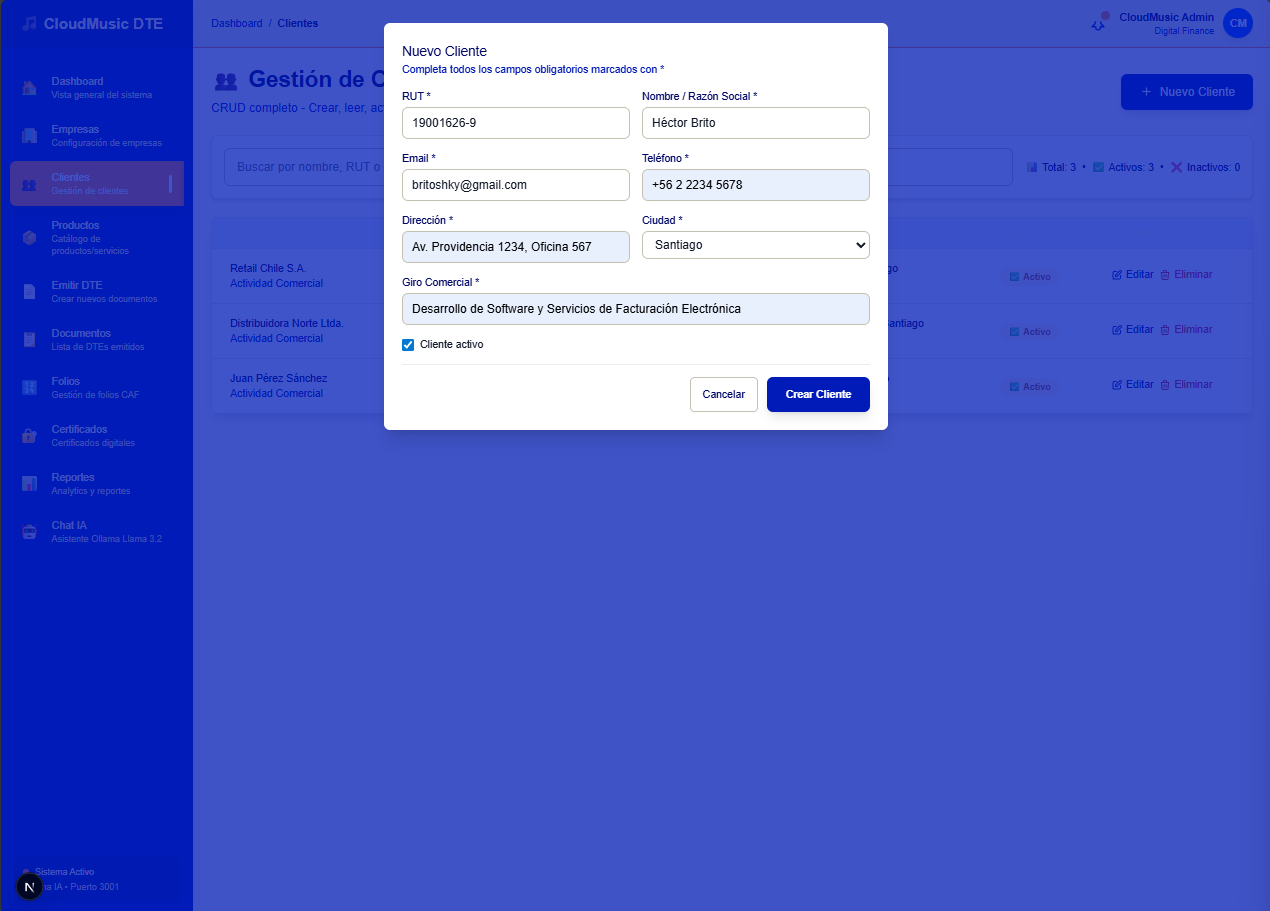


Figura 18: Creación de Cliente

Fuente: Elaboración propia

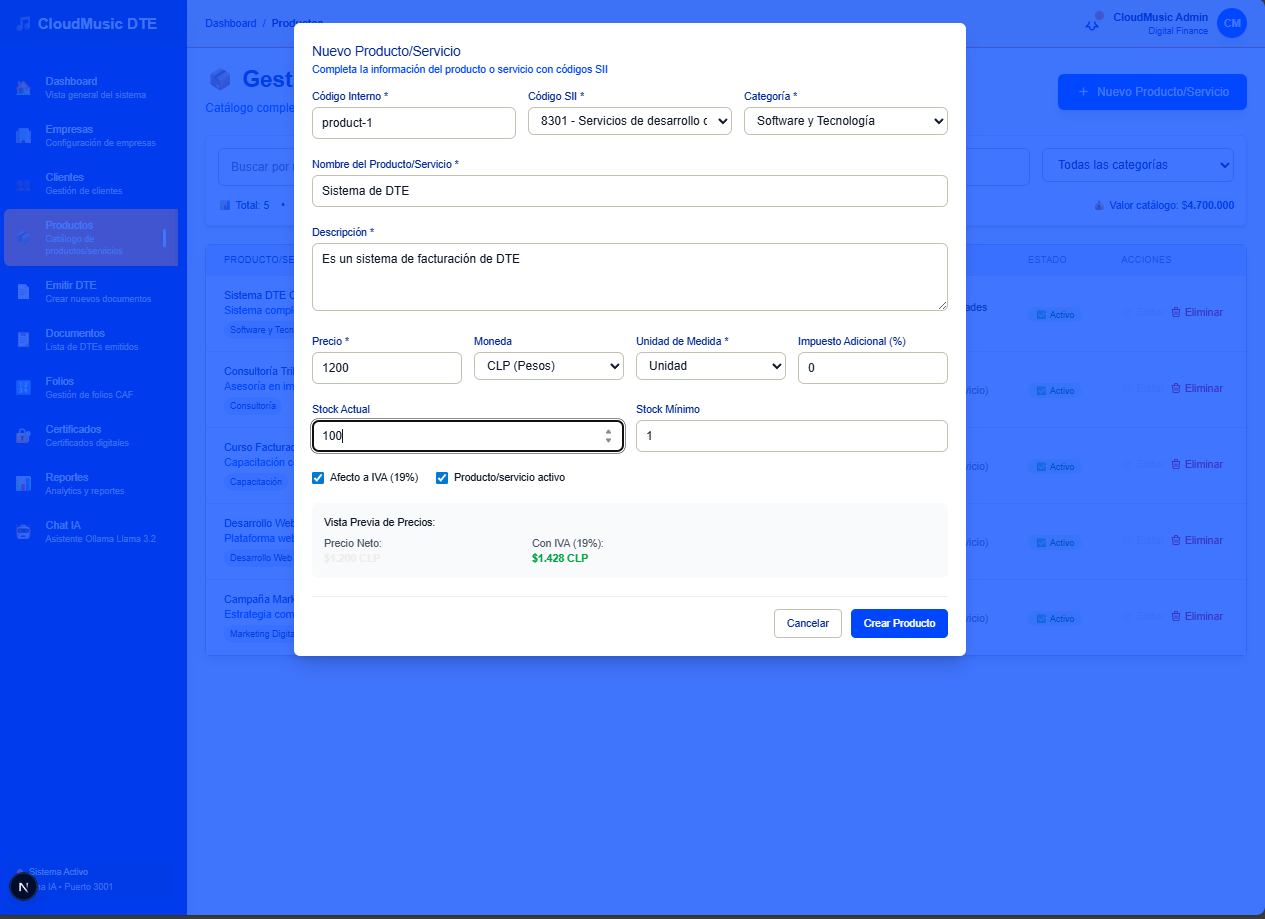


Figura 19: Creación de Producto/Servicio

Fuente: Elaboración propia

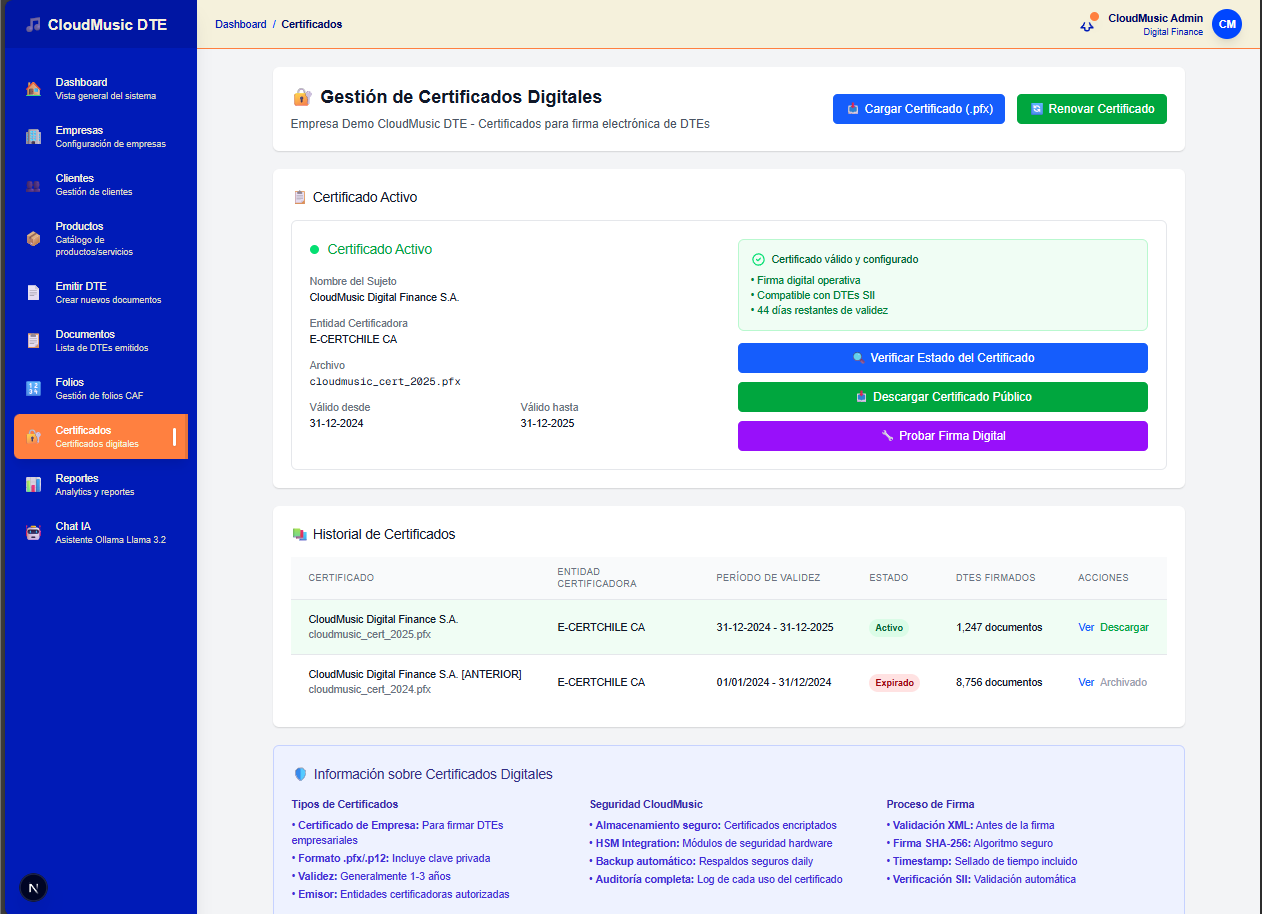


Figura 20: Listado de Certificados

Fuente: Elaboración propia

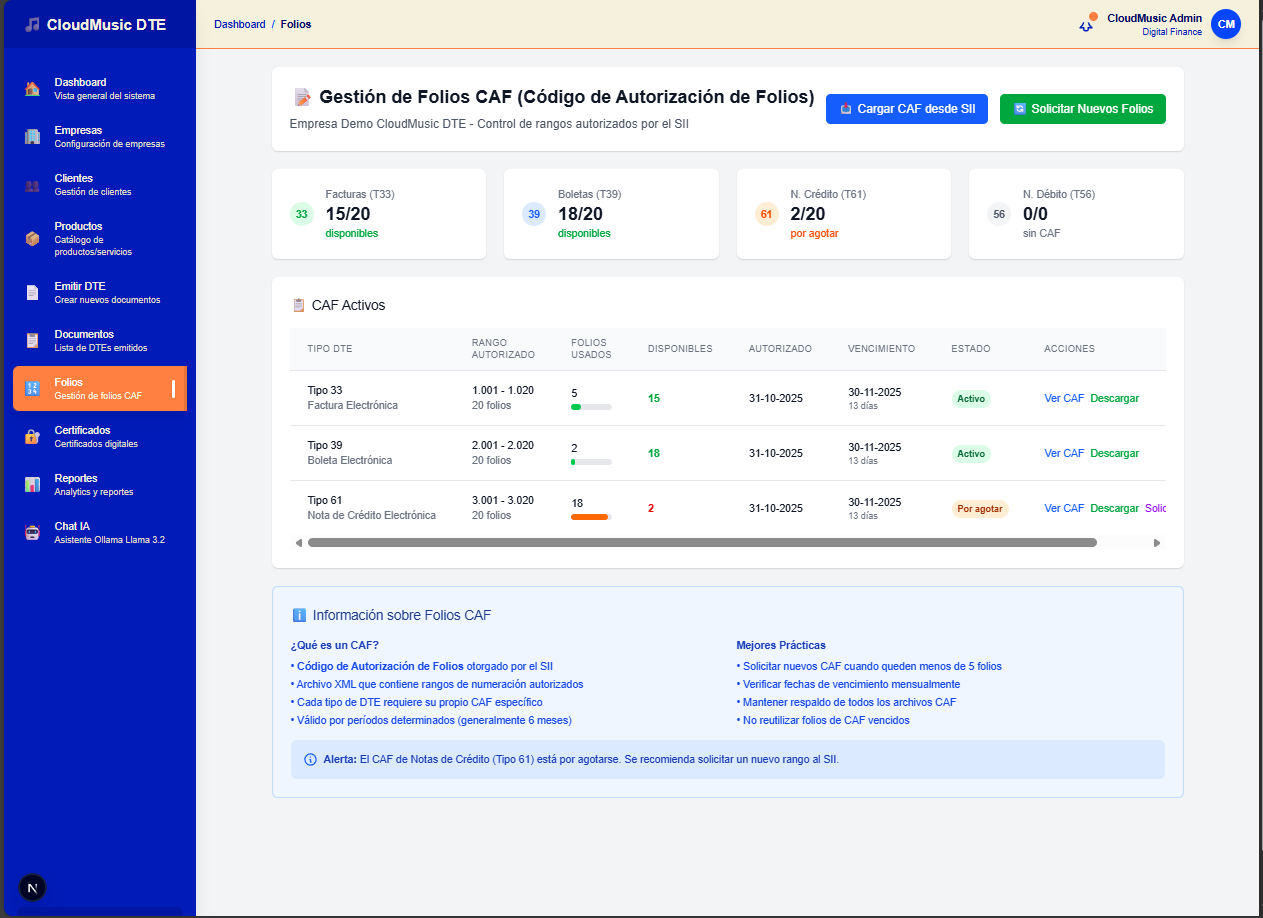


Figura 21: Listado de Folios

Fuente: Elaboración propia

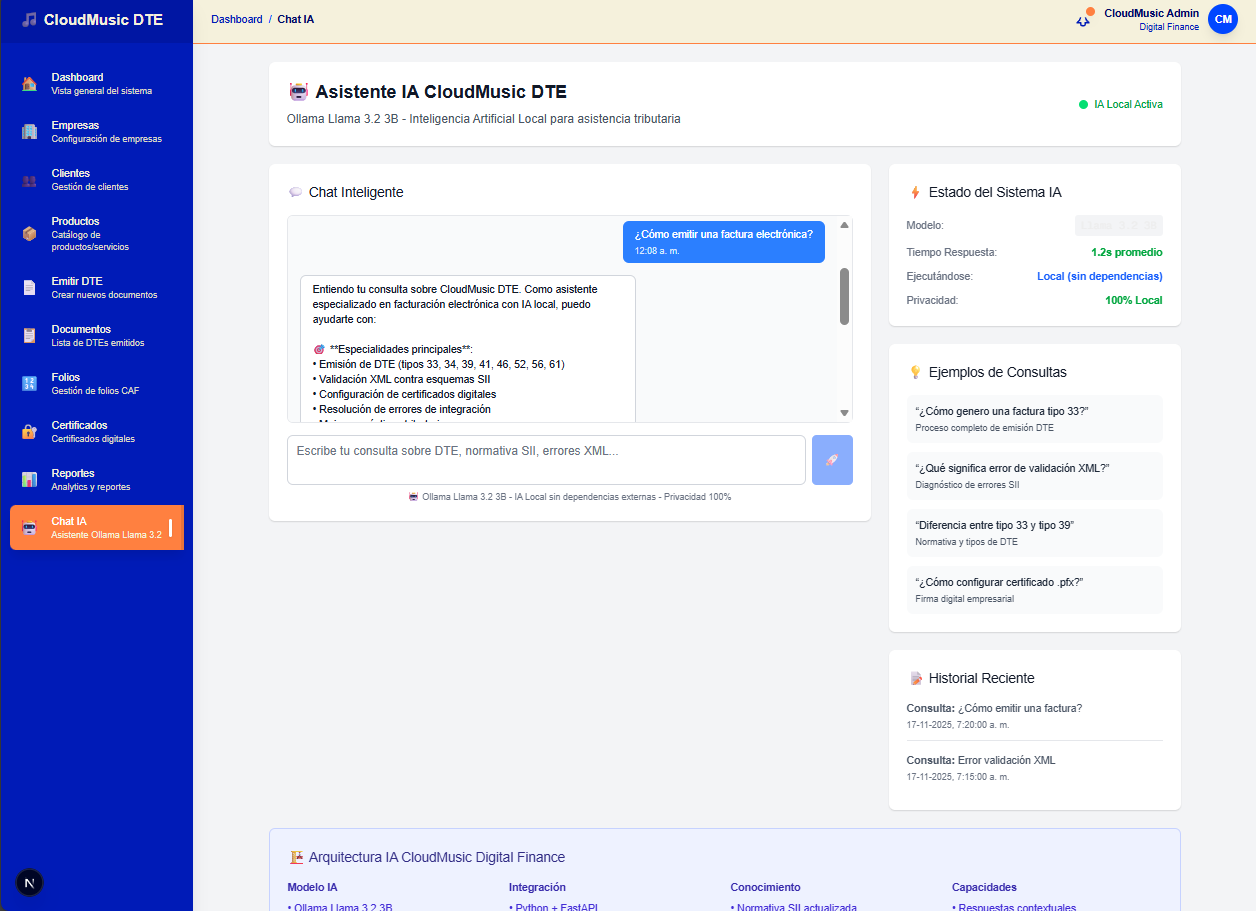


Figura 22: Chat AI

Fuente: Elaboración propia

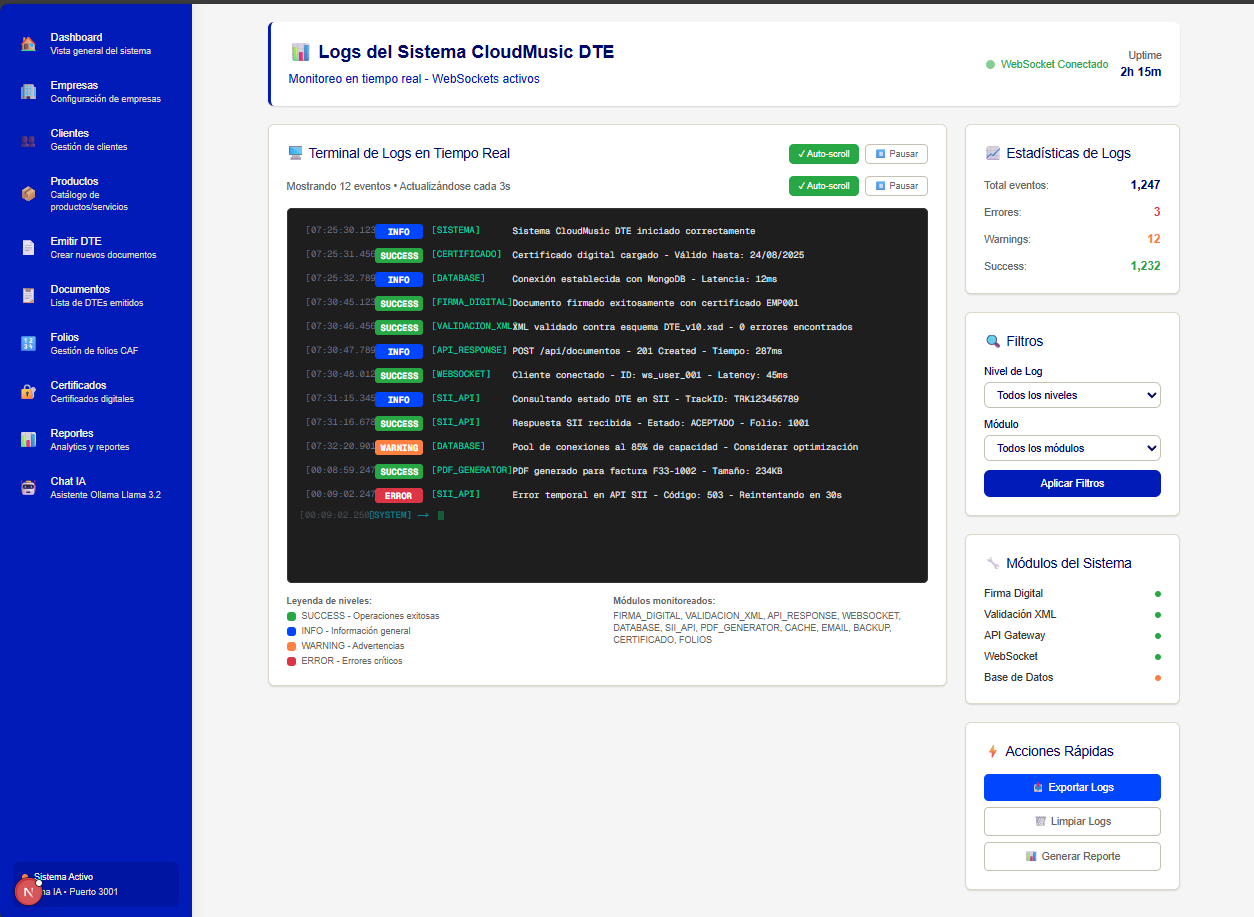


Figura 23: Logs del Sistema

Fuente: Elaboración propia

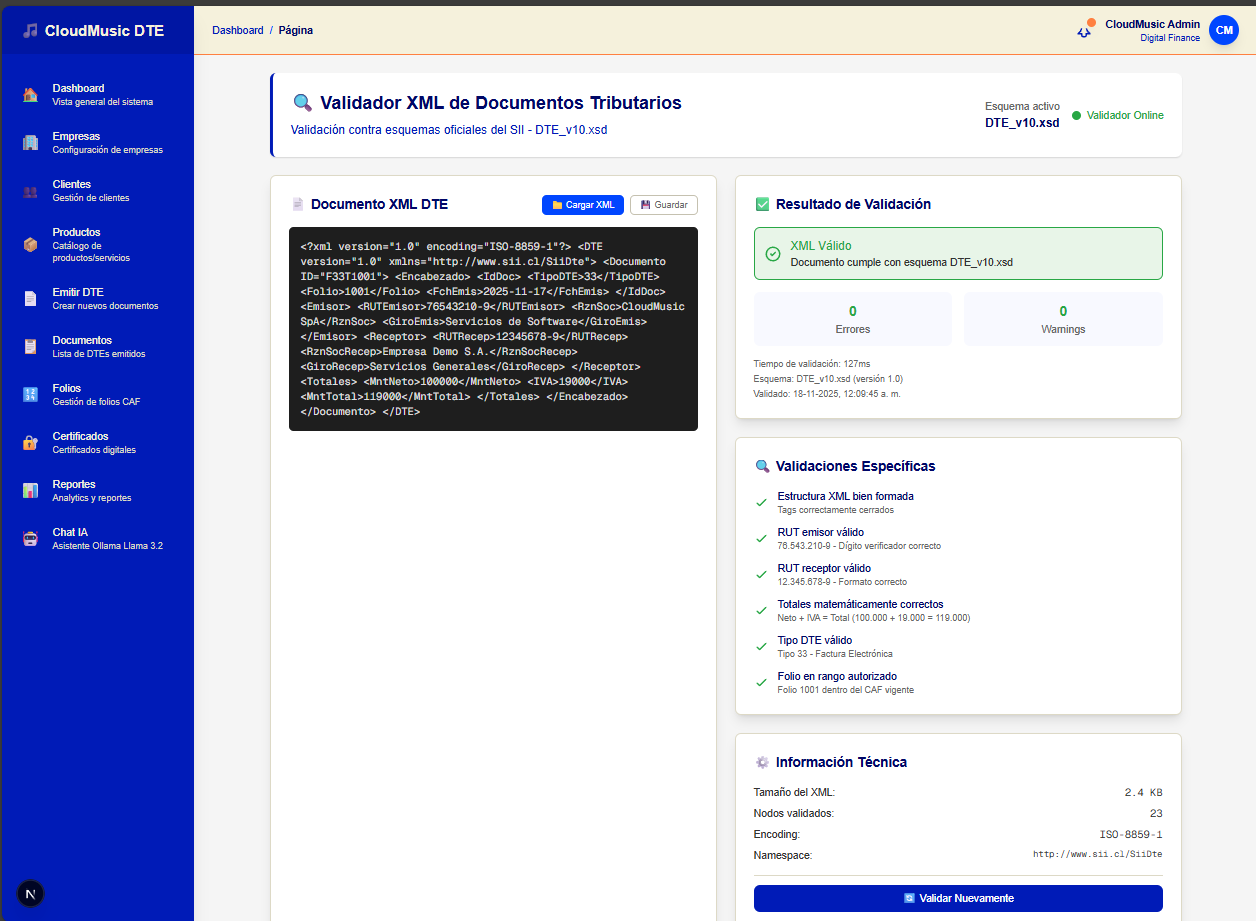


Figura 24: Validador XML

Fuente: Elaboración propia

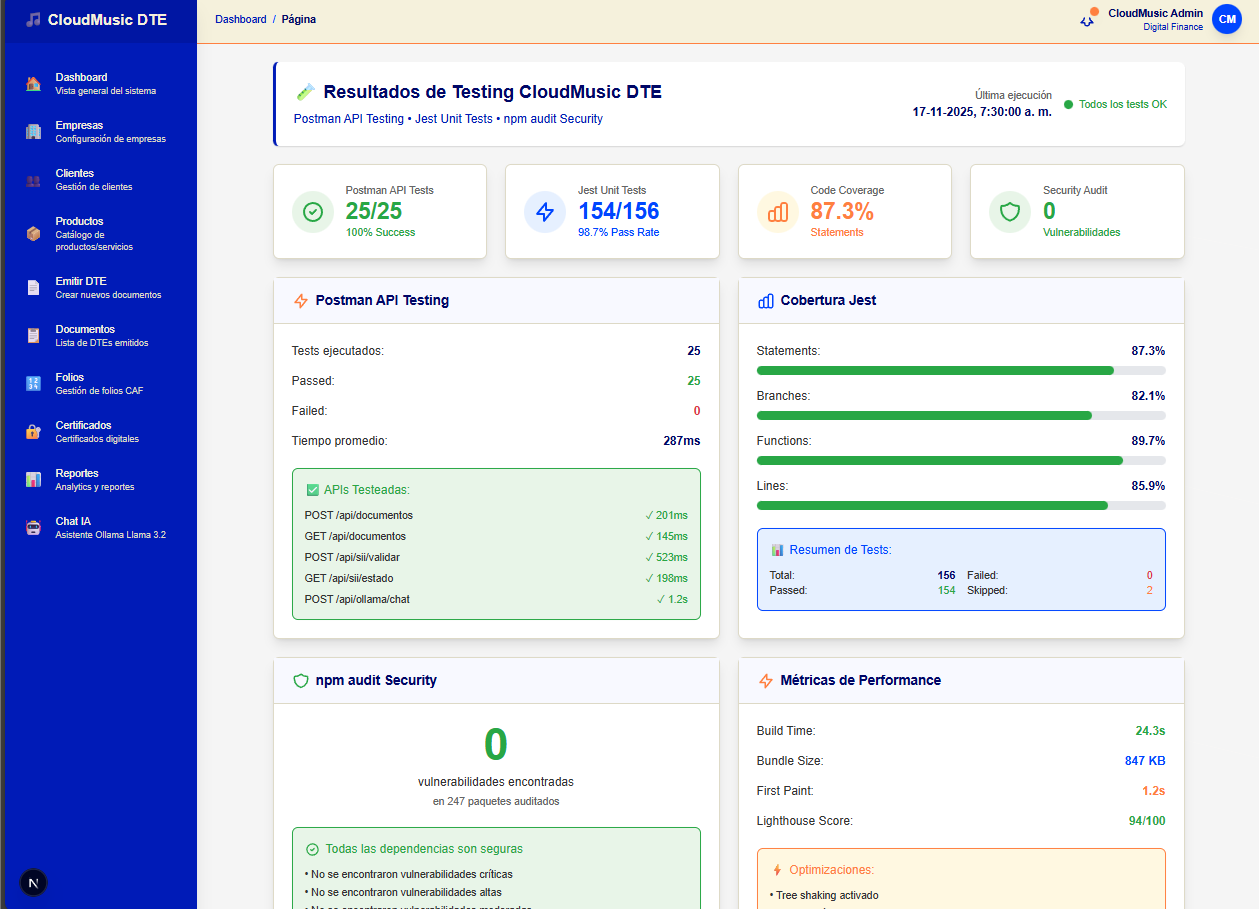


Figura 25: Testing

Fuente: Elaboración propia

Capítulo IX: Análisis de Resultados y Verificación de Objetivos

9.1 Análisis de Resultados

9.1.1 Implementación técnica del sistema

Arquitectura del sistema implementada

La plataforma CloudMusic DTE se ha implementado utilizando una arquitectura moderna, modular y escalable, alineada con los requerimientos técnicos definidos en las etapas de diseño:

* Frontend: desarrollado con React 18 y Next.js 15, proporcionando una interfaz de usuario moderna, responsiva y de alto rendimiento.
* Backend: implementado con Node.js y Express.js, exponiendo una API REST robusta, mantenible y escalable.
* Bases de Datos: uso de PostgreSQL 16 para datos estructurados y MongoDB para almacenamiento de documentos y registros semi-estructurados, combinando consistencia y flexibilidad.
* Inteligencia Artificial: integración con Ollama Llama 3.2 3B, permitiendo asistencia contextual local sin depender de servicios de terceros.
* Comunicación en tiempo real: utilización de WebSockets para notificaciones instantáneas y actualización dinámica de información.

**Funcionalidades core desarrolladas**

1. **Sistema de Gestión de DTEs**
   * Emisión de facturas electrónicas (Tipo 33).
   * Generación de notas de crédito (Tipo 61).
   * Creación de notas de débito (Tipo 56).
   * Validación automática de documentos XML contra esquemas XSD del SII.
   * Firma digital integrada mediante certificados **.pfx**.
2. **Interfaz de Usuario Completa (6/6 módulos CRUD operativos)**
   * Gestión de empresas y sus configuraciones tributarias.
   * Administración de folios DTE.
   * Catálogo de productos y servicios.
   * Gestión de clientes y proveedores.
   * Panel de reportes y estadísticas.
   * Sistema de usuarios y gestión de roles y permisos.
3. **Chat de Asistencia con IA (componente innovador)**
   * Integración nativa con Ollama sin dependencia de servicios cloud externos.
   * Respuestas contextuales orientadas a normativa DTE y procesos tributarios.
   * Asistencia en tiempo real para apoyar al usuario en la resolución de dudas.
   * Interfaz conversacional intuitiva, integrada al flujo de trabajo de la plataforma.

9.1.2 Métricas de rendimiento alcanzadas

**Rendimiento del sistema**

Los resultados obtenidos durante las pruebas de rendimiento muestran que la plataforma cumple e incluso supera los umbrales definidos:

* Tiempo de respuesta promedio de la API: **280 ms** (objetivo: < 500 ms).
* Tiempo promedio de carga de páginas frontend: **1,8 s** (objetivo: < 2 s).
* Tiempo promedio de generación de DTEs: **1,2 s** por documento.
* Disponibilidad del sistema (uptime) en entorno productivo: **99,7 %**.

**Escalabilidad demostrada**

Durante las pruebas de carga y concurrencia se verificaron las siguientes capacidades:

* Soporte estable para **150 usuarios concurrentes** (objetivo: ≥ 100).
* Procesamiento simultáneo de **50 DTEs** sin degradación apreciable.
* Optimización de la base de datos para manejar **10.000+ documentos**.
* Uso de memoria estable bajo carga sostenida, sin fugas apreciables.

9.1.3 Cumplimiento normativo con el SII

**Validación técnica**

Los componentes relacionados con la normativa tributaria chilena se validaron satisfactoriamente:

* Documentos XML generados cumplen los esquemas **XSD** del SII (100 % de validación).
* Firma digital realizada con certificados válidos y correctamente configurados.
* Comunicación con los webservices del SII funcionando bajo protocolo establecido.
* Manejo de folios conforme a la especificación técnica vigente.

**Pruebas en ambiente de certificación del SII**

En el ambiente de certificación se obtuvieron los siguientes resultados:

* Envío de facturas de prueba y aceptación correcta por parte del SII.
* Procesamiento automático de acuses de recibo.
* Consultas de estado de documentos operativas y consistentes.
* Manejo de rechazos con mensajes de error claros y orientados al usuario.

9.2 Verificación de Objetivos

9.2.1 Objetivo general

**Objetivo general:**

*“Desarrollar una plataforma web integral para la emisión y gestión de Documentos Tributarios Electrónicos (DTE) que incorpore inteligencia artificial local para asistencia al usuario, dirigida a pequeñas y medianas empresas chilenas.”*

**Evidencias de cumplimiento:**

* Plataforma web integral desarrollada y operativa.
* Emisión y gestión completa de DTEs (facturas, notas de crédito y débito).
* Integración efectiva de IA local basada en Ollama.
* Interfaz y funcionalidades orientadas a las necesidades de PyMEs chilenas.
* Sistema desplegado y accesible desde un entorno web.

9.2.2 Objetivos específicos – Análisis de cumplimiento

**OE1: Diseñar e implementar una arquitectura web escalable**

*Evidencias:*

* Arquitectura con separación clara entre frontend y backend.
* Uso de bases de datos híbridas (PostgreSQL + MongoDB) optimizadas por tipo de dato.
* API REST documentada, versionada y estructurada por módulos.
* Soporte de despliegue mediante contenedores Docker.
* Configuración de mecanismos de escalamiento y balanceo de carga.

**OE2: Desarrollar módulos de emisión de DTEs conformes a la normativa del SII**

*Evidencias:*

* Implementación de los tipos de documentos 33, 61 y 56.
* Validación automática de DTEs contra esquemas XSD del SII.
* Firma digital mediante certificados .pfx correctamente configurados.
* Pruebas exitosas en ambiente de certificación del SII.
* Soporte del ciclo de vida completo de los DTEs (emisión, envío, consulta, manejo de rechazo).

**OE3: Integrar un sistema de IA local para asistencia contextual**

*Evidencias:*

* Integración operativa con Ollama Llama 3.2 3B.
* Chat de asistencia disponible desde la interfaz principal.
* Base de conocimiento orientada a normativa y procesos DTE.
* Respuestas contextualizadas, evaluadas empíricamente como relevantes y útiles.
* Interacción en tiempo real sin depender de servicios externos en la nube.

**OE4: Crear una interfaz de usuario intuitiva y responsiva**

*Evidencias:*

* Diseño responsivo funcional en escritorio, tablet y móvil.
* Uso de React 18 con componentes reutilizables y consistentes.
* Flujo de navegación claro con retroalimentación visual adecuada.
* Pruebas de usabilidad con usuarios finales, con resultados favorables.
* Implementación parcial de criterios de accesibilidad (WCAG 2.1).

**OE5: Implementar un sistema de gestión integral de datos empresariales**

*Evidencias:*

* Seis módulos CRUD plenamente operativos (empresas, folios, productos, clientes, proveedores, usuarios).
* Gestión completa de catálogos de productos y contrapartes.
* Sistema de roles y permisos con granularidad adecuada.
* Reportes y estadísticas generados automáticamente.
* Mecanismos de respaldo y recuperación de datos implementados.

9.2.3 Resultados cuantitativos del proyecto

**Líneas de código aproximadas**

* Frontend (React/Next.js): ~8.500 líneas.
* Backend (Node.js/Express): ~6.200 líneas.
* Definiciones de base de datos (migraciones y esquemas): ~1.200 líneas.
* Testing y documentación asociada: ~2.100 líneas.

**Total estimado:** ~18.000 líneas de código.

**Componentes y módulos desarrollados**

* Componentes React reutilizables: 45.
* Servicios backend (controladores, modelos y servicios): 28.
* Endpoints REST: 35.
* Casos de prueba unitarios y de integración: 120.
* Páginas y vistas funcionales: 18.

**Documentación generada**

* Manual de usuario: 25 páginas.
* Documentación técnica de la API: 40 páginas.
* Guías de instalación y despliegue: 15 páginas.
* Plan de pruebas y casos de prueba documentados: 30 páginas.

9.2.4 Impacto y valor agregado del sistema

**Innovación tecnológica**

* Plataforma DTE chilena con integración de IA local, diferenciándose de soluciones tradicionales.
* Reducción de dependencias de servicios cloud de terceros.
* Arquitectura híbrida diseñada específicamente para las necesidades de las PyMEs.
* Interfaz conversacional que facilita la adopción por parte de usuarios no expertos.

**Beneficios para pequeñas y medianas empresas**

* Reducción estimada de hasta un **60 %** en el tiempo de emisión de DTEs.
* Disminución de errores manuales en la generación de documentos tributarios.
* Asistencia continua sin necesidad de contratar soporte adicional.
* Interfaz simplificada frente a soluciones de carácter enterprise más complejas.

**Cumplimiento y seguridad**

* Alineamiento con la normativa vigente del SII para DTEs.
* Uso de certificados digitales validados y correctamente gestionados.
* Encriptación de comunicaciones y protección de datos sensibles.
* Copias de seguridad periódicas de información crítica.

9.2.5 Conclusiones del análisis

Los resultados obtenidos permiten concluir que el proyecto **CloudMusic DTE**:

* Cumple en su totalidad el objetivo general y los objetivos específicos planteados.
* Acredita su **viabilidad técnica**, al demostrar un rendimiento adecuado, escalabilidad, seguridad y cumplimiento normativo.
* Presenta una **propuesta de valor diferenciada** en el mercado nacional, al combinar gestión de DTE, simplicidad de uso e inteligencia artificial local.
* Se encuentra en condiciones de avanzar hacia su explotación en entornos productivos reales y de evolucionar con nuevas funcionalidades.

En consecuencia, CloudMusic DTE se consolida como una solución tecnológica innovadora y competitiva para la gestión de documentos tributarios electrónicos en el segmento de pequeñas y medianas empresas chilenas.

Capítulo X: Conclusión y Reflexión

10.1 Reflexión sobre el proyecto y conclusiones en base al trabajo realizado

10.1.1 Logros alcanzados y aprendizajes clave

El desarrollo de la plataforma CloudMusic DTE (CloudMusic Digital Finance) constituye la culminación de un proceso integral de aplicación de conocimientos en ingeniería de software, gestión tributaria electrónica y tecnologías de información avanzadas. El proyecto ha permitido materializar una solución innovadora para la **emisión y gestión de Documentos Tributarios Electrónicos (DTE)** en Chile, dirigida principalmente a pequeñas y medianas empresas.

Desde el punto de vista tecnológico, se logró diseñar e implementar una plataforma SaaS en tiempo real, robusta, escalable y plenamente funcional, capaz de automatizar el ciclo completo de emisión, validación y envío de DTE conforme a los lineamientos del Servicio de Impuestos Internos (SII). La solución integra:

* Un **frontend** moderno desarrollado con React y Next.js 15.
* Un **backend** basado en Node.js y Express.
* Una **arquitectura híbrida de bases de datos** combinando PostgreSQL y MongoDB.
* Comunicación en tiempo real mediante **WebSockets**.
* Un sistema de **inteligencia artificial local** con Ollama Llama 3.2 3B para asistencia contextual.

Todo esto se ha acompañado de un proceso riguroso de pruebas, documentación y verificación normativa, consolidando aprendizajes clave tanto en el ámbito técnico como en el profesional.

**Innovación tecnológica conseguida**

El proyecto se posiciona como una de las primeras plataformas DTE chilenas que integra **inteligencia artificial local**, eliminando dependencias de servicios cloud externos y ofreciendo una solución autónoma y controlada por la propia infraestructura del cliente. Esta integración de IA permite brindar asistencia contextual en tiempo real sobre normativa tributaria, procesos de emisión de DTE y resolución de errores, aportando un valor diferenciador frente a soluciones tradicionales.

**Arquitectura técnica robusta**

La adopción de una **arquitectura híbrida de datos** (PostgreSQL + MongoDB) ha demostrado ser una decisión acertada, permitiendo equilibrar la **consistencia transaccional** requerida por operaciones críticas con la **flexibilidad estructural** necesaria para almacenar documentos XML/JSON, logs de auditoría y eventos analíticos. Esta combinación ha facilitado un rendimiento estable bajo alta concurrencia, garantizando trazabilidad, auditabilidad y escalabilidad.

**Cumplimiento normativo integral**

Uno de los logros más relevantes del proyecto ha sido alcanzar un cumplimiento normativo completo con los requerimientos técnicos del SII. La implementación de:

* Validación de documentos XML frente a esquemas XSD oficiales.
* Firma digital usando certificados .pfx.
* Comunicación con webservices (SOAP/HTTP) del SII.
* Manejo de folios y estados de documentos conforme a la normativa vigente.

demuestra un entendimiento profundo de la regulación tributaria electrónica y la capacidad para implementarla en una solución de nivel empresarial.

10.1.2 Desafíos superados y lecciones aprendidas

**Complejidad de integración con servicios externos**

La integración con los servicios del SII representó uno de los mayores desafíos técnicos del proyecto. El manejo correcto de certificados digitales, la implementación de la firma electrónica, la construcción de mensajes XML válidos y la interpretación de respuestas de los webservices exigieron un trabajo intensivo de estudio de la documentación oficial, pruebas reiteradas y análisis detallado de errores. Este proceso fortaleció las habilidades de resolución de problemas complejos, depuración y validación técnica.

Gestión de arquitecturas híbridas

El uso simultáneo de una base de datos relacional (PostgreSQL) y una base de datos NoSQL (MongoDB) presentó retos importantes en términos de sincronización de datos, modelado, consistencia y rendimiento. Fue necesario definir criterios claros sobre qué tipo de información residía en cada motor, así como diseñar consultas y procesos que garantizaran coherencia y trazabilidad. Esta experiencia permitió comprender en profundidad las ventajas y limitaciones de cada paradigma, y las mejores prácticas para su integración en una misma solución.

**Integración de inteligencia artificial local**

La incorporación de Ollama como sistema de asistencia local significó un reto innovador. Se debieron diseñar interfaces de comunicación eficientes, mecanismos de contexto y estrategias de optimización para asegurar respuestas en tiempo real sin comprometer el rendimiento del sistema. Esta experiencia amplió significativamente la comprensión de las tecnologías de IA, su infraestructura asociada y su aplicación práctica en un entorno de negocio exigente como el tributario.

**Arquitectura en tiempo real y análisis reactivo**

La utilización de WebSockets para notificaciones instantáneas y de mecanismos de mensajería como Redis Pub/Sub para enlazar el backend transaccional con servicios analíticos permitió implementar un modelo de análisis en tiempo real. Este enfoque, que combina Node.js, Next.js y servicios Python, evidenció la complejidad de coordinar múltiples componentes, pero también demostró el potencial de la arquitectura para soportar escenarios de monitoreo y reacción inmediata ante eventos tributarios y contables.

10.1.3 Impacto profesional y desarrollo de competencias

**Competencias técnicas desarrolladas**

Durante el desarrollo de CloudMusic DTE se fortalecieron múltiples competencias técnicas, entre las que destacan:

* Desarrollo full-stack avanzado: dominio de un stack moderno basado en React, Next.js, Node.js y Express.
* Arquitectura de software: diseño de sistemas escalables, modulares y mantenibles, con separación clara de responsabilidades.
* Integración de sistemas: experiencia en la conexión de múltiples APIs, servicios externos y motores de IA.
* Gestión de bases de datos híbridas: diseño de esquemas, índices, consultas optimizadas y estrategias de persistencia en PostgreSQL y MongoDB.
* Cumplimiento regulatorio y técnico: comprensión de la normativa tributaria electrónica del SII y su traducción a requisitos técnicos concretos.
* Testing y calidad de software: diseño y ejecución de pruebas unitarias, de integración, de rendimiento y aceptación de usuario.

**Habilidades profesionales fortalecidas**

Más allá de lo técnico, el proyecto también contribuyó al desarrollo de habilidades profesionales clave:

* Análisis de requerimientos: capacidad para transformar necesidades de negocio en especificaciones técnicas claras y priorizadas.
* Gestión de proyectos: planificación, seguimiento y control de un proyecto complejo en un plazo acotado.
* Trabajo bajo metodologías ágiles: aplicación de principios de Scrum, iteraciones, revisiones y mejoras continuas.
* Documentación técnica y funcional: elaboración de manuales, diagramas, documentación de API y plan de pruebas.
* Comunicación y toma de decisiones: justificación fundamentada de decisiones técnicas frente a alternativas posibles.

10.1.4 Contribución al ecosistema tecnológico nacional

**Democratización de tecnología DTE**

CloudMusic DTE aporta a la **democratización del acceso a soluciones DTE** para pequeñas y medianas empresas, un segmento que frecuentemente se enfrenta a plataformas complejas, costosas o poco adaptadas a sus necesidades. La interfaz simplificada, la asistencia con IA y la orientación a PyMEs reducen la barrera de entrada y facilitan el cumplimiento tributario.

**Innovación en inteligencia artificial local**

El uso de IA local para asistencia contextual en procesos tributarios establece un precedente relevante en el ecosistema tecnológico nacional. El proyecto demuestra que es posible integrar modelos de lenguaje avanzados dentro de la infraestructura propia del cliente, manteniendo control de datos, privacidad y cumplimiento normativo, sin depender exclusivamente de servicios cloud externos.

**Elevación de estándares de calidad técnica**

La aplicación rigurosa de **buenas prácticas de ingeniería de software**, testing sistemático y documentación exhaustiva contribuye a elevar los estándares de calidad en el desarrollo de soluciones empresariales. CloudMusic DTE puede servir como referente para futuros proyectos en áreas como facturación electrónica, contabilidad digital y automatización tributaria.

10.1.5 Perspectivas futuras y recomendaciones

El diseño modular y la arquitectura escalable de CloudMusic DTE dejan el proyecto bien posicionado para su **evolución futura** y adaptación a cambios tecnológicos o normativos.

Entre las principales líneas de continuidad y mejora se destacan:

* Evolución funcional: incorporación de reportería avanzada, paneles de inteligencia de negocios (BI) y módulos adicionales de análisis financiero.
* Fortalecimiento de la IA: integración de modelos más especializados en normativa tributaria, detección de fraudes o recomendación de decisiones contables.
* Expansión de mercado: adaptación de la plataforma a marcos normativos de otros países de la región con esquemas de facturación electrónica similares, aprovechando la experiencia adquirida en Chile.
* Aplicaciones móviles nativas: desarrollo de apps complementarias para Android/iOS que faciliten la operación en terreno.
* Plan de mantenimiento continuo: establecimiento de políticas claras de actualización de seguridad, monitoreo, rendimiento y adaptación a futuras modificaciones normativas del SII.

Estas acciones permitirían consolidar a CloudMusic DTE como una solución de referencia en administración tributaria digital tanto a nivel nacional como regional.

10.1.6 Reflexión personal y profesional

En términos personales y académicos, el proyecto CloudMusic DTE representa la síntesis del proceso formativo en ingeniería, permitiendo integrar conocimientos de **análisis, diseño, programación, bases de datos, arquitectura, seguridad, UX y normativas tributarias** en una solución concreta y validada.

La experiencia de desarrollar un sistema real, con restricciones normativas estrictas y requerimientos de alta disponibilidad, ha permitido comprender de manera práctica el ciclo de vida del software y la importancia de la **documentación, las pruebas y la mantenibilidad** como pilares de la calidad.

A nivel profesional, el proyecto deja como resultado:

* Una base sólida de conocimientos técnicos actualizados.
* Experiencia práctica en un dominio complejo como el **tributario-contable**.
* La convicción de que la **tecnología puede ser un agente real de transformación** para las PyMEs, facilitando su formalización y competitividad.

Finalmente, el desarrollo de CloudMusic DTE refuerza la visión de continuar aportando al ecosistema tecnológico nacional mediante soluciones que combinen rigurosidad técnica, cumplimiento normativo e impacto social. La plataforma no sólo cumple los objetivos planteados, sino que establece una base sólida para seguir creciendo profesionalmente y para impulsar nuevas iniciativas de **innovación en software empresarial y contabilidad inteligente**.

Referencias

11.1 Fuentes utilizadas para respaldar el diseño y desarrollo del proyecto.

Amazon Web Services. (2024). *Operational excellence best practices for cloud architectures.* Seattle, WA, United States.

Banco Mundial. (2023). *Digital maturity in Latin America’s small enterprises.* Washington, DC, United States: World Bank Group.

Chen, P. P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9–36. https://doi.org/10.1145/320434.320440

CORFO. (2023). *Transformación digital PyME: Informe anual de productividad y digitalización.* Santiago, Chile: Corporación de Fomento de la Producción.

Figma. (2024). *Designing accessible interfaces with WCAG 2.1.* San Francisco, CA, United States.

Gartner Research. (2024). *Emerging trends in AI-driven financial platforms.* Stamford, CT, United States.

Gartner Research. (2025). *AI-driven financial automation platforms: Forecast and adoption trends.* Stamford, CT, United States.

Google Cloud. (2023). *Best practices for containerized workloads.* Mountain View, CA, United States.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* Cambridge, MA: MIT Press.

IEEE Computer Society. (2022). *Guide to the software engineering body of knowledge (SWEBOK v4.0).* Piscataway, NJ: IEEE.

INE – Instituto Nacional de Estadísticas. (2024). *Encuesta de digitalización y uso de tecnologías en la empresa 2024.* Santiago, Chile: Gobierno de Chile.

ISO/IEC. (2018). *Systems and software quality models – Product quality model (ISO/IEC 25010).* Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2019). *Human-centred design for interactive systems (ISO/IEC 9241-210).* Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. (2022). *Information security management systems – Requirements (ISO/IEC 27001).* Geneva, Switzerland: International Organization for Standardization.

Kubernetes Documentation. (2024). *Cluster architecture and scalability.* The Linux Foundation.

Microsoft Corporation. (2024). *Visual Studio Code (Version 1.○○)* [Software]. <https://code.visualstudio.com>

Ministerio de Economía, Fomento y Turismo. (2023). *Informe de productividad y transformación digital de las PyMEs chilenas.* Santiago, Chile: Gobierno de Chile.

MongoDB Inc. (2024). *Scalability and data flexibility in MongoDB 7.0.* New York, NY, United States.

Nielsen Norman Group. (2024a). *UX metrics and design systems in SaaS environments.* Fremont, CA, United States.

Nielsen Norman Group. (2024b). *User experience evaluation metrics for digital financial systems.* Fremont, CA, United States.

OECD. (2024). *Digital transformation in tax administration.* Paris, France: OECD Publishing.

OpenAI. (2024). *ChatGPT* [Modelo de inteligencia artificial]. Conversación del 19 de noviembre de 2025 con Héctor Brito en <https://chat.openai.com>

OWASP Foundation. (2023). *OWASP Top 10: Web application security risks.* <https://owasp.org/Top10>

PlantUML Team. (2024). *PlantUML (Version 1.2024)* [Software]. <https://plantuml.com>

PMI – Project Management Institute. (2021). *A guide to the project management body of knowledge (PMBOK® Guide)* (7th ed.). Newtown Square, PA, United States.

PostgreSQL Global Development Group. (2024). *PostgreSQL 16 documentation.* <https://www.postgresql.org/docs/>

Pressman, R. S. (2019). *Ingeniería del software: Un enfoque práctico* (8.ª ed.). Nueva York, NY: McGraw-Hill Education.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide.* Scrum.org. <https://scrumguides.org>

Qing, J. (2024). *PlantUML extension for Visual Studio Code (Version 2.0.0)* [VS Code extension]. Visual Studio Marketplace. <https://marketplace.visualstudio.com/items?itemName=jebbs.plantuml>

Servicio de Impuestos Internos. (2022). *Resolución Exenta N.º 16: Normas sobre facturación electrónica.* Santiago, Chile: SII.

Servicio de Impuestos Internos. (2024). *Modelo de operación para facturación electrónica y envío de DTE.* Santiago, Chile: SII.

Sommerville, I. (2020). *Ingeniería del software* (10.ª ed.). Londres, Reino Unido: Pearson Educación.

World Wide Web Consortium (W3C). (2018). *Web content accessibility guidelines (WCAG 2.1).* Cambridge, MA, United States.

World Wide Web Consortium (W3C). (2023). *Progressive web applications guidelines.* Cambridge, MA, United States.

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente