

ThesisFlow: Plataforma Web para la Gestión y Visualización de Trabajos Finales en el DCIC

Ignacio Joaquín Dotta

Departamento de Ciencias e Ingeniería de la Computación (DCIC)

Universidad Nacional del Sur, Argentina

Directores: Dr. Martín Larrea y Dra. María Luján Gánuza

Codirector: Lic. Diego Sebastián Orbe Leiva

{ignacio.dotta, mlg, mll, diego.orbe}@cs.uns.edu.ar

Abstract. Este trabajo presenta *ThesisFlow*, una plataforma web que centraliza la gestión y visualización de Trabajos Finales y Tesis de Grado del DCIC. El sistema integra en una única solución los flujos administrativos de carga y curado de información, un módulo de autogestión para docentes y un portal público de analíticas interactivas. Se describen la motivación y el contexto institucional, la exploración de soluciones y alternativas, los objetivos del proyecto y los aspectos centrales de la implementación a nivel de arquitectura, modelo de datos e interfaces. Finalmente, se discuten usos concretos de la aplicación, conclusiones sobre su impacto y líneas de trabajo futuro.

Keywords: Sistemas de información académica · Visualización de datos · Ingeniería de software · Aplicaciones web

1 Motivación

La gestión de Trabajos Finales y Tesis de Grado en el Departamento de Ciencias e Ingeniería de la Computación (DCIC) involucra a secretaría, docentes, estudiantes y autoridades. Históricamente, la información asociada a estos proyectos se distribuyó en planillas, documentos aislados y actas del Consejo Departamental, lo que dificulta:

- la trazabilidad del ciclo de vida de cada trabajo;
- el análisis de tendencias temáticas y líneas de dirección;
- la obtención de estadísticas confiables para la planificación académica;
- la transparencia institucional respecto de la producción del departamento.

Estado del arte

Tradicionalmente, al preparar la producción académica anual se revisan manualmente las actas aprobadas durante el período correspondiente, obteniendo principalmente conteos agregados de Trabajos Finales. En años recientes se comenzó a recopilar información más detallada (título, fechas, docentes, temática, alumnos), pero sin un sistema que permita explotarla de manera estructurada ni ofrecer vistas unificadas a los distintos actores.

Perspectiva de los estudiantes

Desde la óptica de los alumnos, el inicio del Trabajo Final suele estar acompañado de incertidumbre. Son frecuentes situaciones como:

- no tener claro qué tipo de producto o alcance se espera de un Trabajo Final;
- contar con una idea preliminar pero no saber si es viable o qué docente podría dirigirla;
- desconocer qué temas se han trabajado recientemente en el departamento.

Una plataforma integral que brinde acceso a proyectos históricos, sus temas y docentes asociados permite:

- explorar Trabajos Finales previos y su documentación para inspirarse;
- identificar tendencias temáticas a lo largo del tiempo;
- partir de un tema de interés y localizar qué docentes trabajan en esa área;
- partir de un docente y explorar los temas en los que ha dirigido trabajos.

En este contexto, *ThesisFlow* surge como una herramienta para unificar la gestión, facilitar el acceso a la información y habilitar analíticas que apoyen la toma de decisiones de estudiantes y autoridades.

2 Exploración de soluciones

Previo al diseño de la plataforma se revisaron soluciones existentes vinculadas a la gestión y difusión de Trabajos Finales y Tesis. Entre ellas se consideraron:

- Repositorio UBA: <https://repositoriouba.sisbi.uba.ar/gsdl/cgi-bin/library.cgi>
- Repositorio Digital de la UNS: <https://repositoriodigital.uns.edu.ar/>

Estos repositorios permiten almacenar y consultar trabajos completos, junto con metadatos básicos sobre autores, directores y fecha. Sin embargo:

- no incorporan flujos administrativos específicos del DCIC (por ejemplo, carga desde actas del Consejo);
- no brindan analíticas interactivas sobre temas, carreras o redes de colaboración;
- no ofrecen vistas diferenciadas para secretaría, docentes y estudiantes.

También se analizaron sistemas de gestión académica y plataformas de e-learning con módulos de Trabajos Finales. En general, se observó que cubren parcialmente el problema, con escasa capacidad de adaptación al dominio local y limitadas herramientas de visualización de datos.

3 Alternativas

A partir de la exploración previa se identificaron tres alternativas principales:

1. Mantener el esquema actual

La primera opción consistía en continuar utilizando planillas compartidas y documentos dispersos. Esta alternativa se descartó debido a:

- dificultades para garantizar integridad y uniformidad de los datos;
- escasa capacidad de visualización avanzada y analíticas complejas;
- poca escalabilidad a medida que crece el volumen de Trabajos Finales;
- limitaciones para extender funcionalidades o incorporar nuevos actores.

2. Adoptar un sistema externo

La segunda alternativa era adaptar un sistema existente (repositorio institucional, sistema académico o plataforma de terceros). Si bien podría reducir esfuerzo inicial, presenta desventajas:

- menor flexibilidad para adaptar el modelo de datos al DCIC;
- dependencia de un proveedor externo para la evolución del sistema;
- capacidades analíticas y de personalización acotadas.

3. Desarrollar un sistema propio

La tercera opción, finalmente adoptada, es desarrollar una solución propia, modular y extensible. Esta alternativa supone un mayor esfuerzo inicial, pero ofrece:

- control total sobre el modelo de datos y las reglas de negocio;
- optimización específica para consultas y analíticas relevantes;
- un diseño preparado para extenderse a nuevas carreras o módulos (por ejemplo, posgrado);
- posibilidad de incorporar integración progresiva con otros sistemas institucionales.

4 Especificación de objetivos

4.1 Objetivo general

Diseñar e implementar una plataforma web integral para la gestión, el seguimiento y la visualización de Trabajos Finales y Tesis de Grado del DCIC, que sirva como fuente confiable de información para secretaría, docentes y comunidad académica.

4.2 Objetivos específicos

- Centralizar la información de Trabajos Finales en una base de datos estructurada y consistente.
- Proveer interfaces diferenciadas para secretaría, docentes y público general.
- Ofrecer analíticas interactivas sobre la producción del departamento.
- Incorporar mecanismos de autenticación y control de acceso acordes a cada perfil.
- Facilitar tareas de respaldo, restauración y auditoría de datos.

5 Implementación

5.1 Arquitectura general

La arquitectura de *ThesisFlow* adopta una estructura de tres capas: frontend, backend y base de datos. La Figura 1 ilustra los componentes principales y sus interacciones.

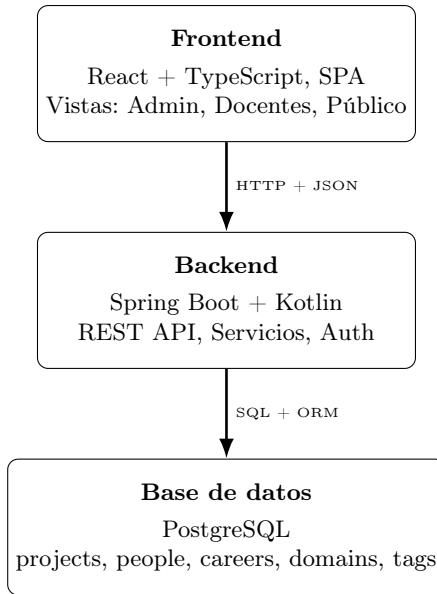


Fig. 1: Arquitectura general de *ThesisFlow*: tres capas principales y sus responsabilidades.

El backend expone endpoints REST para operaciones administrativas, vistas públicas y analíticas. Los servicios encapsulan la lógica de negocio (gestión de proyectos, personas, carreras, importación y respaldos). El frontend, implementado como SPA, organiza vistas por rol y utiliza componentes reutilizables para formularios, tablas y dashboards.

5.2 Modelo de datos y servicios

El modelo de datos incluye entidades como `Project`, `Person`, `Career`, `Domain` y `Tag`, junto con tablas de unión para representar relaciones muchos-a-muchos (por ejemplo, proyectos–docentes y proyectos–etiquetas). Internamente se utilizan claves enteras, mientras que hacia el exterior se exponen UUIDs para preservar estabilidad y privacidad en los identificadores.

Los servicios de negocio abstraen el acceso a datos y coordinan operaciones que involucran múltiples entidades, tales como la creación de un nuevo proyecto con sus relaciones, la importación de lotes desde archivos CSV o la construcción de agregados para analíticas.

5.3 Decisiones de diseño y proceso de desarrollo

El desarrollo de *ThesisFlow* involucró una serie de decisiones arquitectónicas y metodológicas que respondieron tanto a los requisitos funcionales como a las necesidades operativas del DCIC. A continuación se detallan algunos de los aspectos más relevantes.

a) Flujo de autenticación mediante magic links. Para facilitar el acceso de los docentes sin requerir usuarios y contraseñas tradicionales, se implementó un mecanismo de acceso basado en *magic links*. El flujo consiste en:

- El profesor solicita acceso proporcionando su correo institucional.
- El sistema genera un enlace único asociado a un token efímero (no JWT) y lo envía al correo correspondiente. Este correo debe coincidir con el registrado previamente por secretaría al cargar proyectos dirigidos por ese docente.
- Al acceder desde el enlace, el backend valida el token y emite un JWT estándar, que el frontend utiliza para autenticar las sesiones posteriores del docente.

Este mecanismo evita la gestión de contraseñas, reduce fricción para los usuarios y mantiene un esquema seguro basado en la posesión del correo institucional.

b) Elección de un modelo estructurado en SQL. Aunque la estructura jerárquica de un Trabajo Final podría modelarse en una base NoSQL, se optó por PostgreSQL debido a:

- la necesidad de garantizar consistencia y normalización de datos;
- la presencia de relaciones complejas entre entidades (proyectos, docentes, estudiantes, carreras, etiquetas);
- la importancia de realizar estadísticas precisas, agregaciones y filtros sobre datos altamente estructurados.

Un modelo NoSQL hubiera simplificado algunos aspectos del dominio, pero hubiera dificultado mantener consistencia cuando cambian datos compartidos, como la información de un docente asociado a múltiples proyectos.

c) *Proceso de desarrollo.* El desarrollo se organizó en etapas sucesivas:

- Reuniones iniciales para análisis de requerimientos y definición del alcance general.
- Elaboración de un documento PRD (*Product Requirements Document*) con requerimientos funcionales detallados.
- Priorización en tres hitos principales:
 1. Modelo de datos, CRUDs administrativos y autenticación de secretaría.
 2. Acceso público, visualizaciones interactivas y motor de analíticas.
 3. Funcionalidades específicas para docentes, incluyendo vistas personalizadas y *magic links*.
- Creación de prototipos de interfaz en Figma para validar flujos y orientar el diseño visual.
- Diseño del modelo entidad–relación (ER) y transición a la implementación incremental.

d) *Uso experimental de herramientas de inteligencia artificial.* Durante el desarrollo se integraron herramientas de IA generativa (Claude Haiku 4.5 y Codex/ChatGPT 5.0) para acelerar tareas repetitivas. Se observó:

- Alta utilidad en generación de componentes de frontend, CRUDs, pruebas automáticas y depuración de errores típicos de librerías.
- Utilidad moderada en tareas de refactorización o generación de código boilerplate.
- Limitaciones marcadas en tareas complejas como diseño del servicio de analíticas, donde la IA tendió a producir errores conceptuales o resultados inconsistentes.

En conclusión, la IA se demostró altamente eficaz para acelerar un proyecto de escala media cuando se aplica a tareas bien definidas y supervisadas, reduciendo significativamente tiempos de desarrollo.

5.4 Despliegue e infraestructura

En el entorno actual, el frontend de *ThesisFlow* se despliega en Vercel, que realiza el proceso de construcción utilizando la configuración de Vite.js y sirve los archivos estáticos resultantes desde su red de distribución de contenido. El backend y la base de datos PostgreSQL se alojan en Render, donde el servicio backend se construye y ejecuta a partir de un *Dockerfile*, lo que permite reproducir localmente el mismo entorno de ejecución que en producción.

Como línea de mejora, se considera deseable que a futuro la plataforma sea alojada en la infraestructura del propio DCIC, bajo un dominio institucional del tipo `*.cs.uns.edu.ar`, de modo de consolidar el control operativo, la soberanía de los datos y la integración con otros servicios de la Universidad.

5.5 Carga inicial de datos

La carga inicial de datos constituyó una etapa fundamental para poner en funcionamiento la plataforma con un conjunto completo y confiable de Trabajos Finales históricos. El proceso se estructuró en tres fases principales: normalización del dataset original, preparación estructurada de los archivos y carga mediante la funcionalidad de importación masiva del sistema.

Normalización y limpieza del dataset. El punto de partida fue un conjunto heterogéneo de fuentes: planillas históricas, documentos administrativos y actas del Consejo Departamental. Estos datos presentaban variaciones en nombres de docentes, formatos de fechas, diferencias tipográficas y estructuras inconsistentes. Para unificar estos registros, se desarrolló un script específico (documentado en el repositorio del proyecto) que:

- estandarizó nombres de docentes y estudiantes mediante reglas de coincidencia aproximada y diccionarios de equivalencias;
- unificó formatos de fechas (aprobación, defensa, publicación);
- asignó carreras y dominios respetando taxonomías actuales del DCIC;
- detectó y resolvió duplicados mediante heurísticas basadas en título + año + docente;
- generó UUIDs estables para cada proyecto y persona.

Este proceso garantizó un dataset coherente y apto para ser cargado en un modelo relacional.

Preparación del archivo de importación. Una vez normalizado el dataset, el script generó un archivo CSV con columnas alineadas al modelo de datos interno: proyecto, fechas, docentes, carrera, etiquetas temáticas y dominios. El sistema comprueba automáticamente la validez del archivo, la consistencia de las columnas y el tipo de datos antes de continuar.

Importación masiva en la plataforma. Finalmente, la carga se realizó utilizando la funcionalidad de “Importar proyectos”, que:

- procesa cada fila como una operación atómica (validación → creación del proyecto → creación de relaciones);
- informa errores fila por fila sin detener el proceso completo;
- genera un reporte detallado con estadísticas (creados, actualizados, rechazados);
- asegura que las entidades relacionadas (docentes, etiquetas, dominios) se creen o reutilicen según corresponda.

Gracias a este proceso, el sistema quedó poblado con la totalidad del historial disponible, permitiendo que las visualizaciones y analíticas fueran significativas desde el primer despliegue.

5.6 Flujo de creación de proyectos

El flujo de creación de un proyecto refleja el proceso que sigue la secretaría para registrar un Trabajo Final aprobado. La Figura 2 muestra un diagrama de secuencia simplificado.

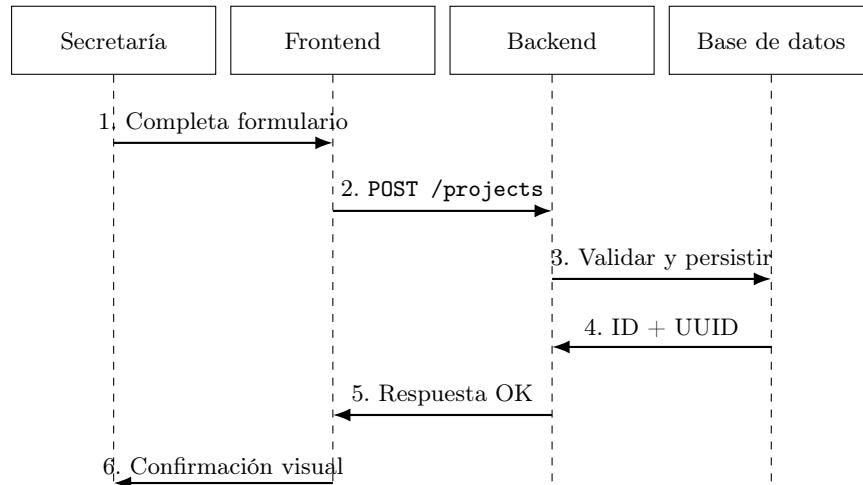


Fig. 2: Flujo simplificado de creación de un proyecto en *ThesisFlow*.

Este flujo asegura que los datos ingresados se validen en el backend, que se creen las relaciones necesarias en la base de datos y que el nuevo proyecto quede disponible inmediatamente para las interfaces de consulta y analíticas.

6 Usos de la aplicación

ThesisFlow se utiliza en distintos escenarios por actores con necesidades diversas:

- **Secretaría:** registra Trabajos Finales aprobados, actualiza metadatos y genera listados consolidados.
- **Docentes:** consultan los proyectos en los que participan como directores o co-directores y verifican la información publicada.
- **Público general:** explora la producción del departamento mediante filtros y visualizaciones interactivas.

Ejemplo de uso por estudiantes

Un estudiante próximo a iniciar su Trabajo Final puede acceder al portal público, filtrar por carrera y temática (por ejemplo, sistemas distribuidos) y visualizar la evolución de proyectos en esa área. A partir de allí, identifica qué docentes dirigen

más trabajos relacionados y revisa títulos y resúmenes de proyectos previos, obteniendo insumos concretos para formular una propuesta viable y ubicar un potencial director.

7 Conclusiones

El desarrollo de *ThesisFlow* dotó al DCIC de una herramienta específica para la gestión y visualización de Trabajos Finales y Tesis de Grado. La plataforma:

- reduce la fragmentación de la información y la dependencia de planillas dispersas;
- mejora la accesibilidad y la transparencia de los datos académicos;
- habilita nuevas formas de análisis sobre la producción del departamento;
- ofrece soporte concreto a la toma de decisiones de estudiantes, docentes y autoridades.

Desde la perspectiva de ingeniería de software, el sistema constituye un caso de estudio de diseño e implementación de una aplicación web modular, con separación clara de responsabilidades, API REST y visualización de datos aplicada a un problema académico real.

8 Trabajo a futuro

Entre las posibles líneas de trabajo futuro se encuentran:

- Optimizar las analíticas mediante caching, vistas materializadas y/o almacenamiento auxiliar para resultados preprocesados.
- Extender la cobertura a Trabajos Finales de posgrado y a otros departamentos de la Universidad.
- Integrar la plataforma con sistemas institucionales existentes (gestión académica, repositorios digitales).
- Incorporar mecanismos de auditoría detallada (historial de cambios, diffs, trazabilidad de usuarios).
- Permitir que los alumnos suban recursos asociados a sus trabajos (código, datasets, documentación complementaria).
- Integrar almacenamiento de archivos dentro del sistema para reducir la dependencia de enlaces externos.
- Implementar estrategias de respaldo automático y procedimientos simplificados de restauración.

Agradecimientos Agradezco especialmente a mis directores, Dr. Martín Larrea y Dra. María Luján Ganuza, y a mi codirector, Lic. Diego Sebastián Orbe Leiva, por su guía, dedicación y acompañamiento durante el desarrollo de este trabajo. Su orientación académica y técnica fue fundamental para la concreción de este proyecto.