

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
VALPARAÍSO - CHILE**



**“SISTEMA DE GESTIÓN Y UNIFICACIÓN DE  
INFORMACIÓN ACADÉMICA PARA ALUMNOS DE  
INGENIERÍA CIVIL EN INFORMÁTICA”**

**PABLO ANTONIO GALLAGHER CAMUS**

**MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO DE EJECUCIÓN EN INFORMÁTICA**

**Profesor Guía: Pedro Godoy Barrera  
Profesor Correferente: Andrea Vásquez Guerra**

**mayo de 2025**

## **DEDICATORIA**

A mi familia, por ser mi mayor fuente de motivación y apoyo incondicional. Gracias por acompañarme en cada paso de este camino, brindándome su amor, paciencia y confianza.

Sus palabras de ánimo y su fe en mí han sido fundamentales para alcanzar este logro.

A mi profesor guía, cuyo conocimiento y orientación fueron esenciales para el desarrollo de este proyecto. Agradezco profundamente su disposición para compartir su experiencia y guiarme con sabios consejos, lo que permitió que esta memoria alcanzara los estándares esperados.

Este trabajo es un reflejo del esfuerzo conjunto y del apoyo invaluable que he recibido de todos ustedes.

## RESUMEN

**Resumen**— Actualmente, el Departamento de Informática enfrenta desafíos en la gestión académica debido a la dispersión de información relacionada con prácticas, evaluaciones de inglés y memorias, lo que dificulta el seguimiento eficiente del avance académico de los estudiantes. El proyecto desarrolla un sistema de información que digitaliza y centraliza los procesos de registro de documentos y solicitudes dirigidas a la Secretaría de Estudios, promoviendo una transformación digital en la gestión académica. La solución permite consolidar datos clave en una plataforma unificada, incorporando funcionalidades como carga masiva de registros, operaciones CRUD administrables, validaciones automatizadas y trazabilidad de acciones. Esto mejora significativamente la precisión, eficiencia y accesibilidad del registro académico, optimizando la toma de decisiones administrativas y el seguimiento del progreso estudiantil.

**Palabras Clave:** Gestión de datos; Transformación digital; Sistema de alumnos; Mejora de proceso; Registro académico

## ABSTRACT

**Abstract**— Currently, the Department of Computer Science faces challenges in academic management due to the dispersion of information related to internships, English assessments, and thesis projects, which hinders efficient tracking of students' academic progress. This project develops an information system that digitizes and centralizes the processes for registering documents and requests submitted to the Office of Academic Affairs, promoting a digital transformation in academic management. The solution consolidates key data into a unified platform, incorporating features such as bulk data upload, manageable CRUD operations, automated validations, and activity traceability. This significantly improves the accuracy, efficiency, and accessibility of academic records, optimizing administrative decision-making and the monitoring of student progress.

**Keywords:** Data management; Digital transformation; Student system; Process improvement; Academic record.

## GLOSARIO

**CRUD:** Acrónimo de Create, Read, Update, Delete. Conjunto básico de operaciones utilizadas en la gestión de datos en sistemas informáticos.

**Carga Masiva:** Proceso que permite la inserción simultánea de grandes volúmenes de datos en un sistema, optimizando tiempos y reduciendo errores.

**Centralización de Datos:** Estrategia que consolida toda la información en un único sistema, mejorando la accesibilidad y la gestión.

**Digitalización:** Proceso de convertir información y documentos físicos en formatos digitales para facilitar su almacenamiento, consulta y procesamiento.

**Historial Académico:** Registro detallado de las actividades y solicitudes relacionadas con el avance académico de los estudiantes.

**jQuery:** Biblioteca de JavaScript que simplifica la manipulación del DOM y las interacciones dinámicas en páginas web.

**AJAX:** Acrónimo de Asynchronous JavaScript and XML. Técnica que permite realizar solicitudes al servidor de manera asíncrona, mejorando la experiencia del usuario.

**Índice:** Estructura en una base de datos que acelera la recuperación de registros mediante la optimización de consultas.

**Memorias:** Trabajos finales o proyectos de grado realizados por los estudiantes como parte de los requisitos académicos.

**Prácticas Profesionales:** Actividades realizadas por los estudiantes en entornos laborales reales como parte de su formación académica.

**Registro Académico:** Base de datos que almacena información clave sobre el avance académico y administrativo de los estudiantes.

**Transformación Digital:** Adopción de tecnologías digitales para optimizar procesos y mejorar la eficiencia en la gestión de información.

**UTFSM:** Universidad Técnica Federico Santa María.

## INDICE DE CONTENIDOS

RESUMEN.....	3
ABSTRACT .....	3
GLOSARIO .....	4
INTRODUCCIÓN .....	8
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA .....	9
1.1 Contexto .....	9
1.2 Situación Actual.....	9
1.3 Actores Involucrados.....	10
1.4 Dificultades Actuales.....	10
1.5 Consecuencias de No Solucionar el Problema .....	10
1.6 Competencia y Soluciones Existentes .....	11
1.7 Objetivos y Alcance de la Solución.....	11
CAPÍTULO 2: MARCO CONCEPTUAL .....	13
2.1 Gestión de Datos Académicos.....	13
2.2 Sistemas Administrativo de datos.....	13
2.3 Carga Masiva de Datos.....	13
2.4 Centralización de Datos.....	14
2.5 Metodologías de Desarrollo de Software .....	14
2.6 Herramientas Utilizadas .....	15
2.7 Comparación con Soluciones Existentes.....	16
2.8 Conclusión .....	17
CAPÍTULO 3: PROPUESTA DE SOLUCION .....	18
3.1 Metodología y Planificación del Proyecto.....	18

3.2 Requerimientos Iniciales y su Refinamiento .....	19
3.2.1 Requerimientos Iniciales.....	19
3.2.2 Refinamiento a través de Reuniones.....	20
3.3 Diseño de la Solución .....	21
3.4 Desarrollo e Implementación.....	25
3.4.1 Desafíos y Soluciones.....	26
3.5 Validación y Resultados.....	26
3.6 Aportes Creativos y Originales .....	27
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN .....	28
4.1 Introducción .....	28
4.2 Estrategia de Pruebas de Validación.....	28
4.3 Pruebas de Carga de Datos .....	29
4.4 Validación de Registros Duplicados .....	29
4.5 Pruebas de Modificación de Información .....	30
4.6 Validación de interfaz usuario y Funcionalidades.....	30
4.7 Conclusiones de la Validación .....	31
CAPÍTULO 5: CONCLUSIONES .....	32
5.1 Resumen de los Resultados.....	32
5.2 Alcances y Limitaciones.....	32
5.3 Validez de los Objetivos Formulados .....	33
5.4 Principales Contribuciones y Aplicaciones del Trabajo.....	33
5.5 Impacto y Aporte a la Organización .....	34
5.6 Recomendaciones y Futuras Ampliaciones.....	34
5.7 Conclusión Final.....	35

REFERENCIAS BIBLIOGRÁFICAS..... 36

## INTRODUCCIÓN

La gestión de datos académicos en la carrera de Ingeniería Civil en Informática de la Universidad Técnica Federico Santa María, específicamente en el ámbito del Departamento de Informática, enfrenta desafíos significativos debido a la dispersión de información relacionada con diversos aspectos del avance de cada estudiante, como las inscripciones de prácticas, los resultados de test de inglés, entre otros. Esta fragmentación limita la eficiencia del control administrativo y aumenta la probabilidad de errores.

Actualmente, no existe un sistema unificado que permita registrar y consultar de forma eficiente la información académica que se gestiona en la Secretaría de Estudios, lo que obliga al uso de múltiples planillas, correos electrónicos y documentos aislados. Esta situación dificulta el seguimiento del avance académico de los estudiantes, retrasa los tiempos de respuesta ante solicitudes y genera duplicidad de esfuerzos en tareas administrativas. La falta de trazabilidad y centralización impide contar con un historial claro de las actividades académicas de cada alumno, afectando la toma de decisiones y la eficiencia de los procesos institucionales.

Para resolver esta problemática, el presente proyecto desarrolla un sistema que centraliza el registro de solicitudes enviadas a dirección de estudios y con ello se unifica la información de cada estudiante a nivel departamental. La solución permite registrar inscripciones, prácticas, exámenes y memorias, con opciones de carga masiva mediante archivos Excel y un CRUD administrable para evitar errores tipográficos. Este sistema facilita una gestión más ágil y precisa de los registros académicos.

La memoria se estructura en cinco capítulos:

1. **Definición del Problema:** Expone la problemática y sus implicancias en la administración de datos.
2. **Marco Conceptual:** Aborda los conceptos clave en la gestión de información educativa.
3. **Propuesta de Solución:** Describe el diseño del sistema y sus funcionalidades.
4. **Validación de la Solución:** Presenta pruebas y resultados del sistema implementado.
5. **Conclusiones:** Resume los logros y sugerencias para futuras mejoras.

## **CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA**

### **1.1 Contexto**

En la Universidad Técnica Federico Santa María, y en particular en la carrera de Ingeniería Civil en Informática impartida por el Departamento de Informática (DI), la gestión académica de alumnos constituye un área crítica para asegurar el seguimiento y control de su progreso académico. Esta gestión incluye procesos relacionados con la inscripción de prácticas profesionales, la evaluación de competencias en inglés y la administración de antecedentes vinculados a trabajos de memoria.

A medida que el número de estudiantes matriculados en la carrera ha aumentado, también lo ha hecho el volumen de datos que deben ser controlados por la jefatura de carrera, secretaría de pregrado del DI y dirección de estudios. Sin embargo, no existe un sistema que unifique esta información de forma centralizada, lo que obliga al uso de diversos canales como planillas Excel, correos electrónicos y documentación física o distribuida. Esta dispersión de datos genera dificultades para acceder oportunamente a la información, aumenta la carga administrativa, y eleva el riesgo de errores, omisiones o duplicidad de registros.

En este escenario, se identifica la necesidad de implementar un sistema que permita centralizar y consolidar la información académica de los estudiantes, facilitando su consulta, seguimiento y actualización. Esta solución contribuiría a optimizar los tiempos de respuesta, mejorar la trazabilidad de los antecedentes académicos y fortalecer la toma de decisiones al interior del Departamento de Informática.

### **1.2 Situación Actual**

Actualmente, en el Departamento de Informática de la Universidad Técnica Federico Santa María, la administración de datos académicos se realiza de forma fragmentada. El registro de antecedentes como prácticas profesionales, evaluaciones de inglés, y solicitudes de memoria se lleva a cabo a través de distintos canales no integrados, tales como el Sistema de Gestión de prácticas, planillas Excel, correos electrónicos enviados a la Secretaría de Estudios y documentos impresos entregados por los propios estudiantes.

El flujo de trabajo más representativo es la validación para titularse. Se debe validar si cumple con el nivel de inglés, prácticas aprobadas y todas las notas terminadas en el SIGA. Pero esta información no siempre está en un solo lugar, no está actualizada o simplemente no ha sido cargada.

Esta dispersión de datos genera demoras en la validación y respuesta a los estudiantes, además de dificultades para consolidar información de manera oportuna. La ausencia de un sistema centralizado impide que los encargados académicos cuenten con una visión integral del avance académico de cada alumno, lo cual afecta directamente la calidad del seguimiento y puede provocar errores en la toma de decisiones o la omisión de avisos

importantes. En consecuencia, la carga operativa sobre el personal administrativo es alta, y el proceso se vuelve propenso a errores tipográficos, duplicación de registros y pérdida de información.

### **1.3 Actores Involucrados**

Los principales usuarios de esta solución serían:

- **Secretarias:** Encargados de ingresar, modificar y revisar los datos de alumnos.
- **Jefes de Carrera:** Encargado de validar estado de los estudiantes de su carrera.
- **Estudiantes:** Como usuarios indirectos, se benefician de un sistema centralizado que asegura un seguimiento más preciso y transparente de sus actividades académicas.

### **1.4 Dificultades Actuales**

Entre las principales dificultades que enfrentan estos actores se encuentran:

- **Envío información a secretaría de estudios:** Muchas veces se ha enviado información a secretaría de estudios, pero ellos no la han procesado y subido al sistema SIGA, por lo que sólo la secretaria de pregrado sabe si el envío procedió o no. Al momento de querer verificar, se debe realizar nuevamente una consulta y/o revisar en sus registros/MAIL.
- **Errores en los registros:** La dispersión de datos aumenta la posibilidad de errores y dificulta la precisión en el ingreso y consulta de información.
- **Demoras en la carga y consulta de datos:** La falta de integración provoca demoras en el procesamiento y búsqueda de información.
- **Duplicidad y pérdida de registros:** La carencia de un sistema unificado da lugar a la duplicación de datos y al riesgo de pérdida de información relevante. Estas dificultades afectan tanto la eficiencia administrativa como la satisfacción de los estudiantes y profesores.

Además de los problemas previamente señalados, la falta de una interfaz de usuario interactiva y dinámica es otra limitación que enfrentan actualmente los sistemas de gestión académica. El procesamiento manual de peticiones y la falta de respuesta en tiempo real provocan demoras en la carga y consulta de información, lo que afecta tanto a estudiantes como a personal administrativo.

### **1.5 Consecuencias de No Solucionar el Problema**

Si estas dificultades no se abordan en el corto o mediano plazo, la gestión académica podría volverse insostenible a medida que aumente el número de estudiantes. Esto no solo afectaría la eficiencia del personal administrativo, sino que también impactaría

negativamente en la experiencia educativa, al no poder ofrecer un seguimiento preciso y completo del progreso académico de los alumnos.

También se puede señalar que, dado que estas consultas generalmente recaen en el jefe de carrera, es imprescindible resolverlo pronto para que no pierda tiempo en tareas operativas y pueda enfocarse en actividades de seguimiento o de aseguramiento de calidad de la carrera

### **1.6 Competencia y Soluciones Existentes**

Si bien en el mercado existen diversos sistemas de gestión académica ampliamente utilizados por instituciones de educación superior, muchos de ellos no responden de manera adecuada a las necesidades específicas del Departamento de Informática ni al enfoque particular de la carrera de Ingeniería Civil en Informática. Estos sistemas suelen estar orientados a cubrir una amplia gama de procesos administrativos generales, lo que los convierte en herramientas robustas, pero también complejas y poco flexibles para contextos específicos.

En este caso, no se busca implementar un sistema de gestión académica completo con todas las funcionalidades integradas que estos productos comerciales ofrecen. Por el contrario, se requiere una solución focalizada, que automatice y optimice un proceso puntual: el registro, control y seguimiento de las solicitudes y documentos académicos que los estudiantes envían a la Secretaría de Estudios. Es decir, una herramienta construida a la medida de los usuarios, que responda con precisión a los flujos de trabajo reales del departamento y que elimine las operaciones innecesarias.

### **1.7 Objetivos y Alcance de la Solución**

**Objetivo General:** Desarrollar un sistema de gestión unificado que centralice la información de prácticas, resultados de inglés y memorias de los estudiantes de informática y permita una administración eficiente y sin errores.

#### **Objetivos Específicos:**

1. Registrar el ingreso de los nuevos estudiantes para tener un estimativo de los estudiantes del departamento de informática.
2. Centralizar la información de solicitudes de prácticas, evaluaciones de Inglés y registros de memoria en un solo sistema.
3. Proveer un medio centralizado y rápido para las consultas de los estudiantes respecto a prácticas y Test de inglés.

**Alcance:** Este sistema abarca la gestión de registros académicos específicos de estudiantes de Ingeniería en Informática, integrando datos relacionados con inscripciones, prácticas, Test de Inglés y actividades finales como memorias. La solución optimizará la administración

académica y permitirá un acceso confiable y rápido a la información, beneficiando a estudiantes, profesores y administrativos. Además se incluirán funciones dinámicas de búsqueda y actualización de datos mediante jQuery y AJAX, eliminando la necesidad de recargar páginas para cada operación y mejorando la rapidez en las consultas y modificaciones de datos.

## CAPÍTULO 2: MARCO CONCEPTUAL

### ***2.1 Gestión de Datos Académicos***

La gestión de datos académicos se ha convertido en una función central en las instituciones de educación superior. Este proceso permite llevar un control integral de las actividades y registros de los estudiantes, mejorando la precisión y accesibilidad de la información.

Según **Picciano (2019)**, la digitalización y centralización de datos académicos facilita la transparencia, optimiza el análisis de la información y permite a las instituciones educativas realizar un seguimiento efectivo del progreso de sus estudiantes. Esto, a su vez, mejora la toma de decisiones y la calidad del servicio educativo que se ofrece.

En el caso específico del presente proyecto, la gestión centralizada de datos académicos de los estudiantes de Ingeniería en Informática está diseñada para superar las limitaciones actuales. Al consolidar información en un sistema único, se asegura la integridad de los datos, se facilita el acceso por parte de administrativos y profesores, y se promueve una experiencia educativa más eficiente y efectiva [Picciano19].

### ***2.2 Sistemas Administrativo de datos***

El concepto de CRUD (Create, Read, Update, Delete) es fundamental en la gestión de datos dentro de sistemas informáticos. Este modelo permite realizar operaciones básicas de manipulación de datos, garantizando que los mismos sean creados, leídos, actualizados y eliminados de manera segura y estructurada.

Según **Garrett (2020)**, los sistemas CRUD son esenciales para la administración eficiente de datos en bases de datos relacionales, ya que proporcionan una interfaz estandarizada para las interacciones del usuario. En el contexto del sistema propuesto, la implementación de un CRUD administrable asegura que los datos académicos puedan ser gestionados con precisión, minimizando errores tipográficos y garantizando la consistencia en los registros.

La funcionalidad CRUD permite a los usuarios administrativos interactuar directamente con los datos de los estudiantes, como actualizar información de contacto, registrar nuevas inscripciones o modificar detalles de evaluaciones y prácticas. Esto no solo optimiza los procesos internos, sino que también reduce significativamente el tiempo necesario para realizar tareas de gestión, mejorando la productividad general del sistema [Garrett20].

### ***2.3 Carga Masiva de Datos***

La carga masiva de datos es una herramienta clave para agilizar la gestión de grandes volúmenes de información en sistemas académicos. Esta funcionalidad permite a los usuarios ingresar múltiples registros simultáneamente mediante el uso de archivos estructurados, como Excel o CSV, en lugar de realizar entradas manuales individuales.

**Kroenke y Auer (2018)** destacan que la implementación de opciones de carga masiva no solo reduce los tiempos de procesamiento, sino que también minimiza los errores humanos al estandarizar los formatos de los datos. Este enfoque es particularmente útil en entornos

educativos donde la información de cientos o miles de estudiantes debe ser gestionada de manera eficiente [Kroenke18].

La carga masiva de datos permite procesar información relacionada con inscripciones, solicitudes de prácticas, evaluaciones de inglés y memorias. Los usuarios administrativos pueden cargar archivos Excel que contienen los registros necesarios, y el sistema valida automáticamente la estructura y los datos antes de almacenarlos en la base de datos.

Este enfoque asegura que los datos ingresados sean consistentes y libres de errores comunes, como duplicados o inconsistencias en los formatos. Además, facilita la integración de datos provenientes de diferentes fuentes, consolidando la información en un sistema centralizado que mejora la accesibilidad y la eficiencia operativa.

#### **2.4 Centralización de Datos**

La centralización de datos consiste en consolidar toda la información relevante en un único sistema, eliminando la dispersión y mejorando el acceso a los registros. Este enfoque permite que los datos sean gestionados de manera más eficiente, facilitando la consulta y actualización de la información desde un punto centralizado.

**Inmon y O'Neil (2019)** argumentan que la centralización de datos no solo mejora la eficiencia operativa, sino que también asegura la consistencia de la información al evitar duplicidades y errores. En el contexto académico, la centralización permite que administrativos y profesores accedan rápidamente a los registros de los estudiantes, optimizando la toma de decisiones y reduciendo el tiempo invertido en la búsqueda de información [Inmon19].

Se centralizan datos relacionados con inscripciones, solicitudes, evaluaciones y memorias en una base de datos relacional robusta. Esta base de datos actúa como el núcleo del sistema, permitiendo que todas las operaciones, como la carga masiva, las consultas y las actualizaciones, se realicen de manera coordinada y eficiente.

Además, la centralización de datos facilita la integración con otras herramientas y sistemas, como plataformas de análisis o servicios en la nube, ampliando las posibilidades de explotación de la información almacenada. Esto asegura que el sistema no solo sea funcional en el presente, sino también escalable para satisfacer futuras necesidades.

#### **2.5 Metodologías de Desarrollo de Software**

El desarrollo de software para sistemas académicos complejos requiere la aplicación de metodologías que permitan una gestión eficiente de los cambios y una entrega continua de valor. En este proyecto se adoptó una metodología ágil, que se caracteriza por ser iterativa, incremental y altamente colaborativa.

Según **Beck et al. (2001)**, las metodologías ágiles, como Scrum o Extreme *Programming* (XP), fomentan la adaptabilidad frente a requisitos cambiantes y la entrega continua de productos funcionales. Estas metodologías dividen el desarrollo en ciclos cortos llamados

*sprints*, lo que permite realizar revisiones frecuentes y asegurar que el producto final cumpla con las expectativas de los usuarios [Beck01].

En el ámbito del desarrollo de sistemas académicos y administrativos, las metodologías ágiles se han consolidado como un enfoque ampliamente utilizado por su capacidad de adaptación y entrega incremental de valor. A diferencia de los modelos tradicionales como el de cascada, donde los requisitos deben estar completamente definidos desde el inicio, las metodologías ágiles permiten responder a cambios durante el desarrollo y mantener una comunicación constante con los usuarios. Esto resulta especialmente relevante en contextos institucionales, donde los procesos pueden variar entre unidades y es frecuente que los requisitos evolucionen con base en el uso.

Diversos estudios y experiencias en el desarrollo de sistemas de información en universidades han reportado beneficios concretos con estas metodologías, como una mayor participación del usuario, mejor trazabilidad de cambios y entregas parciales funcionales que permiten validar decisiones antes de avanzar. Esta flexibilidad y enfoque iterativo han convertido a las metodologías ágiles en una elección recomendada para proyectos académicos donde los requisitos tienden a cambiar según la retroalimentación del usuario y los procesos administrativos particulares.

## ***2.6 Herramientas Utilizadas***

El sistema desarrollado se construyó utilizando herramientas ampliamente adoptadas en el desarrollo de aplicaciones web, considerando tanto los requerimientos técnicos como la facilidad de mantenimiento y la compatibilidad con entornos académicos. A continuación, se detallan las principales tecnologías utilizadas, diferenciando entre la lógica del servidor (back-end) y la capa de presentación (front-end), e integrando criterios de diseño de interfaz y experiencia de usuario.

### **Back-end:**

Para el desarrollo del back-end se utilizó **PHP**, debido a su integración nativa con servidores web, su curva de aprendizaje accesible y su amplia comunidad de soporte. PHP permitió construir una estructura modular para las distintas funcionalidades del sistema, incluyendo la lógica de validación de datos, el procesamiento de formularios y la comunicación con la base de datos.

Como sistema de gestión de base de datos se empleó **MariaDB**, una solución relacional de código abierto, compatible con MySQL y ampliamente usada en aplicaciones web. MariaDB permite trabajar con múltiples tablas relacionadas mediante claves foráneas, soporta consultas eficientes mediante índices y permite manejar grandes volúmenes de datos, lo cual es fundamental para funcionalidades como la carga masiva, los reportes filtrados y el historial de acciones. Además, se implementaron restricciones, validaciones y relaciones explícitas para mantener la integridad referencial entre las entidades del sistema.

### **Front-end y diseño de interfaz:**

En la capa de presentación se emplearon tecnologías estándar como **HTML5**, **CSS3** y **JavaScript**, junto con la biblioteca **jQuery**, que simplifica la manipulación de elementos en el DOM y mejora la compatibilidad entre navegadores. Se utilizó **AJAX** para realizar peticiones asincrónicas al servidor, lo que permite enviar y recibir datos sin necesidad de recargar la página. Esto fue fundamental para lograr una experiencia fluida en operaciones frecuentes como la búsqueda de alumnos, la carga de formularios y la validación en tiempo real.

Desde el punto de vista del diseño de interfaz (**UI**) y experiencia de usuario (**UX**), se aplicaron principios reconocidos en el estado del arte: agrupación lógica de campos, codificación por colores, consistencia visual, retroalimentación inmediata y validaciones antes del envío. Estas decisiones permiten reducir la carga cognitiva del usuario, prevenir errores y mejorar la eficiencia del flujo de trabajo. El sistema fue evaluado de forma iterativa junto a los usuarios, lo que permitió ajustar elementos visuales y organizativos hasta lograr una interfaz clara, accesible y centrada en las necesidades reales del Departamento de Informática.

#### **2.7 Comparación con Soluciones Existentes**

En el mercado actual existen diversas soluciones de gestión académica, pero muchas de ellas no se adaptan completamente a las necesidades específicas del Dpto. Estas plataformas suelen ser costosas y, en algunos casos, presentan limitaciones en cuanto a la personalización y la integración de funcionalidades avanzadas como la carga masiva de datos y un CRUD administrable.

Entre las soluciones existentes en el mercado destacan plataformas como *Banner*, *PeopleSoft Campus Solutions* y *U-Campus* (implementado en varias universidades chilenas), las cuales ofrecen sistemas integrales de gestión académica. Estas herramientas están diseñadas para cubrir un amplio espectro de procesos institucionales, incluyendo matrícula, gestión financiera, asistencia, calificaciones y control curricular. Sin embargo, su implementación suele implicar altos costos de licenciamiento, tiempos extensos de configuración, y en muchos casos, una escasa flexibilidad para adaptarse a necesidades específicas de unidades académicas como el Departamento de Informática.

**Rahimi et al. (2021)** destacan que las soluciones desarrolladas a medida ofrecen una ventaja significativa al adaptarse a los requisitos específicos de una organización, optimizando así los procesos internos y mejorando la satisfacción del usuario final. A diferencia de las soluciones comerciales, el sistema propuesto en este proyecto fue diseñado considerando las particularidades del entorno académico local, garantizando su relevancia y efectividad [Rahimi21].

Una de las principales diferencias con respecto a las plataformas existentes es la integración de herramientas como jQuery y AJAX, que permiten una interacción más fluida y eficiente entre los usuarios y el sistema. Esto no solo mejora la experiencia del usuario, sino que también reduce la carga en el servidor, optimizando el rendimiento general de la aplicación.

Además, el enfoque modular del sistema propuesto facilita su escalabilidad y mantenimiento a largo plazo. Las funcionalidades clave, como la validación automática de datos y la generación de reportes personalizados, proporcionan un valor añadido que no siempre está disponible en las soluciones comerciales. Esto posiciona al sistema como una alternativa robusta, eficiente y accesible para la gestión académica en instituciones de educación superior.

### ***2.8 Conclusión***

El marco conceptual presentado establece una base sólida para el desarrollo del sistema de gestión académica propuesto. Cada concepto, desde la gestión de datos académicos y la centralización hasta las metodologías de desarrollo ágil, contribuye a garantizar que la solución sea eficiente, escalable y alineada con las necesidades específicas del entorno académico.

En comparación con las soluciones existentes, el enfoque personalizado del sistema asegura una mayor adaptabilidad y relevancia, integrando funcionalidades clave que responden directamente a los desafíos actuales de las instituciones educativas. Este marco conceptual no solo respalda la implementación técnica del sistema, sino que también asegura que este sea capaz de evolucionar y adaptarse a las demandas cambiantes del entorno educativo.

## CAPÍTULO 3: PROPUESTA DE SOLUCION

Este capítulo presenta la solución propuesta para enfrentar los desafíos identificados en la gestión académica del Departamento de Informática. Se detallan los aspectos clave del proceso de desarrollo, comenzando por la metodología utilizada y los requerimientos definidos en conjunto con los usuarios. A continuación, se describe el diseño de la solución, tanto a nivel de base de datos como de interfaces del sistema, así como los principales módulos desarrollados. Se abordan también los desafíos técnicos enfrentados durante la implementación y las decisiones adoptadas para optimizar el rendimiento del sistema. Finalmente, se presentan los resultados obtenidos y los aportes innovadores incorporados en la propuesta, destacando su alineación con las necesidades reales del entorno académico.

### **3.1 Metodología y Planificación del Proyecto**

Para el desarrollo del sistema se aplicó una **metodología ágil**, basada en el marco de trabajo **Scrum**, adaptado a las necesidades del proyecto. Se establecieron **sprints de dos semanas**, en los cuales se levantaban los requerimientos directamente con los usuarios y se desarrollaban funcionalidades específicas que luego eran validadas en terreno. Esta dinámica permitió una evolución iterativa e incremental del sistema, en estrecha colaboración con los actores clave.

Durante el primer sprint se realizó el levantamiento inicial de los **dolores del proceso actual** mediante entrevistas informales y observación directa del trabajo realizado por la Secretaría de Estudios. A partir de esto, se definieron las primeras **historias de usuario**. Estas historias se priorizaron y se asignaron a los siguientes sprints de desarrollo.

Cada sprint incluyó una planificación al inicio (donde se definía el alcance del trabajo de las dos semanas), el desarrollo e implementación de las funcionalidades correspondientes, y una instancia de validación con los usuarios al cierre, en la que se presentaba lo desarrollado y se recogían ajustes o nuevos requerimientos.

El siguiente esquema describe el flujo aplicado:

#### **Modelo de trabajo aplicado:**

- **Duración de sprint:** 2 semanas
- **Semana 1:**
  - Reunión con usuarios (levantamiento de historias de usuario)
  - Priorización y planificación del sprint
  - Inicio de desarrollo

- **Semana 2:**
  - Finalización del desarrollo
  - Validación de funcionalidades con usuarios
  - Recolección de feedback y ajustes menores

Esta metodología permitió mantener una comunicación constante con los usuarios finales, asegurar que el sistema respondiera a necesidades reales, y entregar avances funcionales al término de cada ciclo. Gracias a este enfoque, el desarrollo fue más flexible ante cambios y permitió lograr un alto nivel de alineación entre lo construido y las expectativas del Departamento de Informática.

### **3.2 Requerimientos Iniciales y su Refinamiento**

El proyecto surge ante la necesidad de mejorar el proceso de recepción, registro y seguimiento de solicitudes académicas enviadas por los estudiantes del Departamento de Informática. Inicialmente, se identificó como requerimiento fundamental la creación de un sistema que permitiera centralizar esta información, reemplazando el uso de planillas y correos electrónicos por una plataforma digital más eficiente.

Durante las primeras reuniones con los usuarios, se definieron funcionalidades clave como: el registro estructurado de prácticas, evaluaciones de inglés y memorias; la trazabilidad de acciones mediante un historial; y un diseño que permitiera administrar datos sin recargar la página. A medida que el proyecto avanzó, estos requerimientos fueron refinados, incorporando características como la carga masiva de datos a través de archivos Excel, la gestión dinámica de campos mediante tablas parametrizadas y la implementación de tecnologías como jQuery y AJAX para mejorar la experiencia del usuario. Este proceso iterativo permitió adaptar la solución a los flujos reales de trabajo del departamento.

#### **3.2.1 Requerimientos Iniciales**

El sistema propuesto surge a partir de la necesidad de la problemática de la centralización de la información de prácticas, resultados de inglés y memorias de los estudiantes de informática. Los requerimientos iniciales fueron identificados mediante entrevistas con los principales actores involucrados en los procesos académicos, incluyendo administrativos y profesores. Entre los requerimientos más destacados se encuentran:

- **Centralización de Datos:** El sistema debe permitir consolidar toda la información académica de los estudiantes en una única plataforma centralizada, accesible por los usuarios autorizados.
- **Carga Masiva de Registros:** El sistema debe permitir importar múltiples registros a través de archivos Excel estructurados, para agilizar el ingreso de datos y reducir errores de digitación.
- **Historial Académico:** El sistema debe registrar automáticamente cada operación realizada sobre los datos de los estudiantes (creación, modificación, eliminación),

incluyendo la fecha, el usuario y el tipo de acción, para asegurar trazabilidad y control.

- **Interfaz Intuitiva:** Diseñar una interfaz de usuario sencilla e intuitiva, que permita a los usuarios interactuar con el sistema sin necesidad de conocimientos técnicos avanzados.

Estos requerimientos iniciales sirvieron como base para el diseño y desarrollo del sistema, priorizando aquellas funcionalidades que ofrecieran un mayor impacto en la mejora de los procesos académicos y administrativos. Para estructurar el trabajo bajo un enfoque ágil, se levantaron historias de usuario que reflejaran necesidades reales de los distintos perfiles involucrados. Por ejemplo:

- *“Como jefe de carrera, quiero acceder rápidamente al estado académico de los estudiantes para verificar si están en condiciones de titularse, sin tener que solicitar información a distintas áreas.”*
- *“Como secretaria, quiero registrar y consultar en un solo lugar todas las solicitudes de prácticas, inglés y memoria de los estudiantes, para evitar revisar planillas y correos por separado.”*

Estas historias permitieron definir con claridad qué funcionalidades debían abordarse en cada sprint, asegurando que el sistema entregara valor desde las primeras iteraciones y se ajustara a los flujos de trabajo reales del Departamento de Informática.

### **3.2.2 Refinamiento a través de Reuniones**

A medida que se avanzó en los sprints y se profundizó en la interacción con los usuarios, algunos requerimientos iniciales fueron ajustados o ampliados, lo que llevó también a definir con mayor precisión las herramientas a utilizar. Estas decisiones se tomaron en función del contexto técnico, los recursos disponibles y las necesidades funcionales del sistema.

Una de las herramientas incorporadas fue **jQuery**, la cual se seleccionó por su capacidad de simplificar la manipulación del DOM y manejar eventos en el navegador de forma más eficiente y compatible entre distintos navegadores. Dado que el sistema debía ofrecer una interfaz ágil, sin recargas constantes, se requería una biblioteca liviana que facilitara operaciones como mostrar u ocultar secciones dinámicamente, validar formularios en el cliente y mejorar la experiencia general del usuario. En este sentido, jQuery se alinea con los principios del estado del arte en diseño de interfaz (UI), al permitir una experiencia más fluida y responsiva sin necesidad de frameworks más complejos que habrían aumentado el tiempo y complejidad de implementación.

Además, jQuery facilitó el uso de **AJAX**, indispensable para enviar y recibir datos de forma asíncrona entre el cliente y el servidor. Esto fue clave para funcionalidades como búsquedas filtradas de alumnos, carga de formularios, validación en tiempo real y operaciones CRUD sin recargar la página. La elección de jQuery + AJAX respondió a la necesidad concreta de

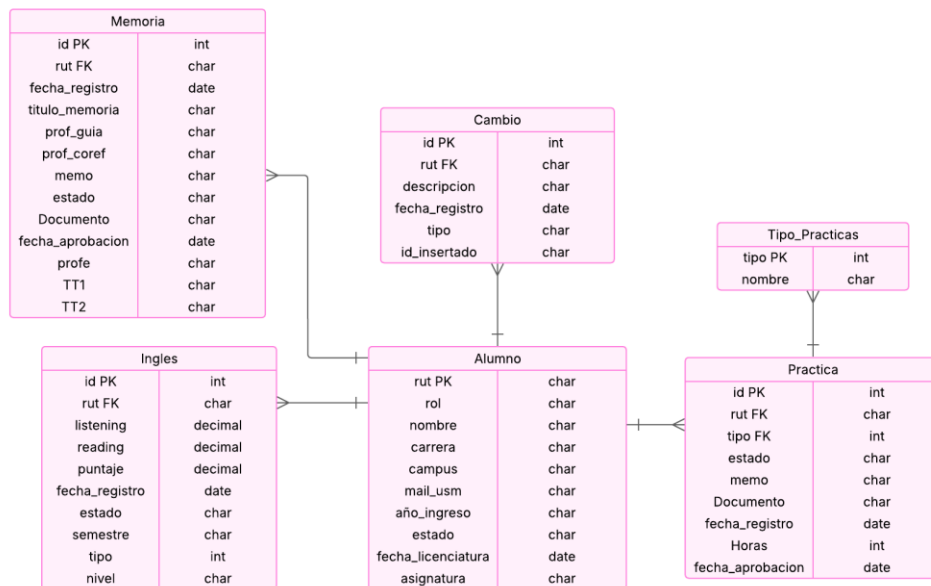
lograr un sistema más eficiente, modular y amigable, sin sacrificar la simplicidad en el desarrollo ni en el mantenimiento posterior.

### 3.3 Diseño de la Solución

El diseño del sistema fue realizado con base en un modelo entidad-relación (MER) que define las relaciones entre las diferentes tablas de la base de datos y un modelo de clases para la capa lógica de la aplicación.

#### Modelo Entidad-Relación:

El sistema se apoya en una base de datos relacional construida sobre **MariaDB**, con una estructura modular orientada a representar información académica de estudiantes del Departamento de Informática. A continuación, se describe un resumen del modelo entidad-relación, incluyendo los principales atributos y sus tipos de datos.



#### Principales tablas y atributos

- **Alumno:**
  - rut (CHAR(15), PK)
  - rol, nombre, carrera, campus, mail\_usm, mail\_inf (CHAR)
  - año\_ingreso (CHAR(4)), estado (CHAR(20)), fecha\_licenciatura (DATE), asignatura (CHAR(100))
- **Ingles:**
  - id (INT, PK), rut (FK, CHAR(15))
  - listening, reading, puntaje (DECIMAL(19,4))

- fecha\_registro (DATE), estado (CHAR(10)), semestre (CHAR(7)), tipo (INT), nivel (CHAR(15))
- **Memoria:**
  - id (INT, PK), rut (FK, CHAR(15))
  - titulo\_memoria, prof\_guia, prof\_coref, memo, estado, Documento, profe (CHAR)
  - fecha\_registro, fecha\_aprobacion (DATE), TT1, TT2 (CHAR)
- **Practica:**
  - id (INT, PK), rut (FK, CHAR(15)), tipo (FK, INT)
  - estado, memo, Documento (CHAR), fecha\_registro, fecha\_aprobacion (DATE), Horas (INT)
- **Tipo Practicas:**
  - tipo (INT, PK), nombre (CHAR(50))
- **Cambio (historial de acciones):**
  - id (INT, PK), rut (FK, CHAR(15))
  - descripcion (CHAR(200)), fecha\_registro (DATE), tipo (CHAR(10)), id\_insertado (CHAR(15))

### Navegación entre tablas

La tabla central del modelo es **Alumno**, ya que a partir de su campo clave primaria (rut) se relacionan las demás entidades: Ingles, Memoria, Practica y Cambio. Esto permite construir una vista unificada del avance académico de cada estudiante.

Por ejemplo:

- Un registro en la tabla **Memoria** está vinculado a un alumno mediante rut, y puede ser consultado junto con su información de práctica o inglés.
- La tabla **Cambio** actúa como historial, registrando acciones asociadas a las distintas solicitudes del estudiante.

### Tabla más relevante del modelo

La entidad más relevante es **Alumno**, ya que funciona como el núcleo del modelo. Todas las funcionalidades del sistema (registro, consulta, seguimiento y validación de requisitos) se estructuran en torno al rut del estudiante. Su correcta estructuración garantiza la trazabilidad y consolidación de información académica en una sola vista.

## Modelo de Arquitectura del Sistema

El sistema se implementa bajo una arquitectura web cliente-servidor, estructurada en tres capas:

### 1. Capa de presentación (Front-end):

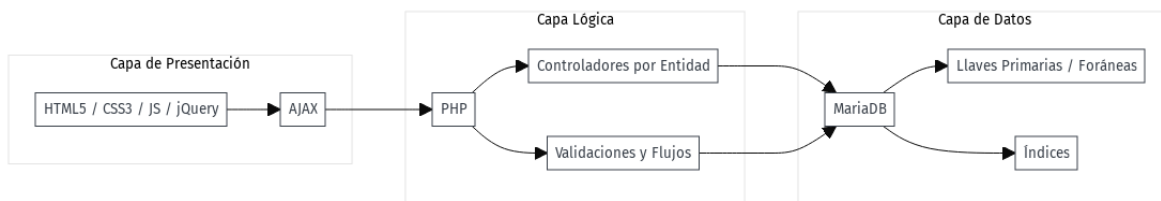
- **HTML5, CSS3, JavaScript y jQuery** para construir una interfaz interactiva.
- Uso de **AJAX** para enviar y recibir datos sin recargar la página, mejorando la experiencia del usuario.

### 2. Capa lógica (Back-end):

- Implementada en **PHP**, se encarga del procesamiento de datos, control de flujos, validaciones, y comunicación con la base de datos.
- Organización modular basada en controladores funcionales por entidad (Alumno, Memoria, Práctica, etc.).

### 3. Capa de datos (Base de datos):

- Base de datos relacional **MariaDB**, estructurada mediante llaves primarias y foráneas, y optimizada con el uso de índices para mejorar el rendimiento en operaciones críticas.



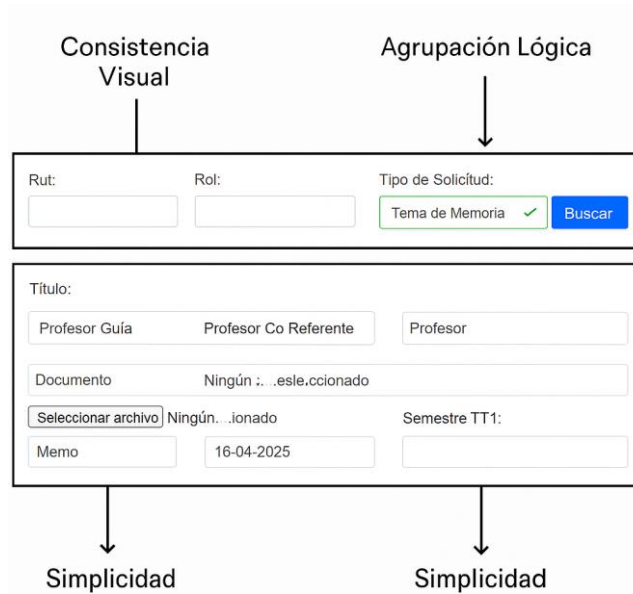
Este modelo arquitectónico permite mantener una clara separación de responsabilidades, facilita el mantenimiento, y asegura que el sistema sea escalable, seguro y eficiente para su uso dentro del Departamento de Informática.

## Diseño de Interfaces:

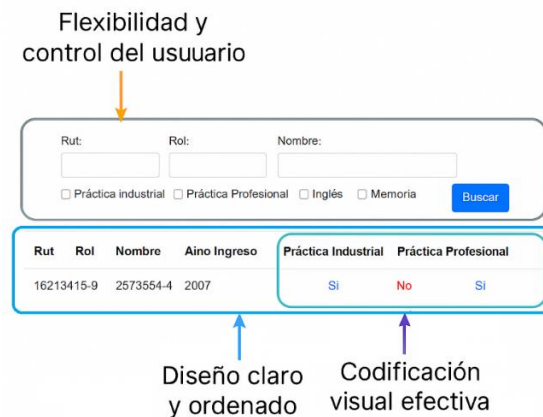
Las interfaces del sistema fueron diseñadas con un enfoque en la usabilidad y accesibilidad. Las principales interfaces incluyen:

1. **Nueva Solicitud:** Permite ingresar una nueva solicitud vinculada a un alumno previamente registrado. Para el diseño se consideraron principalmente las siguientes reglas de diseño:
  - **Consistencia visual:** Los campos están alineados y utilizan un formato uniforme, lo que facilita la navegación y comprensión del formulario.

- **Agrupación lógica:** La información está organizada por secciones relacionadas, ayudando al usuario a entender el flujo de ingreso.
- **Simplicidad:** La interfaz es limpia y presenta solo los campos necesarios, evitando distracciones y facilitando un uso rápido y eficiente.



2. **Búsqueda Avanzada:** Ofrece la posibilidad de buscar alumnos utilizando múltiples filtros. Para el diseño se consideraron principalmente las siguientes reglas de diseño:
- **Flexibilidad y control del usuario:** Permite realizar búsquedas utilizando distintos filtros combinables, adaptándose a las necesidades del usuario.
  - **Codificación visual efectiva:** El uso de colores (azul para “Sí” y rojo para “No”) facilita la interpretación rápida del estado de cada elemento.
  - **Diseño claro y ordenado:** La interfaz es limpia, con campos y resultados bien alineados, lo que mejora la lectura y navegación.



3. **Carga Masiva:** Interfaz para la importación de datos mediante archivos Excel. Para el diseño se consideraron principalmente las siguientes reglas de diseño:

- **Retroalimentación inmediata:** Se informa visualmente qué columnas son requeridas (resaltadas en rojo) y si el archivo ha sido correctamente seleccionado y procesado.
- **Prevención de errores:** Se destacan los campos obligatorios antes de enviar los datos, lo que ayuda a evitar omisiones y fomenta una carga precisa.
- **Flexibilidad y control del usuario:** Permite al usuario mapear manualmente las columnas del archivo con los campos del sistema, adaptándose a distintos tipos de solicitudes.

The screenshot displays a web interface for mass data upload. At the top, there are two input fields: 'Tipo de Carga' (set to 'Alumnos') and 'Archivo de Carga' (with a file named 'prueba subida alumno.xlsx'). A 'Cargar Archivo' button is to the right. Below this, a section titled 'Columnas del archivo de carga:' shows a row of column names: 'rut', 'rol', 'nombres', 'mail usm', 'estado', 'campus', 'carrera', and 'año ingreso'. The first four are highlighted in red, with a note: 'Los nombres en rojo indican campos requeridos'. Below this, a mapping table is shown with dropdown menus for each column. The table below has columns: 'rut', 'rol', 'nombre', 'apellido', 'campus', 'carrera', 'mail\_usm', 'mail\_inf', 'año\_ingreso', and 'estado'. A sample row contains: '16213415-9', '2573554-4', 'Pablo', 'Gallagher', 'CC', 'ING', 'pablo.gallagher@alumnos.usm.cl', '2005', and 'Activo'. Below the table is an 'Enviar Datos' button and an information icon. At the bottom, a 'Registros no insertados' section shows five red error messages: 'El rut es requerido', 'El rol es requerido', 'El nombre es requerido', 'El email es requerido', and 'El año de ingreso es requerido'. Arrows indicate the flow of information: 'Prevención de errores' points to the red highlights; 'Flexibilidad y control del usuario' points to the mapping table; and 'Retroalimentación inmediata' points to the error messages.

### 3.4 Desarrollo e Implementación

El desarrollo de la solución se centró en implementar un proceso de consolidación de datos mediante **PHP** y optimizar las consultas en **MariaDB**. Los scripts se encargan de extraer datos de varias tablas, consolidarlos en estructuras temporales y finalmente realizar la inserción en una tabla principal.

Durante la implementación, se incorporaron **jQuery** y **AJAX** en todas las funcionalidades principales del sistema para mejorar la interacción del usuario con la plataforma. Estas herramientas fueron seleccionadas por su simplicidad, amplia documentación, y facilidad de integración con PHP y HTML, lo que las convierte en una opción eficiente para proyectos de mediana escala y recursos limitados. A diferencia de frameworks más complejos como **React** o **Vue.js**, que requieren configuraciones adicionales, entornos de desarrollo y una curva de aprendizaje más alta, **jQuery** permite implementar rápidamente componentes dinámicos sin sacrificar compatibilidad ni rendimiento. En particular, el uso de **AJAX**

permitió realizar peticiones asíncronas al servidor, evitando recargas de página y mejorando significativamente la fluidez del sistema en operaciones como búsqueda de alumnos, validación de formularios y carga masiva de registros. Esta elección se alineó con la necesidad de mantener un desarrollo ágil, mantener bajo el costo de mantenimiento, y entregar una solución funcional y fácil de adaptar al entorno del Departamento de Informática.

### 3.4.1 Desafíos y Soluciones

Uno de los principales retos fue optimizar las consultas para que manejaran grandes volúmenes de datos sin sobrecargar el servidor. Para esto, se implementaron índices adecuados y se ajustaron las configuraciones de MariaDB para mejorar el rendimiento. También se desarrollaron mecanismos de manejo de errores que permitieran continuar el flujo de trabajo sin interrupciones.

Por ejemplo, en la tabla **Alumnos**, se implementó un índice clúster en el campo rut, ya que este campo es la clave primaria y se utiliza como llave foránea en varias tablas relacionadas, como Inglés, Memorias y Prácticas. Este índice garantiza búsquedas rápidas y relaciones consistentes entre las entidades.

Table name: Alumnos

Index name: PK\_Alumnos\_C2B74E77D93F1626

Index type: Clustered

Unique

Index key columns					
Name	Sort Order	Data Type	Size	Identity	Allow NULLs
rut	Ascending	char(15)	15	No	No

Otro de los desafíos fue asegurar que las interacciones en tiempo real funcionaran de manera confiable, ya que el uso de jQuery y AJAX requería manejar correctamente la comunicación entre funciones. Para resolver esto, se implementaron controles y validaciones, asegurando que solo se enviaran los datos necesarios y que los resultados de cada operación fueran precisos y consistentes.

### 3.5 Validación y Resultados

Se realizaron pruebas de carga para medir el rendimiento del sistema y asegurar su funcionalidad. Las pruebas demostraron que el proceso de consolidación podía validar la consistencia de la información ingresada, logrando validar la precisión de la información.

Se realizaron reuniones periódicas con el usuario y la secretaria, con el objetivo de validar la forma de presentar los datos y las principales funcionalidades del sistema.

Entre las observaciones trabajadas se destaca el manejo del Rut y la integración de los nombres, lo cual permitió implementar un sistema de validaciones al ingreso de solicitudes, como a la carga masiva

### ***3.6 Aportes Creativos y Originales***

El principal aporte creativo de esta solución fue la creación de un proceso de previsualización del archivo Excel para determinar el cruce de cada columna con el esquema de datos permitiendo flexibilidad en la carga dentro del sistema. Además, la implementación de un registro de actividades en **PHP** permitió una trazabilidad completa del proceso, mejorando la capacidad de identificar y resolver errores de manera eficiente.

**Conclusión:** El desarrollo de esta solución demostró un enfoque innovador al combinar la potencia de **MariaDB** con la flexibilidad de **PHP**, optimizando la gestión de grandes volúmenes de datos y asegurando un flujo de trabajo robusto y confiable.

## CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

### 4.1 Introducción

La validación de la solución es fundamental para confirmar su funcionalidad y fiabilidad en el entorno planteado. En este capítulo se detallan las pruebas realizadas para verificar la carga de datos, la identificación y gestión de duplicados, y la modificación de registros en la base de datos. Estas pruebas aseguran que la solución cumple con los requerimientos de integridad y exactitud.

### 4.2 Estrategia de Pruebas de Validación

Las pruebas de validación se centraron en tres aspectos clave:

- **Pruebas de carga de datos:** evaluar la capacidad del sistema para manejar la inserción de grandes volúmenes de datos de manera eficiente.
- **Validación de duplicados:** verificar que el sistema identifica registros duplicados y emite alertas o reportes sobre ellos.
- **Pruebas de modificación de datos:** comprobar la integridad de los datos al realizar modificaciones y actualizaciones en los registros existentes.

La validación de la solución incluyó pruebas específicas sobre las funciones asíncronas implementadas con jQuery y AJAX, evaluando su impacto en la usabilidad y la eficiencia de las peticiones al servidor. Se realizaron pruebas de carga en las peticiones AJAX para confirmar que el sistema mantuviera una respuesta óptima incluso bajo condiciones de alta demanda de datos y se verificó que las respuestas fueran precisas y en tiempo real.

Módulo	Función	Filtro	Tiempo	N° Tablas	N° Registros
Alumno	Buscar	Rol	758 ms	4	10
Alumno	Búsqueda Avanzada	-	648 ms	1	148
Carga Masiva	Lectura Excel	-	689 ms	-	1
Carga Masiva	Alumno	-	353 ms	1	1
Carga Masiva	Lectura Excel	-	1.48 s	-	200
Carga Masiva	Alumno	-	3.15 s	1	200

La imagen muestra una tabla de resultados obtenidos tras realizar pruebas de distintas funcionalidades del sistema en los módulos de **Alumno** y **Carga Masiva**. Cada fila representa una operación específica, donde se detalla la **función evaluada**, si se utilizó algún **filtro** en el envío de datos, el **tiempo de respuesta** medido hasta obtener resultados, el **número de tablas** involucradas en la consulta o inserción, y la **cantidad de registros** procesados. Por ejemplo, en la funcionalidad “Buscar” del módulo Alumno, se empleó el filtro “Rol”,

consultando cuatro tablas y obteniendo diez registros en 758 milisegundos. Estas pruebas permiten evaluar el rendimiento y eficiencia del sistema en distintos escenarios de uso.

### 4.3 Pruebas de Carga de Datos

Las pruebas de carga se llevaron a cabo simulando la inserción de grandes volúmenes de datos en la base de datos de **MariaDB** utilizando scripts en **PHP**. El proceso incluyó la carga de archivos CSV y la inserción directa a través de consultas.

#### Proceso de Prueba:

- Se prepararon archivos CSV con 100 y 200 registros.
- Dentro de estos registros se incorporaron algunos que no debieran superar las validaciones
- Se ejecutaron scripts en PHP que leían y cargaban estos archivos en la base de datos.
- Se verificó que los registros válidos se cargaran completamente y sin interrupciones, mientras que los registros no válidos fueron indicados como tal y la razón del rechazo

**Resultado:** Todos los registros válidos se insertaron correctamente, y el sistema fue capaz de manejar la carga sin perder datos ni generar errores.

Las pruebas de carga demostraron que la integración de AJAX mantuvo el rendimiento en comparación con métodos de recarga de página completos, pero mejoró la interacción con el usuario. Las peticiones AJAX permitieron que las operaciones de inserción y consulta de datos se ejecutaran de forma más amigable, evitando interrupciones en el flujo de trabajo.

### 4.4 Validación de Registros Duplicados

Para asegurar la integridad de los datos, se implementaron validaciones que identifican registros duplicados durante la carga. Esta validación se realizó mediante un script en PHP que compara cada registro con los ya existentes en la base de datos.

#### Descripción de la Prueba:

- Antes de la carga de datos, el sistema verifica la existencia de registros duplicados basándose en un identificador único.
- Si se encuentra un registro duplicado, se genera un informe detallado indicando el ID del registro y la razón de la duplicidad.

#### Registros no insertados

rut	rol	nombre	mail_usm	año_ingreso	Errores
16213415-9	2573554-4	Pablo	pablo.gallagher@alumnos.usm.cl	2005	Alumno ya ingresado

**Resultado:** El sistema identificó correctamente los registros duplicados y generó un informe que permitió al usuario tomar decisiones sobre la carga.

#### ***4.5 Pruebas de Modificación de Información***

Las pruebas de modificación de datos verificaron que el sistema permitiera actualizaciones y eliminaciones de manera segura, manteniendo la integridad de los registros.

##### **Proceso de Prueba:**

- Se seleccionaron registros de prueba y se modificaron atributos clave.
- Se realizaron actualizaciones masivas para comprobar la estabilidad del sistema.
- Se verificó que los cambios se reflejaran correctamente en la base de datos.

Las modificaciones fueron registradas en el historial de solicitudes, respaldando el hecho de que una serie de modificaciones fueron realizadas y el sistema opera con normalidad.

**Resultados:** Las modificaciones de datos se realizaron con éxito, sin afectar la integridad ni la consistencia de los registros. Las pruebas también incluyeron la validación de la función cancelar para revertir cambios en caso de error.

#### ***4.6 Validación de interfaz usuario y Funcionalidades***

Durante el desarrollo del sistema, se llevaron a cabo múltiples sesiones de validación con usuarios reales —principalmente secretarías y jefes de carrera— con el fin de asegurar que la interfaz no solo fuese funcional, sino también intuitiva, accesible y alineada con sus flujos de trabajo diarios. Esta validación se realizó de manera iterativa al finalizar cada sprint, lo que permitió recoger retroalimentación temprana y aplicar mejoras continuas.

##### **Evaluación de Usabilidad**

Se aplicaron principios de diseño centrado en el usuario, tales como:

- **Consistencia visual y navegación intuitiva**, lo que redujo el tiempo necesario para aprender a usar el sistema.
- **Agrupación lógica de campos y codificación por colores**, especialmente en los módulos de búsqueda y carga masiva, lo que ayudó a reducir errores de interpretación.
- **Retroalimentación inmediata** mediante mensajes contextuales y respuestas AJAX que indicaban si una operación fue exitosa, si hubo errores, o si ciertos campos requerían corrección.

## Validación Funcional

Los siguientes módulos fueron evaluados directamente por usuarios en condiciones reales de uso:

Módulo	Función Evaluada	Resultado
Búsqueda avanzada	Uso de múltiples filtros y rapidez de resultados	Interfaz valorada como clara y eficiente. Se propuso y aplicó destacar los resultados en color.
Nueva Solicitud	Ingreso de prácticas y memorias	Flujo validado como intuitivo. Se agregó validación por RUT y carga automática de nombre.
Carga Masiva	Carga de datos y documentos asociados	Se mejoró el mapeo manual de columnas y se incorporó previsualización antes de insertar registros.
Historial	Seguimiento de cambios realizados	Valorado especialmente por jefes de carrera como herramienta para trazabilidad.

## Satisfacción de Usuarios

Al término del proceso, se consultó informalmente a los usuarios clave para saber sus opiniones:

- Consideraron que el sistema es más claro que las planillas anteriores.
- Valoran positivamente la posibilidad de operar sin recargar la página.
- Lo más útil es el historial de cambios y validaciones en tiempo real.

## Ajustes Finales

Los principales cambios incorporados tras las pruebas fueron:

- Inclusión de validaciones visuales en campos obligatorios.
- Mejora en la navegación entre módulos, permitiendo retornar fácilmente al menú principal.
- Incorporación de validaciones automáticas al ingresar datos repetidos en la carga masiva.

### ***4.7 Conclusiones de la Validación***

Las pruebas realizadas confirmaron que la solución es robusta y puede gestionar de manera eficiente la carga, validación y modificación de datos. La identificación de duplicados y las pruebas de modificación mostraron que el sistema es confiable y cumple con los requerimientos establecidos.

## CAPÍTULO 5: CONCLUSIONES

### **5.1 Resumen de los Resultados**

El proyecto de desarrollo e implementación del sistema de gestión y unificación de información académica cumplió satisfactoriamente con los objetivos establecidos. La solución centraliza los datos académicos de los estudiantes en un único sistema, permitiendo registrar inscripciones, solicitudes de prácticas, evaluaciones de inglés y memorias de manera estructurada y accesible. A través de la carga masiva de datos mediante archivos Excel y un sistema CRUD, se logró minimizar los errores tipográficos y reducir el tiempo requerido para actualizar los registros. Las pruebas realizadas confirmaron que el sistema permite una gestión eficiente, manteniendo la integridad y consistencia de los datos y facilitando su consulta y modificación de manera rápida y precisa.

El sistema demostró su eficacia al procesar diferentes volúmenes de datos sin interrupciones y con un rendimiento estable, cumpliendo con los requerimientos iniciales y aportando mejoras en la administración académica. Este proyecto representa un avance significativo en la gestión de datos académicos, optimizando la experiencia tanto para el personal administrativo como para los estudiantes y profesores, quienes ahora disponen de un acceso más ágil y confiable a la información académica.

La integración de jQuery y AJAX en el sistema fue fundamental para mejorar la eficiencia y la experiencia de usuario. Al permitir que las peticiones al servidor se realicen de manera asíncrona, el sistema se vuelve más ágil y responsivo, proporcionando a los usuarios una interacción más fluida y reduciendo significativamente el tiempo necesario para realizar consultas y actualizaciones.

### **5.2 Alcances y Limitaciones**

#### **Alcances:**

El sistema ofrece una solución integral para la gestión de registros académicos en Ingeniería en Informática, facilitando la centralización de información relevante en un solo lugar. Esta estructura modular permite su adaptación a otros programas educativos o sectores que requieran un manejo intensivo de datos, como la administración de inventarios o la gestión de empleados en empresas de diversos rubros. La implementación de carga masiva mediante archivos Excel permite una integración eficiente de grandes volúmenes de datos, lo cual es ideal para instituciones con cientos de registros a gestionar.

#### **Limitaciones:**

Sin embargo, el sistema también presenta ciertas limitaciones. Actualmente, depende de archivos Excel para la carga masiva, lo que implica que los datos deben ser preformateados y organizados correctamente antes de su ingreso al sistema. Además, aunque el sistema de detección de duplicados es funcional, se basa únicamente en identificadores únicos, limitando su capacidad para identificar duplicados en datos con variaciones menores o errores. A nivel de infraestructura, el rendimiento y estabilidad del sistema dependen de

las capacidades del servidor donde se implemente, lo cual puede requerir ajustes adicionales para mantener la eficiencia en entornos de alta demanda.

### ***5.3 Validez de los Objetivos Formulados***

Los objetivos establecidos en el inicio del proyecto fueron cumplidos de manera satisfactoria. El sistema de gestión y unificación de información académica logró:

- **Centralizar los datos** de los estudiantes de manera efectiva, consolidando información dispersa en un único sistema.
- **Simplificar el proceso administrativo**, permitiendo la actualización y consulta de datos con mayor rapidez y precisión.
- **Reducir errores tipográficos** mediante la implementación de un sistema CRUD, lo cual mejora la integridad de los datos almacenados.
- **Facilitar el ingreso de grandes volúmenes de datos** mediante la carga masiva de archivos, optimizando el tiempo de registro.

Estos logros validan la solución propuesta, ya que el sistema no solo mejoró la administración académica sino que también aportó un valor significativo al reducir la carga de trabajo manual y mejorar la precisión en los registros. Los objetivos fueron alcanzados no solo en términos de funcionalidad, sino también en cuanto a la usabilidad del sistema, logrando un impacto positivo en los usuarios finales.

### ***5.4 Principales Contribuciones y Aplicaciones del Trabajo***

#### **Contribuciones:**

Este proyecto hace varias contribuciones importantes al campo de la gestión académica y al desarrollo de sistemas de información:

1. **Solución de código abierto:** La implementación en PHP y MariaDB permite que el sistema sea replicable y adaptable en otras instituciones educativas sin incurrir en los costos de sistemas propietarios.
2. **Optimización de procesos administrativos:** Al reducir la cantidad de tareas manuales, el sistema aumenta la eficiencia operativa de los departamentos académicos.
3. **Modelo de integración de carga masiva y CRUD:** La combinación de estas funcionalidades sienta un precedente para el diseño de sistemas que requieren precisión y rapidez en la gestión de grandes volúmenes de datos.

#### **Aplicaciones Prácticas:**

Este sistema es especialmente útil en instituciones educativas con altos volúmenes de datos, pero también podría adaptarse a otros sectores, como la administración de recursos humanos (gestión de empleados, historial laboral) y el área de ventas (base de datos de

clientes, historial de compras). La precisión y organización que proporciona el sistema pueden optimizar la calidad de los datos y facilitar la toma de decisiones informadas en diversos contextos.

### ***5.5 Impacto y Aporte a la Organización***

El impacto del sistema en la organización es notable, ya que ofrece una herramienta que mejora significativamente el control sobre los registros académicos y optimiza los procesos de administración de información. Al centralizar los datos y reducir los errores, el sistema no solo mejora la eficiencia operativa, sino que también aporta a la transparencia y confiabilidad de los registros. Esto se traduce en un beneficio directo para los estudiantes y profesores, quienes disponen de información confiable y fácil de consultar.

En términos de eficiencia operativa, el sistema permite una notable reducción en el tiempo de carga y validación de datos en comparación con los métodos manuales. Además, al mejorar la calidad de los datos almacenados, la organización se beneficia de una base de información más sólida para el análisis y la toma de decisiones, fortaleciendo así su capacidad para gestionar el crecimiento y evolución de sus estudiantes.

El uso de jQuery y AJAX no solo mejoró la eficiencia del sistema, sino que también redujo la carga en el servidor, ya que solo se envían y reciben los datos necesarios en cada interacción. Esto optimiza el rendimiento y hace que el sistema sea más escalable y capaz de responder a un número mayor de usuarios de manera simultánea.

### ***5.6 Recomendaciones y Futuras Ampliaciones***

#### **Recomendaciones:**

1. **Automatización del Preprocesamiento de Archivos:** Incorporar una funcionalidad que valide y formatee los archivos de carga masiva de manera automática antes de su ingreso al sistema, reduciendo errores y simplificando el proceso de carga para los usuarios.
2. **Notificaciones en Tiempo Real:** Implementar un sistema de alertas por correo electrónico o notificaciones dentro de la plataforma para mantener a los usuarios informados sobre el estado de la carga de datos, errores detectados o cualquier problema que requiera su atención.
3. **Análisis y Reportes Visuales:** La inclusión de un módulo de análisis gráfico permitiría a los usuarios visualizar métricas y tendencias en los registros académicos, facilitando la interpretación de datos y mejorando la toma de decisiones informadas.

#### **Futuras Ampliaciones:**

1. **Detección Avanzada de Duplicados:** Mejorar el sistema de detección de duplicados mediante algoritmos como Jaccard o Levenshtein, que identifiquen coincidencias

parciales y similitudes en texto, permitiendo una mayor precisión al identificar datos duplicados con variaciones mínimas.

2. **Escalabilidad en la Nube:** La migración del sistema a una infraestructura en la nube (Azure, AWS) podría aumentar la capacidad de manejo de datos y la disponibilidad del sistema, permitiendo una respuesta más ágil en escenarios de alta demanda y crecimiento.
3. **Módulo de Seguridad Avanzado:** Incluir controles de acceso y autenticación avanzados para proteger la integridad de los datos, además de registros de auditoría para rastrear y documentar cada cambio en el sistema, mejorando la transparencia y confiabilidad en la gestión de la información académica.

### ***5.7 Conclusión Final***

En conclusión, el sistema de gestión y unificación de información académica ha demostrado ser una solución robusta y efectiva para mejorar la administración de datos académicos en el ámbito de Ingeniería en Informática. Al consolidar los registros de los estudiantes en una plataforma centralizada, se logró optimizar los tiempos de procesamiento y consulta, reducir errores tipográficos y mejorar la precisión y accesibilidad de la información.

Los resultados obtenidos, junto con la retroalimentación de los usuarios, confirman que el sistema cumple con los objetivos iniciales y aporta valor al contexto educativo, facilitando la gestión de la información y contribuyendo a una experiencia académica más eficiente y confiable. Con las recomendaciones y futuras ampliaciones propuestas, el sistema tiene el potencial de evolucionar y adaptarse a las necesidades cambiantes de las instituciones educativas, así como de otros sectores que requieran una gestión de datos robusta y precisa.

Esta plataforma establece una base sólida para futuros desarrollos y mejoras, consolidándose como una herramienta clave en la gestión académica y un modelo replicable para otras aplicaciones, permitiendo que la organización aproveche las oportunidades de innovación y mejora en la gestión de sus registros académicos.

## REFERENCIAS BIBLIOGRÁFICAS

[Beck01] Beck, K., Beedle, M., van Bennekum, A., et al. (2001). *Manifesto for Agile Software Development*. Retrieved from <https://agilemanifesto.org/>

[Codd70] Codd, E. F. (1970). *A Relational Model of Data for Large Shared Data Banks*. *Communications of the ACM*, 13(6), 377-387.

[Duckett14] Duckett, J. (2014). *JavaScript and jQuery: Interactive Front-End Web Development*. Indianapolis, IN: Wiley.

[Garrett20] Garrett, T. (2020). *Building Effective Database Applications with CRUD*. New York, NY: TechHub Publishing.

[Inmon19] Inmon, W. H., & O'Neil, B. (2019). *Data Warehousing for Dummies*. New York, NY: Wiley.

[Kroenke18] Kroenke, D. M., & Auer, D. J. (2018). *Database Concepts* (8th ed.). Upper Saddle River, NJ: Pearson.

[Picciano19] Picciano, A. G. (2019). *The Evolution of Big Data and Learning Analytics in American Higher Education*. *Journal of Asynchronous Learning Networks*, 16(3), 7-23.

[Powers18] Powers, D., & Marrone, C. (2018). *Learning PHP, MySQL & JavaScript* (5th ed.). Sebastopol, CA: O'Reilly Media.

[Rahimi21] Rahimi, S., Martin, S., & Tabrizi, M. (2021). *Customizable Solutions for Academic Management Systems*. *Educational Technology & Society*, 24(1), 101-115.

[Walkenbach18] Walkenbach, J. (2018). *Excel 2019 Bible*. Indianapolis, IN: Wiley.