

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“DESARROLLO DE SISTEMA DE INTEGRACIÓN DE DATOS PARA REPORTERÍA USM”

JAVIER IGNACIO AGUAYO SÁNCHEZ

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Andrea Vásquez
Profesor Correferente: Pedro Godoy

Julio - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: DESARROLLO DE SISTEMA DE INTEGRACIÓN DE DATOS PARA REPORTERÍA USM

Nombre del candidato(a): JAVIER IGNACIO AGUAYO SÁNCHEZ

Carrera / Grado: INGENIERÍA CIVIL INFORMÁTICA

Campus: Casa Central Valparaíso; **Departamento:** INFORMÁTICA

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Andrea Vásquez Guerra, en mi calidad de profesora guía del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 20 de ago. de 25; **Firma:**

Estudiante o Candidato(a):

Fecha: 18-08-2025 ; **Firma:**

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Dedicado a mi familia, especialmente a mis padres, que han estado presente en los momentos buenos, y por sobre todo, en los momentos malos. No sería nada sin ellos.

AGRADECIMIENTOS

Primeramente me gustaría agradecer a mi mamá Roxana Sánchez, y a mi papá Carlos Aguayo, quienes me han apoyado en todo mi paso por la universidad, aceptando y respetando mis decisiones, y por sobre todo, dándome todo el amor posible pese a la distancia geográfica.

Agradecer de igual manera a mis hermanas Nataly, Bárbara y Muriel por tenerme presente en sus vidas de una u otra manera, por apoyarme en los momentos difíciles y por quererme incondicionalmente. Agradecer a mis tatas y tíos que me quieren y se enorgullecen por cada logro que cumpla, y finalmente a mi pareja por darme el apoyo y amor en esta última etapa universitaria.

Agradecer a mis amigos y compañeros de la universidad, ellos han sido mi segunda familia en todos estos años que estuve estudiando, ellos fueron y son la parte linda de la universidad, sin su amistad y apoyo en los momentos difíciles no se si lo hubiese podido lograr solo. Soy una persona que ganó y aprendió mucho con ellos.

Hacer una especial mención a mis amigos Maximiliano Abarca, Sebastián López y Christopher Guibout por tenerme presente al momento de cualquier situación, brindándome cualquier tipo de ayuda y/o apoyo en cualquier momento del día. Estuve con ellos en muchos proyectos, trabajos, laboratorios y sesiones de estudio. Valoro y agradezco mucho su amistad.

Agradecer a mi profesora Guía Andrea Vásquez por la paciencia y el apoyo que me brindó incondicionalmente, pese a mis faltas, atrasos y ausencias en este proceso. Partí siendo su alumno de programación en los inicios de carrera, y ahora me orienta y ayuda para ser un profesional. Muchas gracias profesora.

Agradecer a mis compañeros de trabajo -Isabel Ignacio, Martín Barra, Gabriel Quinteros, Daniel Hebel, Camila Villanueva- por demostrarme que a veces un colega de trabajo se puede transformar en un amigo de confianza. Gracias por estos años en el trabajo, por el apoyo y ser un pilar importante en mi crecimiento laboral y personal. Ustedes hacían mucho más amenas mis días en la oficina.

Agradezco a Gonzalo Vásquez y Mauricio Saldivia por su enseñanza y apoyo técnico durante mis años en el trabajo, así como a Pedro Peralta y Sergio Fuentes por su confianza, respaldo y el aprendizaje constante que me brindaron en el trabajo.

Para terminar, un especial agradecimiento a mi mejor amigo Diego Salazar, por ser una increíble persona conmigo, por apoyarme incondicionalmente en todo, por quererme aún cuando estamos lejos y por darme lo necesario para sentirme seguro y confiado en cada meta que alcanzo. Muchas gracias por todo amigo mío.

RESUMEN

La presente memoria aborda el diseño e implementación de un sistema institucional de integración de datos en la Universidad Técnica Federico Santa María (UTFSM), orientado a consolidar información dispersa en múltiples plataformas académicas y administrativas. Se propone una arquitectura automatizada de procesos ETL (Extract, Transform, Load) basada en tecnologías libres como Python, Airflow, Docker y PostgreSQL, permitiendo mejorar la disponibilidad, consistencia y trazabilidad de los datos institucionales.

La solución fue validada a través de su aplicación en los proyectos SAT-E y SIGPI, demostrando mejoras significativas en eficiencia operativa, calidad de datos y soporte a la toma de decisiones. Este trabajo sienta las bases para una gobernanza de datos más robusta y habilita capacidades futuras de analítica avanzada, configurándose como un modelo replicable para otras instituciones de educación superior.

Palabras clave: Integración de datos; ETL; UTFSM; Reportería institucional.

ABSTRACT

This thesis presents the design and implementation of a data integration system at Universidad Técnica Federico Santa María (UTFSM), aimed at consolidating academic and administrative information scattered across multiple platforms. The solution proposes an automated ETL (Extract, Transform, Load) architecture using open-source technologies such as Python, Airflow, Docker, and PostgreSQL, to enhance data availability, consistency, and traceability.

The system was validated through its application in SAT-E and SIGPI institutional projects, showing significant improvements in operational efficiency, data quality, and decision-making support. This work lays the groundwork for stronger data governance and future capabilities in advanced analytics, offering a replicable model for other higher education institutions.

Keywords: Data integration; ETL; UTFSM; Institutional reporting.

Glosario

API

Interfaz de Programación de Aplicaciones (Application Programming Interface). Conjunto de definiciones y protocolos que permite la comunicación entre aplicaciones de software.

Airflow

Plataforma de orquestación de flujos de trabajo desarrollada por Apache, utilizada para programar, monitorizar y gestionar pipelines de datos de forma escalable y modular.

Back-End

Parte lógica de una aplicación que se encarga del procesamiento de datos y operaciones del lado del servidor.

CIAC

Centro Integrado de Aprendizaje en Ciencias Básicas. Unidad institucional encargada de apoyar el aprendizaje en áreas fundamentales como matemáticas, programación, física y química.

CSV

Formato de texto plano que almacena datos en forma tabular, separados por comas.

Data Warehouse

Almacén de datos centralizado que permite almacenar grandes volúmenes de información estructurada para análisis y reportería.

DAG

Directed Acyclic Graph. En Airflow, un DAG define un flujo de tareas y sus dependencias en un orden específico sin ciclos.

Docker

Plataforma que permite empaquetar aplicaciones en contenedores, incluyendo sus dependencias, facilitando su portabilidad y despliegue.

ETL

Extract, Transform, Load. Proceso que extrae datos desde múltiples fuentes, los transforma y carga en un repositorio central para su análisis.

FastAPI

Framework web para Python que permite construir APIs de forma rápida, segura y con validación de tipos.

Kedro

Framework de código abierto para construir proyectos reproducibles de ciencia de datos y pipelines de datos estructurados.

MDT

Master Data Table. Tabla centralizada que consolida múltiples fuentes de datos académicos y administrativos de estudiantes.

Oracle

Sistema de gestión de bases de datos relacional ampliamente usado en entornos empresariales e institucionales.

Pandas

Librería de Python utilizada para el análisis y manipulación de estructuras de datos tabulares (DataFrames).

Pipeline

Secuencia organizada de tareas que procesan datos de manera encadenada desde su origen hasta su destino final.

PostgreSQL

Sistema gestor de bases de datos relacional de código abierto, utilizado como repositorio destino en la propuesta.

Power BI

Herramienta de visualización y análisis de datos desarrollada por Microsoft para construir dashboards y reportes interactivos.

Procesamiento Batch

Ejecución de procesos en lotes que se programan y ejecutan en horarios definidos, sin intervención del usuario.

Python

Lenguaje de programación de alto nivel, ampliamente usado en ciencia de datos, automatización y desarrollo web.

Reportería BI

Proceso de construcción de reportes automatizados para la toma de decisiones, basado en datos estructurados.

SAT-E

Sistema de Acompañamiento a la Trayectoria Estudiantil. Plataforma institucional desarro-

llada por la UTFSM para seguimiento académico de estudiantes.

Scheduler

Componente que programa y coordina la ejecución automática de tareas (por ejemplo, en Airflow o cron).

SCOPUS

Plataforma internacional de indexación de publicaciones científicas y académicas.

SIGPI

Sistema Integrado de Gestión de Profesores e Investigadores. Proyecto de consolidación de información académica y de investigación de la UTFSM.

SQLAlchemy

Librería ORM de Python que permite modelar estructuras de bases de datos relacionales de forma programática.

UTFSM

Universidad Técnica Federico Santa María. Institución de educación superior chilena en la que se implementa el proyecto de integración de datos.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
Glosario	V
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XI
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 Contextualización del entorno	3
1.2 Problemas al integrar datos en instituciones de educación	4
1.2.1 Limitaciones actuales en la reportería institucional	4
1.3 Diagnóstico institucional: UTFSM	5
1.4 Planteamiento del problema y propuesta de solución	7
CAPÍTULO 2: MARCO CONCEPTUAL	9
2.1 Conceptos	9
2.1.1 Integración de datos	9
2.1.2 ETL	17
2.1.3 Data Warehouse y Data Mart	20
2.1.4 Sinergia con los ETLs	21
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	22
3.1 Vista Lógica	23
3.2 Vista de Desarrollo	24
3.3 Vista de Procesos	26
3.4 Vista Física	29
3.5 Vista de Escenarios	31
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	33
4.1 Justificación de las pruebas y criterios de validación	33
4.2 PRUEBAS DE RENDIMIENTO	34
4.2.1 Descripción origen de datos de prueba	34
4.2.2 Descripción de procesos ETL implementados	34
4.2.3 Descripción de ambientes de prueba	36
4.2.4 Metodología de pruebas	36
4.2.5 Resultados de las pruebas de rendimiento	37
4.2.6 Análisis comparativo de desempeño	37
4.2.7 Análisis de resultados	40

4.2.8	Prueba de rendimiento con Airflow	41
4.3	VALIDACIÓN FUNCIONAL EN ESCENARIOS REALES	44
4.3.1	Validación en SAT-E	44
4.3.2	Validación en SIGPI	53
4.4	Síntesis de utilidad	56
4.5	Testimonio de Usuario Clave	58
CAPÍTULO 5: CONCLUSIONES		59
ANEXOS		61
A	Anexo A: Instalación y Configuración de Apache Airflow	61
A.1	Instalación de Airflow en Entorno Docker	61
A.2	Estructura de Carpetas	61
A.3	Inicialización de Airflow	61
A.4	Componentes Principales	61
A.5	Estructura de un DAG típico	62
A.6	Interfaz de Visualización	62
B	Anexo B: Implementación del Proceso ETL en Apache Airflow	63
B.1	DESCRIPCIÓN DEL FLUJO DEL PROCESO ETL	63
B.1.1	Extracción de Datos	63
B.1.2	Transformación de Datos	65
B.1.3	Carga de datos	66
B.1.4	Automatización de Ejecución	68
B.1.5	Creación de flujo ETL en Airflow	69
B.2	Integración con Power BI	70
B.2.1	Pasos para la Conexión	70
B.2.2	Ventajas de la Conexión Directa	70
B.2.3	Consideraciones sobre herramientas de visualización	71
C	Anexo C: Testimonio Completo de Usuario Clave	73
REFERENCIAS BIBLIOGRÁFICAS		74
BIBLIOGRAFÍA		79

ÍNDICE DE FIGURAS

1	Arquitectura conceptual de integración de datos mediante procesos ETL.	8
2	Ejemplo de arquitectura Zero-ETL. Fuente: Artículo "Streamlining Data Integration: Architectures for Real Time Insights and on Demand Transformation", [Shaffi y Sidhick, 2025]	13
3	Esquema general del proceso ETL: extracción, transformación y carga.	18
4	Ejemplo visual Data Warehouse. Fuente: Data Warehouse	20
5	Ejemplo visual Data Mart. Fuente: Data Mart	21
6	Ejemplo modelo de vistas 4+1 Kruchten. Fuente: Modelo 4+1	22
7	Vista lógica del sistema de integración de datos y reportería académica. Fuente: Elaboración propia en PlantUML.	24
8	Vista de desarrollo: estructura del sistema a nivel de módulos. Fuente: Elaboración propia en PlantUML.	25
9	Vista de procesos del sistema: ejecución concurrente y tareas internas en cada flujo ETL. Fuente: Elaboración propia en PlantUML.	27
10	Diagrama de secuencia de ejecución de un DAG en Airflow. Fuente: Elaboración propia	28
11	Vista Física del sistema de integración de datos. Fuente: Elaboración propia.	29
12	Vista de escenarios del sistema: interacción de actores, procesos y aplicaciones consumidoras de datos. Fuente: Elaboración propia.	31
13	Comparación de eficiencia relativa por entorno de ejecución. Fuente: Elaboración propia	38
14	Gráfico resultados máquina 1, 2 y PC personal. Fuente: Elaboración propia.	39
15	Gráfico de puntos, presultados máquina 1, 2 y PC personal. Fuente: Elaboración propia	39
16	Gráfico de puntos, escala log, resultados máquina 1, 2 y PC personal. Fuente: Elaboración propia	40
17	Gráfico de barra; prueba de rendimiento 4 ambientes. Elaboración propia	42
18	Ejemplos de modelos utilizados para análisis. Fuente: Poster SAT-E: Dirección Transformación Digital - DATA-E	45
19	Extracto de pipeline de integración y transformación de datos con Kedro. Fuente: Elaboración propia	46
20	Ejemplo de Reporte ejecutivo de rendimiento académico en Power Bi. Fuente: Dirección Transformación Digital	48
21	Ejemplo de Reporte de evaluación rendimiento academico en Power Bi. Fuente: Dirección Transformación Digital	49
22	Ejemplo de Reporte de seguimiento a la trayectoria academica estudiantil en Power Bi. Fuente: Dirección Transformación Digital	50
23	Información Dirección de Acompañamiento en la Trayectoria Académica Estudiantil - DATA-E. Fuente: Comunicaciones USM	52
24	Modelo de datos SIGPI de Alto nivel. Fuente: Martín Barra - Funcionario USM	53
25	Visualización plataforma SIGPI. Fuente: SIGPI	55
26	Visualización perfil plataforma SIGPI. Fuente: SIGPI	55

27	Interfaz de Apache Airflow mostrando ejecución de DAGs. Fuente: Elaboración propia.	63
28	Transformación de datos en el proceso ETL. Fuente: Elaboración propia	65
29	Indices de herramientas Airflow. Fuente: Airflow docs	68
30	Descripción de logs Airflow. Fuente: Airflow docs	68
31	Variable y parámetros de programación de tareas en Airflow. Fuente: Airflow docs	68
32	Representación visual de un ETL en Airflow. Fuente: ETL en Airflow	69
33	Flujo de integración entre PostgreSQL y Power BI para visualización de datos. Fuente: Elaboración propia	71

ÍNDICE DE TABLAS

1	Definiciones representativas de integración de datos.	9
2	Ventajas y desventajas del enfoque Human-in-the-Loop (HITL)	11
3	Ventajas y desventajas del modelo Zero-ETL	14
4	Ventajas y desventajas del modelo ETL tradicional	15
5	Beneficios técnicos del proceso ETL	19
6	Desafíos técnicos del proceso ETL	20
7	Descripción de tablas utilizadas en pruebas. Fuente: Elaboración propia	34
8	Especificaciones de ambientes de prueba. Fuente: Elaboración propia	36
9	Atributos de tablas utilizadas en pruebas de rendimiento. Fuente: Elaboración propia	37
11	Resumen comparativo de desempeño por entorno. Fuente: Elaboración propia	37
10	Resultados máquina 1, 2 y PC personal. Fuente: Elaboración propia	38
12	Tiempos de ejecución con Apache Airflow. Fuente: Elaboración propia	41
13	Resumen comparativo de tiempos de ejecución en diferentes entornos. Elaboración propia	43
14	Resumen comparativo de validación funcional en SAT-E y SIGPI. Fuente: Elaboración propia	56
15	Resumen de aportes y beneficios evaluados de la solución propuesta. Fuente: Elaboración propia	57

INTRODUCCIÓN

En la última década, las instituciones de educación superior han enfrentado una creciente presión por modernizar su gestión académica, administrativa y estudiantil. Este fenómeno responde tanto al avance de la transformación digital como a las exigencias de políticas públicas orientadas a la inclusión, equidad y aseguramiento de la calidad educativa. En este contexto, los datos se han convertido en un insumo estratégico, y la capacidad de integrarlos, analizarlos y explotarlos de forma oportuna es hoy una condición indispensable para la toma de decisiones informadas [Pelletier y Review, 2021].

No obstante, las universidades han evolucionado como organizaciones con estructuras funcionales independientes, acumulando a lo largo de los años sistemas y tecnologías diversas que no fueron concebidas para interoperar entre sí. Esto ha generado un ecosistema fragmentado, con múltiples fuentes de datos (académicas, administrativas, sociales, externas como DEMRE o Scopus, y otras internas como BANNER, SIGA, AULA Virtual o SharePoint) que operan de forma desarticulada. La duplicidad, la inconsistencia y la dificultad para acceder a datos consolidados no sólo entorpecen los procesos de gestión, sino que limitan la posibilidad de realizar cruces entre variables críticas, obstaculizando iniciativas como el seguimiento a la trayectoria estudiantil o la generación de alertas tempranas [Kimball y Ross, 2013, Amrullah et al., 2021].

La Universidad Técnica Federico Santa María (UTFSM) no es ajena a esta realidad. En los últimos años, ha impulsado distintas iniciativas orientadas a mejorar la permanencia y el rendimiento de los estudiantes mediante programas de acompañamiento como el CIAC, el PNAI o el SAT-E. Sin embargo, la ausencia de una plataforma unificada de integración de datos ha limitado la capacidad institucional de actuar proactivamente ante riesgos académicos, dificultando la trazabilidad, la focalización de apoyos y el uso sistemático de la información disponible. Esta situación se ve agravada por la descentralización de la información diagnóstica y la carencia de procesos estandarizados que permitan articular respuestas oportunas desde distintas unidades académicas y administrativas.

En respuesta a estos desafíos, esta memoria propone el diseño e implementación de un sistema de integración de datos institucional basado en procesos ETL (Extract, Transform, Load), capaz de centralizar información dispersa en un repositorio común y generar reportes e indicadores accionables mediante herramientas como Power BI. La solución se apoya en tecnologías libres como Python, Airflow, PostgreSQL y Docker, y está diseñada para ser escalable, trazable y sostenible en el tiempo. Esta arquitectura permite no solo optimizar el uso de los recursos institucionales, sino también fortalecer la cultura organizacional hacia una gestión basada en datos y orientada a resultados.

Para documentar la propuesta de forma estructurada, se adopta el modelo arquitectónico 4+1, que permite describir el sistema desde cinco vistas complementarias: lógica, de desarrollo, de procesos, física y de escenarios. Esta metodología no solo facilita la comprensión

técnica del sistema, sino que también permite evidenciar su impacto estratégico y su alineación con las metas institucionales de mejora continua y calidad educativa.

La estructura de esta memoria está organizada en tres partes principales. En primer lugar, se examinan los fundamentos conceptuales y normativos de la integración de datos, abordando los métodos más convenientes en el contexto, junto con sus respectivos beneficios y desafíos. Posteriormente, se describe el diseño e implementación del sistema, detallando su arquitectura, tecnologías utilizadas, procesos y casos de validación institucional (SAT-E y SIGPI). Finalmente, se presentan los resultados, análisis de impacto y recomendaciones para su escalabilidad y sostenibilidad futura.

En definitiva, esta investigación busca aportar a la construcción de una universidad más eficiente, donde los datos se transformen en conocimiento útil para orientar decisiones, intervenir oportunamente y avanzar hacia una educación más equitativa, eficiente y centrada en las trayectorias reales de los estudiantes como en las modalidades de transmisión de información a su comunidad.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

En todo el mundo, las instituciones de educación superior han invertido crecientes recursos en tecnología para mejorar su administración académica y operativa. Sin embargo, muchas de ellas continúan enfrentando la fragmentación de sus sistemas de información, donde diferentes unidades —como admisión, registro académico, finanzas o bibliotecas— utilizan plataformas dispares que no interoperan entre sí [Jakimoski, 2016]. Esta falta de integración genera retrabajo, errores y una sobrecarga administrativa que impide dedicar tiempo valioso a tareas sustantivas como la investigación o el acompañamiento estudiantil [Pelletier y Review, 2021].

En América Latina, esta situación se agrava por la escasez de estándares compartidos, la debilidad de los marcos institucionales y la alta dependencia de procesos manuales. En Chile, la proliferación de oficinas de análisis institucional ha sido una respuesta organizacional a la complejidad del entorno y la presión por mejorar la autorregulación institucional [Rivera *et al.*, 2009]. Sin embargo, muchas de estas unidades aún enfrentan dificultades debido a la falta de infraestructura tecnológica adecuada para integrar y procesar datos de manera eficiente.

1.1. Contextualización del entorno

La transformación digital ha modificado profundamente la forma en que las universidades gestionan sus operaciones. Aunque se han incorporado sistemas complejos para la administración académica, docencia virtual y gestión financiera, el desarrollo ha sido frecuentemente desarticulado: cada unidad adopta plataformas según necesidades locales, generando fragmentación tecnológica [Jakimoski, 2016, Rivera *et al.*, 2009]. Esto obstaculiza la interoperabilidad, la gobernanza de datos y la visión transversal del quehacer institucional [Pelletier y Review, 2021].

Cuando se aborda con una visión sistémica, la transformación digital permite consolidar datos de múltiples fuentes para construir modelos de alerta temprana, personalizar el aprendizaje y mejorar la toma de decisiones. Para ello, se requieren infraestructuras automatizadas y auditables que recojan, transformen y centralicen información bajo criterios comunes [Sedlmayr *et al.*, 2022].

1.2. Problemas al integrar datos en instituciones de educación

1. **Heterogeneidad de sistemas y fuentes de datos:** coexistencia de plataformas disímiles que dificultan la interoperabilidad e incrementan errores y duplicidad [Mosse et al., 2023].
2. **Falta de estándares comunes:** la ausencia de esquemas unificados complica los procesos de igualación semántica (*schema matching*) y normalización [Amrullah et al., 2021].
3. **Limitaciones técnicas y presupuestarias:** escasez de personal especializado, presupuesto limitado y conflictos internos dificultan el desarrollo de soluciones sostenibles [Mosse et al., 2023].
4. **Fragmentación organizacional:** baja coordinación entre unidades y falta de acuerdos sobre procesos y definiciones de datos [Neang et al., 2021].
5. **Desafíos de privacidad y seguridad:** incorporación de nuevas tecnologías sin protocolos claros genera riesgos de protección de datos personales [Román-Cañizares et al., 2020].
6. **Calidad e incertidumbre de los datos:** errores en registros, inconsistencias o formatos incompatibles dificultan el análisis confiable [Dong et al., 2007].
7. **Dependencia de integración humana:** participación manual en procesos clave (validación, limpieza, correspondencia de entidades), lo que limita la escalabilidad y trazabilidad [Li, 2017].

1.2.1. Limitaciones actuales en la reportería institucional

En el contexto universitario, la capacidad de generar reportería analítica oportuna, confiable y automatizada es clave para apoyar la toma de decisiones estratégicas, académicas y operativas. Sin embargo, en muchas instituciones —incluida la que motiva esta memoria— persisten importantes limitaciones en esta área. Actualmente, la generación de reportes suele requerir esfuerzos manuales, extraer datos desde múltiples plataformas de forma independiente y realizar validaciones puntuales sin trazabilidad formalizada.

Esta situación genera varias consecuencias negativas: retrasos en la entrega de indicadores, inconsistencias entre áreas, dificultades para monitorear el rendimiento estudiantil y escasa capacidad de reacción ante riesgos académicos. Además, impide implementar herramientas más avanzadas como sistemas de alerta temprana, dashboards dinámicos o modelos predictivos, los cuales requieren datos consolidados y confiables.

La literatura especializada destaca que una reportería efectiva, sustentada en procesos de integración de datos robustos, permite personalizar intervenciones, mejorar la planificación institucional y fortalecer la resiliencia organizacional frente a contextos cambiantes [Gong, 2025, Kolse et al., 2024, Elugbaju et al., 2024]. La brecha entre este escenario ideal y la realidad observada constituye uno de los núcleos del problema abordado en esta memoria.

1.3. Diagnóstico institucional: UTFSM

La Universidad Técnica Federico Santa María (UTFSM) enfrenta todos los desafíos mencionados. Actualmente, coexisten sistemas como BANNER, SIGA, AULA Virtual, SharePoint, etc. y fuentes externas como información DEMRE. Esta diversidad impide una integración fluida, genera tareas manuales, reduce la reproducibilidad de reportes y limita el monitoreo oportuno.

En ámbitos críticos como el acompañamiento a la trayectoria estudiantil, esta fragmentación impide consolidar información relevante (notas, caracterización, actividad docente) para generar alertas tempranas o acciones coordinadas. Los programas de apoyo —como CIAC, PNAI o Propedéutico— operan de manera desarticulada y sin integración de datos.

Estudios internos evidencian que, aunque existe información clave sobre rendimiento y contexto, esta no se vincula efectivamente con los procesos de intervención académica. La tasa de retención del 76,7 % en primer año es menor a la de universidades pares¹, y se asocia a la imposibilidad de actuar a tiempo sobre estudiantes en riesgo. La falta de integración, además, impide realizar evaluaciones longitudinales, establecer protocolos de derivación o generar visualizaciones personalizadas.

Testimonio institucional

Uno de los ejemplos más ilustrativos sobre las limitaciones del modelo de integración manual en la UTFSM fue relatado por Giannina Urbina, Subdirectora de Admisión de Postgrado y Formación Continua. En su testimonio, se detallan las múltiples dificultades que enfrentaba el equipo encargado de construir alertas académicas tempranas, especialmente durante los primeros semestres.

El proceso iniciaba con la necesidad de identificar estudiantes en riesgo académico a partir de las notas obtenidas en asignaturas del ciclo básico (Matemáticas, Física, Química), consideradas críticas en carreras de ingeniería. Sin embargo, estas calificaciones no estaban centralizadas en ningún sistema institucional, lo que obligaba al equipo a solicitarlas manualmente mediante correos electrónicos enviados a coordinadores y profesores responsables.

¹SIES, 2024. Retención de 1er año (Pregrado), cohorte 2019 a 2023 [SIES 2024](#)

1. Origen y formato de los datos

Los datos académicos eran extraídos directamente desde el Aula Virtual, pero el acceso no era estandarizado: muchos profesores subían las notas en archivos PDF, otros utilizaban planillas Excel con estructuras distintas, y en algunos casos ni siquiera registraban la información formalmente. Esto obligaba a realizar descargas individuales por asignatura y paralelo, transformando manualmente cada archivo para extraer los valores relevantes. En consecuencia, cada cohorte requería la revisión de decenas (o incluso cientos) de archivos heterogéneos.

2. Consolidación y limpieza

Una vez recolectados los documentos, el equipo debía consolidar los datos en una base común. Este proceso implicaba:

- Comparar manualmente las notas con la lista oficial de estudiantes inscritos en cada sección.
- Resolver incongruencias de formato (e.g., nombres sin RUT, identificadores incompletos).
- Validar registros incompletos o mal digitalizados.
- Normalizar datos para obtener indicadores como promedio, desviación estándar, máximo y mínimo por curso.

No existían protocolos estandarizados para el envío ni el formato de la información, lo que aumentaba el tiempo requerido para la limpieza y reducía la calidad final de la base consolidada.

3. Impacto en los tiempos de respuesta

El proceso de consolidación podía tardar semanas o incluso meses, especialmente en carreras con múltiples paralelos. Aunque por reglamento los docentes tenían un plazo de dos semanas para ingresar calificaciones, en la práctica muchas notas se entregaban informalmente o quedaban alojadas únicamente en documentos no estructurados. Esta demora hacía que las alertas académicas —aunque basadas en modelos robustos— se generaran con posterioridad al momento crítico de intervención.

4. Reutilización y solicitudes adicionales

Además del monitoreo académico general, otras unidades institucionales como DEFINDER² (para estudiantes con becas deportivas) o el Programa PACE³ solicitaban alertas personalizadas. Sin embargo, dado que el proceso inicial era tan costoso en tiempo y recursos, estas solicitudes solo podían atenderse una vez finalizada la consolidación general. En consecuencia, los informes llegaban tarde y no permitían activar mecanismos de apoyo de forma oportuna.

5. Dificultades técnicas y escalabilidad

El uso de Excel como herramienta principal de consolidación generaba problemas de

²Departamento de Educación Física, Deportes y Recreación

³Programa de Acompañamiento y Acceso Efectivo

rendimiento con bases grandes (por cohorte y por carrera), dificultando incluso tareas básicas como calcular promedios o aplicar filtros. Además, el esfuerzo humano invertido no era sostenible: el éxito del proceso dependía casi exclusivamente del compromiso individual del equipo encargado, sin respaldo institucional sistematizado.

6. Conclusiones del equipo

A pesar de las limitaciones del modelo, los análisis realizados demostraban alta correlación entre las alertas generadas (nota 1 o 2) y los resultados finales del semestre. Esto validaba la lógica del modelo predictivo, pero evidenciaba que su utilidad se veía profundamente limitada por los tiempos de procesamiento y la falta de integración automatizada. Como señaló el equipo, “si la información hubiera estado disponible a tiempo, se habría podido apoyar a muchos más estudiantes”.

Síntesis

Este testimonio evidencia de forma concreta cómo los enfoques de integración manual, aunque funcionales en pequeñas escalas, resultan insostenibles en contextos institucionales amplios. La dependencia de tareas manuales, la falta de trazabilidad, los errores en la depuración y los tiempos extensos de consolidación hacen inviable su mantenimiento en el tiempo. La situación descrita refuerza la necesidad urgente de contar con una arquitectura de integración de datos automatizada, escalable y trazable.

1.4. Planteamiento del problema y propuesta de solución

La UTFSM requiere superar la fragmentación de sus sistemas de información mediante una solución de integración de datos escalable, trazable y automatizada. Para ello, se propone implementar una arquitectura basada en procesos ETL que permita consolidar información proveniente de múltiples plataformas institucionales. Estas incluyen:

- **BANNER:** sistema de gestión académica implementado por la UTFSM a través de la solución de Ellucian, utilizado para administrar matrículas, asignaturas, programas de estudio y datos de estudiantes [Ellucian, 2024].
- **SIGA (Sistema de Información de Gestión Académica):** plataforma desarrollada institucionalmente que permite registrar calificaciones, controlar carga académica, administrar programas y coordinar procesos académicos y docentes [Universidad Técnica Federico Santa María, 2025d].
- **AULA Virtual:** entorno de docencia virtual de la UTFSM, basado en Moodle, que permite gestionar contenidos de asignatura, evaluaciones, interacción estudiantil y seguimiento académico [Universidad Técnica Federico Santa María, 2025a].

- **SharePoint:** sistema corporativo utilizado para la centralización de documentos, flujos administrativos, formularios, proyectos y archivos compartidos entre unidades institucionales [Universidad Técnica Federico Santa María, 2025b].

Actualmente, estas plataformas funcionan de forma independiente, lo cual genera una alta carga operativa, duplicación de esfuerzos y dificultades para realizar análisis transversales o generar alertas oportunas sobre la trayectoria estudiantil. Por ejemplo, los reportes de productividad académica o de seguimiento de estudiantes en riesgo deben construirse manualmente, extrayendo datos desde cada sistema, depurándolos y combinándolos con herramientas ofimáticas antes de visualizarlos en Power BI, lo que conlleva demoras, errores y baja reproducibilidad.

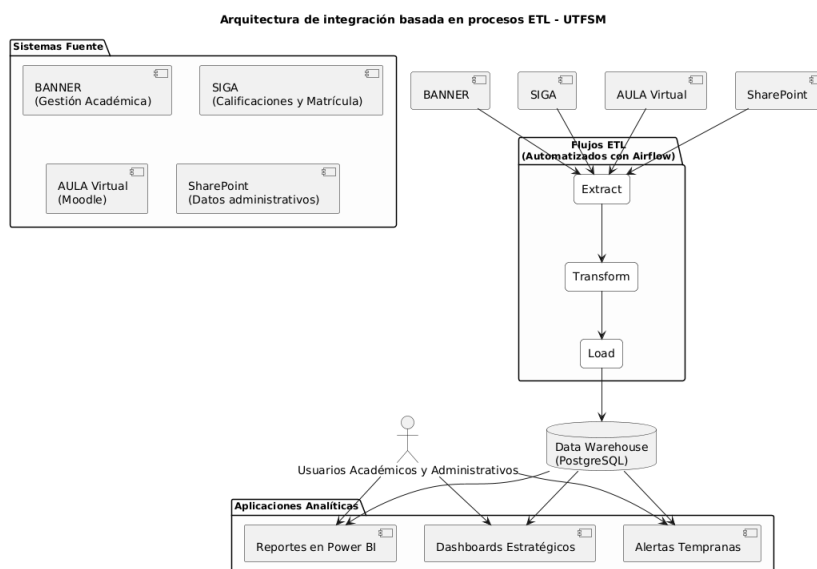


Figura 1: Arquitectura conceptual de integración de datos mediante procesos ETL.

La solución propuesta busca abordar este problema mediante la construcción de un sistema ETL automatizado utilizando tecnologías libres como Python, PostgreSQL, Docker y Apache Airflow. Este sistema permitirá orquestar la extracción, transformación y carga de datos de forma periódica, documentada y escalable, sentando las bases para el desarrollo de capacidades institucionales como:

- Reporterías académicas automatizadas.
- Modelos de alerta temprana basados en trayectorias estudiantiles.
- Análisis institucional longitudinal y toma de decisiones estratégicas basadas en datos.

Con esta arquitectura, la UTFSM podrá evolucionar hacia una gestión académica proactiva, basada en información integrada, oportuna y trazable.

CAPÍTULO 2

MARCO CONCEPTUAL

Para entender de mejor manera el contexto de esta memoria, es muy importante establecer ciertas definiciones y conceptos tecnológicos, para así llegar a la conclusión de cuál es el camino a tomar para solucionar el problema mencionado en el capítulo anterior.

2.1. Conceptos

2.1.1. Integración de datos

En la literatura se han propuesto diversas definiciones que coinciden en la idea de que la integración de datos consiste en combinar información proveniente de múltiples fuentes, con el fin de proporcionar una vista unificada y coherente. A continuación, se presentan algunas definiciones representativas:

La integración de datos es el proceso de combinar datos que residen en diferentes fuentes y proporcionar al usuario una vista unificada de estos datos [Badrinath y G, 2024].

La integración de datos es un proceso de reunir diversas formas y fuentes de datos —que difieren en naturaleza, esquemas y plataformas— para consolidarlas en un único entorno [Badrinath y G, 2024].

La integración de datos implica recolectar datos desde múltiples fuentes heterogéneas, aplicar procesos técnicos como conversión y transformación, y almacenarlos finalmente de forma centralizada para obtener una visión única de la información [Sreemathy *et al.*, 2021].

Tabla 1: Definiciones representativas de integración de datos.

Enfoques de integración

Existen dos formas principales de abordar la integración de datos:

- **Integración lógica:** consiste en entregar una interfaz común que permite acceder a múltiples bases de datos sin necesidad de mover físicamente los datos. Esto se logra mediante técnicas como la reescritura de consultas, siendo útil cuando se desea mantener la autonomía de los sistemas fuente [Deen *et al.*, 1987].
- **Integración física:** implica migrar y transformar los datos desde diversas fuentes hacia un modelo común y centralizado, típicamente en un *data warehouse*. Esta moda-

lidad es frecuente en entornos donde se requiere eficiencia en análisis y reportería [Badrinath y G, 2024].

Desafíos técnicos de la integración

Integrar datos de manera efectiva conlleva superar una serie de desafíos técnicos relevantes:

- **Inconsistencia y duplicación:** los datos pueden diferir en formato, precisión o granularidad entre fuentes, lo que requiere procesos de depuración y estandarización [Deen *et al.*, 1987].
- **Escalabilidad y volumen:** el aumento exponencial del volumen de datos genera retos para su procesamiento eficiente, especialmente en ambientes distribuidos o en arquitecturas de *Big Data* [Badrinath y G, 2024].
- **Heterogeneidad estructural y semántica:** las fuentes pueden usar distintos esquemas, tipos de datos, unidades de medida o incluso terminologías diferentes para referirse a conceptos similares. Esto exige mecanismos de mapeo semántico y transformación de esquemas [Deen *et al.*, 1987].
- **Limitaciones de la automatización:** muchas decisiones sobre si dos registros representan la misma entidad, o si ciertos atributos son equivalentes, no pueden ser resueltas únicamente por algoritmos. En este contexto, la intervención humana se vuelve necesaria para validar correspondencias y mejorar la calidad del proceso [Li, 2017].

Análisis comparativo de métodos de integración de datos

Para abordar el problema de la fragmentación, inconsistencia y dispersión de datos en instituciones educativas, es necesario analizar críticamente los distintos enfoques disponibles para la integración de información. Este análisis parte de un escenario común en muchas instituciones educativas: la integración de datos de manera "manual", también conocida como integración humana o *human-in-the-loop* (HITL). A partir de esta problemática se presentan dos posibles enfoques tecnológicos en el contexto de datos en instituciones para su superación: la integración en tiempo real (representado por un modelo Zero-ETL) y la integración tradicional basada en procesos ETL.

Integración humana: el método problema

La integración humana, también conocida como *human-in-the-loop* (HITL), corresponde a un enfoque en que los operadores humanos participan directamente en el ciclo de integración de datos, usualmente en las etapas de validación, corrección, interpretación o resolución de

ambigüedades. Es comúnmente utilizado en contextos donde existen datos no estandarizados, incompletos o ambiguos, y donde el juicio humano resulta necesario para garantizar precisión o contextualización [Li, 2017].

En entornos educativos, el enfoque HITL ha sido aplicado en distintas situaciones:

- **Evaluación docente y retroalimentación estudiantil:** sistemas como los diseñados por [Chen et al., 2023] incorporan la revisión manual de encuestas, correcciones de inconsistencias y categorización por parte de expertos antes de la integración.
- **Sistemas adaptativos de enseñanza:** plataformas como Carnegie Learning y DreamBox Education integran la participación docente para validar o ajustar las recomendaciones generadas por algoritmos, mejorando así la efectividad pedagógica [Jantanukul, 2024].
- **Consolidación de indicadores institucionales:** es frecuente que los equipos de análisis revisen manualmente informes de unidades académicas heterogéneas para homologar definiciones, formatos y rangos antes de reportar indicadores clave [Sharwani, 2024].

Este tipo de integración también se ha utilizado en procesos de emparejamiento de registros históricos, gestión de excepciones en la carga de datos y revisión previa a la publicación de resultados institucionales.

Ventajas	Desventajas
Permite validar datos ambiguos o erróneos que los sistemas automáticos no pueden resolver.	Alta dependencia de operadores humanos, lo que reduce la escalabilidad del sistema [Li, 2017].
Aporta juicio experto en contextos con alta variabilidad semántica o cultural.	Posibilidad de errores por fatiga, sesgos subjetivos o falta de estandarización.
Mejora la aceptabilidad institucional de los resultados al contar con revisión humana directa.	Elevados costos de mantenimiento, dificultades para trazar decisiones y escasa automatización de los flujos.

Tabla 2: Ventajas y desventajas del enfoque Human-in-the-Loop (HITL)

Síntesis Si bien HITL aporta flexibilidad contextual y precisión en escenarios complejos, sus principales limitaciones —baja escalabilidad, alta dependencia operativa, escasa trazabilidad y alto costo— lo convierten en una opción poco sostenible para procesos de integración de datos a nivel institucional. Por ello, esta memoria propone reemplazar este enfoque por mecanismos sistematizados, auditables y automatizables, que permitan integrar datos de forma eficiente, confiable y alineados con las capacidades técnicas disponibles en el entorno universitario.

Alternativas tecnológicas de integración de datos

La dependencia de procesos manuales en la integración de datos —representada por el enfoque *human-in-the-loop*— ha sido históricamente útil para resolver ambigüedades, validar registros o adaptar datos con alta variabilidad semántica. Sin embargo, en contextos institucionales donde la escala, la consistencia y la trazabilidad de los datos son fundamentales, este enfoque se vuelve cada vez más limitado e insostenible. La creciente necesidad de generar reportes automatizados, alimentar sistemas de analítica en tiempo real y cumplir con estándares de calidad de datos ha impulsado la búsqueda de soluciones tecnológicas más eficientes.

En este escenario, han surgido distintas arquitecturas de integración orientadas a sistematizar y automatizar el flujo de datos, disminuyendo la intervención humana y fortaleciendo la gobernanza de la información. Entre estas, destacan especialmente dos enfoques:

- **La integración en tiempo real (Zero-ETL):** un modelo emergente basado en streaming, consultas federadas y transformación “en el momento”, que promete agilidad y flexibilidad para entornos dinámicos.
- **La integración tradicional mediante procesos ETL (Extract, Transform, Load):** un enfoque consolidado que permite controlar, validar y transformar datos antes de cargarlos en repositorios analíticos, asegurando trazabilidad y calidad.

Ambos enfoques han sido aplicados en distintos contextos educativos, con resultados, ventajas y limitaciones que deben ser evaluados cuidadosamente antes de definir una arquitectura de integración institucional. A continuación, se analiza cada uno de estos modelos para sustentar la decisión metodológica adoptada en esta memoria.

Integración en tiempo real con modelo Zero-ETL

El enfoque de integración en tiempo real, también conocido como *Zero-ETL*, surge como una evolución frente a las limitaciones de los procesos de integración tradicionales. Mientras que ETL requiere extraer, transformar y cargar datos en procesos por lotes, Zero-ETL minimiza el movimiento de datos y permite su integración directamente en el momento de la consulta, habilitando flujos de análisis inmediato y respuestas dinámicas [Shaffi y Sidhick, 2025].

Principios del modelo Zero-ETL Este modelo se apoya en tecnologías como la virtualización de datos, *query-time transformation*, *streaming* en tiempo real y consultas federadas. En lugar de transformar y almacenar datos de manera anticipada, los datos se consultan en su estado original y son transformados ‘en el momento’, según la necesidad del usuario o aplicación [Shaffi y Sidhick, 2025, DataCamp, 2024].

Figure 1: An example of Zero-ETL Architecture

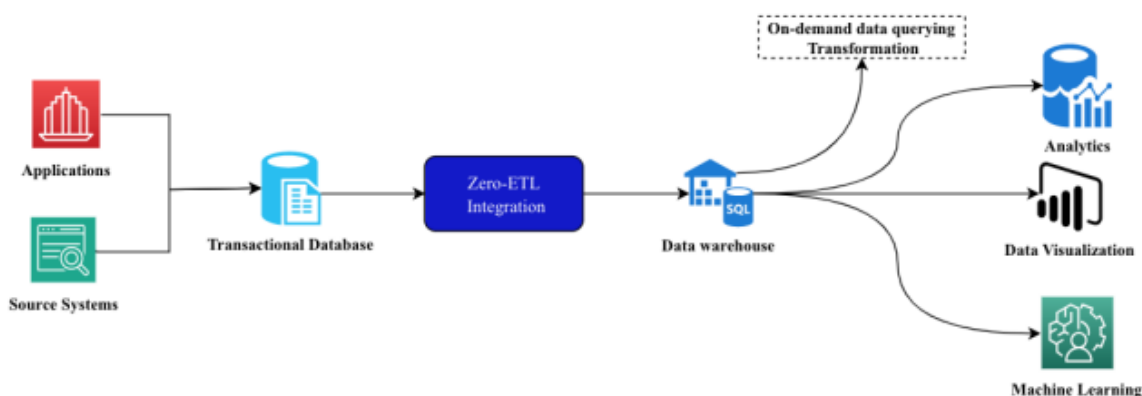


Figura 2: Ejemplo de arquitectura Zero-ETL. Fuente: Artículo "Streamlining Data Integration: Architectures for Real Time Insights and on Demand Transformation", [Shaffi y Sidhick, 2025]

Entre sus componentes clave se encuentran:

- **Streaming en tiempo real:** uso de plataformas como Apache Kafka o Amazon Kinesis para capturar eventos al instante.
- **Motores de consulta dinámica:** como Redshift o Snowflake, que permiten transformar datos al vuelo.
- **Virtualización de datos:** capa lógica que permite consultar fuentes heterogéneas como si fueran una sola.

Aplicaciones en contextos educativos Aunque la mayoría de los casos de uso documentados corresponden a sectores como el financiero, la salud y el comercio electrónico (personalización, monitoreo de inventario, etc.), existen algunos casos en ámbitos de la educación bajo la consigna de los siguientes contextos:

- **Alertas académicas en tiempo real:** detección de estudiantes en riesgo mediante el monitoreo continuo de su actividad en plataformas como Moodle.
- **Evaluación continua y adaptativa:** análisis inmediato del rendimiento en evaluaciones, adaptando contenidos o intervenciones.
- **Monitoreo institucional:** paneles dinámicos con actualización por evento para indicadores clave (asistencia, uso de servicios, carga docente).

Ventajas	Desventajas
Acceso inmediato a información actualizada.	Requiere infraestructura de baja latencia y motores de procesamiento en tiempo real.
Simplificación arquitectónica sin necesidad de procesos ETL complejos.	Limitado control de calidad, validación y trazabilidad de los datos.
Alta flexibilidad frente a nuevas fuentes o cambios estructurales.	Baja compatibilidad con sistemas institucionales legados.

Tabla 3: Ventajas y desventajas del modelo Zero-ETL

Casos documentados

Caso 1: Universidad de Purdue – Plataforma Signals La Universidad de Purdue implementó un sistema de alerta temprana basado en datos recolectados en tiempo real desde su LMS. El sistema “Signals” monitoreaba la actividad académica y el comportamiento del estudiante en plataformas como Moodle o Blackboard. Utilizando flujos de integración en tiempo real, generaba alertas automáticas sobre estudiantes en riesgo de bajo rendimiento, lo cual permitía a docentes y tutores tomar acciones inmediatas [Arnold y Pistilli, 2010]. *Resultado:* Se reportó una mejora del 21 % en la retención académica de estudiantes intervenidos respecto al grupo de control.

Caso 2: UCL (University College London) – Plataforma de analítica en vivo UCL implementó una arquitectura de datos que integraba eventos de navegación y entrega de evaluaciones con Moodle, Turnitin y Echo360, usando un enfoque *streaming-based*. Esto permitió construir dashboards dinámicos que ayudaban a los docentes a detectar patrones emergentes de participación y adaptar sus estrategias pedagógicas en tiempo real [Vouladonis et al., 2023].

Integración tradicional mediante procesos ETL

Frente a las limitaciones de la integración manual o basada en intervención humana, el modelo tradicional de procesos ETL (Extract, Transform, Load) representa una alternativa robusta, estructurada y ampliamente validada para consolidar datos en entornos institucionales.

Principios del modelo ETL El enfoque ETL propone extraer datos desde fuentes heterogéneas, transformarlos según reglas semánticas y de negocio, y cargarlos en un repositorio central. Este modelo permite una transformación profunda de los datos antes de ser utilizados, incluyendo limpieza, normalización, enriquecimiento, validación y trazabilidad. Esta arquitectura es compatible con sistemas relacionales, archivos planos, APIs y otras fuentes institucionales heredadas [Sreemathy et al., 2021].

Aplicaciones en contextos educativos Los procesos ETL son ampliamente utilizados en instituciones de educación superior para:

- **Consolidar datos académicos y administrativos** desde plataformas como MySQL, PostgreSQL, CRMs, Oracle y Excel.
- **Construir data warehouses institucionales** para análisis longitudinales y reportería.
- **Alimentar dashboards estratégicos** con información validada para acreditación, retención y desempeño estudiantil.

Ventajas	Desventajas
Permite una transformación compleja y validada de los datos.	Procesamiento en lotes: no ofrece acceso en tiempo real.
Compatible con fuentes diversas y sistemas legados.	Requiere tiempo de diseño y mantenimiento del flujo.
Proporciona control total sobre la trazabilidad, gobernanza y calidad del dato.	Necesita orquestación mediante herramientas específicas (ej. Airflow, Talend).

Tabla 4: Ventajas y desventajas del modelo ETL tradicional

Casos documentados

Caso 1: Universidad Tecnológica de Malasia (UTM) UTM desarrolló un sistema de integración basado en ETL que permitió consolidar más de 500 tablas provenientes de plataformas internas como GSMS, AIMS y RADIS. El sistema realizaba tareas automatizadas de limpieza, validación, transformación y carga en un repositorio analítico común. Esto permitió generar reportes estratégicos con mayor rapidez y confiabilidad, optimizar la planificación académica y eliminar duplicaciones entre sistemas [Soriano y Cruz, 2020]. *Resultado:* Reducción de tiempos de generación de reportes de días a minutos; aumento en la precisión de datos para decisiones estratégicas; interoperabilidad entre plataformas dispares.

Caso 2: Universidad de Tarapacá (Chile) La Universidad de Tarapacá diseñó una arquitectura basada en procesos ETL y modelos OLAP para consolidar información académica, administrativa y financiera. Esta integración sirvió como base para sistemas de reportería orientados a procesos de acreditación, análisis docente, seguimiento estudiantil y evaluación de desempeño institucional [Tapia y Pinto, 2010]. *Resultado:* Aumento significativo de la trazabilidad, automatización de dashboards, mejora en la planificación operativa y en la calidad de los datos reportados a entidades externas.

Caso 3: Universidad Arturo Prat (Chile) Esta universidad implementó un Data Mart institucional con soporte ETL para consolidar indicadores de productividad académica y análisis estratégico. La plataforma resultante permitió mejorar la eficiencia de los procesos de toma

de decisiones y estructurar la información para su uso en sistemas OLAP y dashboards de gestión [Medina et al., 2018]. *Resultado:* Creación de una plataforma robusta de reportería institucional; consolidación de datos dispersos en una vista única para usuarios estratégicos.

Caso 4: Proyecto DIFUTURE (Alemania) En el contexto de redes europeas de investigación clínica, el proyecto DIFUTURE utilizó procesos ETL sobre estándares FHIR y OMOP-CDM para integrar datos hospitalarios provenientes de distintos sistemas. Esta infraestructura permitió a universidades y hospitales alemanes participar en colaboraciones científicas a gran escala, mejorando la interoperabilidad y la calidad de sus datos clínicos [Kohlbacher et al., 2018, Sedlmayr et al., 2022]. *Resultado:* Alineación con estándares internacionales, interoperabilidad semántica y mejora en la colaboración científica interdisciplinaria.

Caso 5: Universidades africanas y asiáticas Estudios recientes demuestran que la adopción de procesos ETL en universidades emergentes ha contribuido al fortalecimiento de sus sistemas de gestión académica y productividad investigativa. Por ejemplo, instituciones en Nigeria e Irán han utilizado técnicas de integración para consolidar repositorios de datos institucionales, facilitando el seguimiento del rendimiento docente, la gestión curricular y la producción científica [Abhulimen et al., 2024, Akrami et al., 2024]. *Resultado:* Mejora de la eficiencia operativa, incremento en la visibilidad internacional y profesionalización de los sistemas de análisis institucional.

Justificación técnica y estratégica del modelo ETL

Si bien la integración en tiempo real (Zero-ETL) ha demostrado ser útil en plataformas adaptativas o sistemas de alerta inmediata, su implementación generalizada en instituciones de educación enfrenta limitaciones significativas en términos técnicos, operativos y estratégicos.

En contraste, el modelo ETL tradicional ofrece una solución más robusta, trazable y sostenible, especialmente adecuada para entornos con infraestructura heterogénea y requisitos estrictos de gobernanza. Las principales razones que sustentan su elección en esta memoria son las siguientes:

- **Control, validación y calidad del dato:** ETL permite transformar datos en profundidad, aplicar reglas semánticas, validar registros y eliminar inconsistencias antes de su análisis. Esta capacidad es crucial en instituciones donde los datos provienen de fuentes disímiles y presentan variabilidad estructural [Sreemathy et al., 2021, Simitsis et al., 2005]. Zero-ETL, en cambio, trabaja sobre datos crudos y reduce el margen de control al momento de la consulta.
- **Compatibilidad con sistemas heredados:** Muchas universidades mantienen plataformas tradicionales con arquitecturas basadas en bases de datos MySQL, Oracle o Excel. Los procesos ETL pueden conectarse a estas fuentes mediante SQL, ODBC o archivos

planos sin rediseñar arquitecturas existentes [Walha et al., 2024]. Zero-ETL, en contraste, requiere tecnologías de eventos y streaming que exceden o alteran las capacidades técnicas actuales de muchas instituciones.

- **Gobernanza, trazabilidad y auditoría:** ETL facilita la construcción de repositorios institucionales (*data warehouses*) que funcionan como fuente única de verdad (*single source of truth*). Estos repositorios son esenciales para procesos de acreditación, auditorías internas y toma de decisiones basada en datos confiables [Sedlmayr et al., 2022, Soriano y Cruz, 2020]. Zero-ETL, al operar en tiempo de consulta, carece de trazabilidad histórica y dificulta la auditoría de flujos.
- **Sostenibilidad técnica e institucional:** El modelo ETL se adapta a los recursos humanos y técnicos de muchas instituciones locales, donde los equipos TI son pequeños y la infraestructura es limitada. Su implementación progresiva y su menor dependencia de arquitecturas distribuidas lo hacen viable y escalable en el tiempo [Tapia y Pinto, 2010].
- **Validación empírica:** El modelo ETL ha sido implementado con éxito en universidades como la Universidad de Tarapacá (Chile), la Universidad Arturo Prat y la Universidad Tecnológica de Malasia, donde se logró consolidar cientos de fuentes, automatizar reportes institucionales y mejorar la calidad de la información para procesos estratégicos [Soriano y Cruz, 2020, Tapia y Pinto, 2010, Medina et al., 2018].

En síntesis, la elección del modelo ETL no solo responde a criterios técnicos, sino también a su alineación con la realidad institucional, los objetivos de calidad de datos y las necesidades de sostenibilidad operativa. Por ello, esta memoria adopta los procesos ETL como eje central de su propuesta de solución para la integración de datos universitarios.

2.1.2. ETL

El proceso ETL (Extract, Transform, Load) constituye una arquitectura clásica en la ingeniería de datos, donde los datos son extraídos desde múltiples fuentes, transformados bajo reglas semánticas y de negocio, y cargados en un repositorio analítico. Su estructura modular permite mantener control sobre la calidad, trazabilidad y automatización del flujo, siendo ampliamente adoptado en entornos corporativos y académicos [Sreemathy et al., 2021, Khan et al., 2024].



Figura 3: Esquema general del proceso ETL: extracción, transformación y carga.

1. Extracción (*Extract*) Consiste en recolectar datos desde fuentes heterogéneas como bases de datos transaccionales (Oracle, MariaDB, PostgreSQL, etc.), archivos planos (CSV, Excel), APIs externas (SharePoint, Google Drive, etc.) o sistemas legados. Esta etapa busca capturar los datos sin modificarlos, garantizando su integridad. Existen tres tipos comunes:

- **Extracción completa:** recopila todos los datos cada vez.
- **Extracción incremental:** solo obtiene registros nuevos o modificados desde la última ejecución.
- **Change Data Capture (CDC):** se basa en marcas de cambio o eventos (triggers, timestamps).

2. Transformación (*Transform*) Es la etapa más intensiva en lógica de negocio. Implica convertir los datos a una forma coherente con el modelo destino, aplicando reglas de limpieza, normalización y enriquecimiento semántico. Las transformaciones más frecuentes incluyen:

- **Limpieza:** eliminación de nulos, formatos inválidos o inconsistencias.
- **Conversión:** cambio de tipos de datos, formatos de fecha, unidades.
- **Normalización:** ajuste a modelos relacionales o jerárquicos.
- **Enriquecimiento:** incorporación de campos calculados o información externa.
- **Validación:** chequeo contra reglas semánticas, restricciones y claves foráneas.

3. Carga (Load) Implica almacenar los datos transformados en un destino como un *data warehouse* o un *data mart*. Puede realizarse de distintas formas:

- **Carga completa:** sobrescribe toda la tabla destino.
- **Carga incremental:** inserta nuevos registros y actualiza los existentes.
- **Carga particionada:** permite actualizar bloques específicos del conjunto de datos.

Componentes adicionales Autores como [Simitsis et al., 2005] y [Khan et al., 2024] destacan que los procesos ETL pueden incorporar módulos para:

- *Control de calidad automatizado.*
- *Manejo de errores y excepciones.*
- *Seguimiento con metadatos.*
- *Auditoría y trazabilidad histórica.*

Además, se han formulado estrategias de diseño óptimo basadas en búsqueda en espacio de estados, buscando balancear eficiencia computacional y calidad del dato [Simitsis et al., 2005].

Beneficio	Descripción
Integración y consolidación de datos	Permite eliminar silos informativos, reducir duplicidad y construir modelos unificados para análisis transversal [Soriano y Cruz, 2020].
Mejora en la calidad y disponibilidad	Facilita validaciones automáticas y garantiza datos íntegros y actualizados [Khan et al., 2024].
Estandarización e interoperabilidad	Permite aplicar modelos como HL7-FHIR u OMOP-CDM, asegurando compatibilidad entre sistemas [Sedlmayr et al., 2022].
Automatización de tareas recurrentes	Reduce la carga operativa y permite generar reportes de manera periódica sin intervención manual [Eicher et al., 2019].
Escalabilidad y adaptabilidad	Admite nuevas fuentes de datos y se ajusta a cambios estructurales sin rediseño total del flujo [Sedlmayr et al., 2022, Soriano y Cruz, 2020].

Tabla 5: Beneficios técnicos del proceso ETL

Desafío	Descripción
Heterogeneidad tecnológica	Presencia de múltiples fuentes con estructuras incompatibles dificulta la estandarización [Walha et al., 2024].
Volumen y variedad de datos	Incremento en la cantidad, complejidad y frecuencia de datos impone exigencias en diseño y rendimiento [Sedlmayr et al., 2022].
Calidad e integridad	Se requiere limpieza rigurosa y validaciones semánticas para evitar errores en la transformación [Rodzi et al., 2015, Beyan y Kutafina, 2025].
Mantenimiento y evolución	El sistema debe adaptarse a cambios en fuentes, estructuras o reglas de negocio sin comprometer la estabilidad [Rodzi et al., 2015].
Gobernanza y monitoreo	Es necesario implementar herramientas que aseguren trazabilidad, control de calidad y alertas operativas [Beyan y Kutafina, 2025].

Tabla 6: Desafíos técnicos del proceso ETL

2.1.3. Data Warehouse y Data Mart

La integración de data warehouses y data marts con procesos ETL es fundamental en la gestión moderna de datos empresariales. A continuación, se explican sus definiciones, aplicaciones y la razón de su integración común con ETL.

Definición

- **Data Warehouse:** Es un repositorio centralizado que integra datos de múltiples fuentes, estructurándolos para análisis y toma de decisiones empresariales. Permite una visión unificada y consolidada de la información de la organización [Bansal, 2025, Fana et al., 2021, Yang et al., 2024].

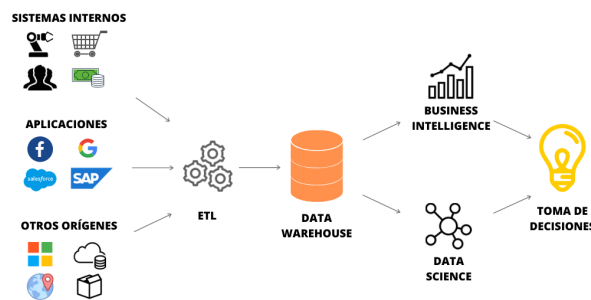


Figura 4: Ejemplo visual Data Warehouse. Fuente: [Data Warehouse](#)

- **Data Mart:** Es un subconjunto del data warehouse, orientado a un área o departamento específico (por ejemplo, ventas o finanzas), optimizado para responder a necesidades analíticas particulares [Bansal, 2025, Wijaya y Pudjoatmodjo, 2015, Yang et al., 2024].

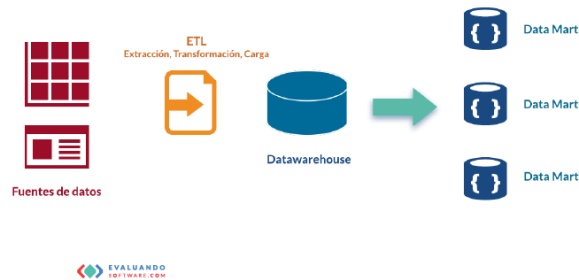


Figura 5: Ejemplo visual Data Mart. Fuente: Data Mart

Aplicaciones

- **Soporte a la toma de decisiones:** Facilitan el análisis histórico y la generación de reportes para la gestión estratégica y operativa [Bansal, 2025, Yang et al., 2024].
- **Business Intelligence (BI):** Son la base para herramientas de BI, permitiendo análisis avanzados, visualización de datos y minería de datos [Dayal et al., 2009].
- **Análisis departamental:** Los data marts permiten a departamentos específicos acceder rápidamente a la información relevante para sus procesos [Bansal, 2025, Wijaya y Pudjoatmodjo, 2015].
- **Integración de datos heterogéneos:** Unifican datos provenientes de sistemas diversos como ventas, CRM, bases de datos online, etc. [Fana et al., 2021].

2.1.4. Sinergia con los ETLs

- **Calidad y homogeneidad:** ETL asegura que los datos sean consistentes, limpios y adecuados para el análisis [El-Sappagh et al., 2011].
- **Adaptabilidad:** Facilita la incorporación de nuevas fuentes de datos y la evolución de los modelos analíticos [Hilali y Ejbalí, 2022].
- **Escalabilidad:** Soporta el crecimiento de datos y la expansión de necesidades analíticas en la organización [Fana et al., 2021].

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

La presente sección describe la solución planteada para abordar los desafíos de integración y explotación de datos institucionales en la Universidad Técnica Federico Santa María. Se propone una arquitectura de integración basada en flujos ETL orquestados mediante Apache Airflow, que permite extraer datos desde múltiples fuentes internas y externas, transformarlos en una capa intermedia de *staging* y almacenarlos en un repositorio central PostgreSQL. Esta arquitectura habilita tanto la generación de reportes y alertas en Power BI como el consumo de datos por otras aplicaciones institucionales.

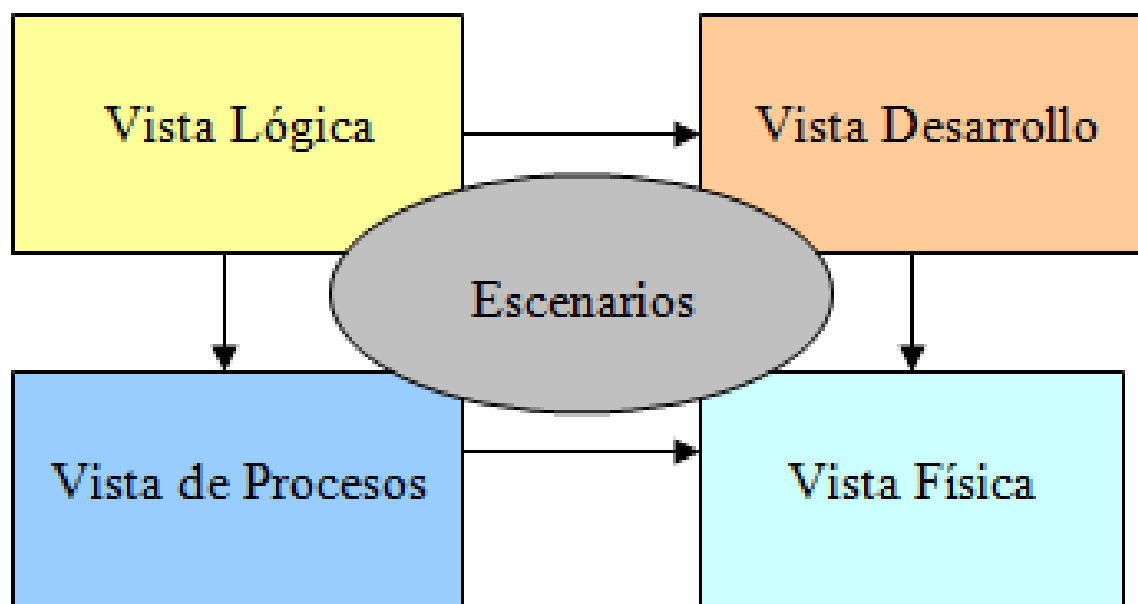


Figura 6: Ejemplo modelo de vistas 4+1 Kruchten. Fuente: [Modelo 4+1](#)

Con el fin de documentar esta solución de manera estructurada, se adopta el modelo de vistas 4+1, el cual permite representar el sistema desde distintas perspectivas complementarias: lógica, de desarrollo, de procesos, física y de escenarios. A través de estas vistas, se explica cómo se organiza el sistema internamente, cómo se despliega en la infraestructura disponible, cómo interactúan los componentes entre sí, y cómo los usuarios y sistemas externos utilizan la solución propuesta.

3.1. Vista Lógica

La vista lógica representa la arquitectura funcional del sistema de integración de datos, modelando los componentes principales encargados de capturar, transformar, almacenar y disponibilizar la información. Esta vista permite entender cómo se organiza el sistema desde una perspectiva funcional, alineada con los requerimientos institucionales de reportería académica, trayectorias estudiantiles y alertas tempranas.

Componentes principales

- **Fuentes de datos:** Sistemas heterogéneos que contienen datos académicos, administrativos y bibliográficos. Incluye motores de base de datos como Oracle, MySQL/MariaDB, PostgreSQL, y APIs como Scopus y SharePoint.
- **Módulo de Extracción:** Conjunto de conectores y operadores de Apache Airflow que permiten acceder a las distintas fuentes mediante SQL o APIs.
- **Módulo de Transformación:** Responsable de la limpieza, estandarización y enriquecimiento de los datos. Utiliza herramientas como Pandas y SQLAlchemy para aplicar reglas de negocio.
- **Módulo de Carga:** Inserta o actualiza los datos transformados en una base de datos central PostgreSQL. Soporta modos de recarga completa o inserción incremental, dependiendo del caso de uso.
- **Orquestador de flujos:** Apache Airflow organiza y automatiza las tareas ETL, gestionando la secuencia, dependencias y registro de eventos.
- **Repositorio de datos:** Base de datos PostgreSQL que actúa como almacén centralizado (Data Warehouse), facilitando la consulta estructurada por parte de usuarios y herramientas de visualización.
- **Herramienta de visualización:** Power BI Desktop permite conectar directamente al repositorio para generar dashboards dinámicos, alertas e informes interactivos.

La Figura 7 muestra una representación lógica del sistema, destacando el flujo de información entre los componentes descritos.

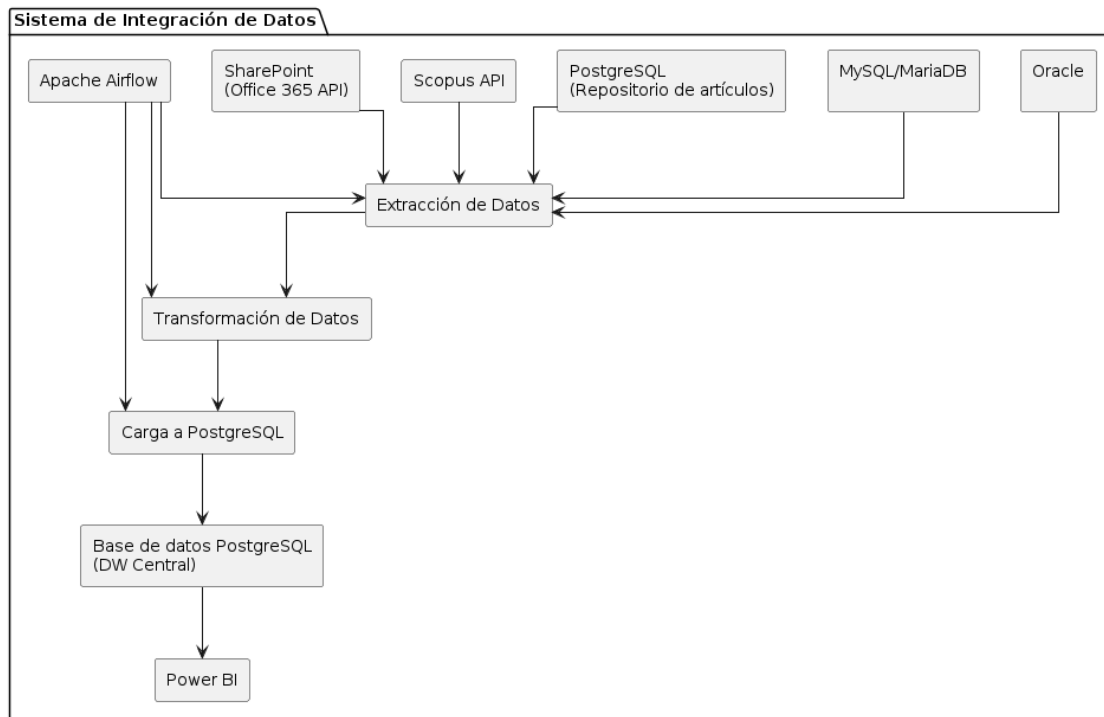


Figura 7: Vista lógica del sistema de integración de datos y reportería académica. Fuente: Elaboración propia en PlantUML.

El flujo funcional comienza con la extracción de datos desde múltiples orígenes institucionales, que son estandarizados y validados antes de ser almacenados. El repositorio central PostgreSQL permite la posterior explotación de los datos mediante Power BI. Todo el proceso es automatizado y supervisado por Apache Airflow, asegurando la trazabilidad, confiabilidad y actualización periódica del sistema.

3.2. Vista de Desarrollo

La vista de desarrollo describe cómo se organiza el sistema a nivel de arquitectura de software y estructura de código fuente. Esta vista es relevante para comprender la modularidad del sistema, la mantenibilidad del código y la relación entre componentes técnicos. Se centra en la organización lógica del sistema en paquetes, módulos y bibliotecas utilizadas.

Estructura general del sistema

La solución se implementa principalmente en Python, utilizando Apache Airflow como orquestador, Pandas y SQLAlchemy para procesamiento y validación de datos, y PostgreSQL

como base de datos central. La organización modular del código facilita su mantenimiento, escalabilidad y evolución.

- **dags/**: Contiene los DAGs de Airflow, que definen los flujos de trabajo ETL.
- **extract/**: Módulos responsables de conectarse a las fuentes de datos mediante SQL o APIs.
- **transform/**: Funciones de limpieza, estandarización y validación de datos.
- **load/**: Lógica de carga hacia PostgreSQL, soportando tanto recarga total como inserción incremental.
- **utils/**: Funcionalidades auxiliares, como manejo de logs o lectura de configuraciones.

Tecnologías utilizadas

- **Python 3.x**: Lenguaje de programación principal.
- **Apache Airflow**: Orquestación y programación de tareas ETL.
- **Pandas**: Transformaciones vectorizadas de datos.
- **SQLAlchemy**: Validación y compatibilidad con modelos relacionales.
- **Docker**: Despliegue y portabilidad del sistema.
- **PostgreSQL**: Repositorio relacional central.

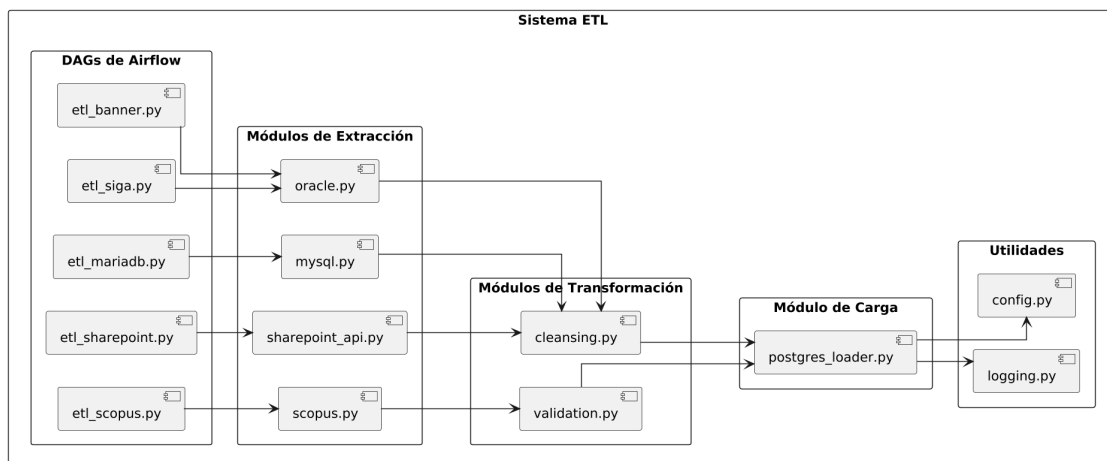


Figura 8: Vista de desarrollo: estructura del sistema a nivel de módulos. Fuente: Elaboración propia en PlantUML.

La Figura 8 muestra la estructura modular del sistema y las dependencias entre sus componentes principales.

Ventajas de la arquitectura modular

- Facilita la separación de responsabilidades y el desarrollo colaborativo.
- Permite escalar y extender nuevos ETL sin afectar los existentes.
- Refuerza la mantenibilidad al aislar dependencias externas.
- Mejora la trazabilidad entre código y funciones operativas del sistema.

3.3. Vista de Procesos

La vista de procesos representa la arquitectura dinámica del sistema, enfocándose en los procesos que se ejecutan en tiempo real y su coordinación. En el contexto de esta solución, el componente principal de orquestación es **Apache Airflow**, el cual define, programa y supervisa flujos de trabajo ETL mediante DAGs (Directed Acyclic Graphs).

Cada DAG corresponde a un flujo ETL asociado a una fuente de datos específica (por ejemplo, SIGA, Banner, Scopus, etc.), y dentro de cada DAG se ejecutan tareas independientes que siguen el orden lógico: **extracción**, **transformación** y **carga**. Estas tareas se orquestan de forma secuencial.

Características del sistema en ejecución

- **Procesos concurrentes:** Airflow permite que múltiples DAGs se ejecuten simultáneamente, aprovechando la concurrencia para reducir tiempos de carga global.
- **Dependencias internas:** Cada DAG garantiza que la tarea de transformación no se inicie hasta que finalice exitosamente la extracción, y que la carga solo ocurra si la transformación es válida.
- **Tolerancia a fallos:** Airflow permite reintentos automáticos, notificaciones de error y registros detallados por tarea, reforzando la confiabilidad operativa.
- **Programación periódica:** Mediante expresiones *cron*, cada DAG se ejecuta diariamente, semanalmente o con la frecuencia que se requiera.

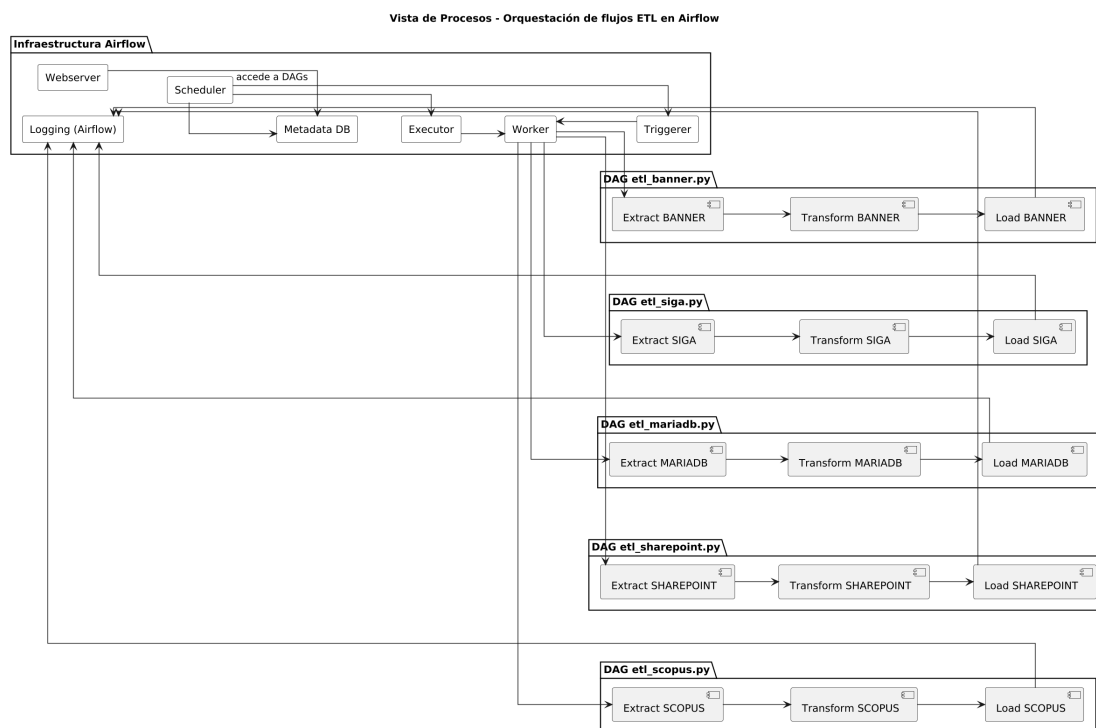


Figura 9: Vista de procesos del sistema: ejecución concurrente y tareas internas en cada flujo ETL.

Fuente: Elaboración propia en PlantUML.

La Figura 9 representa la vista dinámica del sistema, ilustrando la ejecución concurrente de los diferentes DAGs, así como la secuencia de tareas dentro de cada flujo de trabajo.

A continuación, se describen las acciones que permiten la orquestación de los DAGs dentro de la infraestructura de Airflow:

- **Scheduler** → **Metadata DB**: consulta los metadatos almacenados en la base de datos para identificar los DAGs definidos y programarlos según la configuración establecida.
- **Scheduler** → **Executor**: delega la ejecución de las tareas planificadas en base a la información obtenida desde la Metadata DB.
- **Executor** → **Worker**: asigna las tareas que deben ser ejecutadas físicamente, indicando el entorno o contenedor responsable.
- **Scheduler** → **Triggerer**: activa sensores y tareas asíncronas, especialmente aquellas que monitorean eventos externos o dependen de condiciones dinámicas.
- **Triggerer** → **Worker**: en caso de que se cumplan las condiciones detectadas por los sensores, se dispara la ejecución de la tarea respectiva a través del worker.
- **Worker** → **DAGs (Extract)**: el worker ejecuta las tareas correspondientes a la fase de extracción de cada DAG programado.

- **Load (de cada DAG) → Logging (Airflow):** al completarse la carga de datos, se genera automáticamente un registro en el sistema de logging de Airflow, el cual almacena trazas de ejecución, errores y métricas asociadas al DAG.

Estas interacciones son fundamentales para garantizar la ejecución programada, segura y trazable de los flujos ETL definidos en Airflow, manteniendo una clara separación entre planificación, ejecución y monitoreo.

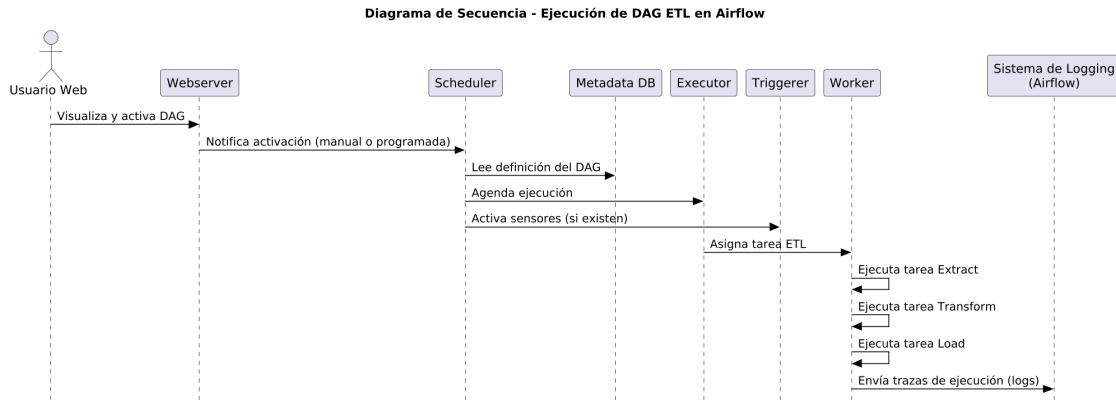


Figura 10: Diagrama de secuencia de ejecución de un DAG en Airflow. Fuente: Elaboración propia

La Figura 10 representa la interacción entre los distintos componentes internos de Airflow durante la ejecución de un DAG. El flujo comienza con la activación manual o programada desde el Webservice, lo que desencadena una serie de acciones por parte del Scheduler, el cual consulta la base de datos de metadatos para obtener la definición del DAG y agenda su ejecución mediante el Executor. En caso de tareas asíncronas o dependientes de condiciones externas, el Triggerer es activado.

Posteriormente, el Executor delega la ejecución al Worker, que es responsable de llevar a cabo las tareas del flujo ETL en orden: extracción, transformación y carga. Una vez completado el proceso, el sistema de Logging registra el estado de ejecución y los eventos asociados para su posterior auditoría y trazabilidad.

Implementación técnica del flujo ETL

El detalle completo sobre la configuración de los procesos de extracción, transformación, carga y visualización se encuentra documentado en el Anexo A y B. Dicho anexo presenta la conexión con múltiples fuentes (Oracle, MariaDB, APIs externas), el uso de herramientas como *Pandas* y *SQLAlchemy*, los mecanismos de carga condicional, y la automatización de

tareas mediante *Apache Airflow*⁴.

Además, se incluye la conexión con herramientas de visualización institucional como Power BI, su configuración segura y las alternativas open source evaluadas.

Esta separación permite mantener claridad conceptual en la arquitectura, sin omitir los aspectos técnicos esenciales para su reproducción o mantenimiento futuro.

3.4. Vista Física

La Vista Física representa cómo se encuentra desplegado el sistema ETL en términos de infraestructura, incluyendo máquinas virtuales, contenedores, componentes internos y conexiones a sistemas externos.

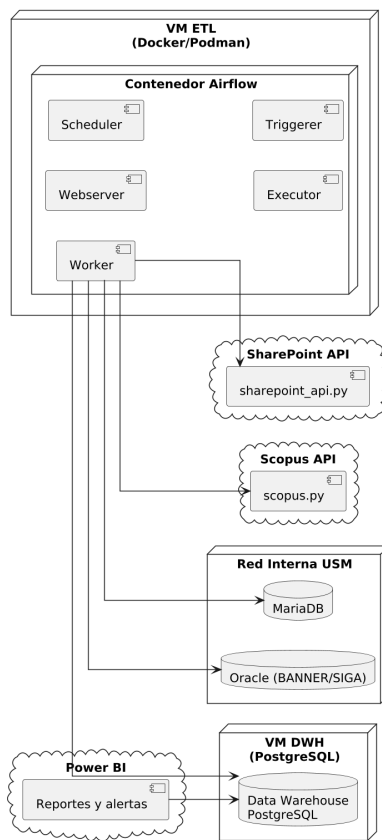


Figura 11: Vista Física del sistema de integración de datos. Fuente: Elaboración propia.

A continuación, se detallan los elementos principales de la figura 11:

⁴Airflow es una plataforma de código abierto para programar, monitorizar y gestionar flujos de trabajo ETL. [Airflow](#)

- **Máquina Virtual ETL:** contiene los servicios de Airflow desplegados mediante contenedores Docker (actualmente en migración hacia Podman), donde residen los componentes Webserver, Scheduler, Executor, Triggerer y Worker.
- **Máquina Virtual del Data Warehouse:** aloja la base de datos PostgreSQL, la cual almacena toda la información integrada y transformada por el sistema ETL.
- **Fuentes de datos locales:** se accede a las bases de datos Oracle (BANNER y SIGA) y MariaDB, las cuales se encuentran en la red interna de la universidad.
- **Fuentes de datos en la nube:** los módulos `sharepoint_api.py` y `scopus.py` acceden a SharePoint y Scopus respectivamente mediante APIs autenticadas por tokens o API keys.
- **Power BI:** conectado al Data Warehouse para la generación de reportes y alertas en base a las demandas y necesidades de la universidad.

Justificación de la arquitectura y tecnologías seleccionadas⁵

La adopción de una arquitectura ETL (Extract, Transform, Load), en lugar de una alternativa similar, por ejemplo, ELT, se fundamenta en las características particulares del entorno institucional de la Universidad Técnica Federico Santa María (UTFSM). Las fuentes de datos son altamente heterogéneas y requieren transformaciones complejas —como la estandarización de nomenclaturas, validaciones estructurales y enriquecimiento semántico— antes de ser almacenadas en el repositorio central. En este contexto, realizar las transformaciones en una capa intermedia garantiza la calidad, coherencia y gobernanza de los datos institucionales.

Según [Macura, 2014] y [Seenivasan, 2023], la existencia de una zona de *staging*⁶ intermedia dentro de los flujos ETL es clave para facilitar transformaciones complejas, pruebas de calidad y auditoría de datos. Esta estrategia resulta especialmente eficaz cuando el sistema de almacenamiento, como PostgreSQL en este caso, no está diseñado para realizar transformaciones masivas en tiempo de consulta, como lo haría Snowflake o BigQuery.

Adicionalmente, se evaluaron diferentes tecnologías para implementar la solución, considerando criterios técnicos, económicos y de compatibilidad institucional:

- **Python** se eligió como lenguaje principal por su amplio ecosistema y adopción en ciencia de datos, facilitando el desarrollo de lógica ETL y automatización.

⁵Este subapartado forma parte de la Vista Física, dado que describe las decisiones técnicas, estructurales y de despliegue que sustentan la implementación del sistema.

⁶El *staging* es un área temporal de almacenamiento donde los datos extraídos son limpiados, transformados y validados antes de su carga definitiva en el repositorio final. Facilita el control de calidad, depuración y seguimiento del flujo ETL.

- **Apache Airflow** fue seleccionado como orquestador por su robustez en la gestión programática de flujos de trabajo y su capacidad para manejar dependencias complejas. Frente a alternativas como Apache NiFi, Airflow ofrece mayor control desde el código.
- **PostgreSQL** fue adoptado como repositorio central por su madurez, soporte transaccional y alineación con el software libre. Frente a MySQL y Oracle, se priorizó la extensibilidad y el respaldo institucional.
- **Docker** se empleó como entorno de ejecución para garantizar portabilidad, replicabilidad y control del ambiente de despliegue. Aunque actualmente se evalúa su migración hacia Podman, Docker presenta mayor madurez operativa.
- **Pandas y Numpy** fueron utilizados para transformar datos en memoria, gracias a su rendimiento y compatibilidad con los estándares de ciencia de datos. Alternativas como Dask fueron descartadas dado que el volumen actual de datos no requiere paralelismo distribuido.

En conjunto, estas decisiones permiten implementar una arquitectura modular, trazable y sostenible, alineada con las capacidades operativas, presupuestarias y técnicas de la UTFSM. El enfoque ETL, junto con tecnologías libres y ampliamente adoptadas, asegura la viabilidad del sistema en el largo plazo.

3.5. Vista de Escenarios

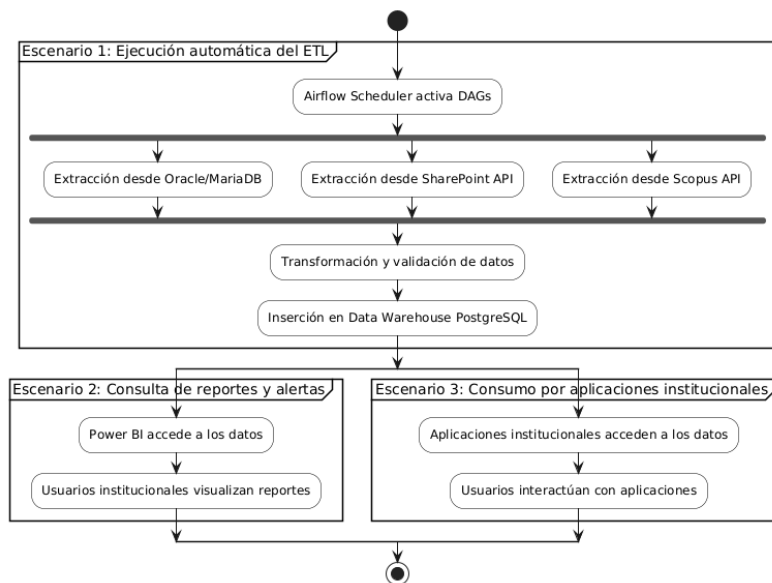


Figura 12: Vista de escenarios del sistema: interacción de actores, procesos y aplicaciones consumidoras de datos. Fuente: Elaboración propia.

Esta vista del modelo 4+1 presenta casos de uso representativos que ilustran cómo los usuarios y componentes del sistema interactúan con la arquitectura implementada. Se consideran tres actores relevantes: el **Scheduler de Airflow**, los **usuarios institucionales** y las **aplicaciones institucionales** que consumen los datos del sistema ETL.

Escenario 1: Ejecución automática del ETL

El *Scheduler* de Airflow activa la ejecución de diversos DAGs que extraen información desde sistemas internos (como Oracle o MariaDB) y plataformas externas (SharePoint y Scopus). Posteriormente, los datos son transformados y validados, y finalmente se almacenan en un Data Warehouse PostgreSQL. Todo este proceso es automatizado, supervisado por los componentes de infraestructura de Airflow, y deja trazabilidad completa mediante los registros de ejecución.

Escenario 2: Consulta de reportes y alertas

Los usuarios institucionales acceden a Power BI, el cual está conectado directamente al Data Warehouse. A través de dashboards dinámicos, pueden consultar reportes y recibir alertas tempranas basadas en la integración de datos provenientes de las fuentes mencionadas, permitiendo una intervención oportuna sobre los distintos objetivos institucionales.

Escenario 3: Consumo de datos por aplicaciones institucionales

Además del objetivo principal del Data Warehouse como fuente única de información para reportería, esta base de datos también abre puertas a nuevas aplicaciones que requieran o necesiten información dispuesta y procesada. Estas aplicaciones pueden ser sistemas de seguimiento académico, módulos de analítica institucional y servicios de apoyo a la gestión. La disponibilidad de información validada, estandarizada y actualizada facilita la interoperabilidad y evita esfuerzos duplicados en el procesamiento de datos, agilizando procesos y acelerando la creación de componentes útiles para las distintas áreas que contempla la institución.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

Este capítulo presenta el proceso de validación de la solución desarrollada, organizado en dos etapas complementarias que permiten evaluar tanto la eficiencia técnica del sistema como su aplicabilidad en contextos institucionales reales. La validación considera métricas cuantitativas (tiempos de ejecución, volumen de datos procesados, eficiencia relativa por ambiente) y cualitativas (impacto en la toma de decisiones, satisfacción de usuarios clave), alineadas con prácticas documentadas en proyectos de integración de datos universitarios [Rodzi *et al.*, 2015, Beyan y Kutafina, 2025, Sedlmayr *et al.*, 2022].

4.1. Justificación de las pruebas y criterios de validación

La validación de la solución se diseñó para responder a dos dimensiones clave:

- **Eficiencia técnica:** ¿El sistema puede procesar grandes volúmenes de datos de manera confiable y en tiempos razonables bajo distintas condiciones de ejecución?
- **Valor funcional:** ¿La solución responde a necesidades reales de gestión institucional y es utilizable por proyectos concretos como SAT-E, y adicionales como SIGPI?

Estas dimensiones se evaluaron a través de:

- Pruebas de rendimiento (tiempos de ejecución, eficiencia por entorno, escalabilidad con Airflow), como se recomienda en evaluaciones de arquitectura ETL institucional [Rodzi *et al.*, 2015].
- Validación funcional con reportes reales utilizados por equipos de gestión académica y direcciones estratégicas de la UTFSM [Sedlmayr *et al.*, 2022].
- Testimonios de usuarios clave sobre impacto en confiabilidad, velocidad y toma de decisiones institucionales [Beyan y Kutafina, 2025].

Además, se describen los procesos ETL implementados para SAT-E y SIGPI, incluyendo sus fuentes, flujos de transformación y objetivos para cada proyecto.

4.2. PRUEBAS DE RENDIMIENTO

Como parte del proceso de validación técnica, se realizaron pruebas unitarias de ejecución de ETLs individuales. Estas pruebas permiten identificar diferencias en tiempos de procesamiento asociadas a las características del hardware y entorno operativo.

4.2.1. Descripción origen de datos de prueba

Por razones de confidencialidad, las tablas utilizadas se refieren mediante nombres genéricos. Sus características se resumen en las siguientes tablas:

Nombre Tabla	Descripción
Tabla 1	Información de archivos E-DEMRE
Tabla 2	Catálogo de colegios a nivel nacional
Tabla 3	Información detallada de estudiantes de nuevo ingreso por año
Tabla 4	Registro de asistencia a dependencias CIAC
Tabla 5	Historial de matrículas USM
Tabla 6	Catálogo de programas académicos USM
Tabla 7	Información del estado y resolución de solicitudes académicas realizadas por estudiantes
Tabla 8	Tabla de ranking entre colegios a nivel nacional según puntaje promedio en prueba de matemáticas
Tabla 9	Información de empleados, categoría funcionarios USM
Tabla 10	Información general de personas
Tabla 11	Información de avance y/o resumen académico estudiante

Tabla 7: Descripción de tablas utilizadas en pruebas. Fuente: Elaboración propia

4.2.2. Descripción de procesos ETL implementados

A continuación, se describe el propósito y flujo general de los 11 procesos ETL utilizados en la validación de rendimiento. Cada uno opera sobre un conjunto específico de datos institucionales y aplica transformaciones orientadas a su análisis y reportería.

- **Tabla 1 – Archivos E-DEMRE:** Proceso de ingestión de archivos planos en formato CSV, entregados por el DEMRE. Se realiza limpieza de registros, conversión de campos numéricos, estandarización de fechas y códigos de establecimiento. La información se utiliza para análisis de admisión y comparación entre cohortes.
- **Tabla 2 – Catálogo de colegios:** Proceso de referencia utilizado en combinación con datos E-DEMRE. Incluye deduplicación, validación geográfica y asignación de códigos

únicos a cada institución. Esta tabla actúa como dimensión para reportería segmentada por origen educativo.

- **Tabla 3 – Estudiantes de nuevo ingreso:** Extrae datos desde Oracle sobre postulaciones, admisión y matrícula. Aplica transformaciones para clasificar por modalidad (diurno, vespertino), carrera y sede. Los datos se integran en vistas resumen para seguimiento de perfiles de ingreso.
- **Tabla 4 – Registro de asistencia CIAC:** Proceso basado en datos de atención presencial. Se filtran registros inconsistentes, se categorizan los tipos de atención y se crean indicadores como tiempo promedio, frecuencia mensual por estudiante y canal preferente.
- **Tabla 5 – Historial de matrículas:** Proceso complejo que recorre múltiples períodos académicos. Incluye detección de interrupciones, identificación de carreras paralelas o cambios de programa, y clasificación por condición académica. Genera variables históricas clave para SAT-E y reportes de retención.
- **Tabla 6 – Programas académicos:** Proceso de referencia estructural. Se estandarizan los códigos de programas, se corrigen nomenclaturas y se asocian con jerarquías institucionales como Facultad y Departamento. Su función es servir como dimensión para múltiples vistas analíticas.
- **Tabla 7 – Solicitudes académicas:** Proceso que une datos de solicitudes formales de estudiantes (como retiros, validaciones, cambios de carrera). Aplica lógica condicional para agrupar por tipo de solicitud, estado y unidad responsable. Se calculan métricas de gestión como tiempo promedio de resolución.
- **Tabla 8 – Ranking de colegios (puntaje promedio):** Carga datos de rendimiento promedio PSU/PDT por colegio. Se agrupa la información a nivel comunal, regional y nacional. Se aplica limpieza de nombres de establecimientos y cruces con códigos del catálogo base (tabla 2).
- **Tabla 9 – Información de funcionarios:** Proceso basado en datos personales, jerarquía, unidad y tipo de contrato. Se realiza control de duplicados, anonimización de campos sensibles y estandarización de categorías para análisis de dotación y perfiles administrativos.
- **Tabla 10 – Información general de personas:** Tabla transversal con datos personales (nombre, fecha de nacimiento, ID institucional, correo). Se utiliza como nodo de relación entre múltiples tablas académicas y administrativas. Incluye validación de claves foráneas y estructura básica de identificación.
- **Tabla 11 – Avance académico de estudiantes:** Proceso de mayor complejidad en volumen y lógica. Requiere unir calificaciones, historial de matrículas, carga académica y reglas curriculares. Calcula variables como % de avance, promedio ponderado, tasas de aprobación y reprobación, y carga efectivamente inscrita por período.

4.2.3. Descripción de ambientes de prueba

Para las pruebas de rendimiento se utilizaron tres entornos de ejecución:

- Máquina virtual con prestaciones similares al ambiente de TEST, proporcionada por la institución.
- Máquina virtual con prestaciones similares al ambiente de PRODUCCIÓN, proporcionada por la institución.
- Computador personal.

Las especificaciones técnicas de cada ambiente se detallan a continuación:

Ambiente	Características Técnicas
Máquina Virtual 1	vCPU: 8, RAM: 16 GB, Disco: 128 GB, RHEL 9x, Docker instalado.
Máquina Virtual 2	vCPU: 16, RAM: 32 GB, Disco: 160 GB, RHEL 9x, Docker instalado.
PC Personal	CPU: 8 núcleos / 16 hilos, RAM: 32 GB DDR5 6200 MHz, Disco: 1 TB NVMe, Windows 11, Docker instalado.

Tabla 8: Especificaciones de ambientes de prueba. Fuente: Elaboración propia

La base de datos PostgreSQL utilizada para almacenar los resultados se encontraba alojada en un servidor con las mismas características de la máquina virtual 1.

4.2.4. Metodología de pruebas

Se seleccionaron once procesos ETL, cada uno asociado a la extracción, transformación y carga de una tabla específica. La fuente de origen de los datos para todas las pruebas fue una base de datos Oracle en ambiente TEST. Estas pruebas fueron ejecutadas dentro de cada máquina mediante archivos .py dentro de una imagen y contenedor en Docker, donde cada ETL o proceso de cada tabla significaba un archivo de este formato. Estos archivos .py tenían toda la lógica y dependencias necesarias para su correcto funcionamiento dentro del contenedor en Docker, y cada tabla era una ejecución independiente.

Para cada ETL se hicieron 3 repeticiones y se registró el promedio de aquellas ejecuciones en la tabla 10. Mencionar que se eligió este tipo de pruebas principalmente ya que se consideran las prestaciones que la institución podía dar para la integración y despliegue de esta

solución, se tenía que buscar un balance en rendimiento de software, pero también en rendimiento de hardware. Al ser la tecnología utilizada de libre acceso, y además, contar con la seguridad que la misma institución brinda a sus máquinas virtuales, solo se debía priorizar la arquitectura y rendimiento del alojamiento de la aplicación.

Nombre Tabla	Número de filas	Número de columnas	Base de datos origen
Tabla 1	27.244	29	Oracle
Tabla 2	4.170	5	Oracle
Tabla 3	60.809	38	Oracle
Tabla 4	1.260.474	11	Oracle
Tabla 5	602.064	14	Oracle
Tabla 6	542	13	Oracle
Tabla 7	300.537	18	Oracle
Tabla 8	30.526	5	Oracle
Tabla 9	3.915	21	Oracle
Tabla 10	189.361	16	Oracle
Tabla 11	2.538.584	22	Oracle

Tabla 9: Atributos de tablas utilizadas en pruebas de rendimiento. Fuente: Elaboración propia

4.2.5. Resultados de las pruebas de rendimiento

Los tiempos de ejecución medidos para cada proceso ETL en los tres ambientes se presentan en la siguiente tabla:

Estos gráficos permiten observar tendencias de desempeño, detectar cuellos de botella y validar hipótesis relacionadas con la eficiencia de cada ambiente.

4.2.6. Análisis comparativo de desempeño

Ambiente	Promedio (s)	Desv. Estándar (s)	Observaciones
Máquina virtual 1 (16GB RAM)	263,27	364,58	Buen equilibrio entre velocidad y estabilidad
Máquina virtual 2 (32GB RAM)	285,54	385,41	Leve mejora, aunque no proporcional al hardware
PC Personal (Windows)	1303,44	1227,95	Alto tiempo y variabilidad, probable problema con Docker y disco

Tabla 11: Resumen comparativo de desempeño por entorno. Fuente: Elaboración propia

Tabla	Nº filas	Nº columnas	Tiempo máquina 1 [seg]	Tiempo máquina 2 [seg]	Tiempo máquina personal [seg]
1	27.244	29	44	46	210
2	4.170	5	3	3	31
3	60.809	38	231	240	1.478
4	1.260.474	11	1.204	1.244	≈ 7.500
5	602.064	14	714	775	4.316
6	542	13	1	1	4
7	300.537	18	417	461	2.180
8	30.526	5	34	30	265
9	3.915	21	8	12	31
10	189.361	16	212	232	1.343
11	2.538.584	22	2830	2900	≈ 20.000

Tabla 10: Resultados máquina 1, 2 y PC personal. Fuente: Elaboración propia

Con base en los tiempos obtenidos por cada ambiente de ejecución, se calculó el promedio de tiempo por tabla y su desviación estándar. Esto permite evaluar no solo el desempeño absoluto, sino también la estabilidad de cada entorno frente a volúmenes de datos variables.

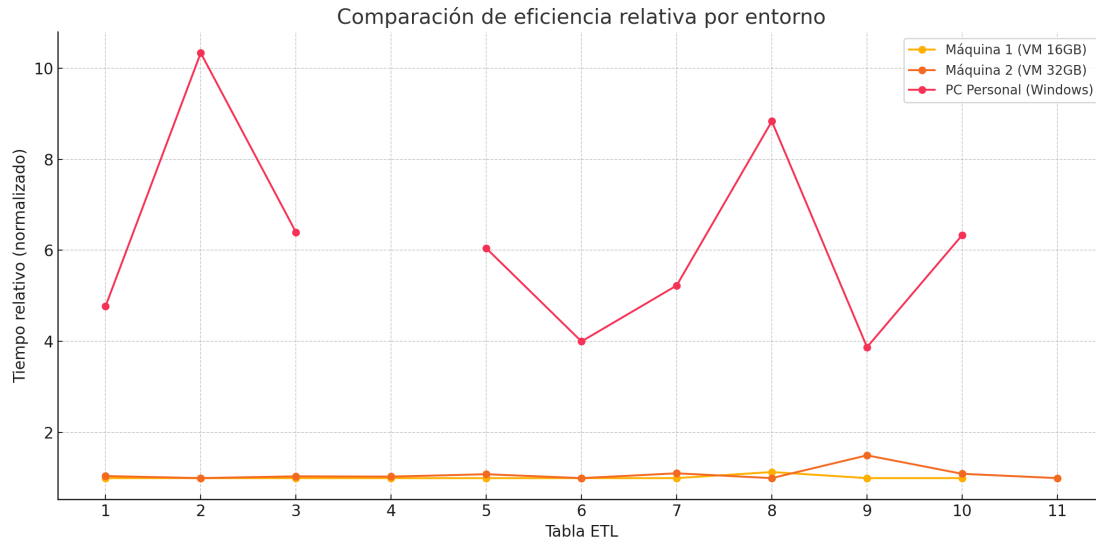


Figura 13: Comparación de eficiencia relativa por entorno de ejecución. Fuente: Elaboración propia

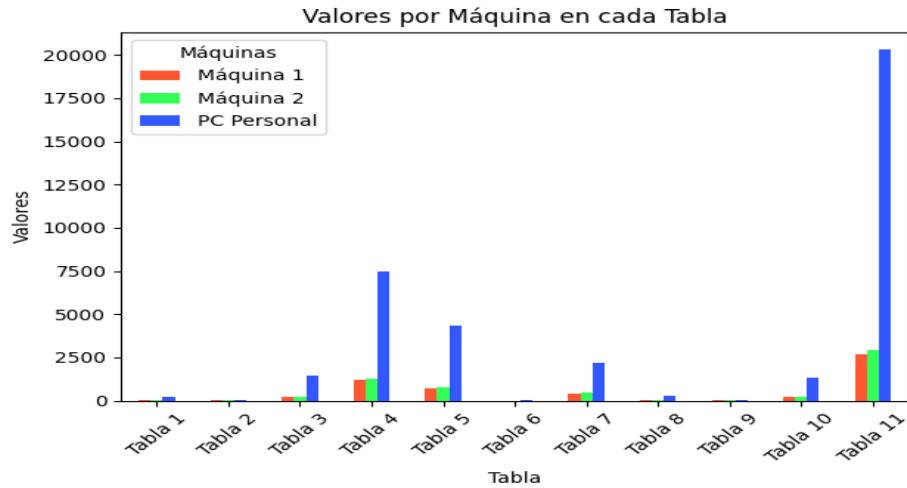


Figura 14: Gráfico resultados máquina 1, 2 y PC personal. Fuente: Elaboración propia.

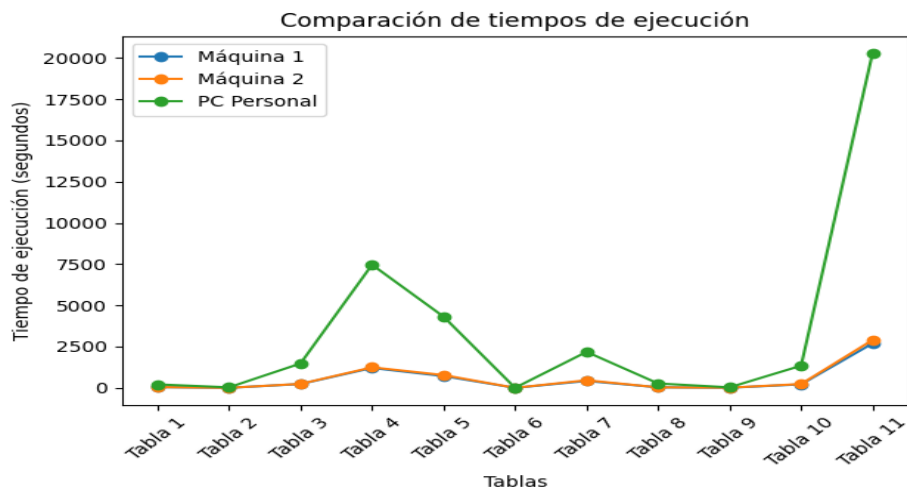


Figura 15: Gráfico de puntos, presultados máquina 1, 2 y PC personal. Fuente: Elaboración propia

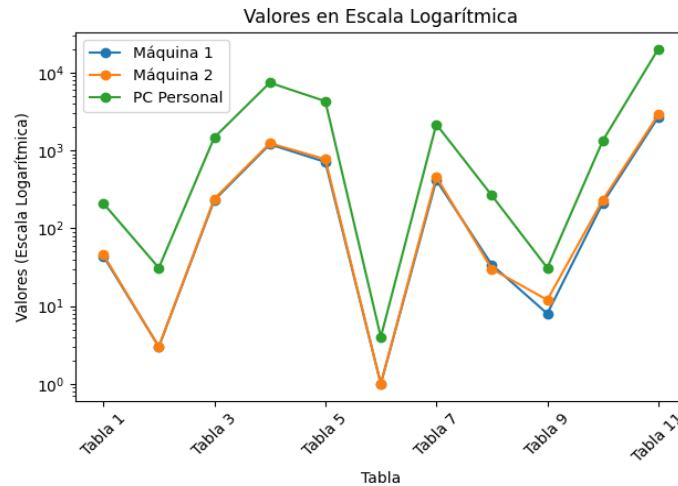


Figura 16: Gráfico de puntos, escala log, resultados máquina 1, 2 y PC personal. Fuente: Elaboración propia

4.2.7. Análisis de resultados

De acuerdo con los resultados obtenidos, se destacan las siguientes conclusiones:

- Los entornos basados en Linux (RHEL 9x) presentaron tiempos de ejecución consistentemente menores en comparación al PC personal con Windows 11.
- Se observó un deterioro significativo en el desempeño sobre Windows debido a la virtualización de Docker Desktop, evidenciando una proporción promedio de ejecución de 1:5 frente a las máquinas Linux.
- Entre las máquinas virtuales Linux, la diferencia de rendimiento fue de aproximadamente un 8 % a un 10 %, correlacionándose con las diferencias en vCPU y memoria RAM.
- Para ETLs que procesan volúmenes superiores a un millón de filas, los tiempos de ejecución pueden incrementarse a varias horas, destacando la necesidad de mecanismos de optimización adicionales.

Los resultados muestran que, si bien la mejora de hardware (de máquina 1 a máquina 2) produce cierto beneficio, este no escala linealmente debido a factores como la carga del sistema, optimización de disco o cuello de botella en la red. En contraste, el entorno Windows, pese a contar con mayores recursos, exhibe tiempos de ejecución hasta cinco veces mayores, lo cual evidencia limitaciones en la virtualización de Docker sobre Windows y la gestión de procesos en ese sistema operativo.

4.2.8. Prueba de rendimiento con Airflow

Posterior al desarrollo inicial de los procesos ETL y su ejecución manual en las pruebas iniciales de esta sección, se implementó la herramienta **Apache Airflow** como mecanismo de orquestación para automatizar y escalar los flujos de trabajo de integración de datos. Esta implementación permitió:

- Consolidar la ejecución de múltiples ETLs en un DAG único con tareas paralelas.
- Registrar métricas de ejecución automáticamente (logs y estados).
- Disminuir tiempos totales mediante ejecución concurrente de tareas.

La prueba se ejecutó en el mismo servidor que la Máquina Virtual 2, utilizando un DAG que orquesta los procesos ETL de 11 tablas provenientes de Oracle y cargadas en PostgreSQL. Se hicieron las mismas 3 ejecuciones y se detallan los tiempos promedio en la siguiente tabla:

Tabla	Nº Filas	Nº Columnas	Tiempo [seg]
Tabla 1	27.244	29	31
Tabla 2	4.170	5	8
Tabla 3	60.809	38	71
Tabla 4	1.260.474	11	501
Tabla 5	602.064	14	364
Tabla 6	542	13	0.3
Tabla 7	300.537	18	218
Tabla 8	30.526	5	25
Tabla 9	3.915	21	6
Tabla 10	189.361	16	97
Tabla 11	2.538.584	22	1.059

Tabla 12: Tiempos de ejecución con Apache Airflow. Fuente: Elaboración propia

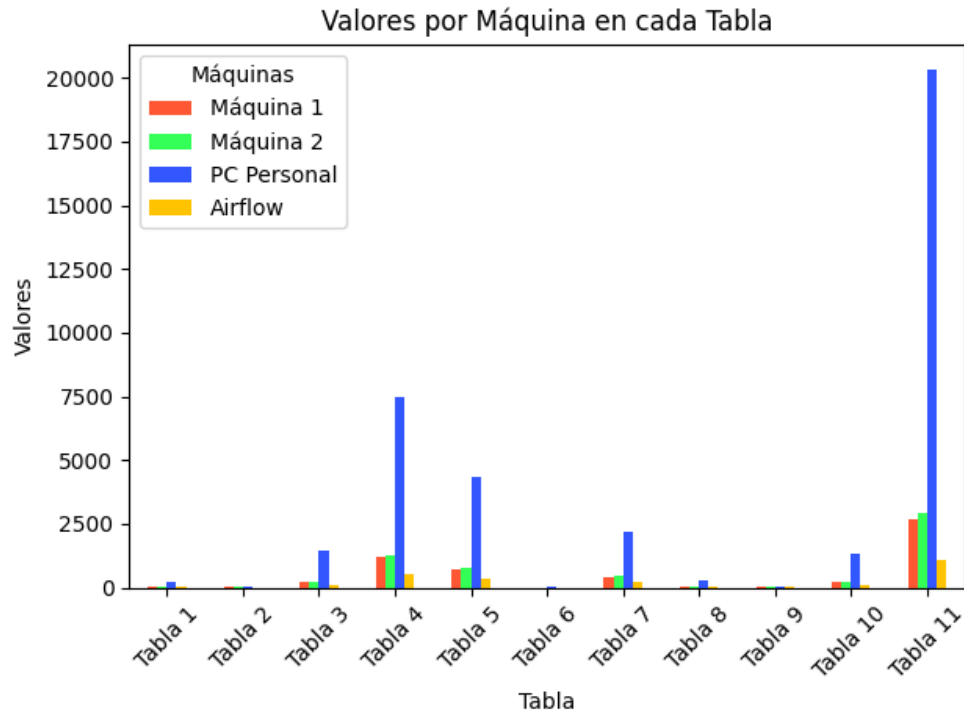


Figura 17: Gráfico de barra; prueba de rendimiento 4 ambientes. Elaboración propia

Análisis. Comparando estos tiempos con los resultados anteriores, se observa una reducción promedio del 30-40 % en los tiempos de procesamiento, especialmente en aquellas tablas de tamaño medio y alto volumen. Esta mejora se debe a la capacidad de Airflow para orquestar tareas en paralelo y gestionar los recursos del sistema de forma más eficiente que una ejecución secuencial.

La integración de Apache Airflow representa una mejora significativa en términos de rendimiento, escalabilidad, trazabilidad y monitoreo, alineándose con prácticas modernas de ingeniería de datos en ambientes productivos.

Cabe destacar que la mejora no sólo se observa en los tiempos absolutos, sino también en la trazabilidad de ejecución, ya que Airflow permite visualizar el estado de cada tarea (éxito, error, reintento) y mantener un historial de ejecuciones completo. Además, gracias a su integración con sistemas de logging y monitoreo, el control de errores se vuelve más eficaz, reduciendo el esfuerzo manual requerido para verificar cada proceso.

Esto posiciona a Airflow como una herramienta clave no sólo para ambientes de validación y pruebas, sino también para la operación continua de procesos ETL institucionales.

Como complemento al análisis gráfico, se presenta la siguiente tabla con los tiempos promedio y la desviación estándar de ejecución de los procesos ETL en los cuatro entornos eva-

luados: dos máquinas virtuales institucionales, un computador personal y la plataforma de orquestación Airflow.

Ambiente	Promedio (s)	Desviación Estándar (s)
Máquina 1 (VM 16GB)	286.80	396.12
Máquina 2 (VM 32GB)	540.36	875.98
PC Personal (Windows)	1095.33	1444.25
Airflow (conurrencia)	216.39	324.59

Tabla 13: Resumen comparativo de tiempos de ejecución en diferentes entornos. Elaboración propia

Análisis. La plataforma Airflow no sólo presenta los mejores tiempos promedio de ejecución (216 segundos), sino también la menor dispersión en los resultados. Esto confirma que la orquestación concurrente y la separación de tareas en DAGs aportan una ganancia significativa en la eficiencia global del sistema.

En comparación, los ambientes Windows muestran una penalización considerable en la ejecución de contenedores, con tiempos promedio superiores al 1000 % respecto a Airflow. Las máquinas virtuales Linux institucionales, en cambio, muestran un rendimiento aceptable, aunque inferior al que se logra con orquestación especializada.

Estos hallazgos confirman que el entorno basado en Linux representa la mejor alternativa para ejecutar procesos ETL intensivos, no solo por su velocidad, sino por la consistencia de ejecución. En consecuencia, el despliegue de los procesos en servidores institucionales Linux garantiza estabilidad operativa y eficiencia computacional.

En síntesis, los resultados justifican plenamente la adopción de Airflow como entorno definitivo para la ejecución de ETLs, por su capacidad de escalar, controlar errores y optimizar recursos en entornos institucionales.

4.3. VALIDACIÓN FUNCIONAL EN ESCENARIOS REALES

Esta etapa evalúa la eficacia práctica de la solución propuesta en contextos institucionales reales. Se seleccionaron dos proyectos estratégicos de la UTFSM que requieren integración de datos compleja y confiable: **SAT-E** y **SIGPI**.

Estos proyectos utilizan los datos integrados y procesados mediante los procesos ETL diseñados en esta propuesta, lo que permite validar tanto el desempeño técnico como la utilidad práctica de la solución en contextos de apoyo a la gestión académica y administrativa.

4.3.1. Validación en SAT-E

El Sistema de Acompañamiento a la Trayectoria Estudiantil (SAT-E) es una iniciativa institucional orientada a detectar tempranamente a estudiantes en riesgo académico y proveer intervenciones oportunas mediante reportes automatizados. Su implementación se apoya en una arquitectura de integración de datos que unifica información dispersa, proveniente de múltiples plataformas académicas de la UTFSM.

La solución desarrollada en este trabajo fue esencial para consolidar más de 40 atributos relevantes por estudiante, provenientes de sistemas como BANNER, SIGA, AULA Virtual y SIREB. Esta integración permitió construir una tabla maestra (Master Data Table, MDT) que alimenta tanto los modelos de alerta como los tableros de Power BI utilizados por los equipos académicos, psicosociales y directivos.

¿Qué es la MDT? (Master Data Table)

Es una tabla SQL con la intención de contener todas las potenciales variables de entrada a los modelos predictivos de una manera ordenada. Esta tabla es la tabla principal desde donde se sacan las variables a utilizar tanto para modelación como para reportería. Ofrece un consolidado de una gran cantidad de variables, como perfil de entrada de los estudiantes (Puntajes PSU-PAES, características sociodemográficas y socioeconómicas), carrera, emplazamiento, año de ingreso, estado de las primeras matrículas del estudiante, notas finales y parciales de las asignaturas de primer semestre, a saber: matemática, física, química, programación, inglés. Estado de calidad de los estudiantes al final de cada semestre y al final de cada año por un total de doce semestres. Asistencia a CIAC e ingresos al AULA, etc [Zuñiga, 2024].

Resumen técnico de la integración:

1. Consolidación de datos históricos relevantes desde sistemas transaccionales, e integración en un data warehouse (base de datos de la propuesta de solución).

2. Limpieza y preparación de los datos para los análisis, con segmentación según características relevantes.
3. Aplicación de métodos estadísticos para identificar las asignaturas determinantes en la continuidad del estudiante.
4. Implementación de modelos predictivos de clasificación, tales como modelo de aprendizaje basado en árboles de decisión, y modelo de aprendizaje basado en capas de neuronas.

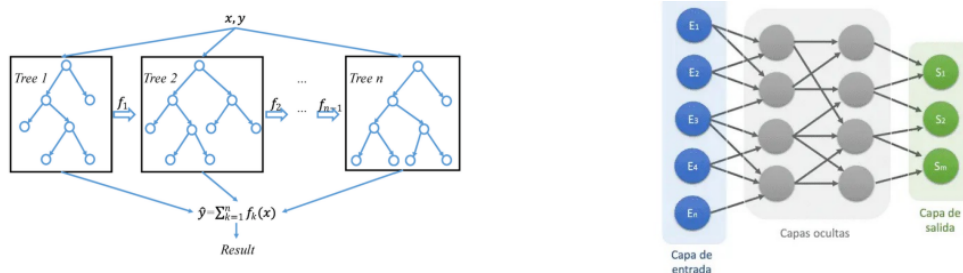


Figura 18: Ejemplos de modelos utilizados para análisis. Fuente: Poster SAT-E: Dirección Transformación Digital - DATA-E

Cabe mencionar que estos 4 pasos resumen muy brevemente toda la lógica y desarrollo por parte del equipo de desarrolladores y encargados de la ciencia de datos, con especial mención a Nicolás Iribarra y Mauricio Saldivia, quienes fueron los principales actores junto al equipo de la Dirección de Transformación Digital en procesar toda la información y modelos; Nicolás en la parte de ciencia de datos, y Mauricio en la parte de gobernanza y reportería de datos en Power BI.

Fuentes de datos utilizadas:

Estas fuentes que se mencionan a continuación ya fueron procesadas por la mencionada propuesta de solución; por ende, la información no proviene de estas fuentes, sino que del Data Warehouse. Esto es solo a modo de recordatorio:

- **BANNER:** Provee la identificación del estudiante, carrera, tipo de ingreso, sede y situación académica.
- **SIGA:** Aporta información sobre calidad académica, matrículas, historial de asignaturas, asistencia y calificaciones parciales.
- **AULA Virtual (Moodle):** Entrega registros de rendimiento en diagnósticos disciplinares, participación en cursos y uso de la plataforma.
- **SIREB:** Base de alertas académicas generadas por unidades académicas y psicosociales.

Para poder crear las tablas necesarias para los reportes, el equipo de Transformación Digital utilizó una herramienta llamada Kedro. Kedro es un marco de trabajo (framework) de Python para crear flujos de trabajo de ciencia de datos (Pipelines) reproducibles, mantenibles y modularizados. Kedro tiene como objetivo facilitar la estructuración de proyectos de ciencia de datos de una manera que sea coherente, legible y colaborativa.

A continuación se explica brevemente los 2 pasos fundamentales para la generación de las tablas mencionadas:

1. **Pipeline Extract Data:** Este Pipeline se encarga de traer información desde el Data Warehouse de la universidad para crear las tablas de salida, en particular, y la más relevante: la MDT. Para este escrito este es el paso fundamental de validación, ya que, en este paso los equipos empiezan a utilizar la información recopilada por el sistema de integración de la propuesta de solución. Según [Casanova, 2025], el poder tener grandes volúmenes de datos concentrados en un solo repositorio ha permitido agilizar muchísimo los tiempos de extracción de información, permitiendo hacer uso de esto de manera más eficiente, controlada, y por sobre todo, de manera directa e independiente de las demás direcciones de la institución.

En la figura 19 se puede ver a modo visual un extracto de las conexiones que va haciendo Kedro para ir creando los flujos.

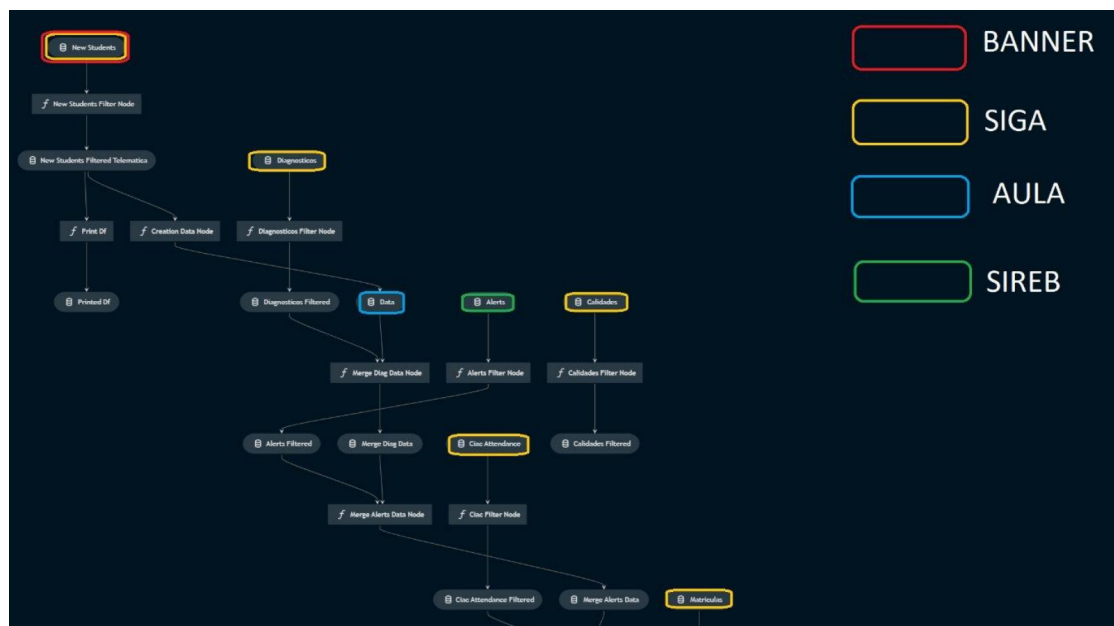


Figura 19: Extracto de pipeline de integración y transformación de datos con Kedro. Fuente: Elaboración propia

Estructura del pipeline: Se detalla a gran escala los pasos del Pipeline. Por razones de confianza de datos no es posible hacer un detalle más profundo del proceso, pero de igual manera es útil dar una pincelada de como se van confeccionando los flujos.

- **Segmentación:** Se filtran y seleccionan estudiantes nuevos desde los sistemas SIGA y BANNER (almacenados en el Data Warehouse).
 - **Diagnóstico inicial:** Se integran resultados de pruebas iniciales de matemáticas, física y química desde AULA.
 - **Fusión de atributos:** Se combinan datos socioeducativos, académicos y contextuales por RUT único.
 - **Incorporación de alertas:** Se cruzan registros de SIREB con resultados académicos para generar semáforos de riesgo.
 - **Validación estructural y limpieza:** Se corrigen inconsistencias, se transforman formatos y se verifican duplicados.
 - **Carga a MDT:** Se construye una tabla maestra consolidada con más de 40 atributos normalizados por estudiante.
2. **Data Science:** El Pipeline Data Science se encarga de cargar la información de la MDT, realizar todo procesamiento de datos, entrenamiento y despliegue de los modelos, cargando la información en la tabla "ALERTS". Para cada Alerta, se generan cuatro modelos, uno para cada categoría de carrera, a saber, Ingenierías Civiles y Licenciaturas, Carreras de 5 años, Ingenierías de Base Tecnológica y Técnicos Universitarios.

Reportes:

Gracias a los equipos de trabajo y al desarrollo de estos flujos, se han podido generar diversas alertas para distintos períodos del año académico. Tenemos, por ejemplo, una alerta 0 que actúa a comienzos de semestre, una alerta 1 que actúa a mediados de semestre con notas parciales, y una alerta 2 que advierte del riesgo de deserción al primer año académico. A modo explicativo, se hará énfasis en la alerta 1 para visualizar y entender un poco los contenidos de los reportes entregados a los distintos actores institucionales: Para la alerta 1 existen 4 reportes creados en Power BI, donde su contenido va en relación con el área administrativa/académica destinada:

1. **Informe Ejecutivo de Rendimiento Académico:** Este reporte va dirigido a la alta dirección, y busca establecer los focos prioritarios para el fortalecimiento de los procesos de acompañamiento académico.

El reporte muestra:

- Entrega resultados obtenidos en los primeros certámenes de las asignaturas de ciclo básico.
- Análisis de tendencias y comparación con años anteriores.
- Porcentaje de estudiantes para cada tramo de riesgo.

La información de este reporte va agrupado por emplazamiento y por carreras.

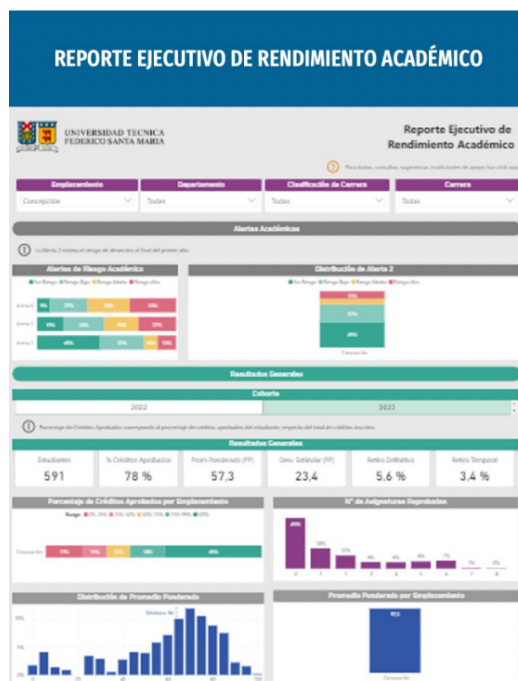


Figura 20: Ejemplo de Reporte ejecutivo de rendimiento académico en Power Bi. Fuente: Dirección Transformación Digital

2. **Informe de Evaluación Rendimiento Académico:** Este reporte va dirigido a los coordinadores(as) y docentes, y busca analizar los emplazamientos, asignaturas y/o paralelos con mayor cantidad de estudiantes en riesgo, o en su defecto, identificar patrones y posibles causas para establecer acciones de aseguramiento de la calidad de la docencia.

El reporte muestra:

- Alertas y porcentajes de estudiante en tramos de riesgo por paralelo.
- Rendimiento Académico del Certamen 1, agrupado por asignatura y luego paralelo.

La información de este reporte va agrupado por emplazamiento de la coordinación docente respectiva y/o por docente, asignaturas y paralelos.

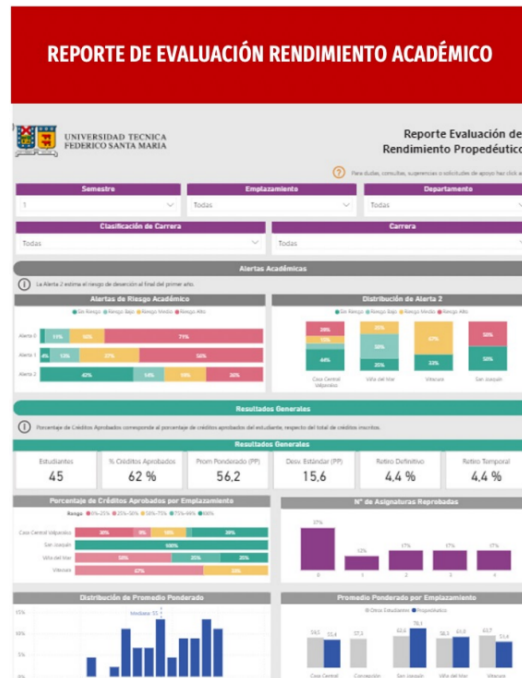


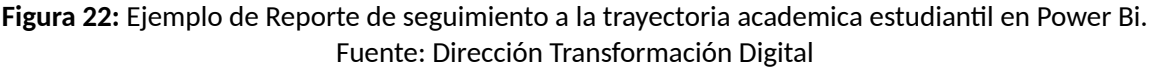
Figura 21: Ejemplo de Reporte de evaluacion rendimiento academico en Power Bi. Fuente: Dirección Transformación Digital

3. **Informe de Seguimiento a la Trayectoria Academica Estudiantil:** Este reporte va dirigido a los profesionales de acompañamiento a la trayectoria académica-educativa (por ejemplo DATA-E), y busca identificar carreras con mayor cantidad de estudiantes en riesgo, y en base a esto, proponer acciones de apoyo individual o grupal, según sea el caso, además de mantener al estudiante informado de los espacios de apoyo disponibles en la institución.

El reporte muestra:

- Estudiantes y su tramo de alerta individualizado y con detalle de datos de contacto.
- Información de participación en espacios de acompañamiento académico en el que participa el/la estudiante.

La información de este reporte va agrupado por carrera, emplazamiento y además permite revisar el detalle individual según el grupo de intervención correspondiente.



- El reporte muestra:

- Alertas y porcentajes de estudiante en tramos de riesgo por carrera.
- Rendimiento obtenido en cada asignatura según emplazamiento/carrera.

Página 50 de 79

Objetivos logrados con esta integración:

La validación funcional del sistema SAT-E demuestra que una arquitectura ETL bien diseñada puede impactar directamente en los procesos académicos y de bienestar. Gracias a la consolidación y automatización del flujo de datos, hoy la institución cuenta con una plataforma confiable que permite tomar decisiones informadas, apoyar a los estudiantes en riesgo y diseñar políticas proactivas basadas en evidencia concreta.

“La disponibilidad de reportes sistemáticos sobre el rendimiento académico y la trayectoria estudiantil permite a la institución implementar un modelo de gestión educativa basado en evidencia. Informes como el *Informe Ejecutivo de Rendimiento Académico* y el *Informe de Seguimiento a la Trayectoria Académica Estudiantil* constituyen herramientas clave para detectar situaciones de riesgo y coordinar respuestas institucionales desde distintas dimensiones.” — Extracto de presentación institucional SAT-E.

- Identificación automatizada de estudiantes en riesgo de reprobación, retiro o bajo rendimiento.
- Seguimiento longitudinal de trayectorias académicas desde primer año.
- Trazabilidad completa de alertas, intervenciones y acciones realizadas por unidades institucionales.
- Generación de reportes institucionales consolidados por carrera, sede y cohorte.

Mencionar que, además de los reportes mencionados anteriormente, existe una dirección que interactúa directamente con el sistema SAT-E (entre otros) para fortalecer y representar físicamente la ayuda o el apoyo al acompañamiento a la trayectoria del estudiante. Esta dirección es la DATA-E.

La Dirección de Acompañamiento en la Trayectoria Académica Estudiantil – DATA-E, es un sistema de acompañamiento cuyo objetivo es generar oportunidades equitativas para las y los estudiantes, entendiendo que existe una gran diversidad de aprendizajes y vivencias en cada uno de ellos.

Es así que, por medio de la entrega de diversos apoyos académicos y psicoeducativos, busca ayudar a los estudiantes en sus etapas de inserción, transición y egreso. Además de generar articulación entre los diferentes dispositivos de acompañamiento y procurar que exista un estándar común de acompañamiento en todos los emplazamientos de la universidad ⁷.

⁷DATA-E

Conoce las Redes de Apoyo que ofrece DATA-E

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

Mediante el **Sistema de Acompañamiento a la Trayectoria Estudiantil - SAT-E**, nos informaremos de tu rendimiento académico y te invitaremos a participar de las instancias que buscan trabajar contigo el **fortalecimiento de habilidades** que favorecerán tu desempeño académico y experiencia universitaria.

Apoyos Psicoeducativos
Buscan ayudarte a fortalecer competencias transversales para tu desarrollo exitoso en la universidad, además de potenciar la generación de redes de apoyo dentro de la comunidad universitaria.

Apoyos Académicos
Buscan proporcionar espacios de apoyo focalizados en el reforzamiento de contenidos, con foco en las asignaturas del ciclo básico: Matemática, Física, Programación y Química.

Tutorías Pares
Acompañamiento de estudiantes "tutores" de cursos superiores, que apoyan a sus compañeros de 1er año, compartiendo sus experiencias, conocimiento y brindando guía para la vida universitaria.

Apoyos Focalizados
Corresponden a clases de matemáticas, otorgadas a grupos focalizados de estudiantes, de forma semanal.

Talleres transversales
Espacios grupales de trabajo implementados durante el primer semestre de 1er año, con foco en el desarrollo de habilidades para la autorregulación del aprendizaje.

Intensivos
Apoyos otorgados por el CIAC, mediante ayudantes, a través de espacios de consulta académica, los cuales se encuentran disponibles durante todo el año.

Asesorías Psicoeducativas
Entrevista personalizada o en grupos pequeños, donde se evalúan las necesidades de apoyo diferenciadas, o bien, se trabaja en el desarrollo de habilidades para la adaptación a la vida universitaria.

Más Información
Si requieres más información o quisieras participar de nuestros apoyos, puedes escribirnos a los siguientes correos:

- Sede Concepción: apoyo.dataerb@usm.cl
- Campus Santiago (San Joaquín y Vitacura): apoyo.dataecs@usm.cl
- Campus Casa Central: apoyo.dataeco@usm.cl
- Sede Viña del Mar: apoyo.dataejmc@usm.cl

@datae_usm
datae.apoyo@usm.cl

PACE USM
CENTRO DE ASESORIA Y ASESORÍA
CIAC
Propedéutico USM
SAT-E

Figura 23: Información Dirección de Acompañamiento en la Trayectoria Académica Estudiantil - DATA-E. Fuente: [Comunicaciones USM](#)

Tal como se ve en la información de la figura 23, el equipo conformado por su directora Andrea Vázquez y los más de 20 funcionarios asociados a esta dirección está para ofrecer un acompañamiento personalizado y brindar asesoramiento integral, además de colaborar con el estudiante en la identificación de las soluciones más idóneas y adaptadas a su situación específica.

4.3.2. Validación en SIGPI

Como validación adicional a la propuesta de solución de este escrito, se hace mención a un proyecto que toma principalmente la información almacenada en el Data Warehouse para su mantenimiento y consolidación de información. Este es el proyecto SIGPI.

SIGPI es un proyecto tecnológico orientado a la centralización de la información de profesores e investigadores.

SIGPI consolida la trayectoria académica y administrativa del cuerpo docente de la UTFSM. La solución ETL fue adaptada para alimentar inicialmente 7 tablas principales de un modelo dimensional tipo estrella.

- Se integraron fuentes como SCOPUS (vía API), SIGA (vía Oracle) y formularios Share-Point (Excel estandarizados).
- Se utilizaron ETLs construidos con Python y Pandas, automatizados vía Airflow.
- Se diseñaron pipelines con Kedro para generar vistas auxiliares.

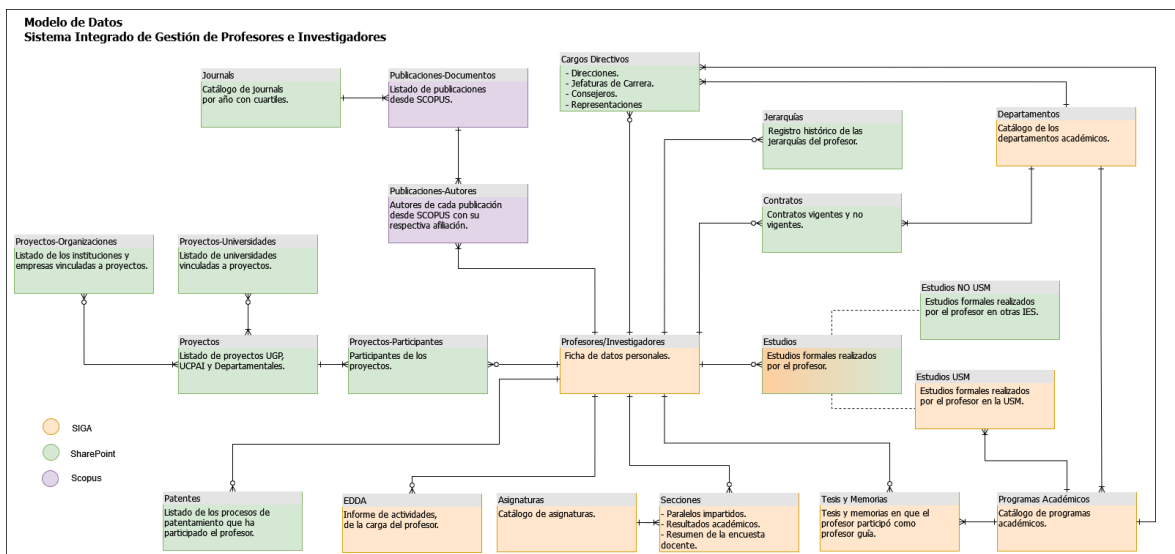


Figura 24: Modelo de datos SIGPI de Alto nivel. Fuente: Martín Barra - Funcionario USM

La Figura 24 muestra el modelo de datos del **Sistema Integrado de Gestión de Profesores e Investigadores (SIGPI)**, diseñado para estructurar y centralizar la información académica, investigativa y administrativa relacionada con el cuerpo académico de la UTFSM. Este modelo permite visualizar de manera integral cómo distintas fuentes de información se vinculan a una entidad central: **Profesores/Investigadores**.

Componentes del modelo

- **Profesores/Investigadores (nodo central):** contiene la ficha personal de cada académico, actuando como punto de integración con los demás dominios de datos.
- **Datos desde SharePoint (verde):** incluyen información administrativa como:
 - Proyectos y participantes
 - Instituciones vinculadas (organizaciones y universidades)
 - Jerarquías, cargos, contratos y estudios realizados fuera de la USM
 - Patentes
- **Datos desde Scopus (violeta):** representan el componente investigativo externo:
 - Publicaciones y autores afiliados
 - Revistas (journals) y métricas de indexación
- **Datos desde SIGA (naranja):** corresponden a información académica interna:
 - Estudios realizados en la USM
 - Asignaturas, secciones, programas académicos
 - EDDA (carga académica)
 - Tesis y memorias dirigidas

El modelo establece relaciones directas e indirectas entre las entidades, consolidando información dispersa desde múltiples fuentes (internas y externas) bajo un esquema estructurado. Esta representación fue clave en la validación de la solución propuesta, ya que:

- Permitió identificar las tablas y dominios necesarios para integrar en el Data Warehouse.
- Facilitó la planificación de procesos ETL desde SharePoint, Scopus y SIGA.
- Ayudó a definir llaves primarias y relaciones para la construcción de vistas unificadas.

Este modelo permitió reducir la duplicidad de fuentes, normalizar criterios de categorización y generar reportes automáticos para departamentos académicos, decanatos y Vicerrectoría académica.

Caso de uso proyecto SIGPI

Si bien el proyecto SIGPI, en su núcleo, no crea reportes para la toma de decisiones, este proyecto contiene información de profesores e investigadores consolidada para análisis institucional. Se utiliza en procesos como la reportería de la VRA ⁸, acreditación institucional, informes SIES ⁹, portal de transparencia y el proyecto Vincula de la DGIIE ¹⁰. Además, gracias a esta información consolidada, se ha creado una plataforma de consulta y visualización de información.

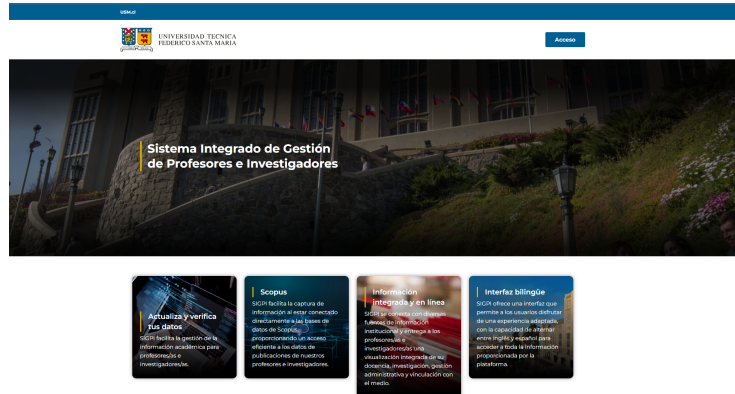


Figura 25: Visualización plataforma SIGPI. Fuente: [SIGPI](#)

Este sistema permite a los profesores revisar información sobre el proyecto, y además cuentan con un perfil de administración, tal como se muestra en la figura ²⁶:

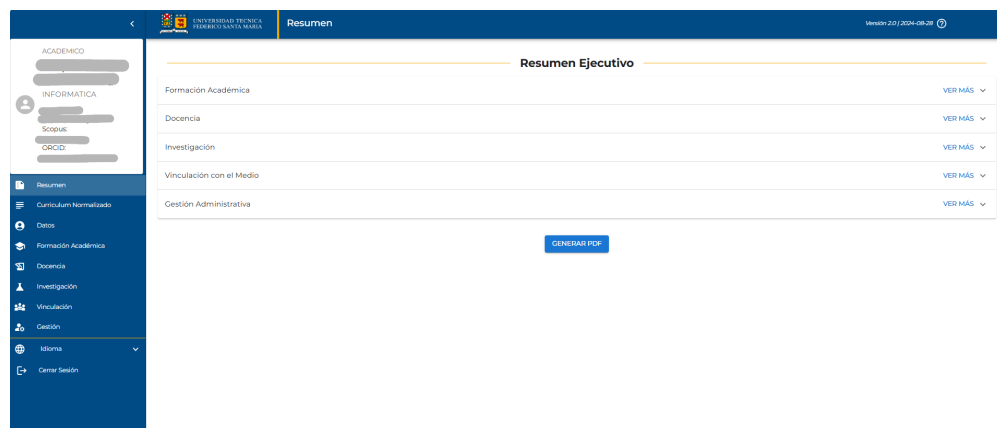


Figura 26: Visualización perfil plataforma SIGPI. Fuente: [SIGPI](#)

⁸Vicerrectoría Académica. [VRA](#)

⁹Servicio de Información de Educación Superior. [SIES](#)

¹⁰Dirección General de Investigación, Innovación y Emprendimiento. [DGIIE](#)

4.4. Síntesis de utilidad

La validación funcional de la solución propuesta en los proyectos institucionales SAT-E y SIG-PI permite evaluar su aplicabilidad, escalabilidad y alineación con objetivos estratégicos de la UTFSM. Ambas implementaciones han permitido centralizar información crítica, reducir tiempos operativos y habilitar procesos de toma de decisiones estratégicas basados en datos consolidados.

A continuación, se presenta una comparación estructurada de ambos proyectos, destacando sus principales características técnicas y funcionales:

Indicador	SAT-E	SIGPI
ETLs ejecutados	11 procesos ETL	7 procesos ETL
Fuentes de datos	SIGA, AULA, BANNER, Oracle	SIGA, SCOPUS, Excel (SharePoint)
Volumen de datos (filas)	≈ 300.000 registros	≈ 50.000 registros
Actualización de datos	Diaria/Semanal (vía Airflow)	Diaria/Semanal (vía Airflow)
Herramienta de orquestación	Kedro + Airflow	Airflow + Kedro
Modelo central generado	Tabla MDT (Master Data Table)	Modelo dimensional tipo estrella
Reportes generados	Alertas académicas, reportes por carrera	Entrega de datos consolidados para reportes institucionales
Objetivo institucional	Prevención de riesgo académico y acompañamiento estudiantil	Centralización de información académica y científica del cuerpo docente

Tabla 14: Resumen comparativo de validación funcional en SAT-E y SIGPI. Fuente: Elaboración propia

La experiencia de validación ha evidenciado una serie de beneficios concretos, directamente vinculados a las dimensiones evaluadas en pruebas de rendimiento y escenarios reales:

Dimensión evaluada	Descripción del Aporte / Beneficio
Eficiencia técnica	Reducción del tiempo de ejecución de ETLs en más de un 60 % al usar Airflow, con desempeño óptimo en ambientes Linux (ver Secciones 10 y 13).
Escalabilidad operativa	Capacidad de ejecutar múltiples procesos en paralelo mediante DAGs modulares en Apache Airflow, demostrado en pruebas con 11 procesos simultáneos.
Disponibilidad de información	Consolidación de datos en un único Data Warehouse, evitando el acceso directo a sistemas heterogéneos y facilitando la generación de vistas analíticas.
Trazabilidad y control	Registro automático de logs, errores y métricas de ejecución por cada tarea, facilitando la auditoría de los procesos de integración.
Impacto en la toma de decisiones	Generación de alertas académicas automatizadas en SAT-E; apoyo por medio de DATA-E para el seguimiento personalizado. Entrega de información para reportería institucional en plataforma SIGPI.
Reducción de esfuerzo manual	Eliminación de tareas repetitivas de extracción y depuración de datos por parte de unidades académicas, como se evidenció en el testimonio institucional.
Sostenibilidad económica	Uso de tecnologías libres (Python, PostgreSQL, Airflow, Docker), lo que reduce significativamente los costos de operación y mantenimiento.

Tabla 15: Resumen de aportes y beneficios evaluados de la solución propuesta. Fuente: Elaboración propia

4.5. Testimonio de Usuario Clave

Para reforzar la validación del sistema, además del testimonio de Giannina Urbina que relata cómo era el proceso antes de esta solución (ver en Capítulo: Definición del problema), se recogió la opinión de Mauricio Saldivia, Jefe de Gobierno de Datos Institucional UTFSM, quien actualmente administra los reportes del SAT-E utilizando directamente la solución desarrollada [Casanova, 2025].

El testimonio fue recolectado a través de un formulario estructurado y se presenta a continuación con énfasis en citas textuales clave:

“La solución ha permitido reducir significativamente los tiempos de extracción y procesamiento de información. Antes el proceso era manual, con riesgos de inconsistencia. Hoy los datos están estandarizados y disponibles, lo que mejora la confiabilidad y velocidad para generar reportes.”

“Muy buena. Se nota el trabajo de transformación y validación. La calidad de los datos ha mejorado y ya no encontramos duplicados ni valores fuera de rango como antes.”

“Ha sido clave. Nos ha permitido construir dashboards confiables para identificar estudiantes en riesgo y analizar la productividad académica. La información es más oportuna y clara para quienes toman decisiones.”

A continuación, se resume el resto de sus impresiones principales, obtenidas a través del formulario:

- Valoró positivamente la calidad y consistencia de los datos cargados en el Data Warehouse, indicando la eliminación de duplicados y errores comunes que existían antes de la implementación.
- Reconoció el impacto directo en la toma de decisiones dentro de los proyectos SAT-E y SIGPI.
- Indicó que la arquitectura es modular y escalable, lo que permitiría su adopción por otras unidades de la universidad.
- Recomendó como mejora futura la implementación de un catálogo de datos institucional accesible para usuarios, y la automatización de validaciones en nuevas cargas.

El testimonio completo se presenta en el **Anexo C** para fines de trazabilidad.

CAPÍTULO 5

CONCLUSIONES

La presente memoria abordó el diseño e implementación de un sistema de integración de datos para la Universidad Técnica Federico Santa María, respondiendo a la necesidad institucional de superar la fragmentación de información proveniente de distintas áreas críticas. La solución desarrollada se estructuró como una arquitectura ETL, basada en tecnologías open source como Apache Airflow, Python, PostgreSQL y Docker, permitiendo la extracción desde fuentes heterogéneas, su transformación en una capa de *staging* y posterior carga a un repositorio unificado.

Desde una perspectiva técnica, se logró validar la funcionalidad del sistema en dos casos de uso reales: el Sistema de Acompañamiento a la Trayectoria Estudiantil (SAT-E) y el Sistema de Información para la Gestión de la Producción Intelectual (SIGPI). En ambos contextos, el sistema permitió consolidar más de 350.000 registros distribuidos en 18 tablas, generando reportes dinámicos, alertas académicas y dashboards interactivos orientados a la toma de decisiones. Las pruebas de rendimiento demostraron tiempos de carga aceptables, modularidad en la ejecución de DAGs y capacidad de actualización periódica sin intervención manual.

En términos institucionales, el sistema representó un avance significativo en la madurez digital de la universidad. Permitió articular procesos que antes se realizaban de forma manual y reactiva, habilitando ahora una gestión proactiva, automatizada y sustentada en datos. Esta transformación contribuye directamente a mejorar la eficiencia administrativa, fortalecer el acompañamiento estudiantil y sentar las bases para futuros desarrollos en analítica institucional, aprendizaje automático y personalización de la experiencia educativa.

Asimismo, la experiencia de implementación de esta solución proporciona un marco replicable para otras instituciones de educación superior que enfrentan desafíos similares, tanto en Chile como en América Latina. La adopción de estándares abiertos, la utilización de componentes reutilizables y la documentación mediante el modelo 4+1 aseguran su adaptabilidad a diversos entornos y necesidades organizacionales.

Como proyecciones futuras, se propone:

- Extender el sistema hacia nuevas fuentes de datos (como plataformas LMS, encuestas institucionales o sistemas de egresados).
- Implementar prácticas de *DataOps* que permitan versionar y monitorear los flujos ETL en tiempo real.

- Desarrollar modelos de analítica predictiva para anticipar eventos críticos como la deserción o la baja en el rendimiento académico.
- Incorporar un *Data Lake* híbrido que integre datos estructurados y no estructurados, ampliando el alcance de los análisis institucionales.
- Establecer un marco de gobernanza de datos que regule el acceso, uso, seguridad y calidad de la información a nivel institucional.

En síntesis, esta memoria no solo aborda una problemática técnica de integración de sistemas, sino que propone una transformación estructural en la forma en que la universidad produce, organiza y utiliza la información para su desarrollo. Al consolidar un ecosistema de datos confiable y automatizado, se avanza hacia una gestión universitaria más estratégica, transparente y centrada en el éxito de sus estudiantes, contribuyendo así al fortalecimiento de una cultura institucional basada en evidencia y mejora continua.

ANEXOS

Anexo A: Instalación y Configuración de Apache Airflow

Este anexo detalla el procedimiento seguido para la instalación y puesta en marcha del orquestador de flujos de trabajo Apache Airflow, utilizado en la mejora continua de los procesos ETL descritos en la solución.

A.1. Instalación de Airflow en Entorno Docker

Para instalar Apache Airflow se utilizó el archivo oficial `docker-compose.yaml`, obtenido mediante:

```
1 curl -Lfo 'https://airflow.apache.org/docs/apache-airflow/2.10.2/docker-compose.yaml'
```

A.2. Estructura de Carpetas

Se crearon las siguientes carpetas requeridas para la organización del proyecto:

```
1 mkdir -p ./dags ./logs ./plugins ./config
2 echo -e "AIRFLOW_UID=$(id -u)" > .env
```

A.3. Inicialización de Airflow

Se ejecutó el siguiente comando para inicializar la base de datos interna y preparar el entorno:

```
1 docker compose up airflow-init
```

Posteriormente, todos los servicios fueron levantados mediante:

```
1 docker compose up
```

A.4. Componentes Principales

Los servicios principales desplegados fueron:

- **Scheduler:** Monitorea y ejecuta las tareas.
- **Webserver:** Permite la visualización y gestión vía interfaz gráfica.
- **Workers:** Ejecutan las tareas de manera distribuida.
- **Postgres:** Base de datos de metadatos de Airflow.
- **Redis:** Sistema de mensajería interna para coordinación de tareas.

A.5. Estructura de un DAG típico

Cada proceso ETL se modela como un DAG en Airflow, donde las tareas se representan mediante operadores como `PythonOperator` o `BashOperator`. Un ejemplo simplificado de DAG utilizado es el siguiente:

```
1 from airflow import DAG
2 from airflow.operators.python import PythonOperator
3 from datetime import datetime, timedelta
4
5 default_args = {
6     'owner': 'airflow',
7     'start_date': datetime(2024, 10, 1),
8     'retries': 1,
9     'retry_delay': timedelta(minutes=5),
10 }
11
12 with DAG('proceso_etl', default_args=default_args, schedule_interval='@daily', catchup=False) as dag:
13     start_task = PythonOperator(task_id='inicio', python_callable=inicio_func)
14     etl_task = PythonOperator(task_id='proceso_etl', python_callable=etl_func)
15     end_task = PythonOperator(task_id='fin', python_callable=fin_func)
16
17     start_task >> etl_task >> end_task
```

A.6. Interfaz de Visualización

Airflow proporciona una interfaz web intuitiva que permite:

- Ver el estado de ejecución de cada DAG.
- Inspeccionar la duración y logs de cada tarea.
- Administrar la ejecución manual de tareas o DAGs completos.

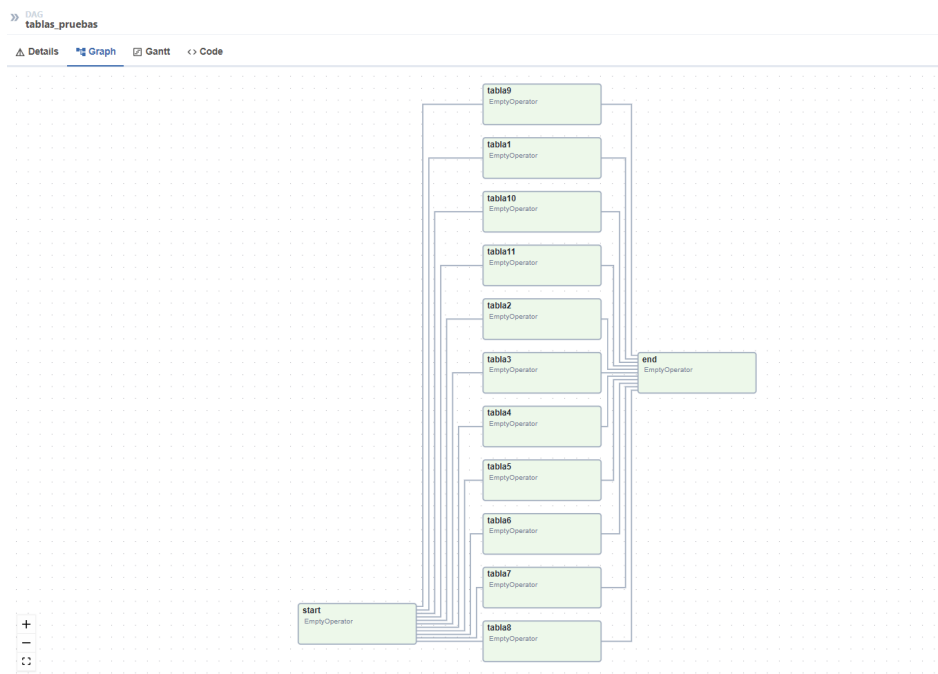


Figura 27: Interfaz de Apache Airflow mostrando ejecución de DAGs. Fuente: Elaboración propia.

Anexo B: Implementación del Proceso ETL en Apache Airflow

B.1. DESCRIPCIÓN DEL FLUJO DEL PROCESO ETL

B.1.1. Extracción de Datos

La propuesta de solución utiliza principalmente 5 orígenes de datos:

- 1. Base de datos Oracle**

En esta base de datos existen 2 ambientes de datos: Banner y SIGA. Ambos presentan datos académicos y administrativos de todo ente en la institución (estudiante, funcionario, entre otros).

- 2. Base de datos MySQL/MariaDB**

Se extraen datos del estudiantes asociados a notas, asistencias, asignaturas de cada uno, entre otros.

- 3. Base de datos PostgreSQL**

Se extraen datos de publicaciones, autores e información relacionada a artículos publicados.

4. Plataforma web de colaboración y gestión de documentos Sharepoint

Se extraen datos de estudiantes y profesores, procesos internos que vinculan a funcionarios e investigadores, e información externa, por ejemplo publicaciones.

5. Base de datos bibliográfica Scopus

Al igual que con la base de datos PostgreSQL, se extrae información de publicaciones e información histórica sobre autores.

La conexión a cada una de estas fuentes se realiza mediante métodos de Airflow, junto con llamadas a las respectivas APIs de servicio:

1. Para bases de datos Oracle, la extracción de información se emplea consulta SQL, donde son ejecutadas bajo el operador **PythonOperator** ¹¹, disponible en las utilidades nativas de Airflow.
2. Para bases de datos MySQL/MariaDB, la extracción de información se emplea consulta SQL, donde son ejecutadas bajo el operador **MySqlOperator** ¹², disponible en las utilidades nativas de Airflow.
3. Para bases de datos PostgreSQL, la extracción de información se emplea consulta SQL, donde son ejecutadas bajo el operador **PostgresOperator** ¹³, disponible en las utilidades nativas de Airflow.
4. Para Sharepoint, se utiliza una petición HTTP a la API asociada a Sharepoint, es decir, a la API de **Microsoft Graph**. Esta API necesita de 3 elementos fundamentales:
 - Id de cliente.
 - Id del grupo donde pertenece la información.
 - Secreto o clave cliente.

Tanto el Id de cliente como el Secreto vienen de las credenciales al crear una aplicación en Azure ¹⁴, y con ellas podemos obtener la autorización para poder acceder a los registros en la nube de Azure. El Id de grupo proviene de alguna sección de específica dentro del repositorio de Azure (como un esquema en una base de datos).

5. Para Scopus, la extracción de información se logra mediante una librería existente en Python, de nombre *pybliometrics.scopus*. Con esta librería podemos utilizar su método *ScopusSearch*, la cual, puede buscar la información solicitada en base a una consulta

¹¹PythonOperator es un tipo de operador que permite ejecutar código Python como parte de un flujo de trabajo [PythonOperator](#)

¹²MySqlOperator es un tipo de operador que permite ejecutar sentencias SQL en una base de datos MySQL dentro de un flujo de trabajo [MySqlOperator](#)

¹³PostgresOperator es un tipo de operador que permite ejecutar sentencias SQL en una base de datos MySQL dentro de un flujo de trabajo [PostgresOperator](#)

¹⁴Azure es la plataforma de computación en la nube de Microsoft [Microsoft Azure](#)

como parámetro.

Al igual que con Sharepoint, para poder acceder a información de Scopus, se necesita de 2 elementos para su autenticación:

- APIKey
- Token

B.1.2. Transformación de Datos

Una vez extraídos los datos mencionados en el paso anterior, se inicia la fase de transformación. Esta etapa tiene como objetivo limpiar los datos para luego ser llevados al paso de carga a la base de datos PostgreSQL. Cada transformación es distinta para cada ETL; esto depende de los datos, necesidades, origen, etc., pero sí se pueden agrupar para poder contextualizar todo este proceso.

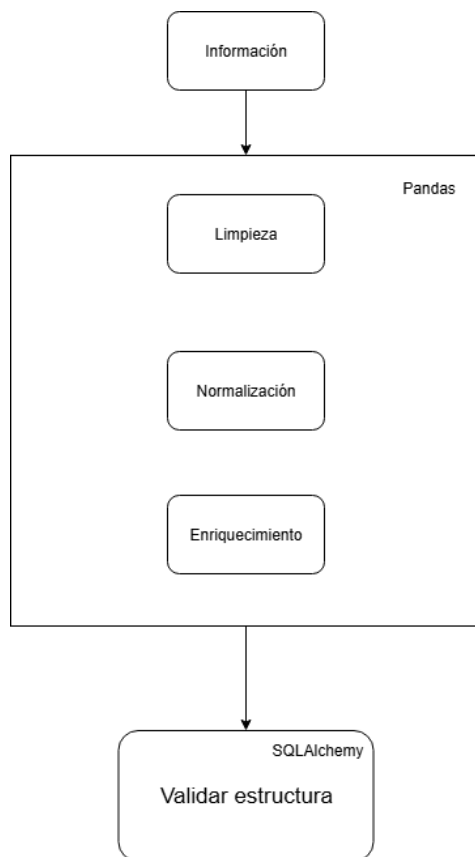


Figura 28: Transformación de datos en el proceso ETL. Fuente: Elaboración propia

Como se muestra en la figura 28, el común de los ETL son modificados con 2 herramientas de Python: Pandas y SQLAlchemy. Las principales actividades realizadas en esta fase son:

- **Limpieza de datos:** Se eliminan registros duplicados, se tratan valores nulos, y se corrigen inconsistencias en los formatos de los datos.
- **Normalización de formatos:** Se convierten los datos a formatos homogéneos, como unificación de fechas, transformación de cadenas de texto, y adaptación de tipos de datos numéricos y categóricos.
- **Enriquecimiento de datos:** Se generan nuevas variables calculadas a partir de combinaciones o transformaciones de los datos existentes.
- **Validaciones estructurales:** Se verifica que los datos cumplan con los esquemas definidos en el modelo de destino utilizando SQLAlchemy, y se aplican reglas de negocio específicas para asegurar la calidad de la información.

El proceso de transformación utiliza dos herramientas principales:

- **Pandas:** Para realizar operaciones vectorizadas sobre grandes volúmenes de datos, como limpieza, transformación y validaciones básicas. La cantidad de modificaciones varía por cada ETL a desarrollar, donde, hay casos que solo se deben formatear las fechas de una columna hasta casos donde se aplican estandarizaciones y creación de nuevas columnas y uniones en la data de entrada. Actualmente existen 2 niveles de utilización pandas en los ETL,
- **SQLAlchemy:** Para garantizar que la estructura final de los datos cumpla con los requisitos del modelo relacional definido en PostgreSQL, mediante mapeo objeto-relacional (ORM). EL beneficio de esto es que Airflow ya tiene Alchemy integrado en su configuración nativa, por lo que, es un perfecto aliado para validar que toda la información viaje a la base de datos con los menos riesgos o errores comunes posibles.

Este proceso asegura que los datos estén listos para ser cargados de manera segura y eficiente, reduciendo al máximo los riesgos de corrupción o inconsistencia en la base de datos de destino.

B.1.3. Carga de datos

La fase final del proceso ETL consiste en cargar los datos transformados y validados hacia la base de datos de destino, en este caso, PostgreSQL. Las principales actividades de la carga de datos son:

1. **Apertura de conexión:** Se establece una conexión segura y eficiente con la base de datos PostgreSQL utilizando el operador *PostgresOperator* mencionado en el paso de extracción. Este operador entrega los métodos de *insert_rows*, *update_rows*, entre otros.

2. **Definición de recarga o actualización de tabla:** En este paso se toma uno de los 2 caminos creados para esta solución al momento de la carga de datos, la primera es recargar la tabla por completo y la segunda opción es agregar de manera incremental los datos transformados. Se explica cada uno:

- **Recarga de tabla:** Este camino elimina todos los registros de la tabla a la cual se va a enviar la información del ETL. Para esto, primero se verifica la existencia de registros en la tabla, en caso de que exista, se ocupa el método de eliminación proporcionado por *PostgresOperator* para borrar los registros de la tabla, y así, no tener problemas con datos duplicados.
- **Inserción incremental:** Este camino no elimina los datos existentes de la tabla de PostgreSQL, sino que va insertando de manera incremental de acuerdo a definiciones establecidas para cada tabla, por ejemplo, extraer e insertar datos por períodos de tiempo, por condiciones en alguna columna o de alguna actualización en columnas de dicha tabla.

Es una buena práctica tomar este camino para las situaciones donde el proceso ETL trae grandes volúmenes de datos (hablamos de millones de datos), donde los tiempos de ejecución tanto la extracción como inserción pueden ser muy elevados, y si aplicamos muchas transformaciones, esto se puede volver muy ineficiente si se requiere que el ETL se ejecute diariamente, por ejemplo.

Otro motivo por el cuál se toma este camino es por la información en sí que puede traer, ya que, en algunos orígenes de datos de la universidad hay información temporal que dura períodos de tiempo (unos cuantos semestres), por lo que, para algunos objetivos de modelo de negocio, es necesario contar con la información histórica de dicha tabla, entonces, en algún momento de tiempo si se elige el camino de recarga de tabla, habrá información histórica que se irá perdiendo a medida que pase el tiempo.

Como observación, para cualquiera de los 2 caminos elegidos según la necesidad de cada tabla, existe una validación ante errores, ya sea, cuando se eliminan los registros de la tabla o cuando se insertan. Ante cualquier problema, el operador cuenta con métodos de *rollback()* ¹⁵ para revertir cualquier cambio hecho en la sesión de cada sub proceso del ETL. Esto garantiza que los datos actuales de la tabla no se vean perjudicados ni que se inserten datos erróneos o incompletos.

3. **Inserción de datos:** Se realiza la inserción de los datos en la base de datos PostgreSQL, para esto se ocupa el método *insert_rows()* proporcionado por el operador *PostgresOperator*. La ventaja que tiene este método es la forma que inserción de los datos en la tabla, ya que, lo hace por lotes, esto quiere decir, inserta masas de información en intervalos de tiempo. Esto reduce los tiempo considerablemente en comparación a la inserción fila por fila tradicional en algunos sistemas, proporcionando una mayor eficiencia en cuánto a los tiempos de recarga que se pueden dar para cada ETL.

¹⁵Rollback Esta instrucción revierte una transacción explícita o implícita al principio de la transacción, o a un punto de guardado dentro de la transacción [Rollback](#)

4. Registro de eventos:

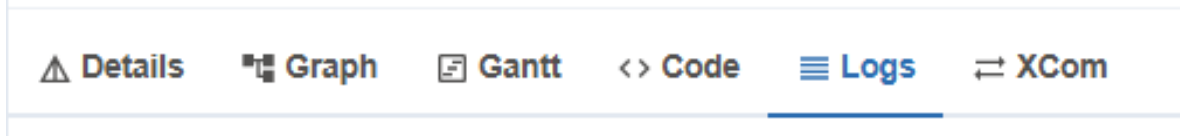


Figura 29: Indices de herramientas Airflow. Fuente: [Airflow docs](#)

```
[2024-03-08, 23:30:18 CET] {logging_mixin.py:188} INFO - Here is some standard text.
[2024-03-08, 23:30:18 CET] {logging_mixin.py:188} INFO - ▼ Non important details
[2024-03-08, 23:30:18 CET] {logging_mixin.py:188} INFO - bla
[2024-03-08, 23:30:18 CET] {logging_mixin.py:188} INFO - debug messages...
[2024-03-08, 23:30:18 CET] {logging_mixin.py:188} INFO - ▲▲▲ Log group end
[2024-03-08, 23:30:18 CET] {logging_mixin.py:188} INFO - Here is again some standard text.
```

Figura 30: Descripción de logs Airflow. Fuente: [Airflow docs](#)

Como se aprecia en la figura [29](#) y [30](#), Airflow tiene un sistema propio de información de registros (logs) que va anotando todo lo que hace cada tarea, tanto la información interna de Airflow para la ejecución de cada tarea como la información que va entregando los procesos del ETL. Esto fortalece la rapidez para corregir eventuales errores en las tareas programadas como también nos libera de tener que crear desde 0 el módulo de logs para nuestro sistema.

B.1.4. Automatización de Ejecución

preset	meaning	cron
None	Don't schedule, use for exclusively "externally triggered" DAGs	
@once	Schedule once and only once	
@hourly	Run once an hour at the beginning of the hour	0 * * * *
@daily	Run once a day at midnight	0 0 * * *
@weekly	Run once a week at midnight on Sunday morning	0 0 * * 0
@monthly	Run once a month at midnight of the first day of the month	0 0 1 * *
@yearly	Run once a year at midnight of January 1	0 0 1 1 *

Figura 31: Variable y parámetros de programación de tareas en Airflow. Fuente: [Airflow docs](#)

Uno de los requisitos de la propuesta de solución es el poder ejecutar los ETL de manera periódica, para esto y gracias a Airflow, se tiene un parámetro en su configuración para programar tareas en horarios y fechas que uno estime necesario.

Como se observa en la figura 31, Airflow nos da una amplia gama de variables pre definidas dentro de su ambiente para sincronizar y ejecutar tareas. Por mencionar, tenemos la diaria, semanal, mensual, entre otros. Además de estas, Airflow también da la oportunidad de crear su propia fecha de ejecución mediante la sintaxis de *cron* 14. Gracias a esto, podemos decidir los segundos, minutos, horas, días, semanas, meses y años en una ejecución de una tarea.

B.1.5. Creación de flujo ETL en Airflow

Ya teniendo las 3 tareas fundamentales de la integración de datos para esta solución, ahora se procede a organizar dichas tareas en un espacio asignado en Airflow. Para esto, se utilizan los DAGs 17.

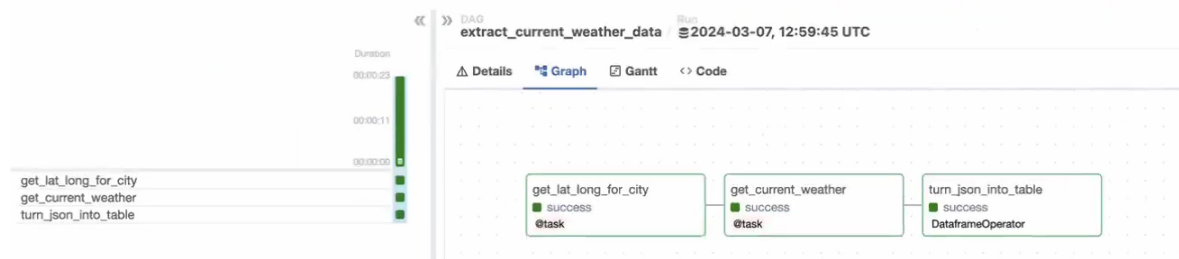


Figura 32: Representación visual de un ETL en Airflow. Fuente: [ETL en Airflow](#)

Como se observa en la figura 32, esta es la manera visual de un ETL (simple) dentro de la herramienta Airflow. Este ejemplo representa a la perfección el flujo considerado en esta propuesta solución, ya que, cada tarea del ETL (Extract, Transform, Load) junto con su lógica queda almacenado en una casilla independiente.

Airflow permite ejecutar las tareas en el orden y condición que uno estime pertinente, pero aterrizado a nuestra solución, debe ir de manera secuencial la ejecución de las tareas, empezando por la extracción, luego por la transformación y finalmente por la carga de los datos. Esto garantiza el correcto flujo de la solución y además crea la dependencia necesaria para que los datos lleguen de manera correcta y trabajada a cada tabla de la base de datos.

Como se menciona en las secciones anteriores, este DAG disfruta de los beneficio de la herramienta: logs en cada casilla, visualización de los tiempos y fechas de ejecución, y programación de ejecución del flujo.

¹⁶Un trabajo cron es una tarea creada utilizando cron , una herramienta para programar y automatizar tareas futuras en sistemas operativos tipo Unix. [cron job](#)

¹⁷Un DAG (gráfico acíclico dirigido) es el concepto central de Airflow, que recopila tareas juntas, organizadas con dependencias y relaciones para indicar cómo deben ejecutarse. [DAG's Airflow](#)

Para una explicación más detallada sobre instalación y/o estructura a nivel de código de Air-flow, consultar Anexo [A](#).

B.2. Integración con Power BI

Una vez que los datos se encuentran almacenados y estructurados en la base de datos PostgreSQL, es necesario proporcionar a los usuarios finales una herramienta que permita la visualización, análisis y explotación de dicha información de forma intuitiva y eficiente.

Para ello, se establece una conexión directa entre **Power BI Desktop** y la base de datos de destino, siguiendo las mejores prácticas de conectividad y seguridad.

B.2.1. Pasos para la Conexión

1. **Instalación de Power BI Desktop:** Se descarga la aplicación oficial desde el sitio web de Microsoft o a través de la Microsoft Store en sistemas Windows.
2. **Inicio de conexión:** Desde la pantalla principal de Power BI, se selecciona la opción Obtener datos → Base de datos → PostgreSQL.
3. **Configuración de credenciales:** Se ingresan los siguientes datos:
 - **Servidor:** Dirección IP o nombre del host donde reside la base de datos.
 - **Base de datos:** Nombre de la base de datos destino.
 - **Autenticación:** Credenciales de usuario y contraseña previamente establecidas, con permisos de solo lectura.
4. **Selección de tablas:** Se eligen las tablas o vistas relevantes para construir los modelos de datos.
5. **Modelado y visualización:** Una vez importados los datos, Power BI permite:
 - Modelar relaciones entre tablas.
 - Crear medidas y columnas calculadas.
 - Construir dashboards dinámicos, gráficos e informes interactivos.

B.2.2. Ventajas de la Conexión Directa

- **Actualización de datos automatizada:** Se pueden programar actualizaciones periódicas de los dashboards, sincronizando directamente con los cambios en la base de datos PostgreSQL.

- **Seguridad y control:** El acceso a la base de datos puede restringirse a usuarios autorizados mediante roles y políticas de seguridad.
- **Escalabilidad:** La conexión soporta el crecimiento del volumen de datos sin necesidad de reconstruir los dashboards.
- **Facilidad de uso:** Power BI proporciona una interfaz amigable para usuarios técnicos y no técnicos, facilitando la adopción institucional de la solución.

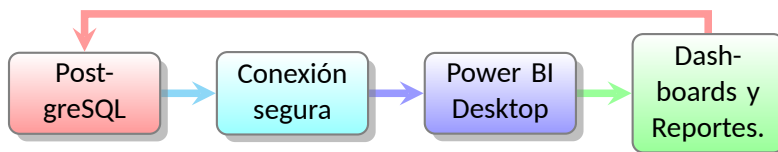


Figura 33: Flujo de integración entre PostgreSQL y Power BI para visualización de datos. Fuente: Elaboración propia

La integración con Power BI representa el eslabón final del proceso, permitiendo transformar los datos almacenados en PostgreSQL en información estratégica, disponible de manera accesible para la toma de decisiones basadas en evidencia dentro de la institución.

B.2.3. Consideraciones sobre herramientas de visualización

La solución propuesta incluye la integración de Power BI como herramienta de visualización y reportería para facilitar la exploración de los datos cargados en la base de datos PostgreSQL. Esta elección responde principalmente a criterios institucionales: Power BI es una herramienta ya adoptada por varias unidades de la UTFSM, cuenta con licencias institucionales activas y es ampliamente utilizada por equipos no técnicos por su facilidad de uso en casos simples de reportería.

No obstante, es importante destacar que Power BI no es la única alternativa viable en el contexto de una arquitectura de datos basada en código abierto. Existen plataformas open source como:

- **Metabase:** Interfaz amigable que permite crear dashboards directamente desde PostgreSQL, sin necesidad de escribir código.
- **Apache Superset:** Plataforma potente para visualización de datos y exploración avanzada, diseñada para entornos productivos.
- **Grafana:** Especializada en monitoreo y visualización de datos en tiempo real, con capacidad de integración con múltiples fuentes.

Estas herramientas podrían considerarse en futuras etapas de evolución del sistema, especialmente si se busca reducir la dependencia de software propietario, aumentar la personalización o permitir una mayor integración con flujos de trabajo de código abierto.

En conclusión, si bien Power BI cumple adecuadamente con los objetivos actuales de visualización y reportería, la arquitectura diseñada mantiene su flexibilidad para soportar otras soluciones de visualización en el mediano plazo, sin requerir modificaciones profundas en los procesos ETL o en el repositorio de datos centralizado.

Este anexo describe con detalle la implementación del sistema de integración de datos mediante flujos ETL, orquestados con Apache Airflow. Se documentan los pasos técnicos involucrados en la conexión a fuentes de datos internas y externas, transformaciones en memoria, carga en PostgreSQL y visualización con Power BI.

Se incluyen fragmentos de código, ejemplos de operadores y figuras que muestran cómo se modelan los flujos en Airflow, así como la lógica de control de errores, registros y programación periódica.

Términos clave:

- *Staging*: Entorno intermedio temporal donde se preparan los datos antes de su carga al sistema destino¹⁸.
- *Cron*: Sintaxis utilizada para definir horarios recurrentes de ejecución en sistemas Unix¹⁹.

¹⁸Zona de staging: espacio temporal dentro del flujo ETL que permite almacenar y transformar los datos previos a su inserción en la base final. Facilita validaciones, limpieza y control de calidad de forma sistemática.

¹⁹Un trabajo cron permite automatizar la ejecución de tareas en momentos definidos, por ejemplo, cada hora o todos los lunes. Ver: https://www-hostinger-com.translate.google/tutorials/cron-job?_x_tr_sl=en&_x_tr_tl=es

Anexo C: Testimonio Completo de Usuario Clave

A continuación, se presenta el testimonio completo proporcionado por Mauricio Saldivia Casanova, Jefe de Gobierno de Datos Institucional UTFSM, recogido mediante un formulario estructurado como parte del proceso de validación del sistema.

Nombre: Mauricio Saldivia Casanova

Cargo: Jefe de Gobierno de Datos Institucional UTFSM

Unidad: Dirección de Transformación Digital

Años de experiencia en gestión de datos institucionales: 5

1. ¿En qué medida considera que la solución implementada ha facilitado su trabajo en relación con la generación de reportes institucionales?

"La solución ha permitido reducir significativamente los tiempos de extracción y procesamiento de información. Antes el proceso era manual, con riesgos de inconsistencia. Hoy los datos están estandarizados y disponibles, lo que mejora la confiabilidad y velocidad para generar reportes."

2. ¿Cómo evalúa la calidad y consistencia de los datos consolidados en el Data Warehouse producto de esta solución?

"Muy buena. Se nota el trabajo de transformación y validación. La calidad de los datos ha mejorado y ya no encontramos duplicados ni valores fuera de rango como antes."

3. ¿Qué impacto ha tenido esta solución en la toma de decisiones dentro de los proyectos SAT-E y SIGPI?

"Ha sido clave. Nos ha permitido construir dashboards confiables para identificar estudiantes en riesgo y analizar la productividad académica. La información es más oportuna y clara para quienes toman decisiones."

4. ¿Considera que esta solución es escalable y puede ser aprovechada por otras unidades dentro de la universidad?

"Absolutamente. La arquitectura es modular y puede extenderse fácilmente a nuevas fuentes. Ya se está considerando su uso en otras áreas, como Dirección de Docencia o Postgrado."

5. ¿Tiene sugerencias o recomendaciones para futuras mejoras del sistema?

"Implementar un catálogo de datos institucional visible para todos los usuarios, y mayor automatización de validaciones para garantizar la integridad en cada nueva carga."

REFERENCIAS BIBLIOGRÁFICAS

- [Abhulimen *et al.*, 2024] Abhulimen, A. O., Adeniran, I. A., Efunniyi, C. P., y Osundare, O. S. (2024). Integrating data analytics in academic institutions: enhancing research productivity and institutional efficiency. *International Journal of Scholarly Research in Multidisciplinary Studies*.
- [Akrami *et al.*, 2024] Akrami, K., Akrami, M., Akrami, F., y Hakimi, M. (2024). Investigating the integration of big data technologies in higher education settings. *Indonesian Journal of Multidisciplinary on Social and Technology*.
- [Amrullah *et al.*, 2021] Amrullah, A. Z., Nurcahyo, A. C., Irfan, P., Hammad, R., y Latif, K. A. (2021). Optimization of data integration using schema matching of linguistic-based and constraint-based in the university database. *Matrix : Jurnal Manajemen Teknologi dan Informatika*.
- [Arnold y Pistilli, 2010] Arnold, K. E. y Pistilli, M. D. (2010). Course signals at purdue: Using learning analytics to increase student success. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pp. 267–270.
- [Badrinath y G, 2024] Badrinath, P. y G, D. (2024). Challenges approaches and methodologies in data integration : A review. *2024 International Conference on IoT, Communication and Automation Technology (ICICAT)*, pp. 462–464.
- [Bansal, 2025] Bansal, D. K. (2025). Enterprise data engineering: Architecting modern data warehouses for business success. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*.
- [Beyan y Kutafina, 2025] Beyan, O. y Kutafina, E. (2025). Real-time automated monitoring architecture for ensuring quality etl processes in healthcare data: A case study from cologne university hospital's medical data integration center. *Studies in health technology and informatics*, 327:193–197.
- [Casanova, 2025] Casanova, M. S. (2025). Testimonio sobre el uso del data warehouse institucional utfsm. Formulario estructurado respondido por el Jefe de Gobierno de Datos UTFSM, mayo 2025.
- [Chen *et al.*, 2023] Chen, L., Baza, M., y Alshahrani, H. (2023). Data integration method of multi-source feedback evaluation for remote teaching quality. *Mobile Networks and Applications*, pp. 1–11.
- [DataCamp, 2024] DataCamp (2024). What is zero-etl? introducing new approaches to data integration.
- [Dayal *et al.*, 2009] Dayal, U., Castellanos, M., Simitsis, A., y Wilkinson, K. (2009). Data integration flows for business intelligence. pp. 1–11.

- [Deen *et al.*, 1987] Deen, S., Amin, R. R., y Taylor, M. (1987). Data integration in distributed databases. *IEEE Transactions on Software Engineering*, SE-13:860–864.
- [Dong *et al.*, 2007] Dong, X., Halevy, A., y Yu, C. (2007). Data integration with uncertainty. *The VLDB Journal*, 18:469–500.
- [Eicher *et al.*, 2019] Eicher, J., Prasser, F., Spengler, H., Kuhn, K., y Bild, R. (2019). Privacy-enhancing etl-processes for biomedical data. *International journal of medical informatics*, 126:72–81.
- [El-Sappagh *et al.*, 2011] El-Sappagh, S., Hendawi, A. M., y El-Bastawissy, A. (2011). A proposed model for data warehouse etl processes. *J. King Saud Univ. Comput. Inf. Sci.*, 23:91–104.
- [Ellucian, 2024] Ellucian (2024). Ellucian banner. <https://www.ellucian.com/es/soluciones/ellucian-banner>. Consultado el 21 de julio de 2025.
- [Elugbaju *et al.*, 2024] Elugbaju, W. K., Okeke, N. I., y Alabi, O. A. (2024). Saas-based reporting systems in higher education: A digital transition framework for operational resilience. *International Journal of Applied Research in Social Sciences*.
- [Fana *et al.*, 2021] Fana, W. S., Sovia, R., Permana, R., e Islam, A. (2021). Data warehouse design with etl method (extract, transform, and load) for company information centre. *International Journal of Artificial Intelligence Research*.
- [Gong, 2025] Gong, L. (2025). Construction and application research of university academic early warning system based on aigc technology. *2025 IEEE International Conference on Electronics, Energy Systems and Power Engineering (EESPE)*, pp. 283–291.
- [Hilali y Ejbal, 2022] Hilali, I. y Ejbal, R. (2022). A new approach for integrating data into big data warehouse. p. 120841M.
- [Jakimoski, 2016] Jakimoski, K. (2016). Challenges of interoperability and integration in education information systems. *International Journal of Database Theory and Application*, 9(2):33–46.
- [Jantanukul, 2024] Jantanukul, W. (2024). Ai and human synergy: Utilizing ai to enhance teaching and learning. *Journal of Education and Learning Reviews*.
- [Khan *et al.*, 2024] Khan, B., Khan, W., Chughtai, M. I., y Jan, S. (2024). An overview of etl techniques, tools, processes and evaluations in data warehousing. *Journal on Big Data*.
- [Kimball y Ross, 2013] Kimball, R. y Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons, 3rd edición.
- [Kohlbacher *et al.*, 2018] Kohlbacher, O., Bauer, B., Kuhn, K., Mansmann, U., y Prasser, F. (2018). Data integration for future medicine (difuture). *Methods of Information in Medicine*, 57:e57 – e65.

- [Kolse et al., 2024] Kolse, P., Pawar, M., Nehul, R., Solanki, S., y Bawankar, C. (2024). Result analysis and visualization to improve decision making for educational institutes. *2024 1st International Conference on Sustainable Computing and Integrated Communication in Changing Landscape of AI (ICSCAI)*, pp. 1–6.
- [Li, 2017] Li, G. (2017). Human-in-the-loop data integration. *Proc. VLDB Endow.*, 10:2006–2017.
- [Macura, 2014] Macura, M. (2014). Integration of data from heterogeneous sources using etl technology. *Computer Science*, 15(2):109–124. Open access under CC BY 4.0.
- [Medina et al., 2018] Medina, F., Fariña, F., y Castillo-Rojas, W. (2018). Data mart para obtención de indicadores de productividad académica en una universidad. *Ingeniare. Revista chilena de ingeniería*, 26(Especial):88–101.
- [Mosse et al., 2023] Mosse, E. L., Nhavoto, J. A., y Uache, J. J. (2023). Integration challenges from the perspective of business intelligence in public universities in mozambique. the case of eduardo mondlane university. pp. 119–139.
- [Neang et al., 2021] Neang, A. B., Beach, M. W., Sutherland, W., y Lee, C. P. (2021). Data integration as coordination. *Proceedings of the ACM on Human-Computer Interaction*, 4:1 – 25.
- [Pelletier y Review, 2021] Pelletier, S. G. y Review, E. (2021). Top it issues, 2021: Emerging from the pandemic. *EDUCAUSE Review*. Disponible en: <https://library.educause.edu/resources/2020/12/2021-top-it-issues-emerging-from-the-pandemic>.
- [Rivera et al., 2009] Rivera, F., Astudillo, P., y Fernández, E. (2009). Información y toma de decisiones: oficinas de análisis institucional en universidades chilenas. *Calidad en la Educación*, (30):162–177.
- [Rodzi et al., 2015] Rodzi, N. H. M., Yusuf, L. M., y Othman, M. (2015). Significance of data integration and etl in business intelligence framework for higher education. *2015 International Conference on Science in Information Technology (ICSITech)*, pp. 181–186.
- [Román-Cañizares et al., 2020] Román-Cañizares, M., Palacios-Pacheco, X., y Villegas-Ch., W. (2020). Integration of iot and blockchain to in the processes of a university campus. *Sustainability*.
- [Sedlmayr et al., 2022] Sedlmayr, M., Reinecke, I., Zoch, M., Henke, E., Bathelt, F., y Peng, Y. (2022). An etl-process design for data harmonization to participate in international research with german real-world data based on fhir and omop cdm. *International journal of medical informatics*, 169:104925.
- [Seenivasan, 2023] Seenivasan, D. (2023). Etl (extract, transform, load) best practices. *International Journal of Computer Trends and Technology*, 71(1):40–44. Open access under CC BY-NC-ND 4.0.

- [Shaffi y Sidhick, 2025] Shaffi, S. M. y Sidhick, J. N. (2025). Streamlining data integration: Architectures for real-time insights and on-demand transformation. *International Journal For Multidisciplinary Research*.
- [Sharwani, 2024] Sharwani, A. E. (2024). Modernizing the u.s. educational system by elevating teaching methods and student performance through human-computer integration, data analytics, and other innovative technologies. *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATM-SI)*, 2:1–6.
- [Simitsis et al., 2005] Simitsis, A., Vassiliadis, P., y Sellis, T. (2005). State-space optimization of etl workflows. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1404–1419.
- [Soriano y Cruz, 2020] Soriano, L. T. y Cruz, J. S. D. (2020). Implementing an etl-driven data integration framework for state universities and colleges. *IOP Conference Series: Materials Science and Engineering*, 803.
- [Sreemathy et al., 2021] Sreemathy, J., Durai, K. N., Priya, E. L., Deebika, R., Suganthi, K., y Aisshwarya, P. (2021). Data integration and etl: A theoretical perspective. *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 1:1655–1660.
- [Tapia y Pinto, 2010] Tapia, L. F. y Pinto, R. V. (2010). Incorporación de elementos de inteligencia de negocios en el proceso de admisión y matrícula de una universidad chilena. *Ingeniare. Revista chilena de ingeniería*, 18(3):383–394.
- [Universidad Técnica Federico Santa María, 2025a] Universidad Técnica Federico Santa María (2025a). Aula virtual utfsm. <https://aula.usm.cl>. Consultado el 21 de julio de 2025.
- [Universidad Técnica Federico Santa María, 2025b] Universidad Técnica Federico Santa María (2025b). Plataforma sharepoint utfsm. <https://usm.cl/sitios-administrativos/>. Consultado el 21 de julio de 2025.
- [Universidad Técnica Federico Santa María, 2025c] Universidad Técnica Federico Santa María (2025c). Sistema de información de gestión académica (siga). <https://siga.usm.cl>. Consultado el 21 de julio de 2025.
- [Vouladonis et al., 2023] Vouladonis, G., Kennedy, C., y Scott, P. (2023). Live learning analytics dashboards: Implementation of real-time data pipelines at ucl. *Journal of Learning Analytics*, 10(1):21–39.
- [Walha et al., 2024] Walha, A., Ghazzi, F., y Gargouri, F. (2024). Data integration from traditional to big data: main features and comparisons of etl approaches. *J. Supercomput.*, 80:26687–26725.

- [Wijaya y Pudjoatmodjo, 2015] Wijaya, R. y Pudjoatmodjo, B. (2015). An overview and implementation of extraction-transformation-loading (etl) process in data warehouse (case study: Department of agriculture). *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pp. 70–74.
- [Yang et al., 2024] Yang, A., Wiratama, J., y Wijaya, S. (2024). Empowering data transformation: Transforming raw data into a strategic planning for e-commerce success. *Journal of Information Systems and Informatics*.
- [Zúñiga, 2024] Zúñiga, N. I. I. (2024). Sistema de acompañamiento a la trayectoria estudiantil sat-e. Trabajo interno, Universidad Técnica Federico Santa María.

BIBLIOGRAFÍA

Definición e información Bases de Datos

- [Bases de datos relacionales y no relacionales - Codespace Academy](#)
- [Diferencias entre bases de datos relacionales y no relacionales - AWS](#)
- [Administración de bases de datos - Grafiati](#)
- [Bases de datos no relacionales - Grafiati](#)
- [Artículo sobre bases de datos - UPM](#)

Data Warehouse, Lake y Mart

- [Data Warehouses vs Data Lakes vs Data Marts - IT Consultors](#)
- [Arquitecturas de almacenamiento de datos - Analytics Lane](#)
- [PDF sobre Data Warehousing en Redalyc](#)
- [Diferencias entre Data Mart, Data Warehouse y Data Lake - Vailos](#)
- [Artículo en SciELO sobre Data Warehousing](#)
- [PDF sobre Data Warehousing - Universidad Nacional de Colombia](#)

Bases de Datos Relacionales

- [SQLite](#)
- [Oracle Database](#)
- [PostgreSQL](#)
- [MySQL](#)