UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

# Entregable 1: Centro Educativo

**Trabajo IAP**

Grado en Ingeniería Informática

*Autor:* Anass Lambaraa Ben Razzouq
Pablo Raga Riera
Enrique Sopeña Urbano
Vicente Rafael Lorente Núñez
*Grupo:*

Curso 2024-2025

# Resumen

Este documento describe el desarrollo de una solución de integración de datos para un Centro Docente, en el marco de la asignatura **Integración de Aplicaciones** del cuarto curso del Grado en Ingeniería Informática. El objetivo es diseñar un módulo de integración utilizando Mule ESB, capaz de procesar y transformar datos generados en distintos formatos (CSV, XML y JSON) provenientes de la aplicación PADRINO. Este módulo también debe adaptarse a múltiples métodos de transferencia, como HTTP y directorios locales, y almacenar la información transformada en la base de datos del centro educativo. La implementación asegura un diseño modular y escalable que facilita la incorporación de nuevas fuentes de datos en el futuro.

**Palabras clave:** integración de datos, Mule ESB, middleware, transformaciones de datos, aplicaciones académicas.

**Palabras clave:** flujo, http, json, csv, inbox

# Abstract

This document describes the development of a data integration solution for an Academic Center, as part of the **Integration of Applications** course in the fourth year of the Computer Engineering degree. The goal is to design an integration module using Mule ESB, capable of processing and transforming data in various formats (CSV, XML, and JSON) generated by the PADRINO application. This module must also adapt to multiple transfer methods, such as HTTP and local directories, and store the transformed information in the Academic Center's database. The implementation ensures a modular and scalable design, facilitating the incorporation of new data sources in the future.

**Keywords:** data integration, Mule ESB, middleware, data transformation, academic applications.

**Key words:** flow, http, json, csv, inbox

# Índice general

**Índice general** II
**Índice de figuras** III

**Introducción** IV
**Diseño del middleware** V
**Resultados** IX
**Conclusiones** X

# Índice de figuras


1 Esquema general de la solución de integración . . . . . IV

2 Flujo principal de la solución . . . . . V
3 Estructura del sistema de archivos . . . . . V
4 Flujo secundario http to inbox$_csv$ . . . . . VI
5 Flujo secundario http to inbox$_XML$ . . . . . VI
6 Flujo secundario http to inbox$_Json$ . . . . . VII
7 Flujo secundario Json to XML (Canonical) . . . . . VII
8 Flujo secundario CSV to XML (Canonical) . . . . . VII

9 Estado final de la base de datos . . . . . IX

# Introducción

El presente trabajo corresponde al primer entregable del proyecto de la asignatura de **Integración de Aplicaciones**. El objetivo principal es diseñar e implementar una solución que permita la integración de datos entre aplicaciones en un entorno académico, utilizando tecnologías avanzadas como Mule ESB y transformaciones de datos mediante herramientas especializadas.

El caso de estudio aborda las necesidades de un **Centro Docente** que requiere optimizar la gestión de sus datos académicos, integrando la información generada por su aplicación PADRINO en formatos heterogéneos (CSV, XML y JSON) con la base de datos institucional. Además, la solución debe adaptarse a múltiples métodos de transferencia, tales como directorios locales y HTTP.

Este documento describe el desarrollo de un módulo de integración diseñado para automatizar los procesos de adquisición, transformación y almacenamiento de datos en la base de datos central. El sistema implementa transformaciones de datos personalizadas y un middleware de integración optimizado, garantizando una interacción eficiente y fluida entre los sistemas involucrados. La arquitectura del módulo ha sido diseñada para minimizar la complejidad y la carga de trabajo asociadas con la incorporación de nuevas funcionalidades. .

Este trabajo es parte de un proyecto global de la asignatura y contribuye significativamente al diseño de soluciones tecnológicas escalables y eficientes dentro del campo de la ingeniería informática.

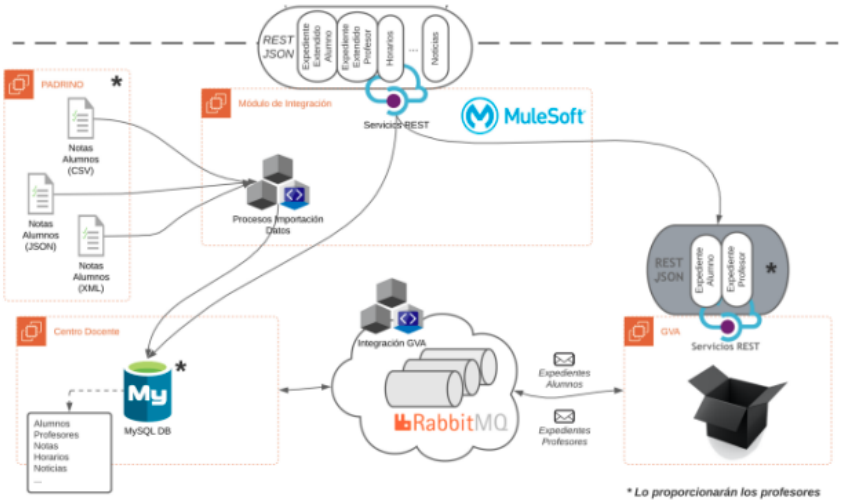



**Figura 1:** Esquema general de la solución de integración

# Diseño del middleware

Para implementar la solución, el sistema debe ser capaz de recibir archivos a través de protocolos HTTP o mediante su incorporación en un directorio local (*inbox*). Con el propósito de garantizar un diseño modular y escalable, se desarrolló un flujo principal estandarizado para trabajar con un único formato de archivo. En este caso, se seleccionó XML como formato debido a su amplia adopción y versatilidad en la actualidad, como se ilustra en la Figura 2.

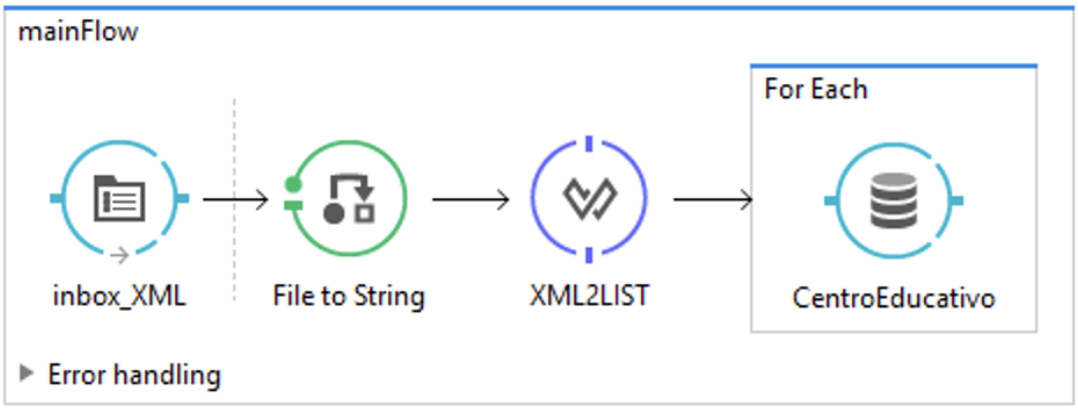
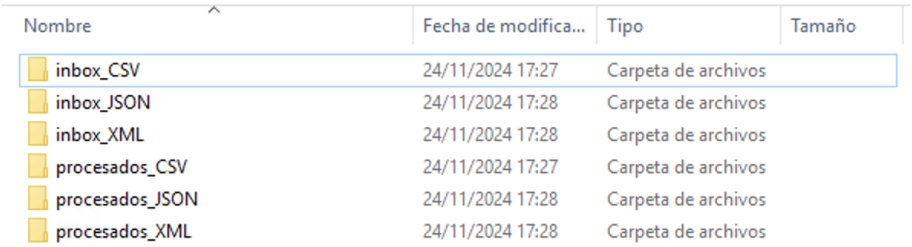



**Figura 2:** Flujo principal de la solución

El flujo principal se complementa con flujos secundarios diseñados específicamente para procesar archivos recibidos a través de conexiones HTTP. Estas conexiones soportan tres formatos de archivo: JSON, XML y CSV. Para gestionar esta variedad, se desarrollaron tres flujos dedicados a procesar cada formato. Cada flujo recibe el archivo correspondiente, lo almacena en el directorio *inbox* asignado al formato respectivo y, en el caso de los formatos CSV y JSON, activa flujos adicionales para transformarlos al formato XML. Una vez transformados, los archivos se trasladan al directorio *inbox_XML*. La Figura 3 ilustra la estructura del sistema de archivos que respalda esta solución. Las Figuras 4, 5, 6, 7 y 8 ilustran en detalle el proceso descrito previamente.



| Nombre | Fecha de modifica... | Tipo | Tamaño |
|---|---|---|---|
| inbox_CSV | 24/11/2024 17:27 | Carpeta de archivos | |
| inbox_JSON | 24/11/2024 17:28 | Carpeta de archivos | |
| inbox_XML | 24/11/2024 17:28 | Carpeta de archivos | |
| procesados_CSV | 24/11/2024 17:27 | Carpeta de archivos | |
| procesados_JSON | 24/11/2024 17:28 | Carpeta de archivos | |
| procesados_XML | 24/11/2024 17:28 | Carpeta de archivos | |

**Figura 3:** Estructura del sistema de archivos

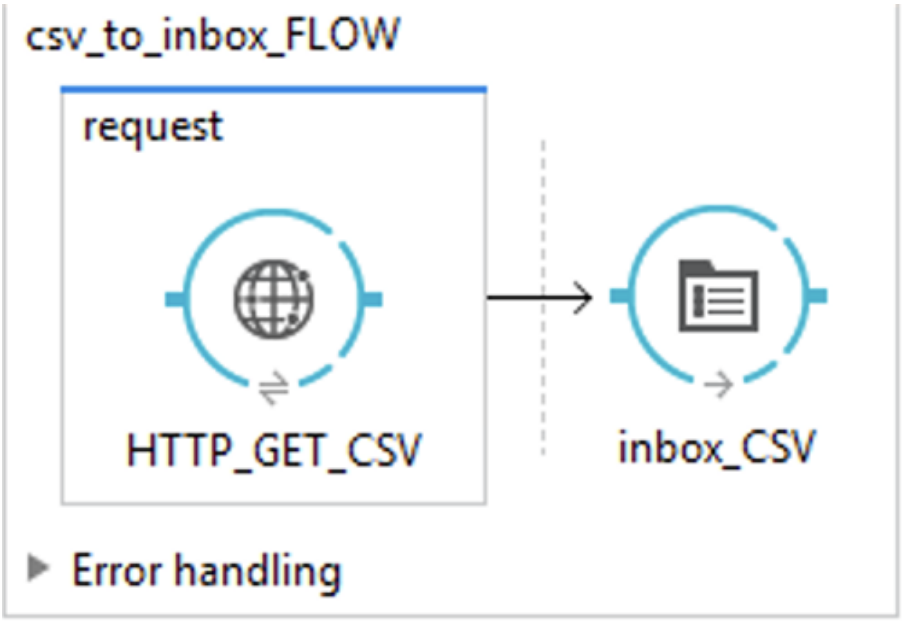



**Figura 4:** Flujo secundario http to inbox$_csv$

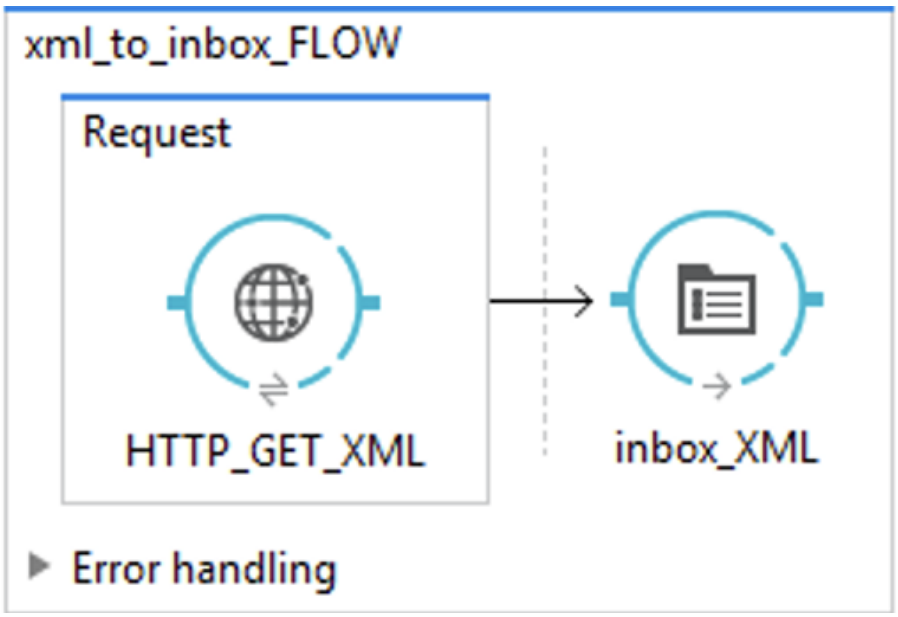



**Figura 5:** Flujo secundario http to inbox$_XML$

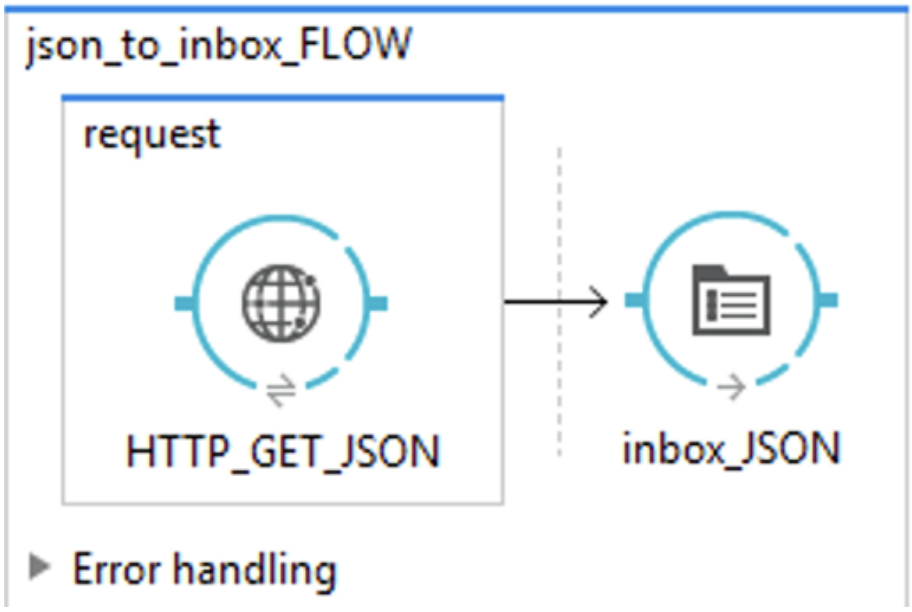



**Figura 6:** Flujo secundario http to inbox$_{J}son$

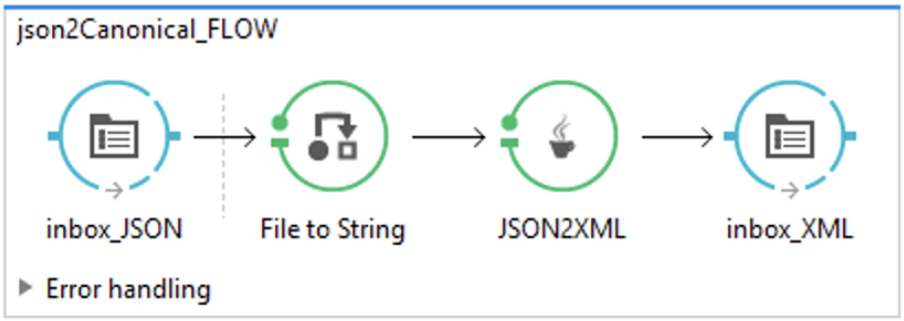



**Figura 7:** Flujo secundario Json to XML (Canonical)

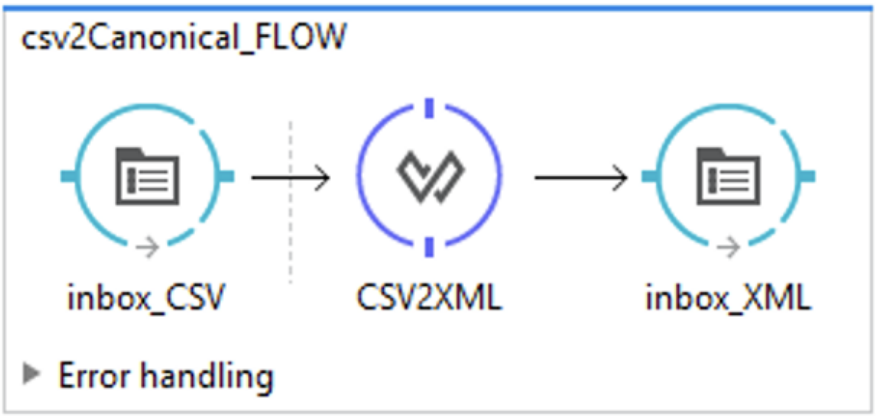



**Figura 8:** Flujo secundario CSV to XML (Canonical)

El diseño modular y la capacidad de adaptarse a múltiples fuentes de datos aseguran la escalabilidad de la solución y su facilidad de mantenimiento. El middleware implementado satisface los requisitos de la solución al habilitar la recepción de archivos en tres formatos distintos (JSON, XML y CSV), ya sea a través de conexiones HTTP o median-

te el sistema de archivos local, organizados en los directorios *inbox_JSON*, *inbox_XML* e *inbox_CSV*.

# Resultados

En este apartado se presenta el estado final de la base de datos tras la ejecución de la solución propuesta. Este análisis permite verificar y validar el correcto funcionamiento del sistema implementado.

La visualización del estado final incluye la estructura, los datos almacenados y las transformaciones realizadas durante el proceso, asegurando que cada uno de los flujos definidos haya operado de manera adecuada y que los datos se hayan integrado correctamente. Esto garantiza que los archivos recibidos en diferentes formatos (JSON, XML y CSV) se hayan procesado, transformado y almacenado según lo especificado en los requisitos del sistema.

Mostrando filas 0 - 17 (total de 18, La consulta tardó 0,0023 segundos.)

```
SELECT * FROM `notas`
```

Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

Mostrar todo | Número de filas: Todos/as Filtrar filas: Buscar en esta tabla

+ Opciones

| ←T→ | alumno | nombre_acto | asignatura | nota |
|---|---|---|---|---|
| Editar Copiar Borrar | 12345678W | Examen | DEW.2020 | 9 |
| Editar Copiar Borrar | 12345678W | Práctica 3 | DEW.2020 | 7.5 |
| Editar Copiar Borrar | 12345678W | st escrito | IAP.2020 | 9 |
| Editar Copiar Borrar | 23456387R | Análisis de necesidades | DCU.2020 | 7 |
| Editar Copiar Borrar | 23456387R | Examen | DEW.2020 | 4.5 |
| Editar Copiar Borrar | 23456387R | Test escrito | DCU.2020 | 6.5 |
| Editar Copiar Borrar | 34567891F | Análisis de necesidades | DCU.2020 | 6.5 |
| Editar Copiar Borrar | 34567891F | Examen | DEW.2020 | 7.5 |
| Editar Copiar Borrar | 34567891F | Práctica 3 | DEW.2020 | 8.5 |
| Editar Copiar Borrar | 34567891F | Prototipo Web | DCU.2020 | 9 |
| Editar Copiar Borrar | 34567891F | Test escrito | DCU.2020 | 7 |
| Editar Copiar Borrar | 34567891F | Test escrito | IAP.2020 | 7.5 |
| Editar Copiar Borrar | 37264096W | Práctica 3 | DEW.2020 | 6.5 |
| Editar Copiar Borrar | 37264096W | Prototipo Web | DCU.2020 | 8.5 |
| Editar Copiar Borrar | 37264096W | Test escrito | IAP.2020 | 9.5 |
| Editar Copiar Borrar | 93847525G | Análisis de necesidades | DCU.2020 | 8.5 |
| Editar Copiar Borrar | 93847525G | Prototipo Web | DCU.2020 | 7.5 |
| Editar Copiar Borrar | 93847525G | Test escrito | DCU.2020 | 9.5 |

**Figura 9:** Estado final de la base de datos

# Conclusiones

El desarrollo del presente trabajo ha permitido aplicar de manera práctica conceptos fundamentales de integración de aplicaciones, utilizando Mule ESB como plataforma principal. La solución diseñada destaca por su modularidad y flexibilidad, permitiendo integrar datos provenientes de diversas fuentes y formatos en una única base de datos de manera eficiente y escalable.

Entre los logros principales del proyecto, se incluye la implementación de flujos que facilitan el procesamiento automatizado de documentos en formatos CSV, XML y JSON. Además, se han diseñado subflujos especializados que adaptan los datos provenientes de diferentes métodos de entrada, como HTTP y directorios locales, hacia un flujo principal que centraliza el procesamiento.

Otro aspecto relevante es la utilización de transformaciones personalizadas, tanto mediante herramientas de transformación estándar como a través de transformadores Java, lo que garantiza la compatibilidad de los datos y su correcta inserción en la base de datos del Centro Docente.

Finalmente, esta experiencia práctica ha sido clave para fortalecer las competencias técnicas en middleware y en la integración de aplicaciones, mostrando la importancia de diseñar sistemas escalables y adaptables que respondan a las necesidades cambiantes de los entornos académicos y tecnológicos.