

Carga de datos de fuentes externas a base de datos relacionales mediante una aplicación

Información de los Autores

Nombre: A. Sánchez Vázquez¹, M. Patiño Ortiz, J.L. Alcaide Juárez
Institución: Instituto Politécnico Nacional
Email: arturosava.ipn@gmail.com
URL ORCID: <https://orcid.org/0009-0005-1338-3682>

Resumen— El uso de Excel en la gestión de datos sigue siendo común en muchas empresas mexicanas, sin embargo, esto dificulta el aprovechamiento de las ventajas de las bases de datos relacionales, que ofrecen mayor capacidad de almacenamiento y mejor integridad de los datos. Es imperante que las empresas, actualmente, implementen herramientas especializadas que les permitan un impacto positivo en su eficiencia, eficacia y planificación, que se refleje en sus áreas financieras, de inventarios, compras, ventas, etc. [1].

Un sistema automatizado que facilite la conversión e importación de datos a un esquema relacional global permitirá:

- Reducir tiempos y costos operativos, eliminando la necesidad de procesos manuales de migración de datos [2].
- Mejorar la integridad y calidad de la información, aplicando reglas de normalización y validación [3].
- Optimizar la integración con sistemas, proporcionando una solución escalable y accesible [4].

El objetivo de este artículo mostrar un Sistema de Información que permite la validación de datos de un archivo Excel y traducirlo a un esquema relacional, aplicando técnicas y herramientas de la ingeniería de software e ingeniería de sistemas, dando como resultados:

- Filtrado y Validación automatizada de datos en archivos Excel antes de insertarlos en la base de datos.
- Detección precisa de tipos de datos (INT, DECIMAL, DATETIME, NVARCHAR, etc.), reduciendo errores de inserción.
- Manejo de errores y generación de reporte detallado de inconsistencias en los datos.
- Optimización del tiempo de procesamiento, reduciendo validaciones manuales.

- Integración con la base de datos, asegurando que los datos insertados sean correctos y consistentes.

Palabras Clave — Automatización, Base de datos, Excel, Sistema de Información

The use of Excel for data management is still common in many Mexican companies; however, this makes it difficult to take advantage of the benefits of relational databases, which offer greater storage capacity and improved data integrity. It is imperative that companies today implement specialized tools that can positively impact their efficiency, effectiveness, and planning, which is reflected in their financial, inventory, purchasing, sales, and other areas.

An automated system that facilitates the conversion and import of data into a global relational schema will:

- Reduce operational times and costs by eliminating the need for manual data migration processes [2].
- Improve data integrity and quality by applying normalization and validation rules [3].
- Optimize system integration by providing a scalable and affordable solution [4].

The objective of this article is to show an Information System that allows the validation of data from an Excel file and translate it into a relational schema, applying techniques and tools from software engineering and systems engineering, giving the following results:

- Automated filtering and validation of data in Excel files before inserting them into the database.
- Accurate detection of data types (INT, DECIMAL, DATETIME, NVARCHAR, etc.), reducing insertion errors.
- Error handling and generation of detailed reports on data inconsistencies.
- Optimization of processing time, reducing manual validations.
- Integration with the database, ensuring that the inserted

data is correct and consistent.

Keywords — Automation, Database, Excel, Information System

I. INTRODUCCIÓN

Microsoft Excel es una herramienta fundamental para la gestión de datos, ampliamente utilizada en empresas y organizaciones debido a su flexibilidad y facilidad de uso para almacenar, organizar y analizar datos. Aunque Excel permite trabajar con grandes volúmenes de datos, es importante señalar que los archivos de Excel no están estructurados como bases de datos y carecen de una integridad referencial adecuada, lo que los hace inapropiados para aplicaciones empresariales más complejas [5].

En 2023, Fernández señala que una base de datos es una colección de información organizada la cual se almacena y gestiona en un sistema informático. Diseñadas para el manejo de grandes cantidades de datos, nos proporciona formas para acceder y seguir almacenando datos para su procesamiento. En cuanto a la carga de información a un sistema de base de datos tiene una gran importancia en distintos contextos como la gestión de datos dentro de organizaciones, empresas o incluso en el ámbito académico.

A diferencia de los archivos Excel, las bases de datos relacionales son adecuadas para consultas avanzadas, el mantenimiento de la integridad referencial y la normalización de datos, lo que las convierte en la opción ideal para gestionar grandes volúmenes de datos con eficiencia [8].

Las bases de datos relacionales ofrecen una solución robusta y escalable para almacenar y gestionar datos, debido a su capacidad para organizar la información en tablas y establecer relaciones entre ellas. El desafío principal cuando se trabaja con archivos desestructurados es la conversión de sus datos desestructurados a un formato adecuado para bases de datos, ya que Excel no sigue un esquema relacional predefinido. Este proceso debe garantizar que los datos se alineen con la estructura de la base de datos sin perder información o generar inconsistencias [3].

Según Morales en el 2023, Excel presenta varias limitaciones frente a una base de datos, entre ellas:

- Formato de datos: Excel puede cambiar formatos automáticamente, lo que genera problemas de compatibilidad. En cambio, las bases de datos relacionales manejan tipos de datos definidos, garantizando consistencia.
- Capacidad de almacenamiento: Excel tiene un límite de filas y columnas, lo que puede ocasionar pérdida de datos, como ocurrió en Reino Unido con casos de

COVID-19.

- Rendimiento: A medida que aumenta el volumen de datos, Excel se ralentiza y pierde eficiencia.
- Integridad de datos: La existencia de múltiples copias y la posibilidad de sobrescribir datos generan riesgos de pérdida o corrupción de información.
- Datos accionables: Excel es un sistema pasivo, mientras que las bases de datos pueden integrarse con sistemas que automatizan alertas y acciones.
- Seguridad: En Excel, los datos pueden ser modificados o extraídos fácilmente, mientras que las bases de datos permiten gestionar permisos y accesos de forma segura.
- Capacidades analíticas: Las bases de datos permiten cruzar múltiples fuentes de información y generar análisis avanzados, optimizando la toma de decisiones.

Estos puntos implican un reto al querer realizar la importación de información a tablas y columnas de una base de datos.

Existen distintos sistemas que sirven para importar información de Excel a base de datos como son:

1) SQL Server Import Wizard

Herramienta que facilita la importación de datos Excel, hacia SQL Server. Sin embargo, no ofrece opciones avanzadas como la validación de datos o la creación automática de tablas, lo que lo hace adecuado solo para tareas de importación básica [9].

2) Power Automate

Permite automatizar tareas repetitivas y flujos de trabajo, y puede integrarse con otros servicios de Microsoft para importar y validar datos. Sin embargo, no tiene capacidades avanzadas como la creación automática de tablas base de [10].

3) Excel VBA (ODBC/OLEDB)

Permite la automatización de tareas dentro de hojas de cálculo de Excel, incluidas las importaciones de datos mediante conexiones ODBC o OLEDB. Sin embargo, no ofrece opciones para la validación o creación de tablas [11].

4) Python (pandas + SQLAlchemy)

Solución poderosa para el análisis y la manipulación de datos, utilizando pandas para la gestión de datos tabulares y SQLAlchemy para la interacción con bases de datos. Aunque es muy eficaz para la importación y procesamiento de datos, no ofrece validación o creación automática de tablas [12].

5) Talend Open Studio

Herramienta de integración de datos que permite realizar procesos ETL. Ofrece capacidades para la integración de datos desde diversas fuentes, pero no incluye validación avanzada ni la creación de tablas [13].

6) Pentaho Data Integration (PDI)

También conocido como Kettle, es una herramienta de integración de datos que soporta tareas de extracción, transformación y carga de datos. Sin embargo, no ofrece opciones para la creación de tablas o automáticamente [14].

A continuación, se muestra la comparativa con respecto a los sistemas mencionados anteriormente,

Tabla 1. Comparativa de Herramientas para Importar Excel a Bases de Datos. Elaboración propia 2025

Sistema	importación de Datos	Validación de Datos	Creación de Tablas
SQL Server Import Wizard	✓	✗	✗
Power Automate	✓	✗	✗
Excel VBA (ODBC/OLEDB)	✓	✗	✗
Python (pandas + SQLAlchemy)	✓	✗	✗
Talend Open Studio	✓	✗	✗
Pentaho Data Integration (PDI)	✓	✗	✗
Aplicación IPN	✓	✓	✓

II. METODOLOGÍA/DESARROLLO

El Sistema de Información desarrollado está diseñado para automatizar la importación de datos desde fuentes externas hacia una base de datos relacional, generando dinámicamente las tablas SQL con base en los tipos de datos detectados. Su objetivo principal es garantizar que la información importada sea precisa, estructurada y compatible con la base de datos destino.

El sistema analiza los datos del archivo fuente y determina automáticamente los tipos de datos adecuados para cada columna. En caso de detectar valores con tipos de datos mixtos en una misma columna, se genera un archivo de errores con los detalles de las inconsistencias encontradas. Además, el sistema identifica relaciones entre tablas, reconociendo claves foráneas cuando un campo, distinto al primero, coincide con la clave primaria de otra tabla.

Una vez validados y estructurados los datos, el sistema crea las tablas necesarias y realiza la importación de la información, asegurando su correcta integración en la base de datos. Con estas funcionalidades, la aplicación optimiza el proceso de migración de datos, reduciendo errores manuales y mejorando la calidad de la información almacenada.

A continuación, se detallan los principales beneficios:

Automatización del proceso de importación

- Generación automática de tablas SQL según los tipos de datos detectados.
- Identificación y asignación de claves foráneas para mantener la integridad referencial.
- Eliminación de la necesidad de definir manualmente la estructura de la base de datos.

Validación y control de calidad de los datos

- Detección de errores en tipos de datos mixtos en una misma columna.
- Generación de un archivo de errores con detalles de inconsistencias detectadas.
- Garantía de que los datos almacenados sean correctos y estructurados.

Optimización de tiempos y reducción de errores manuales

- Minimización de la intervención manual en la estructuración de datos.
- Importación rápida y precisa, reduciendo el margen de error en la gestión de información.
- Mayor eficiencia en el procesamiento de grandes volúmenes de datos.

Mejora en la integridad y consistencia de la base de datos

- Identificación automática de relaciones entre tablas para mantener la coherencia.
- Creación de tablas con estructuras optimizadas para su uso en consultas y reportes.
- Reducción de redundancias y errores de relación entre registros.

Se utilizó una metodología basada en el Modelo de Ciclo de Vida de Desarrollo de los Sistemas con un enfoque sistémico, la cual abarca los siguientes pasos:

Análisis

- Se estudió el proceso de importación de datos desde archivos Excel a bases de datos relacionales.
- Se identificaron los desafíos comunes, como la detección de tipos de datos, la validación de estructuras y la gestión de claves foráneas.
- Se documentó la situación actual y se definieron los requerimientos clave y la automatización de la estructura SQL.

Diseño

- Se desarrollaron los algoritmos encargados de analizar los datos, detectar tipos de columnas y validar inconsistencias.

- Se estableció un método para la identificación de relaciones entre tablas mediante la detección de claves foráneas.
- Se diseñó un método para la generación de un archivo de salida si es que existen errores en los datos.

Desarrollo

- Se seleccionó C# y .NET como entorno de desarrollo, aprovechando su integración con bibliotecas para la manipulación de archivos Excel y conexión con bases de datos SQL.
- Se implementaron los algoritmos de análisis de datos, generación de scripts SQL y validación de integridad referencial.
- Se construyó el método de generación de errores para garantizar la calidad de los datos importados.

Pruebas

- Se realizaron pruebas con diferentes archivos Excel para validar la correcta detección de tipos de datos y relaciones entre tablas.
- Se ejecutaron escenarios controlados para evaluar el desempeño del sistema en la generación de estructuras SQL y la importación de datos.
- Se obtuvieron retroalimentaciones de usuarios finales para mejorar la usabilidad y precisión del sistema.

Implementación

- Se desplegó la aplicación en un entorno de desarrollo, asegurando su correcto funcionamiento en la importación y estructuración de datos.
- Se implementó un monitoreo continuo para detectar posibles errores y optimizar el rendimiento.

El procedimiento que se debe llevar a cabo en el SI sigue los siguientes pasos:

1; La aplicación deberá tener contemplada los datos de conexión del servidor de base de datos a donde se realizará la importación.

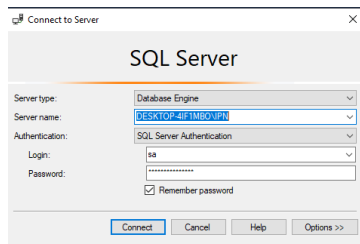


Figura 1.- Servidor SQL Server de base de datos. Elaboración propia 2025

1; Se debe seleccionar por hojas la información a ser importada a tablas nuevas o existentes de la base de datos:



Figura 2.- Hojas de Excel a importar. Elaboración propia 2025

2; La aplicación tomará de la ruta configurada el archivo Excel para posterior de cada hoja se tome la primera columna como la llave primaria de dicha tabla, si existe una llave foránea deberá ir en una columna posterior a la primera columna:

	A	B	C	D	E	F
1	ID	dato1	dato2	dato3	dato4	dato5
2	001	001	001	001	001	001
3	002	003	002	002	004	005
4	003	003	003	003	003	003
5	004	005	006	007	008	009
6	005	005	005	005	005	005
7	006	006	006	006	006	006

Figura 3.- Hoja de excel con llave primaria. Elaboración propia 2025

	A	B	C	D	E	F
1	ID2	ID	dato2	dato3	Fecha	dato5
2	1	001	1	1.1	02/03/2025	Arturo
3	2	002	3	1.2	03/03/2025	Juan
4	3	003	5	1.3	04/03/2025	Luis
5	4	004	7	1.4	05/03/2025	Miguel
6	5	005	9	1.5	06/03/2025	Julian
7	6	006	11	1.6	07/03/2025	Jose

Figura 4.- Hoja de excel con llave primaria y llave foranea. Elaboración propia 2025

3; La aplicación realizará el proceso de detección de tipos de datos de las columnas de la hoja de Excel si estas contemplan distintos tipos de datos se generará un archivo .txt con los errores detectados, no generará tabla ni importación de datos hasta corregir y pasara a la siguiente hoja para realizar el mismo proceso.

```

02/03/2025 12:25:36 a. m.:ERROR en Hoja 'EjemploTabla', Columna 5: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:25:36 a. m.:ERROR en Hoja 'EjemploTabla', Columna 6: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:27:28 a. m.:ERROR en Hoja 'EjemploTabla', Columna 1: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:27:28 a. m.:ERROR en Hoja 'EjemploTabla', Columna 2: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:27:28 a. m.:ERROR en Hoja 'EjemploTabla', Columna 3: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:27:28 a. m.:ERROR en Hoja 'EjemploTabla', Columna 4: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:27:28 a. m.:ERROR en Hoja 'EjemploTabla', Columna 5: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:27:28 a. m.:ERROR en Hoja 'EjemploTabla', Columna 6: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:40:32 a. m.:ERROR en Hoja 'EjemploTabla', Columna 1: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:40:32 a. m.:ERROR en Hoja 'EjemploTabla', Columna 2: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:40:32 a. m.:ERROR en Hoja 'EjemploTabla', Columna 3: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:40:32 a. m.:ERROR en Hoja 'EjemploTabla', Columna 4: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:40:32 a. m.:ERROR en Hoja 'EjemploTabla', Columna 5: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:40:32 a. m.:ERROR en Hoja 'EjemploTabla', Columna 6: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:41:22 a. m.:ERROR en Hoja 'EjemploTabla', Columna 1: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:41:54 a. m.:ERROR en Hoja 'EjemploTabla', Columna 2: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:42:00 a. m.:ERROR en Hoja 'EjemploTabla', Columna 3: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:42:03 a. m.:ERROR en Hoja 'EjemploTabla', Columna 4: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:42:03 a. m.:ERROR en Hoja 'EjemploTabla', Columna 5: Datos mixtos (String: 0, Double: 1, DateTime: 1)
02/03/2025 12:42:15 a. m.:ERROR en Hoja 'EjemploTabla', Columna 6: Datos mixtos (String: 0, Double: 1, DateTime: 1)
    
```

Figura 5.- Archivo .txt con errores detectados. Elaboración propia 2025

4; Si no hay errores la aplicación procederá a revisar si existe o genera la tabla de la hoja de Excel, el nombre de la tabla que se asignará será el mismo que tiene la hoja de Excel.

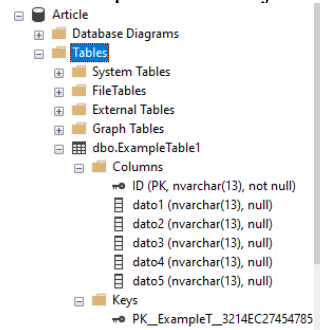


Figura 6.- Tabla generada en base de datos con llave primaria. Elaboración propia 2025

5; Se realizará el proceso de creación de llaves foráneas si es que existen:

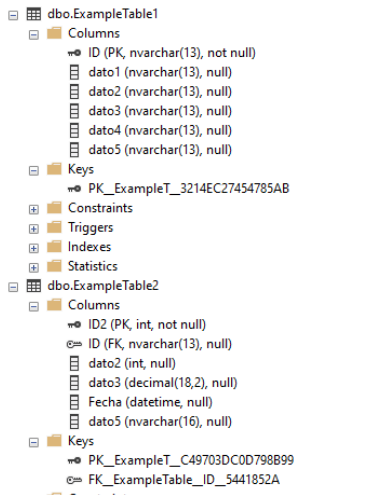


Figura 7.- Tabla generada en base de datos con llave primaria y asignación de llave secundaria. Elaboración propia 2025

III. RESULTADOS Y DISCUSIÓN

La ejecución de la aplicación nos da como resultado la carga de información en las tablas, tipos de datos, llaves primarias y foráneas generadas automáticamente:

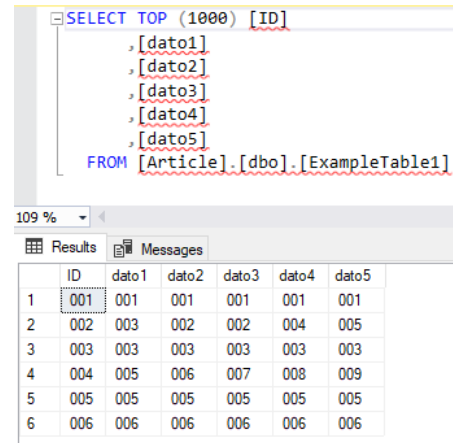


Figura 8.- Consulta de información a tabla ExampleTable1. Elaboración propia 2025

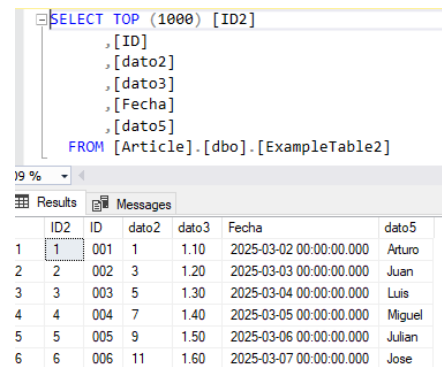


Figura 8.- Consulta de información a tabla ExampleTable2. Elaboración propia 2025

IV. CONCLUSIONES

Este trabajo resalta la importancia de implementar un proceso eficiente y automatizado para la carga de datos desde fuentes externas a bases de datos relacionales. El SI desarrollado no solo facilita la creación automática de tablas y columnas adecuadas, sino que también define de manera precisa las llaves primarias y foráneas, asegurando una estructura de datos coherente y bien organizada.

Uno de los aspectos clave es la capacidad de la aplicación para detectar diferentes tipos de errores en los datos, tales como inconsistencias en los tipos de datos, valores inesperados o formatos incorrectos. Esto se logra mediante una validación exhaustiva antes de la inserción en la base de datos. Además, la herramienta genera un archivo detallado con los errores encontrados, lo que permite a los usuarios corregir los problemas antes de proceder con la importación.

El proceso culmina con la importación de los datos solo si han pasado todas las validaciones necesarias, garantizando que la información ingresada en la base de datos sea correcta y consistente. Este enfoque no solo optimiza el tiempo de procesamiento al eliminar pasos manuales, sino que también mejora la calidad y la integridad de los datos, reduciendo el riesgo de errores en el sistema y mejorando la eficiencia en el manejo de grandes volúmenes de información.

En resumen, el SI desarrollado proporciona una solución robusta para la carga de datos externos a bases de datos relacionales, asegurando que los datos sean validados, consistentes y correctamente estructurados antes de su importación, lo que optimiza el flujo de trabajo y mejora la fiabilidad del sistema en su conjunto.

AGRADECIMIENTOS

Agradezco a Sección de Estudios de Posgrado e Investigación de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco

REFERENCIAS

- [1] <https://asem.mx/empresas-en-mexico-no-planifican-sus-finanzas-estudio-de-gestion-financiera-de-la-asem/>
- [2] Pressman, R. (2014). Ingeniería del Software: Un Enfoque Práctico. McGraw-Hill.
- [3] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Fundamentos de Bases de Datos. McGraw-Hill.
- [4] Freeman, A. (2020). Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. Apress.
- [5] Walkenbach, J. (2019). Excel Bible 2019. Wiley.
- [6] Fernández(2023).https://www.researchgate.net/publication/369039369_Pequena_introduccion_a_las_bases_de_datos
- [7] Morales, A. (2023). 7 razones por las que Excel no es recomendable frente a una base de datos. Biuwer. <https://biuwer.com/es/blog/por-que-no-debes-utilizar-excel-como-base-de-datos>
- [8] Date, C. J. (2021). An Introduction to Database Systems. Pearson.
- [9] Chugugrace et al. (2024) Start the SQL Server Import and Export Wizard. <https://learn.microsoft.com/en-us/sql/integration-services/import-export-data/start-the-sql-server-import-and-export-wizard?view=sql-server-ver16>
- [10] Kisubedi et al. (2024) Introducción a Power Automate. <https://learn.microsoft.com/es-es/power-automate/getting-started>
- [11] SerdarSoysal et al. (2023) Compartir una conexión OLE DB u ODBC con los Servicios de Excel (SharePoint Server 2013). <https://learn.microsoft.com/es-es/sharepoint/administration/share-an-ole-db-or-odbc-connection-using-excel-services-sharepoint-server-2013>
- [12] Awan, A. (2024). sing SQL with Python: SQLAlchemy and Pandas. <https://www.kdnuggets.com/using-sql-with-python-sqlalchemy-and-pandas>
- [13] Talend. (s.f.). Talend Open Studio para la integración de datos. <https://www.talend.com/products/talend-open-studio/>
- [14] Pentaho. (n.d.). Pentaho Data Integration. Pentaho. <https://pentaho.com/products/pentaho-data-integration/>