



UNIVERSIDAD DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA GRADUADO EN INGENIERÍA INFORMÁTICA

Desarrollo de una aplicación web para el acceso, optimización y configuración de un almacén de datos

Realizado por
Alejandro Guzmán Serrano

Tutorizado por
Rafael Marcos Luque Baena
Enrique Soler Castillo

Departamento

Lenguajes y Ciencias de la
Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2023

Resumen

El objetivo de este proyecto es desarrollar una aplicación web que automatice y simplifique la elaboración de documentos para el Servicio de Calidad utilizando datos que provienen de un almacén (o Data Warehouse), así como nuevos datos aportados en ficheros de rendimiento. Los resultados obtenidos se pueden comparar con ficheros generados manualmente o mediante otros procesos, mostrando posibles conflictos, permitiendo ver todos los registros comparados. Estos documentos, que contienen una gran cantidad de datos, se envían a instituciones gubernamentales para revisar y estudiar los grados universitarios en España.

La aplicación se ha diseñado para ser sencilla, eficiente y fácil de usar, con el objetivo de convertirse en una herramienta útil para los trabajadores del Servicio de Calidad, los usuarios finales. Se requiere iniciar sesión con las credenciales del servidor que aloja la aplicación, restringiendo su acceso a quienes tienen permisos en dicho equipo. Permite importar un archivo comprimido o importar archivo a archivo con los datos correspondientes a un curso académico específico para agregarlos al almacén de datos y generar el documentos para el Servicio de Calidad utilizando los datos almacenados.

El desarrollo se basa en el diseño previo del almacén de datos y el proyecto de Extracción, Transformación y Carga de datos (ETL), realizando modificaciones específicas para este nuevo desarrollo con el objetivo de lograr una mayor eficiencia y parametrización. Estos proyectos se han integrado en una única aplicación web siendo simplificados y mejorando su accesibilidad. Para la implementación, se utiliza Flask como framework, Python como lenguaje de programación para el back-end, y HTML, JavaScript y CSS para el front-end.

La metodología de desarrollo empleada es SCRUM, una metodología ágil basada en ciclos de trabajo iterativos. Se enfoca en entregar resultados de alta calidad de forma continua y en adaptarse rápidamente a los cambios en los requisitos del proyecto y las necesidades del cliente.

Palabras Clave

Automatización, aplicación web, documentos, almacén de datos.

Abstract

The objective of this project is to develop a web application that automates and simplifies the creation of quality documents using data stored in a warehouse, as well as new data provided in performance files. The obtained results can be compared with other cases, displaying potential conflicts and allowing for the visualization of all compared records. These quality documents, containing a significant amount of data, are sent to government institutions for review and analysis of university degrees in Spain.

The application has been designed to be simple, efficient, and user-friendly, aiming to serve as a valuable tool for the Quality Service workers, the end users. Login credentials from the hosting server are required for access, ensuring restricted usage to authorized personnel. It facilitates the importation of compressed files or individual file imports containing data for a specific year, enabling the addition of this data to the warehouse and generating the quality document utilizing the stored data.

The development is based on the previous design of the data warehouse and the Extraction, Transformation, and Loading (ETL) project, incorporating specific modifications for enhanced efficiency and parameterization in this new development. These projects have been consolidated into a single web application, streamlining processes and improving accessibility. Flask is utilized as the implementation framework, with Python serving as the back-end programming language, and HTML, JavaScript, and CSS employed for the front-end.

The development methodology employed is SCRUM, an agile approach based on iterative work cycles. It emphasizes the continuous delivery of high-quality results and the ability to adapt swiftly to changes in project requirements and customer needs.

Keywords

Automation, web application, documents, data warehouse.

Índice

Resumen.....	2
Palabras Clave.....	2
Abstract.....	3
Keywords.....	3
Índice.....	4
1. Introducción.....	6
1.1 Objetivos.....	8
1.2 Motivación.....	10
1.2 Estructura de la Memoria.....	10
2. Tecnologías utilizadas.....	13
2.1 Lenguaje de programación para el backend.....	13
2.2 Lenguaje de programación para el frontend.....	14
2.3 Entorno de trabajo.....	15
3. Extensión de trabajos previos.....	17
4. Almacén de datos.....	19
4.1 Fuentes de datos.....	19
4.2 Definición ETL.....	24
4.3 Modificaciones en el ETL.....	25
4.3 Definición Almacén de datos.....	28
4.3 Modificaciones al Almacén de datos.....	30
5. Metodología.....	33
6. Fases de desarrollo.....	37
6.1 Planificación.....	37
6.2 Análisis de requisitos.....	38
6.3 Diagramas.....	40
6.3.1 Casos de Uso.....	40
6.3.2 Diagrama de secuencia.....	41
6.4 Programación e Implementación.....	43
6.4.1 Inicio Sesión.....	43
6.4.2 Menú.....	46
6.4.2 Cargar Datos.....	47
6.4.3 Ejecutar comando.....	51

6.4.4 Crear fichero de salida.....	53
6.4.5 Comparación Ficheros.....	54
6.4.6 Ver resultados.....	57
7. Conclusiones.....	60
8. Referencias.....	62
9. Apéndices.....	64
MANUAL DE USUARIO.....	64
GUÍA DE INSTALACIÓN.....	75

1. Introducción

El almacenamiento de datos personales se ha vuelto una tarea fundamental para prácticamente todas las instituciones, sin embargo, los datos en una base de datos o almacén por sí solos no proporcionan una gran cantidad de información. Es por ello que el análisis de datos se ha convertido en un aspecto crucial en el campo de la informática actual. El éxito de este análisis depende tanto de la cantidad como de la calidad de los datos disponibles. Para lograr un análisis efectivo, es igualmente importante contar con una gran cantidad de datos, así como asegurarse de que dichos datos sean de la mejor calidad posible. Asegurar la calidad de los datos suele ser lo más complejo, ya que los datos almacenados pueden tener errores, pueden estar desactualizados o incluso que no se hayan almacenado de la forma correcta.

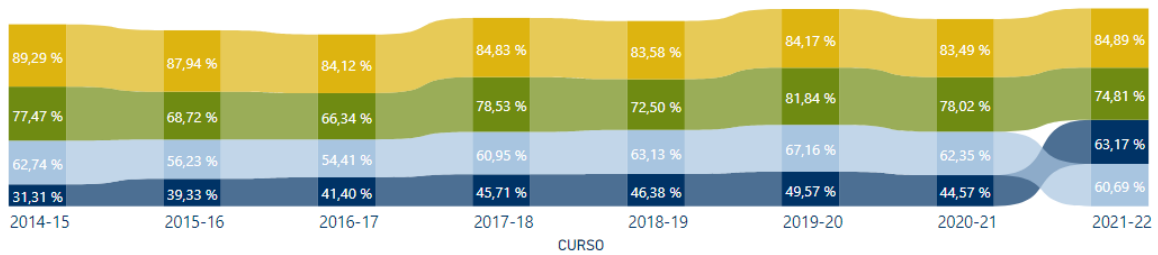
Las instituciones educativas deben almacenar información sobre sus estudiantes que será fundamental para la evaluación del rendimiento de la propia institución. Es aquí donde la Universidad de Málaga debe generar un informe anual sobre los datos de sus alumnos para el Ministerio de Educación. Estos informes son cruciales para el estudio del rendimiento de las universidades y de los grados. Los análisis de estos documentos ayudan a la toma de decisiones por parte del Ministerio. A partir de ellos se puede obtener información sobre los grados con mayor tasa de fracaso, lo cual permite organizar planes de mejora. También es posible identificar los grados con mayor número de alumnos graduados o matriculados, lo cual nos brinda información indirecta sobre las personas jóvenes que están ingresando o ya se encuentran en el mercado laboral.

Estos datos no sólo son útiles para el Ministerio, sino que también pueden ser analizados para identificar patrones y tendencias, para la autoevaluación del rendimiento de la universidad y para tomar medidas para la mejora de la calidad de la enseñanza. Por ejemplo, se puede detectar fácilmente cuáles son las materias en las que los alumnos tienen más dificultades y puedan necesitar mayor apoyo.

En el caso de la Universidad de Málaga, se dispone de un apartado en su propia página web, que recientemente ha sido renovado mostrando los datos ahora de una forma más visual, en el que los usuarios pueden consultar ellos mismos algunos de estos datos aportados por el Servicio de Calidad de la Universidad de Málaga [1]. En el ejemplo de la Figura 1.1 podemos ver los datos de los alumnos de la titulación Ingeniería Informática.

Tasa de Rendimiento por año donde se imparten la asignatura

Año de impartición ● 1 ● 2 ● 3 ● 4



Resultados obtenidos por curso

● Suma de Num. No Present... ● Suma de Num. Suspensos ● Suma de Num. de Apr... ● Suma de Num. Not... ● Suma de Num. So... ● Suma de Num. ...

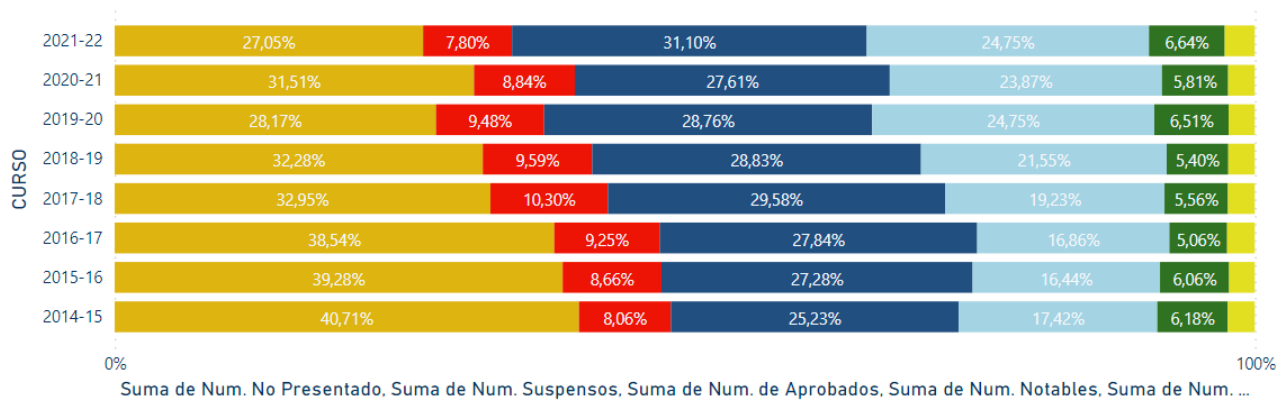


Figura 1.1: Calidad UMA Resultados por título y asignaturas

Aunque este apartado se ha renovado recientemente la elaboración de los ficheros de rendimiento nombrados anteriormente se siguen haciendo en mayor parte de forma casi manual. Con el fin de automatizar este proceso, se han llevado a cabo varios proyectos de fin de grado. Estos proyectos buscan reducir la carga de trabajo manual mediante la implementación de soluciones como el diseño de un almacén de datos específico para esta tarea y la automatización de la carga de datos en dicho almacén. El problema es que para realizar dicha carga o para la generación del documento a partir del almacén es necesario tener ciertos conocimientos básicos sobre *SSIS (SQL Server Integration Services)* y *SQL Server* respectivamente, lo que causaba que aunque fueran funcionales, no acabaran de ser implementados.

La simplificación y automatización de los procesos de carga de datos sobre el almacén, generación y comparación del documento para el Servicio de Calidad para que cualquier persona con conocimientos básicos de informática sea capaz de realizarlo, son los principales objetivos de este proyecto.

El usuario tendrá que aportar ciertos ficheros de datos que dispone el Servicio de Calidad cada curso, y podrá cargar el almacén con esos datos de forma simple, para después generar el documento del curso académico. Estos documentos se podrán seguir haciendo de forma semi manual como se ha estado haciendo hasta ahora, por eso se aportará también una herramienta de comparación de ficheros de calidad para poder identificar fácilmente las diferencias entre el obtenido de forma manual y el generado a partir del almacén.

1.1 Objetivos

El objetivo principal es desarrollar una aplicación web que integre los diferentes proyectos desarrollados para el Servicio de Calidad, con el propósito de brindar una utilidad real y un funcionamiento sencillo para los usuarios. Se busca mejorar la accesibilidad de las herramientas existentes, facilitando el acceso y la utilización de las mismas. Además, la aplicación tendrá la capacidad de identificar los puntos donde se producen fallos en la automatización de la creación de los ficheros de rendimiento. Esto permitirá realizar las modificaciones necesarias en los ficheros de entrada, en el proceso de Extracción, Transformación y Carga de datos (ETL), o en el propio almacén de datos, con el objetivo de mejorar el rendimiento y la calidad de los resultados obtenidos.

La aplicación web actuará como una plataforma centralizada que integrará y gestionará todas las etapas del proceso, desde la importación de los ficheros de rendimiento hasta la generación de los documentos para el Servicio de Calidad. De esta manera, se optimizará la eficiencia y se reducirán los errores o inconsistencias en el proceso de elaboración de los documentos. Al mejorar la accesibilidad de las herramientas existentes y proporcionar una solución integrada, se espera facilitar el trabajo de los usuarios del Servicio de Calidad, optimizando su productividad y permitiéndoles identificar y corregir rápidamente cualquier problema en la automatización de la creación de los ficheros de rendimiento.

Este trabajo de fin de grado tiene varios subobjetivos, los cuales serán descritos a continuación:

- **Simplificar la carga de datos del almacén:** desarrollar una interfaz fácil e intuitiva que permita al usuario identificar claramente los archivos necesarios para la carga de datos. Posteriormente, se busca proporcionar una forma sencilla para que el usuario pueda proporcionar dichos archivos, sin requerir conocimientos previos sobre SQL Server o SSIS.
- **Automatizar la elaboración de los ficheros de rendimiento:** hasta ahora para poder obtener el documentos para el Servicio de Calidad a partir de la base de datos primero se debía ejecutar una consulta en el almacén de datos en *SQL Server Management Studio* y después copiar el resultado obtenido y pegarlo en una plantilla de un fichero *excel*, para por último exportarlo en formato *XML*. Este procedimiento se puede automatizar para que el usuario no necesite tener los conocimientos de esas tecnologías y además minimizar la posibilidad del fallo humano.
- **Comparar documentos, generado con supuesto ideal:** como se ha mencionado anteriormente, el documento para el Servicio de Calidad se podrá seguir haciendo de forma manual, y esta herramienta permitirá comparar el documento generado a partir del almacén con el manual, permitiendo ver el porcentaje de similitud y destacando aquellos registros en los que se obtengan mayor número de diferencias.
- **Adaptar e integrar a otros proyectos:** se tomarán de base dos proyectos para ampliarlos y realizar una conexión entre ellos para integrarlos en un mismo proyecto funcional y útil. En particular se tomará el diseño del almacén de datos y la consulta para la elaboración del documento para el Servicio de Calidad.

1.2 Motivación

El origen de este trabajo de fin de grado se da a causa de la situación del Servicio de Calidad de la Universidad de Málaga, que ya ha sido objetivo de varios trabajos de fin de grado, pero que no están siendo usados, por lo que con este se busca dar una **cohesión** a estos y poder integrales en un sólo proyecto para simplificar su uso.

También se busca una mejora en el trabajo del Servicio de Calidad, que se ven obligados todos los años a invertir gran cantidad de horas en revisar una gran cantidad de datos de forma manual. La automatización de algunas de estas tareas puede reducir drásticamente el tiempo empleado en la elaboración de este informe. Al obtener un modelo de base automáticamente y compararlo con el documento generado manualmente, es posible identificar las diferencias y estudiar su origen. De esta manera, se logra **mejorar la precisión** del documento generado a partir del almacén de datos en cada iteración.

Aún con este proyecto, esta línea sigue abierta para posibles ampliaciones que puedan agregarse al proyecto para hacerlo cada vez más completo y una herramienta clave para el Servicio de Calidad que repercutirá directamente en la calidad propiamente dicha de los grados y másteres impartidos en la Universidad de Málaga.

1.2 Estructura de la Memoria

La memoria del proyecto se dividirá en capítulos, para poder entender mejor cada parte del proceso de diseño y desarrollo por separado. Estos capítulos serán expuestos a continuación:

- **Resumen y Palabras clave**

Resumen de la idea general del proyecto, explicando el problema y la solución propuesta. Una lista de palabras clave para categorizar el proyecto.

- **Capítulo 1: Introducción**

Se profundiza en la problemática y en el contexto desde el que parte el proyecto.

- **Objetivos:** descripción de las metas marcadas a alcanzar mediante el desarrollo del proyecto.

- Motivación: razones por las que se ha decidido realizar el proyecto.
- Estructura de la memoria: breve descripción de los apartados que forman la memoria.
- **Capítulo 2: Tecnologías utilizadas**
 - Lenguaje de programación para el *backend*: razones por las que se ha optado por elegir Python como lenguaje para desarrollo.
 - Lenguaje de programación para el *frontend*: razones por las que se ha optado por elegir Html y Javascript como lenguajes para desarrollo
 - Entorno de trabajo: listado de los entornos usados para el desarrollo y el porqué han sido elegidos.
- **Capítulo 3: Extensión de trabajos previos**
 - Explicación de los trabajos que se toman como base y lo que se ha desarrollado a partir de los mismos
- **Capítulo 4: Modificaciones a proyectos**
 - Fuentes de datos: listado y explicación del contenido de los ficheros de fuente de datos para el almacén.
 - Definición ETL
 - Modificaciones ETL: explicación de las modificaciones aplicadas al proyecto de *Integration Services*.
 - Definición Almacén de datos
 - Modificaciones Almacén de datos: explicación de las modificaciones aplicadas al almacén.
- **Capítulo 5: Metodología**
 - Metodología: definición de la metodología usada, Scrum.
- **Capítulo 6: Fases de desarrollo**
 - Planificación: descripción de la planificación seguida durante el desarrollo del proyecto.

- Análisis de requisitos: listado de los requisitos funcionales y no funcionales.
- Diagramas: breve demostración de los diagramas desarrollados para el desarrollo
- Programación e implementación: explicación técnica del desarrollo de las principales funcionalidades de la aplicación:
- **Capítulo 7: Conclusión**
 - Conclusión: consolidación de las ideas desarrolladas durante el proyecto.
- **Capítulo 8: Referencias**
 - Referencias: listado numerado de las referencias de este proyecto.
- **Capítulo 9: Apéndices**
 - Guía de Instalación
 - Manual de usuario

2. Tecnologías utilizadas

2.1 Lenguaje de programación para el *backend*

El desarrollo de aplicaciones web se puede hacer en una infinidad de lenguajes de programación, pero para el desarrollo de este proyecto se ha optado por **Python**, ya que es un lenguaje muy versátil y popular lo que facilita el desarrollo. En concreto, **Flask** ha sido el framework elegido ya que es simple, escalable y fácil de aprender lo que lo hace ideal para este proyecto. Aquí las principales razones desarrolladas por las que se han elegido este lenguaje y este framework:

- **Facilidad de uso:** Python es un lenguaje de programación muy fácil de aprender y utilizar y Flask, por su parte, es un *framework* ligero y flexible que se integra fácilmente con otras bibliotecas de Python.
- **Comunidad de desarrollo activa:** Python y Flask cuentan con una gran comunidad de desarrolladores activa y comprometida. Esto significa que hay una gran cantidad de recursos y herramientas disponibles para los desarrolladores, lo que facilita el proceso de desarrollo y solución de problemas.
- **Eficiencia y velocidad:** Python es un lenguaje de programación muy eficiente y rápido, lo que lo hace ideal para aplicaciones web que requieren un alto rendimiento. Además, Flask es un framework muy ligero que permite una rápida respuesta del servidor y una carga rápida de páginas web.

El uso de Python con Flask no solo permite una fácil integración con otras bibliotecas de Python, sino que también proporciona una amplia gama de funcionalidades adicionales a través de diversas bibliotecas y módulos. Algunas de estas bibliotecas utilizadas en este proyecto son “pandas”[12], una biblioteca de análisis de datos que permite la manipulación y análisis de grandes conjuntos de datos; *os*[13], que proporciona una manera sencilla de interactuar con el sistema operativo subyacente; y *zipfile*, que proporciona una forma fácil de trabajar con archivos ZIP. Estas bibliotecas, junto con la facilidad de uso, la comunidad de desarrollo activa, la eficiencia y velocidad, son algunas de las razones principales por las que se ha optado por Python con Flask para el desarrollo de esta aplicación web.

2.2 Lenguaje de programación para el *frontend*

Para el frontal se ha optado por un estilo simple usando HTML, Javascript y CSS ya que son las tecnologías principales utilizadas para el diseño frontal de aplicaciones web debido a su capacidad para crear interfaces de usuario interactivas y visualmente atractivas.

HTML, que significa "Lenguaje de Marcado de Hipertexto" (Hypertext Markup Language), es el lenguaje utilizado para estructurar el contenido de una página web. Es la base sobre la que se construyen todas las páginas web y proporciona la estructura básica para el contenido de la página. El HTML es fácil de aprender y permite crear páginas web rápidamente. Además, es compatible con la mayoría de los navegadores web, lo que lo hace una opción ideal para el diseño frontal de una aplicación web.

CSS, que significa "Hojas de Estilo en Cascada" (Cascading Style Sheets), es un lenguaje utilizado para definir la presentación y el estilo de una página web. CSS permite a los diseñadores y desarrolladores aplicar estilos visuales a un sitio web, como colores, fuentes, tamaños de letra y diseños de página. También se utiliza para crear efectos de animación y transiciones en la página. CSS se integra perfectamente con HTML y JavaScript, lo que permite crear páginas web visualmente atractivas y estilizadas.

JavaScript, por otro lado, es un lenguaje de programación utilizado para crear páginas web interactivas. Permite a los desarrolladores agregar funcionalidades avanzadas a una página web, como efectos de animación, validaciones de formulario, manipulación de eventos, y muchas otras cosas. Además, se puede utilizar para acceder a datos de servicios web y para interactuar con la API de un sitio web.

La combinación de HTML, CSS y JavaScript permite crear sitios web dinámicos e interactivos con una gran cantidad de funciones y estilos personalizados. Con estas tecnologías, los desarrolladores pueden crear interfaces de usuario atractivas y eficientes que ofrecen una experiencia de usuario excepcional. Además, estas tecnologías se utilizan en una amplia variedad de sitios web, lo que significa que hay una gran cantidad de recursos y documentación disponible en línea para ayudar a los desarrolladores a crear aplicaciones web efectivas y profesionales.



Figura 2.1: Logos tecnologías usadas

2.3 Entorno de trabajo

Podríamos dividir los entornos de trabajo en varias partes:

- Existen muchísimos entornos de desarrollo para crear aplicaciones web como Eclipse, Atom, Sublime Text o Visual Studio. Para el desarrollo de este proyecto se ha elegido **Visual Studio Code**, por su versatilidad, facilidad de uso, compatibilidad con Python, disponibilidad de extensiones y complementos, y una gran comunidad de desarrolladores comprometidos. Todo esto se traduce en una herramienta completa y potente que permite el desarrollo de aplicaciones web de manera eficiente y efectiva.
- El proyecto de *Integration Services* del proyecto anterior que permite la inserción de los datos en el almacén que ha sido creado previamente con una consulta. Para este proyecto se necesita **Visual Studio 2019**, y las modificaciones aplicadas al mismo serán expuestas en los siguientes capítulos.

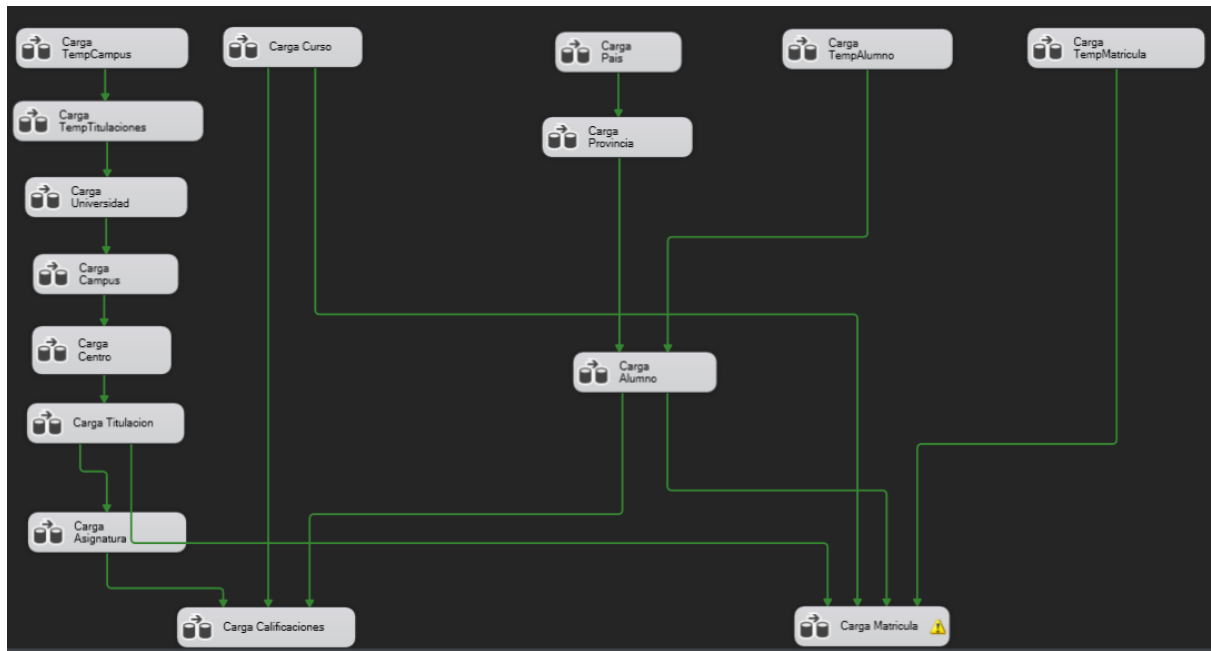


Figura 2.2: Proyecto Integration Services

- *SQL Server Management Studio (SSMS)* es una herramienta de gestión de bases de datos de Microsoft que se ha utilizado en este proyecto. Es una herramienta muy completa y potente que permite administrar bases de datos de SQL Server de manera fácil y eficiente. Con SSMS, se pueden crear, modificar y eliminar bases de datos, así como realizar consultas y generar informes detallados sobre el rendimiento de la base de datos.

Además así mantenemos todo el paquete de entornos de desarrollo de Microsoft, SSMS puede integrarse fácilmente con Visual Studio y otros entornos de desarrollo de Microsoft, lo que permite trabajar de manera más eficiente y productiva.

3. Extensión de trabajos previos

Para la realización de este proyecto se ha partido principalmente de otros dos proyectos:

- Proyecto Almacén y ETL: está desarrollado un proyecto de *integration services* funcional que a partir de los ficheros de entrada alimenta el almacén de datos. El diseño del ETL es bastante complejo ya que contiene muchas reglas específicas para tratar los datos de los ficheros de entrada.
- Proyecto de comparación: programa simple de comparación de ficheros que dado dos ficheros xml, compara sus registros uno a uno y muestra los errores y el porcentaje de similitud de los mismos por consola.

Estos proyectos han sido tomados como base, pero se han modificado según el nuevo diseño para conseguir mejorar su funcionamiento y para poder integrarlos en un sólo documento. Antes del desarrollo de este TFG los nuevos archivos de cada año se debían añadir manualmente para que la ruta coincidiera con el ETL, para después ejecutar este mismo desde *Integration Services* y cargar el Almacén. Esto se ha solucionado con la integración, varias de las modificaciones realizadas al proyecto ETL han sido para mejorar el enrutamiento de los archivos y mejorar la calidad de los datos insertados en el almacén, ya que las reglas que se siguen para insertar los datos no siempre están muy definidas lo que causa con facilidad fallos en las inserciones. El diseño del almacén de datos se ha respetado, aunque sí se ha añadido una nueva tabla pensada para almacenar la comparación de los ficheros de rendimiento.

El diseño del proyecto desarrollado se ha centrado en la integración y mejora de estos proyectos, buscando la simplificación de su uso y haciendo mucho más accesible. Al desarrollar selectores de archivos a través de la web, se puede comprobar el formato de los archivos, asegurándonos de que los archivos de entrada son los adecuados para la ejecución del ETL, y en caso contrario mostrando en qué ha fallado para que el usuario lo conozca. La carga de datos en el almacén se realiza desde la web, ejecutando un comando desde Python.

Anteriormente la creación del fichero de salida se tenía que hacer manualmente, realizando una consulta al almacén de datos y copiando ese resultado a un excel para posteriormente exportarlo como *xml*. La comparación entre este fichero creado y el ideal aportado por el Servicio de Calidad se debía hacer desde el terminar en la máquina virtual.

Todo esto complicaba el uso de esta funcionalidad, por lo que se ha automatizado la creación del documento, y su descarga. La comparación también se ha simplificado e integrado con el resto del proyecto, quedando todo en un mismo entorno simplificando el uso de todo el flujo principal.

En la *Figura 3.1*, se muestra gráficamente cómo los dos proyectos que se han integrado en el proyecto web, con sus respectivas modificaciones. Además este proyecto web desarrollado cuenta con un apartado de visualización de la comparación, permitiendo buscar registros por documento de identidad o filtrar por el número de errores en la similitud, completando una herramienta con mucho potencial para la creación de los ficheros de rendimiento y la mejora de la calidad de los datos del almacén de datos.

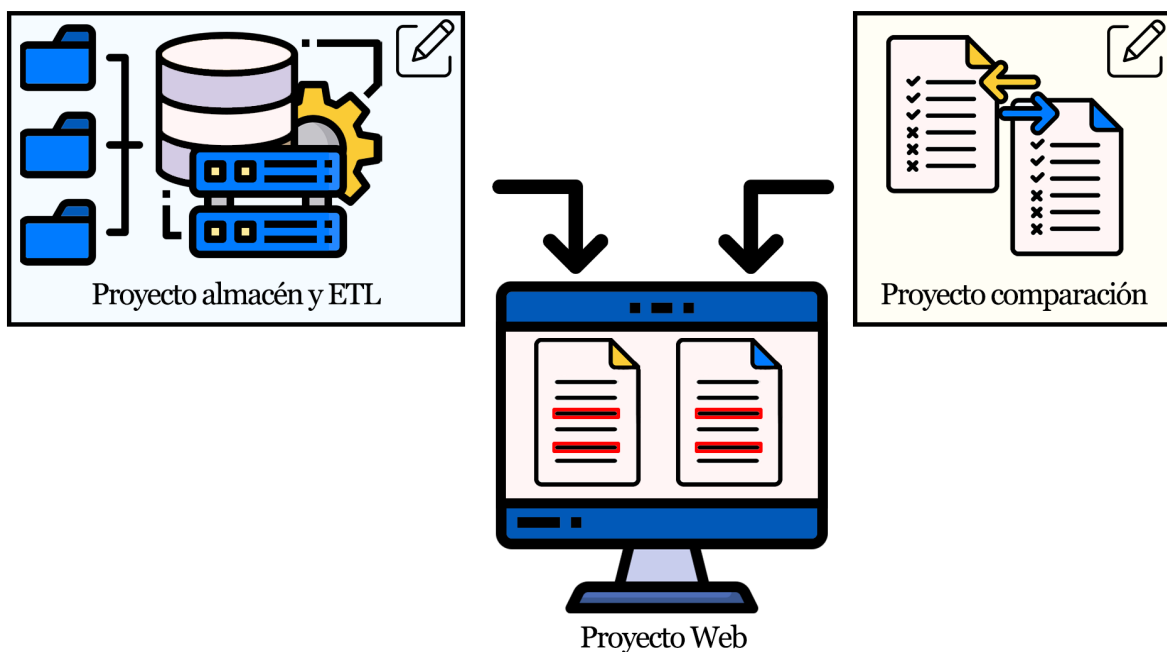


Figura 3.1: Diagrama del proyecto

4. Almacén de datos

4.1 Fuentes de datos

Para poder entender bien qué tenemos que hacer, es muy importante entender bien cuáles son las fuentes de datos y cómo se estructuran. En los anteriores proyectos se centraron en analizar las columnas y el tipo de datos que almacenaban, en este caso nos vamos a centrar en el formato también.

Debemos diferenciar los ficheros que se aportan desde el Servicio de Calidad y los ficheros que nuestro proyecto necesita. Nuestro proyecto necesita los ficheros *csv* en formato *UTF-8-BOM*, en cambio, según los ejemplos que disponemos a partir de 2019 los ficheros que el Servicio de Calidad nos adjunta son *UTF-16*, lo que causa problemas en la conexión de datos en el ETL.

Vamos a diferenciar dos grupos de ficheros, el primer grupo serán los “Generales” que son aquellos que en principio son independientes del curso académico, por lo que son generales para cualquier curso, por lo que deberán quedarse en el proyecto estáticos, aunque podrán sustituirse en el futuro por versiones actualizadas. El segundo grupo serán los “Ficheros de rendimiento”, que serán los que aporte el usuario en la página web. A continuación un listado de los archivos que pertenecen a ambos grupos:

- **Generales:** Estos ficheros no se encuentran accesibles desde la aplicación web ya que no están pensados para ser modificados frecuentemente. Se encuentran en una carpeta del proyecto y en caso de modificación un administrador tendrá que realizarlas, sería tan fácil como sustituir un archivo por el actual.

Nombre	Columnas	Descripción
Titulaciones.csv	<ul style="list-style-type: none">● Tipo Estudio● Titulación● Cód. Titulación● Titulación MEC● Cód. Titulación MEC	Se relacionan las titulaciones con su código y su código MEC.

TipoDNI.xlsx	<ul style="list-style-type: none"> •Codigo_Tipo_Documento •Tipo_Documento •Trabajo_Curso_Anterior 	Identifica los diferentes documentos de identificación con un código.
TipoDedicacionEstudiante.xlsx	<ul style="list-style-type: none"> •Tipo_Estudiante •Cod_Dedicacion 	Relación el tipo de estudiante con un código.
CODIGOS POSTALES SIIU.xlsx	<ul style="list-style-type: none"> •Código postal numero •Código postal texto •Código INE •Nombre 	Relaciona el código postal con su localidad correspondiente.
CargaProvincias.xlsx	<ul style="list-style-type: none"> •Provincia de Residencia Familiar •Cód. País Residencia •Cód. Municipio Residencia •CCAA de Residencia Familiar 	Relaciona la provincia de la residencia familiar con el país, parece exclusivo para extranjeros.
Campus.xlsx	<ul style="list-style-type: none"> •Codigo •Campus •Codigo_Campus_Ficheros 	Relaciona los campus de la UMA con un código.

- **Ficheros de Rendimiento:**

Nombre	Columnas	Descripción
fichero1.csv	<ul style="list-style-type: none"> ● Curso Académico ● Tipo Estudio ● Universidad ● Mes Versión ● Cód. Centro SIIU ● Cód. Municipio ● Cód. Campus ● Cód. Titulación MEC ● Titulación MEC ● Tipo Documento Identificativo ● Documento Identificativo ● Fecha de Nacimiento ● País Nacionalidad ● Sexo ● CCAA de Residencia Familiar ● País Nacimiento ● Cód. Municipio Residencia ● Cód. Postal Actual ● Cód. Dedicación al estudio ● Grado de Presencialidad ● País de Residencia Familiar ● Cód. País Residencia ● Provincia de Residencia Familiar ● Centro 	<p>Dentro de este archivo se encuentra registrada toda la información correspondiente a un estudiante, incluyendo su género, lugar de residencia, fecha de nacimiento y la carrera que está estudiando.</p>
fichero2.csv	<ul style="list-style-type: none"> ● Universidad ● Mes Versión ● Tipo Estudio ● Curso Académico ● Documento Identificativo ● Cód. Titulación MEC ● Núm. Créditos Matriculados ● Núm. Créditos Matriculados 	

	<ul style="list-style-type: none"> • Núm. Créditos 1a matrícula • Núm. Créditos 2a matrícula • Núm. Créditos 3a matrícula • Núm. Créditos 4a matrícula • Núm. Créditos Superados • Núm. Créditos Superados • Núm. Créditos Reconocidos • Núm. Créditos Matriculados Históricos • Núm. Créditos Superados Históricos • Núm. Créditos Presentados históricos (Curso académico) • Núm. Créditos Reconocidos Históricos • Núm. Créditos Transferidos Históricos • Núm. Créditos Presentados (Curso Académico) • Núm. Créditos Convalidados/Adaptados • Núm. Créditos Convalidados/Adaptados Históricos 	<p>En este archivo se encuentran todos los tipos de créditos disponibles según la titulación y el estudiante correspondiente.</p>
<p>fichero3.csv</p>	<ul style="list-style-type: none"> • Mes Versión • Universidad • Mes Versión • Tipo Estudio • Código Interno • Núm. Alumnos Graduados • Nota Media • Curso Académico 	<p>En este archivo, la información más importante es la calificación promedio de cada estudiante en cada programa académico, así como su estado de graduación.</p>

	<ul style="list-style-type: none"> • Documento Identificativo • Cód. Titulación 	
fichero4.csv	<ul style="list-style-type: none"> • Documento Identificativo • Año Acceso SIU (Sistema Universitario Español) 	Relaciona el año de acceso al SUE con el estudiante.
fichero5.csv	<ul style="list-style-type: none"> • Documento Identificativo • Año Acceso Titulación 	Relaciona el año de acceso a la titulación con el estudiante
fichero6.csv	<ul style="list-style-type: none"> • Curso Académico • Tipo Estudio • Documento Identificativo • Tipo de Estudiante 	La información en este archivo ya está disponible en otros, por lo tanto, no es importante.
fichero7.csv	<ul style="list-style-type: none"> • Curso Académico • Tipo Estudio • Documento Identificativo • Estado Matrícula 	Proporciona información sobre el estado de la matrícula de cada estudiante.
Asignaturas.csv	<ul style="list-style-type: none"> • Curso Académico • Tipo Estudio • Documento Identificativo 	Dentro de este archivo, se encuentran registros de

	<ul style="list-style-type: none"> • Cod. Titulación • Cod. Asignatura • Asignatura • Calificación de la Asignatura • Calificación de la convalidación/adaptación 	<p>calificaciones de una materia correspondiente a un estudiante y a un programa de estudios, así como información sobre si la materia ha sido convalidada.</p>
--	--	---

En general cada uno de estos ficheros contiene información relevante de cada estudiante y datos adicionales sobre diferentes aspectos académicos sobre el transcurso del curso, como su resumen créditos matriculados y superados hasta el curso actual. Otra información personal relevante son los códigos postales, tanto de la residencia habitual como la residencia durante el curso. Esta información debe ser de calidad, ya que a partir de ella se alimentará el almacén.

También a raíz del estudio de los ficheros de rendimiento, cabe destacar que en el fichero2.csv hay dos columnas que están duplicadas que son Núm. Créditos Matriculados y Núm. Créditos Superados Históricos, y se debe tener en cuenta para el desarrollo, ya que la inserción de datos tiene en cuenta dicha duplicidad.

4.2 Definición ETL

ETL proviene de *Extraction Transformation and Loading* o traducido Extracción, Transformación y Carga, y es un proceso de la integración de datos en el que se extraen los datos de diversas fuentes, después son transformados según las necesidades y por último son cargados en un destino final que suele ser un almacén de datos o una base de datos [2][10].

La extracción implica la recuperación de datos de diferentes fuentes, incluyendo bases de datos, archivos planos, sistemas de archivos y servicios web. Después de la extracción, los datos son procesados y transformados para limpiarlos, normalizarlos, fusionarlos y enriquecerlos con información adicional. Finalmente, los datos transformados se cargan en el destino final, donde se pueden utilizar para el análisis, la toma de decisiones y otras aplicaciones empresariales.

Los procesos ETL son importantes porque permiten a las organizaciones integrar datos de diversas fuentes y formatos, lo que les permite obtener una vista completa y precisa de su negocio. En nuestro caso permite integrar los datos de todos los años en un almacén, juntando además todos los alumnos de la Universidad de Málaga en uno destinado concretamente a la elaboración de informes requeridos por el Servicio de Calidad, entre ellos el fichero de rendimiento que tratamos de generar automáticamente. Además los procesos ETL permiten automatizar la integración de datos, lo que puede ahorrar tiempo a las organizaciones y reducir costes y errores. Esto es especialmente importante cuando las organizaciones manejan una gran cantidad de datos.

4.3 Modificaciones en el ETL

Al descargar el proyecto de *Integration Services* por primera vez, es posible que se detecte un error inicial relacionado con las rutas de las conexiones a archivos, ya que están escritas manualmente. Esto ocasiona que cada vez que se cambie por ejemplo la ruta de un archivo se tenga que cambiar la ruta de la conexión también, lo que repercute directamente en la instalación del proyecto, ya que todos los archivos se encuentran sin conexión (ver Figura 3.1). Se podrían cambiar uno a uno pero sería un trabajo demasiado tedioso cada vez que se realice un cambio de este tipo, o cada vez que alguien que quiera desarrollar sobre el mismo tenga que instalarlo en un máquina local.

Además los nombres de las conexiones resultaban poco intuitivas, dando lugar a confusiones durante el estudio y desarrollo de las modificaciones, por lo que fueron renombradas por conexión carga “tabla a cargar, fichero de procedencia” .

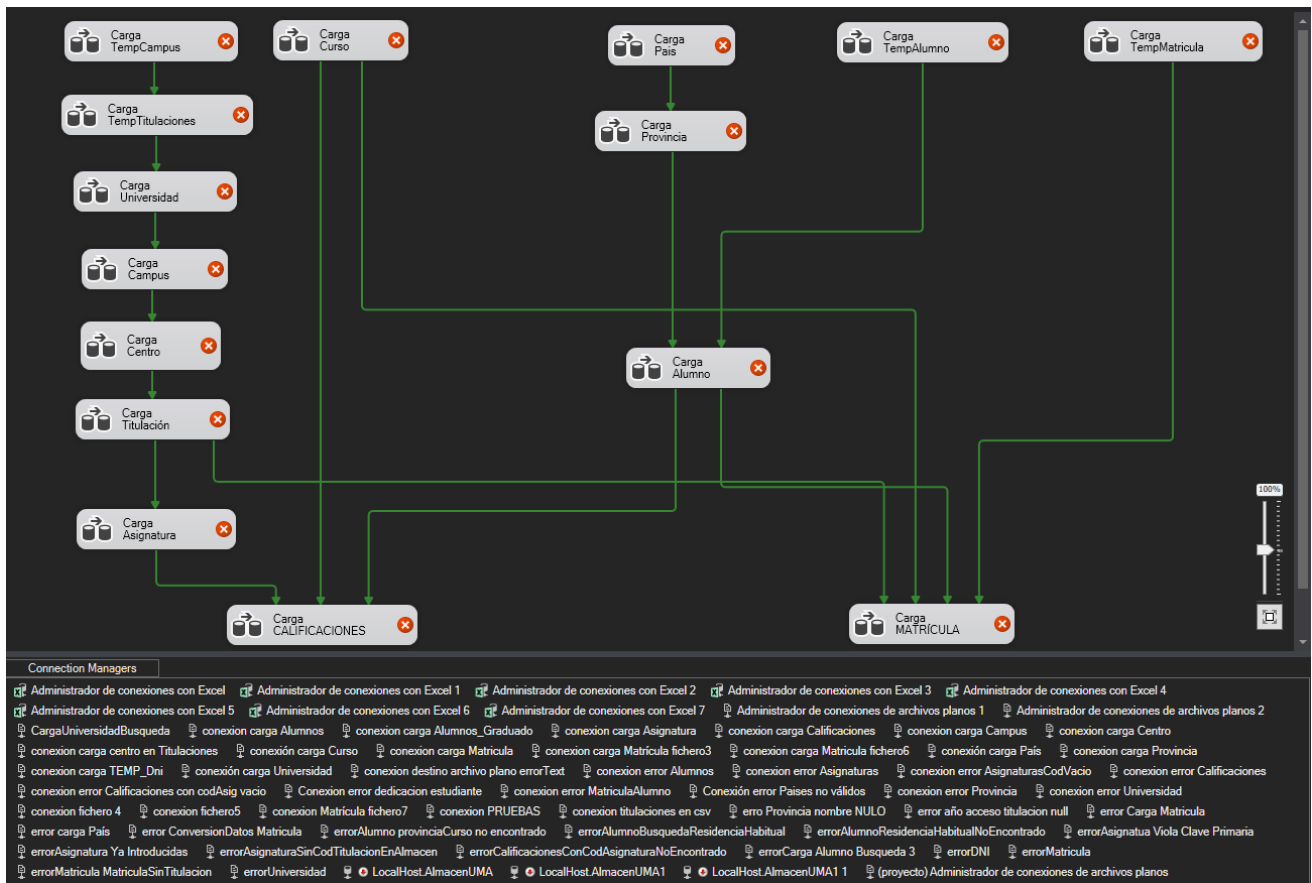


Figura 4.1: Proyecto Integration Services con errores de conexión

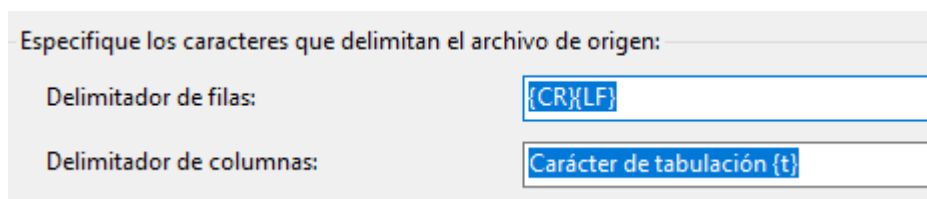
Para poner solución a este problema y poder preparar el proyecto para posibles futuras ampliaciones o modificaciones vamos a parametrizar todas las conexiones posibles. Para esto debemos crear una variable en el proyecto de *Integration Services*, abriendo la ventana de variables de tipo *String*, y añadiendo como valor la ruta en la que se encuentran los archivos (ver Figura 3.2).

Name	Scope	Data type	Value
RutaDatos	Package	String	C:\Users\AleGu\Desktop\TFG - ENTREGA\Datos\Datos_UMA_alumnos\FicherosRendimiento\actual
RutaDatosGrado	Package	String	C:\Users\AleGu\Desktop\TFG - ENTREGA\Datos\Datos_UMA_alumnos\FicherosRendimiento\actual\Grado
RutaErrores	Package	String	C:\Users\AleGu\Desktop\TFG - ENTREGA\Ficheros de error
RutaGenericos	Package	String	C:\Users\AleGu\Desktop\Datos Genericos

Figura 4.2: Ventana con variables del proyecto de Integration Services

Una vez creadas todas las variables necesarias, dos para la carpeta donde se encuentran los ficheros de rendimiento, otras dos para los ficheros de error, podemos ir a la conexión que queremos parametrizar y nos vamos a expresiones, dependiendo de si es una conexión de fichero de texto plano, es decir *csv*, o si es conexión tipo *excel*, seleccionamos *ConnectionString* o *ExelFilePath* respectivamente. En la expresión debemos concatenar la variable que guarda el ruta para ese archivo, más la cadena con el nombre del archivo, debemos de tener precaución ya que deberemos escapar los caracteres especiales, como la “\”. Por ejemplo la expresión para la conexión de datos del archivo Campus.xlsx sería @[User::RutaGenericos]+"\\Campus.xlsx". Cuando guardemos si todo ha salido correctamente saldrá un ícono de “fx” junto a la conexión para indicar que ha sido parametrizada, de esta forma sólo cambiando el valor de la variable podemos enrutar todos los archivos.

La separación de las columnas que nos aporta el Servicio de Calidad en los ficheros “*csv*” es mediante tabulaciones, en el proyecto anterior estos tabuladores eran sustituidos por puntos y comas para poder realizar la conexión de datos. Para simplificar el proyecto hemos cambiando la conexión para que tome por defecto la separación de las columnas con tabulaciones para ahorrarnos el paso de sustituir que en los archivos que se aporta, ya que esta sustitución aunque trivial si es algo costosa a nivel de recursos, teniendo en cuenta las dimensiones de los archivos y la cantidad de sustituciones.



Especifique los caracteres que delimitan el archivo de origen:

Delimitador de filas: {CR}{LF}

Delimitador de columnas: Carácter de tabulación {t}

Figura 4.3: Configuración de Columnas de las conexiones de ficheros planos

Uno de los problemas encontrados en la carga de datos son las notas medias, pues en los casos en los que el alumno tiene nota media o viene el valor sin comillas y en otro caso viene entre comilla, esto se debe a un pequeño problema con el formato de esta columna, y se puede solucionar fácilmente utilizando columna derivada y la expresión *REPLACE([Nota Media],"\",",")*.

4.3 Definición Almacén de datos

Un almacén de datos o *data warehouse* es un sistema utilizado para almacenar grandes cantidades de datos de diferentes fuentes con el objetivo de facilitar su análisis y toma de decisiones. Estos datos son organizados, limpiados y transformados en un formato coherente y homogéneo para poder ser utilizados por los sistemas de análisis de datos, como las herramientas de Business Intelligence (BI), los sistemas de minería de datos y los modelos de aprendizaje automático[11].

La importancia de los almacenes de datos radica en que permiten a las organizaciones obtener una visión más completa y precisa de sus operaciones y de sus clientes, lo que puede ayudarles a tomar decisiones más informadas y estratégicas. Además, el uso de los almacenes de datos puede mejorar la eficiencia y la efectividad de las operaciones y reducir los costos.

En el mundo de la informática, los almacenes de datos son cada vez más importantes debido al gran volumen de datos que se generan y se almacenan en diferentes formatos y sistemas. Por esta razón, las empresas y organizaciones necesitan herramientas y tecnologías que les permitan integrar, limpiar y analizar estos datos de manera eficiente y efectiva. Los almacenes de datos son una solución para este problema, ya que proporcionan un ambiente centralizado para la integración y el análisis de datos.

Es necesario realizar una distinción previa entre hechos y dimensiones en el contexto del almacén de datos. La tabla de hechos es la tabla principal del almacén y contiene campos clave que se utilizan para relacionarse con las demás tablas que componen las dimensiones. En las tablas de hechos se almacenan las métricas o medidas que deseamos analizar. Estas medidas suelen ser valores numéricos que se pueden agregar, como calificaciones de asignaturas, notas numéricas (notas medias) y números de créditos por matrícula. En nuestro caso las tablas de hechos son

Por otro lado, las tablas de dimensiones contienen atributos o características que nos permiten agrupar, rebanar o filtrar la información. Estos atributos pueden estar organizados en jerarquías, lo que nos facilita realizar análisis de datos a nivel de agrupaciones. Las relaciones entre dimensiones y hechos suelen ser de tipo uno a muchos (1:N).

Un ejemplo de jerarquía en una dimensión podría ser la dimensión de fecha, donde los atributos de año, mes y día forman una jerarquía. En la dimensión de producto, por otro lado, podemos encontrar una jerarquía formada por los atributos de categoría, subcategoría y producto.

En nuestro caso particular, contamos con dos tablas de hechos: Matrícula y Calificaciones. Estas tablas de hechos están relacionadas con dos jerarquías compartidas, que incluyen las dimensiones de Curso, para una jerarquía, y País, Provincia y Alumno, para otra.

Además, la tabla de hechos Matrícula tiene una jerarquía adicional que incluye las dimensiones de Universidad, Campus, Centro y Titulación. Por otro lado, la tabla de hechos Calificaciones cuenta con otra jerarquía que incluye las dimensiones de Universidad, Campus, Centro, Titulación y Asignatura.

Estas jerarquías nos permiten analizar los datos de forma agrupada y realizar diferentes niveles de análisis basados en las dimensiones mencionadas.

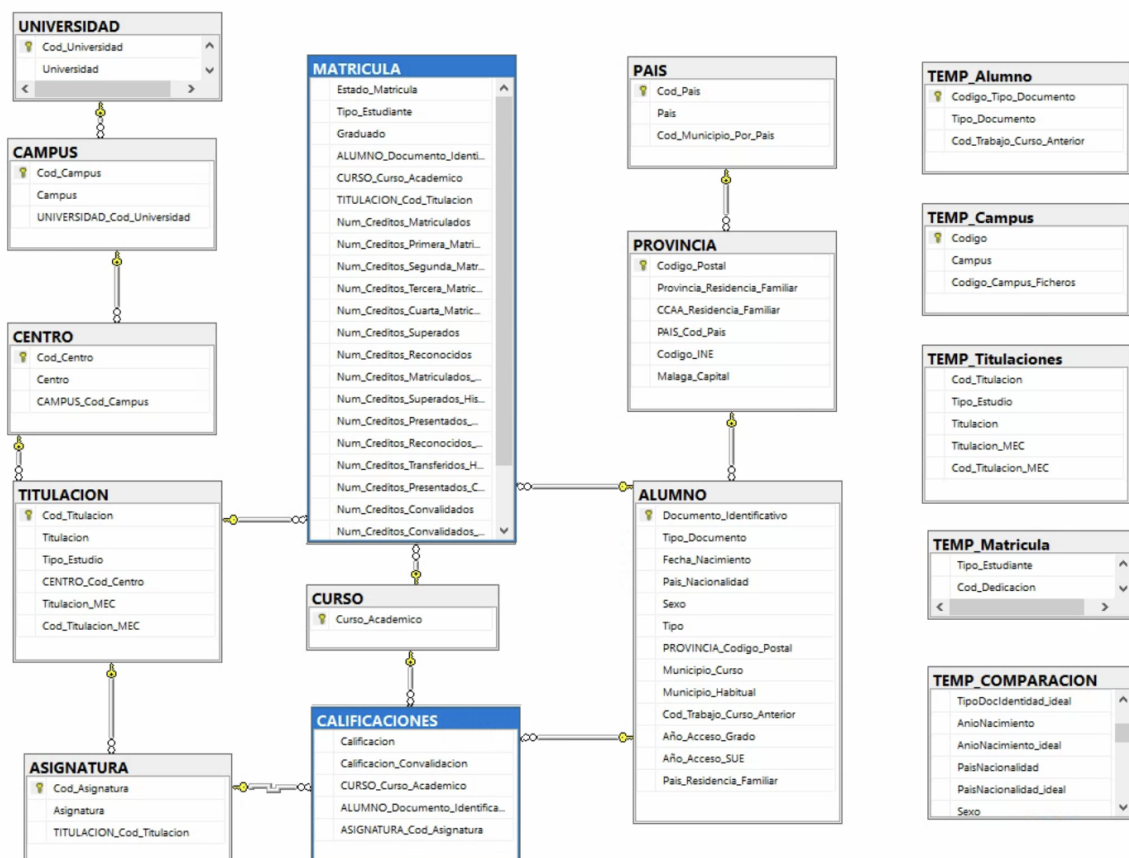


Figura 4.4: Modelo Conceptual del Almacén

Entre las dos tablas de hechos que se han declarado, para nuestro proyecto destaca “MATRICULA”, ya que, a partir de esta tabla se genera el documento para el Servicio de Calidad. Esta tabla contiene entre otros muchos datos toda la información relacionada con los créditos del alumno, como los que se ha matriculado en un curso concreto, cuántos ha recuperado, cuántos son de primera matrícula, de segunda, etc. Cabe destacar entre las tablas temporales la tabla “TEMP_COMPARACION”, aunque no se relaciona conceptualmente con ninguna tabla de hechos, que se usará para la comparación de ficheros y almacenará por cada registro todos los campos del mismo de los dos ficheros comparados. Por último las dimensiones siempre van de lo más específico a lo más general, se puede ver fácilmente en la dimensión Alumno, Titulación, Centro, Campus y Universidad. Esta relación siempre es de uno a muchos, ya que una titulación puede tener muchos alumnos, y un centro puede tener muchas titulaciones, y así sucesivamente.

4.3 Modificaciones al Almacén de datos

Se han hecho dos modificaciones con respecto al almacén de datos:

- Se ha añadido la tabla TEMP_COMPARACION, pensada para la comparación de los ficheros de salida que genera la aplicación web con los ficheros que el Servicio de Calidad ya ha generado, los “ideales”. Para guardar los datos de los registros comparados, esta tabla contiene todos los campos de los registros dos veces, uno con el nombre donde se guardará el dato del fichero generado y otro con el mismo nombre con el añadido de “_ideal” en el que se guardará los campos del fichero ideal. Además tiene una columna con un id autogenerado y otra con el número de errores en esa comparación de registro. Esto simplificará las consultas a dicha tabla.

- Se ha modificado la consulta para obtener los resultados con los que se generará el fichero de salida posteriormente. Esta consulta devuelve todos los campos con los que se formarán los registros por cada una de las filas que estén en la tabla Matrícula con el curso que se seleccione. Se han solucionado unos errores que había con la selección de campos, por ejemplo había una confusión con el campo “Num_Creditos_Reconocidos_Historico”. También se ha modificado la consulta para obtener los datos en el mismo formato que requiere el XML, por ejemplo los créditos deben estar en este formato: '000.00'. La consulta al completo se puede consultar en la carpeta del proyecto (QuerySalida.sql).

```

FORMAT(MAT.Num_Creditos_Matriculados, '000.00') AS CreditosMat,
FORMAT(MAT.Num_Creditos_Primer_Matricula, '000.00') AS CreditosMat1,
FORMAT(MAT.Num_Creditos_Segunda_Matricula, '000.00') AS CreditosMat2,
FORMAT(MAT.Num_Creditos_Tercera_Matricula, '000.00') AS CreditosMat3,
FORMAT(MAT.Num_Creditos_Superados, '000.00') AS CreditosSupCurso,
FORMAT((MAT.Num_Creditos_Matriculados - MAT.Num_Creditos_Superados), '000.00') AS CreditosNoSupCurso,
FORMAT(MAT.Num_Creditos_Reconocidos, '000.00') AS CreditosRecCurso,
FORMAT(MAT.Num_Creditos_Presentados_Curso_Academico, '000.00') AS CreditosPreCurso,
FORMAT((MAT.Num_Creditos_Matriculados - Num_Creditos_Presentados_Curso_Academico), '000.00') AS CreditosNoPreCurso,
FORMAT(MAT.Num_Creditos_Matriculados_Historico, '000.00') AS CreditosMatInicio,
FORMAT(MAT.Num_Creditos_Superados_Historico, '000.00') AS CreditosSupInicio, /* ERA MAT.Num_Creditos_Convalidados_Historico */
FORMAT(MAT.Num_Creditos_Presentados_Historico, '000.00') AS CreditosPreInicio,
FORMAT(MAT.Num_Creditos_Reconocidos_Historico, '000.00') AS CreditosRecInicio,
FORMAT(MAT.Num_Creditos_Transferidos_Historico, '000.00') AS TotalCreditosTransferidos,
MAT.Graduado AS TituladoCursoRef,
FORMAT(ROUND(MAT.Nota_Media_Fichero_Rendimiento, 2), '00.00') AS NotaMediaExpediente,
1 AS PresencialidadEstudio

```

Figura 4.4: Query salida

Otro problema importante ocurría con los alumnos matriculados en dobles grados, ya que sólo se guardaba la matrícula con el código de titulación del doble grado, pero un requisito del fichero de salida para los alumnos de dobles grados es que deben aparecer 3 registros, uno con el código de titulación del doble grado y otros dos con los códigos de titulación que forman el doble grado. Para resolver este problema se ha optado por desarrollar un disparador que se active cuando se inserta una fila en la tabla “MATRICULA”, y compruebe si el código de titulación corresponde a un doble grado, y en ese caso busca los grados que lo componen e inserta dos filas en “MATRICULA” con los códigos correspondientes. Para que el disparador actúe cuando se insertan los datos a partir del ETL es importante desactivar la opción de “Usar inserción masiva cuando sea posible”.

```

BEGIN
-- Obtener el valor de Cod_Titulacion desde la tabla TITULACION
DECLARE @Cod_Titulacion VARCHAR(50);
SELECT @Cod_Titulacion = Cod_Titulacion
FROM TITULACION
WHERE Cod_Titulacion = (SELECT i.TITULACION_Cod_Titulacion FROM inserted i);

-- Obtener los valores para TITULACION_Cod_Titulacion a partir de la consulta
DECLARE @Cod_Titulacion1 VARCHAR(50), @Cod_Titulacion2 VARCHAR(50);
SELECT @Cod_Titulacion1 = Cod_Titulacion
FROM TITULACION
WHERE Cod_Titulacion IN (
    SELECT Cod_Titulacion
    FROM TITULACION
    WHERE Titulacion LIKE '%' + (
        SELECT SUBSTRING(Titulacion, 1, CHARINDEX(' + ', Titulacion + ' + ') - 1)
        FROM TITULACION
        WHERE Cod_Titulacion = @Cod_Titulacion
    ) + '%'
    AND Cod_Titulacion <> @Cod_Titulacion
    AND Titulacion NOT LIKE '% + %'
);

SELECT @Cod_Titulacion2 = Cod_Titulacion
FROM TITULACION
WHERE Cod_Titulacion IN (
    SELECT Cod_Titulacion
    FROM TITULACION
    WHERE Titulacion LIKE '%' + (
        SELECT SUBSTRING(Titulacion, CHARINDEX(' + ', Titulacion) + 3, LEN(Titulacion))
        FROM TITULACION
        WHERE Cod_Titulacion = @Cod_Titulacion
    ) + '%'
    AND Cod_Titulacion <> @Cod_Titulacion
    AND Titulacion NOT LIKE '% + %'
);

-- Insertar la fila duplicada en MATRICULA con el campo TITULACION_Cod_Titulacion cambiado a @Cod_Titulacion1

```

Figura 4.5: Fragmento del disparador

5. Metodología

Para el desarrollo del proyecto se ha optado por la metodología **Scrum**, una metodología ágil de gestión de proyectos, que se utiliza para el desarrollo de software y otros productos complejos. Esta metodología se basa en la iteración y la entrega continua de valor, a través de ciclos de trabajo cortos y enfocados en objetivos concretos. Scrum se centra en la colaboración, la comunicación y la transparencia, para asegurar que el equipo de desarrollo trabaje de manera eficiente y efectiva. Aunque normalmente se usa con equipos de trabajo se puede orientar fácilmente a trabajos individuales como se ha hecho en este desarrollo.

Scrum se compone de tres pilares principales: transparencia, inspección y adaptación [3]. La transparencia se refiere a la visibilidad de todo el proceso de desarrollo para todos los miembros del equipo y los stakeholders. La inspección se refiere a la revisión constante del trabajo realizado y del proceso de desarrollo en sí mismo, para identificar oportunidades de mejora. La adaptación se refiere a la capacidad del equipo de ajustar el proceso de desarrollo para abordar las oportunidades de mejora identificadas durante la inspección.

Scrum se divide en tres roles principales:

- El **Scrum Master** es el líder del proceso Scrum y se encarga de guiar al equipo de desarrollo en el uso efectivo de Scrum. Es responsable de asegurarse de que se sigan las reglas y prácticas de Scrum, y de identificar y eliminar cualquier obstáculo o impedimento que pueda afectar el progreso del equipo. El Scrum Master trabaja para garantizar que el equipo de desarrollo esté enfocado en alcanzar los objetivos del Sprint y ayudarles a mejorar continuamente su rendimiento.
- El **Product Owner** es el responsable de definir y priorizar los requisitos del producto, y de asegurarse de que el equipo de desarrollo esté enfocado en la entrega de valor para el cliente. Es el único responsable de la gestión del Product Backlog y trabaja con el equipo de desarrollo para asegurarse de que los requisitos del producto se traduzcan en funcionalidades que proporcionen valor al cliente. El Product Owner es también el principal punto de contacto con el cliente y se encarga de asegurar que el producto se entregue de manera oportuna y con los criterios de calidad adecuados.

- El **Equipo de Desarrollo** es el grupo de personas responsables de diseñar, desarrollar, probar y entregar el producto. El equipo de desarrollo debe ser multifuncional, auto-organizado y tener las habilidades necesarias para completar el trabajo requerido durante el Sprint. El equipo de desarrollo es responsable de estimar el esfuerzo requerido para completar las tareas asignadas, y de trabajar juntos para lograr los objetivos del Sprint. Además, el equipo de desarrollo también es responsable de crear las pruebas necesarias y realizar la verificación y validación de los resultados de su trabajo.

En el caso de este proyecto, los roles de Scrum Master y Product Owner han sido llevados a cabo por el tutor y el rol de equipo de desarrollo por el realizador del proyecto.

El proceso de SCRUM se compone de cinco fases, cada una de las cuales cumple un propósito específico. En la primera fase, Inicio, se realiza un estudio y análisis del proyecto para identificar las necesidades del sprint y definir las tareas a realizar. La fase de Planificación y Estimación se enfoca en establecer metas fijas y plazos para cumplir con las tareas identificadas. La tercera fase, Implementación, tiene como objetivo producir entregables a partir del Sprint Backlog y realizar revisiones diarias para verificar el avance del equipo. En la cuarta fase, Revisión y Retrospectiva, se realiza una revisión del entregable y se busca mejorar la eficiencia del equipo y el proceso para próximos sprints. Por último, en la fase de Lanzamiento se entregan los resultados finales al cliente y se realiza una retrospectiva del proyecto para inspeccionar todo lo ocurrido y encontrar áreas de mejora. Cada fase es fundamental para el éxito del proyecto y permite una gestión efectiva de los recursos y del tiempo.

Scrum utiliza un ciclo de trabajo llamado **Sprint**, que dura entre 1 y 4 semanas. Durante el Sprint, el Equipo de Desarrollo trabaja en los elementos del Sprint Backlog, que son las tareas y actividades específicas que deben ser completadas durante el Sprint para alcanzar los objetivos definidos. El Sprint Backlog se crea en la reunión de planificación del Sprint, donde el equipo de desarrollo define qué trabajo se va a completar durante el Sprint, cómo se va a hacer y cómo se va a medir.

El objetivo del Sprint es entregar un incremento potencialmente entregable al final del Sprint. Esto significa que el equipo de desarrollo trabaja en todas las tareas necesarias para crear un producto funcional que cumpla con los requisitos del cliente y esté listo para ser entregado a los usuarios finales. Al final del Sprint, el Incremento debe ser probado y validado para asegurarse de que cumple con los criterios de aceptación del cliente. Después de cada Sprint, se llevan a cabo dos reuniones importantes: la Revisión del Sprint y la Retrospectiva del Sprint. En la Revisión del Sprint, el equipo de desarrollo presenta el Incremento potencialmente entregable al Product Owner y a otros interesados, para que lo inspeccionen y proporcionen retroalimentación. En la Retrospectiva del Sprint, el equipo de desarrollo reflexiona sobre el proceso de desarrollo del Sprint anterior y cómo pueden mejorar en el futuro. En ambas reuniones se identifican oportunidades de mejora y se ajusta el proceso de desarrollo para el próximo Sprint.

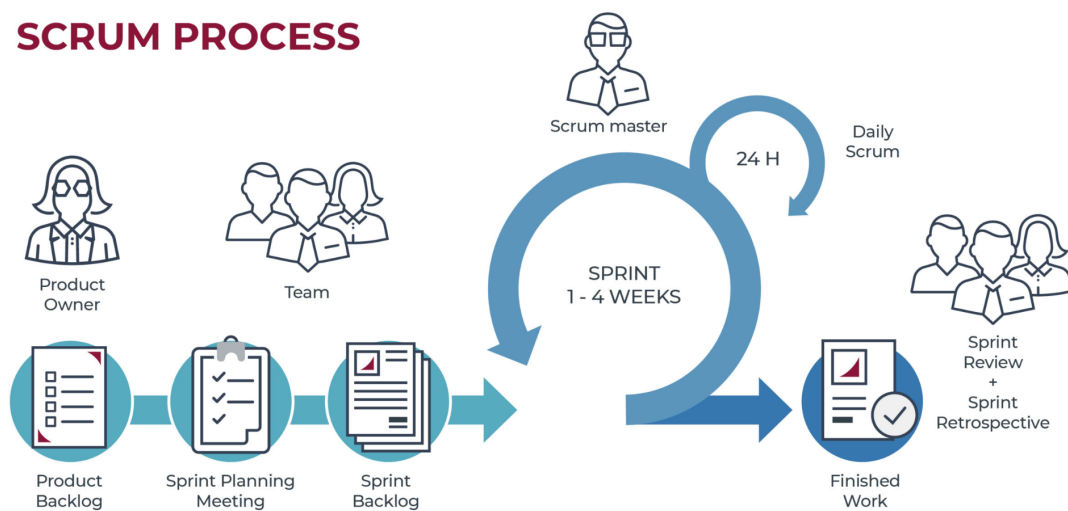


Figura 5.1: Gráfico metodología Scrum

[Imagen]. Recuperado el 15 de mayo de 2022, de <https://www.escueladenegociosydireccion.com>

El objetivo de la repetición de estos ciclos de trabajo es el de crear un producto que cumpla con los requisitos del cliente, que se entregue en el tiempo y presupuesto establecidos y que se desarrolle de forma efectiva y eficiente. Scrum promueve la transparencia, la inspección y la adaptación, y está diseñado para ayudar a los equipos de desarrollo a trabajar de manera más colaborativa y productiva.

La elección de SCRUM como metodología para el proyecto se debe a varios factores clave. En primer lugar, el enfoque iterativo de SCRUM permite una adaptación constante a los cambios y una entrega temprana y frecuente de los resultados. Asimismo, SCRUM establece un marco de trabajo claro y flexible, lo que permite una mayor transparencia y una gestión efectiva del tiempo y de los recursos. Por otro lado, SCRUM se ha consolidado como una metodología eficaz para el desarrollo de proyectos de software, lo que la convierte en una opción segura y confiable para el proyecto.

En resumen, la elección de SCRUM como metodología para el proyecto se basa en su enfoque colaborativo, su adaptabilidad, su marco de trabajo claro y flexible, su eficacia en el desarrollo de proyectos de software y su amplia aceptación en la industria.

6. Fases de desarrollo

6.1 Planificación

A continuación se expondrá una tabla con un breve seguimiento de las fases del desarrollo, incluyendo las fechas, y destacando las reuniones con los tutores en las que se definen o redefinen los requisitos:

Fecha de inicio	Fecha de fin	Fases del proyecto	Tareas principales
1-feb	19-feb	Inicio	Primeras reuniones online sobre el proyecto, definición de los requisitos principales, estudio del almacén de datos, ETL y el proyecto en general
20-feb	20-feb	Reunión	Reunión definición primer Sprint Backlog, requisitos, tareas, etc.
21-feb	6-mar	Planificación y Estimación	Identificación y estimación de tareas, creación del Sprint Backlog, modificaciones sobre el ETL y Almacén de datos, comienzo desarrollo web
7-mar	20-mar	Implementación	Producción del entregable a partir del Sprint Backlog
21-mar	21-mar	Revisión y Retrospectiva	Revisión del entregable, medición y mejora de la eficiencia del equipo
22-mar	12-abr	Implementación	Producción del entregable a partir del Sprint Backlog (<i>backend</i>)
13-abr	13-abr	Reunión, Revisión y Retrospectiva	Revisión del entregable, medición y redefinición de partes del <i>backend</i> y <i>frontend</i>
13-abr	26-abr	Implementación	Producción del entregable a partir del Sprint Backlog (<i>backend</i> y front-end)
27-abr	27-abr	Reunión, Revisión y Retrospectiva	Revisión del entregable, medición y redefinición de partes del backend, visualización datos finales

28-abr	7-may	Implementación	Producción del entregable a partir del Sprint Backlog (backend y front-end)
8-may	18-may	Revisión y Retrospectiva Final	Pruebas, revisión del entregable y definición de requisitos finales para la entrega

6.2 Análisis de requisitos

El éxito de un proyecto de software depende en gran medida de la definición clara de sus requisitos. Para ello, es necesario identificar las necesidades del cliente y plasmarlas en un conjunto de especificaciones técnicas que permitan al equipo de desarrollo construir una solución que satisfaga dichas necesidades. En este proyecto, se han establecido una serie de requisitos funcionales y no funcionales que guiarán el desarrollo del software. A continuación, se detallarán cada uno de ellos para que se puedan entender claramente las funcionalidades que se deben implementar y las restricciones que deben cumplirse.

Para este proyecto, la lista de requisitos es:

ID	Nombre	Descripción
Requisitos funcionales		
RF 1	Inicio de sesión	Se debe poder iniciar sesión con el usuario y contraseña de Windows del servidor que aloja la aplicación
RF 2	Cerrar sesión	Se debe poder cerrar sesión
RF 3	Cargar ficheros de forma individual	Permitir al usuario subir un fichero a la aplicación web para su posterior carga en el almacén de datos

RF 4	Cargar ficheros en formato ZIP	Permitir al usuario subir un fichero en formato ZIP que contenga varios archivos para su posterior carga en el almacén de datos
RF 5	Comprobar el formato de los ficheros	Validar que los ficheros subidos por el usuario tienen un formato compatible con el almacén de datos
RF 6	Informar sobre el formato de los ficheros	Proporcionar información al usuario sobre el formato de los ficheros subidos y si son compatibles con el almacén de datos
RF 7	Cargar Datos	Automatizar la ejecución del proceso de Extracción, Transformación y Carga (ETL) de los archivos subidos al almacén de datos
RF 8	Generar Fichero Salida	Se debe poder generar el fichero de salida del año que seleccione el usuario
RF 9	Descargar Fichero Salida	Se debe poder descargar el fichero de salida que se ha generado
RF 10	Comparar ficheros	Se debe poder comparar dos ficheros con registros, uno ideal y otro generado
RF 11	Visualización Similitud	Se debe mostrar el porcentaje de similitud de los ficheros comparados
RF 12	Visualización Errores	Se debe mostrar el porcentaje de los errores de similitud de los ficheros
RF 13	Visualización de Registros con errores	Se debe poder mostrar aquellos registros que contengan errores de similitud
RF 14	Búsqueda de registros con errores	Se debe poder buscar por Documento de Identidad los registros con errores de similitud
Requisitos no funcionales		
RNF 1	Web	Debe desarrollarse una aplicación web para que sea fácilmente accesible desde la red de la Universidad de Málaga
RNF 2	Interfaz de usuario	La interfaz de usuario debe ser muy clara e intuitiva, para usuarios que no tenga conocimientos avanzados de informática
RNF 3	Manual de usuario	Se debe entregar un manual de usuario con la aplicación

RNF	Guía de instalación	Se debe entregar una guía de instalación con la aplicación
-----	---------------------	--

6.3 Diagramas

6.3.1 Casos de Uso

El diagrama de casos de uso es muy simple, ya que sólo disponemos de un actor llamado usuario, que además es capaz de realizar todas las funcionalidades de la aplicación, ya que la propia aplicación está diseñada para ser instalada en un máquina virtual con un acceso muy restringido, por lo que no es necesario la implementación de diferentes roles de usuarios y diferentes funcionalidades dependiendo de estos mismos.

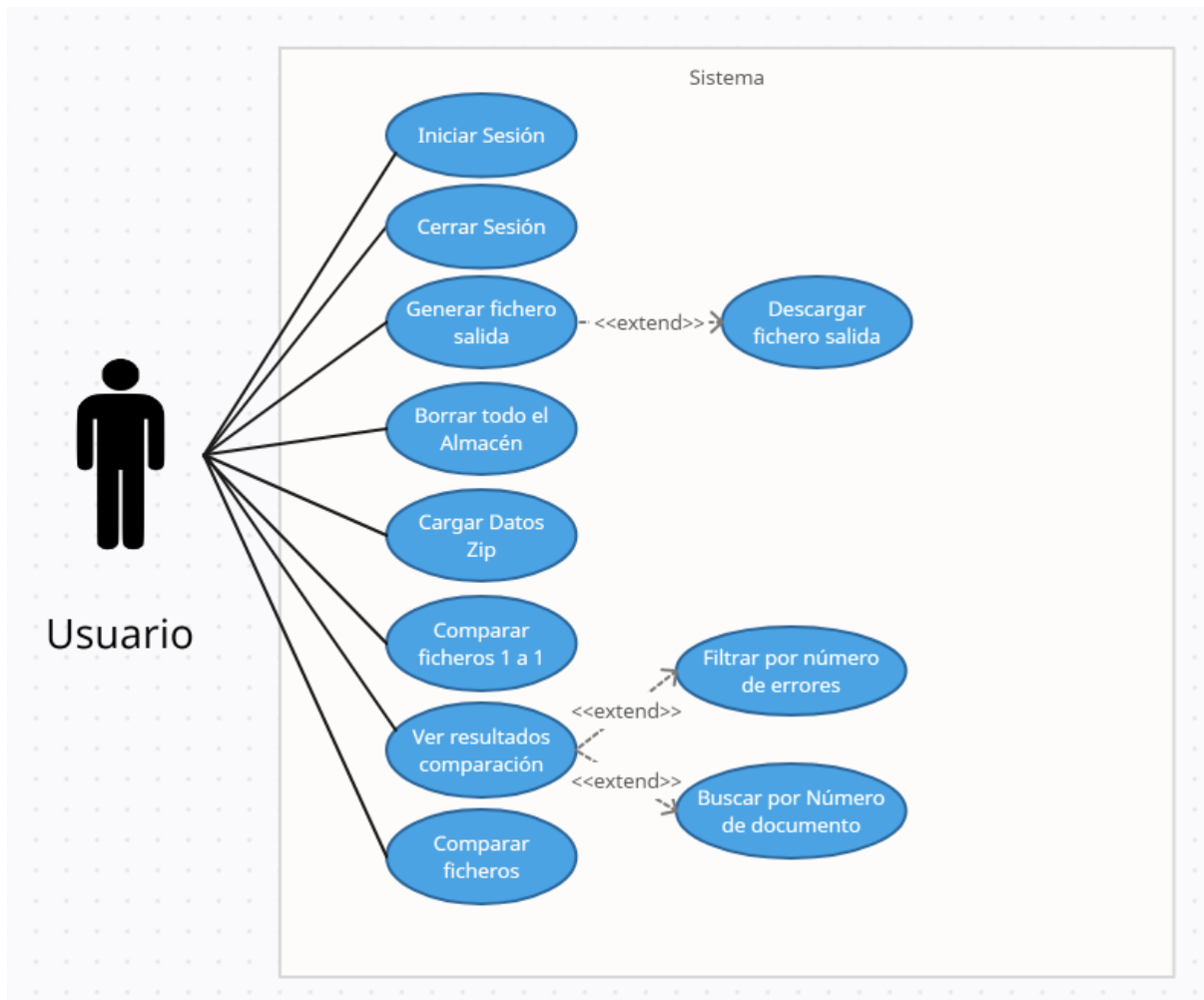


Figura 6.1: Diagrama Casos de uso

6.3.2 Diagrama de secuencia

La creación de diagramas de secuencia es una práctica fundamental en el diseño y desarrollo de sistemas de software. Estos diagramas permiten capturar y comunicar de manera efectiva la lógica subyacente de un sistema, mostrando la secuencia de eventos y las interacciones entre los diferentes elementos. Al ofrecer una representación gráfica de alto nivel, los diagramas de secuencia ayudan a identificar los puntos clave de interacción, los roles de los actores involucrados y las dependencias entre los componentes.

La importancia de desarrollar diagramas de secuencia radica en varios aspectos. En primer lugar, proporcionan una comprensión clara de cómo se realiza un determinado proceso en el sistema, permitiendo identificar posibles problemas o mejoras en el diseño. Al visualizar la secuencia de acciones, se pueden detectar fácilmente puntos de congestión, cuellos de botella o pasos redundantes, lo que facilita la optimización y la toma de decisiones informadas.

Además, los diagramas de secuencia ayudan a comunicar y alinear a los miembros del equipo de desarrollo. Proporcionan un lenguaje común para discutir y visualizar los flujos de trabajo, facilitando la colaboración y evitando malentendidos. Los diagramas de secuencia también sirven como documentación valiosa para futuras referencias y para la comprensión del sistema por parte de nuevos miembros del equipo.

En la *Figura 6.2* se puede ver un ejemplo de un diagrama de secuencia, en este caso representado los casos de uso de Generar fichero de salida y la Descarga del fichero de salida, que depende del primero. Ya que sólo podemos descargar el fichero de salida que acabamos de descargar.

En la *Figura 6.3* se puede ver un ejemplo de un diagrama de secuencia, en este caso representado los casos de uso de Cargar Datos Zip con las dos alternativas, en una se le devuelve el error con el mensaje a el usuario en la propia vista, y si los archivos han sido verificados correctamente se le redirige a la nueva vista para ejecutar el comando del ETL.

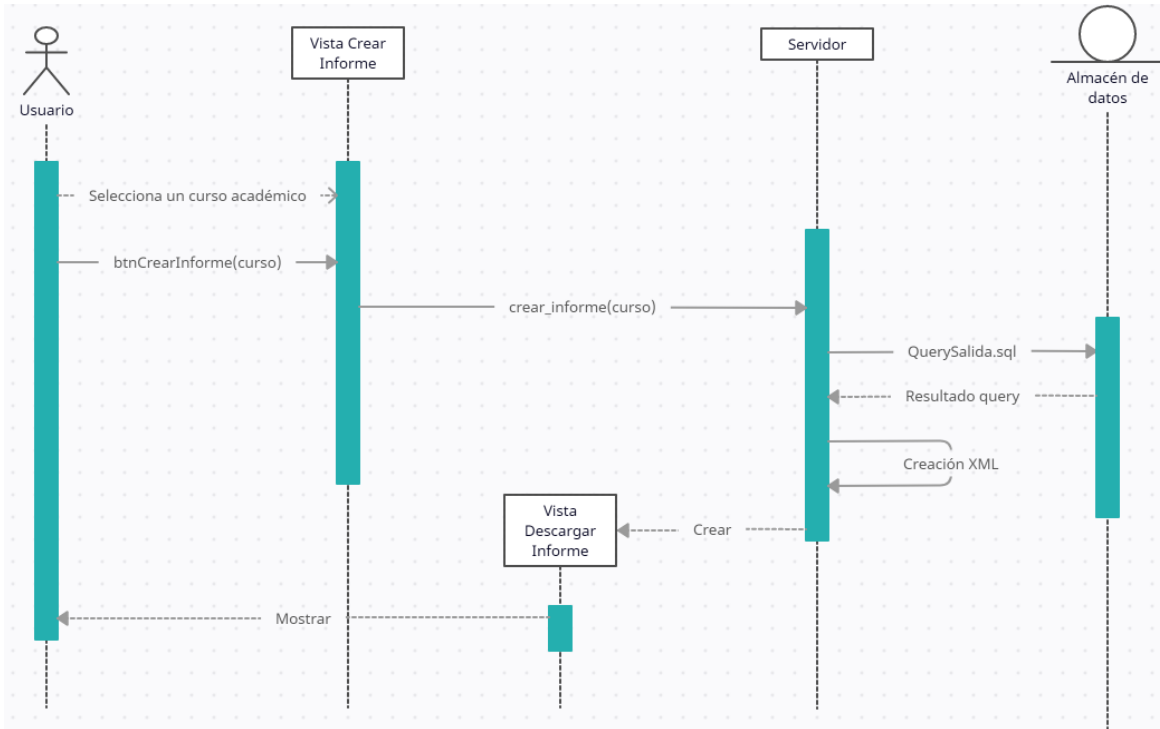


Figura 6.2: Diagrama Secuencia Generar y Descargar Fichero

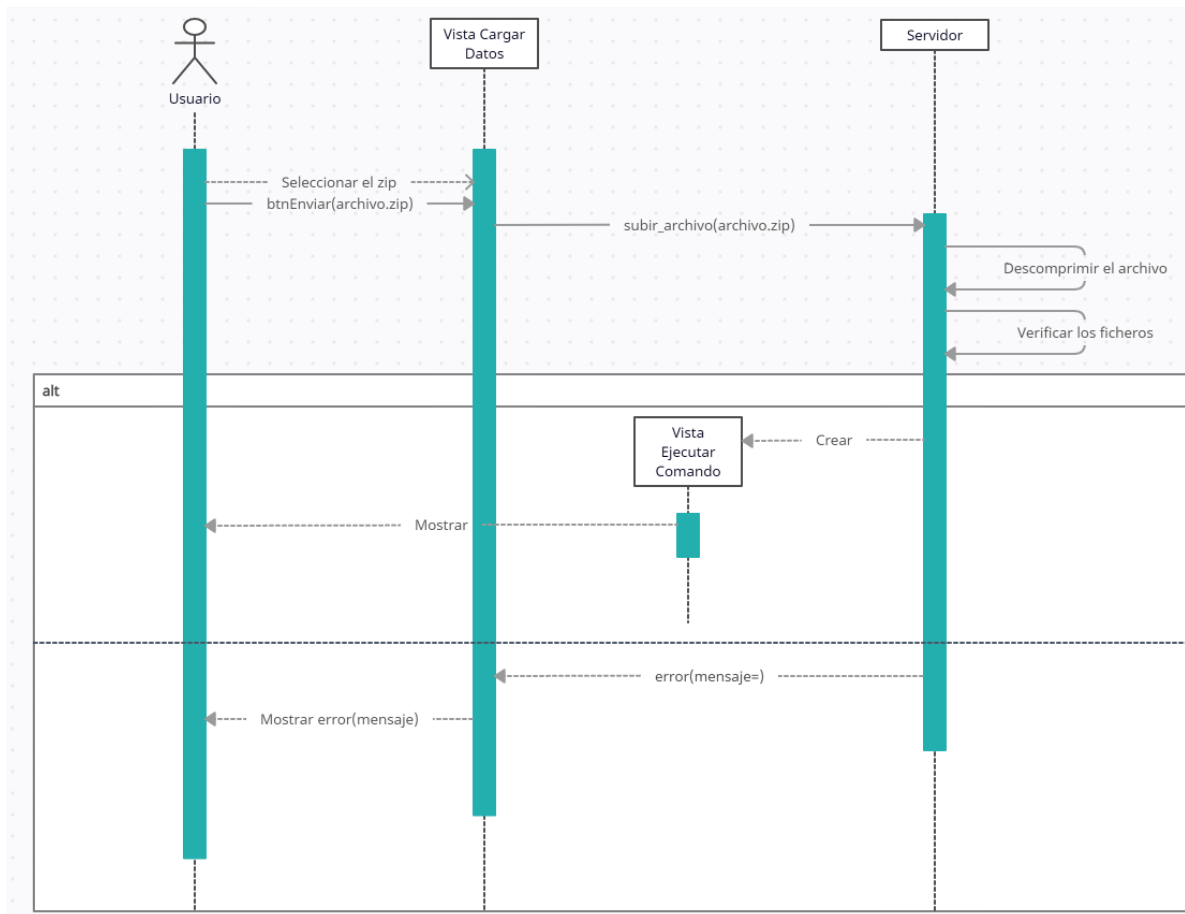


Figura 6.3: Diagrama Secuencia Cargar Datos zip

6.4 Programación e Implementación

6.4.1 Inicio Sesión

Para el inicio de sesión se requería que fuera con las credenciales de Windows de la máquina que aloja el proyecto, es decir el servidor. Esto con el objetivo de poder restringir aún más su accesibilidad, no dejando crear usuarios sólo dando acceso a aquellos usuarios que tengan acceso a la máquina virtual que funciona como servidor. Ya que esta aplicación está pensada para ser usada por muy pocos usuarios es una solución bastante simple y óptima para la problemática de autenticación.

- **Backend:**

Para desarrollarlo se han usado dos paquetes para python “win32api” y “win32security”[14], que permiten interactuar con las credenciales de Windows y comprobar la autenticación del usuario en el sistema. Se ha creado una función de inicio de sesión que utiliza estas bibliotecas para comprobar las credenciales de usuario y permitir el acceso a la aplicación [Figura 6.4]. También se ha implementado un sistema de registro de actividad de inicio de sesión para fines de auditoría y seguridad.

Es importante destacar que esta solución de autenticación basada en las credenciales de Windows es efectiva para una aplicación que se utiliza en una máquina virtual o servidor específico y que se limita a un pequeño número de usuarios. Si en algún momento se requiere escalar la aplicación y permitir un mayor número de usuarios o su uso en diferentes máquinas, se deberá considerar una solución de autenticación más escalable y segura.

```

44 @socketio.on('connect')
45 @app.route('/login', methods=['GET', 'POST'])
46 def login():
47     username = win32api.GetUsername()
48     if request.method == 'POST':
49         username = request.form['username']
50         password = request.form['password']
51         try:
52             token = win32security.LogonUser(
53                 username,
54                 None,
55                 password,
56                 win32security.LOGON32_LOGON_NETWORK,
57                 win32security.LOGON32_PROVIDER_DEFAULT
58             )
59             user_sid = win32security.GetTokenInformation(token, win32security.TokenUser)[0]
60             username = win32api.GetUsername()
61             user_sid_str = str(user_sid)
62             session['username'] = username
63             session['domain'] = domain
64             session['user_sid'] = user_sid_str
65             session['last_activity'] = datetime.now()
66             return redirect(url_for('home'))
67         except win32security.error:
68             print(win32security.error)
69             return render_template('login.html', error='Credenciales no válidas')
70     else:
71         return

```

Figura 6.4: Función login

Además se ha desarrollado un control de inicio de sesión para poder limitar el acceso a las rutas de la web si el usuario no ha iniciado sesión o si su sesión ha caducado, ya que también se ha desarrollado una función que guarda la fecha y hora de la última actividad del usuario, estableciendo una variable global con el tiempo máximo de inactividad permitido. [Figura 6.5]

```

73 def login_required(f):
74     @wraps(f)
75     def decorated_function(*args, **kwargs):
76         if session.get("username") is None:
77             return render_template("login.html")
78         return f(*args, **kwargs)
79     return decorated_function
80
81 @app.before_request
82 def check_session_expiry():
83     if 'last_activity' not in session:
84         return
85     # Obtenemos la última vez que se accedió a la sesión
86     last_activity = session['last_activity'].astimezone(tz)
87
88     now = datetime.now(tz)
89
90     # Si la sesión existe y ha caducado, la invalidamos
91     if last_activity and now - last_activity > MAX_SESSION_TIME:
92         session.clear()
93
94     # Actualizamos el timestamp de la última actividad en la sesión
95     session['last_activity'] = now

```

Figura 6.5: Funciones de caducidad de la sesión y requerimiento de login

Por último para cerrar sesión sería tan fácil cómo eliminar el usuario de las sesiones cuando se pulse el botón de cerrar sesión.

- **Frontend:**

Siguiendo con el estilo simple y minimalista que se ha elegido para el proyecto, el inicio de sesión será lo más simple posible, con el logo de la universidad, el nombre de la herramienta y los campos correspondientes para el acceso a la aplicación.



Figura 6.6: Código HTML de login y visualización

6.4.2 Menú

El diseño del menú en este caso se ha implementado utilizando la biblioteca Bootstrap, que proporciona estilos y componentes predefinidos para ayudar en el diseño de páginas web. En este caso, se ha utilizado la clase "card" para dar formato a cada sección del menú y se ha utilizado la clase "col" para establecer una disposición en columnas para cada sección.

Cada sección consta de una imagen representativa, un título en negrita y un texto descriptivo, lo que hace que sea fácil para el usuario entender qué acción realizará al hacer clic en cada sección. También se ha incluido un botón "submit" dentro de cada tarjeta para permitir al usuario realizar la acción correspondiente.

El diseño del menú en tarjetas consigue un resultado muy visual e intuitivo, lo que facilita la navegación por la página. Además, la inclusión de una sección de "Cerrar sesión" en el pie de página proporciona una forma fácil de cerrar sesión para el usuario. En general, este diseño de menú es una buena práctica de usabilidad y accesibilidad en la interfaz de usuario.



Figura 6.7: Menú del programa

6.4.2 Cargar Datos

Se disponen dos formas de cargar los ficheros para su posterior carga de datos sobre el almacén de datos, la primera forma es mediante un fichero *zip* que debe contener la estructura correspondiente dentro, tal y cómo se ha definido previamente en los requisitos funcionales. Esta carga de datos no es tan trivial cómo descargar los archivos en una carpeta determinada del servidor, sino que deben estar en una estructura determinada, y se debe comprobar que los archivos tienen un formato específico para evitar posibles errores en la ejecución del ETL.

- **Backend:**

En ambos casos se comparten muchas funcionalidades, sólo varía la forma de iterar sobre los ficheros. Lo primero es hacer un guardado de seguridad sobre lo que se ha cargado con anterioridad, por sí hubiera un error no controlado se pudiera estudiar que es lo que falla con dichos archivos. El comprimido con los ficheros que se han cargado con anterioridad se guarda en la carpeta del proyecto llamada “backups” y se guarda con el nombre de la fecha y hora en la que se cargaron.

Una vez salvados los datos anteriores se vacía la ruta en la que se cargarán los archivos para dejarla lista para los nuevos y comprobamos la extensión del fichero que el usuario nos ha subido para asegurarnos de que no pueda ser un fichero que pueda romper el funcionamiento del programa.

Para todas las modificaciones, creaciones de archivos, carpetas y demás es fundamental el uso de paquetes como “os”, que proporciona una interfaz para interactuar con el sistema operativo subyacente en el que se está ejecutando Python

- **Fichero zip:**

Tal y cómo se indica en la web el zip que se debe subir debe cumplir con la siguiente estructura:

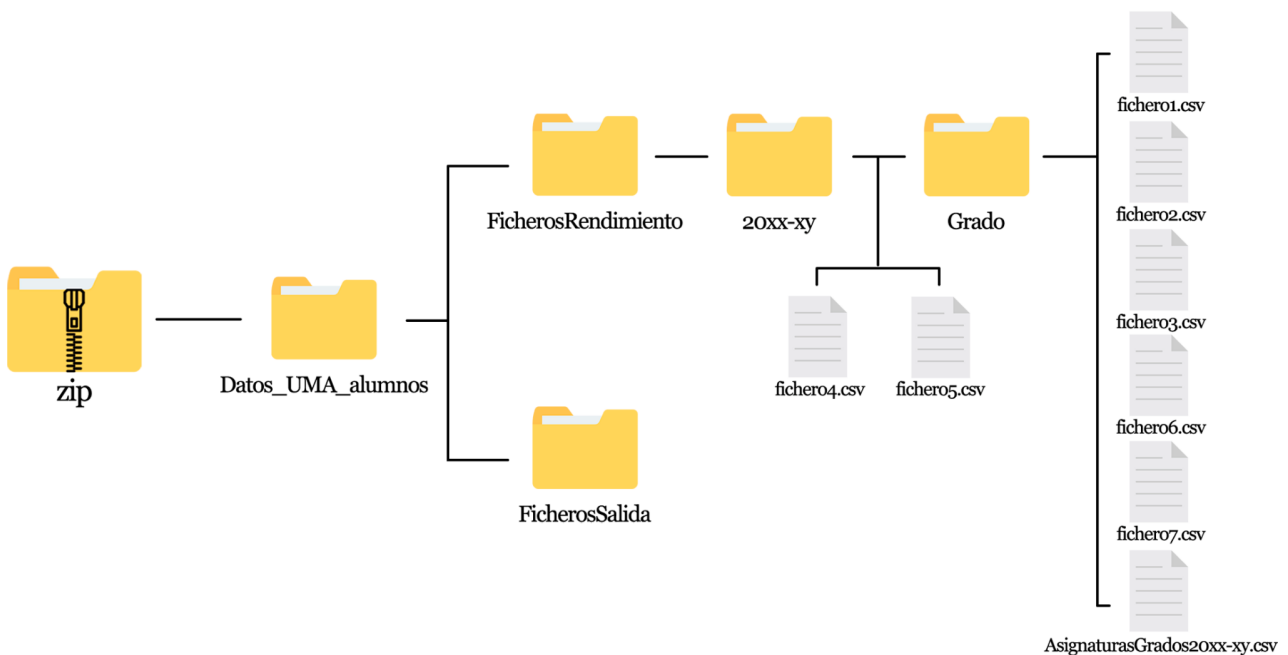


Figura 6.8: Estructura zip

En este caso descomprimos el archivo en la ruta y recorremos todos los archivos de la ruta para rectificarlos en caso de que fuera necesario, si no son UTF-8 serán recodificados para evitar problemas de compatibilidad con el ETL. Esto se consigue gracias al paquete “codecs”, para abrir los archivos recodificarlos y guardarlos con el nombre que deberá usar el ETL y “chardet”, para poder determinar cuál es la codificación previa del fichero.


```

129 def recodificar(file, root):
130     filename1, extension = os.path.splitext(file)
131     print("ruta"+root)
132     # Verifica si la extensión del archivo es permitida
133     #if extension not in ALLOWED_EXTENSIONS and filename1 != '.DS_Store':
134     print("nombre " + filename1 + " extensión " + extension)
135     if extension not in ALLOWED_EXTENSIONS and extension:
136         return 'Extensión de archivo no permitida'
137     elif extension == ".csv" or extension == ".csv":
138         print("es csv")
139         encoding = get_file_encoding(root+"\\"+file)
140         print("El encondig es ----->" + encoding)
141         if encoding == "UTF-16":
142             # Abre el archivo UTF-16 en modo lectura y UTF-16LE encoding
143             with codecs.open(root+"\\"+file, 'r', 'utf-16') as f:
144                 # Lee el contenido del archivo
145                 content = f.read()
146
147             # Abre el archivo UTF-8 en modo escritura y UTF-8-SIG encoding, cambiando el nombre a nombre_mod
148             if "AsignaturasGrados" in filename1:
149                 print("
150                     Es asignaturas")
151                 with codecs.open(root+"\\"+"Asignaturas.csv", 'w', 'utf-8-sig') as f:
152                     # Escribe el contenido del archivo UTF-16 convertido a UTF-8-SIG
153                     f.write(content)
154             else:
155                 with codecs.open(root+"\\"+filename1+"_mod.csv", 'w', 'utf-8-sig') as f:
156                     # Escribe el contenido del archivo UTF-16 convertido a UTF-8-SIG
157                     f.write(content)
158             print("El archivo " + filename1 + " ha sido cambiado de formato " + encoding + " a UTF-8-SIG")
159             os.remove(root + "\\" + file) #Eliminamos el fichero sin modificar

```

Figura 6.9: Función de recodificar

Después se renombra la carpeta dentro de “FicherosRendimiento”, que se debería llamar como el curso que se va a insertar, por “actual” para que el ETL pueda acceder a los ficheros sin problemas de enrutamiento. A continuación se comprueba si se encuentran todos los archivos necesarios para la ejecución del ETL, si no se encuentran todos se devuelven cuáles son los que faltan, y por último se comprueban las columnas de los ficheros, que se devuelven en qué fichero faltan columnas en caso de que falten para que el usuario pueda revisarlas y volver a subirlas. Esto se hace gracias a “comprobar_lista_archivos()”, en el que se recorre todos los ficheros comprobando que los nombres sean los mismos que el ETL necesita y “comprobar_columnas_todos_ficheros()”, que comprueba que cada fichero contenga exactamente las columnas necesarias.

```

64     for archivo in archivos: #Comprobamos todos los archivos que estan en la carpeta grado
65         try:
66             print("Comprobamos " +archivo["nombre"])
67             with open(RUTA_ARCHIVOS_ACTUAL + "\\Grado\\" + archivo["nombre"], 'r', encoding='utf-8-sig') as file:
68                 reader = csv.reader(file, delimiter='\\t')
69                 headers = next(reader)
70                 if all(column in headers for column in archivo["columnas"]):
71                     print(f"El archivo {archivo['nombre']} contiene todas las columnas requeridas")
72                 else:
73                     print(f"El archivo {archivo['nombre']} no contiene todas las columnas requeridas")
74                     faltan.append({"fichero": archivo["nombre"], "columnas": archivo["columnas"]})
75         except FileNotFoundError:
76             print(f"El archivo {archivo['nombre']} no existe en la carpeta Grado")
77             faltan.append({"fichero": archivo["nombre"], "columnas": "No se pudo acceder al archivo"})

```

Figura 6.10: Fragmento de `comprobar_columnas_todos_ficheros()`

- **Ficheros por separado:**

En este caso sigue un flujo muy similar, sólo que de uno en uno, cuando validamos un fichero estamos comprobando su extensión, asignándole la ruta adecuada y comprobando sus columnas. Del mismo modo se mostrará un error cuando las columnas no coinciden. Cuando se valida el primer fichero se comprimen los fichero anteriores y se limpia la estructura y se crea una nueva vacía, ya que en este caso no partimos del comprimido con la estructura creada. En este caso se renombrar todos los archivos para que no se tengan problemas de conexión durante la ejecución del ETL.

- **Frontend:**

El código HTML es una página web básica que incluye dos formularios para cargar archivos. El primer formulario permite cargar un archivo comprimido ZIP y el segundo formulario permite cargar dos archivos separados. Cada formulario incluye un campo para cargar uno o los archivos y un botón de envío. Además, hay un botón "?" por cada uno de los archivos seleccionables que muestra indicaciones de cómo debe ser el fichero que se debe seleccionar.

Figura 6.11: Página de cargar datos

6.4.3 Ejecutar comando

- **Backend:**

En este método, se utiliza el paquete “subprocess” de Python para ejecutar un comando en la terminal del sistema operativo y obtener su salida. La función se activa cuando el cliente emite un evento llamado “run_process” al servidor.

Se llama a “subprocess.Popen” con el comando a ejecutar que es una variable global y se redirecciona la salida estándar y la salida de error a tuberías de datos para que podamos procesar su salida. El comando tiene la siguiente estructura: "Ubicación del archivo DTEExec.exe" /F "Ubicación del archivo Package.dtsx del proyecto que se quiera ejecutar".

Después, se crea una lista vacía para almacenar los nombres de los procesos que se encuentran en la salida del comando. El bucle for recorre las líneas de la salida del comando con “iter(p.stdout.readline, b'')”. Cada línea se decodifica en la codificación Windows-1252 y se guarda en la variable output. Si la línea contiene la cadena 'Origen:', se extrae el nombre del proceso de la línea y se comprueba si ya se encuentra en la lista de procesos.

Si no está en la lista, se agrega a ella y se emite un evento al cliente con el nombre del proceso para que cree una nueva barra de progreso. Si la línea contiene la cadena 'Limpieza:', se busca un número en la línea con una expresión regular y se emite otro evento al cliente para actualizar la barra de progreso del proceso correspondiente. Finalmente, se devuelve una respuesta vacía con un código HTTP 204 (sin contenido) para indicar que la petición se ha procesado correctamente.

```

259 @socketio.on('run_process')
260 def ejecutar_comando():
261     print('Proceso iniciado')
262     p = subprocess.Popen(CMD, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
263     print('Proceso procesando')
264     procesos = []
265     for line in iter(p.stdout.readline, b''):
266         output = line.decode('Windows-1252')
267         if 'Origen:' in output:
268             proceso_nombre = output.strip().split(':')[1].strip() # obtener el nombre
269             if proceso_nombre not in procesos: # agregar proceso si no existe en la lista
270                 procesos.append(proceso_nombre)
271                 if len(output.split()) == 3:
272                     socketio.emit('crear_barra_progreso', proceso_nombre) # emitir evento para crear la barra de pro
273         elif 'Limpieza:' in output:
274             match = re.search(r'\d+', output)
275             if match:
276                 progreso = int(match.group())
277                 print("proceso_nombre "+ proceso_nombre+ " progreso "+ str(progreso))
278                 socketio.emit('actualizar_barra_progreso', {'proceso_nombre': proceso_nombre, 'progreso': progreso})
279     return '', 204

```

Figura 6.12: Función `ejecutar_comando()`

- **Frontend:**

Esta página tiene un botón para ejecutar el comando, junto a una barra que mostrará el progreso dinámicamente, ya que este apartado está hecho de forma dinámica, por lo que pensando en posibles futuras modificaciones a el almacén o al ETL, se creará una barra por cada nueva tabla que se proceda a cargar. Estas barras se crean gracias al script de JavaScript incluido en la página define una conexión a un servidor de Socket.io y establece manejadores para varios eventos emitidos por el servidor. El evento "crear_barra_progreso" que crea una barra con su identificador único y el evento "actualizar_barra_progreso" se emite para actualizar el progreso de un proceso existente. Cuando una actualización de barra llega al 100% entonces se actualiza también la barra de progreso total que se encontraba en la página desde el principio.

6.4.4 Crear fichero de salida

- **Backend:**

Cuando entramos a esta página aparece un selector con los cursos disponibles para generar el fichero de salida, esto se hace fácilmente ejecutando “**SELECT Curso_Academico FROM CURSO**”, las conexiones a base de datos se realizan usando el paquete para Python “pymssql”[15].

Cuando seleccionamos el curso y le damos a generar, recogemos el curso seleccionado del formulario enviado a través de la solicitud POST utilizando `request.form['curso']`. Luego, se realiza una conexión a una base de datos mediante “`pymssql.connect`” con la dirección del servidor y el nombre de la base de datos. Se utiliza un archivo de texto llamado “QuerySalida.sql” que contiene la consulta a la base de datos que se ejecutará para obtener los resultados. El archivo se lee con la función `open` y se sustituye el año del curso seleccionado en la consulta con `replace`.

Luego, se ejecuta la consulta y se recogen los resultados. A continuación, se cierra la conexión a la base de datos y se crea el elemento raíz del archivo XML, para cada registro de la consulta, se crea un elemento de registro en el archivo XML y, para cada columna de la fila, se crea un elemento de columna en el archivo XML.

```
328 # Creación del elemento raíz del archivo XML
329 root = ET.Element("Fichero")
330 root.set("xmlns:xsi", "http://www.w3.org/2001/XMLSchema-instance")
331
332 # Creación de un elemento por cada registro de la consulta
333 for registro in fichero:
334     elem_fila = ET.SubElement(root, "Registro")
335     for i, valor in enumerate(registro):
336         # Creación de un elemento por cada columna de la fila
337         nombre_columna = c.description[i][0] # Obtención del nombre de la columna
338         elem_columna = ET.SubElement(elem_fila, nombre_columna)
339         elem_columna.text = str(valor) # Inserción del valor de la columna
```

Figura 6.13: Parte de la creación del Fichero de Salida

El archivo XML se escribe utilizando “`ElementTree.write`”, la fecha y hora se obtienen con “`datetime.now()`” y se utiliza para distinguir los archivos generados ya que todos se guardan en una carpeta del proyecto llamada “FicherosSalida”. Finalmente, se copia el archivo XML generado en la carpeta de salida especificada y se redirecciona a una vista de descarga de XML.

- **Frontend:**

El código de esta parte es bastante sencillo, sólomente es un selector y un botón para accionar la creación del fichero con el curso seleccionado.

6.4.5 Comparación Ficheros

La primera parte es muy sencilla ya que sólomente son dos seleccionadores de archivos, uno para el fichero de salida ideal y otro para el que se ha generado a partir del almacén de datos. Una vez seleccionados podemos darle a comparar, y nos llevará a una página con información dinámica. La comunicación entre el servidor y el cliente se hace a través de SocketIO, una librería que permite la comunicación en tiempo real entre clientes y servidores a través de *websockets*. [8]

La comunicación entre el cliente y el servidor normalmente se realiza de manera síncrona, es decir, el servidor espera una solicitud del cliente y envía una respuesta inmediata. Pero a veces, esta forma de comunicación no es adecuada para ciertas situaciones en las que es necesario enviar datos continuamente al cliente sin interrupción, mientras que el servidor realiza otras tareas. En estos casos, la comunicación asíncrona es la solución. En este caso, se utiliza la comunicación asíncrona de SocketIO para enviar actualizaciones del progreso de la comparación de dos archivos al cliente. Al realizar la comparación de dos archivos en un hilo separado, el servidor puede seguir recibiendo solicitudes de otros clientes mientras tanto. Sin embargo, si no se utilizara la comunicación asíncrona, el servidor tendría que esperar a que se complete la comparación de archivos antes de enviar la respuesta al cliente.

Se ha utilizado el modelo de programación de concurrencia Gevent[16], que permite la ejecución de múltiples tareas en un solo hilo. De esta manera, se puede realizar una comunicación asíncrona eficiente sin la necesidad de múltiples hilos o procesos.

- **Backend:**

Para el desarrollo de este apartado se ha partido de una base de una comparación de ficheros xml, que ha sido modificada e integrada dentro de la aplicación. Esta comparación se realiza de una manera muy simple, se recorren los registros del fichero ideal, y se busca por Documento de Identidad y Titulación en el fichero generado, cuando ya se tienen ambos registros son comparados y se detecta cuáles de los campos han sido diferentes para mostrar un listado de errores.

El porcentaje de similitud se calcula teniendo en cuenta cuántos registros son totalmente idénticos y cuantos registros se han comprobado hasta el momento. En paquetes de 10 registros se van guardando en el almacén de datos, esto para reducir los accesos a bases de datos ya que los ficheros de registros suelen ser bastante grandes, en la nueva tabla creada llamada "TEMP_COMPARACION", esta tabla se vacía antes de ejecutar la comparación, por lo que sólo podremos consultar los datos de la última comparación realizada.

```
545     newData = buscarCorrespondencia(dni,titulacion,df_mod)
546     if not newData is None:
547         comparacion = newData.compare(columnData)
548         registro = []
549         for clave in columnData.keys():
550             registro.append((COLUMNAS[clave],columnData[clave],newData[clave]))
551         paquete_registros.append((registro))
552         if comparacion.empty():
553             IGUALES = IGUALES + 1
554             paquete_errores.append(0)
555         else:
556             analizarComparacion(comparacion,errores)
557             print("Los errores de esta comparacion son " + str(len(comparacion)))
558             paquete_errores.append(len(comparacion))
559             if len(comparacion) > 4:
560                 graves.append(dni)
561                 socketio.emit('actualizar_lista_registros_graves', registro)
562     CONT = CONT + 1
563     if CONT % 10 == 0:
564         print(paquete_errores)
565         insertar_datos_en_tabla(paquete_registros,paquete_errores)
566         paquete_registros = []
567         paquete_errores = []
568         string = mostrarResultado(CONT,tipos_error,errores,IGUALES)
569         socketio.emit('actualizar_lista_errores', string)
570         # Enviar el porcentaje de similitud a la página web
571         porcentaje = round(obtener_porcentaje_similitud(),2)
572         socketio.emit('actualizar_porcentaje_similitud', {'porcentaje': porcentaje, 'contenido_
```

Figura 6.14: Parte de la creación de la comparación de ficheros

Mientras transcurre la comparación en un hilo lanzado para poder enviar información de manera asíncrona mientras se comparan los ficheros se va enviando información de interés al cliente, para que el usuario pueda tener cierta idea de qué porcentaje está teniendo la comparación en tiempo real, un listado con los registros con errores graves, que son considerados a partir de 5 errores. Y por último también se envía la lista de errores junto a su porcentaje, es decir que el usuario puede ver en tiempo real los registros con mayor número de fallos, ver por qué porcentaje de error va la comparación y ver el listado de errores con porcentajes. De esta forma aunque la comparación transcurra durante un tiempo elevado, ya que pueden ser bastantes registros, el usuario tiene a su disposición información para trabajar en el transcurso de este tiempo.

- **Frontend:**

Tiene tres apartados diferenciados:

- La barra de progreso: usando Bootstrap está animada y es azul mientras se está ejecutando, muestra tanto el porcentaje de similitud cómo el número de registros iguales y totales comprobados hasta el momento. Cuando la comparación es detenida o llega al final se congela la animación y se vuelve de un color verde, mostrando el resultado final.
- Caja con errores: muestra los errores de forma dinámica en tiempo real de los registros que se han comprobado hasta el momento, junto a el porcentaje que representa frente a el total de errores en registros. Este apartado es fundamental pues puede ayudarnos a saber dónde debemos mejorar el almacén y dónde debemos centrar nuestros esfuerzos para alcanzar un mayor porcentaje de similitud.
- Tabla de registros con errores graves: una tabla a la que se añadirá de forma dinámica nuevas filas con los registros con errores graves comprobados. Además incluye un buscador para poder buscar por Documento de Identificación en la tabla, ya que en algunas comparaciones puede alcanzar un gran tamaño.

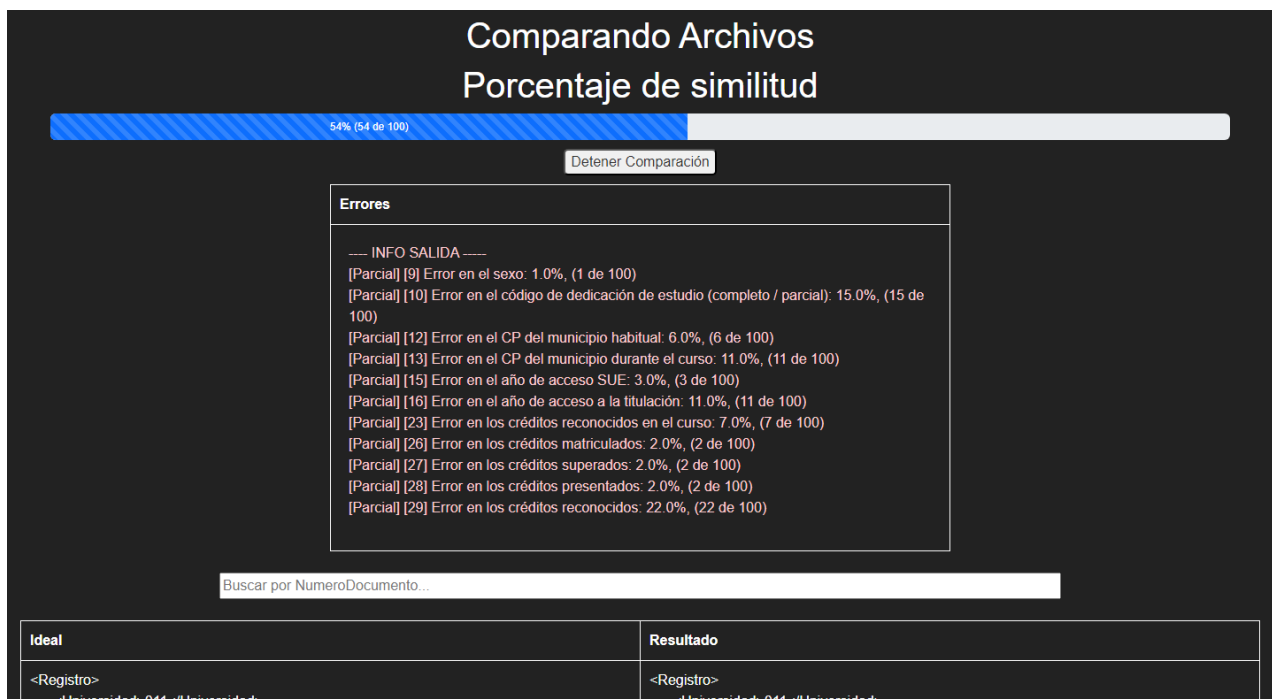


Figura 6.15: Página de comparación de ficheros

6.4.6 Ver resultados

- **Backend:**

Esta función es una ruta de una aplicación web Flask que se encarga de mostrar los resultados de una consulta a una base de datos. Primero, se establece una conexión a una base de datos utilizando la librería “pymssql”. Luego, se ejecuta una consulta SQL en la tabla “TEMP_COMPARACION” que devuelve los campos “NumeroDocumento”, “Titulacion” y “NumErrores”.

Después se configura la paginación para mostrar los resultados de la consulta de manera paginada. Se utiliza la librería Flask para obtener el parámetro de la página actual y se establece un valor predeterminado en caso de que no se encuentre la página actual. También se establece la cantidad de resultados que se mostrarán por página y se calcula el desplazamiento para obtener los resultados de la página actual.

Luego, se utiliza la clase `Pagination` de `Flask` para configurar la paginación y obtener la información necesaria para mostrar la paginación en la plantilla de resultados. Finalmente, se obtienen los resultados de la página actual a partir del desplazamiento y la cantidad de resultados por página. Se cierra la conexión a la base de datos y se devuelve una plantilla HTML que muestra los resultados de la página actual junto con la paginación correspondiente.

Para el apartado del buscador se vuelve a hacer algo similar pero en este caso la consulta tiene un campo `“WHERE NumeroDocumento LIKE '{search_term}%'”` en el que se filtra fácilmente por Número de documento. De forma muy similar se ejecuta la funcionalidad del filtrado por número de errores

- **Frontend:**

Utiliza la plantilla `base.html` y extiende su contenido. Esto se logra mediante la etiqueta `“{% extends 'base.html' %}”` al principio del archivo. El contenido adicional se agrega a la plantilla base mediante bloques que se definen con la etiqueta `“{% block %}”` y se terminan con la etiqueta `“{% endblock %}”`.

Dentro del bloque `content` hay un formulario que contiene un campo de entrada de texto y un botón para buscar en los resultados. Debajo del formulario hay una tabla para mostrar los resultados de la búsqueda. La tabla tiene una fila de encabezado con tres columnas: Número Documento, Titulación y Número Errores. La información de cada resultado se muestra en una fila separada.

También hay un elemento emergente que se muestra al hacer clic en una fila de la tabla. El elemento emergente muestra una tabla con la comparación del registro ideal y el registro resultado en dos columnas. El registro ideal y el registro resultado se muestran uno al lado del otro, y los campos que no coinciden se resaltan en rojo.

Por último encima de la tabla a la derecha se ha añadido un selector de número de errores junto a un botón de filtrar para modificar el listado de registros de la tabla por sólo registros con el número de errores seleccionado.

Ideal	Resultado
<Registro> <Universidad>011</Universidad> <Centro>29009168</Centro> <Municipio>29067</Municipio> <Campus>01101</Campus> <Titulacion>2501159</Titulacion> <TipoDocIdentidad>1</TipoDocIdentidad> <NumeroDocumento>15434417Z</NumeroDocumento> <AnioNacimiento>1995</AnioNacimiento> <PaisNacionalidad>724</PaisNacionalidad> <Sexo>M</Sexo> <DedicacionEstudio>1</DedicacionEstudio> <PaisResidencia>724</PaisResidencia> <MunicipioResidencia>11339</MunicipioResidencia> <MunicipioResidenciaCurso>99999</MunicipioResidenciaCurso> <TrabajoAnterior>0</TrabajoAnterior> <AnioAccesoSUE>2014</AnioAccesoSUE> <AnioAccesoTitulacion>2014</AnioAccesoTitulacion> <CreditosMat>006.00</CreditosMat> <CreditosMat1>000.00</CreditosMat1> <CreditosMat2>006.00</CreditosMat2>	<Registro> <Universidad>011</Universidad> <Centro>29009168</Centro> <Municipio>29067</Municipio> <Campus>01101</Campus> <Titulacion>2501159</Titulacion> <TipoDocIdentidad>1</TipoDocIdentidad> <NumeroDocumento>15434417Z</NumeroDocumento> <AnioNacimiento>1995</AnioNacimiento> <PaisNacionalidad>724</PaisNacionalidad> <Sexo>M</Sexo> <DedicacionEstudio>1</DedicacionEstudio> <PaisResidencia>724</PaisResidencia> <MunicipioResidencia>11339</MunicipioResidencia> <MunicipioResidenciaCurso>11339</MunicipioResidenciaCurso> <TrabajoAnterior>0</TrabajoAnterior> <AnioAccesoSUE>2014</AnioAccesoSUE> <AnioAccesoTitulacion>2014</AnioAccesoTitulacion> <CreditosMat>006.00</CreditosMat> <CreditosMat1>000.00</CreditosMat1> <CreditosMat2>006.00</CreditosMat2>

Figura 6.16: Elemento emergente del registro

Los errores que se encuentran en las comparaciones suelen ser errores que provienen de los ficheros de entrada y de la falta de definición en algunas reglas del ETL. Los errores más comunes suelen ser en los Créditos Reconocidos, los cuáles desde el Servicio de Calidad nos han informado que provienen de un fallo que tienen con la extracción de dichos datos, sobre todo con los alumnos que tienen algunos créditos convalidados. Otro de los errores más comunes son los códigos postales, con los cuáles se tienen algunas reglas más complejas para cambiar el código postal de los alumnos si no viven en la provincia de Málaga, o si no se tiene información disponible. Algunos de estos errores por el momento no hay más forma que arreglarlos de manera manual, pero si los ficheros de entrada contienen los datos correctos, la generación debería ser correcta también. Las decisiones sobre la corrección de estos datos recae sobre el Servicio de Calidad, quién para mejorar la calidad de los datos que se almacena deberá mejorar la definición de las reglas de inserción de los datos para una posterior mejora del ETL y la correcta alimentación del almacén.

7. Conclusiones

El desarrollo de este trabajo de fin de grado ha sido un reto muy enriquecedor y ha permitido la aplicación de los conocimientos adquiridos durante la carrera en un proyecto real y complejo. Se han utilizado diversas tecnologías y herramientas para crear una plataforma capaz de centralizar y estandarizar procesos de almacenamiento de datos en la Universidad de Málaga.

La posibilidad de poder comprender y abordar un problema real, la toma de decisiones sobre el diseño y la estructura de la solución, empezar desde cero y ver cómo paso a paso el proyecto toma forma hasta cumplir con todos los requisitos es un gran reto, y una experiencia muy enriquecedora, que una vez completada tiene la recompensa de la satisfacción de un trabajo bien hecho. Durante todo el desarrollo se han usado conocimientos adquiridos en una gran variedad de asignaturas vistas a lo largo del grado, entre las que destacaría, Diseño y Explotación de Almacenes de Datos, Sistemas de Información para Internet, Administración de Bases de Datos y Sistemas de Información para Internet.

Como pasa con cualquier desarrollo para una organización tan grande como la Universidad de Málaga, este software necesitará futuras revisiones y aplicaciones a medida que la propia Universidad crezca. El proyecto ha sido desarrollado teniendo en cuenta estas posibles futuras modificaciones, incluso el propio proyecto ayudará a detectar en qué campos se puede mejorar el almacén de datos para poder poco a poco ser capaces de generar los documentos de salida con mayor calidad.

La aplicación se podrá ir mejorando poco a poco, ya que gracias a la identificación de errores, se puede identificar cuáles son los campos en los que se producen estos errores y estudiar soluciones. Poco a poco se podrán ir definiendo nuevas reglas y estas pueden ir siendo implementadas en el ETL, logrando un almacén de datos con mayor calidad de datos, que creará ficheros de salida de mejor calidad y más parecidos a los ideales. Además existe la posibilidad de mejorar aún más si cabe la aplicación web, se podrían añadir filtros por tipo de error, o añadir la capacidad de almacenar más de una comparación simultáneamente. También puede servir como aplicación central para nuevos desarrollos que tengan relación con el Servicio de Calidad, añadiendo nuevos apartados.

Este proyecto representa una contribución importante a la gestión de datos en la Universidad, y se espera que en un futuro pueda ser utilizado para la generación de informes y análisis de datos de manera más eficiente y efectiva. Ha sido una oportunidad única, ya que se ha propuesto una solución a un problema real, partiendo de otros trabajos teniendo que entender bien lo que estaba hecho, modificarlo e integrarlo con eficacia con lo nuevo desarrollado tal y cómo se suele hacer en el mundo laboral. Se ha trabajado la gestión de proyectos y la resolución de problemas. En resumen, este trabajo ha sido una experiencia enriquecedora y desafiante que ha permitido el crecimiento personal y profesional.

8. Referencias

[1] UMA - SERVICIO DE CALIDAD, PLANIFICACIÓN ESTRATÉGICA Y RESPONSABILIDAD SOCIAL - Servicio de Calidad Universidad de Málaga.
<https://www.uma.es/calidad>

[2] Universidad de Málaga - Apuntes de la asignatura Diseño y Explotación de Almacenes de Datos.

[3] RootStack - Los pilares del Scrum: claves para la gestión de procesos
<https://rootstack.com/es/blog/los-pilares-del-scrum-claves-para-la-gestion-de-procesos#:~:text=En%20el%20portal%20oficial%20Scrum,experiencia%20y%20en%20evidencia.>

[4] Microsoft - Documentación Tecnología Analysis Services
<https://learn.microsoft.com/es-es/analysis-services/?view=asallproducts-allversions>

[5] Microsoft - Documentación SQL Server Integration Services
<https://learn.microsoft.com/es-es/sql/integration-services/sql-server-integration-services?view=sql-server-ver16>

[6] Flask - Documentación de Flask (2.3.x) <https://flask-es.readthedocs.io>

[7] SocketIO - Documentación SocketIO <https://socket.io/docs/v4/>

[8] Aply - Socket.IO vs. WebSocket: Key differences and which to use
<https://ably.com/topic/socketio-vs-websocket>

[9] Juan Trujillo, Jose Norberto Mazón, Jesús Pardillo - Diseño y explotación de almacenes de datos: Conceptos Básicos de Modelado Multidimensional. Ed. Club Universitario. 2013

[10] IBM - ¿Que es ETL? <https://www.ibm.com/es-es/topics/etl>

[11] Microsoft - ¿Qué es un Data Warehouse?
<https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-a-data-warehouse/>

[12] Pandas - Documentación oficial de pandas
<https://pandas.pydata.org/docs/>

[13] Os - Documentación oficial de os
<https://docs.python.org/3/library/os.html>

[14] *Win32security* - Documentación oficial de *win32security*
<https://pypi.org/project/pywin32/>

[15] *Pymssql* - Documentación oficial de *Pymssql*
<https://pymssql.github.io/en/stable/>

[16] *Gevent* - Documentación oficial de *gevent* <http://www.gevent.org/>

9. Apéndices

MANUAL DE USUARIO

Índice

Índice.....	65
Iniciar sesión.....	66
Menú.....	67
Cerrar sesión.....	67
Cargar datos.....	68
Crear Fichero de Salida.....	70
Comparar ficheros.....	71
Ver resultados.....	73

Iniciar sesión

Para acceder a la página, necesitarás conocer la dirección en la que se encuentra alojada la aplicación web, así como el nombre de usuario y contraseña de Windows del servidor. Sigue estos pasos para iniciar sesión:

1. Abre tu navegador web e introduce la dirección de la página.
2. Ingresa tu nombre de usuario de Windows en el campo "Usuario".
3. Ingresa tu contraseña de Windows en el campo "Contraseña".
4. Haz clic en el botón "Iniciar Sesión" para continuar.

Asegúrate de proporcionar las credenciales correctas del servidor de Windows para poder acceder correctamente a la aplicación web. Si los datos ingresados son correctos, se te redirigirá a la página principal de la aplicación. En caso de que las credenciales sean incorrectas, se mostrará un mensaje de error y deberás volver a ingresar los datos correctos para iniciar sesión.



**E.T.S. INGENIERÍA
INFORMÁTICA**
UNIVERSIDAD DE MÁLAGA

Herramienta de Carga, Comparación y Consulta datos XML

Usuario:

Contraseña:

Iniciar Sesión

Figura 9.1: Inicio de sesión

Si tienes alguna dificultad para acceder a la página o no recuerdas tus credenciales de Windows, te recomendamos ponerte en contacto con el administrador del servidor o el departamento de soporte técnico correspondiente para obtener la asistencia necesaria.

Menú

En el menú principal se encuentran claramente diferenciados las 4 opciones de la web: Cargar datos, Crear Fichero de Salida, Compara Ficheros y Ver Resultados Anterior. Cada uno contiene una breve descripción en el propio menú y para acceder a cada una de estas funcionalidades, simplemente haz clic en el botón correspondiente que se encuentra en el menú principal. Esto te llevará a la interfaz específica de cada opción, donde podrás realizar las acciones correspondientes.

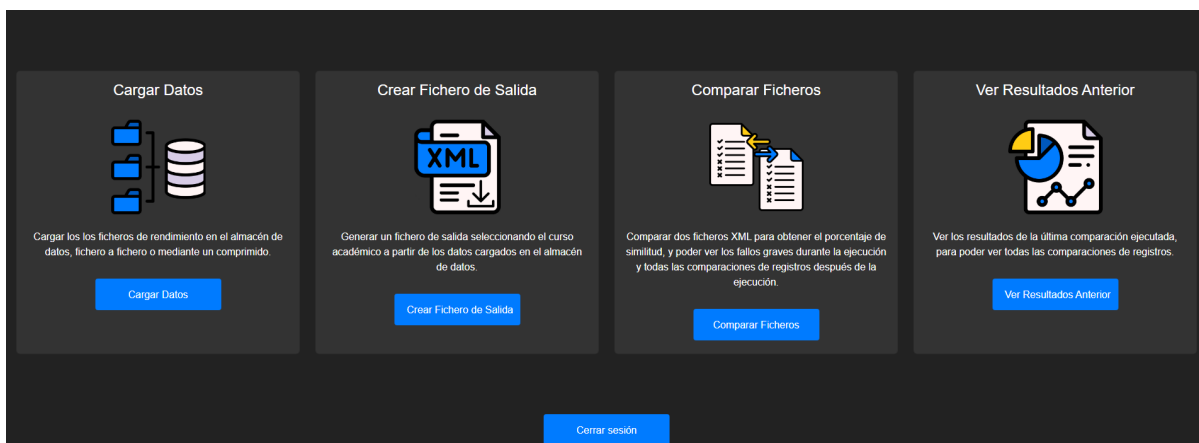


Figura 9.2: Menú principal

Cerrar sesión

Para cerrar la sesión debemos pulsar en el botón “Cerrar sesión” que se encuentra abajo en el menú principal (ver figura 8.2). Al hacer esto, se cerrará tu sesión actual y se te redirigirá a la página de inicio de sesión. Asegúrate de cerrar la sesión cada vez que termines de usar la aplicación para proteger tu privacidad y seguridad.

Cargar datos

En este apartado se encuentran dos formularios, uno de ellos a elección del usuario deberá ser completado seleccionando los archivos requeridos. La primera opción es seleccionar un fichero comprimido zip con una estructura determinada. Para conocer los detalles sobre la estructura o el formato de los archivos se puede pulsar en los botones que con “?” al lado de cada selector.

The screenshot displays a dark-themed interface for uploading data. It is divided into two main sections:

- Cargar un archivo comprimido zip:** This section features a header "Cargar un archivo comprimido zip:". Below it, the text "Archivo zip (Con la estructura):" is followed by a selection button labeled "Seleccionar archivo", a status indicator "Ninguno archivo selec.", and a help icon "?". A blue "Enviar" button is positioned below these elements.
- Cargar los archivos separados:** This section is titled "Cargar los archivos separados:" and contains three rows, each for a separate file:
 - Fichero 1:** Includes a "Seleccionar archivo" button, "Ninguno archivo selec.", a blue "Validar fichero 1" button, and a help icon "?".
 - Fichero 2:** Includes a "Seleccionar archivo" button, "Ninguno archivo selec.", a blue "Validar fichero 2" button, and a help icon "?".
 - Fichero 3:** Includes a "Seleccionar archivo" button, "Ninguno archivo selec.", a blue "Validar fichero 3" button, and a help icon "?".

Figura 9.3: Formularios Cargar Datos

En el caso de optar por la opción de seleccionar fichero a fichero deberemos validarlos uno a uno también, para ello seleccionamos el fichero y pulsamos en validar. Si el fichero se valida correctamente el texto del nombre del fichero pasará a ser verde y se marcará con un “tick” verde. Si por el contrario no se valida correctamente aparecerá una “x” roja y se mostrará un error explicando que ha fallado del formato (ver figura 9.4).

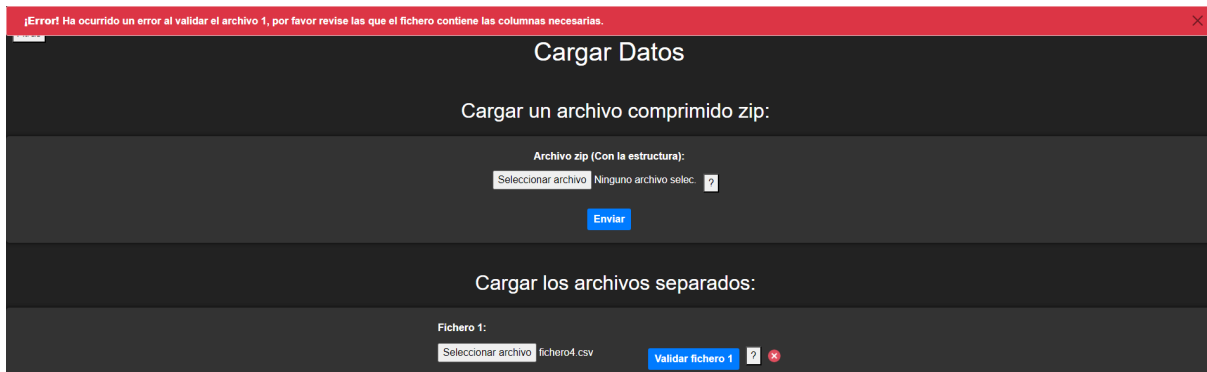


Figura 9.4: Ejemplo de error en formato

Cuando enviamos cualquiera de los dos formularios correctamente pasamos a la parte de ejecutar el comando para cargar el almacén de datos con los datos que acabamos de aportar.

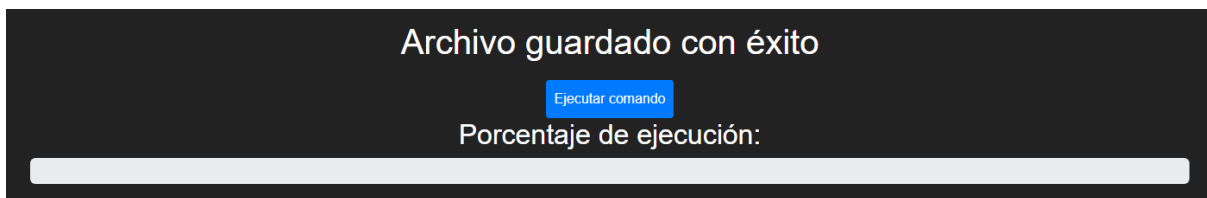


Figura 9.5: Ejecutar comando

Pulsamos en el botón de ejecutar comando y esperamos mientras se cargan todas las tablas del almacén de datos.



Figura 9.6: Porcentajes de ejecución completados

Cuando todos los porcentajes de ejecución lleguen al 100% podemos volver al menú principal para acceder a otra funcionalidad.

Crear Fichero de Salida

La opción "Crear Fichero de Salida" te permite generar un informe a partir de los datos de la base de datos. Sigue estos pasos para crear el informe:

1. Ingresa a la sección "Crear Fichero de Salida" en la aplicación web.
2. Encontrarás un selector que muestra los cursos académicos disponibles. Selecciona el curso académico del cual deseas generar el informe. Si el curso académico que buscas no está en la lista desplegable, asegúrate de haber cargado los datos correspondientes previamente.
3. Una vez seleccionado el curso académico, haz clic en el botón "Crear Informe". Ten en cuenta que este proceso puede tardar unos momentos en completarse.
4. Una vez finalizado, podrás descargar el informe generado (ver figura 9.8).

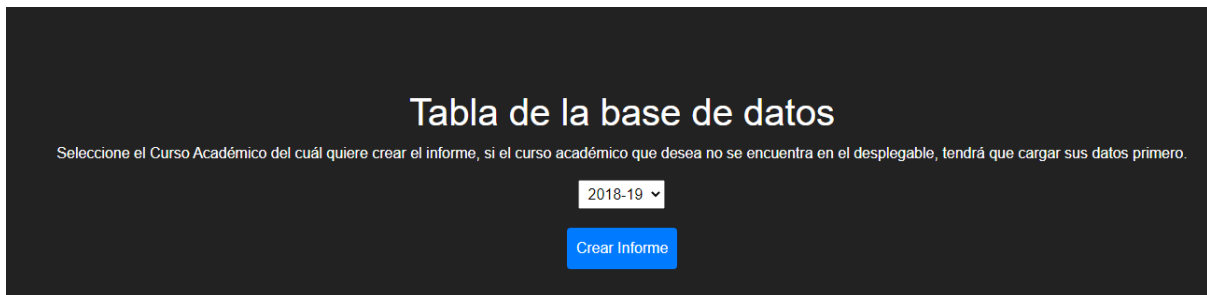


Figura 9.7: Crear informe

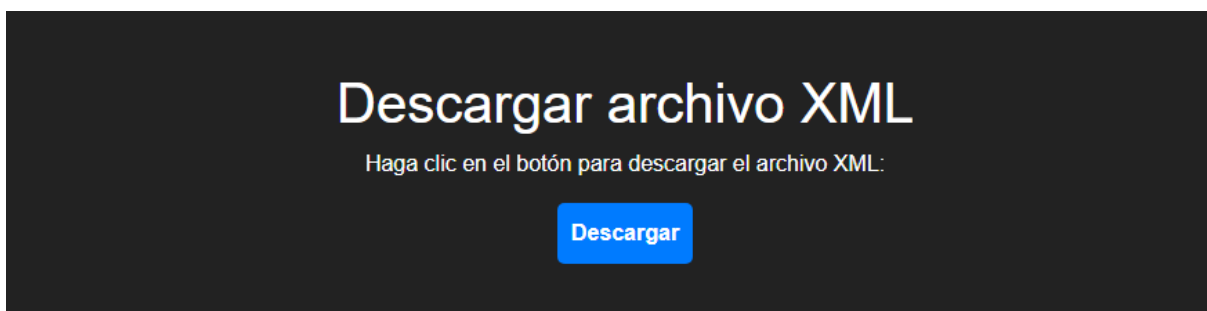
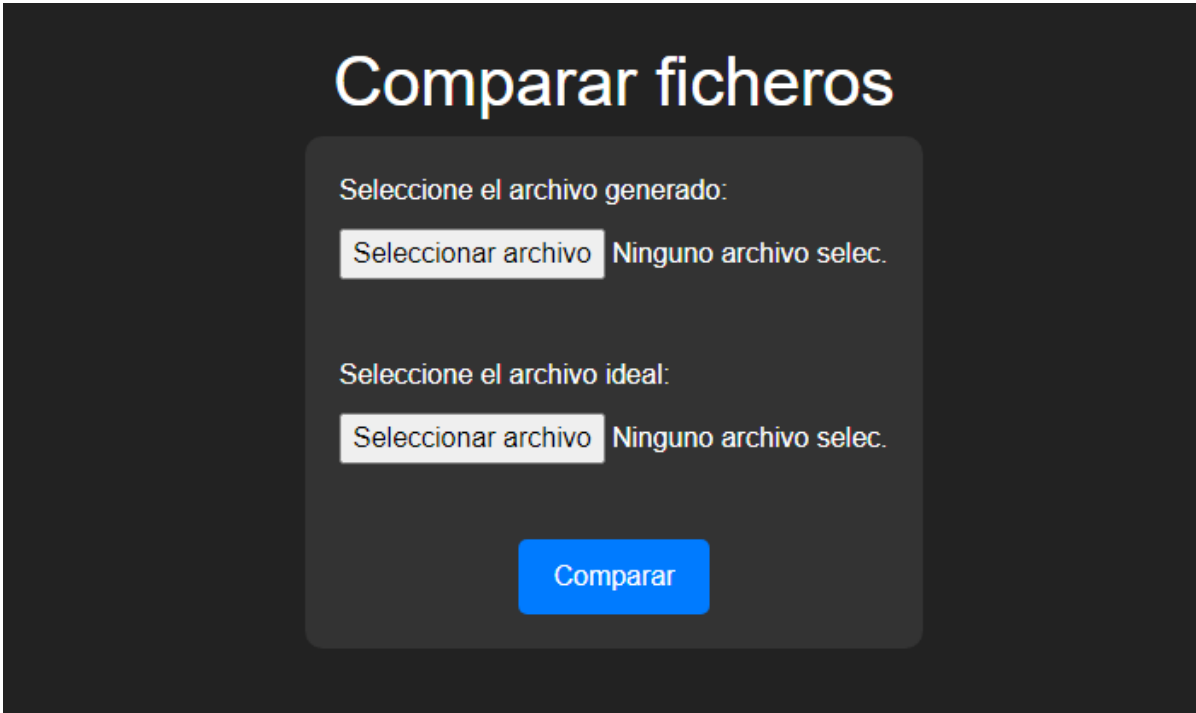


Figura 9.8: Descargar informe

Comparar ficheros

Para comparar los ficheros primero debemos seleccionar un fichero XML generado por la propia aplicación y un fichero XML ideal externo. Es importante que cada uno de los ficheros se seleccionen en su campo. Para iniciar la comparación pulsamos en el botón “Comparar” (ver figura 9.9) que nos redirigirá a otra página en la que podremos ver el estado de la comparación ya que al ser normalmente ficheros muy grandes con un número elevado de registros la comparación puede tardar una o dos horas. Al finalizar la comparación se podrán consultar todas las comparaciones de registros.



Comparar ficheros

Seleccione el archivo generado:

Seleccionar archivo Ninguno archivo selec.

Seleccione el archivo ideal:

Seleccionar archivo Ninguno archivo selec.

Comparar

Figura 9.9: Comparar archivos

Mientras se comparan los archivos podremos ver el porcentaje de similitud actual de la comparación teniendo en cuenta los registros que son iguales y el número de registros comprobados total. Se muestra también el listado de diferencias o errores encontrados en las comparaciones de registros, así podremos estudiar en qué estamos fallando más en la comparación para poder mejorar el próximo informe (ver figura 9.10). También, justo debajo de la barra de progreso hay un botón para detener la comparación y poder analizar la comparación hasta el momento en el que se paró.

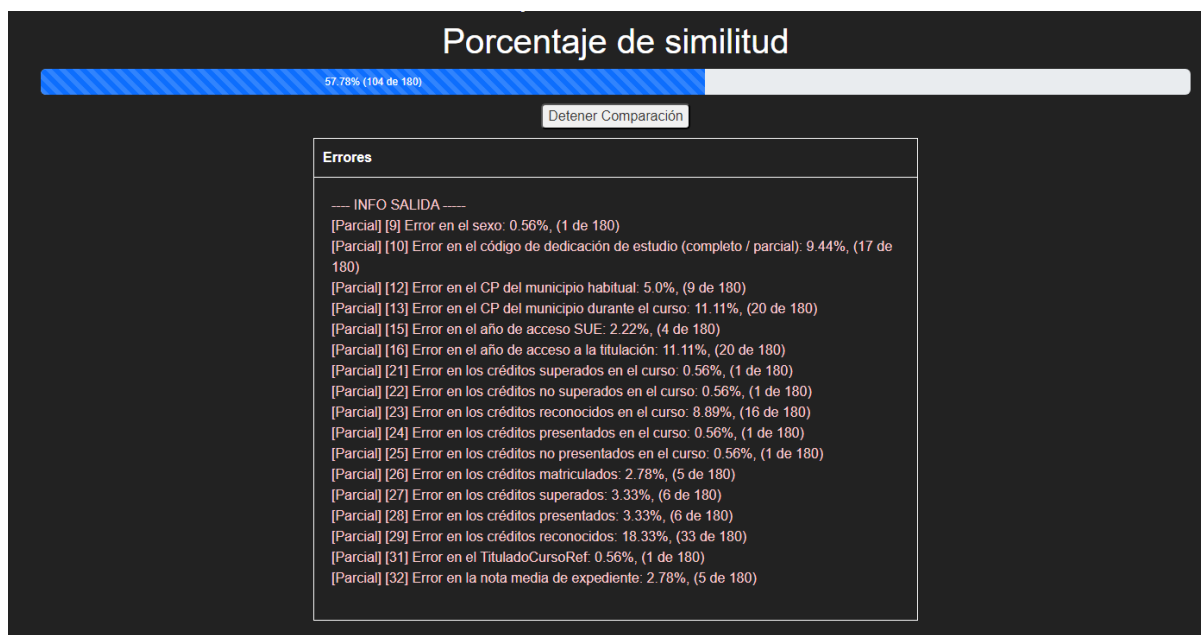


Figura 9.10: Ejecución de la comparación

A medida que la comparación se va ejecutando también se añaden filas a una tabla con los registros con errores graves, errores graves se consideran los registros que tienen al menos 5 campos diferentes. Esta tabla aumentará dinámicamente y tiene una barra de búsqueda para poder buscar por el campo de Número de Documento. En el listado se muestran los registros destacando en rojo las diferencias encontradas.

Para poder buscar en el listado de registros graves se debe escribir lo que se desea buscar en el campo de “Buscar por NumeroDocumento” (ver figura 9.11) y a medida que se inserte el texto la tabla sólo mostrará los registros que coinciden con el número insertado. Para borrar la búsqueda se debe borrar el texto insertado y la tabla volverá a mostrar todos los registros con errores graves.

Buscar por NumeroDocumento	
Ideal	Resultado
<Registro> <Universidad>011</Universidad> <Centro>29009168</Centro> <Municipio>29067</Municipio> <Campus>01101</Campus> <Titulacion>2501159</Titulacion> <TipoDocIdentidad>1</TipoDocIdentidad> <NumeroDocumento>25741586D</NumeroDocumento> <AnioNacimiento>1993</AnioNacimiento> <PaisNacionalidad>724</PaisNacionalidad> <Sexo>H</Sexo> <DedicacionEstudio>1</DedicacionEstudio> <PaisResidencia>724</PaisResidencia> <MunicipioResidencia>29630</MunicipioResidencia> <MunicipioResidenciaCurso>29630</MunicipioResidenciaCurso> <TrabajoAnterior>0</TrabajoAnterior> <AnioAccesoSUE>2011</AnioAccesoSUE> <AnioAccesoTitulacion>2011</AnioAccesoTitulacion> <CreditosMat>054.00</CreditosMat> <CreditosMat1>054.00</CreditosMat1> <CreditosMat2>000.00</CreditosMat2> <CreditosMat3>000.00</CreditosMat3> <CreditosSupCurso>000.00</CreditosSupCurso> <CreditosNoSupCurso>054.00</CreditosNoSupCurso> <CreditosRecCurso>186.00</CreditosRecCurso> <CreditosPreCurso>000.00</CreditosPreCurso> <CreditosNoPreCurso>054.00</CreditosNoPreCurso> <CreditosMatInicio>054.00</CreditosMatInicio>	<Registro> <Universidad>011</Universidad> <Centro>29009168</Centro> <Municipio>29067</Municipio> <Campus>01101</Campus> <Titulacion>2501159</Titulacion> <TipoDocIdentidad>1</TipoDocIdentidad> <NumeroDocumento>25741586D</NumeroDocumento> <AnioNacimiento>1993</AnioNacimiento> <PaisNacionalidad>724</PaisNacionalidad> <Sexo>H</Sexo> <DedicacionEstudio>1</DedicacionEstudio> <PaisResidencia>724</PaisResidencia> <MunicipioResidencia>29630</MunicipioResidencia> <MunicipioResidenciaCurso>29630</MunicipioResidenciaCurso> <TrabajoAnterior>0</TrabajoAnterior> <AnioAccesoSUE>2011</AnioAccesoSUE> <AnioAccesoTitulacion>2018</AnioAccesoTitulacion> <CreditosMat>054.00</CreditosMat> <CreditosMat1>054.00</CreditosMat1> <CreditosMat2>000.00</CreditosMat2> <CreditosMat3>000.00</CreditosMat3> <CreditosSupCurso>000.00</CreditosSupCurso> <CreditosNoSupCurso>054.00</CreditosNoSupCurso> <CreditosRecCurso>000.00</CreditosRecCurso> <CreditosPreCurso>000.00</CreditosPreCurso> <CreditosNoPreCurso>054.00</CreditosNoPreCurso> <CreditosMatInicio>522.00</CreditosMatInicio>

Figura 9.11: Listado de registros con errores graves

Cuando la comparación finalice o el usuario la detenga aparecerá otro botón que pondrá ver resultados. Se puede acceder desde ahí mismo o se puede acceder desde el menú principal.

Ver resultados

Cuándo accedemos a este último apartado tenemos una tabla paginada con el listado de los registros que se han comparado, se muestra los dos atributos diferenciadores de los registros, el NumeroDocumento y la Titulacion, junto a el número de errores detectado en la comparación. Para navegar entre las páginas de la tabla se dispone de unos botones en la parte inferior de la misma. Justo encima de la tabla en la parte izquierda hay un campo para poder buscar por Número de Documento.

Para buscar debemos introducir en el campo “Buscar por número de documento” lo que se quiere buscar y pulsar en el botón buscar. Si se quiere eliminar el filtro de búsqueda y volver al listado completo se puede pulsar en el botón “Limpiar búsqueda”, actualizando de nuevo al listado del comienzo.

Resultados

Numero Documento	Titulacion	Numero Errores
02198836J	2501159	0
03145345A	2501159	0
04638455E	2501159	1
06590368V	2501159	0
09079577M	2501159	0
09080216T	2501159	0
09083902Y	2501159	0
15157117W	2501159	0
15434417Z	2501159	2
23641366B	2501159	2
24759004X	2501159	0
24784995B	2501159	2

Figura 9.12: Resultados de registros comparados

En el listado también se puede hacer click sobre cualquier celda para ver la comparación de ese registro en una ventana emergente. Se mostrarán los dos registros, indicando cuál es cuál, y se destaca con el fondo en rojo las diferencias entre los registros.

Datos del Registro

Ideal	Resultado
<Registro>	<Registro>
<Universidad>011</Universidad>	<Universidad>011</Universidad>
<Centro>29009168</Centro>	<Centro>29009168</Centro>
<Municipio>29067</Municipio>	<Municipio>29067</Municipio>
<Campus>01101</Campus>	<Campus>01101</Campus>
<Titulacion>2501159</Titulacion>	<Titulacion>2501159</Titulacion>
<TipoDocIdentidad>1</TipoDocIdentidad>	<TipoDocIdentidad>1</TipoDocIdentidad>
<NumeroDocumento>04638455E</NumeroDocumento>	<NumeroDocumento>04638455E</NumeroDocumento>
<AnioNacimiento>1996</AnioNacimiento>	<AnioNacimiento>1996</AnioNacimiento>
<PaisNacionalidad>724</PaisNacionalidad>	<PaisNacionalidad>724</PaisNacionalidad>
<Sexo>M</Sexo>	<Sexo>M</Sexo>
<DedicacionEstudio>1</DedicacionEstudio>	<DedicacionEstudio>1</DedicacionEstudio>
<PaisResidencia>724</PaisResidencia>	<PaisResidencia>724</PaisResidencia>
<MunicipioResidencia>16001</MunicipioResidencia>	<MunicipioResidencia>16001</MunicipioResidencia>
<MunicipioResidenciaCurso>99999</MunicipioResidenciaCurso>	<MunicipioResidenciaCurso>16001</MunicipioResidenciaCurso>
<TrabajoAnterior>0</TrabajoAnterior>	<TrabajoAnterior>0</TrabajoAnterior>
<AnioAccesoSUE>2015</AnioAccesoSUE>	<AnioAccesoSUE>2015</AnioAccesoSUE>
<AnioAccesoTitulacion>2015</AnioAccesoTitulacion>	<AnioAccesoTitulacion>2015</AnioAccesoTitulacion>
<CreditosMat>060.00</CreditosMat>	<CreditosMat>060.00</CreditosMat>
<CreditosMat1>060.00</CreditosMat1>	<CreditosMat1>060.00</CreditosMat1>
<CreditosMat2>000.00</CreditosMat2>	<CreditosMat2>000.00</CreditosMat2>
<CreditosMat3>000.00</CreditosMat3>	<CreditosMat3>000.00</CreditosMat3>
<CreditosSupCurso>054.00</CreditosSupCurso>	<CreditosSupCurso>054.00</CreditosSupCurso>
<CreditosNoSupCurso>006.00</CreditosNoSupCurso>	<CreditosNoSupCurso>006.00</CreditosNoSupCurso>

Figura 9.13: Resultados de registros comparados

GUÍA DE INSTALACIÓN

Índice

Índice.....	76
1. Instalación Software Necesario.....	77
1.1 Python y Flask.....	77
1.2 Microsoft Management Studio.....	78
1.2 Visual Studio.....	79
1.3 Dependencias y Configuraciones Proyecto.....	82
2. Nginx y ejecución de la aplicación.....	83
2.1 Instalar Nginx.....	83
2.2 Configuración y ejecución.....	84

1. Instalación Software Necesario

1.1 Python y Flask

Primero se descarga Python desde su página web oficial la versión 3.11.2. Es importante asegurarse de marcar la opción "Add Python.exe to Path" durante la instalación, para que Python pueda ser accedido desde cualquier directorio en la terminal o línea de comandos.

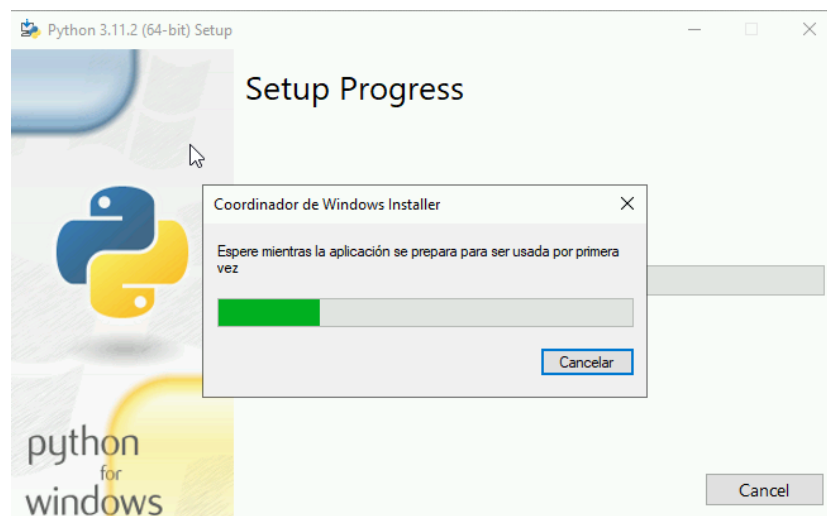


Figura 8.1: Instalación Python

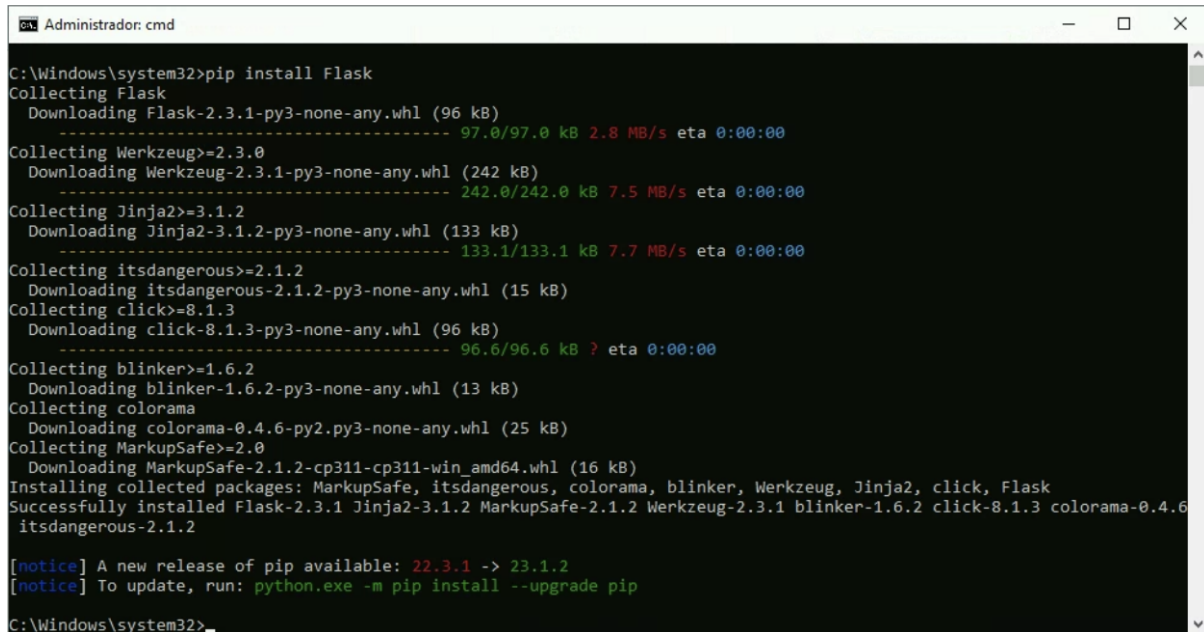
Para comprobar que la instalación de Python se realizó correctamente, se puede ejecutar el siguiente comando en la línea de comandos o terminal: "python". Esto debería iniciar el intérprete de Python y mostrar el número de versión instalada, junto con el indicador de entrada de comando ">>>" que indica que Python está listo para recibir instrucciones.

```
Administrador: cmd - python
Microsoft Windows [Versión 10.0.17763.2565]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Figura 8.2: Comprobación instalación python

Una vez que se ha comprobado que la instalación de Python se ha realizado correctamente, se puede proceder a instalar Flask para la aplicación. Esto se puede hacer fácilmente utilizando el administrador de paquetes de Python llamado "pip". Para instalar Flask, se puede ejecutar el siguiente comando en la línea de comandos o terminal: "pip install Flask". Esto descargará e instalará automáticamente la última versión estable de Flask y sus dependencias necesarias.



```
Administrador: cmd
C:\Windows\system32>pip install Flask
Collecting Flask
  Downloading Flask-2.3.1-py3-none-any.whl (96 kB)
----- 97.0/97.0 kB 2.8 MB/s eta 0:00:00
Collecting Werkzeug>=2.3.0
  Downloading Werkzeug-2.3.1-py3-none-any.whl (242 kB)
----- 242.0/242.0 kB 7.5 MB/s eta 0:00:00
Collecting Jinja2>=3.1.2
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
----- 133.1/133.1 kB 7.7 MB/s eta 0:00:00
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.1.3
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
----- 96.6/96.6 kB ? eta 0:00:00
Collecting blinker>=1.6.2
  Downloading blinker-1.6.2-py3-none-any.whl (13 kB)
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.2-cp311-cp311-win_amd64.whl (16 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, Flask
Successfully installed Flask-2.3.1 Jinja2-3.1.2 MarkupSafe-2.1.2 Werkzeug-2.3.1 blinker-1.6.2 click-8.1.3 colorama-0.4.6
itsdangerous-2.1.2

[notice] A new release of pip available: 22.3.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Windows\system32>
```

Figura 8.3: Instalación Flask

1.2 Microsoft Management Studio

Para crear el almacén de datos, es necesario instalar Microsoft SQL Server Management Studio. Una vez instalado, se debe abrir la herramienta y conectarse al servidor de base de datos correspondiente. A continuación, se puede crear una nueva base de datos con el nombre deseado, en este caso "UMA_DW".

Para crear las tablas y cargar los datos en el almacén, se debe ejecutar la consulta proporcionada en la carpeta raíz del proyecto llamada "Query_genera_almacén". Esta consulta creará las tablas necesarias en la base de datos y cargará los datos correspondientes.

Para ejecutar la consulta, se debe abrir una nueva consulta en Management Studio, seleccionar la base de datos "UMA_DW" y pegar la consulta en la ventana de edición. A continuación, se debe ejecutar la consulta haciendo clic en el botón "Ejecutar" en la barra de herramientas o presionando la tecla F5. Una vez ejecutada la consulta, se deben crear las tablas y cargar los datos en la base de datos "UMA_DW".

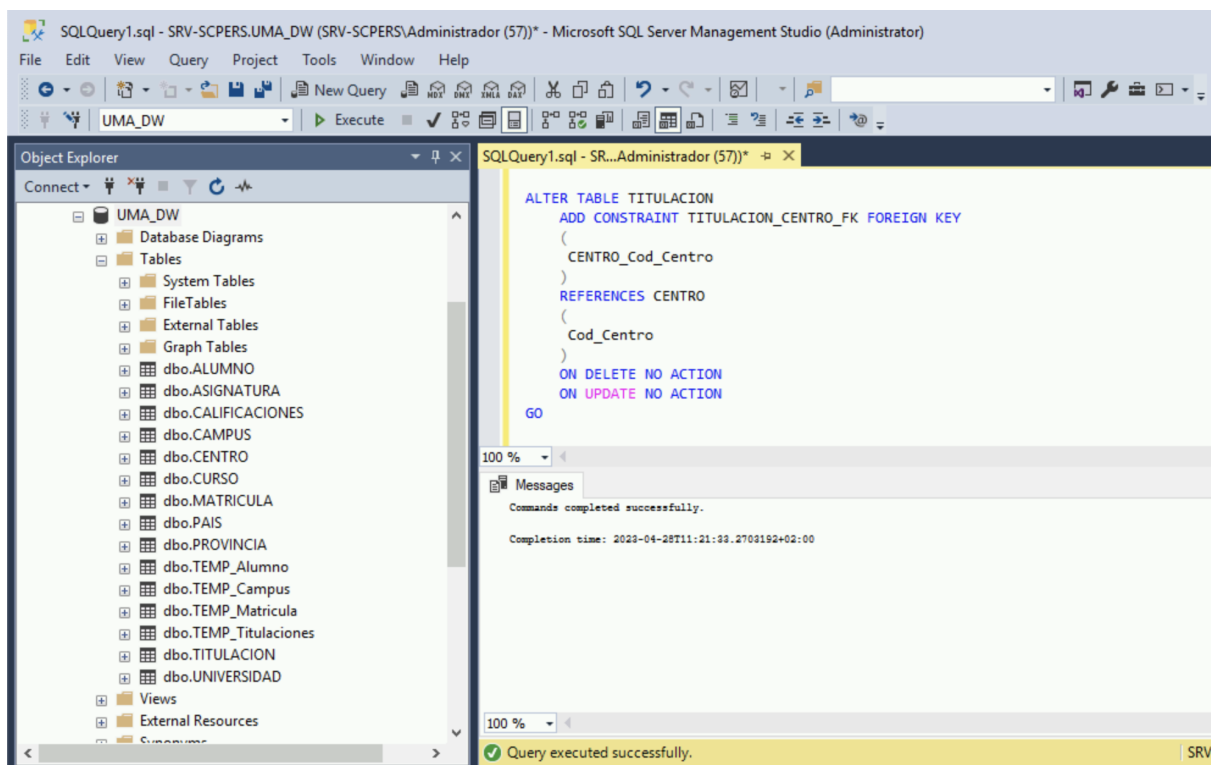


Figura 8.4: Query creación almacén de datos

1.2 Visual Studio

Para configurar el proyecto de Integration Services, es necesario instalar Visual Studio con el paquete de SQL Server Integration Services (SSIS). Para evitar posibles problemas de compatibilidad con las versiones de Visual Studio se recomienda el uso de la versión 15.9.39 de Microsoft SQL Server Data Tools para Visual Studio 2017, siguiendo estos pasos:

1. Abrimos nuestro navegador web y nos dirigimos al sitio web oficial de Microsoft.
2. En la barra de búsqueda escribimos "SQL Server Data Tools para Visual Studio 2017".

3. En la página de descarga, hacemos clic en el botón "Descargar".
4. Una vez descargado, hacemos doble clic en el archivo para iniciarlo.
5. En la ventana de instalación, seguimos las instrucciones que nos aparecen en pantalla.
6. En algún momento de la instalación, se nos pedirá seleccionar los componentes que deseamos instalar. Asegurémonos de seleccionar la opción "SQL Server Integration Services" para instalar el paquete SSIS.

Una vez instalado, se debe abrir el proyecto "Proyecto de Integration Services2" que se encuentra en la carpeta "Proyecto Integración de Datos" del proyecto.

Una vez abierto el proyecto, se deben reconfigurar las variables de ruta para volver a enrutar todas las conexiones. Para ello, se debe hacer clic derecho sobre el fondo del "Flujo de control" y seleccionar "Variables" en el menú desplegable. En la ventana de variables, se deben actualizar las rutas de las conexiones para que apunten a la base de datos "UMA_DW" creada anteriormente. Es importante revisar cuidadosamente cada variable y asegurarse de que todas las rutas están actualizadas correctamente. Una vez actualizadas las variables, se debe guardar el proyecto y compilarlo para asegurarse de que no hay errores en la configuración.

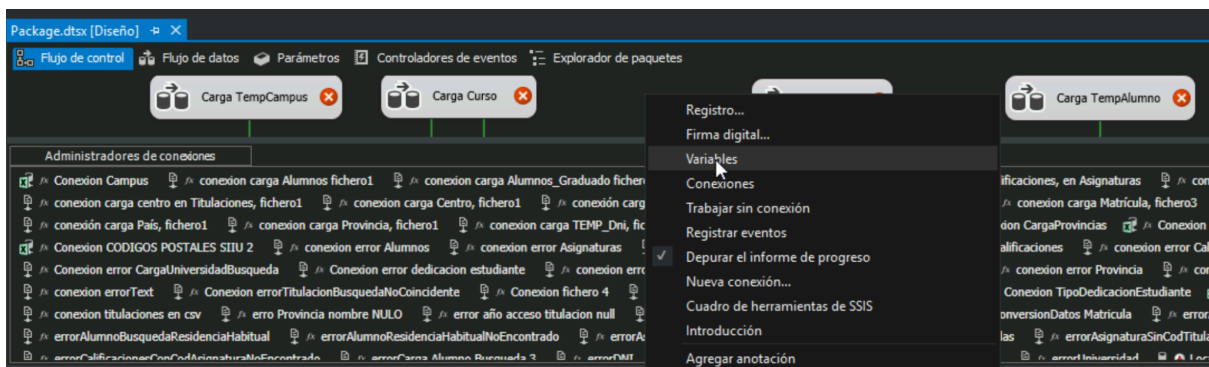


Figura 8.5: Botón variables Visual Studio

Debemos actualizar el valor de las variables de ruta por las nuevas rutas que debemos asignar en nuestra instalación:

 A screenshot of the 'Variables' window in Visual Studio. It displays a table with the following columns: 'Nombre', 'Ámbito', 'Tipo de datos', 'Valor', and 'Expresión'. The table contains four rows of variables related to file paths.

Nombre	Ámbito	Tipo de datos	Valor	Expresión
RutaGenericos	Package	String	C:\Users\Administrador\Desktop\TFG - Ale\Datos Genericos	
RutaDatos	Package	String	C:\Users\Administrador\Desktop\TFG - Ale\Datos\Datos_UMA_alumnos\FicherosRendimiento\actual	
RutaDatosGrado	Package	String	C:\Users\Administrador\Desktop\TFG - Ale\Datos\Datos_UMA_alumnos\FicherosRendimiento\actual\Grado	
RutaErrores	Package	String	C:\Users\Administrador\Desktop\TFG - Ale\Ficheros de error	

Figura 8.6: Pestaña variables Visual Studio

Para configurar las conexiones con el almacén de datos en el proyecto de Integration Services, se debe abrir la ventana de "Administradores de conexiones". Esta ventana muestra una lista de todas las conexiones disponibles en el proyecto.

Para configurar las conexiones con el almacén de datos, se debe seleccionar la primera conexión de la lista y hacer doble clic sobre ella. Esto abrirá la ventana de configuración de la conexión. En esta ventana, se debe ingresar el nombre del servidor de bases de datos en el campo correspondiente y seleccionar el nombre del nuevo almacén de datos que se ha creado anteriormente.

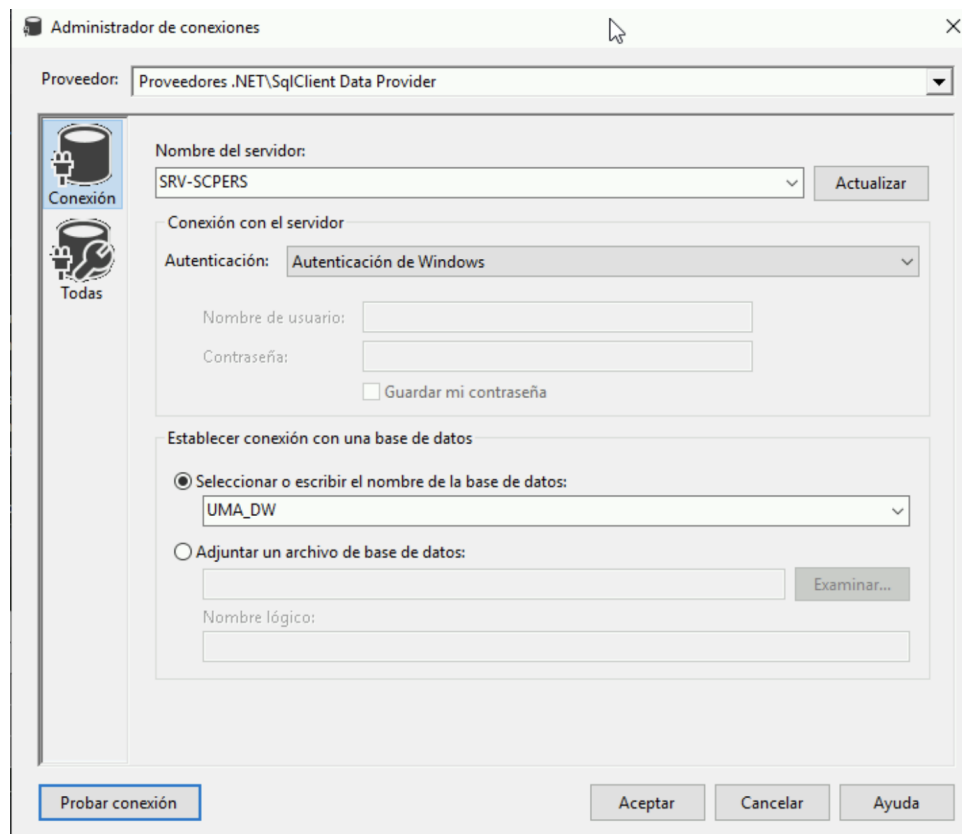


Figura 8.7: Configuración Administrador de conexiones

Se debe repetir el mismo proceso para la segunda conexión en la lista. Es importante asegurarse de que ambas conexiones estén configuradas correctamente para que el proyecto de Integration Services pueda acceder al almacén de datos y extraer los datos necesarios.

Una vez que se han configurado correctamente las conexiones con el almacén de datos, se debe guardar el proyecto y compilarlo para asegurarse de que no hay errores en la configuración. Con las conexiones configuradas correctamente, el proyecto de Integration Services estará listo para ejecutarse y extraer los datos del almacén de datos.

1.3 Dependencias y Configuraciones Proyecto

Para instalar todas las dependencias del proyecto, se debe abrir la consola y ejecutar el comando "pip install -r requirements.txt". Este comando instalará todas las dependencias necesarias para que la aplicación web pueda funcionar correctamente.

Es importante asegurarse de que se encuentra en el directorio raíz del proyecto antes de ejecutar el comando. De lo contrario, se pueden producir errores al intentar instalar las dependencias. Una vez que se ha ejecutado el comando, se deben esperar unos minutos a que se instalen todas las dependencias necesarias.

Para la configuración de las variables del proyecto tendremos que acceder a el archivo "globales.py" y cambiar la definición de esas variables con las nuevas rutas del equipo en el que se instale la aplicación:

```
globales.py > ...
1 from datetime import datetime, timedelta
2 import pytz
3 import win32api
4
5 # EDITAR PARA LA CONFIGURACION :
6 RUTA_MATERIAL = "C:\\Users\\AleGu\\Desktop\\TFG - ENTREGA"
7 RUTA_PROYECTO = "C:/Users/AleGu/Desktop/TFG Proyecto"
8
9 CMD = ['C:\\Program Files (x86)\\Microsoft SQL Server\\150\\DTS\\Binn\\DTEExec.exe', '/F', 'C:\\Users\\AleGu\\Desktop\\TFG - ENTREGA\\Proyecto
10
11 SERVER = 'DESKTOP-MEI3U6C\\MSSQLSERVER1'
12 DATABASE='AlmacenUMA'
```

Figura 8.8: Variables globales del archivo globales.py

“RUTA_MATERIAL” será dónde se guardarán los datos de los resultados, los backups y dónde se trabajará con los archivos en general. Las demás rutas se completan a través de la anterior por lo que no deberían de modificarse, a excepción de la última “RUTA_PROYECTO” que es dónde se encuentra el proyecto de python cómo tal y dónde se realizará la comparación. Se deberá respetar el formato de las barras en cada una de las variables, por ejemplo en “RUTA_MATERIAL” serán dobles barras y en “RUTA_PROYECTO” sólo una barra.

También en “CMD” se debe cambiar el primer atributo de la lista por la ruta en la que se encuentre el fichero DTExec.exe que es del paquete de Integration Services.

Y por último en “SERVER” el nombre del servidor de bases de datos y el “DATABASE” el nombre del almacén que se ha creado.

En caso de que se agreguen nuevas titulaciones o grados se debe sustituir o modificar el fichero “Titulaciones.csv” que se encuentra en la carpeta “Datos Genericos” del proyecto. Estos cambios al ser muy puntuales se deben realizar manualmente, ya sea modificando o sustituyendo el fichero por el nuevo. De igual forma si fuera necesario añadir algún país con el fichero “PAISES.csv” o alguna modificación más específica como el tipo DNI o el “Tipo Dedicación Estudiante”.

2. Nginx y ejecución de la aplicación

2.1 Instalar Nginx

Para poder acceder a la aplicación web desde otros equipos, es necesario realizar algunas configuraciones adicionales. Una de ellas es descargar la última versión de Nginx para Windows desde el sitio web oficial de Nginx: <https://nginx.org/en/download.html>.

Una vez descargado el archivo, debemos descomprimirlo en un directorio de nuestra elección, por ejemplo, en la ruta "C:\Program Files (x86)\nginx".

A continuación, debemos agregar la ruta del directorio de Nginx al PATH del sistema. Para hacer esto, podemos seguir los siguientes pasos:

1. Hacer clic con el botón derecho en "Mi PC" y seleccionar "Propiedades".
2. Hacer clic en "Configuración avanzada del sistema" y luego en "Variables de entorno".
3. Buscar la variable "PATH" en la lista de variables del sistema y hacer clic en "Editar".
4. Agregar la ruta del directorio de Nginx al final de la variable "PATH", separada por un punto y coma (;). Por ejemplo, podríamos agregar la siguiente ruta: C:\Program Files (x86)\nginx.

2.2 Configuración y ejecución

Una vez realizados instalado Nginx podemos abrir una ventana del símbolo del sistema como administrador y cambiar al directorio donde descomprimos Nginx. Para iniciar Nginx, podemos ejecutar el siguiente comando: `nginx.exe`. Si todo ha funcionado correctamente, deberíamos ver un mensaje que indica que Nginx se ha iniciado correctamente.

Para comprobar que Nginx está funcionando correctamente, podemos abrir un navegador web e introducir la dirección <http://localhost>. Si Nginx está funcionando correctamente, deberíamos ver una página de bienvenida de Nginx. Ahora, podemos configurar Nginx para que redirija el tráfico a nuestra aplicación web.

Una vez instalado Nginx se deberá configurar para poder alojar la aplicación:

1. Debemos ir al directorio de instalación de Nginx (por ejemplo, `C:\Program Files (x86)\nginx\conf`) debería haber un fichero con un nombre que termine en `.conf`. Por ejemplo, `nginx.conf`.
2. En el archivo escribir o editar lo siguiente:

```
server {  
    listen 80;  
    server_name DOMINIO;  
    location / {  
        proxy_pass http://localhost:PUERTO_APP;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

Sustituyendo **DOMINIO**, por el dominio (en nuestro caso será la IPv4 pública que servidor, que se puede obtener haciendo “ipconfig” en el símbolo del sistema) y **PUERTO_APP**, por el puerto de Flask, que en este caso es el 5000.

3. Por último en este mismo archivo debemos ir al bloque "http" y añadir la siguiente línea para el tamaño de los archivos permitidos, esto se puede modificar, en principio recomendamos 100M:

```
client_max_body_size 100M;
```

Para permitir el acceso a la aplicación Flask desde internet, es necesario configurar el firewall de Windows para permitir el tráfico entrante en el puerto que se está utilizando para la aplicación. Para hacerlo, se debe seguir los siguientes pasos:

1. Abrir el "Panel de Control" de Windows y hacer clic en "Sistema y Seguridad" y luego en "Firewall de Windows".
2. Hacer clic en "Configuración Avanzada" en el panel izquierdo.
3. Hacer clic en "Reglas de Entrada" en el panel izquierdo y luego hacer clic en "Nueva Regla" en el panel derecho.
4. Se abrirá el "Asistente para Nueva Regla de Entrada". Seleccionar "Puerto" y hacer clic en "Siguiente".

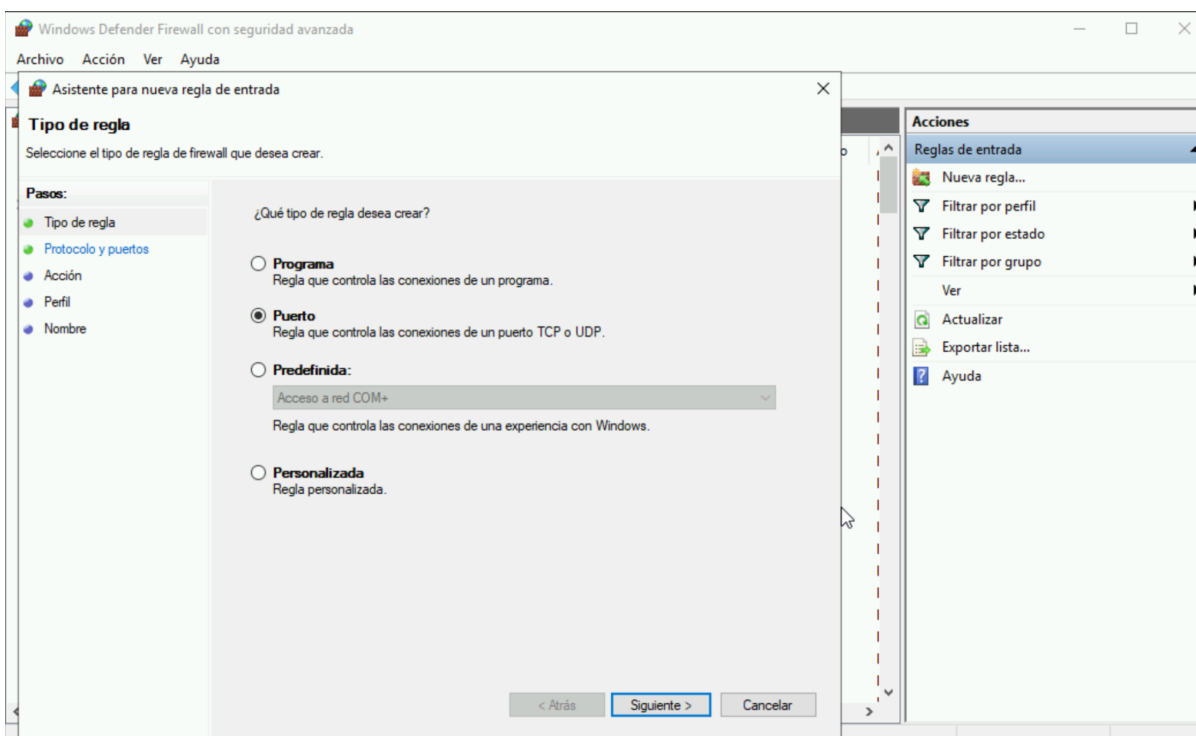


Figura 8.9: Tipo de regla del asistente para nueva regla entrada

5. Seleccionar "TCP" o "UDP", dependiendo del protocolo que use la aplicación, y luego seleccionar "Puertos Locales Específicos". Ingresar el número del puerto que se quiere abrir y hacer clic en "Siguiente".

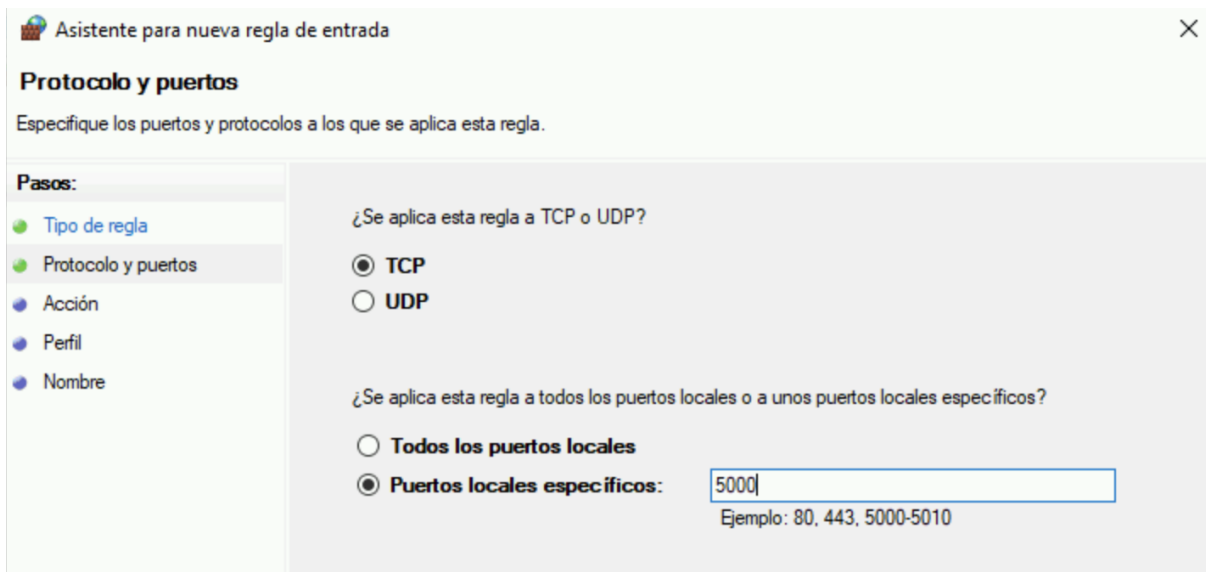


Figura 8.10: Protocolo y puertos del asistente para nueva regla entrada

6. Seleccionar "Permitir la Conexión" y hacer clic en "Siguiente".
7. Seleccionar los perfiles de red en los que se quiere permitir esta conexión y hacer clic en "Siguiente". Si no se está seguro de qué perfiles de red seleccionar, se puede seleccionar "Todos".
8. Ingresar un nombre y una descripción para la regla y hacer clic en "Finalizar".

Por último deberemos iniciar nuestra aplicación python con normalidad, en este caso ir con el Símbolo del sistema hasta la ruta en la que se encuentra, y ejecutar "python app.py". Y si tenemos ejecutándose el nginx ya podríamos acceder a el dominio con nuestra aplicación web corriendo desde cualquier equipo que se encuentre en nuestra misma red.



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga