

Grado en Ingeniería Informática 2022-2023

Trabajo final de grado

“Modelo de calidad de datos orientado a empresas”



Fecha	05/09/2023
Autor	Daniel Alvarez Reyes (Campus de Leganés) 100383460@alumnos.uc3m.es
Tutor	Alejandro Rey Lopez alejandro.rey@uc3m.es

ABSTRACT

Currently, the volume of data handled by companies is increasing. Technology and digitization have generated an enormous amount of information that, although very valuable, is difficult to manage and control. Therefore, it is essential to manage and maintain certain levels of data quality.

Proper data management is considered a necessity because it optimizes data quality and ensures proper functioning. It also brings great advantages such as savings in costs and time invested in error correction.

It is worth noting that the costs associated with poor data quality can be very high. According to recent research, the average cost of poor data quality increases over the years. These costs include customer loss, reduced business efficiency, lost business opportunities, among others.

In an attempt to solve these problems, this document presents a software product that can be used to obtain measures regarding data quality control. The developed tool simplifies the process of data quality assessment by means of a web application to which files can be uploaded for analysis, streamlining the process and providing instant user friendly feedback about the files.

RESUMEN

En la actualidad, el volumen de datos que se maneja en las empresas es cada vez mayor. La tecnología y la digitalización han generado una enorme cantidad de información que, aunque es muy valiosa, es difícil de gestionar y controlar. Por ello, resulta indispensable gestionar y mantener unos niveles de calidad de datos determinados.

La gestión adecuada de los datos es considerada una tarea de primera necesidad, ya que esto permite optimizar su calidad y garantizar su correcto funcionamiento. Además, también trae consigo grandes ventajas, como el ahorro de costes y tiempos invertidos en la corrección de errores.

Es importante destacar que los costes asociados a la mala calidad de datos pueden ser muy elevados. Según investigaciones recientes, el coste medio de la mala calidad de datos aumenta con los años. Estos costes incluyen pérdida de clientes, reducción de la eficiencia empresarial, pérdida de oportunidades de negocio, entre otros.

Con el objetivo de intentar subsanar estos problemas, en este trabajo se presenta una herramienta de control de calidad de datos, donde se podrán detectar varios parámetros que podrían dictaminar una calidad aceptable de los datos reforzada con una aplicación web donde poder introducir los archivos a analizar y ver los resultados del análisis más fácilmente.

ÍNDICE

ABSTRACT	3
RESUMEN	4
ÍNDICE	5
ÍNDICE DE TABLAS	7
ÍNDICE DE ILUSTRACIONES	9
1. INTRODUCCIÓN	10
1.1 Contexto y motivación	10
1.2 Objetivos	11
1.3 Estructura del documento	12
2. ESTADO DEL ARTE	14
2.1 Tecnologías aplicables al modelo de calidad de datos	14
2.1.1 Pandas	14
2.1.2 Openpyxl	17
2.1.3 Apache POI	18
2.1.4 ExcelDataReader	19
2.1.5 Comparativa	21
2.2 Tecnologías aplicables al desarrollo de una aplicación web	22
2.2.1 Django	23
2.2.2 Spring Framework	25
2.2.3 Express	26
2.2.4 Laravel	27
2.3 Bases de datos	28
2.3.1 Bases de datos relacionales	28
2.3.1.1 MySQL	30
2.3.1.2 SQLite	31
2.3.2 Bases de datos no relacionales	32
2.3.2.1 MongoDB	33
2.4 Conclusiones del estado del arte	34
3. ANÁLISIS DEL PROBLEMA	37
3.1 Alcance del proyecto	37
3.1.1 Casos de uso	38
3.1.2 Requisitos	40
3.1.2.1 Requisitos de usuario	42
3.1.2.2 Requisitos de sistema	44
3.2 Implementación	49
Paso 1. Inicio de sesión/Registro	53
Paso 2. Introducir un fichero a analizar	55
Paso 3. Ejecutar el programa de análisis de datos	57

Paso 4. Visualización de los resultados obtenidos	59
4. EVALUACIÓN	62
5. PLANIFICACIÓN DEL PROYECTO	66
5.1 Metodologías para el desarrollo de proyectos	66
5.2 Planificación inicial	67
5.3 Planificación real	68
6. Presupuesto del proyecto	70
6.1 Recursos software	70
6.2 Recursos Hardware	71
6.3 Recursos humanos	71
6.4 Presupuestos indirectos	71
6.5 Resumen de presupuesto	72
7. Impacto socioeconómico	73
8. Marco regulador	74
8.1 Leyes de protección de datos	74
8.2 Licencias	75
9. Conclusiones	75
9.1 Retrospectiva	76
9.2 Trabajo futuro	77
9.3 Conclusiones personales	77
10. Summary	80
10.1 Introduction and objectives	80
10.2 State of the Art	80
10.2.1 Technologies Applicable to the Data Quality Model	81
10.2.2 Technologies Applicable to Web Application Development	82
10.2.3 Databases	83
10.2.3.1 Relational Databases	83
10.2.3.2 Non-Relational Databases (NoSQL)	84
10.2.4 Conclusions from the State of the Art	84
10.3 Implementation	85
10.4 Evaluation	86
10.5 Project Planning	88
10.5.1 Methodologies for project development	88
10.5.2 Initial and actual planning	89
10.6 Project budget	90
10.7 Socioeconomic Impact	90
10.8 Regulatory Framework	91
10.8.1 Data Protection Laws	91
10.8.2 Licenses	92
10.9 Conclusions	93
Referencias / Bibliografía	95

ÍNDICE DE TABLAS

Tabla 1 Comparativa tecnologías modelo calidad	22
Tabla 2 Formato caso de uso	39
Tabla 3 CU-01	39
Tabla 4 CU-02	40
Tabla 5 CU-03	40
Tabla 6 Formato requisitos	41
Tabla 7 RU-01	42
Tabla 8 RU-02	42
Tabla 9 RU-03	42
Tabla 10 RU-04	43
Tabla 11 RU-05	43
Tabla 12 RU-06	43
Tabla 13 RU-07	44
Tabla 14 RU-08	44
Tabla 15 RS-01	45
Tabla 16 RS-02	45
Tabla 17 RS-03	45
Tabla 18 RS-04	46
Tabla 19 RS-05	46
Tabla 20 RS-06	46
Tabla 21 RS-07	47
Tabla 22 RS-08	47
Tabla 23 RS-09	47
Tabla 24 RS-10	48
Tabla 25 RS-11	48
Tabla 26 RS-12	49
Tabla 27 Formato tabla pruebas	62
Tabla 28 P-01	63
Tabla 29 P-02	63
Tabla 30 P-03	63
Tabla 31 P-04	64

Tabla 32 P-05	64
Tabla 33 P-06	64
Tabla 34 Matriz de trazabilidad entre requisitos y pruebas	65
Tabla 35 Planificación inicial	66
Tabla 36 Planificación real	68
Tabla 37 Presupuesto recursos software	69
Tabla 38 Presupuesto recursos humanos	70
Tabla 39 Presupuestos indirectos	71
Tabla 40 Resumen de presupuesto	71

ÍNDICE DE ILUSTRACIONES

Ilustración 1 dataframe creation example	15
Ilustración 2 Conexiones persistentes	25
Ilustración 3 MySQL Growth History	31
Ilustración 4 Official Drivers for MongoDB	33
Ilustración 5 Casos de uso	38
Ilustración 6 Directorio raíz del proyecto	50
Ilustración 7 Página administradora de Django	52
Ilustración 8 Página principal index.html	53
Ilustración 9 Formulario de registro	54
Ilustración 10 Encabezado de welcome.html	55
Ilustración 11 Formulario para introducir el fichero a analizar	55
Ilustración 12 Vista previa del fichero a analizar	56
Ilustración 13 Página de resultados finales	60
Ilustración 14 Diagrama de Gantt (Planificación inicial)	67
Ilustración 15 Diagrama de Gantt (Planificación real)	68

1. INTRODUCCIÓN

1.1 Contexto y motivación

La importancia de los datos manejados y recabados por empresas, negocios u organizaciones ha crecido enormemente en los últimos años [1]. Esto se debe a que toda esa información almacenada puede llegar a ser un recurso muy valioso para la toma de decisiones.

Para que la información pueda ser explotada es necesario que los datos que se crean y analizan dentro de una organización tengan un orden o un sentido que permita su almacenamiento y análisis. Asimismo, es importante detectar cuándo los datos no aportan valor para poder descartarlos.

Es por esto que se requiere que las organizaciones, públicas o privadas, se enfoquen en producir información de alta calidad para poder explotarla de forma más eficiente, puesto que subsanar los errores a posteriori genera plantear soluciones específicas a cada caso que suponen un coste. Por ejemplo, IBM calcula que los datos erróneos cuestan a la economía estadounidense 3.1 billones de dólares al año [2].

Como la cuantía de información y la calidad de los datos almacenados no siempre van de la mano, es necesario tratar de asegurar la calidad de los datos en los procesos previos a su explotación.

Un modelo de calidad de datos es una herramienta utilizada para evaluar y mejorar la calidad de los datos en un sistema u organización. La calidad de los datos en este caso pretende cuantificar que sean exactos, integrales, consistentes y completos.

En este Trabajo Fin de Grado se presenta un software que permite automatizar la evaluación de la calidad de datos de ficheros provistos por el usuario a través de una aplicación web.

1.2 Objetivos

Este proyecto tiene dos objetivos principales claramente diferenciados, además de varios objetivos secundarios que se expondrán a continuación:

1. Desarrollar un software capaz de analizar un fichero para poder determinar su nivel de calidad a través de unos parámetros previamente establecidos. Los principales objetivos de este programa son los siguientes:

- El programa deberá poder leer archivos de tipo Excel o CSV.
- Deberá tener la capacidad de aplicar funciones de análisis a los datos previamente extraídos del archivo provisto con el fin de determinar su nivel de calidad.

2. Desarrollar una página web para facilitar la ejecución del programa de análisis de la calidad de los datos y la visualización de los resultados del mismo. Sus objetivos principales son los siguientes:

- Realizar una verificación inicial de los usuarios.
- Capacidad de conectar un programa externo y enlazarlo con la aplicación web de forma que los usuarios puedan ejecutar el programa de análisis de forma sencilla a través de una GUI.
- Presentar los resultados del control de calidad en una página, que incluya datos detallados, características específicas, cálculo global y gráficos y diagramas diversos para hacer la visualización de los resultados más clara y amena.

Como objetivos secundarios, y por los cuales he escogido este Trabajo Final de Grado encontramos la práctica y perfeccionamiento de tres cosas que considero importantes y muy útiles en el sector informático:

- La oportunidad de poder trabajar y perfeccionar conocimientos acerca de tecnologías de creación de páginas web.
- La manipulación de archivos y estructuras de datos de tipo Excel, CSV, etc. Estos son algunos de los formatos de archivo más utilizados actualmente, por

lo que su modificación y análisis es de gran utilidad, especialmente de cara a su uso posterior en aplicaciones de data science.

- La integración de sistemas y la puesta en práctica de diferentes formas de programación como lo pueden ser un desarrollo web y un programa de calidad de datos en un mismo proyecto, la perfecta coordinación de ambos y la gran utilidad de poder unir varios tipos de tecnologías.

1.3 Estructura del documento

Para un mejor análisis del documento, se presentan y describen los 9 apartados que contiene.

Capítulo 1 - Introducción: inicialmente se hace una pequeña introducción donde se aclaran el contexto y la motivación que ha llevado al autor del documento a elegir ese tema y los objetivos a conseguir mediante la realización del proyecto.

Capítulo 2 - Estado del arte: este apartado está dedicado a analizar y comparar diversas tecnologías útiles para la realización de la aplicación, tanto de la parte de desarrollo web como del programa externo de análisis de datos. Finalmente se hace un resumen de las tecnologías elegidas y se da explicación del motivo de su elección.

Capítulo 3 - Análisis del problema: esta sección va dedicada a realizar un análisis de los casos de uso y requisitos de la aplicación. Finalmente se explica el funcionamiento de la misma.

Capítulo 4 - Evaluación: parte fundamental del proyecto dedicada a la realización de test con el fin de comprobar el correcto funcionamiento de la aplicación.

Capítulo 5 - Planificación del proyecto: se indica la planificación inicial del proyecto en cuestión de días, así como la planificación real final empleada.

Capítulo 6 - Entorno socioeconómico: sección dedicada al cálculo presupuestario en el caso de sacar la aplicación al mercado, además de un análisis del impacto socioeconómico que tendría si saliese.

Capítulo 7 - Marco regulador: la primera parte de esta sección hace referencia a las leyes de protección de datos que intervienen y se deben cumplir en el proyecto. La segunda parte analiza las diferentes licencias necesarias para llevar a cabo correctamente el proyecto.

Capítulo 8 - Conclusiones: en este apartado se ha decidido establecer una división en tres subapartados: el primero hace referencia a la retrospectiva que se ha tenido desde el inicio del proyecto hasta su finalización, el segundo se detallan las funcionalidades pensadas a futuro que no han podido ser implementadas hasta la fecha y en tercer lugar se detallan las conclusiones personales que el autor ha extraído del proyecto.

Capítulo 9 -: Summary: para finalizar el documento se proporciona un resumen del mismo en inglés.

2. ESTADO DEL ARTE

En esta sección se analizan, investigan y describen las diferentes herramientas y tecnologías que pueden ser utilizadas para realizar de forma óptima y útil el proyecto de este Trabajo Final de Grado.

Ya que este proyecto se divide en dos partes diferenciadas y para ellas se utilizan diferentes tecnologías, separaremos este apartado en dos subapartados, aunque se incluirá un tercero en el que aparecerán diferentes tecnologías de almacenamiento en bases de datos también útiles para un proyecto con estas características.

2.1 Tecnologías aplicables al modelo de calidad de datos

En la mayoría de los casos, las empresas utilizan unas estructuras de datos específicas para almacenar y procesar los datos, ya que suelen ser más sencillas y manejables. Estas se suelen guardar en archivos de tipo Excel, CSV, archivos de texto comunes, archivos XML o de tipo JSON, entre otros.

Por este motivo las tecnologías enumeradas en esta sección deberán poder acceder a este tipo de archivos en específico y permitir procesar y analizar los de los mismos para extraer conclusiones sobre la calidad de datos.

Posteriormente a una búsqueda exhaustiva de tecnologías compatibles y óptimas para llevar a cabo esta parte del proyecto, llevaremos a cabo un estudio de todas ellas.

2.1.1 Pandas

“Pandas es un paquete de Python que proporciona datos rápidos, flexibles y expresivos. Se basa en estructuras diseñadas para hacer que trabajar con datos "relacionales" o "etiquetados" sea fácil e intuitivo. Su objetivo es ser el componente fundamental de alto nivel para haciendo análisis de datos prácticos del mundo real en Python” [3]

Como podemos ver en la definición que nos proporciona la página web pypi.org, ampliando con otras definiciones, Pandas es una librería de Python que fundamentalmente nos permite trabajar con dataframes, series y paneles de datos. Con Pandas, es posible leer un archivo Excel y convertirlo en un dataframe para realizar análisis y poder manipular los datos. Además, ofrece funciones para filtrar, ordenar, unir y agrupar datos, lo que la convierte en una herramienta muy potente.

Según la página W3Schools, la definición de dataframe es la siguiente: “Un dataframe es una estructura de datos de 2 dimensiones, como un array bidimensional, o una tabla con filas y columnas.” [4]. En la Ilustración 1 se observa un ejemplo de inicialización de un dataframe con Pandas para crear una mejor idea de su significado ya que es una parte muy importante de esta tecnología.

```
import pandas as pd

data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

Ilustración 1 dataframe creation example [5]

Como se puede observar, un dataframe tiene un alto parecido con una matriz bidimensional, donde cada posición es un dato de cualquier tipo. Dada su estructura, los dataframes son muy útiles para almacenar datos provenientes de archivos de tipo Excel, JSON o XML ya que en este tipo de archivos la estructura de los datos es similar. Entre las ventajas que proporciona el uso de dataframes encontramos [6]:

- **Estructura tabular:** esta propiedad hace referencia a lo explicado anteriormente, refiriéndonos a la similitud que tiene con una base de datos o con tablas de datos.
- **Heterogeneidad de datos:** los dataframes permiten incluir diferentes tipos de datos en sus columnas. Por ejemplo, las columnas pueden contener datos como tipos numéricos, cadenas de texto, fechas, tipo char, etcétera. Esto los hace más flexibles a la hora de manejar diferentes tipos de datos a la vez.
- **Etiquetado de columnas y filas:** cada columna y cada fila pueden tener un nombre asociado, lo que facilita la identificación de las mismas.
- **Acceso y manipulación de datos:** internamente ofrecen funciones y métodos propios para acceder y manipular datos de manera diferente, como filtrado, selección, agregación, combinación y transformación.
- **Integración con herramientas de análisis de datos:** normalmente son más compatibles con una diferentes tipos de bibliotecas y herramientas de análisis de datos.
- **Tratamiento de datos faltantes:** los datos nulos pueden ser manipulados, al igual que los valores flotantes. También proporcionan métodos para identificarlos fácilmente, lo que ayuda en gran medida al análisis de datos.
- **Operaciones vectorizadas:** permiten aplicar una operación a una columna entera en lugar de que sea necesario hacer la misma operación dato a dato.

Una de las principales ventajas de Pandas es su capacidad de soportar una gran variedad de formatos, pudiendo transformar los datos de todos ellos en estructuras de datos específicas. Como tipos de archivos compatibles tenemos CSV, Excel, SQL, JSON, HTML, HDF5, Parquet, Feather, Msgpack, Stata, SAS, Google BigQuery.

Con Pandas, es posible leer un archivo Excel y convertirlo en un dataframe para realizar análisis y manipulación de datos. También ofrece una gran cantidad de funciones para trabajar con series de tiempo y datos financieros. Debido a ello, es posible realizar análisis de series de tiempo y calcular estadísticas financieras, lo que amplía aún más sus ventajas.

2.1.2 Openpyxl

Openpyxl [7] es una librería de Python que permite leer y escribir archivos Excel en formato xlsx. Con Openpyxl, es posible leer los datos de un archivo Excel y convertirlos en una estructura de datos de Python, como una lista o un diccionario. Además, Openpyxl ofrece funciones para trabajar con hojas de cálculo, celdas y fórmulas, lo que la convierte en una herramienta muy versátil.

Openpyxl es muy similar a la librería Pandas, siendo ambas librerías de Python destinadas a trabajar con archivos excel. A continuación, para facilitar la comprensión de Openpyxl debido al enorme parecido con pandas, se enumeran algunas de las características que diferencian ambas tecnologías:

- **Tipos de archivo compatibles:** Pandas puede trabajar con una gran variedad de tipos de archivo, como CSV, Excel, SQL, entre otros, sin embargo Openpyxl está más enfocado a el formato de excel Office Open XML (.xlsx)
- **Estructuras de datos:** Pandas utiliza los dataframes como estructuras de datos, mientras que Openpyxl modifica directamente el archivo Excel, no definiendo una estructura de datos propia sobre la que trabajar.
- **Funcionalidades:** Pandas ofrece una amplia gama de funciones para realizar operaciones sobre los dataframes mientras que Openpyxl, debido a que se centra principalmente en la lectura y escritura de datos en las hojas de Excel, no tiene funciones ni métodos para realizar operaciones sobre los datos.
- **Nivel de abstracción:** Pandas utiliza una interfaz más sencilla ocultando muchos detalles de bajo nivel, mientras que Openpyxl proporciona una capa de bajo nivel, lo que puede requerir un mayor conocimiento para interactuar con su interfaz.

En resumen, mientras que Pandas está más enfocado a recoger datos de diferentes tipos de archivos, introducirlos en dataframes y modificarlos con sus operaciones ya incluidas en la interfaz, Openpyxl está más orientado a modificar directamente los archivos de datos de entrada.

Por otro lado, como similitudes tenemos las siguientes características:

- **Soporte para archivos Excel:** ambas librerías trabajan y pueden leer archivos Excel, aunque cada una lo haga a su manera.
- **Interoperabilidad:** ambas permiten intercambiar datos entre Excel y Python.
- **Manipulación de datos:** ambas permiten manipular datos de un archivo Excel, Pandas a través de dataframes y Openpyxl en el propio archivo, pero ambas tienen capacidad de manipular datos.

2.1.3 Apache POI

La definición de Apache POI según riptutorial es la siguiente: “Apache POI Project es una API de Java para manipular varios formatos de archivo basados en los estándares Office Open XML (OOXML) y el formato de documento compuesto de OLE 2 de Microsoft (OLE2). En resumen, puede leer y escribir archivos de MS Excel, Word y Powerpoint utilizando Java.” [8].

Apache POI permite leer los datos de un archivo Excel y convertirlos en una estructura de datos de Java, como un array o una lista. Como propiedades principales destacan las siguientes:

- Es **compatible con múltiples sistemas operativos**, lo que le da la propiedad de ser una herramienta multiplataforma.
- **Soporta una variedad de formatos de Microsoft específicos**, en concreto Microsoft Word (.doc o .docx), Microsoft Excel (.xls o .xlsx) y Microsoft PowerPoint (.ppt o .pptx).
- La **manipulación de los datos es similar a la de Openpyxl**, permite leer, escribir y modificar datos en el propio archivo. Tiene una parte positiva frente a Openpyxl, y es que Apache POI permite crear archivos vacíos desde cero, aunque su aspecto negativo frente a Pandas es que no permite transmitir los datos a ninguna estructura como los dataframes y por lo tanto, es más costoso poder analizar y realizar operaciones con los datos.

- Apache POI proporciona diversas **funcionalidades avanzadas**, como aplicar formatos, estilos, fórmulas y gráficos en las hojas de cálculo de Excel o insertar imágenes y tablas en los documentos Word entre otras cosas.
- Posee una **amplia comunidad y soporte**. Esta herramienta es muy popular entre la comunidad de desarrollo de Java, por lo que se destina una gran cantidad de recursos, foros y ejemplos para obtener ayuda y soporte aparte de intentar actualizar y renovar la herramienta en el menor lapso de tiempo posible para adaptar la herramienta a nuevas versiones de Office, mejorar el rendimiento, corregir errores entre otros.

Como resumen de la herramienta y a recalcar tanto sus partes positivas como negativas, tenemos lo siguiente:

- Apache POI es una librería de Java, algo no demasiado eficiente ya que dicho lenguaje está decayendo un poco en la actualidad [9], aunque sí es muy utilizada por la comunidad, lo que hace que tenga actualizaciones constantes y pueda mantenerse y perdurar en el tiempo.
- No almacena datos, si no que modifica archivos ya existentes y crea archivos nuevos. Con el objetivo de realizar un análisis de datos no es del todo productivo, pero dada su funcionalidad extra de poder incluir gráficos, fuentes y otros elementos de office de manera interna, hace que se pueda considerar seriamente para cierto tipo de proyectos.
- Por último, en cuestión de accesibilidad, si bien es cierto que es multiplataforma y puede ser utilizada en múltiples sistemas operativos, su productividad baja enormemente al tener una escasez alta de archivos con posibilidad de lectura.

2.1.4 ExcelDataReader

“ExcelDataReader es una API ligera de código abierto escrita en C# para leer archivos de Microsoft Excel.” [10].

Como variante a las herramientas mencionadas anteriormente tenemos ExcelDataReader, la única de las cuatro que se utiliza en C#. En comparación con las otras herramientas se puede deducir que tiene similitud con Pandas, siendo muy similar a ella con cambios sutiles. Para

poder visualizar mejor de qué trata y cómo funciona, a continuación se mostrarán sus características, tanto positivas como negativas:

- **Compatibilidad con Excel:** en el aspecto de la compatibilidad vemos más parecido a Openpyxl, siendo Microsoft Excel el único tipo de archivo con el que es compatible. Como punto fuerte se debe recalcar que es compatible con muchos de los formatos que tiene Excel, así como muchas de sus versiones.
- **No requiere tener Excel instalado:** al no utilizar dependencias externas, no requiere tener Microsoft Excel instalado en el sistema para funcionar, lo que simplifica enormemente su uso.
- **Alto rendimiento:** la librería está optimizada para poder manejar un gran conjunto de datos, lo que la hace ideal para manejar archivos muy grandes aparte de poder manejar también un conjunto elevado de archivos simultáneamente.
- **Fácil usabilidad:** la biblioteca posee una interfaz simple e intuitiva, permitiendo a los desarrolladores leer y procesar la información de las bases de datos de manera más rápida y sencilla.
- **Capacidad de lectura:** a diferencia de otras herramientas, ExcelDataReader únicamente posee capacidad de lectura de los archivos Excel, no pudiendo ni crearlos ni modificarlos.
- **Carencia de una estructura de datos interna:** los datos extraídos se pueden almacenar en estructuras ya definidas por el lenguaje de programación utilizado, como listas o matrices, pero no posee una estructura de datos como podrían ser los dataframes en el caso de la librería Pandas para almacenar y modificar datos eficientemente.
- **Soporte multiplataforma:** tiene la capacidad de poder ser utilizado con diferentes aplicaciones así como con diferentes tipos de sistemas operativos, incluyendo Linux, Windows y macOS.

Como se puede observar, ExcelDataReader es una biblioteca poderosa y versátil para trabajar con archivos Excel en C#. Su punto fuerte es su capacidad para trabajar con archivos pesados y grandes, por lo que si hiciésemos un análisis de datos en archivos grandes podría ser de gran utilidad. Sin embargo, no posee una estructura de datos interna, por lo que esta

herramienta podría estar más destinada a recoger, recopilar y procesar una gran cantidad de datos simultáneamente sin un grado de exhaustividad excesivo .

2.1.5 Comparativa

Dado que las herramientas que hay expuestas anteriormente son similares entre ellas se puede realizar una comparativa con el fin de analizar las ventajas y desventajas de cada uno.

En la siguiente tabla aparece reflejada una comparativa de todas las herramientas mencionadas anteriormente, tomando como parámetros de la comparación la compatibilidad de archivos que posee cada herramienta, los permisos de lectura y escritura que tienen cada uno, el tipo de almacenamiento interno de los datos una vez extraídos, el lenguaje nativo de cada librería y por último sus puntos fuertes y débiles.

Todos estos parámetros han sido elegidos por la posible comparación entre ellos, ya que pueden tener diferencias sustanciales en esos ámbitos o porque en el campo del análisis de datos es de vital importancia, por ejemplo, saber cómo se guardan los datos, y es que no es lo mismo guardarlos en un dataframe con mayor capacidad para procesamiento y manipulación que en una lista o un diccionario con posibilidades más reducidas.

	Pandas	Openpyxl	Apache POI	ExcelDataReader
Compatibilidad	Múltiples compatibilidades	Excel	Office	Excel
Escritura/lectura	Lectura sobre los archivos, no escritura	Lectura y escritura en archivos	Lectura y escritura en archivos	Lectura sobre los archivos, no escritura
Almacenamiento	Dataframes	Listas, diccionarios, etc	Listas, diccionarios, etc	Listas, diccionarios, etc
Lenguaje	Python	Python	Java	C#
Puntos fuertes	<ul style="list-style-type: none"> - Capacidad de guardar los datos en dataframes - Operaciones vectorizadas - Compatibilidad - Operaciones internas - Heterogeneidad de los datos 	<ul style="list-style-type: none"> - Interfaz sencilla y rápida - Capacidad de escritura en los archivos 	<ul style="list-style-type: none"> - Capacidad de escritura en los archivos - Permite crear archivos desde cero - Amplia comunidad - Funcionalidades avanzadas 	<ul style="list-style-type: none"> - No requiere instalación de Excel - Interfaz simple e intuitiva
Puntos débiles	<ul style="list-style-type: none"> - No tiene capacidad de escritura en los archivos 	<ul style="list-style-type: none"> - Carece de estructura de datos propia - No tiene operaciones internas - Compatibilidad reducida 	<ul style="list-style-type: none"> - Carece de estructura de datos propia - Compatibilidad reducida 	<ul style="list-style-type: none"> - Compatibilidad reducida - Carece de estructura de datos propia

Tabla 1 Comparativa tecnologías modelo calidad

Para la elección del estudio de tecnologías para el desarrollo del programa de calidad de datos se he hecho una comparativa entre tecnologías de diferentes lenguajes de programación y distintas características con la misma funcionalidad, poder trabajar con archivos tipo Excel. Cada uno tiene ventajas específicas que ayudarán dependiendo del tipo de proyecto que estemos desarrollando, pero concretamente según nuestros requerimientos, al final del apartado se expondrá un resumen de la elección escogida.

2.2 Tecnologías aplicables al desarrollo de una aplicación web

La segunda parte de la que consta este proyecto es en el desarrollo de una página web completamente operativa que permita introducir un archivo en ella y, en consecuencia, pueda

derivar dicho archivo a nuestro programa análisis de calidad de los datos para que lo analice, procese y valore. Finalmente deberá ser capaz de arrojar los resultados acerca de la calidad de datos obtenida por el programa externo.

2.2.1 Django

Primeramente y para entender de qué trata este framework, debemos explicar que Django [11] es un framework web de alto nivel que permite el desarrollo rápido de sitios web. Debido a que ha sido desarrollado por programadores con experiencia en el sector web, Django se encarga de una gran parte de las complicaciones y procedimientos iniciales del desarrollo, proporcionando herramientas para completar rápidamente tareas tediosas y mecánicas que habría que hacer manualmente en caso de que no existiera, lo que ayuda y facilita en gran medida a todos aquellos desarrolladores que la utilicen al desarrollo interno de su página web. Es gratuito y de código abierto, posee una comunidad activa y dedicada, una extensa documentación y muchas opciones de soporte tanto gratuito como de pago [12].

Sus características han hecho que Django sea uno de los frameworks más utilizados para la creación de páginas web. Dichas características son las siguientes:

- **Completo:** provee diversas herramientas internas que facilitan enormemente el desarrollo la página web, aparte de poseer una amplia y actualizada documentación que facilita el uso de dichas herramientas.
- **Versátil:** la variedad de páginas web que puede construir Django es muy diverso, desde wikis, hasta redes sociales y sitios de noticias. Puede devolver contenido en casi cualquier formato y puede funcionar con cualquier framework en el lado del cliente.
- **Seguro:** Django proporciona diversas herramientas para asegurar la seguridad. Un ejemplo de ello es que proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando errores de inicio de sesión o cookies dónde podría llegar a ser vulnerable. También posee protección contra algunas vulnerabilidades de manera predeterminada, entre las que se incluyen la inyección SQL, scripts entre diferentes sitios o falsificación de solicitudes.
- **Escalable:** utiliza una arquitectura basada en “shared-nothing”. Esto significa que cada parte de la arquitectura es independiente de las otras, por tanto si alguna no

funciona o hay algún problema con ella puede ser reemplazada o cambiada si es necesario sin una modificación severa de las demás piezas.

- **Mantenible:** está configurado para que el código sea lo más mantenible y reutilizable posible. Posee herramientas para que no exista duplicación innecesaria de código aparte de agrupar código relacionado en módulos siguiendo el patrón MVC (Model View Controller). Este modelo expresa cómo organizar y estructurar los componentes de un sistema software y las relaciones existentes entre cada uno de ellos [13].
- **Portable:** dado que está escrito en Python, se ejecuta en diversas plataformas, aparte de poderse ejecutar en múltiples distribuciones de Windows, Linux y MAC OS.
- **Patrón MVT (Model-View-Template):** es una variante originaria de Django del patrón MVC mencionado anteriormente, que funciona como veremos a continuación. La arquitectura de la aplicación se divide en tres partes: el modelo, la vista y los templates. El modelo es la parte de la aplicación que maneja los datos y la lógica, igual que ocurre en la arquitectura MVC. La vista se encarga de recibir las solicitudes de los usuarios, realizar las operaciones necesarias en el modelo y devolver una respuesta adecuada. Por último, la plantilla se encarga de la presentación visual de los datos de la aplicación dividida en diferentes páginas html [14].

Entre las funcionalidades que se pueden implementar fácilmente con Django se encuentran los formularios, autenticación y permisos de usuarios, cacheo, sitio de administración desde la que poder crear, editar y visualizar cualquiera de las partes involucradas de su sitio y la serialización de los datos para poder servir tus datos como XML o JSON, entre otras.

En cuanto a las bases de datos que soporta nos encontramos las siguientes: PostgreSQL, MariaDB, MySQL, Oráculo y SQLite. Para la gestión del framework con la base de datos, Django presenta una propiedad fundamental que permite optimizar la velocidad y la utilidad de las conexiones.

La propiedad de la que hablamos se basa en las conexiones persistentes. Esto quiere decir que una vez que se establece una conexión con la base de datos, un parámetro define la vida útil máxima de la conexión y la mantiene abierta. Si queremos volvernos a conectar, no se requiere iniciar una nueva conexión con la penalización de tiempo que esto conlleva, si no

que utiliza la creada anteriormente para conectarse a la misma base de datos más fácilmente siempre y cuando esté en el periodo de tiempo útil establecido previamente, periodo de vida que además puede configurarse en la aplicación [15].

Persistent connections

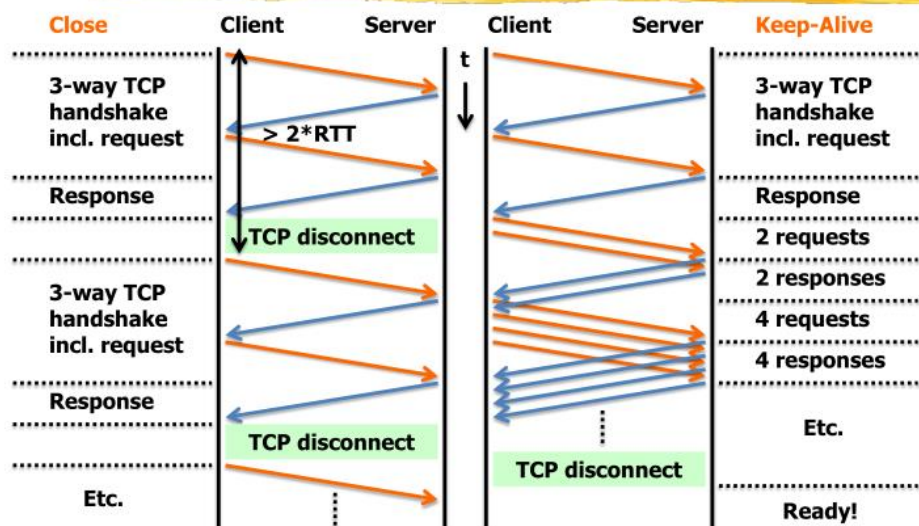


Ilustración 2 Conexiones persistentes [16]

Django ofrece más ventajas en la optimización de código y facilidad de creación de la página. Varios ejemplos de ello podría ser el uso de plantillas, que contienen las partes estáticas de la salida HTML así como alguna sintaxis especial para insertar el contenido dinámico [17] o el uso de formularios, con los que Django proporciona una gama de bibliotecas y herramientas para su creación [18].

2.2.2 Spring Framework

Spring Framework [19] es uno de los framework más populares que utiliza el lenguaje Java, y es ampliamente utilizado para construir aplicaciones robustas y escalables. Posee una alta versatilidad que proporciona a los desarrolladores un conjunto rico de características y módulos, lo que hace que el proceso de construir aplicaciones sea significativamente más fácil. Además, Spring ofrece una variedad de otras características y módulos clave que

pueden mejorar significativamente la funcionalidad de sus aplicaciones. Algunas de las siguientes características son las siguientes:

- **Programación orientada a aspectos (AOP):** esto implica dividir la lógica del programa en diferentes partes llamadas preocupaciones. Estas preocupaciones consisten en conceptos como registro, auditoría, transacciones declarativas, seguridad, almacenamiento en caché, etc. Conceptualmente todos estos conceptos están separados de la lógica empresarial de la aplicación, lo que facilita el mantenimiento y la reutilización de código [20].
- **Inyección de dependencias (DI):** patrón de desarrollo de software donde los objetos no son los encargados de inicializar sus dependencias, sino que estas son provistas a través de otro objeto, como consecuencia mejora la modularidad y la flexibilidad del código [21].
- **Integración con bases de datos:** Spring proporciona soporte para mayor facilidad de integración con las bases de datos mediante diferentes tecnologías como pueden ser JDBC, JPA y otros módulos relacionados.
- **Spring Boot:** dentro de Spring Framework podemos encontrar una de sus principales herramientas, denominada Spring Boot. Esta herramienta se encarga de acelerar y simplificar el desarrollo y la creación de microservicios y aplicaciones web gracias a tres funciones principales: una configuración inicial automática, un menú de configuración optimizado y la capacidad de crear aplicaciones autónomas [22], lo que permite crear aplicaciones con menos configuración y esfuerzo.

2.2.3 Express

Para explicar en qué consiste Express [23] debemos empezar explicando qué es Node.js. Node, cómo lo llamaremos a partir de ahora como modo de simplificación, es un entorno de código abierto y multiplataforma que trabaja en tiempo de ejecución y que permite a los programadores crear toda clase de herramientas de lado del servidor empleando JavaScript como lenguaje de programación [24]. Si bien sabemos que node es un entorno de código abierto, Express es el framework web más popular de Node [25]. Como características principales podemos destacar las siguientes:

- **Minimalismo:** al ser un framework minimalista, hace que sea más fácil y flexible a la hora de usarlo que cualquier otro framework de Node. También proporciona un excelente sistema de enrutamiento, middlewares y negociación de contenidos.
- **Escalabilidad:** debido a su popularidad, este framework ha demostrado ser muy escalable a lo largo del tiempo. No requiere apenas configuración inicial y maneja peticiones y respuestas de los usuarios de forma eficiente.
- **Potente sistema de enrutamiento:** posee el sistema de enrutamiento más potente y robusto incorporado por defecto, lo que es extremadamente útil para gestionar la estructura de la aplicación, agrupando las diferentes rutas en una única carpeta.
- **Middlewares:** comprende una serie de middlewares incorporados con los que los desarrolladores pueden introducir scripts para interceptar el flujo de la aplicación. Un middleware es un código que se ejecuta previamente a que una petición HTTP llegue al manejador de rutas, lo que permite ejecutar un script antes o después de la petición de un cliente.

2.2.4 Laravel

Laravel [26] es un framework multiplataforma basado en PHP [27] creado por Taylor Otwell en 2011. Fue desarrollado con el objetivo de hacer el desarrollo web en PHP más rápido y eficiente. Desde su creación, Laravel ha ganado una gran cantidad de seguidores en la comunidad de desarrollo web, convirtiéndose en uno de los frameworks más utilizados y queridos disponibles.

Laravel está diseñado principalmente para el desarrollo de back-end, aunque también ofrece algunas herramientas útiles para el desarrollo del front-end, aunque muchas de sus características son independientes del front-end. El lenguaje utilizado es un lenguaje de scripting en lugar de un lenguaje PHP. Pese a que los lenguajes de scripting y los de programación tienen relación, se pueden ver diferencias en la facilidad de uso y velocidad de ejecución. Para poder comparar mejor este framework con los mencionados anteriormente también incluiremos sus principales características:

- **Aprendizaje sencillo:** dado su tipo de lenguaje, es necesario menor conocimiento avanzado para comprender su funcionamiento comparado con otros frameworks.
- **Gestión de rutas:** facilita la creación y el mantenimiento de rutas en aplicaciones web mediante el uso de nombres sencillos e identificadores de ruta.
- **Seguridad:** ofrece una serie de funciones de seguridad, incluyendo autenticación de usuarios, autorización de roles, verificación de correo electrónico, encriptación, hashing de contraseñas y restablecimiento de contraseñas.
- **Fácilmente escalable:** es muy flexible y puede crecer fácilmente, cuenta con funciones incorporadas para almacenar temporalmente datos de manera rápida y distribuida. También incluye una plataforma de implementación sin necesidad de servidores llamada Vapor, la cual se basa en AWS y proporciona mucha escalabilidad.
- **Progresivo:** incluye características para usuarios de todos los niveles. Los principiantes pueden acceder a kits de inicio y los usuarios más experimentados pueden aprovechar dichos kits para construir sus propias implementaciones.
- **Amplio ecosistema y comunidad:** la biblioteca de aplicaciones y paquetes disponibles es amplia, tanto los paquetes oficiales proporcionados por los desarrolladores del framework como los de otros usuarios que lo utilizan.

2.3 Bases de datos

En cuanto a bases de datos podemos distinguir dos tipos más comunes que otros: las bases de datos relacionales (SQL) y las bases de datos no relacionales (NoSQL). Dedicaremos dos subapartados para cada uno de los tipos incluyendo su descripción, características y las tecnologías más populares que se puede utilizar con ambos tipos de bases de datos, incluyendo una breve descripción de las mismas.

2.3.1 Bases de datos relacionales

Las bases de datos relacionales [28], también conocidas como bases de datos SQL (Structured Query Language), son uno de los tipos de bases de datos más tradicionales y ampliamente utilizados. En este tipo de bases de datos, la información se organiza en tablas con filas y columnas, y se establecen relaciones entre tablas utilizando claves primarias y claves externas. Los analistas utilizan consultas SQL para combinar diferentes puntos de

datos y resumir el rendimiento empresarial. Las bases de datos relacionales están comúnmente relacionadas con las bases de datos transaccionales que ejecutan comandos o transacciones de forma colectiva. Estas bases de datos están caracterizadas por unas propiedades denominadas por sus iniciales como ACID [29] que deben cumplir todas sus transacciones.

- **Atomicidad:** las modificaciones en los datos se realizan de forma parecida a una sola operación, o se cambian todos los datos o no se cambia ninguno.
- **Consistencia:** los datos permanecen en un estado consistente al pasar de un estado a otro, lo que involucra un refuerzo en la integridad de los datos.
- **Aislamiento:** el estado intermedio de una transacción no es visible para el resto de transacciones, por lo tanto aquellas transacciones que se ejecuten simultáneamente están serializadas.
- **Durabilidad:** los cambios en los datos se vuelven persistentes, es decir, permanentes, después de completar con éxito una transacción.

Si bien su propio nombre lo indica, las bases de datos relacionales son aquellas que pueden establecer relaciones entre sus tablas, lo que le da múltiples ventajas como reducir la redundancia, una mayor facilidad de copia de seguridad y recuperación de datos y una mayor facilidad de uso y empleo.

El lenguaje utilizado para llevar a cabo operaciones en la base de datos es SQL, un lenguaje declarativo que permite filtrar, agregar, eliminar, actualizar y buscar datos de manera eficiente. Este lenguaje se inventó en la década de 1970 [30] basado en el modelo relacional y se convirtió en el sistema comercial de administración de bases de datos relacionales. Primeramente la empresa IBM fue la pionera en esta tecnología y le otorgó el nombre de SEQUEL, aunque en seguida evolucionó principalmente gracias a una compañía llamada Relational Software, que posteriormente pasó a llamarse como actualmente la conocemos, Oracle [31].

2.3.1.1 MySQL

El software MySQL proporciona un servidor de base de datos SQL multi-threaded, multi usuario y robusto, diseñado para entornos de producción críticos, con una alta carga de trabajo y para poder integrarse en un software con el fin de ser distribuido [32]. Fue creado primeramente por la empresa MySQL AB, posteriormente adquirida por MicroSystems en 2008 y finalmente pasó a posesión de Oracle Corporation en 2010 [33]. Cuenta con una doble licencia, por un lado es de código abierto, es decir que su código fuente está abierto públicamente, pero por otro lado cuenta con una versión más comercial gestionada por la compañía Oracle.

Las ventajas que ofrece este software son las siguientes:

- Podemos recalcar en primer lugar que sea de **código abierto**, ya que es fácilmente accesible por los desarrolladores.
- Su **arquitectura cliente-servidor** es de gran utilidad ya que los clientes se conectan con los servidores de manera aislada para un mejor rendimiento.
- Su **compatibilidad con SQL**, el lenguaje más popular referente a las bases de datos relacionales ofrece un desarrollo más simple y eficaz.
- Posee la opción de la **creación de vistas** a partir de la versión 5.0, esto es fundamental para las bases de datos de gran tamaño ya que podemos simplificar su visualización y comprensión enormemente.
- MySQL **no procesa las tablas directamente**, sino que a través de algunos procedimientos almacenados internamente tiene la capacidad de incrementar la eficacia de su implementación. Además, tiene automatizadas ciertas tareas dentro de la base de datos, por lo que ciertas acciones como la actualización al lanzar un nuevo evento o eliminarlo se hace más sencilla.

En la siguiente imagen se muestra la evolución que MySQL ha tenido a lo largo de los años.

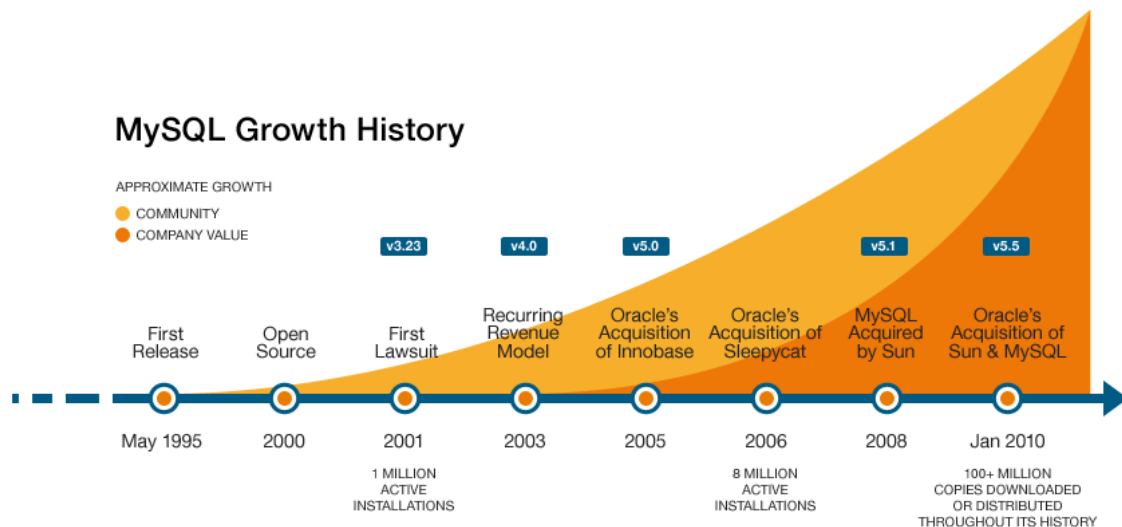


Ilustración 3 MySQL Growth History [34]

2.3.1.2 SQLite

SQLite [35] es un sistema de gestión de bases de datos relacionales muy ligero, escrito en C y de código abierto. Funciona como un servidor externo independiente, incluyendo tanto una base de datos como su propio motor internamente [36]. Entre las principales ventajas de SQLite se encuentran su soporte para múltiples tablas, índices, triggers y vistas, lo que permite a los desarrolladores trabajar con una amplia variedad de datos y estructuras.

Otra ventaja importante de SQLite es su capacidad para leer y escribir directamente sobre archivos que se encuentran en el disco duro. Esto quiere decir que los datos almacenados en una base de datos SQLite están disponibles para cualquier aplicación que tenga acceso al archivo en el que se almacenan los datos [37]. Además, el formato de la base de datos es multiplataforma y se puede utilizar el mismo archivo en un sistema tanto de 32 como de 64 bits.

A diferencia con los sistemas de administración de bases de datos que utilizan el modelo cliente-servidor, SQLite no se concibe como un proceso autónomo, en su lugar, se comunica con el programa y se convierte en una parte integral del mismo. Para que esto funcione, emplea llamadas a funciones bloqueando el archivo al inicio de cada transacción que reducen el tiempo de acceso a la base de datos, así como el tiempo de ejecución de cada acción [38].

En términos de velocidad, SQLite es muy eficiente al realizar operaciones de SQL y es más rápido que las otras opciones previamente vistas. Además, cuenta con diversas interfaces API, lo que permite trabajar con diferentes tecnologías como Python, C++, PHP o Groovy. También es totalmente auto contenida, lo que quiere decir que no posee dependencias externas. Por último, cuenta con librerías para muchos lenguajes de programación, lo que facilita su uso.

Aunque SQLite presenta varias ventajas, también tiene algunas desventajas importantes. Por ejemplo, presenta limitaciones en la cláusula “Where” debido al soporte para clausuras anidadas. Además, cuando se crea la tabla desde el modo consola es incapaz de establecer claves foráneas.

No obstante, en su tercera versión SQLite ha mejorado y ha incorporado nuevas características que lo hacen aún más poderoso. Por ejemplo, ahora soporta bases de datos de hasta 2 terabytes de tamaño y permite la inclusión de campos tipo BLOB [39].

2.3.2 Bases de datos no relacionales

Las bases de datos no relacionales son un sistema de almacenamiento de información caracterizado por no emplear el lenguaje SQL al contrario de las relacionales. Tampoco cumplen el estándar ACID, que es la propiedad definida en el apartado anterior característico de las bases de datos relacionales [40].

Una de sus principales características es que no trabajan con estructuras de datos definidas, lo que quiere decir que los datos no se almacenan en tablas, sino en documentos. Poseen una gran escalabilidad y están destinadas a la gestión de grandes volúmenes de datos.

Son de gran utilidad para bases de datos de gran tamaño o para organizar bases de datos ya creadas pero mal estructuradas internamente debido a los motivos explicados anteriormente y además teniendo en cuenta que son más flexibles a la hora de crear esquemas de información que los relacionales, garantizan un gran rendimiento y ya que cuentan con APIs exclusivas son muy funcionales [41].

Teniendo en cuenta que las ventajas de las bases de datos relacionales son superiores, sólo en determinados casos son recomendables utilizar este tipo de bases de datos, por lo general suele ser más eficiente no utilizarlas.

2.3.2.1 MongoDB

Este sistema de gestión de bases de datos NoSQL fue desarrollado por MongoDB Inc en el año 2007 por Dwight Merriman, Eliot Horowitz y Kevin Ryan, quienes anteriormente pertenecían a la empresa de publicidad en internet DoubleClick [42].

Una de las mayores sorpresas al entrar en la página oficial de MongoDB es la documentación para todos los posibles lenguajes de programación que acepta. Aparte, podemos encontrar toda la documentación necesaria en dicha página, sin tener que recurrir a páginas de terceros o tener que fiarse de fuentes externas.

Official Drivers

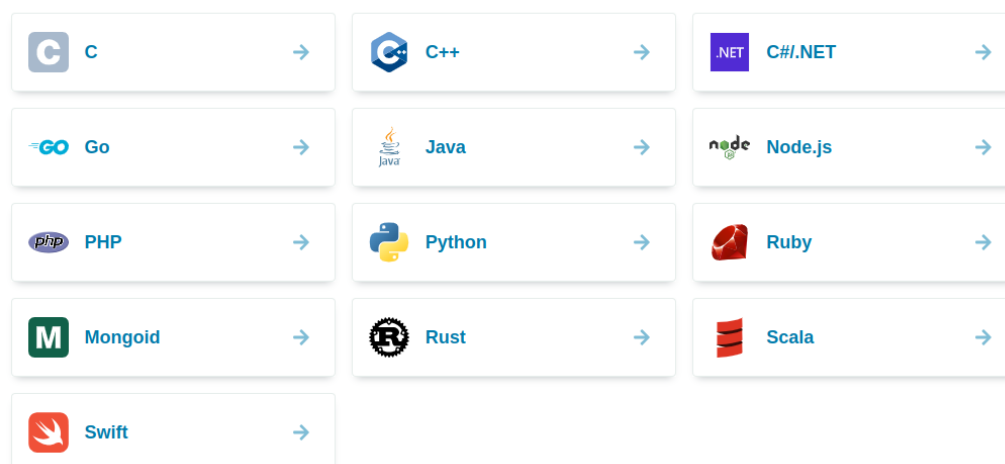


Ilustración 4 Official Drivers for MongoDB [43]

MongoDB es una base de datos que funciona almacenando documentos con una estructura de pares de campos y valores. Dichos documentos en MongoDB tienen ventajas como su capacidad de corresponder con tipos de datos nativos en muchos lenguajes de programación y reducir la necesidad de uniones costosas debido a sus documentos integrados y arreglos de discos. Esos documentos los almacena a su vez en colecciones, las cuales son una analogía de las tablas en las bases de datos relacionales.

Además, MongoDB proporciona persistencia en sus datos, alto rendimiento debido a que la compatibilidad con modelos de datos integrados reduce la actividad de E/S, los índices admiten una velocidad mayor de consultas y provee alta disponibilidad mediante redundancia de datos en conjuntos de réplicas. Parte de su funcionalidad principal es la escalabilidad horizontal, la cual permite la creación de zonas de datos para equilibrar las lecturas y escrituras. También admite múltiples motores de almacenamiento y proporciona una API de almacenamiento conectable para desarrolladores de terceros [44].

2.4 Conclusiones del estado del arte

Una vez analizadas las posibles tecnologías que permiten el análisis de calidad de datos de diferentes formatos de archivos junto con los diferentes frameworks y aplicaciones que permiten desarrollar una aplicación web; unido con plataformas de gestión de bases de datos que se contengan en ambas divisiones podemos llegar a varias conclusiones:

- Para la apertura de diferentes tipos de archivos y el almacenamiento de sus datos en estructuras productivas con el fin de analizarlas es buena opción hacer uso de la **librería Pandas**. Las razones para elegir esta tecnología son las siguientes:
 - La cantidad de formatos que soporta es mucho más amplio que los de las demás tecnologías, lo que permite mayor variedad de tipos de archivos que podemos analizar.
 - Los dataframes proporcionan mayor facilidad para el análisis de datos, teniendo en cuenta que otras tecnologías no tienen ningún tipo de formato para el almacenamiento de datos interno, lo que dificulta su manipulación.

- Pandas posee diferentes tipos de operaciones internas útiles para poder manejar y manipular datos específicos, lo que es de gran utilidad para establecer más parámetros de calidad para mostrar.
- Para este proyecto en concreto necesitamos muchas de las propiedades mencionadas anteriormente. Las otras tecnologías tienen otras propiedades más orientadas a modificar los archivos de entrada, crear archivos nuevos, etc. Debido a todas estas características se ha elegido utilizar la librería Pandas en el proyecto.
- En cuanto a la aplicación que se destinará para elaborar la aplicación web y que más ventajas posee para nuestro proyecto se ha elegido el framework **Django**. Esta elección ha sido debida a las siguientes características:
 - Posee medidas preventivas que mejoran y refuerzan aspectos relacionados con la seguridad, la sencillez y la simplicidad, como el método de autenticación de usuarios o la arquitectura basada en “shared-nothing” explicada anteriormente en las características de este framework.
 - Su documentación es de buena calidad, está actualizada y es de fácil acceso.
 - El patrón MVT, propio de Django, permite que el código se vea reducido pudiendo obviar ciertas partes. Los templates ayudan considerablemente a la creación y modificación de la parte visual de la página, proporcionando también métodos internos para facilitar la creación.
 - También contamos con las conexiones persistentes, propiedad de la que carecen algunos frameworks también investigados.
 - Aunque haya propiedades de Django que no se estén utilizando actualmente en el proyecto, el motivo por elegir este framework va con visión de futuro, ya que en una posible ampliación de la aplicación podrían ser útiles.
- El lenguaje de programación también ha influido en la toma de decisiones. Se ha buscado en la medidas de lo posible que todas las tecnologías estuvieran basadas en el mismo lenguaje de programación. Tanto Pandas como Django están escritas en **Python**, lo que refuerza la elección de ambas tecnologías.
- En caso de necesitar una aplicación para la gestión de bases de datos es conveniente utilizar **MySQL**, aplicación de bases de datos relacionales ya que nuestra base de datos no sería extremadamente grande para emplear una base de datos no relacional.

Django es compatible con este tipo de tecnología, por lo que actualmente y en vistas a futuro podría influenciar a favor en la creación y ampliación de la aplicación.

3. ANÁLISIS DEL PROBLEMA

En este apartado se describe el desarrollo y las funcionalidades de la solución propuesta al principal problema anteriormente planteado. Se proporcionan casos de uso, requisitos y una explicación detallada tanto de la aplicación web como del programa externo.

3.1 Alcance del proyecto

El desarrollo planteado comprende tres partes claramente diferenciadas y explicadas a continuación:

- La **creación de una página web** operativa localmente con autenticación de usuario donde se podrá introducir el archivo que queramos analizar para determinar su calidad.
- El **desarrollo de un programa** que permita analizar diferentes archivos en los formatos más comunes utilizados actualmente y detectar ciertos patrones de calidad previamente establecidos.
- La **visualización de los resultados** obtenidos en forma estadística a través de distintos desplegados que muestran los resultados de cada parámetro de calidad establecido, con la ampliación de gráficos y otras herramientas visuales, para que los usuarios puedan valorar su nivel de calidad a través de la interfaz web.

Esta herramienta está orientada a la gran mayoría de empresas, ya que actualmente los datos forman parte indispensable de toda empresa y el control de calidad de sus datos se ha convertido en una parte fundamental y necesaria de controlar.

Asimismo, el programa podría ser utilizado por empresas de otros sectores e incluso por pequeñas empresas. La importancia de garantizar la calidad de los datos es importante en todos los contextos en los que los datos resulten de utilidad.

3.1.1 Casos de uso

Los casos de uso son una herramienta esencial para cualquier proyecto ya que permite describir las interacciones entre los usuarios u otros sistemas externos y el sistema. En la ilustración 5, podemos ver reflejado el funcionamiento del programa desde un punto de vista externo haciendo referencia y utilizando los casos de uso que se describen a continuación.

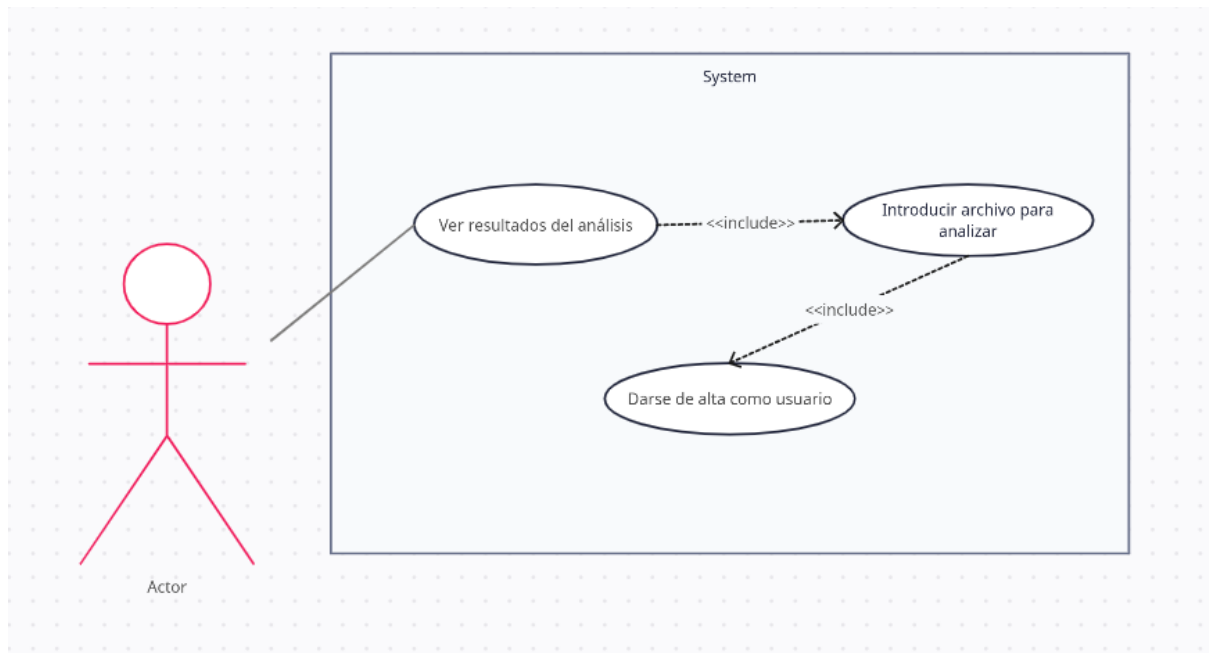


Ilustración 5 Casos de uso

El objetivo de la ilustración es poder comprender el funcionamiento del sistema, en el que el usuario primeramente se da de alta en la aplicación web, posteriormente introduce un archivo que analizar, el programa externo analiza ese archivo y obtiene un resultado de calidad y por último, los resultados de esta ejecución se transfieren a la página web para mostrarlos en una forma optimizada y adecuada, que llega finalmente al usuario.

La siguiente tabla representa el formato a utilizar para representar los casos de uso:

ID	
Título	
Descripción	
Precondiciones	
Postcondiciones	

Tabla 2 Formato caso de uso

- ID: cada caso de uso poseerá un identificador único e intransferible siguiendo el formato “CU-XX” donde “XX” es el número asociado.
- Título: un breve título que resume el significado del caso de uso en cuestión.
- Actores: simboliza quien debe ejecutar la acción, en nuestro caso hay dos posibilidades: que sea el usuario o el sistema externo
- Descripción: no muy extensa descripción explicando más detalladamente el motivo del caso de uso.
- Precondiciones: condiciones previas necesarias para poder llevar a cabo su realización.
- Postcondiciones: condiciones posteriores a la realización del caso de uso.

ID	CU-01
Título	Darse de alta como usuario / Iniciar sesión
Descripción	En primera instancia el usuario deberá ingresar en la página web y acceder al registro dándose así de alta como nuevo usuario. En caso de que el usuario ya esté dado de alta, deberá iniciar sesión con su nombre de usuario y contraseña.
Precondiciones	Ninguna.
Postcondiciones	El usuario se da de alta en el sistema y se guardan sus credenciales en la base de datos.

Tabla 3 CU-01

ID	CU-02
Título	Introducir archivo para analizar
Descripción	El usuario podrá introducir el archivo que desea analizar en el entregador que aparecerá en el apartado “Análisis” de la aplicación web para iniciar su análisis.
Precondiciones	El usuario ha iniciado sesión previamente y tiene un archivo en el formato correcto para analizar.
Postcondiciones	El archivo pasa a manos del programa externo para ser analizado.

Tabla 4 CU-02

ID	CU-03
Título	Ver resultados del análisis
Descripción	La página web mostrará los resultados obtenidos por el programa externo al usuario mediante una serie de elementos visuales que facilitarán la comprensión de los mismos.
Precondiciones	El programa se ha ejecutado correctamente y ha arrojado unos resultados válidos que ha enviado a la página web.
Postcondiciones	La página web arroja los resultados de la forma más visual posible y el usuario tiene acceso a ellos.

Tabla 5 CU-03

3.1.2 Requisitos

Primeramente para que se entienda realmente el significado de un requisito, podemos definirlo como [45]:

- Una condición o capacidad que un usuario necesita.
- Una condición o capacidad que debe poseer un sistema.
- Una representación documentada de una condición o capacidad.

Una vez explicada la definición de requisito, para este apartado debemos distinguir entre requisitos de usuario y requisitos de sistema. Por un lado, los requisitos de usuario tienen como finalidad que el propio usuario intervenga en la condición o capacidad que procedemos a describir, al igual que los de sistema que tiene la misma finalidad interviniendo el propio sistema.

Las tablas tendrán el mismo formato para ambas, a excepción del campo “RU Involucrados”, que solo estará disponible en los requisitos del sistema. El formato de las tablas para ambos tipos de requisitos será el siguiente:

ID	
Nombre	
Descripción	
Prioridad	
Estabilidad	
RU Involucrados	

Tabla 6 Formato requisitos

- **ID:** cada requisito debe tener un identificador único. Ya que tenemos dos tipos diferentes de requisitos, cada uno se mostrará de la siguiente manera:
 - Los requisitos de usuario seguirán el formato “RU-XX”, donde “XX” es el número identificador del requisito de usuario.
 - Los requisitos de sistema seguirán el formato “RS-XX”, donde “XX” es el número identificador del requisito de sistema.
- **Nombre:** todo requisito deberá tener un nombre claro y conciso que lo identifique.
- **Descripción:** breve descripción explicando más detalladamente la funcionalidad del requisito.
- **Prioridad:** nivel de preferencia del requisito en el entorno del proyecto. Puede tomar valores “superior”, “promedio” o “inferior”.
- **Estabilidad:** establece la capacidad de persistencia del requisito. Puede tomar valores “superior”, “promedio” o “inferior”.
- **RU Involucrados:** en caso de los requisitos de sistema, indica los identificadores de los requisitos de usuario que están involucrados en cada caso.

3.1.2.1 Requisitos de usuario

En el siguiente apartado se listan los requisitos de usuario que posee el proyecto con el formato definido previamente.

ID	RU-01
Nombre	Navegación
Descripción	Los usuarios podrán acceder a las funcionalidades del proyecto mediante la navegación a través de una interfaz web.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input type="checkbox"/> Superior <input type="checkbox"/> Promedio <input checked="" type="checkbox"/> Inferior

Tabla 7 RU-01

ID	RU-02
Nombre	Registro
Descripción	Los usuarios deben ser capaces de registrarse en la aplicación rellendo un formulario con sus datos.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior

Tabla 8 RU-02

ID	RU-03
Nombre	Inicio de sesión
Descripción	Los usuarios registrados deben poder iniciar sesión en la aplicación introduciendo correctamente sus credenciales.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior

Tabla 9 RU-03

ID	RU-04
Nombre	Cierre de sesión
Descripción	Los usuarios registrados deben poder cerrar sesión de manera que no se pueda acceder al análisis de archivos.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior

Tabla 10 RU-04

ID	RU-05
Nombre	Cargar archivo
Descripción	Los usuarios autenticados deben tener la opción de cargar archivos tipo Excel y CSV para su análisis.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input type="checkbox"/> Superior <input checked="" type="checkbox"/> Promedio <input type="checkbox"/> Inferior

Tabla 11 RU-05

ID	RU-06
Nombre	Ver resultados
Descripción	Los usuarios deben poder acceder a una sección donde puedan ver los resultados de los análisis del fichero introducido.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior

Tabla 12 RU-06

ID	RU-07
Nombre	Configuración del archivo
Descripción	Los usuarios podrán configurar cómo desean que el programa analice el fichero, por ejemplo con la elección de una hoja concreta del mismo.
Prioridad	<input type="checkbox"/> Superior <input checked="" type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input type="checkbox"/> Superior <input checked="" type="checkbox"/> Promedio <input type="checkbox"/> Inferior

Tabla 13 RU-07

ID	RU-08
Nombre	Ayuda al programa externo
Descripción	Los usuarios podrán ayudar al análisis del fichero seleccionando parámetros subjetivos que el programa no puede detectar por sí mismo.
Prioridad	<input type="checkbox"/> Superior <input type="checkbox"/> Promedio <input checked="" type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior

Tabla 14 RU-08

3.1.2.2 Requisitos de sistema

En el siguiente apartado aparecerán listados los requisitos de sistema existentes en el proyecto con el formato definido previamente, es decir que se incluirá el apartado RU Involucrados.

ID	RS-01
Nombre	Procesamiento de archivos
Descripción	El sistema debe ser capaz de aceptar archivos tipo Excel y CSV como entrada para su análisis.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-02, RU-03, RU-05, RU-07

Tabla 15 RS-01

ID	RS-02
Nombre	Almacenar variables importantes
Descripción	El sistema debe almacenar y procesar algunas variables como la separación en caso de CSV, el nombre específico de la página del fichero o las respuestas a preguntas en la página vista previa.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input type="checkbox"/> Superior <input checked="" type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-02, RU-03, RU-07, RU-08

Tabla 16 RS-02

ID	RS-03
Nombre	Conectividad de ambas tecnologías
Descripción	El sistema deberá poder conectar la aplicación web y el programa externo una vez introducimos el archivo para poder analizarlo, y posteriormente recoger los resultados obtenidos.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-05, RU-06, RU-07, RU-08

Tabla 17 RS-03

ID	RS-04
Nombre	Analizar de calidad de datos
Descripción	El sistema deberá poder leer y analizar la calidad de los datos del archivo que el usuario ha introducido en la página web.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-02, RU-03, RU-05, RU-07, RU-08

Tabla 18 RS-04

ID	RS-05
Nombre	Almacenamiento de datos
Descripción	Tanto los resultados del análisis como la gestión de usuarios deberán almacenarse y comprobarse en una base de datos garantizando la seguridad e integridad de los datos.
Prioridad	<input type="checkbox"/> Superior <input checked="" type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-05

Tabla 19 RS-05

ID	RS-06
Nombre	Interfaz de usuario
Descripción	El sistema deberá proporcionar una interfaz gráfica que facilite iniciar el análisis de un archivo de entrada y su posterior visualización.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input type="checkbox"/> Superior <input checked="" type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-02, RU-03, RU-04, RU-05, RU-06, RU-07, RU-08

Tabla 20 RS-06

ID	RS-07
Nombre	Mostrar vista previa
Descripción	El sistema deberá mostrar una vista previa del documento antes de iniciar su análisis a través de la interfaz web y tras que el usuario haya indicado el archivo que va a analizar.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-02, RU-03, RU-05, RU-06

Tabla 21 RS-07

ID	RS-08
Nombre	Permitir/Denegar análisis de archivos
Descripción	El sistema deberá permitir el análisis de los archivos sólo a aquellos usuarios que haya iniciado sesión.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input type="checkbox"/> Superior <input checked="" type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-02, RU-03, RU-05, RU-06, RU-07, RU-08

Tabla 22 RS-08

ID	RS-09
Nombre	Iniciar sesión
Descripción	El sistema deberá iniciar sesión una vez que se haya introducido correctamente el usuario y la contraseña.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-03

Tabla 23 RS-09

ID	RS-10
Nombre	Registro
Descripción	El sistema deberá recoger los datos del formulario introducidos por el usuario, agregarlos a la base de datos e iniciar sesión en la aplicación web.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-02

Tabla 24 RS-10

ID	RS-11
Nombre	Cerrar sesión
Descripción	El sistema deberá cerrar sesión en la aplicación cuando se presione el botón de “Cerrar sesión”
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
RU Involucrados	RU-01, RU-04

Tabla 25 RS-11

ID	RS-12
Nombre	Mostrar los resultados
Descripción	El sistema deberá mostrar los resultados del análisis de un archivo proporcionado por los usuarios a través de la interfaz web.
Prioridad	<input checked="" type="checkbox"/> Superior <input type="checkbox"/> Promedio <input type="checkbox"/> Inferior
Estabilidad	<input type="checkbox"/> Superior <input type="checkbox"/> Promedio <input checked="" type="checkbox"/> Inferior
RU Involucrados	RU-06

Tabla 26 RS-12

3.2 Implementación

Este apartado tiene la finalidad de explicar el funcionamiento del proyecto en general, tanto de la aplicación web como del programa externo, cómo se comunican entre ellos y qué criterios se han tenido en cuenta para su creación.

Primeramente para la creación de la aplicación web como hemos mencionado antes se ha utilizado el framework Django. Esta aplicación se ejecuta en un servidor HTTP desplegado de forma local con el comando “python3 manage.py runserver”, siendo manage.py un archivo local del proyecto destinado a la activación del servidor. Por defecto utiliza la URL local <http://127.0.0.1:8000/>, donde se podría modificar el puerto o la dirección a placer dentro de la configuración de la aplicación.

La arquitectura que se ha seguido para implementar la aplicación web ha sido el modelo MVT. Como hemos explicado anteriormente este modelo consiste en dividir en tres partes la lógica de la aplicación: el **modelo**, el cual tiene una importancia ínfima en el proyecto, ya que no llegamos a generar ningún objeto que pueda ser representado como modelo, por lo tanto está vacío. Después pasamos a la **vista**, que en este caso es la parte que más desarrollo ha tenido en esta aplicación web. En ella podemos encontrar métodos creados para cumplir con la funcionalidad de cada página, ya sea para iniciar sesión, registrarse, redirigir a otra página si accionamos un botón, recoger la información del archivo que hemos introducido, etcétera.

Básicamente cualquier acción que realice la aplicación web está configurada aquí. Por último están las **templates**, que se refiere al conjunto de páginas html que conforman la aplicación. En las templates se proporciona el valor estético de la aplicación.

El programa externo de control de calidad de datos realmente se encuentra en la misma carpeta donde podemos encontrar las vistas para mayor comodidad a la hora de comunicarse y de encontrar el programa.

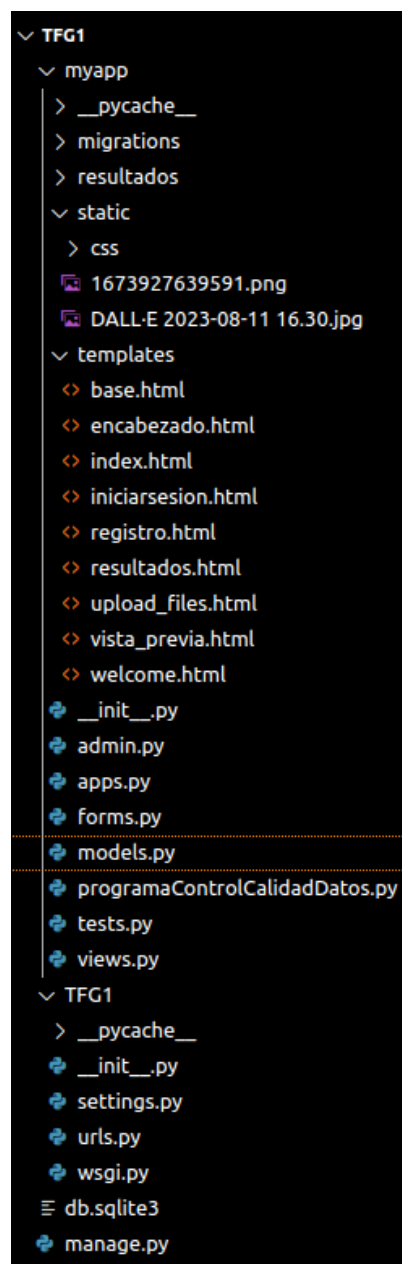


Ilustración 6 Directorio raíz del proyecto

La Ilustración 6 muestra la estructura de directorios del proyecto. Como podemos observar tenemos dos partes claramente diferenciadas, la carpeta *myapp* donde se encuentra la mayoría de código creado por el autor de la aplicación y la carpeta “TFG1”, generada automáticamente cuando se creó el proyecto Django con todos los archivos de configuración. Para entender mejor la organización de la ilustración se van a explicar la funcionalidad de los archivos:

- **Carpeta resultados:** esta carpeta está destinada a guardar un histórico de resultados, funcionalidad no implementada aún pero con vista a futuro.
- **Carpeta templates:** como se explicó anteriormente, los templates conforman la visualización de cada página independiente. Tanto `base.html` como `encabezado.html` se refieren a los menús de navegación que se encuentran en la página, siendo diferentes dependiendo de si se ha iniciado sesión o no. La diferencia entre `welcome.html` e `index.html` radica en si el usuario inicia sesión o no, nuevamente. Si la sesión está iniciada se dirigirá a `welcome.html`, de lo contrario a `index.html`.
- **admin.py:** fichero destinado a la gestión de la propiedad de administrador de Django [46]. Aquí podemos introducir lo que queremos que se encuentre en la página de administración de Django, la cual ampliaremos su definición más adelante.
- **apps.py:** este fichero se utiliza para definir el nombre de la aplicación o su configuración, entre otras.
- **forms.py:** aquí se definen los formularios personalizados en Django. Se introducen los campos, validaciones y más elementos de los formularios que queramos personalizar.
- **models.py:** fichero donde se definen los distintos modelos de la aplicación. Un modelo es una clase que define la estructura de una tabla en la base de datos [47]
- **tests.py:** en este fichero se generan test unitarios para modelos, vistas o formularios específicos.
- **views.py:** como se ha explicado anteriormente, fichero donde se encuentran las vistas del modelo MVT.
- **settings.py:** archivo de configuración del proyecto.
- **urls.py:** lugar donde se podrían encontrar las distintas urls de las páginas independientes creadas.
- **wsgi.py:** fichero de configuración de la aplicación web con los servidores web.

Antes de comenzar a explicar los pasos a seguir por el usuario que quiera utilizar la aplicación hay que mencionar el sitio de administración de Django, únicamente accesible por el administrador de la aplicación.

Este sitio de administración se puede encontrar creando un usuario administrador con el comando “python3 manage.py createsuperuser” e introduciendo “/admin” al final de la url de nuestro servidor web, siempre y cuando se haya iniciado sesión con ese usuario administrador. Una vez ingresado en esta url, la página de administración de Django nos proporciona en forma de interfaz web una manera sencilla e intuitiva de administrar diversas funciones del propio back-end, como la creación o eliminación de parámetros, usuarios, modelos, organización de los mismos y acciones similares.

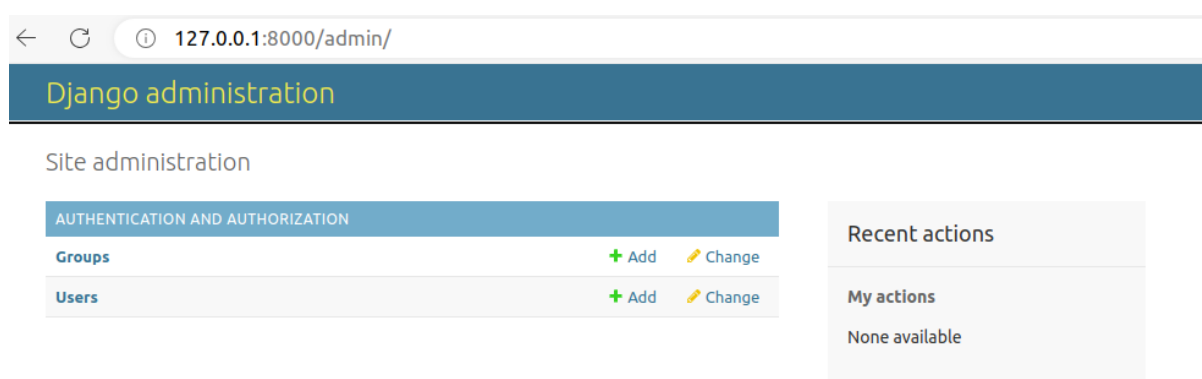


Ilustración 7 Página administradora de Django

En el caso de nuestro proyecto solo podemos administrar esos dos campos, ya que al no tener ningún modelo implementado no aparecen en dicha página. Sin embargo, si decidimos crear algún modelo en concreto, aparecería a continuación del apartado de usuarios para poder administrarlo de igual modo.

A continuación se procederá a explicar paso a paso cómo tendría que hacer un usuario para analizar un archivo aleatorio y poder ver los resultados que arroja.

Paso 1. Inicio de sesión/Registro

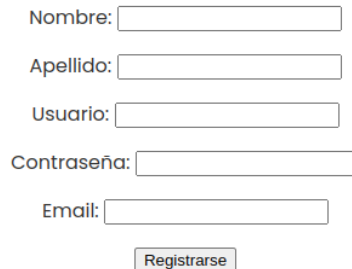
Inicialmente nos encontramos en la página principal una imagen introductoria e intuitiva sobre la aplicación web, con un encabezado en el que existen tres opciones: Inicio, Registro e Iniciar sesión. A la izquierda del encabezado está situado el logo de la empresa, quedando según la siguiente ilustración:



Ilustración 8 Página principal index.html

Aquí podemos observar que la página no nos deja realizar ninguna funcionalidad sin registrarnos, por lo que la primera acción necesaria si nunca hemos ingresado en la aplicación web es registrarse. Por medio de un formulario la aplicación nos pide que ingresemos nombre, apellido, usuario, email y contraseña para poder registrarnos. El formulario de registro es el siguiente:

Página de registro



Nombre:

Apellido:

Usuario:

Contraseña:

Email:

Ilustración 9 Formulario de registro

Una vez introducidos los datos del registro, Django se encarga de introducir los datos del usuario en la base de datos que hemos enlazado a través del comando interno `"User objects create_user"`. Si los datos son aceptados por el programa se redirigirá automáticamente a la página principal una vez que hemos iniciado sesión. Se refiere con que los datos sean aceptados a que el mismo nombre de usuario no se haya registrado previamente y a que la dirección de correo electrónico esté en formato correcto. Si alguno de estos parámetros falla saltará un mensaje de error hasta que nos registremos correctamente.

Si el proceso del registro ha salido correctamente nos veremos redirigidos a la página de inicio cuando se ha iniciado sesión, `"welcome.html"`. Esta página es muy parecida a la página de `index.html`, sus principales diferencias son la aparición de nuestro nombre de usuario en pantalla, junto a la variación del encabezado. Aquí la opción de registro se cambia por la de análisis y la de inicio de sesión por cierre de sesión, permitiéndonos así poder comenzar con el análisis de los ficheros.

El funcionamiento del inicio de sesión es muy similar, requiriendo únicamente el usuario y la contraseña. Dicha página está formada por un formulario con ambos campos, que una vez introducidos y comprobados también nos llevará a la página `welcome.html`.



Bienvenido, Dani

Ilustración 10 Encabezado de welcome.html

Paso 2. Introducir un fichero a analizar

Para este paso es necesario acceder a la página de análisis, formada por un entregable de archivos donde necesariamente debemos incluir únicamente aquellos tipos de archivos aceptados por el programa (inicialmente los tipos son Excel y CSV, pendientes a ampliación futura), una caja de texto que permite introducir una página específica del Excel/CSV que se quiera analizar y por último un seleccionable destinado a los archivos CSV para introducir la clase de separador que tiene, pudiendo elegir entre cuatro opciones: coma, punto y coma, espacio o tabulación.

Analisis de archivos

Introduce tu archivo aquí

datos_aleatorios.csv

Introduce el nombre de la hoja en caso de que quieras elegir una en específico

En caso de CSV, ¿que se utiliza como separación?

☐ ";"

☒ ","

☐ tab

☐ " "

Ilustración 11 Formulario para introducir el fichero a analizar

Realizado este proceso, el programa transformará el archivo introducido en un dataframe. Esto se hace principalmente para poder mostrar la vista previa de la siguiente página en la cadena. También genera un archivo Excel con la información del dataframe para el posterior análisis del programa externo.

Una vez realizados todos estos pasos, nos mostrará una página donde podemos ver una vista previa del archivo para comprobar que todo ha salido correctamente. Dicha página tiene posibilidad e idea de incluir algunas cuestiones que ayudarán a mejorar el análisis del documento, pero actualmente dados los parámetros que se muestran en los resultados no es necesario.

El principal motivo de incluir preguntas en esta parte de la aplicación es la de mejorar el análisis del documento, ya que por ejemplo se puede ocurrir un caso concreto de conflicto en cuanto a determinar las columnas o celdas numéricas, como ha ocurrido ya en algunos análisis, debido que el criterio para diferenciar un número de algo que no lo es a veces es muy subjetivo. Inicialmente, para subsanar este error mi planteamiento ha sido declarar como número aquel dato que pueda ser convertido a float. Hay muchas más posibilidades que se podrían llegar a estudiar en un futuro, pero esta manera suele acertar la mayoría de las veces, con lo que la considero, al menos, acertada. Para casos futuros responder estas cuestiones determinará una mayor precisión en el análisis y una menor tasa de error.

Vista previa de archivo Excel

	Nombre	Dirección	Fecha de nacimiento	Trabajo	Teléfono	Email	Código	País	Ciudad	Número
0	Begoña Ayuso	Paseo Dolores Uriarte 73 Puerta 4 \nTarragona, 66186	Radiographer, therapeutic	+34 221 421 729	sribas@carvajal.com	DQORJaH3iD	Rumania	Málaga	43	NaN
1	Josefina Martínez Uabrés	Ronda Cristian Pinedo 43 Apt. 67 \nMurcia, 55387	Designer, textile	+34846 62 42 95	wcamps@marti.com	kgxDWTqIFI	Mali	Melilla	19	NaN
2	Samuel Zabala- Valverde	Paseo de Aitor Ripoll 51\nNavarra, 02979	Development worker, international aid	+34784 335 668	palaciosdolores@gmail.com	YcJIGNWslUI	Tuvalu	Tarragona	47	NaN
3	Santiago Vicente- Montoya	Calle Lorenzo Roda 54\nBadajoz, 38689	IT sales professional	+34616 446 121	domingovicens@hotmail.com	WMj3qCWy7N	Nicaragua	Asturias	60	NaN
4	Hector Vigil Tejedor	Avenida Cesar Iriarte 938\nAlmería, 56880	Clinical scientist, histocompatibility and immunogenetics	+34 926 750 395	jose-maria75@aznar- andrade.com	ZINKEIB6Jh	Argelia	Zamora	60	NaN

Para mayor precisión en el análisis del archivo conteste a estas preguntas.

¿Hay alguna columna que contenga valores calculables?

En caso de haberlo, introduzca las posiciones de las columnas con valores calculables

El formato debe ser el índice de cada columna teniendo en cuenta que la primera se corresponde al 0, con comas como separación. Por ejemplo "0,1,4,8" !

Inicio de página

Fin de página

En la ilustración 12 podemos observar un ejemplo de la vista previa que arroja un fichero CSV aleatorio. También aparecen como ejemplo las preguntas antes mencionadas, de momento inhabilitadas.

Paso 3. Ejecutar el programa de análisis de datos

Ya en la página de la vista previa, oprimiendo el botón de carga, la aplicación se encarga de generar un archivo de tipo Excel guardado localmente con los datos introducidos para que el programa externo pueda localizar dicho fichero y analizarlo. Este método es provisional, ya que para un solo usuario en un mismo periodo de tiempo funciona, pero en vistas a futuro si desean ejecutar el programa varios usuarios en el mismo periodo de tiempo esto colapsaría, por lo tanto sería necesario contemplar otro método de traspaso.

A continuación, la aplicación web ejecuta la línea de código “resultado=subprocess.run(['python3',ruta_programa_externo,ruta_excel],stdout=subprocess.PIPE, text=True), donde “ruta_programa_externo” guarda la ubicación del programa externo que se procede a ejecutar y “ruta_excel” guarda la ruta donde se encuentra el archivo Excel descrito anteriormente. Esta línea ejecuta en la terminal del sistema la operación de ejecutar el programa externo exportando los resultados en la variable resultado.

Para hacer funcionar un modelo de calidad de datos se llevan a cabo varios pasos.

- Primero, se hace un **análisis exhaustivo** de los datos existentes para comprender su calidad y determinar los problemas o deficiencias que pueden estar presentes. Esto implica identificar errores, duplicados, valores faltantes, inconsistencias y cualquier otra anomalía que pueda afectar la calidad de los datos.
- Una vez identificados los problemas, se definen **métricas o indicadores de calidad** que permiten cuantificar la calidad de los datos en función de criterios específicos. Estas métricas pueden incluir la precisión, integridad referencial, consistencia y puntualidad de los datos, entre otros.
- Después de establecer las métricas de calidad, se desarrolla un **modelo** que pueda

evaluar y medir automáticamente la calidad de los datos. Este modelo puede incluir reglas, algoritmos o técnicas estadísticas que analizan los datos en busca de patrones o características específicas que indiquen problemas de calidad.

- Una vez aplicado el modelo a los datos, se **generan informes** o resultados que proporcionan una visión detallada de la calidad de los datos. Estos informes pueden incluir métricas cuantitativas, gráficos o visualizaciones que resalten los problemas encontrados y las áreas que requieren mejoras.

El funcionamiento del programa externo es el siguiente: primero extrae la ruta del archivo Excel que le hemos pasado en la línea de código anterior para poder convertirlo a dataframe. Una vez convertido, extraemos las variables que nos interesan de ese dataframe y procedemos a ejecutar una serie de funciones que nos ayudan a determinar la calidad de los datos. Los criterios que de momento están implementados para calcular la calidad de los datos son los siguientes, teniendo una escalabilidad baja pudiendo ser ampliados en el futuro:

El programa externo funciona de la siguiente manera: primero coge la ruta que hemos introducido en el back-end de la aplicación web, que es la ruta del Excel que hemos guardado anteriormente con los datos del fichero introducido y lo vuelve a convertir de dataframe. Una vez hecho esto, extrae las variables necesarias para poder llevar a cabo el análisis, como el número de filas o columnas del dataframe. A continuación, empiezan a aplicarse las funciones creadas para analizar la calidad de los datos del dataframe, que aunque existan más funciones secundarias en el fichero, el resumen de los resultados que se extraen de ellas son los siguientes parámetros, que a su vez son los **criterios de calidad establecidos** anteriormente:

- **Número de celdas vacías:** la concepción ideal de un archivo es que tenga todas las celdas rellenas, por lo tanto un indicador de baja calidad de los datos sería un número de celdas vacías elevado.
- **Compleitud del documento:** relacionado con la propiedad anterior, comparamos el número de celdas vacías respecto al número de celdas totales. Este es un cálculo más exacto ya que no es lo mismo una celda vacía en un fichero con mil celdas que en un fichero con tres celdas.
- **Desviación típica de cada columna:** valor de gran utilidad ya que si es muy alto,

podría indicarnos que algún número ha sido mal introducido.

- **Valores repetidos en cada columna:** podemos interpretar y poner diversos parámetros para averiguar la calidad de un documento, ya que su calidad es algo subjetivo y que se establece a través de una serie de parámetros, en este caso podemos establecer como parámetro de calidad que si el número de valores repetidos se acerca mucho a la cantidad de filas de una columna podría significar que los valores restantes estén mal introducidos. También como otro parámetro de calidad, comprobamos que si el número de veces que se repite el valor en una columna es excesivamente alto sin llegar al 100%, podría ser que hubiera una repetición involuntaria y errónea de algunos datos. Una vez llegue al 100% podría ser una columna con un dato predeterminado, por lo que sería un criterio de buena calidad.
- **Tipo de datos:** el fichero ideal debería presentar una homogeneidad en los tipos de datos de todas sus columnas, por lo tanto si en alguna columna el tipo de dato cambia, la calidad del fichero se ve disminuida.
- **Porcentaje de calidad del fichero:** mediante todos estos parámetros se calcula un porcentaje aproximado de la calidad del fichero midiendo distintos niveles de relevancia según el parámetro sea más o menos influyente en la calidad.

Los parámetros establecidos inicialmente de calidad de datos tienen expectativas de ser ampliados en un futuro debido a que el tiempo de elaboración de este proyecto no ha permitido el desarrollo y la finalización de la creación de más parámetros.

Aparte de dichos parámetros, también se ven reflejados en la página de resultados otros datos relevantes como la media de los números de cada columna o la cantidad de números de cada columna, por ejemplo.

Paso 4. Visualización de los resultados obtenidos

Ya calculados todos los parámetros de calidad necesarios se guardan en un array y se pasan a la aplicación web en formato JSON. La aplicación lo recoge y lo envía a la template correspondiente para poder mostrar los resultados. En ejemplo de cómo resultaría la página de los resultados sería el siguiente:

Resultados del análisis

Podemos estimar la calidad del archivo en un 80%



El archivo analizado está compuesto por 10 columnas y 300 filas.

- Cantidad de números en las columnas
- Medias de las columnas
- Datos más repetidos (valor más repetido , número de veces que se repite)
- Completitud del documento

La completitud del documento es del 90%, con 300 celdas vacías de 3000 totales.

- Desviación típica de las columnas numéricas
- Tipo de datos de las columnas



Ilustración 13 Página de resultados finales

Como podemos observar en la ilustración, la barra de progreso indica el porcentaje aproximado de calidad del archivo calculado por el programa externo. A continuación, nos refleja el número de filas y columnas del documento, seguido por unos desplegables con todos los parámetros de calidad establecidos previamente. Podemos hacer click en cada uno de ellos para ver un resumen de cada parámetro, la mayoría vista desde el punto de vista de las columnas del fichero. Finalmente, tenemos un símbolo de advertencia donde nos aparecerán algunos parámetros algo más subjetivos que podrían indicar un error en la introducción de los datos.

También se ha de recalcar, fuera de la explicación de los pasos a seguir por el usuario en la aplicación, que para el almacenamiento de datos en una base de datos específica existe un archivo llamado db.sqlite3, que es una base de datos en sí misma. Con el fin de visualizar mejor los registros de la base de datos se ha decidido utilizar la aplicación DB Browser for SQLite [48], una aplicación con una interfaz sencilla y manejable.

Por último, adjunto [aquí](#) el enlace con el código de la aplicación al completo:

4. EVALUACIÓN

En esta sección se exponen las pruebas necesarias para asegurar el correcto funcionamiento tanto de la aplicación web como del programa externo. Para una mejor comprensión y visualización, se presentarán las pruebas en forma de tabla, que seguirá el siguiente formato:

ID	
Nombre	
Requisitos relacionados	
Descripción	
Resultado esperado	
Estado	

Tabla 27 Formato tabla pruebas

- ID: cada prueba debe tener un identificador único. Seguirán el formato “P-XX”, donde “XX” es el número identificador de la prueba.
- Nombre: toda prueba deberá tener un nombre claro y conciso que lo identifique.
- Requisitos relacionados: requisitos del sistema que intervienen en cada prueba.
- Descripción: breve descripción explicando más detalladamente la funcionalidad de la prueba.
- Resultado esperado: indica cómo debería reaccionar el sistema ante cada prueba.
- Estado: indica si la prueba ha cumplido con el resultado esperado.

A continuación se presentan las pruebas realizadas con sus respectivos resultados.

ID	P-01
Nombre	Registro
Requisitos relacionados	RS-05, RS-08, RS-10
Descripción	1. El usuario ingresa en la página web y accede a la página de registro. 2. El usuario introduce sus datos y presiona el botón “Registrarse”
Resultado esperado	Si los datos han sido incorrectos, ya sea por el formato del email o porque ya esté registrado ese nombre de usuario en la base de datos, salta un mensaje de error. En caso contrario, almacena esos datos en la base de datos y accede a la página welcome, aparte de registrar en la aplicación web que se ha iniciado sesión.
Estado	Verificado

Tabla 28 P-01

ID	P-02
Nombre	Inicio de sesión
Requisitos relacionados	RS-05, RS-08, RS-09
Descripción	1. El usuario ingresa en la página web y accede a la página de inicio de sesión. 2. El usuario introduce su usuario y contraseña y presiona el botón “Iniciar sesión”
Resultado esperado	El sistema comprueba si son correctos los datos introducidos por el usuario, en caso de no serlo aparecerá un mensaje de error, en caso de si serlo accederemos a la página welcome aparte de registrar en la aplicación web que se ha iniciado sesión.
Estado	Verificado

Tabla 29 P-02

ID	P-03
Nombre	Cierre de sesión
Requisitos relacionados	RS-08, RS-11
Descripción	1. El usuario, que previamente ha iniciado sesión, presiona el botón del encabezado “Cerrar sesión”
Resultado esperado	El sistema redirige al usuario a la página index aparte de registrar en la aplicación web que se ha cerrado sesión.
Estado	Verificado

Tabla 30 P-03

ID	P-04
Nombre	Subir un archivo para analizar
Requisitos relacionados	RS-01, RS-02, RS-06, RS-07
Descripción	1. El usuario, que previamente ha iniciado sesión, presiona el botón del encabezado “Análisis” 2. El usuario introduce el fichero a analizar así como los parámetros necesarios para su correcto funcionamiento (nombre de la hoja y separador en caso de fichero CSV)
Resultado esperado	El sistema redirige al usuario a la página vista previa, genera una vista previa del archivo introducido.
Estado	Verificado

Tabla 31 P-04

ID	P-05
Nombre	Analizar la calidad de un archivo
Requisitos relacionados	RS-03, RS-04, RS-05
Descripción	1. El usuario, ubicado en la página de vista previa, deberá contestar a las preguntas existentes (si hay) y presionar el botón “Cargar”. 2. La aplicación web ejecutará el programa externo, el cual arrojará un array como salida que volverá a coger la aplicación web.
Resultado esperado	El sistema redirige al usuario a la página resultados, con posesión del array generado por el programa externo dispuesto a mostrarlo además de guardar localmente los resultados obtenidos.
Estado	Verificado (a excepción de guardar localmente los resultados obtenidos)

Tabla 32 P-05

ID	P-06
Nombre	Ver resultados del análisis
Requisitos relacionados	RS-05, RS-06, RS-12
Descripción	1. El sistema deberá mostrar los resultados del análisis de manera ordenada y clara una vez se haya presionado el botón “Cargar” de la página vista previa. 2. Se podrá interactuar con los diferentes parámetros de calidad para ver y analizar cada uno de los mismos.
Resultado esperado	El sistema redirige al usuario a la página resultados, mostrando un resumen con gráficos, menús desplegables y buscadores de los parámetros dictaminados previamente para analizar la calidad del fichero.
Estado	Verificado

Tabla 33 P-06

A continuación, se muestra la matriz de trazabilidad entre los requisitos del sistema y las pruebas realizadas.

	P-01	P-02	P-03	P-04	P-05	P-06
RS-01						
RS-02						
RS-03						
RS-04						
RS-05						
RS-06						
RS-07						
RS-08						
RS-09						
RS-10						
RS-11						
RS-12						

Tabla 34 Matriz de trazabilidad entre requisitos y pruebas

5. PLANIFICACIÓN DEL PROYECTO

En esta sección del proyecto se encuentra primeramente un resumen de las metodologías más comunes en el desarrollo de proyectos seguida de una planificación estimada inicial y una planificación real, dividiendo el proyecto en tareas y subtareas con la fecha en la que se pensaba iniciar cada una de ellas con su duración.

5.1 Metodologías para el desarrollo de proyectos

Las metodologías para el desarrollo de proyectos más comúnmente utilizadas son tres: Scrum, Waterfall (Cascada) y el Modelo incremental. En esta sección se hace un resumen de estos métodos y se concluye con el método elegido para elaborar el presente documento.

- **Scrum** [49]: Scrum es una metodología ágil de gestión de proyectos que se centra en la colaboración, la adaptabilidad y la entrega continua de productos o proyectos. Se divide en ciclos llamados "sprints", que generalmente duran alrededor de 2 semanas, durante los cuales se planifica, desarrolla y entrega un conjunto de funcionalidades. Scrum enfatiza la comunicación constante, la retroalimentación y la adaptación a medida que el proyecto avanza.
- **Waterfall (Cascada)** [50]: El modelo Waterfall es una metodología de gestión de proyectos tradicional y secuencial. Se divide en fases secuenciales, como requerimientos, diseño, implementación, pruebas y mantenimiento. Cada fase debe completarse antes de pasar a la siguiente, y los cambios suelen ser difíciles de incorporar una vez que una fase ha comenzado. Es apropiado para proyectos con requisitos estables y bien definidos, pero puede resultar inflexible en situaciones cambiantes.
- **Modelo incremental** [51]: El Modelo Incremental es una metodología que divide el proyecto en módulos. Cada módulo se desarrolla y entrega por separado, y se va construyendo sobre la base del módulo anterior. Cada iteración agrega funcionalidad al producto y permite su lanzamiento temprano. El modelo incremental está principalmente destinado a aquellos proyectos en los que la toma de decisiones y la implementación de manera continua tienen cabida.

Debido a la necesidad de implementación de la aplicación formada principalmente por dos tecnologías diferentes, se ha decidido emplear el modelo incremental. Inicialmente se contempló la división del proyecto en tres módulos diferentes: la aplicación web, el programa externo y la redacción del documento. Cada módulo se fue implementando de manera aislada y uniendo poco a poco según avanzaban. Se ha elegido este método por descarte y conveniencia, ya que al tener una separación clara del proyecto en esas tres partes los modelos Waterfall y Scrum resultaban menos eficientes.

5.2 Planificación inicial

Inicialmente, el proyecto fue planificado para empezar el 15 de mayo de 2023 con una duración total de 109 días, terminando así el 1 de septiembre de 2023.

Nombre de la tarea	Fecha Inicio	Duración (días)	Fecha Fin
Planificación inicial	15/05/2023	2	17/05/2023
Desarrollo aplicación web y programa externo	17/05/2023	54	10/07/2023
Configuración inicial	17/05/2023	2	19/05/2023
Creación de aplicación web	19/05/2023	25	13/06/2023
Creación de programa externo	13/06/2023	20	03/07/2023
Unión de ambas tecnologías	03/07/2023	2	05/07/2023
Estilos y visualización de la página	05/07/2023	5	10/07/2023
Desarrollo del documento	10/07/2023	53	01/09/2023
Introducción	10/07/2023	4	14/07/2023
Estado del arte	14/07/2023	20	03/08/2023
Análisis del problema	03/08/2023	20	23/08/2023
Evaluación	23/08/2023	5	28/08/2023
Resto del documento	28/08/2023	4	01/09/2023
Desarrollo total del proyecto	15/05/2023	109	01/09/2023

Tabla 35 Planificación inicial

La tabla tiene asociada un diagrama de Gantt que se mostrará a continuación:

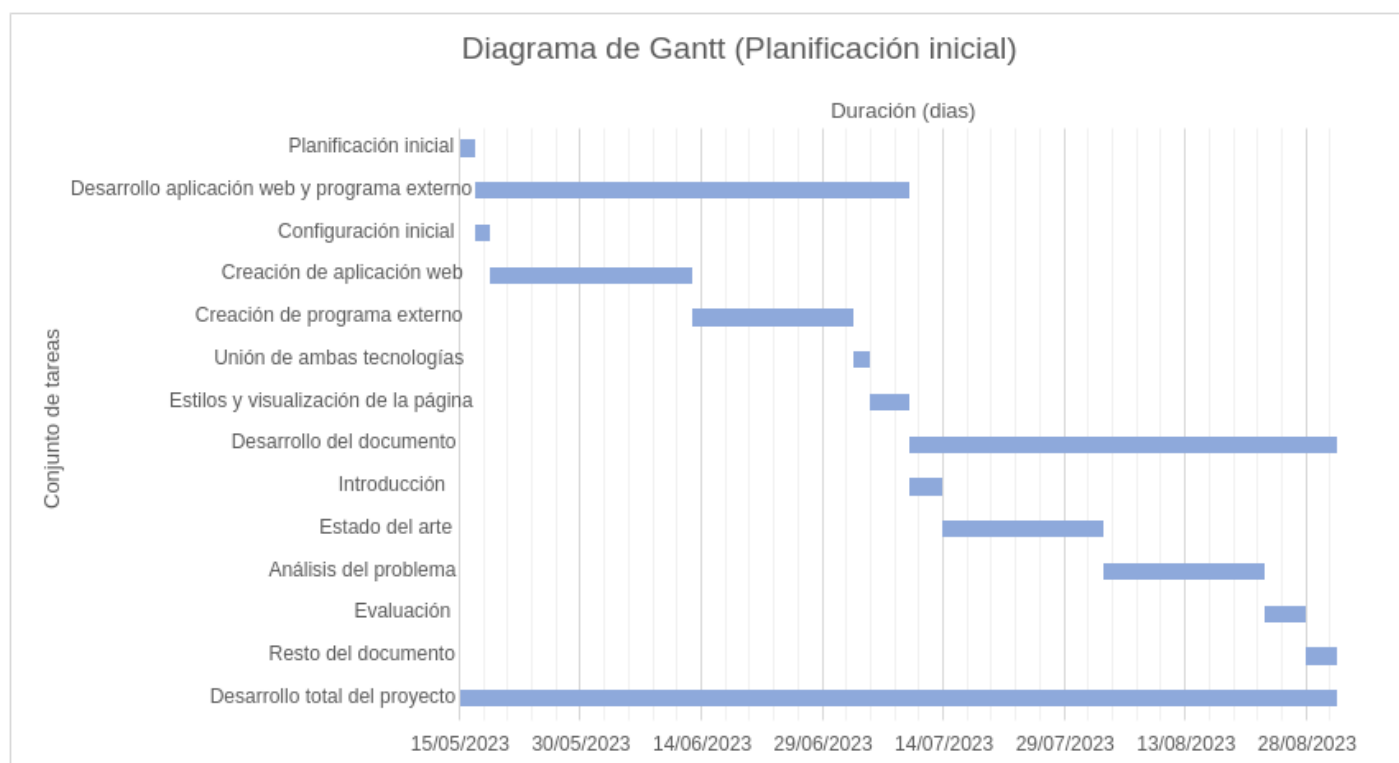


Ilustración 14 Diagrama de Gantt (Planificación inicial)

5.3 Planificación real

La planificación real finalmente se parece en gran medida a la inicial, teniendo como principal diferencia el día de comienzo del proyecto, siendo un mes posterior, teniendo que dividir el tiempo empleado para cada tarea de manera diferente. Aun así, el día estimado para la finalización del proyecto coincide con la fecha real de finalización.

Nombre de la tarea	Fecha Inicio	Duración (días)	Fecha Fin
Planificación inicial	15/06/2023	2	17/06/2023
Desarrollo aplicación web y programa externo	17/06/2023	40	27/07/2023
Configuración inicial	17/06/2023	1	18/06/2023
Creación de aplicación web	18/06/2023	20	08/07/2023
Creación de programa externo	08/07/2023	15	23/07/2023
Unión de ambas tecnologías	23/07/2023	1	24/07/2023
Estilos y visualización de la página	24/07/2023	3	27/07/2023
Desarrollo del documento	27/07/2023	36	01/09/2023
Introducción	27/07/2023	3	30/07/2023
Estado del arte	30/07/2023	16	15/08/2023
Análisis del problema	15/08/2023	10	25/08/2023
Evaluación	25/08/2023	5	30/08/2023
Resto del documento	30/08/2023	2	01/09/2023
Desarrollo total del proyecto	15/06/2023	78	01/09/2023

Tabla 36 Planificación real

Al igual que en la planificación anterior, se muestra un diagrama de Gantt de la planificación real.

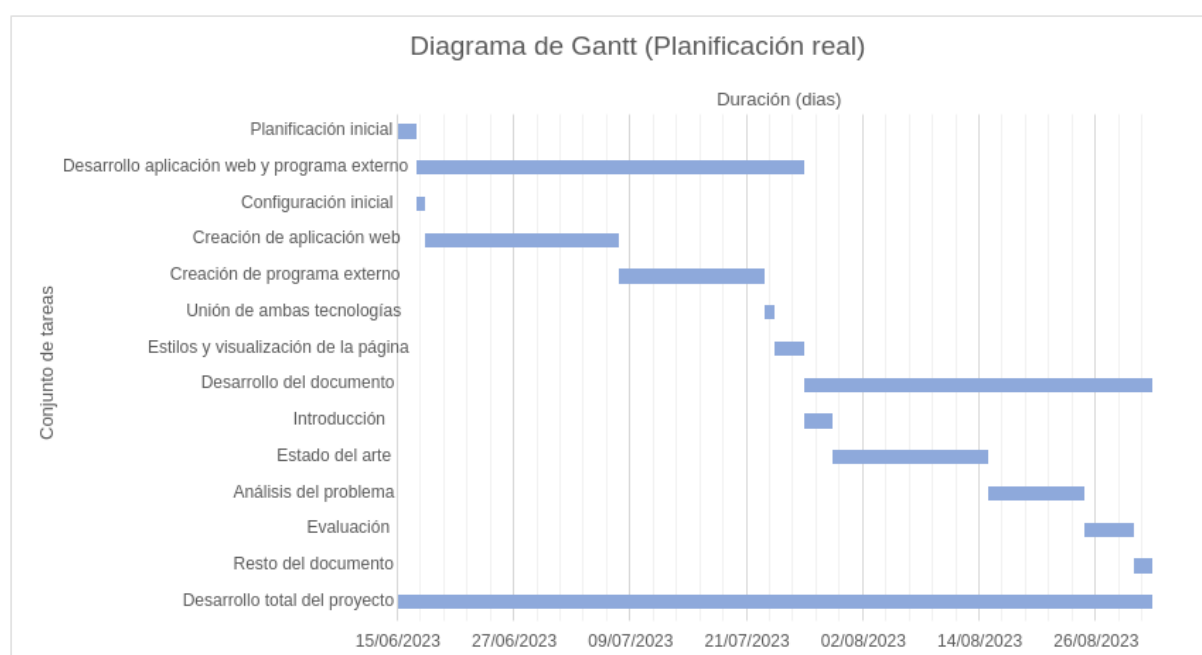


Ilustración 15 Diagrama de Gantt (Planificación real)

6. Presupuesto del proyecto

En esta sección se hará un cálculo aproximado del presupuesto que debería tener el proyecto en caso de que saliese al mercado. Dividiremos este apartado en subapartados según el tipo de elementos que necesitemos.

Para un mejor entendimiento de este apartado se deben aclarar tres puntos importantes:

- La moneda empleada es el Euro (€) junto con dos cifras decimales.
- El impuesto aplicado al presupuesto es el IVA (Impuesto sobre el Valor Añadido), cuyo valor es del 21%.

6.1 Recursos software

En la tabla 37 se realiza un desglose del presupuesto para los recursos software utilizados en este proyecto:

Presupuesto recursos software	
Recurso software	Precio
Ubuntu 20.04.6 LTS	0,00 €
Microsoft Word y Microsoft Excel	0,00 €
Visual Studio Code	0,00 €
Django	0,00 €
Pandas	0,00 €
LibreOffice Draw	0,00 €
DB Browser for SQLite	0,00 €
GitHub	0,00 €

Tabla 37 Presupuesto recursos software

Afortunadamente no se requiere ninguna inversión en cuanto a recursos software, todas las aplicaciones que han intervenido en el desarrollo de este proyecto son gratuitas o han sido proporcionadas gratuitamente.

6.2 Recursos Hardware

Para el desarrollo del proyecto no es necesario ningún recurso hardware en específico, únicamente es necesario un ordenador con conexión a internet, cosa que se excluye del presupuesto.

6.3 Recursos humanos

El presupuesto destinado a los recursos humanos se ha calculado imaginando que el proyecto ha sido llevado a cabo por una empresa. Para garantizar el funcionamiento efectivo de la herramienta desarrollada, se requiere la colaboración de, al menos, cuatro expertos en el campo informático, incluyendo un diseñador web, un analista, un programador especializado en Python y un evaluador encargado de realizar pruebas del sistema. Se ha considerado como referencia salarial para estos profesionales los datos proporcionados por el sitio web especializado talent.com [52].

Presupuesto recursos humanos				
Rol	Analista	Diseñador web	Programador	Pruebas
Días de trabajo	60	40	30	7
Horas/día	2	2	2	2
Salario mensual medio	2.333,00€	1.938,00€	2.375,00€	2.375,00€
Salario/hora	13,25€	11,01€	13,49€	13,49€
Total por trabajador	1590,00€	880,80€	809,40€	188,86€
Total	3469,06€			

Tabla 38 Presupuesto recursos humanos

6.4 Presupuestos indirectos

Para todo proyecto siempre hay ciertos costes indirectos mensuales que cubrir, como la electricidad, el internet, mobiliario, etc.

Presupuestos indirectos		
Nombre	Presupuesto mensual	Total
Internet	20,00€	60,00€
Electricidad	15,00€	45,00€
Mobiliario	50,00€	150,00€
Total	85,00€	255,00€

Tabla 39 Presupuestos indirectos

6.5 Resumen de presupuesto

Una vez desglosados y explicados todos los costes y presupuestos para el proyecto se recopilará un resumen en la tabla 40:

Concepto	Presupuesto
Presupuesto recursos software	0,00€
Presupuesto recursos hardware	0,00€
Presupuesto recursos humanos	3469,06€
Presupuestos indirectos	255,00€
Presupuesto total mensual (sin IVA)	3724,06€
Beneficio 20%	744,81€
Total (sin IVA)	4468,87€
Total (con IVA)	5407,33€

Tabla 40 Resumen de presupuesto

7. Impacto socioeconómico

El impacto socioeconómico de este proyecto es muy amplio ya que el principal objetivo del modelo de calidad de datos es evitar errores en los ficheros y así ahorrar dinero y tiempo invertido en malas prácticas. Por todo esto podemos listar una serie de ámbitos en los que el proyecto tendría un impacto socioeconómico:

- **Creación de empleo:** al desarrollar y comercializar la herramienta, como hemos visto previamente en el punto 6.1.3, es necesario contratar a profesionales para el desarrollo del proyecto. Esto podría generar empleo directo e indirecto, contribuyendo a la generación de puestos de trabajo en tu región.
- **Aumento de la productividad:** la aplicación puede llegar a convertirse en una solución valiosa para las empresas por lo que podría aumentar la productividad en sectores que dependen de la gestión de datos y análisis. Esto podría traducirse en un crecimiento económico a nivel empresarial.
- **Reducción de costos:** la aplicación provocaría que las empresas que la utilicen experimenten una reducción de costos a largo plazo al evitar errores costosos y mejorar la toma de decisiones.
- **Mejora de la competitividad:** las empresas que utilicen la aplicación podrían volverse más competitivas en el mercado al tomar decisiones más informadas y eficientes. Esto podría conducir a un crecimiento económico tanto a nivel empresarial como regional.
- **Mejora de la toma de decisiones:** al proporcionar a las empresas herramientas para analizar y gestionar datos de manera efectiva, se contribuye a una toma de decisiones más fundamentada. Esto podría significar un impacto positivo en la calidad de las decisiones empresariales y, en última instancia, en el éxito de las empresas.
- **Desarrollo tecnológico:** el proyecto podría fomentar el desarrollo tecnológico y la innovación en el campo de la gestión de datos y análisis, lo que podría tener un impacto positivo en la economía en general al impulsar el avance tecnológico.

8. Marco regulador

En esta sección se detallan, primero, las leyes principales de protección de datos que el proyecto debe cumplir al guardar información en bases de datos y segundo, se exponen las licencias de las diferentes herramientas utilizadas para elaborar dicho proyecto.

8.1 Leyes de protección de datos

Las dos leyes que aparecen a continuación deberán cumplirse obligatoriamente en este proyecto y en todos los proyectos que involucren un registro de usuarios o cualquier tipo de almacenamiento de información en bases de datos.

- **Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (RPGPD)** [53]. Este tiene como objetivo proteger los derechos y libertades de las personas físicas en lo que respecta al tratamiento de sus datos personales. Establece principios de legalidad, transparencia y consentimiento informado para la recopilación y uso de información personal. Las organizaciones deben implementar medidas de seguridad, designar un Delegado de Protección de Datos en casos específicos y notificar violaciones de seguridad. El RPGPD también regula la transferencia internacional de datos. Sanciona el incumplimiento con multas considerables y reemplaza la Directiva 95/46/CE, asegurando una gestión responsable de la privacidad en la era digital.
- **Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD) 3/2018 en España** [54]: esta ley complementa el RPGPD al establecer disposiciones adicionales para la protección de datos y derechos digitales. Además de regular la recopilación y procesamiento de datos personales, la ley aborda los derechos digitales de los ciudadanos en entornos digitales, como el derecho a la desconexión digital y el derecho a la privacidad en el ámbito laboral. La LOPDGDD busca asegurar que los derechos fundamentales de las personas, tanto en el mundo digital como en el físico, estén debidamente protegidos y respetados.

La primera ley es general para cualquier empresa u organización establecida en un país de la

Unión Europea o que tengan datos de personas de la Unión Europea, en cambio la segunda ley es un refuerzo de la primera que afecta a empresas cuya ubicación se encuentra en España, ciudadanos españoles y además es la encargada de garantizar sus derechos digitales.

8.2 Licencias

A continuación se muestran las diferentes herramientas que han sido utilizadas para la elaboración del proyecto con sus respectivas licencias:

- **Microsoft Word y Microsoft Excel** [55]: ambos programas se han utilizado para redactar el presente documento y realizar diagramas, gráficos e incluso ayudar al propio proyecto ya que es uno de los tipos de archivos compatibles con él. Ambos se han obtenido mediante la licencia “Office 365 for Students” que proporciona la UC3M a sus estudiantes.
- **Visual Studio Code** [56]: este editor de código fuente está bajo la licencia MIT (Massachusetts Institute of Technology) [57], licencia de código abierto que permite la modificación y redistribución del software.
- **Django**: el framework requiere de una licencia BSD de 3 cláusulas [58] (también conocida como Licencia de Nuevo Estilo BSD).
- **Pandas**: al igual que la aplicación Visual Studio Code, la librería pandas se distribuye bajo la licencia MIT.
- **LibreOffice Draw**: para la creación de imágenes conceptuales ubicadas en este documento se ha utilizado esta herramienta que utiliza la Licencia Pública General de GNU (GPL) [59].
- **DB Browser for SQLite**: esta herramienta de visualización de bases de datos se distribuye bajo la Licencia Pública General de GNU, versión 3 (GLP-3.0)
- **GitHub** [60]: esta herramienta permite la subida y control de versiones del código de la aplicación. No requiere ninguna licencia ya que no es un software en sí mismo que puedas instalar en el ordenador, si no una herramienta web.

9. Conclusiones

Finalmente, en esta sección del documento se presentan las conclusiones extraídas del desarrollo del proyecto. Se dividirá en tres subapartados: en primer lugar, se comparan las soluciones y los resultados obtenidos en el momento de la finalización del proyecto con los objetivos iniciales, en segundo lugar se contemplan las opciones y proyectos que tendrán lugar en el futuro de la aplicación, y en tercer lugar un resumen de la opinión personal del autor del documento respecto al proyecto llevado a cabo.

9.1 Retrospectiva

El objetivo principal del trabajo era elaborar un programa que pudiese leer un archivo Excel y analizar la calidad de sus datos.

Una vez iniciado el procedimiento de creación del programa, el autor consideró que lo más sencillo e intuitivo para que cualquier usuario pudiera ejecutar dicho el programa sería implementar una aplicación que hiciera sencillo el proceso de subir el fichero que se pretenda analizar y, por último, mostrara de una manera sencilla estudiar los resultados obtenidos acerca de la calidad de datos del archivo proporcionado.

Ambos objetivos han sido cumplidos con éxito, teniendo una aplicación web perfectamente operativa y funcional, pudiéndose conectar al programa externo de análisis de datos y finalmente volviendo a coger los resultados que arroja el programa para mostrarlos de manera sencilla en la aplicación.

Por otra parte, la creación de este documento ha supuesto el cumplimiento de otros objetivos como la investigación necesaria al inicio del proyecto para poder elegir correctamente las tecnologías que se usarían, la comprobación del correcto funcionamiento de la aplicación a través de test y el estudio posterior del presupuesto y el marco regulador con mentalidad de sacar la aplicación al mercado. El hecho de cumplir estos objetivos ha supuesto un reto y una concienciación del entorno de cualquier proyecto muy grande para el autor.

9.2 Trabajo futuro

Por desgracia, debido a diversos factores, la aplicación tiene sus limitaciones y aún quedan muchas funcionalidades interesantes que no han podido ser implementadas para este Trabajo Fin de Grado. En esta sección se enumeran y explican todas aquellas funcionalidades que quedan como trabajo futuro :

- **Almacenamiento de histórico de resultados en base de datos:** como se ha comentado anteriormente, la aplicación deberá introducir aquellos resultados de los ficheros que cada usuario introduce en una base de datos de manera independiente de cada usuario, así cada vez que se inicie sesión, cualquier usuario puede volver a ver los resultados que previamente introdujo.
- **Extensión de parámetros de calidad:** dado que los parámetros de calidad llegan a ser muy subjetivos, se prevé en un futuro poder ampliar su número para hacer la aplicación más completa, introduciendo por ejemplo control de idiomas, tildes mal puestas o detección y resaltado caracteres especiales.
- **Mejora de la visualización de la aplicación web:** el diseño de la aplicación web inicialmente es muy sencillo y minimalista, pero para poder salir a mercado necesita una mejora en cuanto a aspecto.
- **Cuestiones en la página vista previa para favorecer el análisis de datos:** las cuestiones en la página previa aún no son funcionales ya que con estos parámetros no se requiere de estas, pero ampliando parámetros de calidad está pensado implementar estas cuestiones y hacerlas funcionales.
- **Mejora visual en la página de resultados:** debido al tiempo de realización de todo el proyecto, no se ha podido implementar todas las mejoras visuales que se pretendían en la página de resultados, como gráficos, mayor cantidad de indicadores, etcétera.

9.3 Conclusiones personales

El desarrollo de este proyecto me ha proporcionado una gran satisfacción y unos conocimientos avanzados sobre temáticas de mi interés. El tema fue elegido a propósito ya que el desarrollo de páginas web y la manipulación de archivos como Excel o CSV está muy en auge actualmente y adquirir mayores conocimientos acerca de estos temas puede resultar

beneficioso en el mercado laboral en un futuro.

Mi nivel de conocimiento sobre ambos campos ha crecido de manera muy notoria, teniendo la capacidad de coger soltura en tecnologías anteriormente desconocidas. En cuanto al desarrollo web, el trabajar con el framework Django, me ha permitido adquirir una visión mucho más clara de la organización y componentes de una página web, en caso de Django utilizando el modelo MVT. También he podido ahondar en la generación de contenido dinámico de código html, cogiendo una gran soltura en dicho aspecto.

Con el programa externo y la utilización de Pandas he comprendido mejor la definición de dataframe y he ampliado mis conocimientos y destreza manejando este tipo de estructura de datos, algo realmente importante en ciencia de datos. La soltura lograda en la manejabilidad y análisis de datos en archivos tipo Excel me ha llevado incluso a crear otros programas utilizando la librería Pandas para fines ajenos a este proyecto, visto todo el potencial que ofrece y recalcando la importancia y utilidad que tiene esta librería en la actualidad.

También se ha de recalcar el proceso de unión de dos tipos de tecnologías diferentes, algo que nunca antes había llegado a realizar. El proceso de unión ha supuesto un mayor entendimiento de algo tan importante como la fusión de programas, algo que considero muy útil en cualquier tipo de proyecto.

Por otro lado y ajeno a las conclusiones de haber generado y puesto en marcha la aplicación, la redacción del presente documento ha supuesto un gran reto y una gran capacidad de aprendizaje. La necesidad de investigar diferentes tipos de tecnologías, compararlas, documentar todo el proceso de la creación de un proyecto y tener en cuenta todos los posibles factores relacionados con el mismo genera una mayor concienciación de la creación de un gran proyecto y de los pasos que deben seguir, mejorando en gran medida la visión de preparación a largo plazo de cualquier proyecto.

Finalmente, se podría determinar que la aplicación no se ha creado con fines de ponerla pública para todo el mundo en un tiempo breve, si no para la mejora y ampliación de conocimientos acerca de la creación de un proyecto grande y el dominio de las dos tecnologías que lo componen, por lo que sí he tenido un alto nivel de dificultad que se ha

visto compensada con el crecimiento exponencial de los conocimientos adquiridos de ambos sectores.

Como conclusión general, este proyecto ha generado un gran incremento de conocimiento así como una gran satisfacción al completar mi primer gran proyecto sin ayuda utilizando tecnologías vistas por primera vez. Se puede concluir que el grado de satisfacción completando este proyecto ha sido muy alto y no habría sido posible sin una capacidad alta de esfuerzo, una buena planificación inicial y una constancia inquebrantable.

10. Summary

10.1 Introduction and objectives

The primary objective of the project at hand is to develop a web application and an external program dedicated to data quality analysis within files, with a particular focus on widely used formats such as CSV and Excel. This initiative has emerged in response to the escalating significance of data in the business world and the imperative need to ensure its integrity and accuracy.

The envisioned web application will serve as a user environment equipped with an authentication system, enabling registered users to upload files for subsequent analysis. It will act as the user interface and deliver data quality results in an intuitive and visually accessible manner.

Conversely, the external program will be engineered to execute the actual analysis of uploaded files. Its design will accommodate various common file formats, including CSV and Excel. Its principal function will be the detection of predefined data quality patterns and the execution of calculations based on these patterns.

Once the external program completes the file analysis, the results will be statistically presented through the web application. Users will have access to diverse dropdowns and charts that showcase the quality results for each predefined parameter, facilitating an efficient assessment of data quality levels.

The project is conceived with a broad range of applications in mind, as data quality is indispensable across virtually all sectors. Moreover, the external program possesses the capability to analyze outliers within files, expanding its potential utility beyond businesses to encompass small enterprises and other industries.

10.2 State of the Art

In the "State of the Art" section, a comprehensive analysis of tools and technologies

optimally applicable to the Final Degree Project is conducted. Given that the project is bifurcated into two distinct components, each demanding disparate technologies, this section is sub-divided accordingly. Additionally, a third subsection is included, cataloging various database storage technologies pertinent to a project of this nature.

10.2.1 Technologies Applicable to the Data Quality Model

Companies often employ specific data structures for data storage and processing, typically in simpler and more manageable formats. These structures are commonly stored in specific file formats such as Excel, CSV, standard text files, XML files, and JSON files. This section examines technologies capable of accessing these specific file types and facilitating processing and analysis to draw conclusions regarding data quality.

After an exhaustive search for compatible and optimal technologies to execute this aspect of the project, a comprehensive investigation of each is conducted.

- **Pandas:** Pandas is a Python library renowned for offering fast, flexible, and expressive data structures. It simplifies working with "relational" or "labeled" data by providing dataframes, series, and data panels. Pandas can read Excel files and convert them into dataframes for analysis and manipulation. It boasts functions for filtering, sorting, joining, and grouping data, making it a potent tool for data analysis.
- **Openpyxl:** Openpyxl, a Python library, is proficient in reading and writing Excel files in the xlsx format. It reads Excel data and converts it into Python data structures such as lists or dictionaries. Openpyxl offers functions for handling spreadsheets, cells, and formulas.
- **Apache POI:** The Apache POI Project is a Java API crafted for manipulating various file formats based on Office Open XML (OOXML) and Microsoft's OLE 2 composite document format. It facilitates the reading and writing of MS Excel, Word, and PowerPoint files using Java. Apache POI can read Excel data and convert it into Java data structures, such as arrays or lists. It supports multiple operating systems and various Microsoft-specific formats.
- **ExcelDataReader:** ExcelDataReader is an open-source, lightweight API written in C# for reading Microsoft Excel files. It is compatible with various Excel formats and

versions, making it suitable for working with Excel files. ExcelDataReader offers high-performance capabilities for handling extensive and intricate data files and supports multiple platforms, including Linux, Windows, and macOS.

A comparative analysis of these tools is provided, evaluating parameters such as file compatibility, read/write permissions, data storage mechanisms, native language, and strengths and weaknesses. This comparison aids in selecting the most appropriate tool based on specific project requisites.

10.2.2 Technologies Applicable to Web Application Development

The second facet of this project entails the development of a fully functional web page, enabling users to upload files and subsequently send them to the data quality analysis program for evaluation and processing. The web application must effectively present the results of data quality analysis.

- **Django:** Django is a high-level web framework designed to expedite the development of secure and maintainable websites. Noteworthy for its completeness, versatility, security measures, scalability, maintainability, portability, and employment of the Model-View-Template (MVT) pattern, Django simplifies web application development by automating tedious and repetitive tasks. It supports multiple databases, including PostgreSQL, MariaDB, MySQL, Oracle, and SQLite.
- **Spring Framework:** Spring Framework, based on Java, is renowned for constructing robust and scalable applications. It offers a rich set of features and modules that simplify application development. Spring encompasses Aspect-Oriented Programming (AOP), Dependency Injection (DI), database integration, and Spring Boot, which streamlines the development of microservices and web applications. Spring's AOP facilitates the separation of concerns within an application, enhancing code modularity and reusability. DI assists in managing object dependencies, thereby improving code maintainability.
- **Express:** Express is a minimalist web framework designed for Node.js, a runtime environment that executes server-side JavaScript. Express streamlines web application development through a powerful routing system, middleware support, and

content negotiation. Its scalability, efficiency, and robust routing capabilities make it a popular choice for constructing various types of web applications.

- **Laravel:** Laravel is a PHP-based, multi-platform framework created to streamline web development, making it faster and more efficient. Recognized for its simplicity and ease of learning compared to other frameworks, Laravel is primarily designed for back-end development but encompasses some front-end capabilities. It uses a scripting language, enhancing its accessibility to beginners, and boasts features like easy routing management, built-in security measures, scalability, and a rich ecosystem of packages.

10.2.3 Databases

Databases can be broadly categorized into relational databases (SQL) and non-relational databases (NoSQL). Each type is discussed separately, encompassing descriptions, characteristics, and popular technologies associated with both, along with a brief overview of each.

10.2.3.1 Relational Databases

Relational databases, also known as SQL databases (Structured Query Language), represent one of the most traditional and extensively employed database types. In these databases, data is organized into tables characterized by rows and columns, and relationships between tables are established using primary and foreign keys. SQL queries are utilized to retrieve and manipulate data for diverse business purposes. Relational databases are often associated with transactional databases that execute commands or transactions collectively. These databases are characterized by properties known as ACID (Atomicity, Consistency, Isolation and Durability)

- **MySQL:** Robust open-source RDBMS designed for critical production environments. Acquired by Oracle Corporation, offering dual licensing. Key advantages include open-source accessibility, client-server architecture for performance, SQL compatibility, view creation for data visualization, support for stored procedures, and automation of database tasks. Supports various storage engines and persistent

connections for optimization.

- **SQLite:** Ultra-lightweight, open-source RDBMS written in C, functioning as a self-contained, serverless engine. Advantages include direct file access for data sharing, cross-platform compatibility, seamless integration with applications, efficient SQL operations, diverse API interfaces, and self-contained operation. Limitations in handling complex nested subqueries and foreign keys during table creation via console commands.

10.2.3.2 Non-Relational Databases (NoSQL)

Non-relational databases, often referred to as NoSQL databases, offer flexibility and scalability for managing large datasets and accommodating evolving data schemas. One noteworthy example is:

- **MongoDB:** Prominent NoSQL database employing a document-oriented approach with key-value pairs. Offers data persistence, high performance, horizontal scalability, support for multiple storage engines, and a pluggable storage API. Known for adaptability to changing data structures and comprehensive documentation in various programming languages.

10.2.4 Conclusions from the State of the Art

After an extensive analysis of various technologies for data quality analysis, web application development, and database management, several conclusions can be drawn:

- **Pandas** for Data Handling: Pandas is an excellent choice for opening and managing different file formats and analyzing data due to its extensive format support, ease of data manipulation, and rich data analysis capabilities. It offers significant advantages for this project.
- **Django** for Web Development: Django stands out as a robust web development framework, offering security measures, simplicity, excellent documentation, and the Model-View-Template (MVT) pattern. Its support for persistent connections and compatibility with MySQL make it suitable for both the current project and potential

future expansions.

- **MySQL** for Database Management: MySQL is a reliable choice for database management, known for its open-source nature, client-server architecture, SQL compatibility, and automation features. Its strong support for data persistence and high performance align well with the project's requirements.

This comprehensive overview highlights the key technologies and tools identified for the project's successful execution, aligning them with the specific objectives and needs of the project.

10.3 Implementation

This section aims to explain the overall functioning of the project, both the web application and the external program, how they communicate with each other, and the criteria considered during their creation.

The web application is built using the Django framework and runs locally on an HTTP server. It employs the Model-View-Template (MVT) architecture, where the model is minimal, the view handles most of the development, and templates provide the application's aesthetic.

Data storage is done in the "db.sqlite3" file, visualized using the DB Browser for SQLite.

The external data quality control program is located in the same folder as the views for convenient communication.

The structure of the web application includes folders for results and templates, along with files like admin.py, apps.py, forms.py, models.py, tests.py, views.py, settings.py, urls.py, and wsgi.py.

Django's administration site allows management of back-end functions, such as creating or deleting parameters, users, models, etc.

Query Workflow on the Website:

Step 1: Login/Register:

Users must register to use the application, providing name, last name, username, email, and password.

Successful registration redirects to the main page.

The login process requires only username and password.

Step 2: Enter a File for Analysis:

Users access the analysis page to upload a file (with accepted types), specify a page in Excel/CSV, and choose a delimiter character.

The application displays a preview of the file for verification.

Step 3: Execute the Data Analysis Program:

After verifying the preview, users click the upload button, generating a local Excel file for analysis.

The web application runs an external program on this file.

The external program extracts data and performs data quality checks, considering parameters like empty cells, document completeness, standard deviation, repeated values, and data types.

Results are stored and sent back to the web application in JSON format.

Step 4: Visualization of Obtained Results:

The web application displays the quality parameters and other relevant data obtained from the external program in a results page.

10.4 Evaluation

Within this section, a comprehensive series of tests were undertaken to ensure the effective operation of both the web application and the external program. These tests were conducted systematically, each bearing a unique identifier in the format of "P-XX." Below is

a summary of these tests, including their names, descriptions, and the expected outcomes:

P-01: Registration

- Description: This test involved a user navigating to the web page and accessing the registration page. The user then provided their details and clicked the "Register" button.
- Expected Outcome: If the user's data was incorrect, either due to email format or an already registered username, an error message would appear. Conversely, successful registration would store the data in the database and redirect the user to the welcome page, also registering the session as logged in.
- Status: Verified

P-02: Login

- Description: The user accessed the web page and entered their username and password on the login page before clicking the "Login" button.
- Expected Outcome: The system verified the correctness of the user's input. If incorrect, an error message would appear; otherwise, the system redirected the user to the welcome page, registering the session as logged in.
- Status: Verified

P-03: Logout

- Description: A user who had previously logged in pressed the "Logout" button on the header.
- Expected Outcome: The system would redirect the user to the index page, registering the session as logged out.
- Status: Verified

P-04: Upload a File for Analysis

- Description: A logged-in user accessed the "Analysis" page and uploaded a file for analysis while providing the necessary parameters.
- Expected Outcome: The system redirected the user to the preview page and generated a preview of the uploaded file.
- Status: Verified

P-05: Analyze File Quality

- Description: In the preview page, the user answered any existing questions (if any) and clicked the "Load" button.
- Expected Outcome: The system executed the external program, which generated an array as output, which the web application retrieved. The user was redirected to the results page with the option to view and analyze the obtained results.
- Status: Verified (except for locally saving the obtained results)

P-06: View Analysis Results

- Description: The system displayed the results of the analysis in an organized and clear manner upon pressing the "Load" button on the preview page. Users could interact with different quality parameters to view and analyze each one.
- Expected Outcome: The system redirected the user to the results page, presenting a summary with charts, dropdown menus, and search options for the predefined quality parameters.
- Status: Verified

All these tests consistently yielded the expected outcomes as per the defined test scenarios, ensuring the robustness and correctness of both the web application and the external analysis program.

10.5 Project Planning

10.5.1 Methodologies for project development

The most commonly used project development methodologies are three: Scrum, Waterfall, and the Incremental Model. This section provides a summary of these methods and concludes with the method chosen for the elaboration of this document.

- **Scrum:** Scrum is an agile project management methodology that focuses on collaboration, adaptability, and continuous delivery of products or projects. It is divided into cycles called "sprints," which typically last around 2 weeks, during which planning, development, and delivery of a set of functionalities occur. Scrum

emphasizes constant communication, feedback, and adaptation as the project progresses.

- **Waterfall:** The Waterfall model is a traditional and sequential project management methodology. It is divided into sequential phases such as requirements, design, implementation, testing, and maintenance. Each phase must be completed before moving to the next, and incorporating changes after a phase has begun is usually challenging. It is suitable for projects with stable and well-defined requirements but can be inflexible in changing environments.
- **Incremental Model:** The Incremental Model is a methodology that divides the project into modules. Each module is developed and delivered separately, building upon the previous module's foundation. Each iteration adds functionality to the product and allows for early release. The incremental model is primarily intended for projects where continuous decision-making and implementation are accommodated.

Due to the need for the implementation of the application, which consists mainly of two different technologies, the incremental model has been chosen. Initially, the project was divided into three separate modules: the web application, the external program, and the document writing. Each module was implemented in isolation and gradually integrated as the project progressed. This method was chosen through a process of elimination and convenience, as having a clear separation of the project into these three parts made the Waterfall and Scrum models less efficient options.

10.5.2 Initial and actual planning

In this section of the project, both an initial and real estimated planning are presented, breaking down the project into tasks and subtasks with their respective start dates and durations.

Initially, the project was scheduled to start on May 15, 2023, with a total duration of 109 days, concluding on September 1, 2023.

The real planning closely resembles the initial one, with the main difference being the project's start date, which is one month later, necessitating a different distribution of time for

each task. Nevertheless, the estimated project completion date coincides with the actual completion date.

10.6 Project budget

In this section an approximate calculation will be made of the budget that the project should have if it were to go on the market.

- **Software Resources:** Fortunately, no expenses are incurred for software resources, as all applications used are free.
- **Hardware Resources:** No specific hardware expenses are required, only a computer with internet access.
- **Human Resources:** The project presupposes collaboration with at least four computer engineering experts, with salaries based on data from talent.com.
- **Indirect Expenses:** Monthly indirect costs, including internet, electricity, and furniture, amount to €255.
- **Budget Summary:** The total monthly budget (excluding VAT) is €3,724.06, with a 20% profit margin, resulting in a total of €4,468.87 (excluding VAT) and €5,407.33 (including VAT).

This budget will have a significant socioeconomic impact, including job creation, increased productivity, cost reduction, improved competitiveness, better decision-making, and fostering technological development and innovation.

10.7 Socioeconomic Impact

The socioeconomic impact of this project is extensive, as the main objective of the data quality model is to prevent errors in files, saving money and time invested in poor practices. Therefore, we can list several areas in which the project could have a socioeconomic impact:

- **Job Creation:** By developing and commercializing the tool, as seen in section 6.1.3, it is necessary to hire professionals for project development. This could generate direct and indirect employment, contributing to job creation in your region.
- **Increased Productivity:** The application may become a valuable solution for businesses, potentially increasing productivity in sectors dependent on data management and analysis. This could translate into economic growth within the business sector.
- **Cost Reduction:** The application would lead to cost reduction for companies using it in the long run by avoiding costly errors and improving decision-making.
- **Improved Competitiveness:** Companies using the application could become more competitive in the market by making more informed and efficient decisions. This could lead to economic growth at both the business and regional levels.
- **Enhanced Decision-Making:** By providing businesses with tools to analyze and manage data effectively, the project contributes to more informed decision-making. This could have a positive impact on the quality of business decisions and, ultimately, on business success.
- **Technological Development:** The project could promote technological development and innovation in the field of data management and analysis, which could have a positive impact on the economy as a whole by driving technological advancement.

10.8 Regulatory Framework

This section outlines the primary data protection laws the project must comply with when storing information in databases and lists the licenses for the different tools used in the project.

10.8.1 Data Protection Laws

Two key data protection laws must be strictly adhered to in this project and any involving user registration or data storage:

- **Regulation (EU) 2016/679 (GDPR):** This regulation aims to safeguard the rights and freedoms of individuals concerning the processing of their personal data. It establishes

principles of legality, transparency, and informed consent for the collection and use of personal information. Organizations must implement security measures, designate a Data Protection Officer in certain cases, and report data breaches. GDPR also regulates international data transfers and imposes significant fines for non-compliance.

- **Organic Law on Data Protection and Guarantee of Digital Rights (LOPDGDD) 3/2018 in Spain:** This law complements GDPR by providing additional provisions for data protection and digital rights. In addition to regulating the collection and processing of personal data, it addresses citizens' digital rights in digital environments, including the right to digital disconnection and workplace privacy. LOPDGDD aims to ensure that fundamental rights are adequately protected in both digital and physical realms.

10.8.2 Licenses

The section details the licenses for various tools used in the project:

- **Microsoft Word and Microsoft Excel:** These tools, used for document preparation, diagrams, and data analysis, are obtained under the "Office 365 for Students" license provided by UC3M to its students.
- **Visual Studio Code:** This open-source source code editor operates under the MIT (Massachusetts Institute of Technology) license, permitting software modification and redistribution.
- **Django:** The framework for web application development operates under a 3-clause BSD license, which allows free usage and modification.
- **Pandas:** Similar to Visual Studio Code, the Pandas library is distributed under the MIT license.
- **LibreOffice Draw:** This tool is used for creating conceptual images in the document and operates under the GNU General Public License (GPL).
- **DB Browser for SQLite:** A database visualization tool distributed under the GNU General Public License, version 3 (GPL-3.0).

10.9 Conclusions

Unfortunately, due to various factors, the application is not fully completed, and there are still many functionalities that have not been implemented. These include:

- **Storing results in a database:** The application should store the results from user-uploaded files in a database independently for each user. This allows users to access their previously entered results when they log in.
- **Expansion of quality parameters:** Since quality parameters can be subjective, there are plans to expand their number in the future to make the application more comprehensive. This could include adding language control, detecting misplaced accents, or handling special characters.
- **Improvement of web application visualization:** The initial design of the web application is simple and minimalistic, but it needs improvement in terms of aesthetics to be market-ready.
- **Enhancements in the data analysis preview page:** The features on the preview page are not currently functional because they are not required with the existing parameters. However, there are plans to implement these features when expanding the quality parameters.
- **Visual improvements on the results page:** Due to time constraints during the project, not all visual enhancements planned for the results page, such as graphs and additional indicators, could be implemented.

In conclusion, the project has provided the author with great satisfaction and advanced knowledge. The chosen topic, involving web development and file manipulation (Excel and CSV), was deliberately selected as these skills are highly sought after in the job market. The author's expertise in both fields has notably grown, with proficiency gained in technologies such as Django for web development and dynamic/static HTML coding.

Working with external programs and using Pandas expanded the author's understanding of dataframes and data manipulation techniques, especially in Excel files. This proficiency with Pandas has even led to the author using it in other projects, emphasizing its importance in programming.

The project also involved the integration of two different technologies, a challenge the author had not previously encountered. This fusion process provided valuable insights into program integration, a skill considered highly beneficial in any project.

Aside from the technical aspects, documenting the project presented a significant challenge and a learning opportunity. Researching different technologies, comparing them, and thoroughly documenting the project's creation process improved the author's project management skills and long-term planning perspective.

In summary, the application was not primarily created for immediate public use but served as a means to enhance the author's knowledge in developing a large-scale project and mastering the two technologies involved. Overall, this project significantly increased the author's knowledge and satisfaction, marking a successful endeavor resulting from high effort, meticulous planning, and unwavering commitment.

Referencias / Bibliografía

- [1] La importancia de los datos, Ayuware
<https://www.ayware.es/blog/importancia-de-los-datos/>
- [2] ¿Por qué es importante la calidad de los datos?
<https://www.lotame.com/es/why-is-data-quality-important/>
- [3] Qué es Pandas <https://pypi.org/project/pandas/>
- [4] Pandas, W3Schools
https://www.w3schools.com/python/pandas/pandas_dataframes.asp
- [5] Pandas Dataframe, W3Schools
https://www.w3schools.com/python/pandas/pandas_dataframes.asp
- [6] Definición de Data Frame, Thedataschools
<https://thedataschools.com/que-es/data-frame/>
- [7] Documentación OpenPyXL, <https://openpyxl.readthedocs.io/en/stable/index.html>
- [8] Apache-poi, PDF Documentativo <https://riptutorial.com/Download/apache-poi-es.pdf>
- [9] ¿El lenguaje Java en declive?
<https://esconditegeek.com/seguridad-y-privacidad/el-lenguaje-java-en-declive/>
- [10] ExcelDataReader Documentación
<https://products.fileformat.com/es/spreadsheet/net/exceldatareader/>
- [11] Documentación oficial de Django <https://docs.djangoproject.com/es/4.2/>
- [12] What is a dataframe?, W3schools.
<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
- [13] ¿Qué es el patrón MVC en programación y por qué es útil?
<https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>
- [14] Documentación bases de datos Django
<https://docs.djangoproject.com/es/4.2/ref/databases/>
- [15] Los modelos MVC y MVT con Django
<https://programacionfacil.org/blog/los-modelos-mvc-y-mvt-con-django/>
- [16] Conexiones persistentes <https://www.slideserve.com/elmo/introductie>
- [17] Documentación de plantilla de Django
<https://docs.djangoproject.com/es/4.2/topics/templates/>
- [18] Documentación de formularios de Django

<https://docs.djangoproject.com/es/4.2/topics/forms/>

[19] Documentación de Spring Framework

<https://docs.spring.io/spring-framework/reference/index.html>

[20] AOP con Spring Framework

<https://tutoriales.edu.lat/pub/spring/aop-with-spring/aop-con-spring-framework#:~:text=Este%20es%20un%20m%C3%B3dulo%20que%20tiene%20un%20conjunto,su%20aplicaci%C3%B3n%20donde%20puede%20conectar%20el%20aspecto%20AOP.>

[21] Tipos de inyección de dependencias con Spring

<https://proitsolution.com.ve/inyeccion-de-dependencias-spring/#:~:text=La%20inyecci%C3%B3n%20de%20dependencias%20es%20un%20patr%C3%B3n%20de,es%20provisto%20por%20los%20m%C3%B3dulos%20spring-core%20y%20spring-beans.>

[22] ¿Qué es Java Spring Boot?

<https://www.ibm.com/es-es/topics/java-spring-boot#:~:text=Java%20Spring%20Boot%20%28Spring%20Boot%29%20es%20una%20herramienta,obstinado%20%3%20La%20capacidad%20de%20crear%20aplicaciones%20aut%C3%B3nomas>

[23] Web oficial Express <https://expressjs.com/es/>

[24] Introducción a Express/Node

https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

[25] Primeros pasos con Express, unos de los frameworks de NodeJS más utilizados

<https://www.arsys.es/blog/express-framework-nodejs#:~:text=Aunque%20hay%20muchos%20frameworks%20de%20NodeJS%2C%20seguramente%20Express,repasamos%20sus%20principales%20caracter%C3%ADsticas%20y%20explicamos%20c%C3%B3mo%20instalarlo.>

[26] Documentación de Laravel <https://laravel.com/docs/10.x/readme>

[27] El Framework PHP Laravel – Construcción de Aplicaciones Web Para Todos

<https://kinsta.com/es/base-de-conocimiento/que-es-laravel/>

[28] ¿Qué es una base de datos relacional?

<https://www.ibm.com/mx-es/topics/relational-databases>

[29] Propiedades de ACID y sistema de gestión de bases de datos relacionales

<https://www.oracle.com/ar/database/what-is-a-relational-database/>

[30] Historia de SQL <https://aws.amazon.com/es/what-is/sql/>

[31] Historia del lenguaje SQL

<https://www.universidadviu.com/co/actualidad/nuestros-expertos/lenguaje-sql-historia-y-conceptos-basicos#:~:text=Lenguaje%20SQL%2C%20historia%20y%20conceptos%20b%C3%A1sicos>

[Isicos%201%20Historia_del%20lenguaje%20SQL%20...%204%20Consultas%20SQL%20](#)

[32] MySQL 5.0 Reference Manual, Capitulo 1

<https://downloads.mysql.com/docs/refman-5.0-es.pdf>

[33] Qué es MySQL: Características y ventajas

<https://openwebinars.net/blog/que-es-mysql/>

[34] MySQL Growth History <https://openwebinars.net/blog/que-es-mysql/>

[35] Documentación SQLite <https://www.sqlite.org/docs.html>

[36] SQLite Data Base: Características de SQLite

<https://sqlitedatabasegrupo.blogspot.com/p/caracteristicas-de-sqlite.html#:~:text=CARACTER%3%8DSTICAS%201%20SQLite%20soporta%20m%C3%BAtiples%20tablas%2C%20%C3%ADndices%2C%20triggers,que%20emplea%20registros%20de%20tama%C3%B1o%20variable.%20M%C3%A1s%20elementos>

[37] Qué es SQLite <https://www.softzone.es/programas/lenguajes/que-es-sqlite/>

[38] Introducción a SQLite, Universidad de el Salvador

https://eisi.fia.ues.edu.sv/materialpublico/pdm115/2019/labs/PDM115_guia_lab03_SQLite.pdf

[39] SQLite: ventajas y desventajas

<https://www.hostgator.mx/blog/sqlite-que-es-y-diferencias-con-mysql/>

[40] Bases de datos no relacionales | Bases de datos de gráficos | AWS

<https://aws.amazon.com/es/nosql/>

[41] Base de datos no relacional. ¿Qué es? Características y ejemplos | Ayuda Ley Protección Datos

https://ayudaleyprotecciondatos.es/bases-de-datos/no-relacional/#Que_es_una_base_de_datos_no_relacional_Definicion

[42] Acerca De Nosotros - Nuestra Historia | MongoDB

<https://www.mongodb.com/es/company>

[43] Official Drivers for MongoDB <https://www.mongodb.com/docs/drivers/>

[44] Introducción a MongoDB <https://www.mongodb.com/docs/manual/introduction/>

[45] Transparencia extraída del curso de Ingeniería de Software de la titulación de Ingeniería Informática de la Universidad Carlos III de Madrid.

[46] Sitio de administración de Django

https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Admin_site

[47] Modelos de Django <https://docs.djangoproject.com/en/4.2/topics/db/models/>

- [48] DB Browser for SQLite <https://sqlitebrowser.org/>
- [49] Metodología para el desarrollo de proyectos Scrum
<https://asana.com/es/resources/what-is-scrum>
- [50] Metodología para el desarrollo de proyectos Waterfall (Cascada)
<https://asana.com/es/resources/waterfall-project-management-methodology>
- [51] Metodología para el desarrollo de proyectos Modelo incremental
<https://blog.comparasoftware.com/como-funciona-el-modelo-incremental/>
- [52] Salario en España 2023, talent.com <https://es.talent.com/salary>
- [53] Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016 <https://www.boe.es/buscar/doc.php?id=DOUE-L-2016-80807>
- [54] Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.
<https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673>
- [55] Office 365 for students <https://www.microsoft.com/en-us/education/products/office>
- [56] VisualStudioCode <https://code.visualstudio.com/>
- [57] Licencia MIT <https://www.licen.cc/es/licencias/mit/>
- [58] Licencia BSD
<https://techlib.net/techedu/licencias-bsd/#:~:text=Esta%20licencia%20tambi%C3%A9n%20se%20conoce%20a%20veces%20como,condiciones%20y%20el%20siguiente%20descargo%20de%20responsabilidad.%202>
- [59] Licencia Pública General de GNU (GPL)
<https://www.gnu.org/licenses/licenses.es.html>
- [60] Página oficial de GitHub <https://github.com/>