UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

"Desarrollo de una aplicación para la automatización de detección de cambios en archivos PDF"

TRABAJO FIN DE GRADO

Junio - 2025

AUTORA: Paula Pérez Monge

DIRECTOR: Federico Botella Beviá

RESUMEN

El presente trabajo aborda la necesidad de automatizar el proceso de detección de cambios entre documentos PDF, específicamente en el contexto de la gestión documental de convenios. La comparación manual de estos documentos es un proceso tedioso, propenso a errores y que consume considerables recursos humanos, por lo que se ha desarrollado una aplicación que permite la detección automática de cambios entre versiones de documentos PDF, generando informes que faciliten la identificación de modificaciones de manera eficiente y precisa.

La aplicación implementa un sistema de procesamiento que opera sobre una estructura de carpetas organizadas jerárquicamente, donde cada subcarpeta contiene dos documentos fundamentales: el borrador como versión inicial del documento y el convenio firmado como versión final con las modificaciones incorporadas. El sistema utiliza algoritmos de comparación de texto y análisis diferencial para identificar las variaciones entre ambos documentos, procesando el contenido extraído de los archivos PDF.

La funcionalidad principal de la aplicación consiste en generar automáticamente informes de cambios que toman como base el documento convenio firmado, resaltando visualmente las modificaciones detectadas respecto al borrador inicial. Esto permite a los usuarios identificar rápidamente qué elementos han sido añadidos, modificados o eliminados durante el proceso de revisión, proporcionando una solución escalable para organizaciones que manejan volúmenes significativos de documentos contractuales.

Los resultados obtenidos demuestran una reducción significativa del tiempo de revisión y minimizan el riesgo de pasar por alto modificaciones importantes. La automatización del proceso de comparación mejora la eficiencia operativa y garantiza un mayor control sobre los cambios documentales, resultando especialmente valiosa en entornos administrativos donde la trazabilidad de dichos cambios es crítica para la toma de decisiones y el cumplimiento normativo.

AGRADECIMIENTOS

La realización de este Trabajo de Fin de Grado no habría sido posible sin el apoyo invaluable y la guía constante de varias personas a quienes deseo expresar mi más sincero agradecimiento.

En primer lugar, quiero extender mi profunda gratitud a mi tutor, su paciencia infinita y su valiosa dirección a lo largo de todo el proceso. Sus conocimientos, su claridad al resolver mis dudas y su constante motivación han sido pilares fundamentales para la culminación de este proyecto.

A mi madre, a mi padre y a mi hermano, mi agradecimiento va más allá de las palabras. Vuestro amor incondicional, vuestro sacrificio y vuestra fe en mí han sido la fuerza motriz que me ha impulsado a superar cada obstáculo. Vuestro apoyo emocional y económico, junto con vuestra compañía, vuestros consejos y vuestro ánimo, han sido esenciales en cada etapa de mi formación académica y un gran alivio en los momentos difíciles.

Finalmente, a mi novio, gracias por tu amor, tu paciencia y tu comprensión ilimitada. Tu apoyo constante, tu capacidad para escucharme en los momentos de mayor estrés han sido vitales para mantener la perspectiva y seguir adelante.

A todos ellos, gracias por ser parte de este camino y por hacer posible este logro.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Capítulo 1: Introducción

Este capítulo expone el marco contextual, la justificación y los objetivos que fundamentan este Trabajo de Fin de Grado centrado en el desarrollo de una herramienta para la automatización de detección de cambios en archivos PDF. A través de un análisis detallado, se establece el entorno de aplicación y se especifican las tecnologías empleadas, detallando tanto la motivación profesional como personal que ha impulsado este trabajo. Finalmente, se definen los alcances y las limitaciones del proyecto.

1.1.- ENTORNO DE APLICACIÓN

Hoy en día, la gestión y el procesamiento de documentos es una actividad cotidiana en todos los sectores. La creciente digitalización de los documentos y la necesidad de gestionar grandes volúmenes de datos de manera eficiente han impulsado la demanda de herramientas que permitan comparar, analizar y gestionar documentos de manera ágil y precisa. En este contexto, la tecnología y las herramientas informáticas desempeñan un papel fundamental, pues facilitan la automatización de procesos repetitivos que exigen alta precisión, liberando recursos valiosos y optimizando significativamente los tiempos de ejecución. Esta digitalización no solo incrementa la eficiencia operativa, sino que también reduce el margen de error humano, garantizando resultados más consistentes y confiables.

La transición de documentos físicos a formatos digitales ha conllevado un incremento exponencial en el volumen de archivos electrónicos que las organizaciones deben gestionar diariamente. Sin embargo, esta transformación digital ha generado una nueva necesidad: desarrollar métodos eficientes para analizar, verificar y contrastar estos documentos, especialmente en formatos estándar como PDF y Word. La capacidad de examinar metódicamente estos archivos se ha convertido en una prioridad para garantizar la integridad de la información.

El entorno de aplicación seleccionado para este proyecto se compone de Google Drive y Google Colaboratory (Colab), el cual permite realizar todo el flujo de trabajo en la nube, lo que ofrece ventajas como la accesibilidad desde cualquier dispositivo con acceso a Internet, la integración con otros servicios de Google y la facilidad para compartir los resultados con otros usuarios de manera rápida y segura.

Google Drive [1], como repositorio de almacenamiento en la nube, facilita la organización y el acceso remoto a los documentos, posibilitando su manipulación y actualización desde cualquier lugar con conexión a Internet. Esta herramienta proporciona una infraestructura robusta, segura y escalable para gestionar los documentos involucrados en el proceso de comparación, lo que resulta esencial dado el volumen de los archivos a procesar.

Por otro lado, Google Colaboratory (Colab) [2] se emplea como el entorno de desarrollo y ejecución destinado al procesamiento de los datos. Colab constituye una plataforma en la nube que habilita la ejecución interactiva de código Python, además de ofrecer acceso gratuito a recursos computacionales. Esta funcionalidad permite la realización eficiente de tareas de análisis de documentos, incluso en el caso de archivos de gran volumen o cuando se precisan procesos que involucran algoritmos de elevada complejidad.

El proceso de comparación de documentos automatizado ofrece una solución eficiente para el análisis de cambios entre distintas versiones de un mismo archivo, facilitando la detección de modificaciones realizadas en los textos, como inserciones, eliminaciones o cambios de redacción. Esto no solo ahorra tiempo a los usuarios, sino que también minimiza el riesgo de errores humanos en la identificación de discrepancias.

Para solucionar este problema se ha desarrollado una solución tecnológica capaz de optimizar el proceso de comparación de documentos, mejorando la eficiencia y reduciendo la probabilidad de errores. Este entorno de aplicación ofrece una plataforma robusta y flexible que puede ser utilizada en diversos contextos donde la gestión y comparación de documentos sea una necesidad.

1.2.- JUSTIFICACIÓN DEL PROYECTO

La digitalización de los procesos administrativos ha experimentado un significativo aumento en los últimos años, especialmente acelerada por la necesidad de gestionar la documentación de manera remota. En este contexto, la Universidad Miguel Hernández gestiona un volumen considerable de convenios con diversas entidades, tales como empresas, instituciones públicas, organizaciones internacionales y otros organismos, los cuales requieren un proceso riguroso de firma y verificación. Este procedimiento resulta

crítico, dado que involucra numerosos documentos que deben ser firmados y verificados, lo que genera una carga administrativa considerable para el personal encargado.

Actualmente, la gestión y verificación de estos convenios en formato PDF, desde su versión inicial hasta la versión final firmada, se lleva a cabo mediante un proceso manual que presenta diversos desafíos, tales como:

- Un elevado consumo de recursos humanos y de tiempo en la verificación de documentos.
- La posibilidad de errores derivados de la comprobación manual de los documentos.
- La dificultad para mantener un control eficiente sobre el estado de cada convenio.
- La ausencia de un sistema automatizado para la verificación de los cambios entre versiones de los documentos.
- La complejidad en el seguimiento de los convenios cuando se manejan múltiples casos de manera simultánea.

El presente proyecto surge como una respuesta a la necesidad de automatizar y optimizar dicho proceso mediante el desarrollo de una solución tecnológica que permita llevar a cabo la comparación automática entre los documentos originales y sus versiones firmadas, los cuales se encuentran almacenados en Google Drive. La implementación de esta herramienta no sólo agilizará de manera significativa el proceso de verificación, sino que, además, proporcionará un nivel superior de fiabilidad en el control documental y en la trazabilidad de los cambios realizados en los documentos.

La automatización de este proceso representa un paso decisivo en la modernización de los procesos administrativos en la universidad, contribuyendo a una gestión más eficiente, ágil y segura de los convenios institucionales.

1.2.1.- Aportación al mundo laboral

La implementación de la solución tecnológica propuesta en este Trabajo de Fin de Grado, destinada a la automatización de la comparación de convenios en formato PDF, representa una contribución significativa al mundo laboral. En primer lugar, este tipo de aplicaciones

mejora la eficiencia en la gestión de documentos, un aspecto esencial en sectores donde se manejan grandes volúmenes de información.

La automatización de procesos repetitivos y propensos a errores, como la comparación manual de versiones de documentos, libera a los profesionales de tareas que consumen un tiempo considerable, permitiéndoles concentrarse en actividades de mayor valor estratégico. En consecuencia, se incrementa la productividad de los empleados, lo que repercute directamente en la reducción de costes operativos. Asimismo, la precisión de los resultados obtenidos mediante herramientas automatizadas minimiza el riesgo de errores humanos, lo cual es fundamental en entornos laborales donde la exactitud en la gestión de documentación es crucial, como en el ámbito legal y académico.

Además, la implementación de esta herramienta puede fomentar el desarrollo de competencias tecnológicas avanzadas entre el personal al involucrar el uso de herramientas informáticas especializadas, como Google Drive y Google Colaboratory para el procesamiento de documentos. Estas herramientas no solo optimizan los procesos internos, sino que también ofrecen a los profesionales la oportunidad de adquirir y aplicar conocimientos sobre soluciones tecnológicas emergentes, lo cual incrementa la competitividad de las organizaciones al mantenerse a la vanguardia en el uso de nuevas tecnologías.

En el ámbito laboral más amplio, la utilización de soluciones automatizadas para la gestión de documentos en formatos digitales responde a una tendencia global hacia la digitalización y optimización de los procesos administrativos, puede abrir la posibilidad de expandir esta tecnología a otras áreas fuera del ámbito académico. De esta manera, la herramienta desarrollada en este proyecto no solo tiene un impacto inmediato en el ámbito administrativo de la Universidad Miguel Hernández, sino que también puede extenderse a otras instituciones y empresas, promoviendo la adopción de prácticas laborales más ágiles y eficientes.

La aportación de este proyecto al mundo laboral se manifiesta en la mejora de la eficiencia operativa, la reducción de errores humanos, la optimización del tiempo de los profesionales y el fomento del uso de nuevas tecnologías en la gestión documental, lo que contribuye a una transformación digital más amplia en diversos sectores.

1.2.2.- Motivación personal

La motivación personal que ha orientado el desarrollo de este Trabajo de Fin de Grado se basa en un profundo interés por la automatización de procesos y la inteligencia artificial, áreas que considero fundamentales en la evolución de la tecnología y en la mejora de la eficiencia en una amplia gama de sectores. A lo largo de mis estudios, he experimentado una creciente fascinación por cómo estas disciplinas pueden transformar y optimizar tareas que tradicionalmente han sido manuales, repetitivas y propensas a errores.

La automatización, en particular, me parece interesante por su capacidad para simplificar y agilizar procesos operativos, haciendo posible la realización de tareas complejas de manera rápida y precisa. La oportunidad de integrar la automatización en el desarrollo de soluciones prácticas ha sido uno de los principales factores que ha impulsado mi dedicación a este proyecto. Además, he observado que la automatización no solo permite la optimización de procesos, sino que también reduce significativamente la posibilidad de errores humanos, incrementando así la fiabilidad y la eficiencia operativa.

Por otro lado, la inteligencia artificial, campo que considero imprescindible en la revolución tecnológica actual, ha sido siempre una de mis principales áreas de interés. La capacidad de los sistemas de inteligencia artificial para aprender, adaptarse y mejorar sus propios algoritmos a partir de los datos representa una de las características más innovadoras y prometedoras de esta tecnología.

Este proyecto, además, me ha brindado la oportunidad de contribuir de manera directa a la mejora de los procesos administrativos dentro del ámbito universitario, en particular en lo que respecta a la gestión de convenios institucionales. La posibilidad de optimizar el trabajo de verificación a través de un sistema automatizado representa un avance significativo.

Mi deseo de aplicar estos conocimientos en el ámbito académico, con el fin de mejorar la eficiencia y la precisión de los procesos administrativos, ha sido el motor principal de este trabajo. Asimismo, este proyecto constituye un paso inicial hacia la integración de soluciones automatizadas que, en el futuro, podrían incorporar técnicas de inteligencia artificial para ofrecer soluciones aún más avanzadas y adaptadas a las necesidades del entorno laboral y académico.

1.2.3.- Oportunidad de aprender y reflejar conocimientos adquiridos aplicando tecnologías de interés

Este Trabajo de Fin de Grado ha supuesto una gran oportunidad para aplicar y consolidar los conocimientos adquiridos a lo largo de mi formación académica, al mismo tiempo que me ha permitido explorar y profundizar en nuevas tecnologías. La integración de herramientas avanzadas, como Google Drive, Google Colaboratory y Python, ha sido fundamental para el desarrollo de una solución tecnológica eficaz y eficiente. El uso de Google Drive como plataforma de almacenamiento en la nube ha proporcionado un entendimiento más profundo sobre los sistemas distribuidos y la gestión de documentos en entornos colaborativos, mientras que Google Colaboratory ha facilitado un entorno adecuado para la ejecución y prueba de código Python, esencial para la automatización de procesos.

El uso de bibliotecas especializadas me ha permitido poner en práctica los principios de programación y la manipulación de documentos en formatos PDF y DOCX, aspectos que son cruciales en el ámbito de la automatización de tareas administrativas. Este proceso me ha brindado la oportunidad de reflexionar sobre las mejores prácticas en el desarrollo de algoritmos de comparación, optimización de código y gestión de flujos de trabajo, contribuyendo así a una mejora en mis habilidades de programación.

A su vez, el proyecto ha sido un punto de partida para el aprendizaje de tecnologías emergentes, especialmente en el campo de la inteligencia artificial. Aunque no se ha incorporado inteligencia artificial de manera directa en la implementación inicial, el desarrollo de la herramienta ha abierto la puerta a la investigación sobre su integración en futuras fases, particularmente mediante técnicas como el aprendizaje automático. Esta faceta del proyecto ha ampliado mi comprensión sobre cómo la inteligencia artificial puede ser aplicada para mejorar procesos de automatización y análisis de documentos, consolidando mi interés en seguir profundizando en estos campos.

Este proyecto ha representado una síntesis entre la aplicación de los conocimientos previos y el aprendizaje de nuevas tecnologías, proporcionando una base sólida para mi crecimiento académico y profesional en áreas clave como la automatización de procesos

y la inteligencia artificial.

1.3.- OBJETIVOS

El objetivo principal de este trabajo es desarrollar una herramienta automatizada que facilite la conversión de DOCX a PDF, la comparación y la visualización de cambios entre versiones de documentos PDF, en particular, entre documentos firmados y sin firmar. La herramienta permitirá gestionar una carpeta principal que contenga diversas subcarpetas. En cada una de estas subcarpetas se encontrarán dos versiones de un convenio: una versión borrador y una versión firmada, que pueden estar en formato DOCX o PDF. El sistema se encargará de comparar documentos PDF y subrayar automáticamente los cambios detectados entre ambas versiones, generando un informe detallado con los resultados de la comparación. Dicho informe será el convenio firmado en PDF, pero subrayado con los cambios encontrados, y se almacenará en la misma carpeta donde se encuentran los documentos originales.

Además, se generará un informe general que incluirá un listado de todas las carpetas comparadas, con el nombre de cada informe generado y el número de cambios detectados en cada comparación junto con un informe en formato Excel para poder registrar todas las ejecuciones, garantizando la trazabilidad y auditoría del proceso.

1.3.1.- Objetivos principales

En este apartado se definen los objetivos principales de este proyecto. Estos objetivos han sido desarrollados para abordar los desafíos identificados de este proyecto. Cada objetivo responde a una necesidad específica y, en conjunto, representan lo que buscamos lograr con este proyecto. Al cumplir estos objetivos, se ofrece una solución práctica y efectiva al problema planteado.

- **Automatizar el proceso de comparación de documentos:** desarrollar una herramienta que detecte y subraye automáticamente las diferencias entre convenios firmados y no firmados en formato PDF.

- **Generar informes detallados:** crear informes personalizados donde los cambios detectados queden claramente subrayados, almacenándolos organizadamente en Google Drive.

- **Generar un informe general consolidado:** implementar la generación de un informe resumen que enumere todas las carpetas comparadas, los informes generados y la cantidad de modificaciones detectadas en cada comparación.

- **Integrar la solución con Google Drive:** utilizar la plataforma de Google Drive para la gestión documental, garantizando el correcto almacenamiento y organización de convenios e informes.

- **Optimizar el rendimiento del algoritmo comparativo:** implementar un sistema de detección de diferencias eficiente que identifique y resalte los cambios con precisión y rapidez.

1.3.2.- Objetivos secundarios

Además de los objetivos principales descritos anteriormente, se plantean una serie de objetivos secundarios que permiten desglosar y abordar de manera más específica los distintos aspectos que conforman la problemática estudiada. Estos objetivos complementarios facilitan una comprensión más detallada del fenómeno en estudio, permitiendo analizarlo desde diversas perspectivas y aportar información valiosa para la consecución del objetivo general. A continuación, se enumeran los objetivos secundarios que orientan el desarrollo de este trabajo.

- **Estudiar tecnologías disponibles:** realizar un estudio exhaustivo sobre herramientas y bibliotecas óptimas para la comparación de documentos PDF y Word, así como para la integración con Google Drive.

- **Evaluar la viabilidad técnica:** determinar si las funcionalidades propuestas son implementables con las tecnologías disponibles, verificando la eficiencia y eficacia del sistema para los usuarios finales.

- **Aplicar conocimientos académicos:** poner en práctica los conocimientos teóricos y prácticos adquiridos durante el grado.

- **Ampliar competencias tecnológicas:** profundizar en el uso de Python y de APIs de Google.
- **Desarrollar experiencia práctica:** resolver problemas reales relacionados con la gestión automatizada de documentos, adquiriendo habilidades transferibles a futuros proyectos.
- **Diseñar una solución escalable:** desarrollar un prototipo funcional aplicable en entornos reales, con potencial de ampliación y mejora futura.
- **Facilitar la experiencia de usuario:** garantizar que la herramienta sea accesible para usuarios con conocimientos básicos, minimizando la curva de aprendizaje.

1.4.- QUÉ NO SE PRETENDE

El desarrollo de este proyecto tiene un enfoque claro y delimitado, por lo que es importante establecer los aspectos que no se abordarán. A pesar de que la herramienta propuesta busca facilitar el proceso de comparación de documentos, existen ciertas funcionalidades y características que no forman parte de este trabajo. Estos límites están definidos por la naturaleza del proyecto y los recursos disponibles, con el objetivo de mantener un enfoque eficiente y alineado con los objetivos principales. A continuación, se detallan los aspectos que no se contemplan en el desarrollo de la herramienta:

- **Desarrollo de una aplicación gráfica o interfaz de usuario avanzada:** este proyecto no contempla la creación de una interfaz gráfica de usuario compleja. La herramienta estará diseñada para funcionar de manera automática en segundo plano sin una interfaz visual compleja, por lo que se ha descartado la inclusión de pantallas interactivas, formularios avanzados o diseños visuales de alto nivel.
- **Comparación de documentos fuera de los formatos PDF:** la herramienta estará limitada a la comparación de documentos en formatos PDF. Se admitirán ficheros Word (DOCX) para su comparación, previa conversión a formato PDF. No se contempla en este proyecto la comparación de otros tipos de archivos, como hojas de cálculo (Excel), presentaciones (PowerPoint) o documentos en otros formatos

menos comunes. El enfoque se centrará únicamente en los documentos que son más habituales en el contexto de los convenios, como son los archivos PDF.

- **Cambiar el contenido de documentos:** la herramienta no tendrá la capacidad de editar, modificar o cambiar el contenido de los documentos. Su único propósito será comparar los documentos y resaltar las diferencias entre las versiones, generando un informe detallado subrayando los cambios detectados. Cualquier modificación de contenido deberá realizarse de forma manual fuera de la herramienta.

Capítulo 2: Antecedentes y estado de la cuestión

La búsqueda de información constituye una etapa fundamental, ya que permite conocer el estado del arte, las fuentes disponibles y determinar los criterios de calidad que debe seguir la investigación. En este capítulo, se presenta la búsqueda de información sobre las herramientas de comparación y visualización de modificaciones en documentos PDF, así como los fundamentos teóricos que sustentan estas tecnologías.

El objetivo principal de este capítulo es adquirir todo el conocimiento posible sobre las herramientas y técnicas existentes, evaluar sus capacidades y limitaciones, y establecer la base conceptual y técnica para el desarrollo de una herramienta adaptada al contexto planteado en este proyecto.

Esta fase es crucial para poder formular adecuadamente el problema planteado y proponer una solución innovadora que permita gestionar documentos, comparar diferentes versiones y generar informes detallados.

2.1.- SITUACIÓN ACTUAL

En el contexto actual de transformación digital que experimentan las organizaciones, la automatización de procesos administrativos se ha convertido en una necesidad prioritaria especialmente para instituciones que manejan grandes volúmenes de información. Las plataformas de procesamiento inteligente [3] de documentos están siendo ampliamente adoptadas por organizaciones que buscan automatizar flujos de trabajo documentales, mejorar la eficiencia operativa y reducir errores. El mercado ha experimentado un crecimiento significativo, reflejado en un aumento del 28 % en el tamaño medio de los contratos de estas soluciones.

La comparación de documentos es una tarea crítica dentro de procesos como la firma de convenios, contratos, actas y otros documentos institucionales. Esta comparación permite identificar con precisión los cambios introducidos entre diferentes versiones de un mismo archivo, lo cual resulta indispensable para garantizar la transparencia, la trazabilidad y la validez de los contenidos.

Un aspecto particularmente relevante en el ámbito institucional es la gestión de documentos firmados digitalmente. La firma digital plantea desafíos únicos para los

sistemas de comparación tradicionales, ya que muchos de ellos no pueden procesar correctamente documentos protegidos con certificados digitales o firmas electrónicas avanzadas. Esta limitación es especialmente problemática en entornos como las universidades, donde los convenios institucionales suelen requerir múltiples firmas para su validación.

Actualmente, existen diversas herramientas, tanto de escritorio como en línea, que permiten comparar documentos. Sin embargo, muchas de ellas presentan limitaciones significativas. Algunas requieren suscripciones costosas, otras no ofrecen resultados fácilmente visualizables, y muchas no permiten una integración directa con sistemas de almacenamiento en la nube, como Google Drive. En el caso particular de los convenios institucionales, donde la exactitud y claridad de los cambios son fundamentales, estas soluciones suelen ser poco flexibles o no se adaptan de manera adecuada a las necesidades específicas del entorno universitario.

Además, la mayoría de estas herramientas están pensadas para el uso individual, no para un flujo automatizado que gestione múltiples documentos de forma simultánea ni para generar informes que faciliten la revisión y trazabilidad de los cambios. Ante esta situación, surge la necesidad de diseñar una herramienta que no solo compare documentos de manera automatizada, sino que también permita generar informes subrayando los cambios detectados, y que se integre de forma natural con plataformas en la nube, como Google Drive.

2.2.- HERRAMIENTAS DISPONIBLES

En el mercado actual, existen diversas herramientas diseñadas para la comparación de documentos en distintos formatos. Estas herramientas permiten a los usuarios detectar cambios entre versiones de un mismo archivo, identificar inserciones, eliminaciones o modificaciones. No obstante, al evaluar estas soluciones dentro del contexto específico de este proyecto que implica la comparación de convenios institucionales, muchos de los cuales están firmados digitalmente, es evidente que presentan limitaciones importantes. A continuación, se analizan algunas de las herramientas más relevantes.

2.2.1.- Soluciones comerciales

2.2.1.1.- Adobe Acrobat Pro

Adobe Acrobat Pro [4] es, sin duda, una de las herramientas más conocidas y ampliamente utilizadas para trabajar con documentos en formato PDF, la cual representa una de las soluciones más consolidadas para trabajar con documentos PDF. Dentro de sus múltiples funcionalidades, ofrece una opción específica para comparar archivos permitiendo visualizar diferencias a nivel de texto, formato y elementos gráficos.

Imagen 2.1: Logo de Adobe Acrobat Pro

Ventajas:

- Alta precisión en la detección de cambios textuales y de formato.
- Interfaz intuitiva que facilita la visualización y navegación entre los cambios detectados.
- Generación de informes visuales que resumen las diferencias entre documentos.
- Herramienta consolidada y de uso profesional en entornos corporativos y académicos.

Sin embargo, a pesar de sus capacidades, Adobe Acrobat Pro presenta limitaciones críticas en el contexto de este proyecto.

Limitaciones:

- No permite comparar archivos PDF firmados digitalmente. Esta es una de las mayores restricciones en el ámbito de los convenios institucionales, ya que estos suelen firmarse electrónicamente para garantizar su validez legal. Cuando un documento ha sido firmado digitalmente, Adobe Acrobat lo considera “protegido”

y bloquea su análisis y modificación, incluyendo la función de comparación. Esto imposibilita su uso en la verificación de diferencias entre una versión sin firmar y otra firmada del mismo documento.

- Su uso es únicamente para archivos PDF.
- Es una herramienta de pago, con un modelo de suscripción costoso que puede representar una barrera económica para muchas instituciones, especialmente si se requiere su uso masivo.
- Su funcionamiento está diseñado principalmente para comparaciones manuales y puntuales, además carece de una integración directa con plataformas en la nube como Google Drive, lo cual limita su escalabilidad y automatización.

Estas limitaciones hacen que Adobe Acrobat Pro, a pesar de su robustez, no sea adecuado para entornos donde se manejen múltiples convenios firmados digitalmente y se requiera una solución automatizada y escalable.

2.2.1.2.- Comparación con Microsoft Word

Microsoft Word [5] incorpora una función nativa de comparación de documentos que permite identificar cambios entre versiones de archivos DOCX. Esta herramienta es particularmente útil en entornos colaborativos donde múltiples usuarios pueden editar un mismo documento.

Imagen 2.2: Logo de Microsoft Word

Ventajas:

- Interfaz familiar para la mayoría de los usuarios.
- Seguimiento de cambios en tiempo real durante la edición colaborativa.

- Capacidad para aceptar o rechazar cambios individualmente.
- Generación de informes de cambios con diferentes niveles de detalle.

Limitaciones:

- Limitado a archivos en formato Word, sin capacidad para comparar PDF.
- Dificultades para procesar documentos con estructura compleja o con elementos gráficos.
- No permite comparar documentos firmados digitalmente.
- Opciones limitadas de automatización para procesos institucionales.
- Dificultades en la integración con Google Drive.

2.2.1.3.- Kofax Power PDF

Kofax Power PDF [6] es una solución empresarial diseñada específicamente para la gestión avanzada de documentos PDF, incluyendo funciones de comparación y análisis.

Imagen 2.3: Logo Kofax Power PDF

Ventajas:

- Interfaz similar a Microsoft Office, facilitando la curva de aprendizaje.
- Capacidades avanzadas de comparación de documentos PDF.
- Soporte para comparación de documentos escaneados mediante OCR.
- Opciones de automatización mediante scripts y macros.
- Integración con sistemas de gestión documental.

Limitaciones:

- Limitaciones similares a Adobe Acrobat en cuanto al manejo de documentos firmados.

- Solución costosa orientada principalmente a entornos empresariales.
- Escasa integración con plataformas en la nube como Google Drive.
- Requiere instalación local, limitando el acceso remoto o distribuido.

2.2.2.- Herramientas en línea

Además de las soluciones de escritorio, han surgido numerosas herramientas en línea que ofrecen funcionalidades básicas y avanzadas para comparar documentos. Entre las más populares se encuentran:

- Diffchecker [7]
- Draftable [8]
- Aspose Compare [9]
- TextCompare [10]

Estas herramientas permiten a los usuarios subir dos documentos y obtener una comparación visual de los mismos, destacando los cambios mediante colores, anotaciones o resúmenes textuales.

Ventajas:

- Son generalmente gratuitas, lo cual las hace accesibles para la mayoría de los usuarios.
- La mayoría no requieren instalación, ya que operan completamente en el navegador.
- Resultan útiles para comparaciones rápidas de documentos sin gran complejidad estructural.
- Algunas ofrecen compatibilidad tanto con PDF como con Word (DOCX).

No obstante, estas herramientas presentan varias limitaciones importantes cuando se consideran para un uso profesional, automatizado y sensible:

- Privacidad y seguridad: al tratarse de plataformas en línea, los documentos deben

ser subidos a servidores externos. Esto representa un riesgo considerable de confidencialidad, especialmente cuando se trabaja con convenios institucionales que contienen datos sensibles.

- Incapacidad de comparar documentos firmados digitalmente: al igual que Adobe Acrobat Pro, muchas de estas herramientas no permiten el análisis de documentos PDF que contienen firmas electrónicas, ya que suelen estar cifrados o protegidos contra modificaciones.
- Falta de automatización: estas herramientas están diseñadas para el uso manual y no ofrecen opciones para ser integradas fácilmente en flujos de trabajo automatizados.
- Resultados limitados o poco personalizables: pueden no ser adecuados para informes formales o procesos administrativos rigurosos.

2.3.- VALORACIÓN

Tras el análisis detallado de las herramientas actualmente disponibles para la comparación de documentos, se concluye que, aunque existen opciones robustas y bien consolidadas, ninguna de ellas responde adecuadamente a las necesidades específicas del proyecto que aquí se plantea.

Por un lado, Adobe Acrobat Pro, pese a ser una herramienta avanzada y precisa para la comparación de archivos PDF, presenta una limitación crucial: no permite comparar documentos que han sido firmados digitalmente, ya que estos se consideran protegidos e inalterables por el propio software. Esto hace inviable su uso donde es frecuente trabajar con versiones finales que contienen firmas electrónicas.

Asimismo, Acrobat Pro no está diseñado para integrarse de forma fluida en procesos automatizados ni flujos de trabajo en la nube, lo que impide su utilización en entornos donde se requiere procesar grandes volúmenes de archivos de forma eficiente, sin intervención manual constante. Su enfoque está más orientado a la revisión individual y manual de documentos, lo que limita drásticamente su escalabilidad.

Por otro lado, las herramientas en línea ofrecen accesibilidad y rapidez, pero también presentan importantes desventajas. La mayoría de estas aplicaciones requieren que los documentos se suban a servidores externos, lo cual compromete la privacidad y la confidencialidad de la información, especialmente cuando se trata de documentación institucional o legalmente sensible. Además, muchas de estas herramientas no soportan correctamente documentos complejos ni firmados digitalmente, y están diseñadas para comparaciones esporádicas, no para flujos automatizados ni procesos masivos.

Además de estas limitaciones, otro aspecto importante es que la mayoría de las soluciones disponibles están diseñadas para comparar únicamente dos documentos por vez, sin ofrecer mecanismos eficientes para recorrer y procesar estructuras de múltiples carpetas que contengan archivos.

Por todos estos motivos, se ha identificado la necesidad de desarrollar una herramienta propia, adaptada específicamente al entorno y requisitos de este proyecto. Esta herramienta busca solucionar de forma directa las carencias observadas en las opciones disponibles, y debe cumplir con los siguientes criterios:

- **Automatización del proceso de comparación**: la herramienta debe ser capaz de recorrer automáticamente una carpeta con múltiples subcarpetas en Google Drive, comparando sistemáticamente pares de documentos sin necesidad de intervención manual.

- **Soporte para archivos firmados digitalmente**: uno de los requisitos fundamentales del proyecto es la posibilidad de comparar versiones de convenios firmados, algo que las herramientas actuales no permiten de manera confiable.

- **Compatibilidad con múltiples formatos**: la herramienta analiza archivos en formato PDF (y en formato Word (DOCX) previamente convertidos a PDF), ya que ambas extensiones son comunes en los procesos administrativos de la universidad.

- **Generación automática de informes detallados**: que reflejen de forma clara las diferencias detectadas y subrayen los cambios sobre el documento final firmado.

- **Integración con Google Drive**: permitiendo acceder, almacenar y compartir los archivos y resultados desde la nube, favoreciendo la colaboración entre usuarios

y departamentos.

- **Preservación de la confidencialidad de la información**: al operar exclusivamente dentro del entorno de Google Drive institucional, se evita el uso de servidores externos y se garantiza que los documentos permanezcan dentro de un entorno controlado, seguro y gestionado por la propia universidad.

El desarrollo de una herramienta personalizada no solo ha sido una decisión estratégica, sino una necesidad operativa para cubrir adecuadamente las exigencias del flujo documental universitario. Esta solución representa una mejora significativa frente a las herramientas existentes, al aportar automatización, precisión, adaptabilidad, seguridad y escalabilidad, elementos fundamentales para una gestión moderna y eficiente de convenios institucionales.

Capítulo 3: Hipótesis de trabajo

3.1.- PLANTEAMIENTO DE HIPÓTESIS Y REQUISITOS TÉCNICOS

A partir del análisis de las herramientas disponibles y de las necesidades específicas detectadas en el proyecto, se plantea la hipótesis de trabajo que guiará el desarrollo de la solución al problema planteado. Es fundamental evaluar las diferentes alternativas tecnológicas que podrían implementarse, con el fin de seleccionar la más adecuada para los objetivos establecidos.

La hipótesis principal de este trabajo sostiene que es posible desarrollar una solución tecnológica que automatice eficientemente la comparación de documentos institucionales firmados digitalmente, generando informes detallados de las diferencias encontradas, y que dicha solución puede integrarse perfectamente en el ecosistema de Google Workspace utilizado por la institución.

Los requisitos técnicos fundamentales que debe cumplir incluyen:

- Capacidad para procesar documentos con firmas digitales en formato PDF.
- Extracción precisa del texto contenido en los documentos.
- Comparación eficiente de textos extensos identificando adiciones, eliminaciones y modificaciones.
- Generación automática de informes visuales que destaquen los cambios.
- Integración fluida con Google Drive para acceso y almacenamiento de documentos.
- Escalabilidad para manejar múltiples carpetas y documentos.
- Bajo costo de implementación y mantenimiento.

En este capítulo se analizarán dos enfoques tecnológicos principales: Blue Prism como herramienta de automatización robótica de procesos (RPA) y Python como lenguaje de programación con sus diversas bibliotecas. Se examinarán las ventajas, limitaciones y aplicabilidad de cada una de estas opciones en el contexto del proyecto, así como el entorno de desarrollo más adecuado, para finalmente establecer conclusiones que determinarán la dirección del desarrollo.

3.2.- BLUE PRISM

Blue Prism [11] es una de las plataformas líderes en el campo de la Automatización Robótica de Procesos (RPA), que permite a las organizaciones automatizar tareas repetitivas y basadas en reglas, emulando las acciones humanas a través de "robots de software" [12]. Esta herramienta ha ganado popularidad en entornos empresariales donde la automatización de procesos administrativos es prioritaria.

Imagen 3.1: Logo Blue Prism

3.2.1.- Características y ventajas

Blue Prism ofrece una serie de características que podrían ser relevantes para este proyecto:

- **Automatización visual**: permite diseñar flujos de trabajo mediante una interfaz gráfica intuitiva, sin necesidad de conocimientos profundos de programación, lo que facilita la creación y mantenimiento de procesos automatizados.
- **Capacidad para interactuar con aplicaciones**: Blue Prism puede interactuar con diversas aplicaciones, incluyendo navegadores web, lo que permitiría acceder a Google Drive y manipular documentos almacenados en la nube.
- **Robustez y escalabilidad**: la plataforma está diseñada para entornos empresariales, lo que garantiza un alto nivel de fiabilidad y la capacidad de manejar volúmenes significativos de trabajo.
- **Manejo de excepciones**: incorpora mecanismos avanzados para la gestión de errores y excepciones, lo que resulta crucial en procesos automatizados que deben funcionar sin supervisión constante.

- **Seguridad**: implementa controles de acceso y protocolos de seguridad robustos, lo que es fundamental cuando se trabaja con documentos confidenciales como convenios institucionales.

3.2.2.- Intento inicial con Blue Prism

Inicialmente, se intentó desarrollar la solución utilizando Blue Prism como plataforma principal, considerando sus capacidades de automatización y su amplio uso en entornos empresariales. Se diseñó un flujo de trabajo que contemplaba las siguientes etapas:

1. Acceso a Google Drive mediante la interfaz web a través del navegador controlado por Blue Prism.
2. Navegación por la estructura de carpetas para localizar los convenios a comparar.
3. Descarga local de los documentos para su procesamiento.
4. Utilización de herramientas para la comparación de los textos.
5. Generación de informes y subida de los mismos a Google Drive.

Sin embargo, durante la fase de implementación, se encontraron diversos obstáculos que limitaron severamente la viabilidad de esta opción:

- **Dificultades con documentos firmados**: Blue Prism no logró procesar adecuadamente los documentos PDF con firmas digitales, ya que no disponía de capacidades nativas para extraer el texto de estos archivos protegidos.
- **Integración limitada con Google Drive**: la interacción con Google Drive a través de la interfaz web resultó ser compleja, lenta y propensa a errores, especialmente al manejar múltiples carpetas y archivos.
- **Complejidad en la comparación de textos**: las capacidades de Blue Prism para el análisis detallado de diferencias textuales resultaron insuficientes para los requisitos del proyecto, necesitando siempre herramientas externas.

- **Problemas de escalabilidad**: el procesamiento de múltiples carpetas y documentos simultáneamente generó cuellos de botella en la ejecución, limitando la escalabilidad de la solución.

- **Rigidez en la generación de informes**: la personalización de los informes con el subrayado específico de los cambios detectados resultó extremadamente compleja de implementar en el entorno de Blue Prism.

- **Costos elevados**: al tratarse de una solución comercial con un modelo de licenciamiento que implica una inversión considerable, lo que podría no ser viable para un proyecto académico o institucional con recursos limitados.

- **Curva de aprendizaje**: aunque su interfaz es visual, el dominio pleno de la plataforma requiere formación especializada y tiempo de adaptación, lo que podría retrasar el desarrollo del proyecto.

- **Limitaciones en el procesamiento de texto**: si bien Blue Prism puede interactuar con aplicaciones, sus capacidades nativas para el procesamiento avanzado de texto y la comparación detallada de documentos son limitadas, especialmente cuando se trata de formatos complejos como PDFs firmados digitalmente.

- **Flexibilidad limitada**: la adaptación de BluePrism para tareas muy específicas, como la extracción y comparación de texto en documentos con firmas digitales, resultó prácticamente imposible sin recurrir a desarrollos externos complejos.

Estos problemas pusieron de manifiesto que, si bien Blue Prism es una herramienta poderosa para muchos tipos de automatización empresarial, sus limitaciones en el procesamiento avanzado de texto y en la manipulación de documentos protegidos hacían inviable su uso para este proyecto específico. La rigidez de la plataforma frente a los requisitos particulares del proyecto, especialmente la necesidad de trabajar con documentos firmados digitalmente y de generar informes altamente personalizados, llevó a reconsiderar la elección tecnológica en búsqueda de alternativas más flexibles y adecuadas para este caso en particular.

3.3.- PYTHON

Tras los obstáculos encontrados con Blue Prism, se propuso Python como alternativa tecnológica. Python [13] es un lenguaje de programación de alto nivel, interpretado y de propósito general, que ha ganado enorme popularidad en el ámbito de la ciencia de datos, el análisis de texto y la automatización de procesos. Su flexibilidad, junto con el ecosistema de bibliotecas disponibles, lo convierte en una opción atractiva para el desarrollo de la herramienta de comparación de documentos.

Imagen 3.2: Logo Python

Python destaca por su sintaxis clara y legible, lo que reduce significativamente la curva de aprendizaje. Esta característica facilita tanto el desarrollo inicial como el mantenimiento posterior del código, permitiendo crear soluciones robustas en menos tiempo.

El ecosistema de bibliotecas de Python es extraordinariamente rico, ofreciendo herramientas especializadas para casi cualquier tarea como el procesamiento de documentos y comparación de textos.

La comunidad activa que respalda Python garantiza soporte constante, documentación actualizada y mejoras continuas en sus bibliotecas. Esto resulta especialmente valioso cuando se enfrentan desafíos técnicos durante el desarrollo.

En el contexto de la automatización, Python ofrece capacidades superiores a soluciones propietarias como Blue Prism, particularmente en escenarios que requieren procesamiento avanzado de texto y comparación documental. Su naturaleza de código abierto también elimina las restricciones de licenciamiento que pueden complicar la implementación en entornos empresariales.

3.3.1.- Python como lenguaje clave para la IA y sus ventajas para el proyecto

La adopción de Python como base tecnológica para el desarrollo de la herramienta presenta numerosas ventajas [14]:

- **Claridad y facilidad de uso**: destaca por su sintaxis sencilla y legible, lo que permite a los desarrolladores centrarse en resolver problemas complejos de IA sin perder tiempo en la estructura del lenguaje. Dicha simplicidad también facilita la revisión de código.

- **Bibliotecas especializadas**: cuenta con diversas bibliotecas orientadas al procesamiento de documentos, que permiten la extracción y manipulación de texto en formatos PDF y Word, incluso en documentos con protecciones o firmas digitales.

- **Integración con Google Drive**: la API de Google Drive para Python facilita la interacción directa con archivos almacenados en la nube, permitiendo listar carpetas, descargar archivos, y subir los informes generados, todo desde un entorno de programación unificado.

- **Algoritmos de comparación de texto**: ofrece bibliotecas como difflib [15] que implementan algoritmos sofisticados para la comparación de textos, permitiendo identificar con precisión las inserciones, eliminaciones y modificaciones entre dos documentos.

- **Generación de informes**: bibliotecas como ReportLab [16] o FPDF [17] permiten la creación de documentos PDF personalizados, lo que resulta ideal para la generación de informes.

- **Flexibilidad y personalización**: permite adaptar el código a necesidades específicas, como el manejo de documentos firmados digitalmente o la implementación de lógicas de comparación adaptadas al contexto de los convenios institucionales.

- **Accesibilidad y costo**: al ser un lenguaje de código abierto la mayoría de sus bibliotecas son gratuitas, lo que elimina las barreras económicas y facilita la

adopción de la solución por parte de cualquier institución.

- **Comunidad activa y recursos abundantes**: posee una de las comunidades más grandes y dinámicas del mundo del software. Esto se traduce en una enorme cantidad de tutoriales, foros, documentación y proyectos de código abierto que facilitan el aprendizaje y la resolución de problemas.

- **Versatilidad e integración**: se integra fácilmente con otros lenguajes y plataformas, lo que convierte en una opción flexible para proyectos que requieren combinar distintas tecnologías. Además, es útil en múltiples áreas desde análisis de datos y automatización hasta desarrollo web y ciencia de datos.

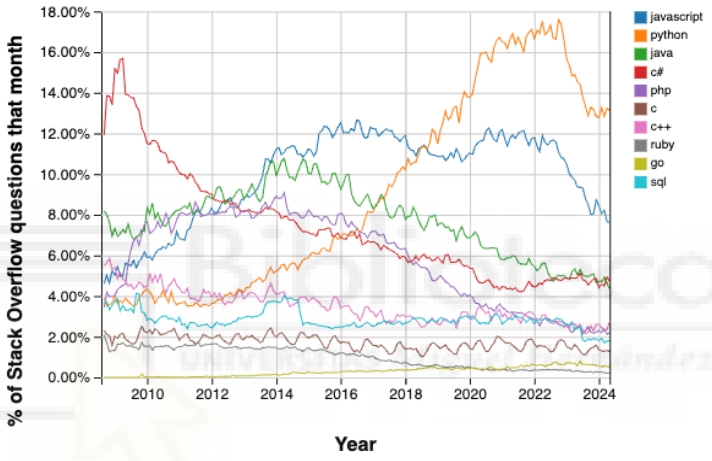

Imagen 3.3: Gráfica comparativa [18] del uso de distintos lenguajes de programación a lo largo de los años

3.3.2.- Consideraciones técnicas

Al implementar la solución en Python, es importante tener en cuenta varios aspectos técnicos:

- **Extracción de texto en documentos firmados**: se requiere verificar la capacidad de las bibliotecas seleccionadas para extraer correctamente el texto de documentos PDF firmados digitalmente, ya que estos suelen presentar protecciones que dificultan su manipulación.

- **Manejo de formatos diferentes**: el sistema debe ser capaz de procesar tanto documentos PDF como Word (DOCX), lo que implica utilizar diferentes bibliotecas y unificar los resultados en formato PDF para la comparación.
- **Rendimiento con documentos extensos**: es necesario evaluar el rendimiento de los algoritmos de comparación cuando se aplican a documentos de gran extensión, típicos en convenios institucionales.
- **Generación de informes visuales**: la creación de informes que subrayen visualmente los cambios requiere un manejo adecuado de estilos y formatos, por lo que se debe seleccionar cuidadosamente la biblioteca para la generación de PDFs.
- **Manejo de excepciones**: la automatización completa del proceso exige un sistema robusto de manejo de errores que permita continuar la ejecución incluso cuando se encuentren documentos con formatos inesperados o problemas de acceso.

3.4.- ENTORNO DE DESARROLLO: GOOGLE COLABORATORY

Google Colaboratory [2], también conocido como Google Colab, es una herramienta gratuita proporcionada por Google que permite escribir y ejecutar código en Python directamente desde el navegador. Está basada en Jupyter Notebooks [19] y es especialmente popular en el ámbito de la ciencia de datos, el aprendizaje automático y la inteligencia artificial [20].

Imagen 3.4: Logo Google Colaboratory

3.4.1.- Justificación de la elección

Google Colaboratory se ha seleccionado como el entorno principal de desarrollo para este proyecto, considerando diversos factores que lo hacen particularmente adecuado para esta implementación:

- **Accesibilidad**: al ser una plataforma basada en la nube, Google Colaboratory permite acceder al entorno de desarrollo desde cualquier dispositivo con conexión a internet, sin requerir instalaciones locales complejas o recursos computacionales significativos.
- **Integración con el ecosistema Google**: dado que el proyecto se centra en la manipulación de documentos almacenados en Google Drive, la integración nativa de Google Colaboratory con este servicio resulta fundamental, simplificando la autenticación y el acceso a los archivos.
- **Entorno preconfigurado**: proporciona un entorno con Python y numerosas bibliotecas científicas ya instaladas, lo que reduce significativamente el tiempo de configuración inicial y garantiza la consistencia del entorno.
- **Recursos computacionales gratuitos**: la plataforma ofrece acceso gratuito a CPU, GPU y RAM, suficientes para las necesidades de procesamiento de este proyecto, eliminando la necesidad de invertir en infraestructura adicional.
- **Interfaz basada en notebooks**: los notebooks de Jupyter integrados en Colab facilitan el desarrollo iterativo, la documentación del código y la visualización de resultados, lo que resulta ideal para un proyecto de investigación y desarrollo como este.
- **Colaboración**: la plataforma permite compartir fácilmente el código con otros miembros del equipo o supervisores, facilitando la revisión y colaboración en el desarrollo.

3.5.- SELECCIÓN DE HERRAMIENTA

Tras analizar las características, ventajas y limitaciones de ambas alternativas tecnológicas, y habiendo intentado inicialmente la implementación con Blue Prism, se pueden establecer los siguientes criterios sobre la selección de la herramienta más adecuada para el desarrollo de este proyecto [21]:

Criterio	Blue Prism	Python
Costo	Elevado (licencias comerciales)	Gratuito (código abierto)
Curva de aprendizaje	Alta, requiere formación especializada	Moderada, ampliamente documentado
Procesamiento de texto	Limitado para comparaciones avanzadas	Extenso, con bibliotecas especializadas
Integración con Google Drive	Posible pero compleja	Directa mediante API oficial
Flexibilidad para requisitos específicos	Limitada	Alta, totalmente personalizable
Mantenimiento a largo plazo	Dependiente del proveedor	Independiente, comunidad activa
Escalabilidad	Alta para procesos empresariales	Adecuada para el alcance del proyecto
Compatibilidad con documentos firmados	Limitada	Factible mediante bibliotecas especializadas
Entorno de desarrollo	Local, requiere instalación	En la nube mediante Google Colab

Generación de informes	Básica, poco personalizable	Avanzada, altamente personalizable

Tabla 3.1: Comparación Blue Prism vs Python

3.5.1.- Conclusión sobre la selección de herramientas

En función del análisis realizado y tras el intento fallido con Blue Prism, se concluye que Python, implementado en el entorno de Google Colaboratory, representa la opción más adecuada para el desarrollo de la herramienta de comparación de documentos, debido a:

1. **Superación de obstáculos técnicos**: las bibliotecas especializadas de Python permiten abordar eficazmente la extracción de texto de documentos firmados digitalmente, superando la principal limitación encontrada en Blue Prism.

2. **Integración óptima con el ecosistema existente**: la combinación de Python y Google Colaboratory proporciona una integración natural con Google Drive, adaptándose perfectamente al flujo de trabajo institucional actual.

3. **Flexibilidad y personalización**: el enfoque basado en código permite desarrollar una solución a medida que se adapte exactamente a los requisitos del proyecto, incorporando lógicas de comparación optimizadas para el contexto de los convenios institucionales.

4. **Viabilidad económica**: al ser una solución basada en software libre y plataformas gratuitas como Google Colaboratory, no existen costos de licenciamiento, lo que garantiza la sostenibilidad económica del proyecto y su potencial adopción por cualquier institución académica.

5. **Accesibilidad y colaboración**: el entorno basado en la nube facilita el acceso desde cualquier ubicación y dispositivo, así como la colaboración entre distintos miembros del equipo durante el desarrollo y posterior mantenimiento.

La combinación de Python con bibliotecas, junto con la API de Google Drive, proporciona todos los componentes necesarios para implementar una solución robusta que cumpla con los objetivos establecidos para el proyecto. Esta selección tecnológica permite abordar de manera efectiva los desafíos técnicos identificados, especialmente la comparación de documentos firmados digitalmente, que constituye una de las principales limitaciones de Blue Prism y otras herramientas comerciales disponibles.

El desarrollo basado en Python no sólo resolverá las necesidades inmediatas del proyecto, sino que también sentará las bases para futuras mejoras y ampliaciones de la herramienta, contribuyendo así a la modernización y optimización de los procesos administrativos en el entorno universitario.

Capítulo 4: Metodología y resultados

Este capítulo detalla la metodología empleada para desarrollar la herramienta de comparación de documentos, abarcando desde la planificación inicial hasta la fase de pruebas. Se describe el proceso seguido para capturar los requisitos, diseñar la solución, implementarla y verificar su funcionamiento, adoptando un enfoque estructurado que ha permitido alcanzar los objetivos establecidos.

4.1.- PLANIFICACIÓN DEL PROYECTO

La planificación del proyecto se ha estructurado siguiendo una metodología ágil adaptada, que ha permitido abordar el desarrollo de forma incremental y flexible. El proyecto se inició en noviembre de 2024 y se ha planificado con una duración aproximada de 7 meses, hasta junio de 2025.

4.1.1.- Metodología de desarrollo

Se ha optado por una metodología ágil basada en Scrum [22], adaptada a las necesidades específicas de un proyecto académico individual. Esta elección responde a la necesidad de mantener un desarrollo iterativo que permita revisar y ajustar los objetivos y el alcance del proyecto a medida que se avanza en su implementación. Las principales características de la metodología aplicada han sido:

- **Ciclos de desarrollo iterativos**: se han establecido sprints de tres semanas de duración para garantizar un avance constante y la posibilidad de reorientar el trabajo según los resultados obtenidos.
- **Revisiones periódicas**: al finalizar cada sprint, se ha realizado una reunión con el tutor para evaluar el progreso, identificar posibles mejoras y planificar el siguiente ciclo de desarrollo.
- **Desarrollo incremental**: la herramienta se ha implementado de forma progresiva, comenzando por funcionalidades básicas (como la extracción de texto) y avanzando hacia características más complejas (como la comparación y generación de informes).

- **Documentación continua**: a lo largo del proyecto, se ha mantenido una documentación actualizada de los avances, decisiones técnicas y problemas encontrados, facilitando así la redacción final de esta memoria.

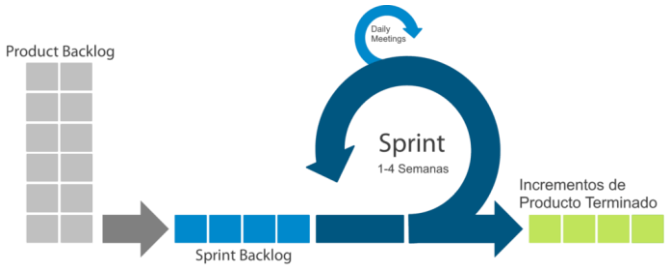

Imagen 4.1: Diagrama metodología Scrum

4.1.2.- Diagrama de Gantt

La planificación del proyecto se ha estructurado en fases claramente diferenciadas, asignando a cada una un periodo específico y estableciendo hitos de control para evaluar el progreso. La estructura ha sido la siguiente:

1. **Fase Inicial:** establecimiento de los objetivos específicos del proyecto, identificación de las necesidades de los usuarios, definición detallada de los requisitos funcionales y no funcionales del sistema y el estudio del entorno de la aplicación

2. **Análisis:** estudio de las diferentes tecnologías disponibles para la manipulación de documentos PDF, comparación de texto y acceso de Google Drive, con el fin de seleccionar las más adecuadas para el proyecto. Además de la evaluación de herramientas y sistemas similares en el mercado, identificando sus fortalezas y debilidades para incorporar buenas prácticas y evitar limitaciones comunes.

3. **Desarrollo:** en esta etapa se realizó la primera implementación utilizando la plataforma Blue Prism para automatizar los procesos de identificación y comparación de documentos. Tras el análisis de las limitaciones encontradas con Blue Prism se inició la migración del desarrollo a Python aprovechando sus bibliotecas especializadas para el procesamiento de documentos. Por último, se

desarrollaron los componentes principales, así como:

- Acceso y autenticación en Google Drive.
- Identificación y clasificación de documentos.
- Conversión de formato DOCX a PDF.
- Algoritmos de comparación de texto.
- Generación de informes individuales con las diferencias resaltadas.
- Generación de un informe global a modo resumen con estadísticas.
- Generación del LOG.

4. **Pruebas y Finalización:** realización de la documentación y de pruebas exhaustivas con diferentes documentos, ajuste de los algoritmos además de la preparación de los materiales para la defensa del proyecto.

La imagen 4.2 muestra el diagrama de Gantt que representa visualmente la distribución temporal de las tareas a lo largo del desarrollo del proyecto, permitiendo observar la secuencia y el solapamiento de las tareas:

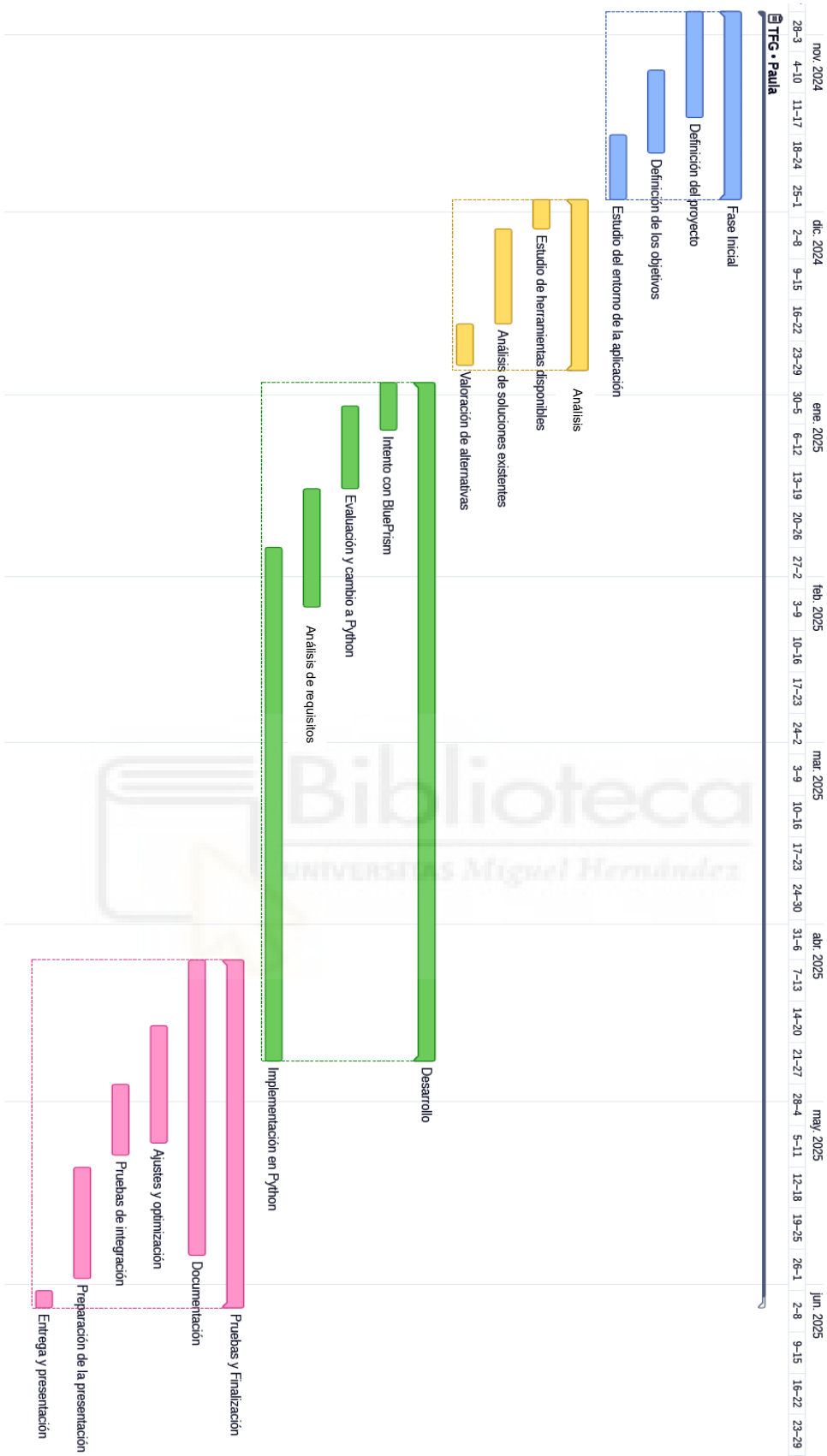

Imagen 4.2: Diagrama de Gantt

A continuación, la tabla 4.1 presenta una visión más detallada de la duración aproximada en días para la realización de cada tarea:

Tareas	**Fecha Inicio**	**Fecha Fin**	**Duración (Días)**
Fase Inicial	**28/10/2024**	**25/11/2024**	**29**
Definición del proyecto	28/10/2024	8/11/2024	12
Definición de los objetivos	6/11/2024	13/11/2024	8
Estudio del entorno de la aplicación	16/11/2024	25/11/2024	10
Análisis	**27/11/2024**	**24/12/2024**	**28**
Estudio de herramientas disponibles	27/11/2024	02/12/2024	6
Análisis de soluciones existentes	03/11/2024	20/12/2024	48
Valoración de alternativas	17/12/2024	23/12/2024	7
Desarrollo	**30/12/2024**	**23/04/2025**	**115**
Intento con Blue Prism	30/12/2024	06/01/2025	8
Evaluación y cambio a Python	03/01/2025	18/01/2025	16
Análisis de requisitos	21/01/2025	08/02/2025	19
Implementación en Python	27/02/2025	23/04/2025	57
Pruebas y Finalización	**08/04/2025**	**03/06/2025**	**57**
Documentación	08/04/2025	28/05/2025	51
Ajustes y optimización	19/04/2025	11/05/2025	23
Pruebas de integración	28/04/2025	12/05/2025	15
Preparación de la presentación	12/05/2025	27/05/2025	16
Entrega y presentación	01/07/2025	02/07/2025	2

Tabla 4.1: Duración de cada tarea

4.2.- ANÁLISIS DE REQUISITOS

El análisis de requisitos constituye una fase crítica en el desarrollo del proyecto, ya que establece las bases funcionales y técnicas sobre las que se construirá la herramienta. Esta etapa ha permitido identificar, documentar y priorizar de manera sistemática las necesidades que debe satisfacer el sistema, asegurando que la solución final responda efectivamente a los problemas planteados.

Para la recopilación de requisitos se han utilizado diversas técnicas, incluyendo entrevistas con los usuarios finales del sistema, análisis de procesos actuales de gestión de documentos y estudio de herramientas similares existentes en el mercado. A continuación, se presentan los requisitos identificados, organizados en categorías funcionales y no funcionales.

4.2.1.- Requisitos funcionales

Los requisitos funcionales describen las capacidades y comportamientos específicos que debe implementar el sistema para cumplir con su objetivo principal: la comparación automatizada de documentos PDF almacenados en Google Drive. Estos requisitos determinan qué debe hacer el sistema y cómo debe responder ante diferentes escenarios.

RF-1	**Acceso autorizado a Google Drive**
Descripción	El sistema debe ser capaz de acceder a una carpeta específica de Google Drive utilizando credenciales autorizadas, garantizando la seguridad y privacidad de los documentos almacenados.

Tabla 4.2: RF-1 Acceso autorizado a Google Drive

RF-2	Identificación de archivos por extensión
Descripción	El sistema debe identificar y listar todos los archivos con extensión DOCX y PDF dentro de cada subcarpeta especificada, creando un listado completo de los documentos disponibles para su procesamiento.

Tabla 4.3: RF-2 Identificación de archivos por extensión

RF-3	Validación de pares de documentos por código y sufijo
Descripción	Para permitir la comparación de documentos dentro de una misma carpeta, deben cumplirse las siguientes condiciones: 1. En cada carpeta deben existir exactamente dos ficheros donde los nombres deben ser diferenciados únicamente por los siguientes sufijos: • "borrador" • "firmado" o "convenio" 2. Los ficheros a comparar deben tener el mismo código identificador. Si no están presentes ambos ficheros (borrador y firmado/convenio) para un mismo código, no se permite realizar la comparación.

Tabla 4.4: RF-3 Validación de pares de documentos por código y sufijo

RF-4	Reconocimiento avanzado de patrones de nomenclatura
Descripción	El sistema debe implementar un algoritmo de extracción de códigos que normalice las inconsistencias en la nomenclatura, identificando el código base inequívocamente incluso ante variaciones de formato. Debe ser capaz de limpiar nombres de archivos eliminando espacios superfluos, sufijos comunes y marcadores de versión, garantizando la

	correcta agrupación de documentos relacionados independientemente de pequeñas variaciones en su nomenclatura implementando un mecanismo avanzado de reconocimiento de patrones, capaz de procesar: ● Patrones simples como "aaaa-nnnnnn Borrador.docx" y "aaaa-nnnnnn Convenio.pdf", donde "aaaa" representa un año y "nnnnnn" un código numérico de identificación, dicho número puede tener la longitud que se desee. ● Patrones complejos que integren letras, números, guiones y guiones bajos como: "C-0327_25-firmado.pdf", "A_1234-567 Convenio.pdf" o "123_456.docx". ● Variaciones y excepciones en la nomenclatura, como sufijos adicionales ("firmado", "informe", "borrador", "convenio"), versiones numeradas entre paréntesis "(1)" o espacios inconsistentes alrededor de guiones o guiones bajos.

Tabla 4.5: RF-4 Reconocimiento avanzado de patrones de nomenclatura

RF-5	**Conversión automática DOCX a PDF**
Descripción	El sistema debe convertir automáticamente los archivos DOCX a formato PDF.

Tabla 4.6: RF-5 Conversión automática DOCX a PDF

RF-6	**Preservación de formato y contenido**
Descripción	La conversión de DOCX a PDF debe preservar el formato, estilo y contenido del documento original, asegurando que no se pierda información relevante durante el proceso de transformación entre formatos.

Tabla 4.7: RF-6 Preservación de formato y contenido

RF-7	Gestión de errores de conversión
Descripción	El sistema debe gestionar adecuadamente cualquier error durante el proceso de conversión, notificando al usuario con mensajes claros y específicos, y continuando con el siguiente documento para evitar la interrupción completa del flujo de trabajo.

Tabla 4.8: RF-7 Gestión de errores de conversión

RF-8	Comparación por pares de documentos relacionados
Descripción	El sistema debe comparar el contenido textual entre pares de documentos, un borrador y el convenio firmado, los cuales comparten el mismo identificador, permitiendo el análisis de las modificaciones realizadas entre versiones.

Tabla 4.9: RF-8 Comparación por pares de documentos relacionados

RF-9	Detección precisa de diferencias textuales
Descripción	El sistema debe detectar con precisión todas las diferencias textuales entre ambos documentos, incluyendo texto añadido en el documento "Convenio" que no existe en "Borrador", texto eliminado del documento "Borrador" que no aparece en "Convenio" y texto modificado entre ambas versiones.

Tabla 4.10: RF-9 Detección precisa de diferencias textuales

RF-10	Creación de informes individuales con resaltado visual de diferencias
Descripción	El sistema creará un nuevo documento PDF por cada par de documentos comparados. Este archivo se denominará utilizando el mismo identificador de los archivos originales, siguiendo el formato "xxxxxxxx_Informe.pdf", donde: ● "xxxxxxxx" representa el identificador de los documentos comparados. El documento PDF generado contendrá una copia del archivo "Convenio" con todas las diferencias respecto al "Borrador" resaltadas en color amarillo. Esto permite al usuario identificar rápida y eficientemente todas las modificaciones entre ambas versiones, proporcionando un registro visual permanente de los cambios detectados durante el análisis comparativo.

Tabla 4.11: RF-10 Creación de informes individuales con resaltado visual de diferencias

RF-11	Generación de informe resumen
Descripción	El sistema generará un informe resumen global en formato Word denominado "AAAAMMDD_HHMM Informe Comparación.docx", donde: ● AAAA: representa el año. ● MM: representa el mes. ● DD: representa el día. ● HH: representa la hora en formato 24h. ● MM: representa los minutos. Este formato de nombre garantiza que cada informe tenga una identificación única basada en su fecha y hora exacta de creación. El informe resumen contendrá: 1. El nombre de cada informe individual generado para cada

	carpeta. 2. Un desglose detallado de los cambios detectados para cada par de documentos comparados, incluyendo: ○ Modificaciones de texto. ○ Adiciones de contenido. ○ Cambios de formato. ○ Alteraciones de un solo carácter (espacios, signos de puntuación, etc.). Esta estructura proporciona una visión integral y organizada de todo el análisis realizado, facilitando la identificación de tendencias en las modificaciones entre documentos. El informe servirá como registro centralizado de todas las comparaciones efectuadas durante la sesión de análisis.

Tabla 4.12: RF-11 Generación de informe resumen

RF-12	**Contabilización total de cambios**
Descripción	El informe debe listar el número total de cambios detectados en cada documento, permitiendo cuantificar el volumen de modificaciones y priorizando la revisión de aquellos documentos con mayor cantidad de alteraciones.

Tabla 4.13: RF-12 Contabilización total de cambios

RF-13	**Formato accesible y exportable**
Descripción	El informe debe presentarse en un formato fácilmente legible y exportable, permitiendo su compartición, archivo o procesamiento posterior mediante otras herramientas ofimáticas o de análisis.

Tabla 4.14: RF-13 Formato accesible y exportable

RF-14	Creación automática de archivo LOG en Excel
Descripción	El sistema debe generar automáticamente una vez ejecutado el programa un archivo Excel denominado “Registro Comparador PDF (LOG).xlsx”, proporcionando un registro completo y estructurado de toda la actividad realizada durante cada sesión de procesamiento.

Tabla 4.15: RF-14 Creación automática de archivo LOG en Excel

4.2.2.- Requisitos no funcionales

Los requisitos no funcionales definen las características cualitativas y restricciones técnicas que condicionan el funcionamiento del sistema, garantizando su calidad, rendimiento y usabilidad.

RNF-1	Continuidad operativa
Descripción	El sistema debe operar sin interrupciones durante todo el proceso de análisis, implementando mecanismos de recuperación en caso de errores parciales.

Tabla 4.16: RNF-1 Continuidad operativa

RNF-2	Autenticación segura
Descripción	El sistema debe utilizar métodos seguros para la autenticación y acceso a Google Drive, cumpliendo con los estándares OAuth 2.0.

Tabla 4.17: RNF-2 Autenticación segura

RNF-3	Confidencialidad
Descripción	Toda la información procesada debe permanecer confidencial y no debe ser transmitida a terceros.

Tabla 4.18: RNF-3 Confidencialidad

RNF-4	Usabilidad de interfaz
Descripción	La interfaz de la aplicación debe ser intuitiva. requiriendo una formación mínima para su uso efectivo.

Tabla 4.19: RNF-4 Usabilidad de interfaz

RNF-5	Retroalimentación del proceso
Descripción	El sistema debe proporcionar retroalimentación clara sobre el progreso de los procesos de conversión y comparación.

Tabla 4.20: RNF-5 Retroalimentación del proceso

RNF-6	Informes comprensibles
Descripción	Los informes generados deben ser fácilmente interpretables, utilizando un esquema de colores consistente y explicaciones claras de las diferencias encontradas.

Tabla 4.21: RNF-6 Informes comprensibles

RNF-7	Modularidad del código
Descripción	El código fuente debe estar modularizado para facilitar el mantenimiento y la extensión de funcionalidades.

Tabla 4.22: RNF-7 Modularidad del código

RNF-8	Adaptabilidad a estructuras
Descripción	El sistema debe ser adaptable para trabajar con diferentes estructuras de carpetas en Google Drive.

Tabla 4.23: RNF-8 Adaptabilidad a estructuras

RNF-9	Escalabilidad
Descripción	El sistema debe ser escalable para manejar un mayor de documentos sin una degradación significativa del rendimiento.

Tabla 4.24: RNF-9 Escalabilidad

4.2.3.- Roles de usuario

El sistema de la aplicación cuenta con dos actores principales: Usuario de la aplicación y Sistema.

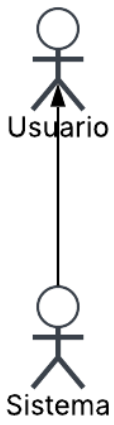

Imagen 4.3: Diagrama roles de usuario

Actor	Usuario de la aplicación
Descripción	Actor principal que interactúa con el sistema exclusivamente para ejecutar la aplicación.
Casos de uso	CU-13

Tabla 4.25: Actor – Usuario del sistema

Actor	Sistema
Descripción	El sistema es quien responde a las acciones del usuario, interpreta sus entradas y realiza internamente todas las operaciones necesarias para entregar una salida correcta.
Casos de uso	CU-1, CU-2, CU-3, CU-4, CU-5, CU-6, CU-7, CU-8, CU-9, CU-10, CU-11, CU-12

Tabla 4.26: Actor – Sistema

4.2.4.- Casos de uso

Los casos de uso constituyen una técnica de modelado que permite representar la interacción entre los usuarios y el sistema, describiendo la secuencia de acciones que se realizan para lograr un objetivo específico. Esta sección presenta los casos de uso identificados para este proyecto, basados en los requisitos funcionales descritos anteriormente.

El siguiente diagrama muestra una visión general de los casos de uso del sistema y su relación con los actores identificados:

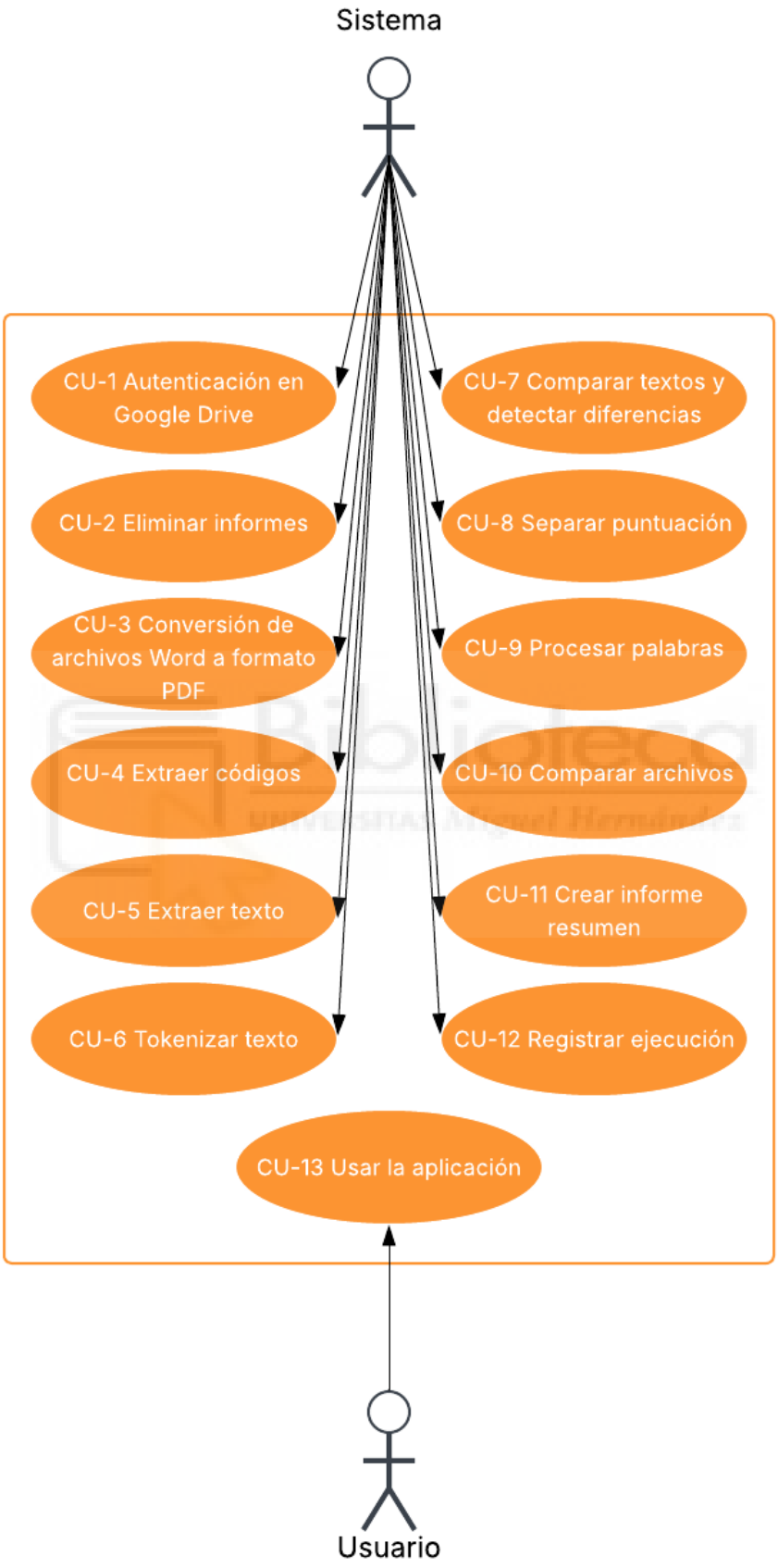

Imagen 4.4: Diagrama casos de uso

CU-1	**Autenticación en Google Drive**	
Actores	Sistema	
Descripción	El sistema permite al usuario autenticarse de forma segura con su cuenta de Google para obtener acceso autorizado a los documentos almacenados en Google Drive.	
Dependencias		
Precondición	El usuario debe disponer de una cuenta de Google válida y activa con acceso a Google Drive.	
Secuencia normal	**Paso**	**Descripción**
	P1	El usuario selecciona la opción de autenticación en Google Drive.
	P2	El sistema redirige al usuario al flujo de autenticación de Google.
	P3	El usuario introduce sus credenciales en la interfaz de Google.
	P4	Google valida las credenciales y devuelve un token de acceso al sistema.
	P5	El sistema almacena temporalmente el token para permitir el acceso a los documentos del usuario.
Postcondición	El sistema obtiene acceso autorizado a la cuenta de Google Drive del usuario.	
Excepciones	• El usuario cancela el proceso de autenticación. • Las credenciales del usuario son incorrectas o la autenticación falla. • Error en la comunicación con los servicios de Google.	
Comentarios		

Tabla 4.27: CU-1 Autenticación en Google Drive

CU-2	**Eliminar informes**	
Actores	Sistema	
Descripción	Recorre todas las subcarpetas de una carpeta principal en búsqueda de archivos PDF que sean informes, para posteriormente eliminarlos. Al realizar este paso nos aseguramos de que no haya ningún problema a la hora de volver a generarlos.	
Dependencias	CU-1	
Precondición	● La ruta existe y es accesible. ● La ruta tiene permisos de modificación.	
Secuencia normal	**Paso**	**Descripción**
	P1	Recorre la carpeta principal en búsqueda de subcarpetas.
	P2	Recorre cada subcarpeta en búsqueda de archivos con la palabra “informe” en el nombre.
	P3	Si encuentra archivos: los elimina y confirma la acción mediante un mensaje indicando el nombre del archivo eliminado y en la carpeta en la que se encuentra. Si no encuentra archivos: notifica que no hay ningún informe que eliminar.
Postcondición	Los archivos informes se han eliminado.	
Excepciones	● Ruta no existente. ● Fallo en la eliminación.	
Comentarios		

Tabla 4.28: CU-2 Eliminar informes

CU-3	**Conversión de archivos Word (.docx) a formato PDF**	
Actores	Sistema	
Descripción	Convertir un archivo de Microsoft Word (.docx) a formato PDF, manteniendo el formato original del documento.	
Dependencias	CU-1	
Precondición	● El archivo .docx existe y es accesible. ● La ruta de salida es válida y tiene permisos de escritura. ● Las dependencias necesarias están instaladas.	
Secuencia normal	**Paso**	**Descripción**
	P1	Validar que el archivo .docx existe.
	P2	Utilizar una librería adecuada para convertir el archivo PDF.
	P3	Guardar el PDF en la ubicación deseada.
	P4	Devolver la ruta del PDF generado o lanzar un mensaje de error si algo falla.
Postcondición	El archivo convertido a PDF existe en la ruta especificada.	
Excepciones	● Archivo no encontrado. ● Extensión incorrecta. ● Fallo en la conversión.	
Comentarios		

Tabla 4.29: CU-3 Conversión de archivos Word (.docx) a formato PDF

CU-4	**Extraer códigos**	
Actores	Sistema	
Descripción	El sistema extrae y valida códigos de identificación de archivos basándose en patrones específicos que deben coincidir con la estructura numérica de la carpeta que contiene dichos archivos. Se normalizan nombres de archivos eliminando sufijos comunes y corrige errores de formato para obtener un código válido. Dicha extracción: ● Se realiza con expresiones regulares para limpieza y extracción de patrones. ● Es insensible a mayúsculas/minúsculas. ● Normaliza automáticamente los separadores al formato de la carpeta.	
Dependencias	CU-1	
Precondición	● Estructura de carpetas organizadas con códigos numéricos. ● Archivos con nomenclatura estandarizada. ● El nombre de la carpeta debe contener al menos un número. ● Los archivos deben seguir un patrón de nomenclatura con códigos alfanuméricos separados por guiones (-) o guiones bajos (_).	
Secuencia normal	**Paso**	**Descripción**
	P1	Recibir nombre de carpeta y nombre de archivo como parámetros.
	P2	Normalizar nombre de carpeta eliminando espacios y caracteres de control.

	P3	Detectar separador utilizado en la carpeta (_ o -).
	P4	Extraer números de la carpeta usando expresiones regulares.
	P5	Limpiar el nombre del archivo eliminando espacios al inicio y final.
	P6	Corregir espacios alrededor de separadores.
	P7	Eliminar sufijos comunes (firmado, informe, borrador, convenio). Además, es insensible a las mayúsculas.
	P8	Eliminar números entre paréntesis (ej: "(1)", "(2)") y la extensión del archivo.
	P9	Extraer números del código del nombre del archivo con expresiones regulares.
	P10	Validar que el número de bloques numéricos coincida entre código y carpeta.
	P11	Validar que los números sean iguales.
	P12	Retornar código normalizado.
Postcondición	**Éxito**: Retorna un código válido normalizado que coincide con la estructura de la carpeta. **Fallo**: Retorna None cuando no se puede extraer un código válido.	
Excepciones	Retorna None cuando: ● No se encuentra patrón de código en el nombre del archivo. ● Número de bloques numéricos diferentes entre código y carpeta por lo tanto los números no coinciden. ● Nombre de archivo vacío o solo espacios.	

<table>
<tr><td>Comentarios</td><td>Ejemplos de uso válido:

<table><tr><td>Carpeta</td><td>Archivo</td><td>Código extraído</td></tr><tr><td>72_25</td><td>72_25 convenio.pdf</td><td>72_25</td></tr><tr><td>434_25</td><td>C-0434-25 borrador (1).pdf</td><td>C-0434_25</td></tr><tr><td>133_25</td><td>C-0133_25- firmado.pdf</td><td>C-0133_25</td></tr><tr><td>408_25</td><td>C-408 _ 25-convenio.pdf</td><td>C-408_25</td></tr></table>
Ejemplos de uso inválido:

<table><tr><td>Carpeta</td><td>Archivo</td><td>Código Extraído</td></tr><tr><td>72_25</td><td>C-0072_025 convenio.pdf</td><td>None</td></tr><tr><td>2025_17633</td><td>2025_17833-borrador.docx</td><td>None</td></tr></table></td></tr>
</table>

Tabla 4.30: CU-4 Extraer códigos

CU-5	**Extraer texto**
Actores	Sistema
Descripción	Extrae y devuelve el texto contenido en un archivo PDF. Utiliza la biblioteca PyMuPDF (fitz) para abrir el PDF, recorrer cada página y extraer su contenido textual. Finalmente, cierra el documento y devuelve el texto acumulado.
Dependencias	CU-1
Precondición	● El archivo PDF existe y es accesible en la ruta especificada. ● El archivo tiene permisos de lectura. ● La biblioteca PyMuPDF está disponible en el sistema.

	● El archivo PDF no está corrupto. El archivo PDF contiene texto legible (no está escaneado como imagen).	
Secuencia normal	**Paso**	**Descripción**
	P1	La ruta proporcionada es válida y accesible.
	P2	Abrir el documento PDF.
	P3	Obtener el número total de páginas del documento.
	P4	Inicializar la variable para acumular el texto extraído.
	P5	Por cada página obtiene el objeto página para extraer el texto de cada una de ellas.
	P6	Una vez extraído concatenar el texto extraído al acumulador.
	P7	Continuar con la siguiente página hasta recorrer todas y extraer el texto de cada una de ellas.
	P8	Cerrar el documento PDF.
	P9	Retornar el texto extraído.
Postcondición	El archivo convertido a PDF existe en la ruta especificada.	
Excepciones	● Archivo no encontrado. ● Archivo corrupto. ● Permisos insuficientes.	
Comentarios		

Tabla 4.31: CU-5 Extraer texto

CU-6	**Tokenizar texto**	
Actores	Sistema	
Descripción	El sistema procesa una cadena de texto de entrada para separarla en tokens individuales, tratando las palabras y los signos de puntuación como elementos independientes. Este proceso normaliza el texto eliminando espacios redundantes y facilita el análisis posterior en tareas de procesamiento de lenguaje natural.	
Dependencias	CU-1, CU-5	
Precondición	● Se recibe una cadena de texto válida como parámetro de entrada. ● El texto puede contener palabras, espacios, signos de puntuación y caracteres especiales. ● El sistema tiene acceso a las funciones de procesamiento de cadenas.	
Secuencia normal	**Paso**	**Descripción**
	P1	Recibir la cadena de texto como parámetro de entrada.
	P2	Normalizar espacios múltiples reemplazándolos por un solo espacio.
	P3	Identificar los signos de puntuación comunes en el texto.
	P4	Separar cada signo de puntuación agregando espacios antes y después.
	P5	Separar el texto procesado en tokens y eliminar elementos vacíos o solo espacios adicionales.
	P6	Retornar una lista limpia de tokens.

Postcondición	● Retorna una lista de strings donde cada elemento es un token válido. ● Cada palabra aparece como un token independiente. ● Cada signo de puntuación aparece como un token separado. ● No existen tokens vacíos o que contengan solo espacios. ● El orden original de las palabras y puntuación se mantiene.
Excepciones	● Parámetro de entrada inválido. ● Si recibe una cadena vacía retorna una lista vacía. ● Fallo en la conversión.
Comentarios	

Tabla 4.32: CU-6 Tokenizar texto

CU-7	**Comparar textos y detectar diferencias**
Actores	Sistema
Descripción	El sistema compara dos textos a nivel de tokens para detectar diferencias, identificando qué tokens fueron eliminados, agregados o permanecen sin cambios. Utiliza tokenización avanzada que trata signos de puntuación como elementos separados y el algoritmo SequenceMatcher [23] para encontrar bloques coincidentes entre las secuencias de texto.
Dependencias	CU-1, CU-6
Precondición	● Se reciben dos cadenas de texto válidas (texto1, texto2) como parámetros: ○ texto1: texto del archivo borrador. ○ texto2: texto del archivo firmado.

Secuencia normal	**Paso**	**Descripción**
	P1	Recibir las dos cadenas de texto como parámetro de entrada.
	P2	Realizar CU-5 para poder obtener la lista de tokens del primer y segundo archivo.
	P3	Obtener los bloques coincidentes usando machet.get_matching_blocks().
	P4	Para cada bloque coincidente, procesar los tokens en tres fases: ● Marcar tokens de texto1 (archivo borrador) no coincidentes como 'rojo' (eliminados). ● Marcar tokens de texto2 (archivo firmado) no coincidentes como 'amarillo' (agregados). ● Marca tokens coincidentes como 'ninguno' en ambas listas.
	P5	Procesar los tokens restantes.
	P6	Retornar una lista de tuplas con el valor del token y si está marcado en amarillo o no.
Postcondición	**Éxito:** Retorna dos listas de tuplas (token, color) para cada texto: ● **resultado1:** Contiene todos los tokens del texto1 con sus estados de comparación. ● **resultado2:** Contiene todos los tokens del texto2 con sus estados de comparación. ● Los colores indican: 'rojo' = eliminado, 'amarillo' = agregado, 'ninguno' = sin cambios. ● Se preserva el orden original de tokens en ambos textos.	
Excepciones	● Parámetro de entrada inválidos. ● Error en la tokenización.	

Comentarios	

Tabla 4.33: CU-7 Comparar textos y detectar diferencias

CU-8	**Separar puntuación**	
Actores	Sistema	
Descripción	Insertar espacios alrededor de los signos de puntuación en un texto.	
Dependencias	CU-1	
Precondición	El texto de entrada es una cadena de caracteres (str).	
Secuencia normal	**Paso**	**Descripción**
	P1	Recibir el texto como parámetro de entrada.
	P2	Para cada signo, se inserta un espacio antes y después, asegurando su separación.
	P3	Dividir en una lista de tokens separando por espacios.
	P4	Retornar un listado de tokens.
Postcondición	Se devuelve una lista de palabras y signos de puntuación como elementos individuales.	
Excepciones		
Comentarios	A diferencia de la función “tokenizar_texto”, esta función no normaliza espacios múltiples ni filtra tokens vacíos.	

Tabla 4.34: CU-8: Separar puntuación

<table>
<tr><td>CU-9</td><td colspan="2">Procesar palabras</td></tr>
<tr><td>Actores</td><td colspan="2">Sistema</td></tr>
<tr><td>Descripción</td><td colspan="2">Procesa una lista de palabras extraídas de documentos PDF separando los signos de puntuación de las palabras y calcula las coordenadas aproximadas para cada nuevo token generado. Esto permite localizar con precisión cada palabra individual en el PDF para poder subrayarlas usando las coordenadas exactas de posicionamiento.</td></tr>
<tr><td>Dependencias</td><td colspan="2">CU-8</td></tr>
<tr><td>Precondición</td><td colspan="2">La lista de palabras de entrada debe tener tuplas con formato:
(x0, y0, x1, y1, text, block, line, word_in_line)
donde:
● x0, y0, x1, y1 (float): coordenadas del área que ocupa la palabra en la página.
● text (str): texto de la palabra.
● block (int): número de bloque (estructura del documento).
● line (int): número de línea dentro del bloque.
● word_in_line (int): posición de la palabra en la línea.</td></tr>
<tr><td rowspan="4">Secuencia normal</td><td>Paso</td><td>Descripción</td></tr>
<tr><td>P1</td><td>El sistema recibe una lista de tuplas con información de palabras extraídas del documento.</td></tr>
<tr><td>P2</td><td>Para cada palabra en la lista, desempaqueta las coordenadas (x0, y0, x1, y1), texto y metadatos (block, line, word_in_line).</td></tr>
<tr><td>P3</td><td>Usa la función “separar_puntuacion” para dividir el texto en tokens individuales (palabras y signos de puntuación).</td></tr>
</table>

	P4	Calcula la longitud promedio de carácter dividiendo el ancho total entre la longitud del texto.
	P5	Para cada token generado, calcula nuevas coordenadas horizontales basadas en la longitud del token y la posición relativa.
	P6	Crea una nueva tupla para cada token con coordenadas recalculadas, manteniendo coordenadas verticales y metadatos originales.
Postcondición	Se genera una lista de tuplas donde cada token (palabra o signo de puntuación) tiene sus propias coordenadas.	
Excepciones		
Comentarios		

Tabla 4.35: CU-9 Procesar palabras

CU-10	**Comparar archivos**
Actores	Sistema
Descripción	Recorre todas las subcarpetas de un directorio base, identifica archivos con palabras clave específicas (informe, borrador, convenio), extrae y valida códigos de los nombres de archivo comparándolos con el nombre de la carpeta contenedora, procesa el contenido textual de cada archivo encontrado, compara textos entre documentos, identifica diferencias mediante tokenización y aplica subrayado automático a las discrepancias encontradas en los PDFs.
Dependencias	CU-1, CU-4, CU-5, CU-6, CU-7, CU-8, CU-9

Precondición	El directorio de la carpeta base debe existir y ser accesible.	
Secuencia normal	**Paso**	**Descripción**
	P1	Recibir el nombre de la carpeta base como parámetros.
	P2	Recorrer cada subcarpeta dentro de la carpeta principal.
	P3	En cada carpeta detectar archivos por código mediante el caso de uso CU-4.
	P4	Identificar los pares de archivos “borrador”, “firmado” o “convenio” por nombre y extensión (.pdf).
	P5	Extraer y comparar el texto de ambos archivos usando los casos de uso CU-6 y CU-7.
	P6	Resaltar en el PDF firmado los tokens que en el paso anterior se marcaron como “amarillo”.
	P7	Guardar un nuevo PDF informe con los cambios resaltados y registrar un informe del proceso.
	P8	Devuelve una lista de tuplas con la información de los informes generados.
Postcondición	● Todos los archivos han sido procesados y analizados. ● Todas las modificaciones han sido resaltadas en amarillo. ● Se genera un registro de archivos procesados y diferencias encontradas.	
Excepciones	Si los códigos no coinciden entre archivo y carpeta no se realizará la comparación.	
Comentarios		

Tabla 4.36: CU-10 Comparar archivos

CU-11	**Crear informe resumen**	
Actores	Sistema	
Descripción	El sistema genera un informe en formato Word (.docx) que contiene un resumen completo de los resultados de comparación de archivos PDF. El informe incluye una tabla con los datos de cada comparación realizada, destacando visualmente la presencia o ausencia de cambios mediante colores diferenciados.	
Dependencias	CU-1	
Precondición	● La carpeta de destino para guardar el informe existe y tiene permisos de escritura. ● La lista de informes contiene al menos un elemento válido. ● Cada tupla de informe tiene el formato correcto. ● Las librerías de generación de documentos Word están disponibles.	
Secuencia normal	**Paso**	**Descripción**
	P1	Recibir la ruta de la carpeta destino y la lista de informes.
	P2	Crear un nuevo documento Word.
	P3	Generar una tabla con encabezados: Nombre del archivo, Total de cambios, Cambios de 1 carácter.
	P4	Para cada informe en la lista, crear una fila con los datos y aplicar formato de color (rojo si hay cambios, verde si no hay cambios).
	P5	Generar el nombre del archivo con formato de fecha y hora:

		AAAAMMDD_HHMM_Informe_comparacion_carpetas.docx
	P6	Guardar el documento en la carpeta especificada y retorna la ruta completa del archivo generado.
Postcondición	● Se ha creado un archivo .docx en la carpeta especificada. ● El archivo contiene un informe completo con tabla formateada de todos los resultados de comparación. ● Las diferencias están resaltadas visualmente con colores apropiados. ● Se retorna la ruta completa del archivo generado.	
Excepciones	● Si la carpeta destino no existe o no tiene permisos de escritura, se genera error de acceso.	
Comentarios	● El formato visual del informe facilita la identificación rápida de archivos con diferencias.	

Tabla 4.37: CU-11 Crear informe resumen

CU-12	**Registrar ejecución**
Actores	Sistema
Descripción	El sistema registra la información completa de la ejecución del cuaderno de comparación de archivos PDF en un archivo Excel (.xlsx).
Dependencias	CU-1
Precondición	● La lista de resultados contiene los datos de los informes procesados. ● El tiempo total de ejecución ha sido calculado correctamente.

	● La carpeta base de procesamiento es válida y accesible. ● Las librerías de manipulación de archivos Excel están disponibles. ● Existe permiso de escritura en la carpeta destino.	
Secuencia normal	**Paso**	**Descripción**
	P1	El sistema recibe la lista de resultados, tiempo total de ejecución y carpeta base procesada.
	P2	Crea un nuevo archivo Excel.
	P3	Por cada carpeta procesada añadir una línea. Cada línea contiene los campos: Fecha de ejecución, nombre carpeta, código identificador, documento word (si existe), archivo borrador, archivo firmado, informe creado, número de cambios, número de cambios de un solo carácter. Al final se añade una línea resumen con los datos del número de carpetas procesadas, el tiempo total que ha tardado en analizar todas las carpetas y el nombre del cuaderno.
	P4	Guardar el archivo Excel con nombre que incluye.
	P5	Retornar la ruta completa del archivo generado.
Postcondición	● Se ha creado un archivo .xlsx con registro completo de la ejecución. ● El archivo contiene tres hojas con información estructurada y métricas calculadas. ● Los datos están organizados en formato tabular para análisis posterior. ● Retorna la ruta completa del archivo Excel generado.	
Excepciones		

Comentarios	

Tabla 4.38: CU-12 Registrar ejecución

CU-13	**Usar la aplicación**	
Actores	Usuario	
Descripción	El usuario ejecuta la aplicación de comparación de documentos desarrollada en Google Colab para analizar diferencias entre carpetas de documentos PDF mediante la ejecución secuencial de tres celdas de código.	
Dependencias	CU-1	
Precondición	• El usuario tiene acceso al cuaderno de Google Colab. • El usuario tiene una cuenta de Google activa. • Las carpetas con documentos PDF están disponibles en Google Drive o almacenamiento local.	
Secuencia normal	**Paso**	**Descripción**
	P1	El usuario accede al cuaderno de Google Colab.
	P2	El usuario ejecuta la primera celda “Conectarse a Google” haciendo clic en el botón de “play”.
	P3	El sistema solicita autenticación a Google Drive mostrando una ventana emergente solicitando permisos de acceso.
	P4	El usuario selecciona la cuenta de Google que desea utilizar.
	P5	El sistema presenta la pantalla de autorización de permisos para acceder a Google Drive.

	P6	El usuario revisa los permisos solicitados (lectura de archivos de Drive).
	P7	El usuario hace clic en “Permitir” o “Autorizar” para conceder los permisos.
	P8	El usuario ejecuta la segunda celda “Configurar cuaderno” haciendo clic en el botón de “play”.
	P9	El sistema instala dependencias y configura el entorno de trabajo.
	P10	El usuario ejecuta la tercera celda “Comparar carpetas” haciendo clic en el botón de “play”.
	P11	El sistema procesa los documentos y muestra los resultados de la comparación.
Postcondición	● La aplicación ha ejecutado exitosamente la comparación de documentos. ● Los resultados están disponibles para su visualización en el cuaderno. ● El entorno queda configurado para futuras ejecuciones.	
Excepciones	● Error de autenticación con Google: el usuario debe volver a intentar la conexión. ● Carpetas no encontradas: verificar la ruta de las carpetas especificadas.	
Comentarios	La aplicación está diseñada para ser ejecutada de forma secuencial, donde cada celda depende de la ejecución exitosa de la anterior. El usuario no requiere conocimientos de programación, solo debe ejecutar las celdas en orden.	

Tabla 4.39: CU-13 Usar la aplicación

4.3.- DISEÑO

Es importante destacar que para este proyecto no se ha desarrollado una interfaz gráfica propia. En su lugar, se ha utilizado directamente la interfaz nativa de Google Colaboratory. Esta decisión responde a criterios de eficiencia, accesibilidad y funcionalidad, aprovechando las capacidades que ofrece la plataforma para la ejecución de código Python.

A pesar de no contar con una interfaz personalizada, se ha puesto especial énfasis en lograr que la experiencia de usuario sea lo más intuitiva posible. Para ello, se ha estructurado el cuaderno de manera que ofrezca una distribución sencilla y limpia, facilitando el acceso a las diferentes funcionalidades del sistema de manera ágil y eficiente, sin necesidad de navegar a través de múltiples páginas o documentos.

En lugar de una navegación compleja, se ha optado por implementar un flujo de trabajo lineal organizado en secciones claramente numeradas y tituladas que permanecen visibles en el cuaderno. Cada sección representa una fase específica del proceso, y al ejecutar cada una de ellas mediante el botón correspondiente, el contenido y los resultados se muestran de manera dinámica justo debajo de la celda ejecutada.

Esta distribución coherente y consistente a lo largo de todo el cuaderno aporta una sensación de familiaridad y facilita enormemente la operación para el usuario, incluso para aquellos con conocimientos técnicos limitados.

4.3.1.- Estructura de la interfaz

El diseño está organizado en tres secciones principales que siguen un orden lógico de ejecución:

1. **Conectarse a Google Drive**: esta sección inicial establece la conexión con el almacenamiento en la nube del usuario, permitiendo el acceso a los archivos PDF que serán analizados posteriormente.

2. **Configurar cuaderno**: este apartado prepara el entorno de ejecución, estableciendo los parámetros necesarios, cargando las librerías y las funciones

requeridas para el correcto funcionamiento del comparador.

3. **Comparar carpetas**: la sección principal de la aplicación, donde se implementa la funcionalidad de comparación de documentos PDF, procesando y mostrando los resultados del análisis.

Al utilizar la interfaz nativa de Google Colaboratory, el proyecto aprovecha cuatro tipos principales de elementos interactivos que ofrece la plataforma:

1. **Botones de ejecución**: elementos estándar de Colaboratory que permiten activar cada bloque funcional siguiendo la secuencia lógica del proceso.

2. **Controles de visualización de código**: funcionalidad nativa "Mostrar código" que ofrece la posibilidad de mostrar u ocultar el código Python subyacente, adaptándose así a usuarios con diferentes niveles de conocimiento técnico.

3. **Controles de salida**: opciones integradas como "Mostrar salida oculta" que permiten gestionar la visibilidad de los resultados generados, especialmente útil cuando estos son extensos.

4. **Mensajes de confirmación**: salidas de texto generadas por el código que proporcionan retroalimentación inmediata sobre el estado de las operaciones realizadas, como "Google Drive montado correctamente".

Este diseño implementa varios principios fundamentales de diseño de interfaces:

- **Simplicidad**: se reduce la interfaz a los elementos esenciales, eliminando distracciones y facilitando el enfoque en la tarea principal.

- **Progresividad**: organización secuencial que guía al usuario paso a paso a través del proceso completo.

- **Consistencia**: se mantiene un estilo uniforme en todos los elementos interactivos y secciones.

- **Retroalimentación**: cada acción del usuario genera una respuesta visual que confirma el resultado de la operación.

- **Flexibilidad**: la interfaz se adapta a diferentes perfiles de usuario, permitiendo

mostrar u ocultar elementos técnicos según se requiera.

4.3.2.- Justificación del diseño

La decisión de utilizar Google Colaboratory como plataforma base, en lugar de desarrollar una interfaz gráfica propia, responde a una estrategia técnica y funcional fundamentada en una serie de criterios técnicos y funcionales.

Las siguientes ventajas justifican la elección de esta plataforma como base del desarrollo, optimizando recursos y maximizando la accesibilidad del sistema:

- **Eficiencia en el desarrollo**: concentrar los esfuerzos en la funcionalidad core de comparación de PDFs en lugar de invertir tiempo en el desarrollo de una interfaz personalizada.
- **Accesibilidad inmediata**: permitir que cualquier usuario con una cuenta de Google pueda utilizar la herramienta sin instalaciones adicionales ni necesidad de familiarizarse con una nueva interfaz.
- **Escalabilidad futura**: establecer una base funcional sólida que, en iteraciones futuras del proyecto, podría evolucionar hacia una aplicación con interfaz gráfica personalizada.
- **Facilidad de uso**: a pesar de no contar con una interfaz personalizada, se ha estructurado el cuaderno para que usuarios sin conocimientos técnicos avanzados puedan aprovechar la funcionalidad.

Para garantizar el correcto funcionamiento, se han implementado las siguientes estrategias técnicas específicas que optimizan el uso de los recursos disponibles y mejoran la experiencia del usuario final:

- **Aprovechamiento de funcionalidades existentes**: la conexión directa con el almacenamiento en la nube elimina la necesidad de cargas manuales de archivos, aprovechando las capacidades integradas de Google Colaboratory.
- **Modularidad del código**: a pesar de no contar con una interfaz propia, se ha organizado el código en bloques funcionales independientes que facilitan el mantenimiento y la extensibilidad.

- **Aprovechamiento de controles nativos**: se utilizan los controles de visibilidad propios de Google Colaboratory para gestionar la exposición del código y los resultados, adaptando así la experiencia según el perfil técnico del usuario.

- **Flujo secuencial estructurado**: se ha estructurado el cuaderno para que siga una secuencia lógica clara, reduciendo el tiempo necesario para completar el proceso de comparación, incluso sin disponer de elementos de navegación personalizados.

Este enfoque pragmático prioriza la funcionalidad sobre la estética, proporcionando una herramienta efectiva para la comparación de documentos PDF en un entorno ya familiar para muchos usuarios, sin las complicaciones que podría suponer el desarrollo de una interfaz propia.

4.3.3.- Flujo de trabajo

El flujo de trabajo diseñado sigue una estructura intuitiva:

1. El usuario accede al cuaderno de Google Colab “Comparador PDF”.
2. Ejecuta la primera sección para conectarse a Google Drive.
3. Continúa con la configuración del cuaderno mediante la ejecución de la segunda sección.
4. Finalmente, utiliza la funcionalidad principal de comparación de carpetas para analizar sus archivos PDF.
5. Visualiza los resultados del análisis directamente en el cuaderno.

El siguiente diagrama de flujo representa de manera global el diseño de la aplicación mostrando la secuencia lógica de pasos que debe seguir el usuario para utilizar correctamente la aplicación. Esta secuencia garantiza que todas las dependencias necesarias estén satisfechas antes de ejecutar la funcionalidad principal, minimizando la posibilidad de errores.

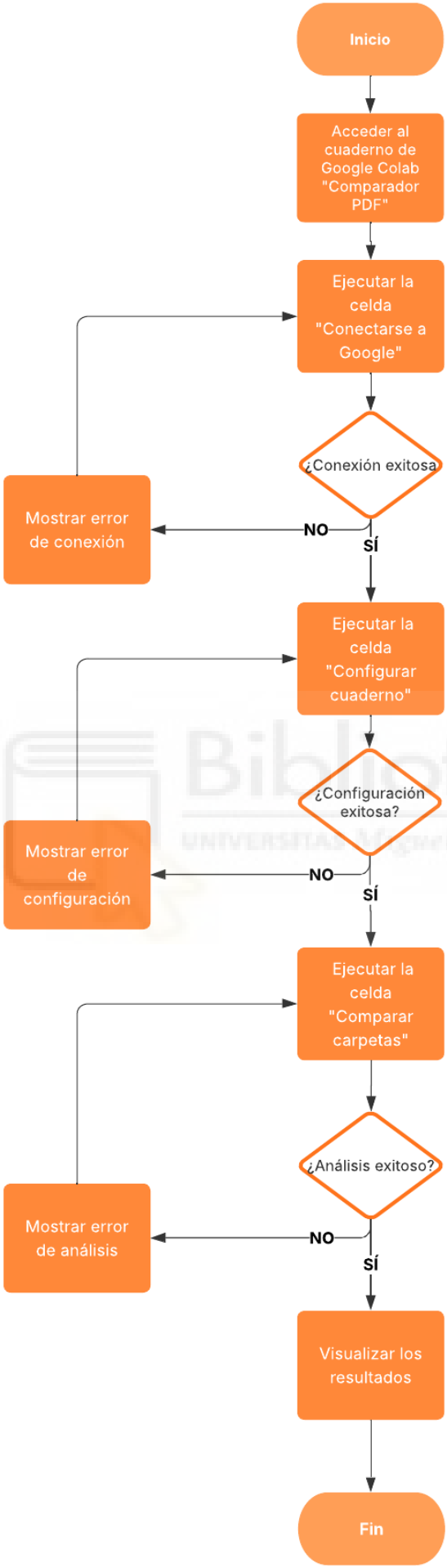

Imagen 4.5: Diagrama de flujo

4.3.4.- Accesibilidad y usabilidad

Aunque no se ha desarrollado una interfaz propia, la implementación en Google Colabaratory incorpora principios fundamentales de accesibilidad y usabilidad que facilitan su uso por diversos tipos de usuarios:

Accesibilidad:

- **Acceso universal**: la aplicación es accesible desde cualquier dispositivo con navegador web actualizado, independientemente del sistema operativo.
- **Sin barreras de instalación**: no requiere instalación de software especializado ni configuraciones técnicas complejas.
- **Rendimiento optimizado**: funciona adecuadamente incluso en equipos con recursos limitados, gracias al procesamiento en servidores de Google.
- **Flexibilidad de interacción**: permite el uso completo tanto con ratón como exclusivamente con teclado, adaptándose a diferentes necesidades y preferencias.

Usabilidad:

- **Estructura intuitiva**: organización mediante secciones numeradas con un flujo de trabajo lógico y secuencial.
- **Feedback inmediato**: cada acción genera una respuesta visual que confirma su ejecución o informa sobre errores.
- **Simplicidad de uso**: reduce la carga cognitiva mediante un flujo de trabajo lineal y claramente definido.
- **Lenguaje claro**: utiliza encabezados descriptivos que comunican explícitamente la función de cada sección.
- **Transparencia operativa**: mantiene al usuario informado sobre el estado del sistema mediante mensajes informativos.
- **Flexibilidad y control**: permite ejecutar cualquier paso o modificar parámetros

en cualquier momento.

- **Diseño coherente**: mantiene un uso consistente de elementos interactivos a lo largo de toda la aplicación.

Estas características permiten que la herramienta sea utilizada por un amplio espectro de usuarios con diferentes niveles de experiencia técnica o necesidades específicas, a pesar de las limitaciones inherentes al uso de la interfaz predefinida de Google Colaboratory.

4.4.- IMPLEMENTACIÓN

Este apartado no solo presenta el resultado final, sino que también expone el razonamiento detrás de las decisiones técnicas adoptadas, ofreciendo así una visión transparente del proceso de desarrollo. A continuación, se proporcionará una descripción específica y técnica de cómo se ha construido la solución propuesta detallando los componentes principales del sistema junto con los algoritmos y estructuras de datos más relevantes.

4.4.1-. Bibliotecas y dependencias usadas para el desarrollo

La implementación del sistema requiere diversas bibliotecas especializadas para el manejo de documentos, procesamiento de datos y creación de interfaces de usuario. Se distingue entre bibliotecas externas que requieren instalación manual y dependencias del sistema que están preinstaladas. Se han seleccionado las siguientes bibliotecas especializadas que abordan los diferentes aspectos del proyecto.

4.4.1.1.- Bibliotecas externas

Las siguientes bibliotecas requieren instalación manual mediante pip:

Biblioteca	Categoría	Función principal
PyPDF2 [25]	Procesamiento de documentos	Lectura y procesamiento de documentos PDF para extracción de texto.
PyMuPDF (fitz) [26]	Procesamiento de documentos	Al ser una biblioteca más avanzada permite el procesamiento alternativo de PDFs cuando PyPDF2 presenta limitaciones.
Python-docx [27]	Procesamiento de documentos	Creación, manipulación de documentos en formato Word (.docx) y extracción de texto estructurado permitiendo la conservación del formato y estilo.
FPDF [17]	Procesamiento de documentos	Creación de documentos en formato PDF.

Tabla 4.40: Bibliotecas externas

4.4.1.2.- Dependencias del sistema

Las siguientes dependencias están preinstaladas en Google Colaboratory o forman parte de la biblioteca estándar de Python:

Biblioteca	Categoría	Función principal
Pandas [28]	Análisis de datos	Útil para la organización y procesamiento estructurado de los datos extraídos de los documentos, facilitando su análisis y comparación.
NumPy [29]	Análisis de datos	Operaciones matemáticas y manejo de arrays multidimensionales.

OpenPyXL [30]	Hojas de cálculo	Procesamiento de hojas de cálculo, formateo de celdas y generación de archivos Excel (.xlsx).
Google-auth [31]	Google Services	Autenticación y gestión de credenciales para Google Drive.
Googleapiclient [32]	Google Services	Búsqueda de documentos en Google Drive.
Difflib [15]	Utilidades	implementa algoritmos para comparar secuencias, particularmente útil para identificar diferencias precisas entre textos.
Re [33]	Utilidades	Procesamiento y limpieza de texto mediante patrones.
Pathlib [34]	Utilidades	Gestión de rutas de archivos.
Datetime [35]	Utilidades	Permite manejar fechas y horas.
Pytz [36]	Utilidades	Gestión de zonas horarias para timestamp.
Os [37]	Utilidades	Operaciones con archivos y directorios.
Io [38]	Utilidades	Procesamiento de datos en memoria (BytesIO).
Itertools [39]	Utilidades	Operaciones avanzadas con iteradores y combinaciones.
Sys [40]	Utilidades	Control del comportamiento del intérprete Python.
Contextlib [41]	Utilidades	Gestión de contextos de ejecución.
Getpass [42]	Utilidades	Manejo seguro de credenciales de usuario.
Time [43]	Utilidades	Control de tiempos de ejecución y pausas

Tabla 4.41: Dependencias del sistema

Estas bibliotecas, en conjunto, proporcionan todas las funcionalidades necesarias para implementar este sistema desarrollado, abordando los desafíos específicos identificados en los requisitos del proyecto.

4.4.2.- Configuración del entorno para el proyecto

Para adaptar Google Colaboratory a las necesidades específicas del proyecto, se ha realizado la siguiente configuración:

- **Instalación de bibliotecas adicionales**: aunque Google Colaboratory incluye numerosas bibliotecas preinstaladas algunas requieren instalación adicional mediante comandos pip, que se han incorporado en las primeras celdas del cuaderno.

- **Persistencia de datos**: para garantizar la persistencia de archivos temporales y configuraciones entre sesiones, se ha implementado una conexión con Google Drive como almacenamiento persistente, montándolo como un directorio dentro del entorno de Google Colaboratory.

- **Gestión de credenciales**: se ha implementado un sistema seguro para la gestión de credenciales de acceso a la API de Google Drive, utilizando archivos de configuración almacenados en una ubicación protegida dentro de Google Drive.

- **Optimización de recursos**: se han implementado estrategias para optimizar el uso de memoria y procesador, considerando las limitaciones de recursos gratuitos que proporciona Google Colaboratory, especialmente para el procesamiento de documentos extensos.

- **Estructuración modular del código**: el cuaderno principal se ha organizado en secciones funcionales claramente delimitadas, con funciones auxiliares importadas desde módulos Python separados para mantener la claridad y facilitar el mantenimiento.

4.4.3.- Integración con Google Drive y gestión de archivos

Un aspecto fundamental del proyecto es la integración eficiente con Google Drive, que se ha implementado de la siguiente manera:

- **Autenticación con OAuth 2.0**: se utiliza el protocolo OAuth 2.0 [46] para la autenticación segura con la API de Google [45], permitiendo acceder a los documentos del usuario sin almacenar credenciales sensibles.

- **Montaje directo del Drive**: mediante la biblioteca google.colab [47], se monta Google Drive directamente como un sistema de archivos accesible desde el entorno, facilitando la manipulación directa de archivos.

- **API de Google Drive**: para operaciones más complejas como búsquedas avanzadas o gestión de metadatos, se utiliza la API oficial de Google Drive [45] a través de la biblioteca googleapiclient [32].

- **Estructura de carpetas optimizada**: Se ha diseñado una estructura de carpetas específica dentro de Google Drive para organizar eficientemente los documentos originales, las versiones comparadas y los informes generados.

Esta integración robusta con Google Drive garantiza que la solución se adapte perfectamente al flujo de trabajo existente en la institución, minimizando los cambios necesarios en las prácticas habituales de los usuarios.

4.4.4.- Manejo de errores

Se ha implementado un sistema robusto de manejo de errores que garantiza la continuidad del procesamiento incluso ante fallos individuales. El sistema está diseñado bajo el principio de "tolerancia a fallos", donde los errores en archivos específicos no interrumpen la ejecución completa del proceso. Como características del sistema de manejo de errores podemos destacar las siguientes:

1. **Validación de formatos**

El sistema implementa un conjunto robusto de validaciones previas al procesamiento de documentos:

- **Verificación de formatos**: validación automática de extensiones de archivo para garantizar compatibilidad con el sistema.
- **Detección de integridad**: identificación temprana de archivos corruptos o incompatibles que podrían comprometer el procesamiento.

2. **Gestión de excepciones en tiempo real**

La aplicación proporciona retroalimentación continua mediante un sistema de notificaciones estructurado por fases operativas:

- **Fase de carga**: seguimiento del progreso de carga con indicadores específicos para cada documento.
- **Fase de análisis**: monitorización del estado de procesamiento con reportes de avance en tiempo real.
- **Fase de conversión**: notificaciones inmediatas sobre el éxito o fallo de las transformaciones aplicadas.
- **Fase de generación**: confirmaciones automáticas de la creación exitosa de reportes y documentos finales.

3. **Continuidad de ejecución**

Una característica fundamental del sistema es su capacidad de mantener la ejecución completa incluso cuando archivos individuales presentan problemas, así como:

- Los archivos que no pueden ser convertidos a un formato específico.
- Si un archivo no puede ser leído por problemas de formato, el sistema registra el error y continúa con el siguiente archivo.
- El sistema mantiene un contador de carpetas procesadas exitosamente.

4. **Mensajes informativos progresivos**

   Durante toda la ejecución, el usuario recibe información detallada sobre:

   - Descripción específica de las operaciones que se están ejecutando en cada momento.
   - Registro de las acciones completadas exitosamente.
   - Notificación inmediata de cualquier error o problema detectado.

Esta implementación asegura que los usuarios puedan procesar grandes lotes de documentos sin preocuparse por que un archivo problemático detenga todo el proceso, manteniendo al mismo tiempo una trazabilidad completa de los resultados obtenidos.

4.5.- PRUEBAS

Para validar el correcto funcionamiento y la robustez del programa desarrollado, se ha diseñado un sistema de pruebas exhaustivo basado en la creación de un entorno de testing controlado. Este entorno consistió en crear 8 carpetas principales, cada una con diferentes subcarpetas específicamente para simular diferentes escenarios de uso real y casos extremos que el programa podría encontrar durante su operación en producción.

El diseño de las pruebas se fundamentó en la modificación sistemática de los nombres de archivos dentro de cada carpeta y subcarpeta, permitiendo evaluar diferentes aspectos críticos del sistema:

- **Verificación de reconocimiento de archivos**: se ha verificado que el programa identificase correctamente los nombres de archivos en diferentes formatos y estructuras de nomenclatura, incluyendo archivos con espacios, guiones, etc.

- **Verificación de códigos**: las pruebas incluyen archivos con diferentes patrones de codificación para asegurar que el algoritmo de reconocimiento funcione de manera robusta ante distintas convenciones de nomenclatura.

- **Conversión de formatos**: las pruebas de conversión han permitido la validación del proceso de conversión de documentos DOCX a formato PDF manteniendo

tanto la integridad del contenido como la preservación del formato original durante la transformación.

Cada carpeta representa variaciones graduales de complejidad. Esta metodología incremental ha permitido identificar y corregir errores de manera sistemática, garantizando que cada funcionalidad del programa opera correctamente.

La estrategia de modificación progresiva de nombres de archivos y códigos identificativos ha proporcionado una cobertura exhaustiva de los posibles escenarios que el programa podría encontrar en un entorno de producción real.

A continuación, se muestra un ejemplo de ejecución y los resultados de salida:

1. En esta imagen se muestra la carpeta principal a analizar, llamada COMPARACIONCONVENIOS, la cual contiene 34 subcarpetas en las que se encontraran los archivos a analizar. En cada carpeta debe haber un borrador y un firmado/convenio para poder realizar la comparación.

Imagen 4.6: Estructura de carpetas

2. Como ya se ha explicado anteriormente la interfaz es muy intuitiva y para ejecutar la aplicación solamente hace falta darle a los 3 botones de “play” en el orden que indica.

Imagen 4.7: Interfaz de la aplicación

3. En todo momento se informa al usuario con mensajes del estado de la ejecución.

```
------------------------------------------------------------
Comprobando si existen archivos DOCX para convertir a PDF en la carpeta 384_25
No existen archivos DOCX para convertir a PDF
------------------------------------------------------------
Comprobando si existen archivos DOCX para convertir a PDF en la carpeta 395_25
No existen archivos DOCX para convertir a PDF
------------------------------------------------------------
Comprobando si existen archivos DOCX para convertir a PDF en la carpeta 397_25
No existen archivos DOCX para convertir a PDF
------------------------------------------------------------
Comprobando si existen archivos DOCX para convertir a PDF en la carpeta 403_25
No existen archivos DOCX para convertir a PDF
------------------------------------------------------------
Comprobando si existen archivos DOCX para convertir a PDF en la carpeta 319_25
No existen archivos DOCX para convertir a PDF
------------------------------------------------------------
Comprobando si existen archivos DOCX para convertir a PDF en la carpeta 2025_66888
Convirtiendo 2025_66888 borrador.docx a 2025_66888 borrador.pdf
------------------------------------------------------------
```

Imagen 4.8: Mensajes del estado de la ejecución (Conversión de formato)

```
------------Procesando la carpeta: 2025_14186 ------------
Código encontrado: 2025_14186
  - 2025-14186 Borrador.docx
  - 2025-14186 Convenio.pdf
  - 2025-14186 Borrador.pdf

Archivo borrador : 2025-14186 Borrador.pdf
Archivo firmado : 2025-14186 Convenio.pdf
Comparando 2025-14186 Borrador.pdf con 2025-14186 Convenio.pdf...

El PDF ha sido guardado como 2025_14186_Informe.pdf


------------Procesando la carpeta: 337_25 ------------
Código encontrado: C-0337_25
  - C-0337_25 borrador.pdf
  - C-0337_25.pdf

Archivo borrador : C-0337_25 borrador.pdf
Archivo firmado : C-0337_25.pdf
Comparando C-0337_25 borrador.pdf con C-0337_25.pdf...

El PDF ha sido guardado como C-0337_25_Informe.pdf
```

Imagen 4.9: Mensajes del estado de ejecución

4. Como resultado final se genera un documento Word con los resultados de la ejecución y un Excel con un registro de cada ejecución.

Informe de Comparación de Archivos PDF

Informe creado el 29/05/2025 a las 19:59

CARPETAS PROCESADAS:

Nombre del informe	Número de cambios	Cambios de 1 caracter
C-0422_25_Informe.pdf	873	305
C-0434_25_Informe.pdf	30	11
C-0417_25_Informe.pdf	0	0
C-0415_25_Informe.pdf	25	6
C-0075_25_Informe.pdf	28	6
C-0413_25_Informe.pdf	49	13
C-0431_25_Informe.pdf	34	11
C-0430_25_Informe.pdf	33	11
C-0411_25_Informe.pdf	22	3

Imagen 4.10: Informe Word con los resultados de la ejecución

	A	B	C	D	E	F	G	H	I	J	K	L
1	Fecha Acceso	Carpeta	Código	DOCX	Borrador	Firmado	Informe	Cambios	1 car	NumCarpProc	Tiempo total	Cuaderno
2	29/05/2025 19:59:19	422_25	C-0422_25		C-0422_25 borrador.pdf	C-0422_25-firmado.pdf	C-0422_25_Informe.pdf	873	305			
3	29/05/2025 19:59:19	434_25	C-0434_25		C-0434_25 borrador.pdf	C-0434-25-firmado.pdf	C-0434_25_Informe.pdf	30	11			
4	29/05/2025 19:59:19	417_25	C-0417_25		C-0417-25 borrador.pdf	C-0417-25-firmado.pdf	C-0417_25_Informe.pdf	0	0			
5	29/05/2025 19:59:19	415_25	C-0415_25		C-0415_25 borrador.pdf	C-0415-25-firmado.pdf	C-0415_25_Informe.pdf	25	6			
6	29/05/2025 19:59:19	75_25	C-0075_25		C-0075_25 borrador.pdf	C-0075_25-firmado.pdf	C-0075_25_Informe.pdf	28	6			
7	29/05/2025 19:59:19	413_25	C-0413_25		C-0413-25 borrador.pdf	C-0413-25-firmado.pdf	C-0413_25_Informe.pdf	49	13			
8	29/05/2025 19:59:19	431_25	C-0431_25		C-0431_25.pdf	C-0431_25-firmado.pdf	C-0431_25_Informe.pdf	34	11			
9	29/05/2025 19:59:19	430_25	C-0430_25		C-0430_25 borrador.pdf	C-0430-25-firmado.pdf	C-0430_25_Informe.pdf	33	11			
10	29/05/2025 19:59:19	411_25	C-0411_25		C-0411_25.pdf	C-0411_25-firmado.pdf	C-0411_25_Informe.pdf	22	3			
11	29/05/2025 19:59:19	414_25	C-0414_25		C-0414_25 borrador.pdf	C-0414-25-firmado.pdf	C-0414_25_Informe.pdf	33	11			
12	29/05/2025 19:59:19	387_25	C-0387_25		C-0387_25 borrador.pdf	C-0387-25-firmado.pdf	C-0387_25_Informe.pdf	415	52			
13	29/05/2025 19:59:19	398_25	C-0398_25		C-0398_25 borrador.pdf	C-0398_25.pdf	C-0398_25_Informe.pdf	0	0			
14	29/05/2025 19:59:19	405_25	C-0405_25		C-0405_25 borrador.pdf	C-0405_25.pdf	C-0405_25_Informe.pdf	0	0			
15	29/05/2025 19:59:19	406_25	C-0406_25		C-0406_25 borrador.pdf	C-0406-25-firmado.pdf	C-0406_25_Informe.pdf	372	49			
16	29/05/2025 19:59:19	408_25	C-0408_25		C-0408_25 borrador.pdf	C-0408-25-firmado.pdf	C-0408_25_Informe.pdf	31	11			
17	29/05/2025 19:59:19	410_25	C-0410_25		C-0410_25 borrador.pdf	C-0410-25-firmado.pdf	C-0410_25_Informe.pdf	33	11			
18	29/05/2025 19:59:19	384_25	C-0384_25		C-0384_25 borrador.pdf	C-0384_25.pdf	C-0384_25_Informe.pdf	0	0			
19	29/05/2025 19:59:19	395_25	C-0395_25		C-0395_25 borrador.pdf	C-0395-25-firmado.pdf	C-0395_25_Informe.pdf	20	3			
20	29/05/2025 19:59:19	397_25	C-0397_25		C-0397_25 borrador.pdf	C-0397-25-firmado.pdf	C-0397_25_Informe.pdf	11	4			
21	29/05/2025 19:59:19	403_25	C-0403_25		C-0403_25 borrador.pdf	C-0403_25-firmado.pdf	C-0403_25_Informe.pdf	55	18			
22	29/05/2025 19:59:19	319_25	C-0319_25		C-0319-25 borrador.pdf	C-0319_25.pdf	C-0319_25_Informe.pdf	0	0			
23	29/05/2025 19:59:19	2025_66888	2025_66888	2025_66888 borrador.docx	2025_66888 borrador.pdf	2025_66888 firmado.pdf	2025_66888_Informe.pdf	502	139			
24	29/05/2025 19:59:19	333_25	C-0333_25		C-0333_25 borrador.pdf	C-0333_25.pdf	C-0333_25_Informe.pdf	0	0			
25	29/05/2025 19:59:19	2025_14186	2025_14186	2025-14186 Borrador.docx	2025-14186 Borrador.pdf	2025-14186 Convenio.pdf	2025_14186_Informe.pdf	0	0			
26	29/05/2025 19:59:19	337_25	C-0337_25		C-0337_25 borrador.pdf	C-0337_25.pdf	C-0337_25_Informe.pdf	0	0			
27	29/05/2025 19:59:19	353_25	C-0353_25		C-0353_25 borrador.pdf	C-0353_25.pdf	C-0353_25_Informe.pdf	0	0			
28	29/05/2025 19:59:19	295_25	C-0295_25		C-0295_25 borrador.pdf	C-0295_25-firmado.pdf	C-0295_25_Informe.pdf	267	66			
29	29/05/2025 19:59:19	306_25	C-0306_25		C-0306-25 borrador.pdf	C-0306_25-firmado.pdf	C-0306_25_Informe.pdf	20	6			
30	29/05/2025 19:59:19	305_25	C-0305_25		C-0305_25 borrador.pdf	C-0305_25-firmado.pdf	C-0305_25_Informe.pdf	62	11			
31	29/05/2025 19:59:19	342_25	C-0342_25		C-0342_25 borrador.pdf	C-0342_25.pdf	C-0342_25_Informe.pdf	0	0			
32	29/05/2025 19:59:19	01_25	01_25	01_25 borrador.docx	01_25 borrador.pdf	01_25-firmado.pdf	01_25_Informe.pdf	1479	335			
33	29/05/2025 19:59:19	133_25	C-0133_25		C-0133_25 borrador.pdf	C-0133_25.pdf	C-0133_25_Informe.pdf	0	0			
34	29/05/2025 19:59:19	182_25	C-0182_25		C-0182_25 borrador.pdf	C-0182_25-firmado.pdf	C-0182_25_Informe.pdf	27	6			
35	29/05/2025 19:59:19	2025_14165	2025_14165	2025-14165 Borrador.docx	2025-14165 Borrador.pdf	2025-14165 Convenio.pdf	2025_14165_Informe.pdf	0	0			
36	**29/05/2025 19:59:19**									**34**	**0:02:23**	**20250526-Comparador PDF v3.ipynb**
37												

Imagen 4.11: Informe Excel con el registro de todas las ejecuciones

Capítulo 5: Conclusiones y trabajo futuro

5.1.- CONCLUSIONES

El presente Trabajo de Fin de Grado ha alcanzado exitosamente su objetivo principal de desarrollar una aplicación automatizada para la comparación de archivos PDF utilizando Python como lenguaje de programación. Se ha logrado crear una solución integral que satisface tanto los requisitos funcionales como no funcionales establecidos al inicio del proyecto.

La solución implementada ha demostrado su eficacia práctica al conseguir reducir significativamente los tiempos de comparación manual de documentos PDF, transformando procesos que anteriormente requerían horas de revisión manual en operaciones automatizadas que se completan en cuestión de minutos. Esta mejora en la eficiencia temporal representa uno de los logros más destacados del proyecto, ya que permite a los usuarios dedicar su tiempo a tareas de mayor valor añadido.

La aplicación desarrollada representa una contribución significativa en el campo de la automatización de procesos documentales, proporcionando una herramienta robusta y eficiente para la comparación automatizada de documentos PDF.

La elección de la metodología Scrum como marco de trabajo ha demostrado ser acertada, facilitando un desarrollo iterativo e incremental que ha permitido la adaptación flexible a los cambios de requisitos y la entrega continua de funcionalidades operativas.

Google Colaboratory ha proporcionado un entorno de desarrollo excepcional que ha contribuido significativamente al éxito del proyecto. Su accesibilidad desde cualquier dispositivo con conexión a internet, junto con el acceso a recursos computacionales especializados, ha eliminado barreras técnicas que podrían haber limitado el alcance del desarrollo. La integración natural con Google Drive ha facilitado el manejo de archivos PDF y ha permitido implementar de manera eficiente las funcionalidades de acceso y procesamiento de documentos.

Desde una perspectiva personal, este Trabajo de Fin de Grado ha constituido una experiencia formativa invaluable que ha permitido la aplicación práctica de conocimientos teóricos en el desarrollo de una solución real y funcional. El dominio de Python para el procesamiento de documentos, la experiencia con metodologías ágiles y el trabajo en entornos de desarrollo colaborativo en la nube representan competencias

técnicas y profesionales que trascienden el ámbito académico.

La versatilidad demostrada por Python en el contexto de este proyecto confirma su posición como una herramienta fundamental para el desarrollo de aplicaciones de procesamiento de datos y automatización. La capacidad del lenguaje para integrar diferentes bibliotecas y servicios, junto con su sintaxis clara y expresiva, ha facilitado la implementación de funcionalidades complejas de manera eficiente y mantenible.

En definitiva, este Trabajo de Fin de Grado ha logrado desarrollar una aplicación para la automatización de detección de cambios en archivos PDF que no solo cumple con todos los objetivos establecidos, sino que también demuestra el potencial de las tecnologías actuales para automatizar procesos documentales complejos. La solución implementada representa una herramienta práctica y valiosa con aplicaciones directas en diversos sectores profesionales. El proyecto confirma la viabilidad de desarrollar soluciones robustas y escalables utilizando metodologías ágiles y tecnologías de desarrollo colaborativo en la nube, estableciendo una base técnica sólida que justifica la inversión en este tipo de desarrollos automatizados.

5.2.- POSIBLES DESARROLLOS FUTUROS

Las perspectivas futuras del proyecto son prometedoras y abren múltiples líneas de trabajo futuro que podrían expandir significativamente las capacidades de la aplicación desarrollada. La integración con otros servicios de almacenamiento en la nube representaría una evolución natural del sistema, lo que ampliaría considerablemente la base de usuarios potenciales y la flexibilidad de implementación en diferentes entornos organizacionales.

El desarrollo de una interfaz web completa proporcionaría mayor accesibilidad y usabilidad, eliminando la dependencia de entornos de desarrollo específicos como Google Colaboratory. Una aplicación web permitiría la implementación de funcionalidades adicionales como la gestión de usuarios, historiales de comparaciones, colaboración en tiempo real y acceso desde dispositivos móviles, transformando la herramienta en una solución empresarial más completa.

Bibliografía

[1] Google. (s.f.). *Google Drive: comparte archivos online con almacenamiento seguro en la nube*. Google Workspace. Disponible en: https://workspace.google.com/intl/es/products/drive/. Último acceso: 7 de mayo de 2025.

[2] Google. (s.f.). *Colab*. Colab. Disponible en: https://colab.google/ . Último acceso: 7 de mayo de 2025.

[3] Roy, A. y Emmott, E. (4 de septiembre de 2024). *Competitive Landscape: Intelligent Document Processing Platforms*. Gartner Research. Disponible en: https://www.gartner.com/en/documents/4705399. Último acceso: 7 de mayo de 2025.

[4] Adobe. (s.f.). *Encuentra tu plan ideal de Acrobat Pro DC*. Adobe Acrobat. Disponible en: https://www.adobe.com/es/acrobat/pricing.html. Último acceso: 8 de mayo de 2025.

[5] Microsoft. (s.f.). *Free Online Document Editing with Microsoft Word*. Microsoft 365. Disponible en: https://www.microsoft.com/es-es/microsoft-365/word. Último acceso: 8 de mayo de 2025.

[6] Kofax Store. (s.f.). *Kofax Power PDF Advanced 5 for Windows*. Kofax Store. Disponible en: https://kofaxstore.com/product/kofax-power-pdf-advanced-5/. Último acceso: 8 de mayo de 2025.

[7] Diffchecker. (s.f.). *Diffchecker - compara el texto para encontrar la diferencia entre dos archivos de texto*. Diffchecker. Disponible en: https://www.diffchecker.com/es/. Último acceso: 8 de mayo de 2025.

[8] Draftable. (s.f.). *Document Comparison Software*. Draftable. Disponible en: https://www.draftable.com/. Último acceso: 8 de mayo de 2025.

[9] Aspose. (s.f.). *Compare Documents Online*. Aspose.App. Disponible en: https://products.aspose.app/words/comparison. Último acceso: 8 de mayo de 2025.

[10] Text Compare! (s.f.). *Text Compare!* Text Compare. Disponible en: https://text-compare.com/. Último acceso: 8 de mayo de 2025.

[11] Blue Prism. (s.f.). *Agentic Automation Company | Leaders in Agentic AI, Automation, RPA & BPM*. Blue Prism. Disponible en: https://www.blueprism.com/es/. Último acceso: 10 de mayo de 2025.

[12] Van der Aalst, W. M., Bichler, M., & Heinzl, A. (2018). Robotic process automation. *Business & Information Systems Engineering*, 60(4), 269-272.

[13] Google for Developers. (s.f.). *Introducción a Python*. Google for Developers. Disponible en: https://developers.google.com/edu/python/introduction?hl=es-419. Último acceso: 10 de mayo de 2025.

[14] Analytics Vidhya. (9 de diciembre de 2024). *10 Advantages of Python Over Other Programming Languages*. Analytics Vidhya. Disponible en: https://www.analyticsvidhya.com/blog/2024/01/advantages-of-python-over-other-programming-languages/. Último acceso: 11 de mayo de 2025.

[15] Python Software Foundation. (2025). *difflib — Helpers for computing deltas*. Python Documentation. Disponible en: https://docs.python.org/es/3.13/library/difflib.html. Último acceso: 11 de mayo de 2025.

[16] Robinson, A., Becker, R., & ReportLab Team. (2025). *reportlab*. PyPI. Disponible en: https://pypi.org/project/reportlab/. Último acceso: 15 de mayo de 2025.

[17] FPDF. (s.f.). *FPDF: a free PHP class to generate PDF files*. FPDF. Disponible en: https://www.fpdf.org/. Último acceso: 16 de mayo de 2025.

[18] Gallardo, P. J. (15 de enero de 2024). *Los lenguajes de programación más demandados en 2024*. Hack a Boss. Disponible en: https://www.hackaboss.com/blog/lenguajes-programacion-mas-demandados. Último acceso: 20 de mayo de 2025.

[19] Project Jupyter. (s.f.). *Project Jupyter*. Jupyter.org. Disponible en: https://jupyter.org/. Último acceso: 20 de mayo de 2025.

[20] Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685

[21] Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J., Ouyang, C., ter Hofstede, A. H., Wynn, M. T., Meinert, M., Fidge, C., & Reijers, H. A. (2020). Robotic process automation: Contemporary themes and challenges. *Computers in Industry*, 115-162.

[22] Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Disponible en: https://www.scrumguides.org/scrum-guide.html. Último acceso: 22 de mayo de 2025.

[23] Python Software Foundation. (s.f.). *difflib — Funciones auxiliares para calcular deltas*. Python 3.9.19 documentation. Disponible en: https://docs.python.org/es/3.9/library/difflib.html. Último acceso: 22 de mayo de 2025.

[24] Python Software Foundation. (2024). *The Python Standard Library*. Python.org. Disponible en: https://docs.python.org/es/3.13/library/index.html. Último acceso: 22 de mayo de 2025.

[25] PyPI. (s.f.). *PyPDF2*. Python Package Index. Disponible en: https://pypi.org/project/PyPDF2/. Último acceso: 22 de mayo de 2025.

[26] PyMuPDF Development Team. (s.f.). *Welcome to PyMuPDF*. PyMuPDF Documentation. Disponible en: https://pymupdf.readthedocs.io/en/latest/. Último acceso: 22 de mayo de 2025.

[27] python-docx Development Team. (s.f.). *python-docx*. python-docx Documentation. Disponible en: https://python-docx.readthedocs.io/en/latest/. Último acceso: 25 de mayo de 2025.

[28] pandas Development Team. (s.f.). *About pandas*. pandas. Disponible en: https://pandas.pydata.org/about/index.html. Último acceso: 25 de mayo de 2025.

[29] NumPy. (s.f.). *NumPy*. NumPy. Disponible en: https://numpy.org/es/. Último acceso: 25 de mayo de 2025.

[30] Gazoni, E., & Clark, C. (29 de mayo de 2024). *openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files*. Read the Docs. Disponible en: https://openpyxl.readthedocs.io/en/stable/. Último acceso: 25 de mayo de 2025.

[31] Google. (s.f.). *Google Auth Library for Node.js* (versión 5.6.1). googleapis.dev. Disponible en: https://googleapis.dev/nodejs/google-auth-library/5.6.1/. Último acceso: 26 de mayo de 2025.

[32] Google for Developers. (s.f.). *API Client Libraries*. Google for Developers. Disponible en: https://developers.google.com/api-client-library?hl=es-419. Último acceso: 26 de mayo de 2025.

[33] Python Software Foundation. (s.f.). *re module — Regular expression operations.* Python Documentation. Disponible en: https://docs.python.org/es/3.13/library/re.html. Último acceso: 26 de mayo de 2025.

[34] Python Software Foundation. (s.f.). *The pathlib module — Object-oriented filesystem paths*. Python Documentation. Disponible en: https://docs.python.org/3/library/pathlib.html. Último acceso: 27 de mayo de 2025.

[35] Python Software Foundation. (s.f). *The datetime module — Basic date and time types*. Python Documentation. Disponible en: https://docs.python.org/3/library/datetime.html. Último acceso: 27 de mayo de 2025.

[36] Bishop, S. (24 de marzo de 2025). *pytz*. PyPI. Disponible en: https://pypi.org/project/pytz/. Último acceso: 27 de mayo de 2025.

[37] Python Software Foundation. (s.f.). *The os module — Miscellaneous operating system interfaces*. Python Documentation. Disponible en: https://docs.python.org/3/library/os.html. Último acceso: 27 de mayo de 2025.

[38] Python Software Foundation. (s.f.). *Módulo io — Herramientas fundamentales para trabajar con flujos de E/S*. Documentación de Python. Disponible en: https://docs.python.org/es/3.9/library/io.html. Último acceso: 28 de mayo de 2025

[39] Python Software Foundation. (s.f.). *Módulo itertools — Funciones para crear iteradores de forma eficiente*. Documentación de Python. Disponible en: https://docs.python.org/es/3.10/library/itertools.html#module-itertools. Último acceso: 28 de mayo de 2025.

[40] Python Software Foundation. (s.f.). *Módulo sys — Parámetros específicos del sistema*. Documentación de Python. Disponible en: https://docs.python.org/es/3.10/library/sys.html. Último acceso: 28 de mayo de 2025.

[41] Python Software Foundation. (s.f.). *Módulo contextlib — Utilidades para sentencias with*. Documentación de Python. Disponible en: https://docs.python.org/es/3/library/contextlib.html. Último acceso: 28 de mayo de 2025.

[42] Python Software Foundation. (s.f.). *Módulo getpass — Entrada de contraseña portátil*. Documentación de Python. Disponible en: https://docs.python.org/es/3.9/library/getpass.html. Último acceso: 28 de mayo de 2025.

[43] Python Software Foundation. (s.f.). *Módulo time — Acceso y conversiones de la hora*. Documentación de Python. Disponible en: https://docs.python.org/es/3.10/library/time.html. Último acceso: 28 de mayo de 2025.

[44] Google for Developers. (2024). *Google Drive API documentation*. Google. Disponible en: https://developers.google.com/drive/api. Último acceso: 2 de junio de 2025.

[45] Hardt, D. (2012). *The OAuth 2.0 authorization framework* (RFC 6749). Internet Engineering Task Force. Disponible en: https://datatracker.ietf.org/doc/html/rfc6749. Último acceso: 2 de junio de 2025.

[46] Auth0. (s.f.). *¿Qué es OAuth 2.0?*. Auth0. Disponible en: https://auth0.com/es/intro-to-iam/what-is-oauth-2. Último acceso: 2 de junio de 2025.

[47] admin. (17 de junio de 2019). *INTRODUCCIÓN A GOOGLE COLAB PARA DATA SCIENCE*. DataHack.es. Disponible en: https://auth0.com/es/intro-to-iam/what-is-oauth-2. Último acceso: 4 de junio de 2025.