



Facultad de Ciencias Económicas y Empresariales

OPTIMIZACIÓN DE PROCESOS EMPRESARIALES: LA AUTOMATIZACIÓN DE DATOS

Autor: Carlota Martínez Baena

Director: Jorge Aracil Jordá

MADRID | Junio 2024

RESUMEN

Nos encontramos en un mundo cada vez más dinámico y competitivo donde la tecnología ha revolucionado como trabajamos y a que dedicamos nuestro tiempo. Si antes empleábamos una gran parte de la jornada laboral en tareas rutinarias como la recolección de datos y la generación de informes, hoy la aparición de herramientas de automatización ha hecho que estas tareas se vuelvan cada vez más automatizadas y menos dependientes de la intervención humana directa. En este trabajo de investigación, se estudiará en primer lugar, el cambio en el rol del empleado como resultado de la implementación de sistemas automatizados, estudiando el impacto de herramientas como SQL y Google Scripts en la optimización de procesos de recolección, actualización y entrega de informes de datos. En segundo lugar, se estudiará al empleado actual en la empresa analizando mediante un cuestionario, el tiempo empleado en su puesto de trabajo realizando tareas monótonas y rutinarias. Para finalizar la investigación, se estudiará el caso de Job&Talent, una multinacional donde estas prácticas ya se ha implementado de forma exitosa

Palabras clave: Optimización de procesos, automatización de datos, tareas de back office, SQL, herramientas de automatización, Google Apps Scripts, tecnología

ABSTRACT

We find ourselves in an increasingly dynamic and competitive world, where technology has revolutionized how we work and how our time at work is spent. If before we used to dedicate a large part of the workday on routine tasks such as generating reports from updated data, today the emergence of automation tools has optimized the time we need to spend on these tasks, making them increasingly automated and less dependent on direct human intervention. In this research project, we will first study the change in employee's role as a result of these automated systems, focusing on tools such as SQL and Google Scripts for the collection of data, updating and report delivery processes. Furthermore, through a questionnaire, the time spent on these types of tasks by a typical employee will be studied, looking at the time spent in a workday performing monotonous and routine tasks. To conclude the research, what has been learned will be analyzed by looking at the case of Job&Talent, a multinational where these codes have successfully been implemented to streamline recurring tasks.

Key Words: Process optimization, data automation, back-office tasks, SQL, automation tools, Google Apps Scripts

ÍNDICE GENERAL

I.	JUSTIFICACIÓN DEL TEMA	7
II.	OBJETIVOS DEL TRABAJO	7
III.	METODOLOGÍA	8
IV.	CONTEXTO HISTÓRICO	8
V.	VOCABULARIO TÉCNICO EN LA AUTOMATIZACIÓN DE PROCESOS	9
	5.1 La Interfaz de Programación de Aplicaciones	9
	5.2 La Evolución de la Red	10
	5.3 El Cloud Computing	11
	5.4 PaaS: Amazon Redshift	12
VI.	LA AUTOMATIZACIÓN DENTRO DE LA EMPRESA	13
	6.1 La Automatización de Datos	13
	6.2 La Digitalización en las Empresas	13
	6.3 Robotic Process Automation	14
	6.4 La automatización dentro de las finanzas	15
VII.	ANÁLISIS	15
	7.1 Conocimiento general de la automatización de procesos de back office	18
	7.2 Mayores desafíos de las tareas de back office	20
	7.3 La Automatización de Procesos de back office como Herramienta de Trabajo ...	22
VIII.	IMPLEMENTACIÓN PRÁCTICA: JOB&TALENT	23
	8.1 El caso de Job&Talent	24
	8.2 Creando un código extrapolable a diferentes países	25
	8.3 La definición de Variables	26
	8.4 De código a tabla: Importando los datos	27
	8.4.1 Importando tablas de bases de datos mediante SQL	28
	8.4.2 Importando información de otras hojas de cálculo	31
	8.5 Generando Informes y Copias Automatizadas	36

8.6	Integrando las funciones como accesos directos a la hoja de cálculo	38
8.7	Incorporando mecanismos de control	39
8.8	Actualizar tablas de google sheets a bases de datos: las <i>Sheet to Tables</i>	40
8.8.1	Apache Airflow: Un intermediario para la automatización de datos.....	41
IX.	CONCLUSIONES	42
X.	FUTURAS LÍNEAS DE INVESTIGACIÓN.....	45
XI.	DECLARACIONES DE USO DE CHAT GPT	46
XII.	BIBLIOGRAFÍA	47
XIII.	ANEXO I	50
XIV.	ANEXO II.....	52

ÍNDICE DE FIGURAS

Figura 1:	Comparativa de los tres escenarios de computación y almacenamiento	11
Figura 2:	Características esenciales y modelos del cloud computing.....	11
Figura 3:	Panel de Servicios Google Scripts.....	24
Figura 4:	Captura del Archivo “Variables Definitions” en Google Apps Script	27
Figura 5:	Código de la función ‘df_actuals’ creado en Google App Scripts	30
Figura 6:	Ejemplo de filtración de datos mediante bucle anidado codificado en Pop SQL .	31
Figura 7:	Código de la función ‘df_import_merge’ escrito en Google App Scripts	33
Figura 8:	Creando una función para exportar versión a una carpeta con Google App Scripts	37
Figura 9:	Captura de pantalla del menú desplegable FinOps.....	38
Figura 10:	Funcionamiento del submenú “Import Unique Metrics”	39
Figura 11:	Función ‘print_last_update’ creada para Google App Scripts	40
Figura 12:	Flujo de automatización de datos entre herramientas	42

ÍNDICE DE TABLAS

Tabla 1: Ficha técnica de encuesta online	16
Tabla 2: Distribución de Años Trabajados entre los Encuestados	17
Tabla 3: Departamento al que pertenecen los encuestados	18
Tabla 4: Desglose de los resultados a la pregunta: “¿Has oído hablar de la automatización de procesos de back office?	19
Tabla 5: Relación entre automatización de tareas de back office y la realización de éstas... ..	19
Tabla 6: Tareas de back office más automatizables según los encuestados	20
Tabla 7: Tareas de back office donde más presentados en el trabajo del encuestado	21
Tabla 8: Percepción de la automatización de tareas de back office como una mejora en la agilidad laboral	22

I. JUSTIFICACIÓN DEL TEMA

Estudiar un doble grado en ADE y Business Analytics ya hace que te cuestiones el poder de las tecnologías de big data y herramientas de análisis de datos dentro de la empresa tradicional. Cuando empezamos a estudiar se nos dijo que nosotros no iríamos a ser los técnicos que creasen los códigos, es decir no íbamos a ser ingenieros de datos. Nuestra labor como profesionales iba a centrarse en el uso de los datos, para llevar a cabo análisis estadísticos, trabajar con grandes columnas de datos, y generar modelos que ayudarán a tomar decisiones justificadas y estratégicas. En la teoría sonaba algo innovador y diferente, pero como alumna nunca me imaginé como lo terminaría empleando dentro del trabajo, al fin y al cabo, mi verdadero interés giraba en torno al mundo de los negocios.

Cuando empecé este último año de carrera, había realizado varias prácticas en distintas compañías, y aunque la mención de business analytics resultaba novedosa, no había puesto mis conocimientos en práctica, por lo que deduje que mis pensamientos originales eran acertados: el análisis de negocio como complemento a un grado en ADE era interesante, pero su utilidad en mi campo de trabajo sería limitada. Así lo he creído hasta mis últimas prácticas en Job&Talent. En Enero de 2024 comencé una beca en el departamento financiero de esta empresa tecnológica, más específicamente dentro del grupo de *finance analytics*, y es donde realmente he descubierto el poder del análisis de negocios y de las herramientas que tenemos a nuestro alcance para recopilar datos, limpiarlos, crear modelos predictivos y tomar decisiones de manera informada gracias al análisis de datos. En mi día a día utilizo herramientas como Google App Scripts y SQL para crear modelos que ayuden en la gestión financiera de la empresa, optimicen las operaciones y reduzcan el tiempo que los equipos tienen que emplear en compartir datos, crear informes, tomar decisiones justificadas y cohesionar los datos de diferentes departamentos y sedes de forma rápida y sencilla.

La utilidad y practicidad de estas actividades motivo mi deseo de explorar esta nueva corriente e investigar como de generalizadas están el uso de estas herramientas en diferentes industrias.

II. OBJETIVOS DEL TRABAJO

Este trabajo tiene como objeto de estudio determinar cómo las herramientas de tecnología de la información se integran en un entorno empresarial para optimizar y facilitar los procesos de gestión, procesamiento y análisis de datos comunes en las medianas y grandes empresas.

La tecnología permite a las empresas tener acceso a más información, llegar a más clientes y ofrecer servicios más personalizados y rápidos. Así mismo, permite la coordinación entre diferentes sedes de forma instantánea, proporcionando métodos de comunicación y visualización de datos que facilitan la comprensión y la homogeneización de los procesos empresariales que se llevan a cabo en diferentes departamentos de una empresa. Con los resultados obtenidos de esta investigación se pretende entender el contexto actual de las empresas, como éstas son capaces de automatizar sus procesos de forma sencilla y económicamente factible, y cómo se incorporan estas herramientas en una empresa multinacional. Se plantean por tanto las siguientes hipótesis que se irán resolviendo durante la investigación:

- Existen herramientas de tecnología de la información que permiten automatizar procesos repetitivos e ineficientes dentro de la empresa.
- El empleado pierde tiempo en actualizar bases de datos y compartir información que podría hacerse de forma automática.
- Las empresas pueden incorporar modelos que ayuden con la predicción de datos, generación de reportes y mejorar la conectividad y comunicación.
- La automatización de datos reduce los errores y aumenta la eficiencia operativa dentro de la empresa.

III. METODOLOGÍA

En esta investigación, se ha combinado la revisión de la literatura, la recopilación de nuevos datos a través de una encuesta, así como el análisis de un caso real para proporcionar una base sólida que permitiese entender el impacto y las ventajas de incorporar estas herramientas en un entorno empresarial.

Para llevarse a cabo se comenzó realizando una revisión de la literatura existente, enfocándola en el desarrollo de las herramientas tecnológicas dentro del mundo empresarial, su funcionamiento, las ventajas que proporcionan y de qué manera se han ido incorporando para ofrecer una comprensión profunda de sus ventajas y utilidad en los diversos contextos empresariales. Esta revisión incluyó libros académicos, fuentes por Internet, bibliotecas digitales, artículos de revistas científicas y estudio de casos publicados.

Para complementar la revisión de la literatura, se lanzó una encuesta creada en Google Forms para recoger más datos sobre la percepción del empleado en cuanto al tiempo que emplean en

el tipo de tareas que estas herramientas pretenden efficientar y el uso y conocimiento que tienen sobre la automatización de procesos empresariales y su mejora en la eficiencia operativa. Esta encuesta se ha analizado utilizando técnicas de análisis estadísticos y herramientas de visualización de datos para mejorar la comprensión y facilitar la extracción de ideas clave. El análisis de los resultados se ha llevado a cabo utilizando Excel, haciendo uso de sus tablas dinámicas y otras funcionalidades.

Finalmente, se estudió el caso de Job&Talent exponiendo cómo esta empresa ha implementado de forma exitosa estas herramientas y cómo han logrado optimizar procesos, mejorar la eficiencia operativa y facilitar la toma de decisiones basada en datos. Esta investigación, que se vio motivada por mi rol dentro de esta empresa en implementar estos procesos, muestra los códigos, de forma teórica por motivos de confidencialidad, y el resultados de implementarlos, en las operaciones del día a día de esta compañía. Además, se expone cómo se integran diferentes lenguajes de programación y plataformas de servicios de computación para lograr estos resultados.

IV. CONTEXTO HISTÓRICO

Nos encontramos en la década de 1960, el término “automatización de tareas” dentro del contexto empresarial empieza a hacerse eco con la introducción de los primeros ordenadores grandes. Estos empezaron a usarse para realizar tareas automáticas y repetitivas en procesos con grandes volúmenes de datos. Hemos oído hablar de la automatización en todos los sectores. Desde la primera revolución industrial ya se escuchaba el temido “auge de las máquinas”. Cada desarrollo ha sido tanto una oportunidad como un reto y como ha ocurrido en las tres revoluciones industriales anteriores, y ocurre en la que estamos sumidos ahora mismo, el miedo a cómo afectará a los puestos de trabajo reside. No obstante, como ya nos ha demostrado la historia, se trata de una transición que genera nuevas oportunidades, mejora la eficiencia y, en esta revolución, libra al trabajador de actividades repetitivas, proporcionando con más tiempo para trabajar en tareas más valiosas para el empleado y la empresa (Terrar & Singh, 2020).

La automatización de procesos de tareas repetitivas en aplicaciones de oficina es cada vez más habitual y fácil de llevar a cabo gracias a los conocimientos que hay sobre cómo trabajar estas herramientas. Los diseñadores de software que antes realizaban códigos extensos y complejos, entendidos por unos pocos, han ido desarrollando lenguajes de programación más comprensibles y sencillos para el usuario común. Esto ha permitido que la integración de

soluciones automatizadas se realice de manera más eficiente y efectiva. Los cambios económicos y sociales que hemos experimentado en los últimos siglos han representado tres importantes revoluciones industriales: la primera a finales del siglo dieciocho, con la producción mecánica, la segunda en el siglo diecinueve con la producción industrial masiva, y la tercera a mitades del siglo veinte con los ordenadores y el Internet (Frey & Osborne, 2013). Ahora, estamos ante la cuarta revolución industrial que está integrando nuevas tecnologías que están transformando los sistemas de gestión empresarial.

V. VOCABULARIO TÉCNICO EN LA AUTOMATIZACIÓN DE PROCESOS

Según un informe de Harvard Business Review (Pérez, 2023), la automatización de tareas es una de las formas más rápidas de mejorar la eficiencia y productividad en todos los departamentos, y ofrece numerosas ventajas incluyendo la mejora de productividad, la reducción de errores humanos, y el ahorro significativo de tiempo que permite a los empleados centrarse en tareas de mayor valor añadido que reducen el estrés de los equipos y mejora los resultados. En este contexto, es crucial entender conceptos claves como las APIs, la computación de la nube o el concepto de la red. En este apartado se pretende resumir las definiciones con las que trabajaremos más adelante cuando estudiemos el caso de Job&Talent.

5.1 La Interfaz de Programación de Aplicaciones

Una API (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y protocolos que permite a diferentes aplicaciones o sistemas comunicarse entre sí. Según la definición proporcionada por Amazon Web Services (s.f), una API define las reglas que se deben seguir para comunicarse con otros sistemas de software. Es decir, los desarrolladores, crean API para que otras aplicaciones puedan comunicarse con sus aplicaciones mediante la programación.

“La API de Google Sheets es una interfaz RESTful, que te permite leer y modificar los datos de una hoja de cálculo” (Google Developers, 2023). Esta API permite realizar una variedad de tareas, entre las cuales se incluyen:

- Crear, actualizar y eliminar hojas de cálculo dentro de un archivo de Google Sheets
- Lectura y escritura de datos de celdas, rangos específicos o hojas completas. Esto es de utilidad para recuperar datos que se encuentran en una hoja de cálculo y usarlos en otra aplicación, actualizar informes o automatizar el agregar nuevos datos a los ya existentes.

- Formatear las celdas, tanto en color, tipo de letra, bordes, y todos los elementos que mejoran la lectura y presentación de los datos (Google Developers, 2023).

Esta interfaz de programación permite a los desarrolladores interactuar con Google Sheets de manera funcional. Debido a la variedad de operaciones que puede realizar sobre hojas de cálculo, resulta una herramienta especialmente útil para automatizar tareas repetitivas e integrar Google Sheets con otras aplicaciones. Como veremos más adelante, uno de sus grandes beneficios es su capacidad de que un script extraiga información de una base de datos externa o con una herramienta de CR;, y actualice una hoja de cálculo de Google Sheets, que nos permita generar informes diarios con información actualizada.

Hemos descrito la API de Google Sheets como una interfaz RESTful. Está, también conocida como una API RESTful, es una interfaz de programación que sigue los principios REST (Representational State Transfer), un estilo arquitectónico para construir servicios web que sean escalables y fáciles de mantener. Una API RESTful permite que las distintas aplicaciones se comuniquen entre sí a través de la web utilizando las convenciones estándar de HTTP (Fielding, 2000).

5.2 La Evolución de la Red

Aunque la tecnología es algo completamente integrado en todos los ámbitos de nuestra vida, las tecnologías de la información son relativamente recientes. Hasta mediados del siglo pasado, no aparecieron los primeros ordenadores. Cuando aparecieron los transistores, los ordenadores empezaron a permitir cierta programación, pero nada comparable con lo que tenemos hoy en día. Hasta los años 60, no se extendió la informática a los negocios, aunque está era mediante grandes ordenadores en centros de datos empresariales, gestionados por nuevos departamentos de informática. Más adelante, es cuando surgieron los terminales con pantalla y teclado, conectados a un *mainframe* central.

Posteriormente, en los años 80 los ordenadores personales (PC) hicieron que la informática fuese accesible en pequeñas y medianas empresas, llegando también hasta los hogares. En esta misma década, apareció el Internet, aunque no fue hasta los años 90 cuando este se popularizó y verdaderamente impulsó un gran crecimiento de usuarios y aplicaciones, surgiendo webs como Google y Amazon. En este entorno, los usuarios no solo consumen información, sino que también la producen, lo cual creó una nueva necesidad de potentes servidores que pudiesen soportar a numerosos usuarios conectados desde diversos dispositivos. A raíz de esto las

empresas externalizan su computación y almacenamiento a grandes centros de datos accesible por Internet, lo que se conoce como cloud computing (Torres i Viñals, 2012).

En la siguiente tabla se muestra un breve resumen de los tres escenarios informáticos que se han presentado.

Figura 1: Comparativa de los tres escenarios de computación y almacenamiento

	Modelo de computación y almacenamiento	Características	Costes
Mainframe	Centralizado	Sistemas que procuraban aprovechar al máximo los recursos a causa del alto coste de estos.	Inversión inicial tanto para el hardware como para el software.
PC	Distribuido	PC y servidores distribuidos conectados en red (primero local y después Internet).	Inversión inicial para la compra de hardware y costes de licencias para el SO y las aplicaciones software que se usaban.
Cloud	Centralizado	Inmensos CPD con recursos TIC de bajo coste (<i>commodity</i>) que se optimizan aprovechando la economía de escala.	Se paga solo por lo que se gasta a medida que se va usando a lo largo del tiempo.

Fuente: (Torres i Viñals, Del Cloud Computing al Big Data, 2012)

5.3 El cloud computing

“El *cloud computing* es un modelo que permite el acceso ubicuo y bajo demanda a una red compartida de recursos informáticos configurables (como por ejemplo red, servidores, almacenamiento, aplicaciones y servicios) que pueden ser provisionados y liberados con el mínimo esfuerzo de gestión o interacción con el proveedor del servicio”. (National Institute of Standards and Technology, 2011, traducción propia, p.6). En este artículo del NIST, se establece un claro marco sobre los aspectos clave de la computación en la nube. Describe cinco características esenciales, 3 modelos de servicio y 4 modelos de implantación. En la Figura 2, se resumen estos aspectos.

Figura 2: Características esenciales y modelos del cloud computing

COMPUTACIÓN EN LA NUBE		
Características Esenciales	Modelos de Servicio	Modelos de Implementación
Autoservicio a demanda	Software como Servicio (SaaS)	Nube Privada
Acceso amplio a la red	Plataforma como Servicio (PaaS)	Nube Pública
Agrupación de recursos	Infraestructura como Servicio (IaaS)	Nube Comunitaria
Elasticidad rápida		Nube Híbrida
Servicio medido		

Fuente: (Elaboración propia. Basado en el NIST, 2011)

Los modelos de servicio dentro del ecosistema de cloud computing describen las diferentes maneras en las que los servicios de computación son ofrecidos a los usuarios. Estos modelos permiten a las organizaciones elegir el nivel de control que quieren tener sobre sus aplicaciones en la nube, lo que optimiza el servicio para sus necesidades específicas.

En 2022, el artículo “Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies” describen los tres modelos de servicio principales presentados en la Figura 2 como:

1. Software como Servicio: el proveedor de la nube presenta a los consumidores con aplicaciones accesibles a través de la interfaz del programa o el buscador web, y el consumidor tiene un control limitado sobre las aplicaciones específicas del usuario.
2. Plataforma como Servicio (PaaS): En la infraestructura de la nube, los consumidores pueden construir y distribuir aplicaciones. Además, pueden ejercer control sobre las aplicaciones que distribuyen pero no sobre la infraestructura de la nube.
3. Infraestructura como Servicio (IaaS): el consumidor está provisto de recursos computacionales esenciales, vitales para el procesamiento, almacenamiento y networking. El usuario controla el almacenamiento, las aplicaciones y los sistemas operativos, pero no la infraestructura de la nube. (Angel, N.A., Ravindran, D., Vincent, P.M.D.R., Srinivasan, K., Hu, Y.-C.)

Como narran Armbrust, Fox, Griffith, Joseph, Katz, Konwinski, Lee, Patterson, Rabkin, Stoica y Zaharia (2010), es habitual que todo lo que tenga que ver con el *computing* se le atribuya hoy en día al cloud computing, debido a la falta de claridad entre las diferencias principales entre *cloud computing* y el *computing* convencional. Por tanto, identificamos como cloud computing a todas aquellas aplicaciones que sean entregadas como servicio a través del Internet y el *hardware* y sistemas de *software* en los centros de datos que proporcionan esos servicios.

5.4 PaaS: Amazon Redshift

Como veremos más adelante con el caso de Job&Talent, una importante herramienta en su proceso de automatización de datos es el servicio “Amazon Web Services”, y en particular, su servicio “Amazon Redshift”. Amazon Redshift se lanzó en 2013 por Amazon Web Services, con el objetivo de satisfacer cuatro demandas específicas del consumidor. En primer lugar proporcionan un gran rendimiento en la ejecución de consultas analíticas

complejas. En segundo lugar, escala sus capas de almacenamiento en respuesta a las diferentes cargas de trabajo y poder así manejar picos de trabajo. Es decir, escala hacia arriba cambiando el tamaño de cada clúster de forma elástica y hacia afuera para aumentar el rendimiento a través de la autoescalación de múltiples clústers. En tercer lugar, necesitaban que fuese fácil de usar para el usuario, para lo cual incorporaron autonomía de aprendizaje automático que ajusta cada cluster según la carga de trabajo que haya. Finalmente, para aumentar su utilidad, era importante que se integrase bien con el resto de los servicios que ofrece AWS.

VI. LA AUTOMATIZACIÓN DENTRO DE LA EMPRESA

6.1 La Automatización de Datos

“La automatización de datos es el proceso de utilizar tecnología para realizar de forma automática tareas que anteriormente se realizaban de forma manual. Estas tareas pueden incluir desde la entrada y procesamiento de datos, hasta el análisis y generación de datos.” (Bibook, 2023). Al eliminar, o reducir, la intervención humana en tareas repetitivas se consigue una mejora en la precisión y consistencia de los datos, eliminando el error humano y acelerando los tiempos de ejecución. La automatización de datos es fruto de la digitalización. Hablamos de la automatización de datos como el proceso de mejorar la eficiencia de procesos empresariales, reducir el error humano, integrar múltiples fuentes de datos y otorgar una mejor visión de la organización en su conjunto. El desarrollo e implementación de la digitalización es el propulsor de que esto sea posible. Kääriäinen, Parviainen, Teppola, y Tihinen, (2017), escribían como los beneficios potenciales de la digitalización para la eficiencia interna podía llevar a una mejora en la satisfacción del empleado mediante la automatización de tareas rutinarias.

6.2 La digitalización en las Empresas

“La transformación digital es la adopción estratégica de tecnologías digitales. [...] La transformación digital juega un papel importante en la experiencia del consumidor, la experiencia del empleado, la **optimización de procesos** y la digitalización de productos” (Citrix, s.f.). A día de hoy la digitalización en las empresas es un elemento fundamental para si uno quiere tener las herramientas para poder competir en su entorno. No obstante, por muy necesario que sea, y las muchas ventajas que proporcione como la optimización de procesos, la eficiencia en el trabajo y la aceleración de tiempos que hemos venido mencionando, implementarla adecuadamente y explotar todos sus beneficios puede ser complicado.

En un artículo de la Universidad de Navarra (2019), Vilaplana y Stein resumen los resultados de un estudio que realizó la consultora McKinsey (2018), donde “el 80% de las organizaciones encuestadas habían comenzado la transformación digital de sus empresas pero, sorprendentemente, sólo el 14% de ellas reportaron haber mejorado su desempeño, y más asombroso aún, sólo el 3% reportaron que el cambio había sido un éxito, lo que confirma el reto que supone la transformación digital en las organizaciones”.

Sorprendente es la palabra adecuada cuando intentamos entender como un cambio que abre una puerta tan amplia de posibilidades, no es explotado de forma que se perciba una mejora en los procesos y operaciones internas de la empresa. La importancia de la implementación de la digitalización es reconocida, pero las compañías encuentran un desafío en entender el potencial impacto y los beneficios de la digitalización (Kääriäinen et al., 2017). Hay que saber diferenciar entre la digitalización de una empresa y sus capacidades dentro de la digitalización. La digitalización como ya hemos dicho es la aplicación de tecnologías digitales mientras que su capacidad es definida por Grant (1996) como “la capacidad de una empresa para desarrollar repetidamente una tarea productiva que se relaciona, directa o indirectamente, con su capacidad de crear valor al afectar la transformación de insumos.

6.3 Robotic Process Automation

La Automatización de Procesos Robóticos (RPA por sus siglas en inglés), es descrito por el Instituto para la Automatización de Procesos Robóticos y la Inteligencia Artificial como “la aplicación de tecnología que permite a los empleados en una empresa configurar un software o a un ‘robot’ para capturar e interpretar aplicaciones existentes para procesar una transacción, manipular datos, desencadenar respuestas y comunicarse con otros sistemas digitales” (2024). La RPA se aprovecha de la digitalización de la información para promover una ventaja competitiva al realizar tareas de forma automática siguiendo criterios y directrices estrictamente definidos (Moreira, Mamede & Santos, 2023).

La RPA y la automatización de datos tienden a ir de la mano en lo que se refiere a la incorporación de estas tecnologías dentro del entorno empresarial. Se trata de herramientas complementarias que permiten mejorar la eficiencia operativa y la gestión de la información lo facilita la transformación digital. La automatización de datos usa la tecnología para recoger, procesar y analizar datos de forma sistemática y eficiente mientras que la RPA utiliza robots de software que automatizan tareas repetitivas basadas en reglas que comúnmente realizan los empleados, como es la recogida y análisis de datos. Es decir, es común integrar estas

herramientas para automatizar y mejorar la manipulación y gestión de grandes volúmenes de datos.

6.4 La automatización dentro de las finanzas

El sector financiero también se ha visto fuertemente influenciado por los desarrollos de la automatización, tanto a nivel sectorial como a nivel departamental. En la industria financiera hemos visto la automatización de procesos con la digitalización de pagos, por ejemplo. A nivel interno en una empresa, el departamento financiero también ha sido capaz de sacar provecho a las muchas herramientas que trae consigo la digitalización y automatización de estos datos. La automatización financiera se refiere al uso de tecnología para completar procesos empresariales, de tal manera que las tareas se vuelven menos repetitivas y consumen menos tiempo para aquellas personas que trabajan en ello. Además, permite aumentar la eficiencia, productividad y resultado (Trustpair, 2023). Con la automatización de estos procesos no hablamos de que la tecnología reemplace al trabajador de este departamento, sino que colaboren en obtener los mejores resultados reduciendo el margen de error, los tiempos de operación, aumentando la eficiencia operativa y la capacidad de análisis. Como describen Davenport y Kirby (2016), el futuro de la mejora en la productividad y éxito empresarial no es ni humano ni tecnológico, sino una combinación de ambos. Él lo denomina como la *aumentación*, dentro de un contexto tecnológico.

VII. ANÁLISIS

La revisión bibliográfica nos ha explicado la utilidad y la importancia de la automatización de datos en el entorno empresarial, haciendo hincapié en que la transformación digital del último siglo ha llevado a que más que una opción sea una necesidad dentro de las empresas. Sin embargo, el estudio de McKinsey que exploramos anteriormente a través de Vilaplana y Stein (2019), nos daba a conocer que por muy necesario que sea que las empresas incorporen estas tecnologías dentro de sus operaciones diarias, y por muy común que sea que se estén implementando, no por ello quiere decir que las empresas o su personal estén preparadas para hacer uso de sus beneficios y ventajas operativas.

Con el fin de entender esto desde el punto de vista del trabajador, se ha lanzado un formulario donde se ha buscado entender qué sentimiento hay frente a estas tecnologías y cuánto tiempo dedica un trabajador a realizar tareas que podrían ser automatizadas por estas herramientas. La encuesta fue difundida por varios canales intentando sobre todo llegar a perfiles de personas

que realicen tareas de forma automática y cuya edad sea inferior a los 30 años. Esto tenía un doble propósito. Por un lado se quería recoger los datos de aquellas personas que son consideradas nativos digitales¹, y que por tanto la posibilidad de haber estudiado y aprendido sobre estas tecnologías sea más alta, y por otro lado se buscaba que fueran individuos cuya experiencia laboral fuese lo suficientemente reducida que fuese habitual que en su jornada laboral tuviesen que realizar tareas repetitivas y automatizables. En total se obtuvieron un total de 32 respuestas. La Tabla 1 muestra la distribución de estas respuestas.

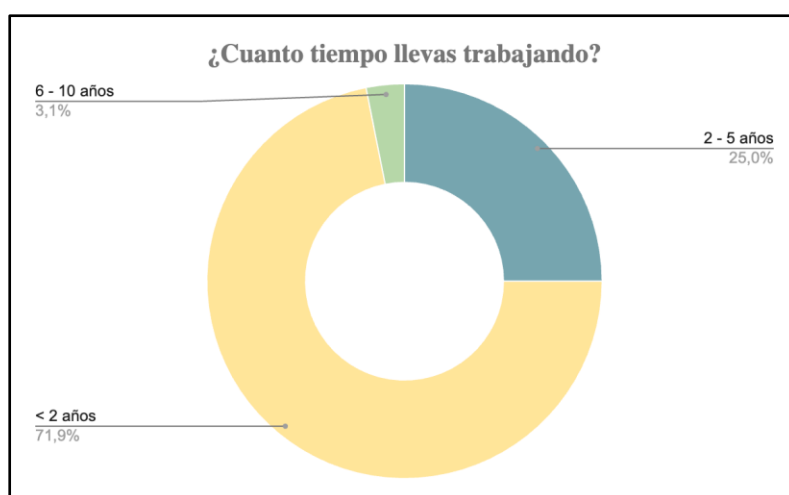
Tabla 1: Ficha técnica de encuesta online

Ficha Técnica	Encuesta <i>on-line</i>
Universo	Población mayor de 18 años
Área geográfica	España (Madrid)
Tamaño de la muestra	163 respuestas
Tipo de muestra	Aleatoria simple
Tipo de encuesta	Personal, distribuida por Internet
Intervalo de confianza	95%, para $p=q=0,5$
Error muestral	$\pm 0,173$
Instrumento de medición	Encuesta digital elaborado con Google Forms

Esta encuesta tiene como objetivo complementar la investigación bibliográfica y la puesta en práctica de estas herramientas en un proyecto en el departamento de finanzas de una empresa, por lo que la encuesta se ha empleado como complemento para respaldar algunas de las conclusiones y datos que hemos sacado. Como se ha mencionado previamente, el público objetivo de esta muestra eran aquellas personas que llevaban poco tiempo en el mercado laboral, debido al tipo de tareas que se suele llevar a cabo a este nivel. En la Tabla 2 se muestra como están distribuidos los años de experiencia entre los 32 encuestados.

¹ *Nativo Digital* - aquellos individuos que han crecido en un entorno donde las tecnologías digitales están omnipresentes desde una edad temprana.

Tabla 2: Distribución de Años Trabajados entre los Encuestados

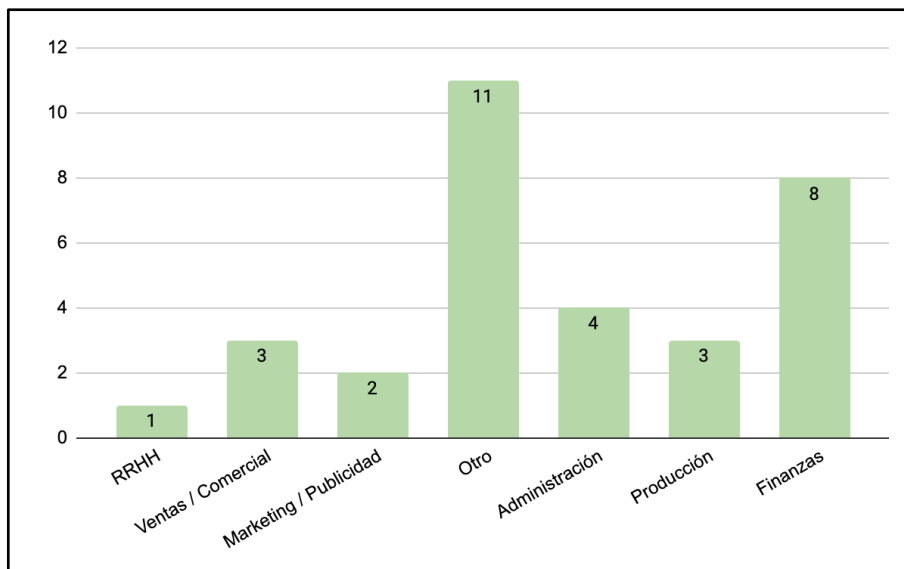


Base: 32 encuestados

Como podemos ver, todos los encuestados cumplen con los años de experiencia que buscábamos. Aunque no es generalizable a todos los ámbitos de trabajo, en su mayoría, en especial en empresas como Job&Talent y otras grandes multinacionales, los empleados con menos años de experiencia suelen realizar más tareas respectivas por su falta de experiencia, donde en roles más básicos se les encomiendan tareas rutinarias y repetitivas.

De nuestro interés en particular son personas que trabajan en un entorno empresarial en departamentos donde se mueven grandes volúmenes de datos. Para ello a los encuestados les pedimos que identificaran en qué departamento trabajaban. Se dieron como opción: Producción, Ventas / Comercial, Finanzas, Recursos Humanos, Marketing / Publicidad, Administración, Compras u Otros. Estos departamentos son aquellos que por lo general, sin contar con el departamento de informática, trabajan de forma indirecta con una gran cantidad de datos que tienen que emplear continuamente, copiando y pegando entre documentos, realizando informes, llevando a cabo análisis, etc. Aquellas personas que trabajan como informáticos, no son personas que para nuestro estudio resultan de gran valor, dado que el fin es investigar cómo se incorporan tecnologías y herramientas de automatización de datos en tareas repetitivas en campos como las finanzas. Es decir, donde la naturaleza del empleado es trabajar con datos con el fin de sacar conclusiones y realizar otras actividades. En la tabla 3, se muestra a qué departamento seleccionaron los encuestados que pertenecen.

Tabla 3: Departamento al que pertenecen los encuestados



Base: 32 personas

Una vez determinada el área de especialización de los encuestados, podemos realizar preguntas más específicas al tipo de tareas que tienen que realizar, cuantas de ellas podrían ser automatizadas, y como de frecuentemente se ven haciendo estas tareas.

7.1 Conocimiento general de la automatización de procesos de back office

En primer lugar, se preguntó si alguna vez habían oído hablar de la automatización de *procesos de back office*². Con este término se pretendía relacionar la automatización con los procesos internos de la empresa, para así diferenciar de la automatización de otros campos como puede ser la automatización de máquinas. Por lo general, en todos los departamentos las respuestas se dividieron equitativamente entre quienes conocían y quienes no conocían el término, o hubo una mayoría que indicó no haber oído hablar de él.

En la tabla 4, se muestra el desglose de resultados dividido por departamento.

² *Procesos de back office - las operaciones internas y administrativas de una empresa que no están relacionadas directamente con el cliente pero son esenciales para el funcionamiento eficiente de la organización*

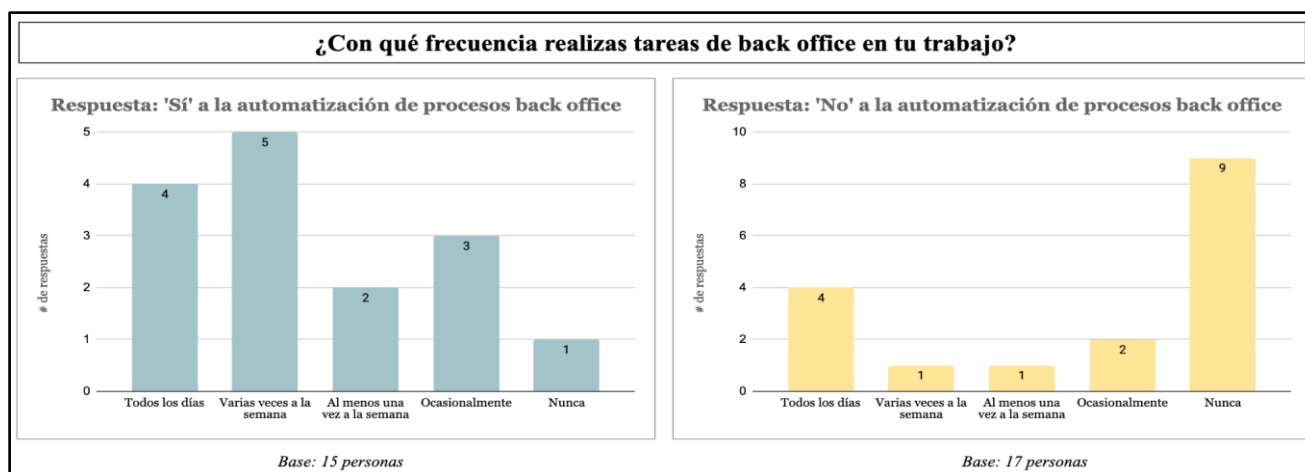
Tabla 4: Desglose de los resultados a la pregunta: “¿Has oído hablar de la automatización de procesos de back office?”

¿Has oído hablar de la automatización de procesos de back office?			
	No	Si	Total Rango
Administración	2	2	4
% del rango	50,00%	50,00%	100,00%
Finanzas	5	3	8
% del rango	62,50%	37,50%	100,00%
Marketing / Publicidad	1	1	2
% del rango	50,00%	50,00%	100,00%
Otro	6	5	11
% del rango	54,55%	45,45%	100,00%
Producción	0	3	3
% del rango	0,00%	100,00%	100,00%
RRHH	1	0	1
% del rango	100,00%	0,00%	100,00%
Ventas / Comercial	2	1	3
% del rango	66,67%	33,33%	100,00%
Suma total	17	15	32

Base: 32 personas

No obstante, aunque es útil conocer el nivel de familiaridad que tienen los encuestados con este concepto, es aún más interesante identificar a aquellos que no realizan tareas repetitivas susceptibles de ser automatizadas. Este grupo probablemente desconoce las herramientas disponibles para reducir el tiempo invertido en tareas repetitivas. Además, un número significativo de encuestados que no han oído hablar de la automatización de procesos de back office, pero que realizan muchas de estas operaciones, reforzaría los hallazgos del estudio de McKinsey explorados por Vilaplana y Stein (2019). Para visualizar estos resultados, se han creado dos gráficos: uno muestra los resultados de los que respondieron 'Sí' y otro los de los que respondieron 'No' a la pregunta anterior. Estos se presentan en la Tabla 5.

Tabla 5: Relación entre automatización de tareas de back office y la realización de éstas



Base: 32 personas

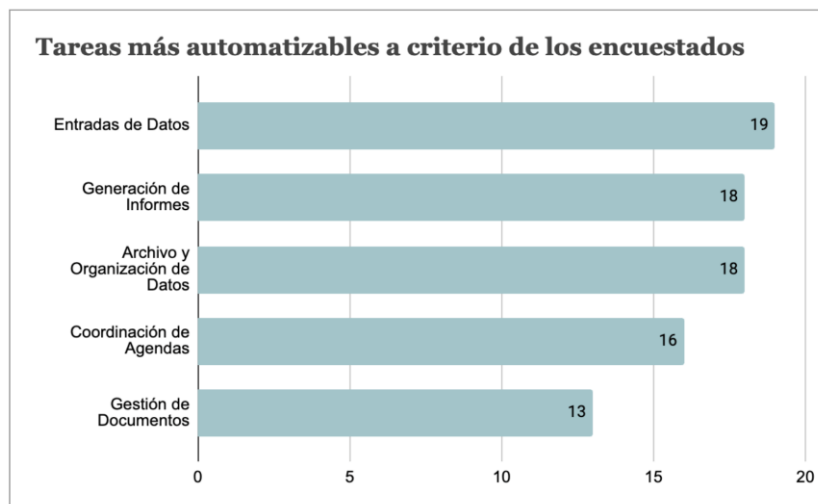
Es importante destacar, en primer lugar, que en la mayoría de los casos existe una relación positiva entre conocer la automatización de tareas de back office y realizar estas tareas de manera habitual en el trabajo. No obstante, también es interesante observar que 4 de las 17 personas que no habían oído hablar de la automatización de procesos de back office realizan este tipo de tareas todos los días. Esto sugiere que la eficiencia y la gestión del tiempo de este grupo de encuestados podrían mejorar significativamente si se aplican algunos de los conceptos revisados anteriormente.

7.2 Mayores desafíos de las tareas de back office

Con la primera parte de la encuesta, se ha buscado entender la utilización de herramientas y tecnologías que facilitan la automatización de los procesos descritos, desde la perspectiva de los encuestados. Como veremos más adelante con un ejemplo práctico, resulta relativamente sencillo aplicar estos conocimientos al entorno de trabajo y los beneficios son amplios. Sin embargo, hemos comprobado que estas herramientas no se emplean ni se explotan en su máximo potencial.

Para la segunda parte de la encuesta, el foco se centró en identificar el tipo de tareas que los encuestados consideran útiles de automatizar. También se exploró en qué áreas se puede aplicar esta tecnología de forma más sencilla y dónde sería más beneficioso hacerlo. En primer lugar se presentaron 5 de las tareas de back office más comunes, para estudiar cuales se ven automatizables de forma más sencilla. En este caso, no hubo límite de selección, pudiendo elegir una opción, todas o ninguna.

Tabla 6: Tareas de back office más automatizables según los encuestados



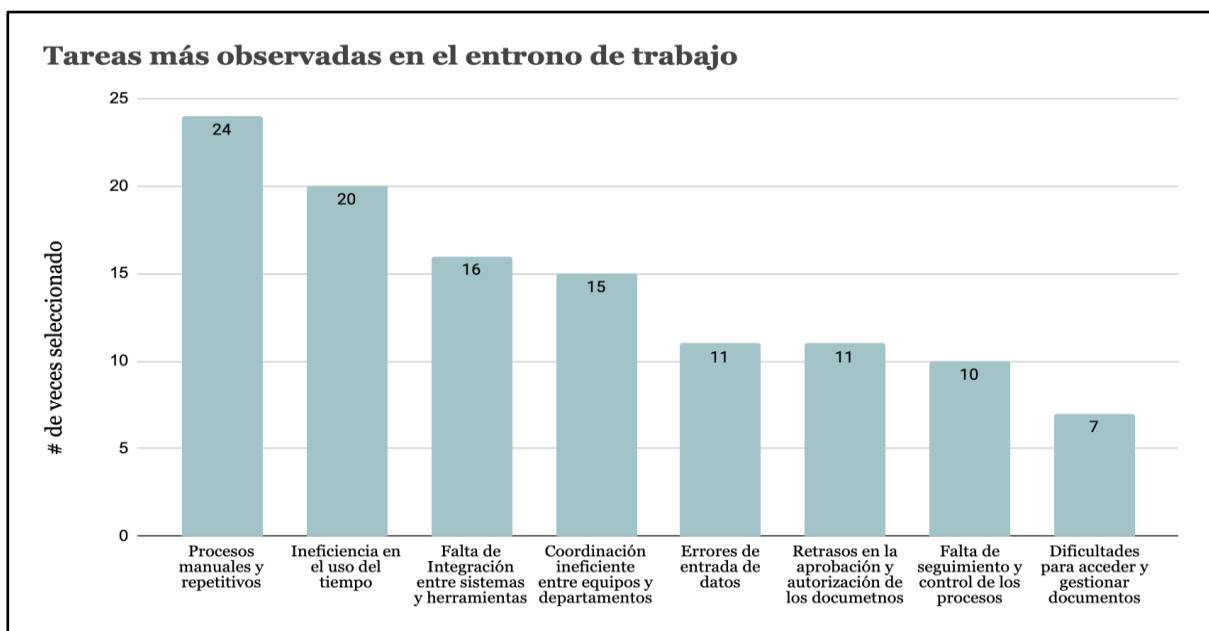
Base: 32 personas

Como podemos observar en la tabla 6, casi a la par tenemos el número de veces que la entrada de datos, la generación de informes y el archivo y organización de datos han sido seleccionadas como las tareas más fácilmente automatizables. Destacamos aquí la relación que existe entre estas tareas y los datos, habiendo un claro indicio de que todo lo relacionado con los datos se visualiza dentro del entorno de la automatización.

De forma similar, comprobamos que las tareas que más a menudo desarrollan los encuestados y que podrían ser automatizadas con las tecnologías investigadas son aquellas que implican el tratamiento y manipulación de datos. Los datos se han convertido en el nuevo compañero de trabajo para prácticamente cualquier departamento, debido principalmente a la utilidad de las conclusiones y análisis que se pueden extraer gracias a las herramientas disponibles para su análisis. No obstante, al igual que existen herramientas para analizar y visualizar datos, también hay herramientas que facilitan el mantenimiento, actualización y organización de estos datos.

En la tabla 7 se muestran estas tareas. Al igual que con la pregunta anterior, se les planteó a los encuestados una lista de tareas de la cual podían elegir tantas como quisieran. El objetivo de la pregunta era identificar a qué tareas dedican más tiempo los encuestados, a las cuales se presentará un ejemplo de automatización más adelante.

Tabla 7: Tareas de back office donde más presentados en el trabajo del encuestado



Base: 32 personas

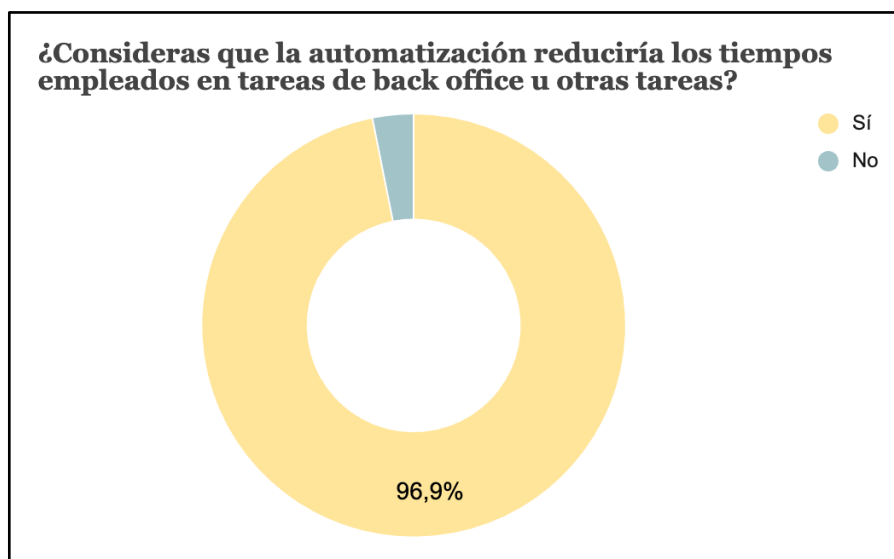
Con la tabla 7 también descubrimos que el encuestado identifica que estas tareas son ineficientes en gestionar el tiempo y que este se ve ocupado por procesos manuales y repetitivos, precisamente los que este estudio pretende reducir exponiendo las herramientas que pueden emplearse.

7.3 La Automatización de Procesos de back office como Herramienta de Trabajo

Tras estudiar el conocimiento de las tareas de back office, el tiempo empleado en ellas, así como el uso o conocimiento de la automatización de estas tareas, se les hizo una última pregunta a los encuestados. Como se ha venido presentando, la automatización es útil y no resulta muy compleja de integrar. Sin embargo, en el proceso de digitalización de las empresas, muchas no aprovechan adecuadamente las herramientas a su disposición. Al no explotar estas herramientas, la mejora en rendimiento y eficiencia organizativa no se logra como se esperaría, pues los procesos que más tiempo consumen, devolviendo proporcionalmente el menor valor añadido, siguen haciéndose de forma manual.

Al preguntarles a los encuestados si creen que se beneficiarían de la automatización de tareas de back office y si esto resultaría en una agilidad aumentada, casi el 100% de ellos estuvieron de acuerdo en que sí les sería beneficioso. Estos resultados se resumen en la tabla 8.

Tabla 8: Percepción de la automatización de tareas de back office como una mejora en la agilidad laboral



Base: 32 respuestas

VIII. IMPLEMENTACIÓN PRÁCTICA: JOB&TALENT

Job&Talent es una empresa fundada en 2009, que se define a sí misma como una plataforma a nivel mundial que conecta a trabajadores con empresas que requieren personal de forma directa y eficiente. Su modelo de negocio se centra alrededor de la tecnología, que usa para que los trabajadores puedan encontrar ofertas de empleo y completar el proceso en minutos. Se trata de una ETT (Empresa de Trabajo Temporal)³. Es decir, cuenta con dos tipos de trabajadores. Los que contrata para después ceder a las empresas usuarias, y los que trabajan para contratar a esos trabajadores, crecer la empresa, llevar las cuentas y establecer las estrategias a seguir. (Job&Talent, 2024)

Es una empresa que opera en 10 países de Europa, Estados Unidos y América Latina. En 2022 colocó a más de 340,000 trabajadores en más de 2,500 empresas. A través de una plataforma digital y una aplicación móvil, conecta con los trabajadores interesados. Es decir, se trata de una empresa que trabaja con un gran volumen de datos, en varios países y con un importante enfoque analítico, siendo este uno de sus siete principios operativos. Con todos los datos con los que operan, buscan optimizar sus decisiones a través del análisis.

Cuando comencé a trabajar en Job&Talent en el equipo de *Financial Analytics*, inmediatamente fui expuesta a la realidad de lo que es una empresa multinacional que trabaja con una increíble cantidad de datos. Estos datos están en continuo movimiento, traspasándose de departamento en departamento, usándose para generar análisis futuros, compartiendo entre países, etc. Al mismo tiempo, se trata de una empresa con una mentalidad de *start-up*⁴, es decir, aunque opere en muchos países, están en continuo contacto entre cada sede, realizando tareas y proyectos en conjunto.

Para poder ser eficientes en un entorno tan dinámico, es importante reducir el tiempo que se emplea para compartir información, actualizar y homogeneizar informes y datos entre países, y trasladar datos entre documentos y departamentos. A continuación, se presenta como esta empresa ha integrado procesos de automatización de tareas para eficientar el mantenimiento de una hoja de datos con la información financiera de la empresa con todas sus sedes, y el informe

³ ETT - una entidad cuya actividad principal es la contratación de trabajadores para cederlos temporalmente a otras empresas que necesitan cubrir puestos de trabajo por un período limitado.

⁴ Start-up - empresa de nueva creación que se caracteriza por su búsqueda de un modelo de negocio innovador, escalable y de rápido crecimiento

semanal correspondiente a dichos datos. Por razones de confidencialidad, solo se muestra el planteamiento teórico de los códigos creados.

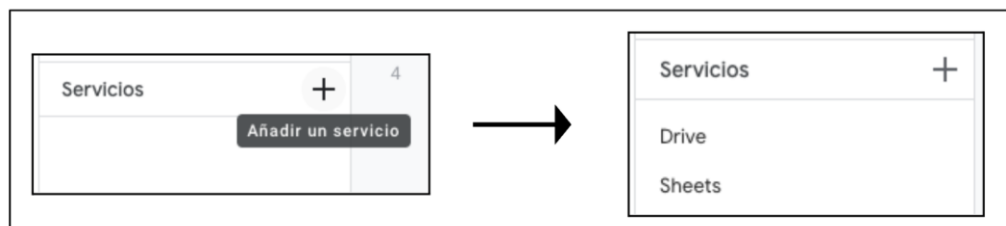
8.1 El caso de Job&Talent

El primer paso para poder escribir un código que facilite el trabajo de un empleado es dar acceso a los servicios de Drive y Sheets, para así poder acceder a los documentos que se actualizan de forma regular, y de los cuales queremos recoger la información.

- El servicio de Drive nos permite crear, encontrar y modificar archivos y carpetas dentro de Google Scripts.
- El servicio de Sheets nos permite crear, acceder y modificar archivos de Google Sheets.

La integración de sistemas de información supone una disminución significativa de costes y esfuerzos de comunicación entre trabajadores, departamentos, sucursales y en función a las propias formas de procesar la información (Proaño Castro, Orellana Contreras y Martillo Pazmiño, 2018). Es por eso que el poder conectar los documentos y archivos que se generan en un departamento con otros de forma automática y sencilla es tan útil. Añadiendo el servicio de Drive y de Sheets al Google Workspace nos permite esto mismo.

Figura 3: Panel de Servicios Google Scripts



Fuente: Elaboración propia.

Una vez tenemos nuestro cuadro de control, tenemos que pensar en que queremos hacer. En este caso, estamos hablando de una empresa que se caracteriza por tener una estructura muy compleja. Cuenta con operaciones en varios países así como con varias subsidiarias. Además, se trata de una empresa con muchos ingresos y gastos variados todos los días, que requiere llevar un reporte adecuado de los estados financieros, la situación actual de la empresa y los *forecasts* a futuro de cada país. Debido a la versatilidad de los datos así como la variedad de fuentes, el objetivo de esta automatización es generar un documento homogéneo y rápido que se puede compartir y emplear en otros análisis.

El objetivo por tanto es automatizar las tablas que muestran el *cash flow* diario, semanal y mensual de cada sede, así como los *forecasts* que se generan. De esta manera, se actualizará la información de forma automática, alimentándose de las tablas de bases de datos de la empresa, así como de otras hojas de cálculo, sin necesidad de tener que mantener actualizado este documento de forma manual. Del mismo modo, se podrán generar *snapshots* para enviar por correo, y así cumplir con las obligaciones de *reporting* que tiene cada departamento dentro de la empresa.

De aquí en adelante, el código empleado se ha codificado para proteger información sensible de la empresa.

8.2 Creando un código extrapolable a diferentes países

Como se ha venido diciendo, una de las claves de este proyecto es que una vez creado un código, el objetivo es poder extrapolarlo para que se pueda utilizar para sacar información de cualquier país/sede. La utilidad de estas herramientas no es solo la agilidad con la que se obtiene información actualizada, sino también la facilidad con la que se puede utilizar para generar informes homogéneos para diferentes países, con tan solo cambiar el valor de unas variables.

En este proyecto, se crearon varios archivos, donde en cada uno se definirían diferentes partes del código. En total creamos cinco archivos diferentes:

1. Variable Definitions: En este campo introduciremos todas aquellas variables que tengamos que cambiar para obtener diferentes datos con el mismo código. Por ejemplo, la variable “country”. Según el país para el cual queramos generar el informe, tendremos que cambiar las siglas que aquí aparecen definidas. De esta manera, podremos identificar fácilmente que variables manipular según que busquemos.
2. Menú: Estamos buscando facilitar el trabajo de empleados que no tienen conocimientos de programar o los cuales queremos reducir al máximo el tiempo empleado para actualizar datos, generar informes, etc. Es por ello que con este apartado, hemos creado un Menú accesible desde la barra de datos del spreadsheet, con el cual al dar a un botón podremos actualizar las diferentes pestañas de datos y generar informes.

Queremos distinguir entre las dos funciones básicas de la automatización de este reporting financiero. Por un lado tendremos el código interno, aquello que servirá para actualizar los datos que aparecen en las pestañas con los cuales podrán llevar a cabo los análisis, generar

informes y tomar decisiones. Por otro lado, tenemos el objetivo de generar informes semanales que muestren la situación actual de la empresa, y la cual se presente en un formato determinado para compartir con las personas interesadas dentro de la empresa. Para ello, hemos dividido en dos archivos diferentes los códigos que realizan cada función.

3. df: En este campo creamos todas las funciones que tengan que ver con importar datos a las diferentes pestañas. Se trata de una lista de funciones donde cada una ejecuta a una pestaña diferente los datos finales. Ordenarlo así nos permite ver con mayor facilidad que información necesita cada departamento que se importe en sus archivos.
4. Daily: En este archivo introducimos todas aquellas funciones que nos permitirán copiar los datos para generar un documento con la situación actual de la empresa. Además, podremos automatizar que se genere de forma automática una captura del estado actual de la empresa cada tiempo determinado. A esto se le conoce como triggers (o desencadenantes en español). Se trata de acciones específicas en SQL que quieres que se ejecuten de forma automática cuando ocurre un evento determinado en la base de datos.
5. Functions: En este último campo, escribimos todas las funciones secundarias que necesitaremos incluir en nuestras funciones principales (Daily y df). Son aquellas funciones como leer un archivo o copiar datos a una pestaña, que son fundamentales para el código, pero que no van a utilizar los empleados de forma directa.

8.3 La definición de Variables

En el primer archivo, hemos hablado sobre la definición de variables. Así mismo, también definimos aquí constantes. Podemos utilizar cualquier nombre para referirnos tanto a variables como a las constantes, siempre y cuando no entren en conflicto con las palabras reservadas en PASCAL para especificar datos o acciones (López, s.f., p.40).

Una variable es un objeto que se utiliza para guardar datos temporales durante la ejecución de un bloque de código. Pueden utilizarse para almacenar valores de datos que pueden cambiarse a lo largo del tiempo (Ben-Gan, I. (2017). T-SQL Fundamentals. Microsoft Press). Por otro lado tenemos las constantes, que es un valor inmutable, es decir su valor es fijado y no puede ser modificado durante la ejecución del programa. Se utiliza para definir valores que se sabe de antemano que no cambiarán durante la ejecución del código (Dalton, 2020, p. 122), evitando así la reintroducción repetida de valores invariables.

En el caso de este código, definimos dos constantes y una variable. Como constantes definimos la zona horaria así como el identificador de la hoja donde queremos leer la información. A pesar de tratarse de un código que queremos emplear en distintos países, recordamos que también buscamos la homogeneidad de los datos, de ahí que la zona horaria sea invariable, independientemente del país con el que queramos trabajar. Por otro lado, definimos la variable `países`. Se trata de un vector donde introduciremos las siglas de los países que queramos incluir en nuestro análisis. El objetivo de este código es que solo modificando ese vector, podamos generar diferentes informes y análisis a partir de las distintas funciones.

Figura 4: Captura del Archivo “Variables Definitions” en Google Apps Script

```
1  const time_zone = 'Europe/Madrid';
2  const sheet_id = SpreadsheetApp.getActive().getId();
3  var all_countries = ['ES'];
```

Fuente: Elaboración Propia

8.4 De código a tabla: Importando los datos

En estos procesos de automatización, el elemento fundamental son los datos. El poder acceder a ellos y a sus tablas es lo que nos permite elaborar unos modelos que funcionen de forma automática y eficiente. Tratamos con diferentes tipos de datos para esto. Al tratarse de reportes financieros, tenemos que tener en cuenta tanto los datos reales como las predicciones que podremos generar a partir de estos datos para generar modelos.

Por tanto, estamos hablando de la necesidad de integrar dos tipos de datos. Por un lado queremos trabajar con los datos reales y conocidos, datos que son subidos por los empleados y que no requieren ningún tipo de modelo. Por otro lado queremos visualizar las predicciones y *forecasts* que se generan para analizar en qué dirección va la empresa y poder tomar decisiones de forma eficiente y acertada. Asimismo, volvemos a hacer hincapié en la importancia de poder acceder a los datos de cualquier país así como el poder generar tablas que contengan información de las diferentes sedes que se integren en una sola para generar una visión global de los estados financieros de la empresa. Es por todo esto, que para unificar de forma adecuada esta información, este código recoge sus datos de dos fuentes diferentes.

En primer lugar, tenemos los datos que vienen de tablas de SQL, generados a partir de tablas de datos que almacenan información de los clientes de la empresa. Estas tablas alimentan

múltiples hojas de datos en spreadsheets y otros programas y almacenan información de plataformas digitales cuyos datos no son fácilmente analizables hasta que son descargados.

En segundo lugar, tenemos los datos que vienen de otras hojas de datos. En su mayoría, estas son las tablas con las que los empleados trabajan en su día a día. Suben la información de sus operaciones diarias, manipulan los datos y lo usan como herramienta de trabajo principal. Al alimentarnos de estas tablas, reducimos el tiempo empleado por estos trabajadores en mantener actualizadas múltiples hojas de datos, permitiéndoles trabajar con una única herramienta que actualiza de forma automática el resto de tablas que van a necesitar en su trabajo.

Por ejemplo, supongamos que un trabajador, se encarga de mantener al día el número de operaciones que cada cliente realiza con la empresa. Esta información resulta muy útil en diferentes campos, y es por ello que tanto el departamento de contabilidad, como el de operaciones y el de finanzas, necesitan que esa información esté actualizada en sus tablas para que cada equipo pueda realizar los análisis necesarios y pertinentes. Sin embargo, las tablas donde albergan esta información son diferentes en cada departamento, y en algunos casos, tratan otra información confidencial a la cual muy pocas personas pueden tener acceso. Además, el departamento de contabilidad necesita que esta información aparezca en tres tablas diferentes y el de finanzas en dos. Es decir, un mismo dato se necesita en seis tablas diferentes, a las cuales el trabajador que posee el dato no tiene porque tener acceso.

En este caso podrían ocurrir varias cosas. Una opción sería que el trabajador metiese en cada tabla individual el dato. No obstante, el meter el mismo dato varias veces aumenta la probabilidad de error, lo cual podría llevar a que en una tabla el número de operaciones en un día para un cliente sean cinco y en otra sean cuatro. Asimismo, el trabajador, que no tiene porque tener acceso a todas las tablas donde se necesite este dato, podría optar por mandarle la cifra a los equipos, pero después estos tendrán que repetir el proceso, aumentando otra vez el trabajo y margen de error. Sería mucho más fácil, rápido y reduciría el margen de error que el dato se tuviese que introducir una sola vez, y de forma automática apareciese en todas las tablas donde se tenga que utilizar. Este es el objetivo de la automatización de estos procesos que conseguimos al generar estos códigos.

8.4.1 Importando tablas de bases de datos mediante SQL

En muchas empresas, existe lo que se conoce como Customer Relationship Management (CRM). Como nos explican Payne y Frow (2005), “el CRM se refiere a emplear

la tecnología para organizar, automatizar, y sincronizar los procesos de ventas, marketing, servicios al cliente, y a dar apoyo técnico.” El objetivo del CRM es mejorar las relaciones con los clientes mediante la centralización de información y la automatización de interacciones. El CRM permite que todos los departamentos tengan acceso a la información actualizada sobre los clientes. No obstante, toda esta información no está siempre disponible en el formato o forma en la que queremos tenerla y además su estructura puede garantizar la eficiencia con la que se realizan análisis y se accede a esta información. A consecuencia de esto, es de gran utilidad integrar la información de CRM en una base de datos SQL. Integrar la información de CRM en una base de datos de SQL permite centralizar los datos, automatizar procesos y mejorar la relación con los clientes (Buttle & Maklan, 2019). En este modelo, aprovechamos estas tablas que se han creado en SQL, y según qué información necesitamos, modificamos las tablas y las adaptamos para después traspasar esta información al código que estamos generando para automatizar nuestros procesos.

Al trabajar con tablas en SQL, somos capaces de filtrar y ajustar la información que tenemos, para sacar diferentes conclusiones y poder analizar de forma rápida y sencilla. Del mismo modo, somos capaces de cambiar lo que vemos según que nos es útil en cada caso.

En nuestro caso, estamos trabajando con varias tablas, de las cuales queremos unificar diferentes trozos de información, limpiar los datos ajustando fechas, rellenando casillas nulas o modificando nombres, para generar la tabla con los datos que finalmente queremos ver plasmados en nuestra hoja de datos. De esta manera, en nuestra hoja de datos ya tendremos la información limpiada y ajustada a nuestras necesidades, lo que reducirá el peso del documento y el esfuerzo que el trabajador que tenga que mirar estos documentos deba poner en identificar qué información le es útil.

En la Figura 5 de la página siguiente, se muestra una de las funciones que se han creado, de forma codificada, para insertar la información que es de utilidad en este caso para el trabajador. En este código identificamos varios elementos que resultan de gran utilidad en el proceso de automatizar los procesos de datos.

Como podemos observar, empezamos por identificar el nombre de la pestaña donde queremos que estos datos terminan pegándose cuando ejecutemos el código. Para ello, usamos una función que accede al documento del cual este App Scripts depende, y busca la pestaña con el nombre indicado. Una vez hecho esto, descargamos todos los datos que genera el código de sql

obtenidos de la URL proporcionada, y lo guardamos en la variable 'values'. Al mismo tiempo, creamos una matriz, 'data', que contiene un solo elemento - el primer elemento de 'values'.

Figura 5: Código de la función 'df_actuals' creado en Google App Scripts

```
function df_actuals(){
  var sheet_name = 'df_actuals';
  var sheet = SpreadsheetApp.getActive().getSheetByName(sheet_name);
  var values = getData("https://popsql.com/share/queries/-identificador_de_la_query");
  Logger.log("Starting...");
  var data = [values[0]];

  for (country = 0 ; country < all_countries.length ; country++){
    for ( i = 1 ; i < values.length ; i++){
      if(values[i][0] == all_countries[country]){
        data.push(values[i]);
      };
    };
    Logger.log("Done with: %s",all_countries[country]);
  };

  if(sheet.getLastRow() > 1 && data.length > 0){
    sheet.getRange(1,1,sheet.getLastRow(),data[0].length+formula[0].length).clearContent();
    SpreadsheetApp.flush();
  };

  if (data.length > 0){
    Sheets.Spreadsheets.Values.update({values: data}, sheet_id, sheet_name, {valueInputOption: "USER_ENTERED"});
    SpreadsheetApp.flush();
  };

  // Paste time tag
  print_last_update('A2:B2', time_zone, sheet_name);

  return 0;
};
```

Fuente: Elaboración propia, en colaboración con Job&Talent

La siguiente línea es fundamental, pues es donde filtramos la información para poder usar un mismo código para diferentes países, empleando así una misma herramienta en las distintas sedes que proporcione sólo los datos que necesitamos en cada país. Las bases de datos que hemos creado en sql almacenan información de todo tipo de clientes para todos los países y con muchos parámetros. En este caso, aunque ya habremos establecido que campos queremos ver, y estos serán los mismos para cada país con el fin de conseguir esa homogeneidad de información entre sedes que busca la empresa, buscamos ahora filtrar por país. Aquí entra en juego la variable que definimos como *all_countries* anteriormente en el archivo 'Variable Definitions'. Este bucle anidado, recorre los datos obtenidos ('values') y, para cada país en la lista 'all_countires', agregan las filas correspondientes a ese país a la matriz data.

Figura 6: Ejemplo de filtración de datos mediante bucle anidado codificado en Pop SQL

country	tax_id	company_name	invoice_id	amount ↑	payment_date	status
SE	TAX_ID_1	Company_name_1	Invoice_ID_1	###,###.##€	202X-XX-XX	PAID
SE	TAX_ID_2	Company_name_2	Invoice_ID_2	###,###.##€	202X-XX-XX	CANCELLED
ES	TAX_ID_3	Company_name_3	Invoice_ID_3	###,###.##€	202X-XX-XX	OVERDUE
SE	TAX_ID_4	Company_name_4	Invoice_ID_4	###,###.##€	202X-XX-XX	PAID
ES	TAX_ID_5	Company_name_5	Invoice_ID_5	###,###.##€	202X-XX-XX	DUE

↓
Tras ejecutarse las líneas de código del bucle anidado de 'all_countries' y values, terminamos con una tabla que solo tiene las filas de los datos que nos interesan en cada situación.

country	tax_id	company_name	invoice_id	amount	payment_date	status
ES	TAX_ID_3	Company_name_3	Invoice_ID_3	###,###.##€	202X-XX-XX	OVERDUE
ES	TAX_ID_5	Company_name_5	Invoice_ID_5	###,###.##€	202X-XX-XX	DUE

Fuente: Elaboración Propia

En la figura 6, se representa que está haciendo este bucle. Si la variable 'values' contiene los datos que se muestran en la primera tabla, en la variable 'data' únicamente se introducirán los que se muestran en la segunda tabla, una vez aplicado el filtro mediante el bucle anidado que ya hemos explicado. La primera tabla contiene la información de los campos mostrados para todos los países, no obstante, nuestro proyecto busca facilitar el reporting financiero a cada delegación, por lo que solo nos interesa la información individualizada de cada país pero no queremos perder la visibilidad que nos da la tabla original, pues para conocer el estado global de la empresa, es importante poder acceder a todos los datos sin filtrar. De esta manera hemos conseguido que a partir de un solo código automáticamente generemos múltiples tablas con los mismos campos, pero solo conteniendo la información que es útil en cada caso (en este proyecto la información de cada país).

Las siguientes líneas de código son las que necesitaremos para poder mantener nuestra pestaña de datos con información actualizada. La primera línea verifica si hay datos en la hoja de cálculo ('sheet') y en la matriz 'data'. Si hay, borra el contenido de ciertas celdas en esta hoja. A continuación, tenemos unas líneas donde sí hay datos en la matriz 'data', actualiza los valores de la hoja de cálculo con los nuevos datos utilizando la API de sheets, que explicamos al principio. Este mismo procedimiento es el que llevaremos a cabo con cada pestaña de datos que queramos llenar con información que provenga de una base de datos en SQL.

8.4.2 Importando información de otras hojas de cálculo

Una de las grandes ventajas de las APIs, es que nos permite integrar datos de diversas fuentes gracias a sus diversas funcionalidades. Venimos explicando la utilidad que tiene en este proyecto el acceder a bases de datos de SQL, pero no es la única fuente de la que tendremos que sacar información.

Aunque el trabajar con bases de datos es algo cada vez más generalizado, el uso de hojas de cálculo sigue siendo una gran herramienta. Aunque es cierto que en Job&Talent, la tecnología es uno de sus enfoques principales, no quita que las hojas de cálculo de cómo Microsoft Excel o Google Sheets, sigan siendo herramientas intuitivas, fáciles de usar y fácilmente accesibles para usuarios que no tienen conocimientos específicos más avanzados. Así mismo, tratándose de una empresa con muchos departamentos y sedes a nivel global, facilita la colaboración en tiempo real.

En línea con esto, en esta empresa hay muchos documentos que se trabajan en diferentes idiomas, y se utilizan para mantener al día las operaciones del país, pero cuya información es útil en otras áreas. Como buscamos minimizar el número de veces que la misma información tiene que ponerse en diferentes documentos, y el tiempo empleado en esto, es de gran utilidad que mediante Google Scripts, recojamos esta información y la traslademos a nuestra hoja de cálculo con los campos, formatos y condiciones que queremos.

En la figura 7, se presenta otra función que se ha creado para importar la información requerida en el documento, pero que en vez de establecer una conexión con las bases de datos que tengamos en SQL, recoge los datos de otras hojas de cálculo de Google Sheets.

Vemos un código parecido al que teníamos para la figura 6. No obstante, rápidamente identificamos la primera diferencia. Mientras que antes hacíamos uso de la JDBC Service de Google Scripts, una biblioteca que permite a los scripts de Google conectarse con bases de datos de SQL, ahora usamos la *clase*⁵ ‘Spreadsheet App’, la cual nos proporciona funciones y métodos para interactuar con Google Sheets.

⁵ Una clase en Google Apps Script es una estructura que agrupa métodos y propiedades para realizar tareas específicas o manipular datos.

Figura 7: Código de la función ‘df_import_merge’ escrito en Google App Scripts

```
function df_import_merge() {
  var sheet_name = "Import Merge";
  var sheet = SpreadsheetApp.openById("identificador_hoja_cálculo");
  var sheetd=SpreadsheetApp.getActiveSpreadsheet();
  var DetinationTab = sheetd.getSheetByName(sheet_name);
  var dict_tabs = ['Import Fuente 1', 'Import Fuente 2', 'Import Fuente 3'];
  var data = [['Account', 'Bank', 'Date', 'Narrative', 'Amount', 'Cash Balance', 'Subcategory']];

  // Take data from every file"
  for(var i=0; i<dict_tabs.length; i++){
    var tab = sheet.getSheetByName(dict_tabs[i]);
    if(dict_tabs[i]=="Import fuente 3"){
      var sourceValues = extract_y_lines(tab.getRange("A2:I").getDisplayValues());
    } else {
      var sourceValues = extract_last_empty_rows(tab.getRange("A2:H").getDisplayValues());
    }
    //Logger.log(sourceValues);
    data = data.concat(sourceValues);
    Logger.log("data length after adding %s is %d", dict_tabs[i], data.length);
  };

  filterDate = new Date('2024-01-01');
  var filteredData = [data[0]];
  for (var i = 1; i < data.length; i++) {
    var rowDataDate = new Date(data[i][2]);
    if (rowDataDate >= filterDate){
      filteredData.push(data[i]);
    };
  };
  // Clear old data in the target sheet
  if (filteredData.length > 0) {
    DetinationTab.getRange(1, 1, DetinationTab.getLastRow(), filteredData[0].length).clearContent();
    SpreadsheetApp.flush();
  };
  // Paste new values
  if (filteredData.length > 0) {
    DetinationTab.getRange(1, 1, numRows, numCols).setValues(filteredData);
  };

  print_last_update('A9:B9', 'Europe/Madrid', sheet_name);

  return 0;
};
```

Fuente: Elaboración propia, en colaboración con Job&Talent

Existen varias funciones dentro de la clase ‘Spreadsheet App’, aunque en este caso, hemos hecho uso de tres:

1. .openById(): lo empleamos para abrir una hoja de cálculo usando su ID que es único e invariable. Otra opción habría sido usar `SpreadsheetApp.openByUrl()`, que lleva a cabo la misma función pero usando en su lugar la URL completa de la hoja de cálculo. La elección de usar uno frente a otro en este caso, es que la segunda opción es más propensa a errores, pues aparte de ser más larga, puede cambiar su estructura.

2. `.getActiveSheet ()`: obtiene la hoja de cálculo que actualmente está abierta, es decir aquella a la cual pertenece este Appscript y en la cual vamos a cargar los datos que estamos recogiendo.
3. `flush ()`: nos asegura que todas las operaciones que estén pendientes se lleven a cabo, es decir, fuerza a que cualquier cambio que se haya llevado hasta ese punto en el script haya terminado de ejecutarse y aplicarse. De esta manera, si tenemos operaciones que se realizan después que dependen de que estos cambios estén hechos, nos aseguramos que los cambios queden reflejados y terminados. Lo usamos para tener un mayor control en scripts donde hay tantas funciones. Se trata de algo de buena praxis que ayuda a evitar errores.

Del mismo modo hacemos también uso de la clase ‘Spreadsheet’, la cual se obtiene a través de ‘SpreadsheetApp’. De esta clase hacemos uso de la una función:

1. `.getSheetByName ()`: Esta función accede a una hoja específica dentro de una hoja de cálculo, buscándola por su nombre.

Finalmente, hacemos uso de una última clase, ‘Range’. Esta clase se emplea cuando queremos representar un rango de celdas en una hoja de cálculo. Trabajamos dentro de su biblioteca con tres funciones:

1. `.getRange ()`: Empleamos esta función para seleccionar un rango específico dentro de una hoja. Anteriormente se explicaba como un equipo puede tener en una hoja de datos muchos datos, de los cuales no todos tienen porque sernos de utilidad para la tarea que nosotros buscamos. Una de las ventajas de conectar hojas de cálculo de esta manera, aparte de la reducción del margen de error, es el filtrar qué campos son útiles en diferentes hojas de cálculo. Al seleccionar solo un rango y no toda la hoja, aligeramos el documento, lo cual es muy importante cuando trabajamos con grandes cantidades de datos, pues reducimos los tiempos de ejecución.
2. `.getDisplayValues ()`: Obtiene los valores visibles en las celdas que se han especificado en el rango. En este código se usa a la par con la función anterior, ‘.getRange’.
3. `.getLastRow ()`: Obtiene el número de la última fila que tiene datos en la hoja o rango especificado. Emplear esto es de mucha utilidad especialmente cuando estamos trabajando con una hoja de datos que no tiene un tamaño fijo, es decir dinámica. El

número de filas que tiene puede aumentar o disminuir, y esto nos permite escribir scripts que se adapten automáticamente al número de finales. Además, evitamos así recorrer filas vacías, que al igual que antes, reduce el tiempo de ejecución al evitar que se corran filas de datos vacías.

Aparte de estas funciones, también hemos hecho uso de otras funciones que no son propias de AppScripts, sino que hemos creado nosotros, y que se encuentran en el archivo 'Functions' que definimos en un principio. Al definirlo anteriormente, podemos llamar esta función todas las veces que queramos en el código, algo que reduce el número de líneas y la complejidad de la programación, lo cual en scripts compartidos es muy importante. Como con todo lo demás, no solo buscamos automatizar procesos, sino simplificar el trabajo y mejorar la eficiencia del empleado. Algo a destacar es también el uso de *SpreadsheetApp* y de *DestinationTab* a lo largo del código. Empleamos el primero siempre que estemos trabajando con la hoja de cálculo de donde queremos sacar la información - nuestra fuente de datos. Emplearemos la segunda cuando hagamos referencia a la hoja de destino, a la que pertenece el script. Nuestro destino final de los datos.

Una vez descompuesto el código con todas sus funciones, vamos a resumir el objetivo y return del código. Buscamos acceder a hojas de cálculo que ya contienen la información que requerimos. Empezamos por acceder a la hoja de cálculo donde tenemos esta información, y establecer conexión con la hoja de cálculo donde vamos a traer esta información. Identificamos tanto las hojas que queremos de la hoja de origen, así como el rango que queremos traernos, y establecemos la hoja de destino dentro de la hoja de cálculo. Aparte de traernos sólo un rango y no la hoja entera, también filtraremos por fecha, de tal manera que terminaremos en una hoja de datos con la información que había en varias hojas de datos en otra hoja de cálculo, unificado todo en uno. Al igual que con el ejemplo anterior, para mantener dinámico el código, implementamos el código que nos borre la información que ya aparezca en la hoja de destino, y pegamos la nueva información.

Resulta importante mencionar también el uso que se hace de la función 'Logger.log'. Se trata de una función específica del entorno de scripting de Google Apps Script que nos permite escribir mensajes de seguimiento. Es una función que emplearemos de forma habitual, pues resulta muy útil para hacer seguimiento de la ejecución de un código y poder identificar en códigos muy extensos donde puede haber un error en la ejecución. En este caso en específico, la usamos para ver cuántas filas de datos tenemos tras importar los datos de una pestaña al

conjunto. Así, si vemos que tras importar una hoja de datos que tiene 500 filas, nuestra pestaña de datos solo ha aumentado en longitud una fila, sabremos que estamos importando o filtrando los datos mal.

8.5 Generando Informes y Copias Automatizadas

Acabamos de terminar de crear todas las funciones que van a permitir que de forma automática, se descarguen los datos de las diversas fuentes que vamos a necesitar para que este equipo pueda llevar a cabo los análisis que necesita. No obstante, otro de los objetivos que tiene marcado el equipo con esta hoja de cálculo es hacer un reporte semanal con la información financiera que aquí recogen.

Estos reportes financieros siempre van a tener el mismo formato, y semanalmente habría que cambiar las tablas y conclusiones que se realizan. Además, los mandan todos los equipos, sin importar el país, y recordemos que en la empresa se busca la integración entre países así como la homogeneidad en los documentos que se realizan. Por tanto, nosotros podemos crear un código que cuando se ejecute, nos genere automáticamente las tablas que vamos a necesitar para ese reporte, de tal manera que sólo quede modificar las conclusiones o información escrita que hay dentro del reporte, y copiar las tablas tal cual se muestran en la hoja de datos, limitando así el tiempo que se dedica a ajustar tablas y montar el documento.

Del mismo modo, necesitan tener una copia de la versión semanal de los balances y demás estados financieros y tablas, que guardan en una carpeta. Al igual que antes, no necesitamos toda la hoja de cálculo, sólo determinadas pestañas. Para esto, podemos crear otra función, que nos cree una hoja de cálculo automáticamente donde se guarda una captura de las tablas que queremos en ese momento específico. La función para llevar a cabo este proceso se muestra en la figura 8.

Figura 8: Creando una función para exportar versión a una carpeta con Google App Scripts

```
function ExportVersion() {
  var source_ss_1 = SpreadsheetApp.openById("id_hoja_de_cálculo");
  var folderId = 'id_carpeta_destino';
  var folder = DriveApp.getFolderById(folderId);

  //Create Report SS
  var date = Utilities.formatDate(new Date(), time_zone, "w");
  var version_name = 'W' + date + ' - Reporting Semanal';
  var report_ss = createSpreadSheetInFolder(version_name, folder);
  var report_id = report_ss.getId();
  var report_ss_Id = 'https://docs.google.com/spreadsheets/d/' + report_id;

  var sourceSheets_1 = source_ss_1.getSheets();
  for (var s = 0; s < sourceSheets_1.length; s++){
    var sourceSheet = sourceSheets_1[s];
    if (!sourceSheet.isSheetHidden()) {
      var sourceSheetName = sourceSheet.getSheetName();
      if (sourceSheetName == 'Hoja 1' || sourceSheetName == 'Hoja 2' || sourceSheetName == 'Hoja 3') {
      };
    };
  };

  if (report_ss.getSheetByName('Sheet1')==null) {
  }else{
  };

  report_ss.deleteSheet(delete_sheet); // Remove the default "sheet1" */
  SpreadsheetApp.getUi().alert(report_ss_Id);
  return 0;
};
```

Fuente: Elaboración propia en colaboración con Job&Talent

Como explicamos arriba, esta función tiene como objetivo capturar la situación de los estados financieros en un momento determinado, creando una versión de la hoja de cálculo con la fecha de la captura. Primero, se accede a la hoja de cálculo y se establece en qué carpeta se creará la copia. Una vez hecho esto, creamos la nueva hoja de cálculo donde vamos a copiar la información, y establecemos que nombre queremos que tome. Como vamos a generar varias versiones a lo largo del tiempo, el nombre de la hoja de cálculo incluirá la fecha en que se genere el documento. La fecha será un elemento variable del nombre, mientras que el nombre base será constante.

A continuación, se seleccionan las pestañas o el rango de información que se incluirá en esta versión. Como hemos mencionado, la ligereza de los documentos de trabajo es esencial, sobre todo cuando tratamos con documentos compartidos con muchas personas y que contienen una gran cantidad de datos. No es conveniente generar una versión que incluya todos los elementos de la hoja de cálculo original, ya que el propósito es proporcionar un documento informativo que muestre la posición de la empresa en un momento específico. Por lo tanto, podemos evitar

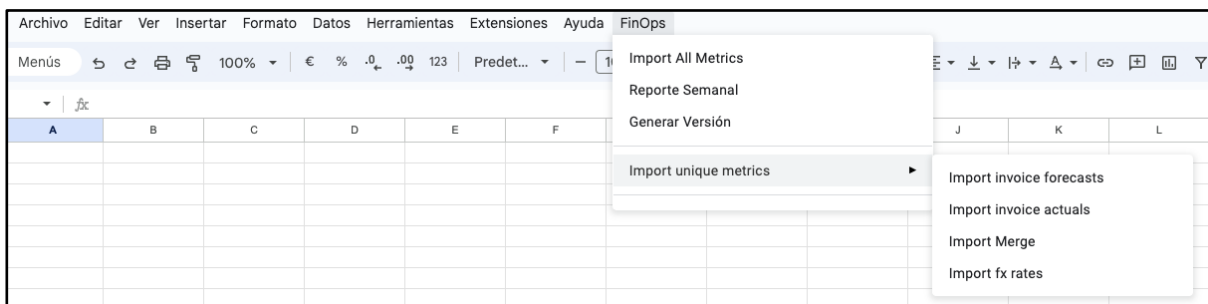
copiar fórmulas y reducir significativamente el tamaño del documento y el tiempo de ejecución de la función.

Además, hemos incorporado un paso nuevo que no se había realizado en funciones anteriores. Al crear una nueva hoja de cálculo, aparece una hoja predeterminada denominada 'Sheet 1' o 'Hoja 1', dependiendo del idioma de trabajo. Para evitar que estas hojas predeterminadas se guarden en las versiones que creamos, el código incluye líneas que las elimina antes de guardar la versión.

8.6 Integrando las funciones como accesos directos a la hoja de cálculo

Hasta ahora, hemos desarrollado todas las funciones necesarias para importar datos desde diversas fuentes, generar versiones de las tablas que seleccionemos y crear hojas de cálculo que incorporen las tablas relevantes para el reporte semanal con un formato específico. Sin embargo, nos falta un último paso que es fundamental para conectar estos códigos y hacerlos fácilmente accesibles desde el cuadro de control que usa el trabajador en su día a día. Para ello, hemos creado una función que añade un nuevo menú desplegable, llamado FinOps, donde agrupamos todas las funciones que hemos desarrollado, facilitando la navegación y el uso de estos nuevos accesos directos.

Figura 9: Captura de pantalla del menú desplegable FinOps



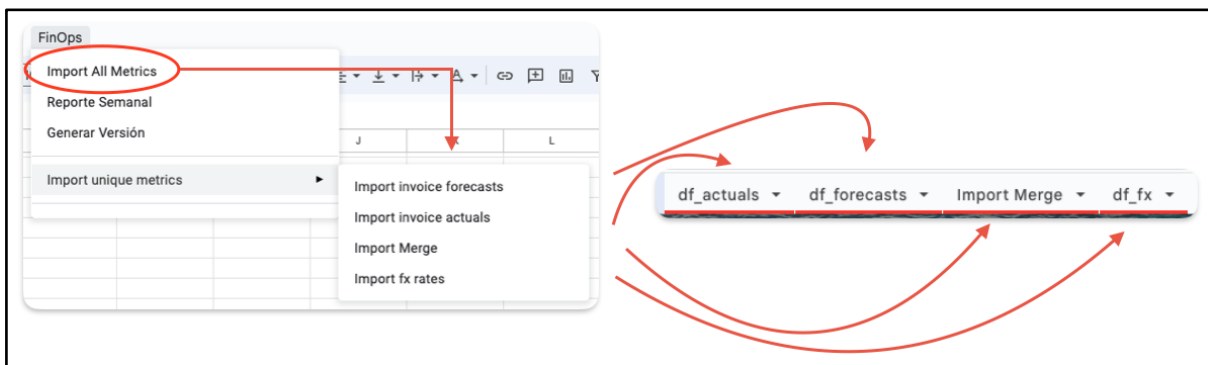
Fuente: Google Sheets

Con este menú, los usuarios que trabajen en la hoja de cálculo podrán abrir el menú desplegable y, con un solo botón, actualizar automáticamente la información de las pestañas de datos, crear el contenido necesario para el reporte semanal y generar una versión que copie la información actual de las pestañas de datos. Para agilizar el trabajo, las pestañas de datos pueden actualizarse todas de una vez, utilizando el acceso directo "Import All Metrics", o bien actualizar solo una de las pestañas de datos mediante el submenú "Import Unique Metrics".

La elección entre estas opciones dependerá del tiempo disponible para la ejecución y de cuánto hayan cambiado los datos en las diferentes pestañas. Cuantas más pestañas se actualicen, mayor será el tiempo de ejecución. Puede haber situaciones en las que solo sea necesario actualizar una pestaña de datos, mientras que el resto de las pestañas no requieran actualización. Este submenú proporciona esa flexibilidad.

En la figura 10, se muestra cómo cada acceso directo importaría los datos a cada pestaña de datos.

Figura 10: Funcionamiento del submenú “Import Unique Metrics”



Base: Elaboración Propia

8.7 Incorporando mecanismos de control

Uno de los grandes problemas que pueden surgir al trabajar con procesos automáticos es que los datos no se actualizan correctamente. El uso de diversas bases de datos que contienen una gran cantidad de información y que dependen de múltiples fuentes aumenta la probabilidad de que ocurran fallos durante la ejecución, lo que puede llevar a que una función no se complete adecuadamente. En algunas ocasiones, es fácil identificar cuando un código no ha funcionado bien porque las filas aparecen vacías o con errores visibles. Sin embargo, cuando se importan miles de filas de datos y el código se detiene a medio camino, puede ser difícil detectar un error en la ejecución antes de comenzar a trabajar con esos datos.

Para evitar la pérdida de tiempo trabajando con información incorrecta o incompleta, es una buena práctica incluir una pestaña que controle la última vez que una función se ejecutó correctamente. Esto tiene una doble funcionalidad: por un lado, permite identificar rápidamente si una función no se ha ejecutado bien, y por otro, muestra cuándo fue la última actualización de la información. De esta manera, cualquier usuario que acceda al documento puede saber inmediatamente la fecha de los datos importados.

Si nos fijamos en las últimas filas de nuestras funciones 'df_actuals' y 'df_import_merge', cuyo código tenemos en la figura 5 y figura 7 respectivamente, veremos que hemos estado llamando a la función 'print_last_update'. Se trata de otra función que no es propia de Google App Scripts pero que hemos creado para poder registrar la última vez que una función ha sido ejecutada.

Figura 11: Función 'print_last_update' creada para Google App Scripts



Fuente: Job&Talent

En la figura 11 se muestra cómo funciona print_last_update, utilizando como ejemplo la llamada a esta función por parte de df_import_merge, previamente ilustrada en la figura 7. Con esta función, accedemos a la pestaña "Control" y, cada vez que es invocada, nos devuelve la fecha y hora en la que una hoja específica (sheet_name) ha sido actualizada. En el caso de df_import_merge, al invocarla, indicamos que registre en el rango de celdas A9:B9 de la pestaña "Control", la fecha y hora de la ejecución. De este modo, logramos tener un mayor control sobre cuándo se actualizó por última vez una hoja de datos y si la actualización se ha realizado correctamente.

8.8 Actualizar tablas de google sheets a bases de datos: las Sheet to Tables

Se conoce como *sheet to table* al proceso de transformar la información almacenada en hojas de datos, como es google sheets, en tablas que después se pueden usar en bases de datos. El almacenar la información en una base de datos frente a una hoja de cálculo, tiene numerosas

ventajas como nos explican Coronel y Morris. Entre otras, cabe destacar la escalabilidad y rendimiento que ofrecen al estar diseñadas para manejar grandes volúmenes de datos de forma rápida y eficiente. Del mismo modo también nos permiten tener un mayor control de los datos que entran dentro de la base de datos, gracias a las diferentes reglas de validación que podemos incorporar para evitar la entrada de datos incorrectos (2018).

8.8.1 Apache Airflow: Un intermediario para la automatización de datos

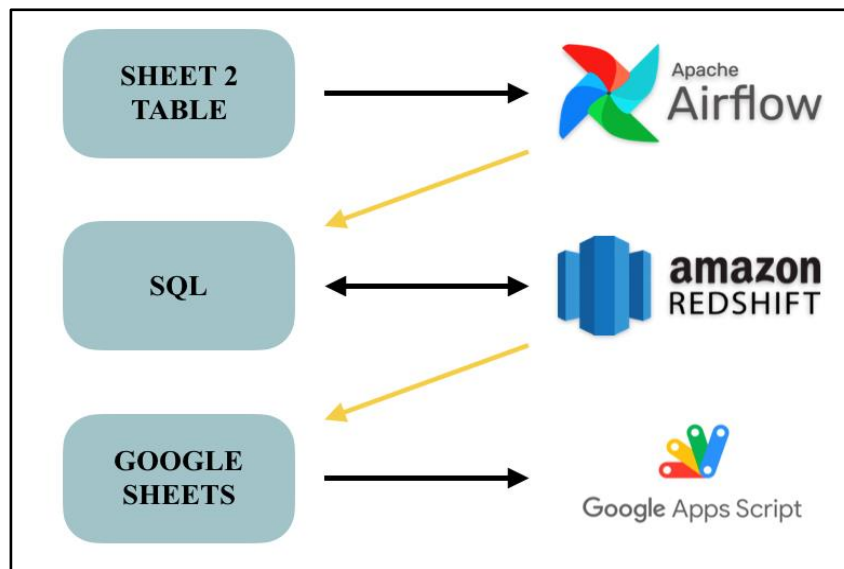
En el caso de Job&Talent, emplean la plataforma Apache Airflow para conectar la información que se guarda en las sheet to tables a las bases de datos en sql. Se trata de una plataforma de código abierto diseñada para gestionar y programar flujos de trabajo de datos de forma programática (Apache Software Foundation, s.f.). Es una plataforma altamente flexible que se usa para un amplio número de tareas de automatización, entre las que se incluyen los procesos de ETL (Extract, Transform, Load) y la automatización de tareas repetitivas.

Airflow y las Sheet to Tables se usan conjuntamente para automatizar y gestionar procesos de datos como el que estamos viendo. En nuestros códigos nos alimentamos de fuentes de datos que provienen tanto de SQL como de otras hojas de cálculo. Para mantener actualizadas las tablas de SQL con la información que se vaya añadiendo diariamente, tendremos que empezar por crear una sheet to table. Una vez creada, la conectaremos a Airflow, de tal manera que de forma programada, la información que aparezca en estas sheet to table, se actualice a las bases de datos que tenemos en SQL.

Del mismo modo que actualizamos nuestras bases de datos, tendremos que mantener actualizadas las hojas de cálculos de donde saquemos información. En muchos casos, serán hojas de datos como las nuestras, que tendrán códigos subyacentes que tendrán que ser ejecutados para actualizar los datos que se contengan en sus hojas. Para ello podremos hacer uso de los triggers de Google Sheets. Los triggers son funciones que se ejecutan automáticamente en respuesta a eventos o condiciones predefinidas. En nuestro caso, queremos hacer uso de triggers de tiempo. Estos son programables para que las funciones se ejecuten en intervalos regulares de tiempo, lo que nos permite recopilar datos de forma periódica y actualizar los datos en nuestras hojas de cálculo.

En la figura 12, se muestra la relación que hay entre las diferentes fuentes de datos y sus correspondientes programas para mantener todos las bases de datos actualizadas.

Figura 12: Flujo de automatización de datos entre herramientas



Fuente: Elaboración propia

Integrando estos elementos, lograremos crear un sistema de automatización en el cual el tiempo que tenga que emplear el trabajador sea muy limitado, siendo su intervención necesaria únicamente en el análisis y generación de los informes. Además, con estos códigos funcionamientos, lograremos reducir el número de veces que los datos tienen que copiarse de otras fuentes de forma manual, reduciendo así los márgenes de error, agilizaremos el proceso de crear los informes semanales, tendremos una plantilla que nos sacará la información de cualquier país que se necesite con solo cambiar las siglas del código, reduciendo el tiempo empleado en crear plantillas de análisis y en homogeneizar estas entre sí. Del mismo modo, tendremos unas hojas de cálculo más ligeras al traernos solo la información que necesitamos en cada análisis, lo que agilizará también los tiempos de ejecución.

En resumen, muchas de las problemáticas que los encuestados señalaron previamente, como la ineficiencia en tareas automatizables comparada con procesos manuales repetitivos, el uso ineficiente del tiempo y la falta de integración entre sistemas y herramientas, se solucionan o mejoran notablemente mediante la implementación de modelos como el que se presenta aquí.

IX. CONCLUSIONES

En este trabajo de investigación, se ha resaltado el uso de herramientas tecnológicas para automatizar tareas manuales y repetitivas, que consumen tiempo, son ineficientes y poco productivas. Hemos estudiado la evolución de la digitalización en las empresas para

comprender el entorno de trabajo actual. Además, hemos explorado diferentes escenarios de computación y almacenamiento, con un enfoque en el cloud computing, destacando sus beneficios y oportunidades para la automatización de procesos de back office.

Se han analizado los diferentes modelos de servicio y se ha proporcionado una descripción de los términos técnicos más relevantes para esta investigación. Un componente fundamental ha sido el estudio de cómo una empresa ha implementado estas herramientas para lograr la eficiencia operativa y permitir a los trabajadores dedicarse a tareas de mayor valor.

En este contexto, se ha examinado el código implementado por el departamento de *financial analytics*, diseñado para ser extrapolable a cualquier país, actualizarse automáticamente, recolectar información de diversas fuentes (hojas de cálculo y bases de datos en SQL), generar informes y mantener un historial de datos.

El objetivo es no solo automatizar el trabajo manual y repetitivo en tareas de back office, sino también reducir el margen de error, acelerar los procesos de carga y las hojas de trabajo, y lograr una homogeneización rápida y fiable en la generación de informes entre las diferentes sedes de la multinacional.

A pesar de los claros beneficios de estas herramientas, se ha realizado una encuesta dirigida a jóvenes, quienes realizan muchas de estas tareas en su jornada laboral, para comprender su percepción sobre la ineficiencia en estas tareas y su conocimiento sobre la capacidad de estas tecnologías para automatizar procesos.

Con la revisión bibliográfica, la encuesta y el estudio del caso de Job&Talent, se ha buscado dar respuesta a las hipótesis planteadas al inicio del estudio. Al combinar estas tres metodologías, se han enriquecido los insights y las conclusiones generadas.

- La automatización de tareas de back office proporciona numerosos beneficios para los empleados, como la reducción de márgenes de error y la optimización de la gestión del tiempo, permitiendo a los trabajadores dedicarse a tareas de mayor valor añadido.
- Aunque la transformación digital y la implementación de herramientas de automatización están presentes en la mayoría de las empresas, tanto los trabajadores como las organizaciones a menudo no aprovechan todas las ventajas disponibles, limitando así la mejora de los procesos.

- Una adecuada automatización de procesos requiere la integración y ejecución de diversas herramientas, así como el acceso a múltiples fuentes de datos. La falta de conocimiento en esta área representa una barrera significativa para la implementación de estos cambios.
- Es posible utilizar códigos sencillos para acelerar procesos laboriosos, pero generar estos códigos por primera vez puede resultar tedioso o complicado. Por lo tanto, es necesario dedicar tiempo a capacitar a los empleados en el manejo de estos lenguajes de programación para maximizar el aprovechamiento de estas herramientas.

Estas conclusiones dan respuesta a las hipótesis que se plantearon al principio.

La primera hipótesis, ‘Existen herramientas de tecnología de la información que permiten automatizar procesos repetitivos e ineficientes dentro de la empresa’. Sí, existen numerosas herramientas que facilitan la automatización de tareas que antes requerían intervención manual, mejorando la eficiencia y reduciendo los errores. Sin embargo, su uso e implementación dentro de las organizaciones aún no está generalizado.

La segunda hipótesis, ‘El empleado pierde tiempo en actualizar bases de datos y compartir información que podría hacerse de forma automática’. Queda aprobada. Los empleados invierten una cantidad significativa de tiempo en tareas que podrían ser automatizadas como son el intercambio de información y la actualización de bases de datos.

La tercera hipótesis, ‘Las empresas pueden incorporar modelos que ayuden con la predicción de datos, generación de reportes y mejorar la conectividad y comunicación’. Efectivamente, hemos visto como herramientas como Google sheets junto con Google App Scripts y sistemas de gestión de bases de datos automatizados pueden encargarse de estas tareas eficientemente, generando predicciones empleando información histórica.

Finalmente, la última hipótesis que se planteó, ‘La automatización de datos reduce los errores y aumenta la eficiencia operativa dentro de la empresa’ queda aprobada. La automatización de datos reduce los errores al minimizar el riesgo de introducir datos incorrectamente al pasarlo de documento a documento. Además, al eliminar la necesidad de intervención manual, aseguramos que los procesos se ejecuten de manera consistente y precisa. Esto libera tiempo para que los empleados se enfoquen en otras tareas, lo cual mejora la eficiencia y productividad organizativa.

Las hipótesis aprobadas ofrecen una valiosa presentación a los beneficios de la automatización de datos, y la creciente sencillez con la que estos procesos pueden ser incorporados por las organizaciones en sus flujos de trabajo, subrayando las percepciones y conocimientos del empleado actual así como analizando un caso donde esto ya se está implementando con éxito.

X. FUTURAS LÍNEAS DE INVESTIGACIÓN

Mientras que este estudio proporciona una buena introducción a cuáles son las posibilidades de la automatización de procesos de back office así como una descripción de cómo estas herramientas pueden aplicarse para automatizar distintas tareas, sería interesante considerar también cuál será el impacto a largo plazo de esta automatización. Esto podría incluir estudios donde se midiese si estos códigos corren sin problemas, o si hay que dedicar mucho tiempo en arreglarlos.

Otro aspecto que sería interesante estudiar sería explorar las capacidades que los empleados tienen que tener para poder escribir y entender los códigos que automaticen estos procesos. Hemos visto por encima que muchas empresas se han digitalizado e incorporado muchos de los avances tecnológicos que ha traído consigo la transformación digital, pero que no han visto una mejora significativa en la eficiencia ni en los procesos internos. Estudiar los motivos por los cuales esto ocurre y buscar soluciones para mejorar la adopción y uso de estas tecnologías sería una interesante línea de investigación.

Finalmente, una posible continuación a esta investigación podría ser estudiar como estos cambios afectan las dinámicas de equipo, las estructuras organizativas y las relaciones laborales de los empleados al transformar la dependencia humana para compartir información entre equipos de trabajo y departamentos.

XI. DECLARACIONES DE USO DE CHAT GPT

Por la presente, yo, Carlota Martínez Baena, estudiante de ADE y Business Analytics de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "Optimización de procesos empresariales a través de la automatización de datos", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. **Brainstorming de ideas de investigación:** Utilizado para idear y esbozar posibles áreas de investigación.
2. **Referencias:** Usado conjuntamente con otras herramientas, como Science, para identificar referencias preliminares que luego he contrastado y validado.
3. **Constructor de plantillas:** Para diseñar formatos específicos para secciones del trabajo.
4. **Corrector de estilo literario y de lenguaje:** Para mejorar la calidad lingüística y estilística del texto.
5. **Generador previo de diagramas de flujo y contenido:** Para esbozar diagramas iniciales.
6. **Sintetizador y divulgador de libros complicados:** Para resumir y comprender literatura compleja.
7. **Generador de encuestas:** Para diseñar cuestionarios preliminares.
8. **Traductor:** Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 15 de Junio de 2024

Firma:  _____

XII. BIBLIOGRAFÍA

- Amazon Web Services. (s.f.). ¿Qué es una API RESTful? Recuperado de <https://aws.amazon.com/es/what-is/restful-api/>
- Angel, N.A.; Ravindran, D.; Vincent, P.M.D.R.; Srinivasan, K.; Hu, Y.-C. Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies. *Sensors* 2022, 22, 196. <https://doi.org/10.3390/s22010196>
- Apache Airflow. (s.f.) Recuperado de 14 de junio de <https://airflow.apache.org/>
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50-58.
- Ben-Gan, I. (2017). *T-SQL Fundamentals*. Microsoft Press.
- Bibook. (2023, November 16). The role of data automation in modern business. Recuperado de <https://get.bibook.com/resources/the-role-of-data-automation-in-modern-business>
- Buttle, F., & Maklan, S. (2019). *Customer Relationship Management: Concepts and Technologies*. Routledge.
- Citrix. (s/f). What is digital transformation? Recuperado de <https://www.citrix.com/glossary/what-is-digital-transformation.html>
- Coronel, C., & Morris, S. (2018). *Database Systems: Design, Implementation, & Management* (13^a ed.). Boston, MA: Cengage Learning.
- Dalton, P. (2020). *SQL Programming and Database Management for Absolute Beginners*. Independently Published.
- Davenport, Thomas H., 1954- & Kirby, Julia. (2016). *Only humans need apply : winners and losers in the age of smart machines* / Thomas H. Davenport and Julia Kirby. New York, NY : Harper Business
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation, University of California, Irvine). Recuperado de https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

- Frey, C. B., & Osborne, M. A. (2013). *The future of employment: How susceptible are jobs to computerization?* Oxford Martin School. Oxford, UK.
- Google Developers. (2023). Google Sheets API. Recuperado de <https://developers.google.com/sheets/api/guides/concepts?hl=es-419>
- Grant, R.M., (1996) Prospering in Dynamically Competitive Environments: Organizational Capability as Knowledge Integration. *Organization Science* 7(4):375-387. Recuperado de: <https://doi.org/10.1287/orsc.7.4.375>
- IRPA&AI. (2024). Definition and benefits. Recuperado el 24 de Mayo, 2024, de <https://irpaa.com/definition-and-benefits/>
- Kääriäinen, J., Parviainen, P., Teppola, S., & Tihinen, M. (2017). Tackling the digitalization challenge: How to benefit from digitalization in practice. *International Journal of Information Systems and Project Management*, 5(1), 63-77. <https://doi.org/10.12821/ijispm050104>
- López, V. (s.f.). *Introducción a la Programación*. Escuela Técnica Superior de Informática, Universidad Autónoma de Madrid. Último acceso: 3 de mayo de 2024.
- Moreira, S., Mamede, H. S., Santos, A. (2023). Process automation using RPA – a literature review. *Procedia Computer Science*, 219, 244-254. <https://doi.org/10.1016/j.procs.2023.01.287>
- McKinsey & Company. (2018). *The keys to a successful digital transformation*. Recuperado de <https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/unlocking-success-in-digital-transformations>
- National Institute of Standards and Technology. (2011). *The NIST Definition of Cloud Computing* (NIST Special Publication 800-145). Gaithersburg, MD: U.S. Department of Commerce. Recuperado de <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- Payne, A., & Frow, P. (2005). A strategic framework for customer relationship management. *Journal of Marketing*, 69(4), 167-176.

- Perez, J. (2023). How Automation Drives Business Growth and Efficiency. Harvard Business Review. Recuperado de <https://hbr.org/2023/04/how-automation-drives-business-growth-and-efficiency>
- Terrar, D., & Singh, N. (2020). Automation: The past, present and future. IBM. Recuperado de <https://www.ibm.com/blogs/think/uk-en/automation-the-past-present-and-future/>
- Torres i Viñals, J. (2012). Del Cloud Computing al Big Data. Universidad Oberta de Catalunya. Recuperado de https://www.soe.uagrm.edu.bo/2018/08/Del.Cloud_.Computing.al_.Big_.Data_.pdf
- Trustpair. (2023). Finance Automation: procesos, benefits, and examples. Recuperado de: <https://trustpair.com/blog/finance-automation/>
- Vilaplana, F., & Stein, G. (2020). Digitalización y personas. Revista Empresa Y Humanismo, 23(1), 113-137. <https://doi.org/10.15581/015.XXIII.1.113-137>

ANEXO I: FORMULARIO DE ESTUDIO SOBRE LA AUTOMATIZACIÓN DE TAREAS DE BACK OFFICE

1. ¿Cuánto tiempo llevas trabajando?

- < 2 años
- 2 - 5 años
- 6 - 10 años
- 11 años - 25 años
- > 25 años

2. ¿En qué departamento trabajas?

- Producción
- Ventas / Comercial
- Finanzas
- Recursos Humanos
- Marketing / Publicidad
- Administración
- Compras
- Otro

3. ¿Has oído hablar de la automatización de procesos de back office?

- Si
- No

4. ¿Con qué frecuencia realizas tareas de back office en tu trabajo?

- Todos los días
- Varias veces a la semana
- Al menos una vez a la semana
- Ocasionalmente
- Nunca

5. ¿Cuáles crees que son las tareas que más fácilmente se podrían automatizar en tu trabajo?

- Entradas de Datos
- Generación de Informes
- Gestión de Documentos
- Archivo y Organización de Datos
- Coordinación de Agendas
- Otra...

6. ¿Cuáles de los siguientes aspectos relacionados con las tareas de back office son los que más observas o experimentas en tu entorno laboral?

- Procesos manuales y repetitivos
- Falta de Integración entre sistemas y herramientas
- Errores de entrada de datos
- Dificultades para acceder y gestionar documentos
- Retrasos en la aprobación y autorización de los documentos
- Falta de seguimiento y control de los procesos
- Ineficiencia en el uso del tiempo
- Coordinación ineficiente entre equipos y departamentos
- Otra...

7. ¿Consideras que la automatización de procesos podría ser beneficiosa para agilizar las tareas de back office, y/o otras tareas en tu empresa?

- Si
- No

Se puede acceder al formulario a través del siguiente enlace: <https://forms.gle/vcSK8cPFcir6Q2Kw9>

ANEXO II: CÓDIGOS GOOGLE APPSCRIPT AUTOMATIZACIÓN PROCESOS

Sección 1: Variables definitions.gs

```
1  const time_zone = 'Europe/Madrid';
2  const sheet_id = SpreadsheetApp.getActive().getId();
3  var all_countries = ['ES'];
```

Sección 2: Menu.gs

```
1  function onOpen() {
2    var ui = SpreadsheetApp.getUi();
3
4    ui.createMenu('FinOps')
5      .addItem('Import All Metrics','import_all_df')
6      .addItem('Reporte Semanal', 'copiarDatos')
7      .addItem('Generar Versión', 'ExportVersion')
8      .addSeparator()
9      .addSubMenu(ui.createMenu('Import unique metrics')
10         .addItem('Import invoice forecasts','df_forecasts')
11         .addItem('Import invoice actuals','df_actuals')
12         .addItem('Import Merge', 'df_import_merge')
13         .addItem('Import fx rates', 'df_fx')
14       )
15     .addSeparator()
16     .addToUi();
17 }
```

Sección 3: df.gs

```
3  function df_actuals(){
4    var sheet_name = 'df_actuals';
5    var sheet = SpreadsheetApp.getActive().getSheetByName(sheet_name);
6    var values = getData("https://popsql.com/share/queries/-identificador_de_la_query");
7    Logger.log("Starting...");
8    var data = [values[0]];
9
10   for (country = 0 ; country < all_countries.length ; country++){
11     for ( i = 1 ; i < values.length ; i++){
12       if(values[i][0] == all_countries[country]){
13         data.push(values[i]);
14       };
15     };
16     Logger.log("Done with: %s",all_countries[country]);
17   };
18
19   if(sheet.getLastRow() > 1 && data.length > 0){
20     sheet.getRange(1,1,sheet.getLastRow(),data[0].length+formula[0].length).clearContent();
21     SpreadsheetApp.flush();
22   };
23
24   if (data.length > 0){
25     Sheets.Spreadsheets.Values.update({values: data}, sheet_id, sheet_name, {valueInputOption: "USER_ENTERED"});
26     SpreadsheetApp.flush();
27   };
28
29   // Paste time tag
30   print_last_update('A2:B2', time_zone, sheet_name);
31
32   return 0;
33 };
```

```

35 function df_forecasts(){
36   var sheet_name = 'df_forecasts';
37   var sheet = SpreadsheetApp.getActive().getSheetByName(sheet_name);
38   var values = getData("https://popsql.com/share/queries/-identificador_de_la_query");
39   Logger.log("Starting...");
40   var data = [values[0]];
41
42   for (country = 0 ; country < all_countries.length ; country++){
43     for ( i = 1 ; i < values.length ; i++){
44       if(values[i][0] == all_countries[country]){
45         data.push(values[i]);
46       };
47     };
48     Logger.log("Done with: %s",all_countries[country]);
49   };
50
51   if(sheet.getLastRow() > 1 && data.length > 0){
52     sheet.getRange(1,1,sheet.getLastRow(),data[0].length+formula[0].length).clearContent();
53     SpreadsheetApp.flush();
54   };
55
56   if (data.length > 0){
57     Sheets.Spreadsheets.Values.update({values: data}, sheet_id, sheet_name, {valueInputOption: "USER_ENTERED"});
58     SpreadsheetApp.flush();
59   };
60
61   // Paste time tag
62   print_last_update('A3:B3', time_zone, sheet_name);
63
64   return 0;
65 };

```

```

67 function df_import_merge() {
68   var sheet_name = "Import Merge";
69   var sheet = SpreadsheetApp.openById("identificador_hoja_cálculo");
70   var sheetd=SpreadsheetApp.getActiveSpreadsheet();
71   var DetinationTab = sheetd.getSheetByName(sheet_name);
72   var dict_tabs=['Import Fuente 1','Import Fuente 2','Import Fuente 3'];
73   var data = [['Account','Bank','Date','Narrative','Amount','Cash Balance','Subcategory']];
74
75   // Take data from every file"
76   for(var i=0; i<dict_tabs.length; i++){
77     var tab = sheet.getSheetByName(dict_tabs[i]);
78     if(dict_tabs[i]=="Import fuente 3"){
79       var sourceValues = extract_y_lines(tab.getRange("A2:I").getDisplayValues());
80     } else {
81       var sourceValues = extract_last_empty_rows(tab.getRange("A2:H").getDisplayValues());
82     }
83     //Logger.log(sourceValues);
84     data = data.concat(sourceValues);
85     Logger.log("data length after adding %s is %d",dict_tabs[i],data.length);
86   };
87
88   filterDate = new Date('2024-01-01');
89   var filteredData = [data[0]];
90   for (var i = 1; i < data.length; i++) {
91     var rowDataDate = new Date(data[i][2]);
92     if (rowDataDate >= filterDate){
93       filteredData.push(data[i]);
94     };
95   };
96   // Clear old data in the target sheet
97   if (filteredData.length > 0) {
98     DetinationTab.getRange(1, 1, DetinationTab.getLastRow(), filteredData[0].length).clearContent();
99     SpreadsheetApp.flush();
100  };
101  // Paste new values
102  if (filteredData.length > 0) {
103    DetinationTab.getRange(1, 1, numRows, numCols).setValues(filteredData);
104  };
105
106  print_last_update('A4:B4', time_zone, sheet_name);
107
108  return 0;
109 };

```

```

111 function df_fx() {
112   var sourceSpreadsheetId = "identificador_hoja_cálculo";
113   var sourceSheetName = "Sheet1";
114   var sourceSheet = SpreadsheetApp.openById(sourceSpreadsheetId).getSheetByName(sourceSheetName);
115   var values = sourceSheet.getRange("A1:F").getDisplayValues();
116   var sheet = SpreadsheetApp.getActiveSpreadsheet();
117
118   var sheet_name = "df_fx";
119   var DetinationTab = sheet.getSheetByName(sheet_name);
120
121   var data = [values[0]];
122
123   for (country = 0 ; country < all_countries.length ; country++){
124     for ( i = 1 ; i < values.length ; i++){
125       if(values[i][0] == all_countries[country]){
126         data.push(values[i]);
127       };
128     };
129     Logger.log("Done with: %s",all_countries[country]);
130   };
131
132   // Clear old data
133   var destinationLastRow = DetinationTab.getLastRow();
134   if (destinationLastRow > 1 && data[0].length > 1) {
135     DetinationTab.getRange(1, 1, destinationLastRow, data[0].length).clearContent();
136     SpreadsheetApp.flush();
137   };
138
139   // Paste new values
140   if (data[0].length > 1) {
141     Sheets.Spreadsheets.Values.update({values: data}, sheet_id, sheet_name, {valueInputOption: "USER_ENTERED"});
142     SpreadsheetApp.flush();
143   };
144
145   // Paste time tag
146   print_last_update('A5:B5',time_zone, sheet_name);
147
148   return 0;
149 };

```

Sección 4: Daily.gs

```
1 function copiarDatos() {
2   // Identifica las hojas de origen y destino
3   var sourceSpreadsheet = SpreadsheetApp.getActiveSpreadsheet();
4   var destinationSpreadsheet = SpreadsheetApp.openById("id_hoja_de_calculo");
5
6   var sourceSheet = sourceSpreadsheet.getSheetByName("Nombre Hoja Origen");
7   var destinationSheet = destinationSpreadsheet.getSheetByName("Nombre Hoja Destino");
8
9   // Prompt the user for the last day of actuals updated using a custom date picker
10  var ui = SpreadsheetApp.getUi();
11  var response = ui.prompt('Introduce el último día para el que hay actuals: (DD/MM/YYYY):', ui.ButtonSet.OK_CANCEL);
12
13  // Check the user's response
14  if (response.getSelectedButton() !== ui.Button.OK) {
15    // User canceled or closed the prompt
16    return;
17  }
18  var dateString = response.getResponseText();
19
20  // Split the input string to get day, month, and year
21  var dateComponents = dateString.split('/');
22
23  // Create a new Date object with the parsed components (Note: months in JavaScript are 0-indexed)
24  var lastDayDate = new Date(dateComponents[2], dateComponents[1] - 1, dateComponents[0]);
25
26  // Validate if the input is a valid date
27  if (isNaN(lastDayDate.getTime())) {
28    ui.alert('Fecha Incorrecta. Porfavor, introduzca una fecha válida (DD/MM/YYYY).');
29    return;
30  }
31  // Pega la última fecha de actualización en la celda B2 del origen y destino
32  sourceSheet.getRange("B2").setValue(lastDayDate);
33  destinationSheet.getRange("B2").setValue(lastDayDate);
34
35  Utilities.sleep(3000);
36
37  // Copia las columnas especificadas
38  var columnsToCopy = ["A64:B110", "C60:EG110"];
39  var destinationColumns = ["A5:B51", "C1:EG51"];
40  for (var i = 0; i < columnsToCopy.length; i++) {
41    // Limpia el rango de destino
42    destinationSheet.getRange(destinationColumns[i]).clearContent();
43
44    // Copia los valores
45    var range = sourceSheet.getRange(columnsToCopy[i]);
46    var values = range.getValues();
47    destinationSheet.getRange(destinationColumns[i]).setValues(values);
48  }
49  // Actualiza la celda B3 con la marca de tiempo actual y la última fecha de actualización
50  var timestamp = new Date();
51  sourceSheet.getRange("B3").setValue(timestamp);
52  destinationSheet.getRange("B3").setValue(timestamp);
53  };
```

```

55 function ExportVersion() {
56   var source_ss_1 = SpreadsheetApp.openById("id_hoja_de_cálculo");
57   var folderId = 'id_carpeta_destino';
58   var folder = DriveApp.getFolderById(folderId);
59
60   //Create Report SS
61   var date = Utilities.formatDate(new Date(), time_zone, "w");
62   var version_name = 'W' + date + ' - Reporting Semanal';
63   var report_ss = createSpreadSheetInFolder(version_name, folder);
64   var report_id = report_ss.getId();
65   var report_ss_Id = 'https://docs.google.com/spreadsheets/d/' + report_id;
66
67   var sourceSheets_1 = source_ss_1.getSheets();
68   for (var s = 0; s < sourceSheets_1.length; s++){
69     var sourceSheet = sourceSheets_1[s];
70     if (!sourceSheet.isSheetHidden()) {
71       var sourceSheetName = sourceSheet.getSheetName();
72       if (sourceSheetName == 'Hoja 1' || sourceSheetName == 'Hoja 2' || sourceSheetName == 'Hoja 3') {
73         var sValues = sourceSheet.getDataRange().getDisplayValues();
74         var numCols = sValues[0].length;
75         sourceSheet.copyTo(report_ss);
76         SpreadsheetApp.flush();
77
78         if (report_ss.getSheetByName('Copy of '+ sourceSheetName)==null) {
79           var destinationSheet = report_ss.getSheetByName('Copia de '+sourceSheetName).setName(sourceSheetName);
80         }else{
81           var destinationSheet = report_ss.getSheetByName('Copy of '+sourceSheetName).setName(sourceSheetName);
82         };
83         // overwrite all formulas that the copyTo preserved
84         // destinationSheet.getRange(1,1,sValues.length,sValues[0].length).setValues(sValues);
85         if (sValues.length > 0){
86           Sheets.Spreadsheets.Values.update({values: sValues}, report_id, sourceSheetName, {valueInputOption: "USER_ENTERED"});
87           SpreadsheetApp.flush();
88         };
89         // Copy cell background colors
90         var sourceRange = sourceSheet.getRange(1, 1, sValues.length, numCols);
91         var sourceBackgrounds = sourceRange.getBackgrounds();
92         var destinationRange = destinationSheet.getRange(1, 1, sValues.length, numCols);
93         destinationRange.setBackgrounds(sourceBackgrounds);
94
95       };
96     };
97   };
98
99   if (report_ss.getSheetByName('Sheet1')==null) {
100     var delete_sheet = report_ss.getSheetByName('Hoja1');
101   }else{
102     var delete_sheet = report_ss.getSheetByName('Sheet1');
103   };
104
105   report_ss.deleteSheet(delete_sheet); // Remove the default "sheet1" */
106   SpreadsheetApp.getUi().alert(report_ss_Id);
107   return 0;
108 };

```

Sección 5: Functions.gs

```

3  function getData(url){
4    var csvString = UrlFetchApp.fetch(url);
5    var datos = Utilities.parseCsv(csvString);
6    return datos;
7  };

```

```

9  function extract_last_empty_rows(raw_data){
10   var i = 0, data = [], col_to_check = 0;
11
12   while(raw_data[i] != null && raw_data[i][col_to_check] != "" && raw_data[i][col_to_check] != NaN){
13     data.push(raw_data[i])
14     i++;
15   };
16   return data;
17 };

```



```

19 function extract_y_lines(raw_data){
20
21     var filtered_data = [];
22
23     for (var i = 0; i < raw_data.length; i++) {
24         var line = raw_data[i];
25
26         // Check if the 9th position of the line contains "Y"
27         if (line.length >= 9 && line[8] === "Y") {
28             del = line.splice(8,1); // Delete auxiliary column
29             filtered_data.push(line);
30             // Logger.log(line);
31         };
32     };
33     // Now, filtered_data contains only the lines with "Y" in the 9th position.
34     return filtered_data;
35 };

```

```

37 function apply_date_format(sheet,cells,format){
38     sheet.getRange(cells).setNumberFormat(format);
39     return 0;
40 };

```

```

42 function createSpreadSheetInFolder(name, folder){
43     var ss = SpreadsheetApp.create(name);
44     var id = ss.getId();
45     var file = DriveApp.getFileById(id);
46     folder.addFile(file);
47     return ss;
48 };

```

```

50 function print_last_update (cell,timezone, tab){
51     var sheet = SpreadsheetApp.getActive().getSheetByName("Control");
52     var timetag = Utilities.formatDate(new Date(), timezone, 'dd-MM-yyyy (HH:mm:ss)').toString();
53     sheet.getRange(cell).setValues([[tab,timetag]]);
54
55     return 0;
56 }

```

Para acceder al AppScript donde se encuentran estos códigos siga el Enlace > Extensiones > Apps Script
 Enlace: [1ywX80pYZ5WMnPKE3bapgTJBMg8k1F4MHu2AZF2Jt6Ss](https://script.google.com/d/1ywX80pYZ5WMnPKE3bapgTJBMg8k1F4MHu2AZF2Jt6Ss)