



GRADO EN ESTADÍSTICA

Trabajo de Fin de Grado

RPA Cognitiva: estudio y aplicación

Departamento de Ciencias de la Computación e Inteligencia Artificial

Dirigido por:

Luis Valencia Cabrera

Autor: **Jerohan Velasco Dorado**

Curso 2021/2022

ÍNDICE GENERAL

1 INTRODUCCIÓN Y OBJETIVOS.....	1
1.1 Objetivos.....	2
2 RPA: ESTADO DEL ARTE.....	3
2.1 Breve historia de RPA.....	3
2.2 Qué es y en qué consiste RPA.....	4
2.3 Cómo y cuándo emplear RPA.....	5
2.4 Ventajas e inconvenientes de RPA.....	6
2.5 Hiperautomatización: el siguiente paso de RPA.....	7
3 INTELIGENCIA ARTIFICIAL.....	10
3.1 Breve historia.....	10
3.2 Acotación de nuestro caso.....	11
3.2.1 Redes neuronales artificiales.....	11
3.2.2 Redes neuronales convolucionales.....	17
3.2.3 Procesamiento del lenguaje natural.....	23
3.2.4 Reconocimiento Óptico de Caracteres (OCR).....	26
4 PRESENTACIÓN DEL CASO DE ESTUDIO.....	28
4.1 Introducción.....	28
4.1.1 Software empleado.....	28
4.2 Caso.....	31
4.2.1 Abordaje del caso.....	31

5 IMPLEMENTACIÓN DE IA EN RPA	32
5.1 Introducción.....	32
5.2 Modelo de clasificación propuesto	33
5.3 Motor OCR propuesto.....	34
5.4 Implementación en Blue Prism.....	36
6 RESULTADOS	45
6.1 Resultados de los modelos.....	45
6.1.1 Clasificación	45
6.1.2 Extracción de información.....	49
6.2 Resultados de la automatización de procesos.....	51
6.2.1 Proceso sin RPA.....	52
6.2.2 Proceso con RPA atendida	52
6.2.3 Proceso hiperautomatizado	54
7 CONCLUSIONES	56
7.1 Respecto a los modelos planteados	56
7.2 Respecto a la implementación.....	60
7.3 Comentarios personales	61
8 BIBLIOGRAFÍA	62

9 ANEXO65

9.1 Blue Prism 65

9.1.1 Construcción de objetos en Blue Prism66

9.1.2 Desarrollo de flujos de procesos 70

9.1.3 Módulos.....80

9.2 Process Definition Document (PDD) 80

9.2.1 Descripción general del proceso80

9.2.2 Análisis técnico.....82

9.2.3 Requisitos de sistema.....84

9.2.4 Paso a Paso del proceso.....85

9.3 Solution Definition Document (SDD) 93

9.3.1 Descripción de la solución93

9.3.2 Alcance del proceso.....94

9.3.3 Flujo de datos y aplicaciones94

9.3.4 FlowChart.....95



TFG Grado en Estadística

RPA Cognitiva: estudio y aplicación

Jerohan Velasco Dorado



ÍNDICE DE ILUSTRACIONES

Ilustración 1.1: RPA (Robotic Process Automation)	1
Ilustración 2.1: Búsquedas del término RPA desde 2015.....	3
Ilustración 2.2: Búsquedas del término hyperautomation desde 2017.....	4
Ilustración 2.3: Evolución RPA-AI	7
Ilustración 2.4: Resolución de problemas: RPA vs ML[14]	8
Ilustración 2.5: Relación entre RPA, IA & ML[14].....	9
Ilustración 3.1: Representación de una neurona biológica.....	12
Ilustración 3.2: Estructura jerárquica de un sistema basado en RNA.[17].....	13
Ilustración 3.3: Esquema general de una red neuronal artificial	13
Ilustración 3.4: Esquema de una neurona artificial.....	15
Ilustración 3.5: Ejemplo de separación de datos en entrenamiento, validación y test.....	17
Ilustración 3.6: Capas de una CNN	18
Ilustración 3.7: Arquitectura de una CNN basada en el modelo VGG16 [24]	18
Ilustración 3.8: Ejemplo de capa con 32 filtros. [23].....	20
Ilustración 3.9: Submuestreo por max-pooling de dimensión 2. [23].....	21
Ilustración 3.10: Ciclos de convoluciones en una CNN. [25]	21
Ilustración 3.11: Proceso de aplanamiento.....	22
Ilustración 3.12: Proceso completo de una CNN.....	22
Ilustración 3.13: Ejemplo de documento etiquetado con extracción de información. [27].....	25
Ilustración 3.14: Ejemplo de salida del motor PaddleOCR.....	27
Ilustración 4.1: Ventana de creación de procesos de Blue Prism	29
Ilustración 4.2: Ventana de temporizador de ejecución de procesos de Blue Prism	29
Ilustración 4.3: Ventana de gestión de proveedores de Contoso Invoicing	30
Ilustración 4.4: Ventana de gestión de facturas de Contoso Invoicing.....	30
Ilustración 5.1: Estructura general del modelo propuesto [32].....	33

Ilustración 5.2: Resumen del pipeline de PaddleOCR [34]	34
Ilustración 5.3: Ejemplo de salida de PaddleOCR formateada en tabla.....	35
Ilustración 5.4: Parsing de la salida a un JSON con la información requerida	36
Ilustración 5.5: Preparando los datos de entrenamiento - creando la estructura de carpetas.	37
Ilustración 5.6: Creación del modelo holístico	38
Ilustración 5.7: Entrenamiento del modelo holístico	39
Ilustración 5.8: Reconocimiento del elemento donde BP debe insertar el nombre del documento	40
Ilustración 5.9: Blue Prism reconoce adecuadamente el elemento que necesitamos	41
Ilustración 5.10: Reconocimiento del botón Run	42
Ilustración 5.11: Salida de la clasificación de una imagen	43
Ilustración 5.12: Formato html de la salida obtenida	44
Ilustración 6.1: Especificaciones del modelo holístico	46
Ilustración 6.3: Precisión del modelo holístico sobre el dataset de evaluación	47
Ilustración 6.4: Entrenamiento del modelo por regiones: Cabecera	47
Ilustración 6.5: Entrenamiento del modelo por regiones: Pie	48
Ilustración 6.6: Entrenamiento del modelo por regiones: parte izquierda	48
Ilustración 6.7: Entrenamiento del modelo por regiones: parte derecha	48
Ilustración 6.8: Entrenamiento del modelo final	49
Ilustración 6.9: Precisión del modelo final sobre el dataset de evaluación	49
Ilustración 6.10: SROIE dataset: imagen de un recibo	49
Ilustración 6.11: SROIE dataset: OCR de un recibo	50
Ilustración 6.12: SROIE dataset: información extraída de un recibo	50
Ilustración 6.13: Especificaciones del modelo.....	50
Ilustración 6.14: Parte final del proceso de entrenamiento del modelo de extracción de información	51
Ilustración 6.15: Resultados del modelo de extracción de información sobre el conjunto de evaluación	51
Ilustración 6.16: Tiempo de ejecución del proceso.....	53

Ilustración 6.17: Eliminar documentos marcados como excepciones en la BBDD de Blue Prism	54
Ilustración 7.1: Métricas de DocFormer extraídas de	56
Ilustración 7.2: Implementación de Docformer en R en desarrollo	57
Ilustración 9.1: Ventana de procesos y objetos en Blue Prism	65
Ilustración 9.2: Método attach del objeto Contoso_Global	66
Ilustración 9.3: Ventana para elegir el tipo de aplicación en el Modeller	66
Ilustración 9.4: Ventana para elegir el modo de ejecución en el Modeller	67
Ilustración 9.5: Ventana para insertar la ruta de acceso a la aplicación en Modeller	67
Ilustración 9.6: Application Modeller para Contoso	68
Ilustración 9.7: Detección de elementos en Modeller	69
Ilustración 9.8: Configuración de opciones de identificación de elementos en Modeller	69
Ilustración 9.9: Vista general del proceso orquestador	70
Ilustración 9.10: Herramientas de flujo de Blue Prism	71
Ilustración 9.11: Método WriteAccountsData del objeto Contoso_Global	71
Ilustración 9.12: Propiedades de Start en el método WriteAccountsData	72
Ilustración 9.13: Configuración para hacer click en New Record	73
Ilustración 9.14: Configuración para introducir datos en la tabla de Proveedores de Contoso	73
Ilustración 9.15: Configuración para hacer click en Save Changes	74
Ilustración 9.16: Propiedades de End del método WriteAccountsData	74
Ilustración 9.17: Página principal del subproceso de gestión de facturas	75
Ilustración 9.18: Primera tarea del subproceso Facturas	76
Ilustración 9.19: Segunda tarea del subproceso Facturas	77
Ilustración 9.20: Tercera tarea del subproceso Facturas	77
Ilustración 9.21: Página para detectar elementos en cola	78
Ilustración 9.22: Propiedades de Start de la página Pending Items	78
Ilustración 9.23: Ventana de propiedades de la acción Obtener Datos: Inputs y Outputs	79
Ilustración 9.24: Ventana de verificación de elementos en cola	79



TFG Grado en Estadística

RPA Cognitiva: estudio y aplicación

Jerohan Velasco Dorado



RESUMEN

En los últimos 5 años muchas empresas han decidido dar el salto a la digitalización de la mano de RPA o Automatización Robótica de Procesos. Esto ha permitido que grandes empresas mejoren el servicio a sus clientes, reduzcan costes o puedan ampliar el volumen de trabajo. La razón de la adopción de esta tecnología en lugar de otra es la increíble versatilidad y capacidad de adaptación que tiene, puesto que permite que un software desarrolle el trabajo que un humano haría en un entorno digital *as is*, o sea, tal y como lo desempeñaría el humano. La ventaja de este hecho es que permite su implementación sin tener que modificar el entorno de trabajo, ahorrando el coste que tendría la adquisición de nuevas infraestructuras, software, cursos de formación para empleados, etc.

Por otro lado, también se ha vivido en estos últimos años un auge en el desarrollo de soluciones basadas en inteligencia artificial. Las grandes empresas tecnológicas han invertido muchos recursos en avanzar en determinadas tecnologías empleando técnicas basadas en Machine Learning o Deep Learning. Y no solo se trata del desarrollo real que ha tenido la inteligencia artificial, sino de la popularidad que ha alcanzado a nivel social. Ahora todas las empresas quieren “sumarse al carro” de la IA como símbolo de calidad y vanguardia en sus productos.

Bajo estas premisas se pretende estudiar la sinergia que se produce cuando combinamos RPA con inteligencia artificial en la aplicación de un caso ficticio pero lo más realista posible. Para ello, se desarrollará un caso práctico de automatización de un proceso al que se le aplicará RPA y RPA + IA para comparar sus resultados y analizar de forma aplicada la ventaja que estas soluciones pueden proporcionar.

ABSTRACT

In the last 5 years, many companies have decided to digitalise their processes through RPA or Robotic Process Automation. This has allowed large companies to improve service to their customers, reduce costs or increase the workload. The reason for the adoption of this technology and no other is its incredible versatility and adaptability, as it allows software to perform the work that a human would do in a digital environment "as is", that is, as a human would do it. The benefit of this is that it can be implemented without modifying the work environment, saving the cost of new infrastructure, software, training courses for employees, etc.

On the other hand, there has also been a boom in recent years in the development of solutions based on artificial intelligence. Large technology companies have invested a lot of resources in developing certain technologies using techniques based on Machine Learning or Deep Learning. And it is not only about the real development that artificial intelligence has had, but also about the popularity it has achieved on a social level. Now all companies want to join the AI bandwagon as a symbol of quality and avant-garde in their products.

On these premises, the aim is to study the synergy that is produced when we combine RPA with artificial intelligence in the application of a scenario that is fictitious but as realistic as possible. To this purpose, a practical case of automation of a process to which RPA and RPA + AI will be applied will be developed to compare the results.

1 INTRODUCCIÓN Y OBJETIVOS

La clave del éxito en los negocios está en saber hacia dónde se dirige el mundo, y llegar ahí primero.

-Bill Gates-

En la carrera por adelantarse a la transformación digital, la automatización robótica de procesos o, por sus siglas en inglés, RPA (*Robotic Process Automation*) es una novedosa tecnología que permite, de forma rápida y sencilla, agilizar los procesos críticos de una empresa. En un sentido amplio, RPA facilita la creación y uso de una fuerza de trabajo virtual, flexible y escalable que imita el comportamiento humano, lo que posibilita la interacción con los sistemas ya existentes y que, finalmente, se traduce en una reducción de los tiempos de procesamiento o respuesta, de los errores humanos y, por tanto, del coste de un proceso.



Ilustración 1.1: RPA (Robotic Process Automation)

Habitualmente tendemos a asociar el término *robot* con intrincadas máquinas industriales de cadenas de montaje o como una máquina antropomórfica sacada de una novela futurística, sin embargo, en nuestro caso hablamos de *software*, es decir, scripts y algoritmos más o menos complejos para realizar determinadas tareas. [1]

Por otro lado, si bien no todos los procesos son susceptibles de ser automatizados, hay que destacar que la aplicación de tecnologías basadas en inteligencia artificial ha ampliado y mejorado las aplicaciones de RPA, permitiendo realizar tareas cognitivas, es decir, tareas que imitan el funcionamiento del cerebro humano en factores como el aprendizaje, el razonamiento, la resolución de problemas, la toma de decisiones o el procesamiento del lenguaje.

1.1 Objetivos

Los objetivos que se persiguen con este trabajo son los siguientes:

- Analizar y compartir el estado actual de la tecnología RPA, sus posibles aplicaciones y su proyección en un futuro cercano.
- Explorar y mostrar el funcionamiento de las ramas de la inteligencia artificial con las que se pueden desarrollar proyectos de clasificación y extracción de información de documentos en imágenes.
- Crear un caso práctico y desarrollar una solución de automatización de procesos usando un software específico de RPA.
- Realizar una implementación en el proceso automatizado de las técnicas y tecnologías de inteligencia artificial exploradas.
- Estudiar la sinergia producida por la integración de IA y RPA y comparar los resultados obtenidos con los del proceso automatizado y sin automatizar.
- Extraer conclusiones sobre el impacto de la aplicación de RPA a un proceso sin automatizar así como de su evolución al integrar tareas cognitivas a través de inteligencia artificial.

2 RPA: ESTADO DEL ARTE

La gente no asume que su trabajo acabará haciéndolo las máquinas, en 20 años.

-Bill Gates (2014)-

2.1 Breve historia de RPA

Un breve repaso por la historia de RPA nos revela que sus comienzos se remontan a principios de los 2000. En aquel entonces se trataba de una tecnología incipiente y poco conocida que apenas podía cubrir tareas muy básicas. No fue sino hasta el 2015, cuando su desarrollo fue suficiente, que las empresas comenzaron a interesarse por el RPA como una opción robusta en la digitalización y automatización estable de sus procesos. [2]

Un rápido estudio de las búsquedas en Google del término “RPA” nos muestra claramente cómo, a partir del 2015, el interés en RPA sube progresivamente. [3]

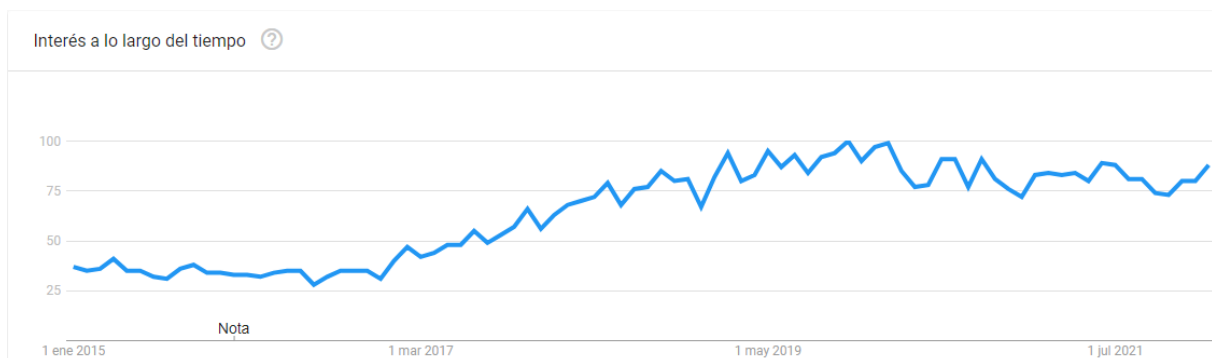


Ilustración 2.1: Búsquedas del término RPA desde 2015

De este gráfico destacan 3 puntos clave:

1. Desde finales de 2016 hay un aumento constante del interés por RPA
2. A finales de 2019 este interés creciente se estabiliza
3. El interés queda estabilizado en valores altos (entre 75-100), por lo que se deduce que es un término popular y, por tanto, muy buscado.

¿Qué ocurre en 2019? Gartner, una empresa consultora y de investigación de las tecnologías de la información con sede en Stamford, publica una lista con las 10 tendencias tecnológicas estratégicas para 2020 y sitúa en primera posición la hiperautomatización, definiéndola como el próximo paso en la automatización de procesos.[4], [5]

En el siguiente gráfico de Google trends se aprecia cómo, a partir de la publicación de esta lista, la búsqueda del término *hyperautomation* comienza a crecer, coincidiendo con la estabilización del número de búsquedas del término *RPA*. [6]



Ilustración 2.2: Búsquedas del término *hyperautomation* desde 2017

2.2 Qué es y en qué consiste RPA

Es complicado dar una definición única y universal sobre RPA, dada la falta de literatura formal acerca de esta tecnología y las variadas descripciones que las grandes empresas proveedoras de este servicio ofrecen. Además, tampoco se puede definir RPA como la evolución de una única tecnología diseñada exclusivamente para la automatización, sino más bien es el resultado de una amalgama de tecnologías que se han ido combinando hasta formar el conjunto de herramientas que conocemos por RPA [2]. Por tanto, empezaremos definiendo qué es un *robot*.

“Un robot es un ente artificial, de comportamiento adaptable y que, actuando sobre su entorno, es capaz de realizar de forma total o parcialmente autónoma tareas que, en otro caso, realizarían personas y ejecutando esas tareas de una forma similar a como las realizaría un humano.”[7]

Aplicando esta definición a nuestro caso, podemos decir que un robot es un software programable para realizar determinadas tareas tal y como las haría un humano, de forma independiente o asistida, en un determinado entorno. Por tanto, una posible y sencilla forma de definir RPA sería decir que se trata de una tecnología que permite crear un trabajador digital que puede adaptarse a cualquier entorno computacional para realizar tareas sencillas y repetitivas sin necesidad de descansar, *sin cometer errores* y con un coste de implementación menor al de un humano.

Entonces, ¿en qué consiste RPA? En la automatización de tareas sencillas y repetitivas, dentro de un proceso de negocio, a través de un robot software que gestiona un variado conjunto de herramientas con el objetivo de mejorar el retorno de inversión de dicho proceso.

En el Anexo, [sección 9.1](#), podemos encontrar una explicación detallada con ejemplos.

2.3 Cómo y cuándo emplear RPA

La tecnología RPA está diseñada para automatizar tareas repetitivas en procesos de negocio que siguen reglas bien definidas, estructuradas y específicas cuya carga de trabajo manual sea importante. Las características típicas de los procesos susceptibles de ser automatizados son aquellos cuyas tareas: [8]

- Están basadas en reglas.
- Son propensas a errores o tienen una baja tolerancia a los fallos.
- Tienen entradas de datos estandarizadas.
- Se desarrollan en varias aplicaciones no integradas entre sí.
- Se ejecutan con mucha frecuencia y ocupan un alto volumen de tiempo.
- Son estables y presentan pocas excepciones.

La mayoría de los procesos en casi todos los sectores cumplen una o varias de estas características, por lo que existen muchos procesos automatizables. Los pasos para la automatización de un proceso se rigen por la siguiente estructura: [8]

- **Identificación:** es fundamental determinar qué procesos pueden ser automatizados, es decir, cumplen las características antes mencionadas.
- **Viabilidad:** se estudia en profundidad la estabilidad del proceso que queremos automatizar, las excepciones que puede presentar, la forma en que recibe los datos o la necesidad de interacción humana para una toma de decisiones. Esto determina la complejidad de la automatización del proceso y su posible coste de implementación.
- **Análisis ROI:** se estudia la relación coste/beneficio actual del proceso teniendo en cuenta, entre otros factores, las personas necesarias para su desarrollo o tiempo de ejecución y se compara con la relación coste/beneficio de su automatización.
- **Diseño e implementación:** en este punto se desarrollan los robots, se hacen las pruebas oportunas de funcionamiento y se define el modelo operativo para monitorizarlos, mantenerlos en producción y auditar sus resultados.
- **Seguimiento:** se realiza un seguimiento de los resultados de la implementación de la automatización para determinar su rentabilidad continuada en el tiempo, a medio y largo plazo.

Hay que destacar que existen 3 formas de automatizar un proceso, que dependen de la naturaleza propia del proceso:

1. **Automatización desatendida:** es una forma de automatización que no requiere la supervisión ni la intervención humana. El robot es completamente independiente y realiza las tareas *end-to-end* exactamente como haya sido configurado, sin descansos ni excepciones. Se suelen usar en tareas de back office. [8], [9]

2. **Automatización atendida:** es una forma de automatización que requiere de la intervención o supervisión de una persona. El robot es un soporte para el humano, ayudándole a realizar determinadas tareas repetitivas dentro de un proceso que requiere toma de decisiones, son altamente variables o presenta muchas posibles excepciones.[8], [9]
3. **Hiperautomatización:** es una forma de automatización que aprovecha y combina el uso de tecnologías de inteligencia artificial. RPA convencional imita las acciones humanas, y necesita para ello datos estructurados, pero si se combina con inteligencia artificial, que imita el intelecto humano, se amplía drásticamente la cantidad de procesos automatizables y la calidad de los resultados de la automatización al poder procesar datos semiestructurados o sin estructura. Esto permite a las empresas automatizar casi cualquier tarea repetitiva y escalable.[5]

2.4 Ventajas e inconvenientes de RPA

El uso de RPA tiene una serie de ventajas para la empresa, pero también presenta una serie de inconvenientes que se deben tener en cuenta. Algunas de sus ventajas son:[11], [12]

- **Rentabilidad:** es la principal ventaja del uso de RPA. Como ya dijimos, un robot puede realizar tareas sin descanso con un coste inferior al de un humano, por lo que mejora el rendimiento y, por tanto, la rentabilidad de las operaciones.
- **Eficacia:** los robots no se equivocan nunca, siempre que las reglas estén bien definidas sobre un proceso completamente determinista. Además, a diferencia de los humanos, pueden realizar más tareas simultáneamente en un menor tiempo. Esto suele repercutir en un mejor servicio al cliente y un aumento de los beneficios.
- **Eficiencia:** dejar las tareas repetitivas y de bajo nivel a los robots permite que los empleados centren sus esfuerzos en tareas de mayor impacto para la empresa, permitiéndoles ser más creativos y eficientes en su trabajo.
- **Seguimiento y control:** Los robots dejan constancia de todas sus acciones y resultados, ofreciendo así una poderosa herramienta para el análisis de negocio y la toma de decisiones.
- **Fácil implementación:** otra de las grandes ventajas de RPA es su facilidad de implementación y adaptabilidad al entorno. Es muy fácil de configurar, sin requerir ninguna habilidad de programación (salvo casos muy concretos). Es un software no invasivo con capacidad de actuar en cualquier entorno digital.

Por el contrario, las desventajas que presenta son las siguiente: [11], [12]

- **Inversión inicial:** la inversión inicial de desarrollo e implementación, así como el mantenimiento del robot son altos. Aunque la inversión se amortice de forma rápida, el coste inicial puede suponer una gran barrera para pequeñas y medianas empresas.

- **Miedo al reemplazo:** la visión que muchos empleados tienen de la automatización es que reemplazan humanos, por lo que acabarán perdiendo su trabajo. Sin embargo, la finalidad de RPA es permitir a los empleados a enfocarse en tareas de mayor exigencia y creatividad.
- **Seguridad:** es importante realizar un estudio sobre la forma en que los robots acceden y manejan información sensible para evitar violaciones de datos y brechas en la ciberseguridad.
- **Grandes cambios:** mientras que un humano puede adaptarse con relativa facilidad a un gran cambio en la tecnología actual de la empresa (cambio de software o de sistema de trabajo), en el caso de los robots puede llegar a ser necesario implementar un nuevo robot adaptado a los nuevos cambios, con todos los costes que eso conlleva.
- **Limitaciones:** al contrario que un humano, un robot no puede tomar decisiones por sí mismo, por lo que su radio de acción está limitado a tareas estables con reglas muy bien definidas. No obstante, la aplicación de inteligencia artificial amplía las capacidades de la automatización.

2.5 Hiperautomatización: el siguiente paso de RPA

Como ya adelantamos, la hiperautomatización o “automatización inteligente” consiste en la combinación de RPA e inteligencia artificial para aprovechar la sinergia que se produce entre ambas tecnologías. Hasta ahora hemos hablado de RPA e inteligencia artificial como dos tecnologías completamente diferentes que pueden trabajar unidas, pero esto no es del todo exacto.

En la mayoría de las referencias y artículos que hablan de este tema, tratan a RPA y a la inteligencia artificial como extremos de un proceso evolutivo. Esta aproximación es comprensible desde el momento en que se reduce la extensa amplitud de campos que resulta ser la inteligencia artificial a “Machine Learning” casi exclusivamente. Esta idea se representa habitualmente con la siguiente ilustración. [13], [14]

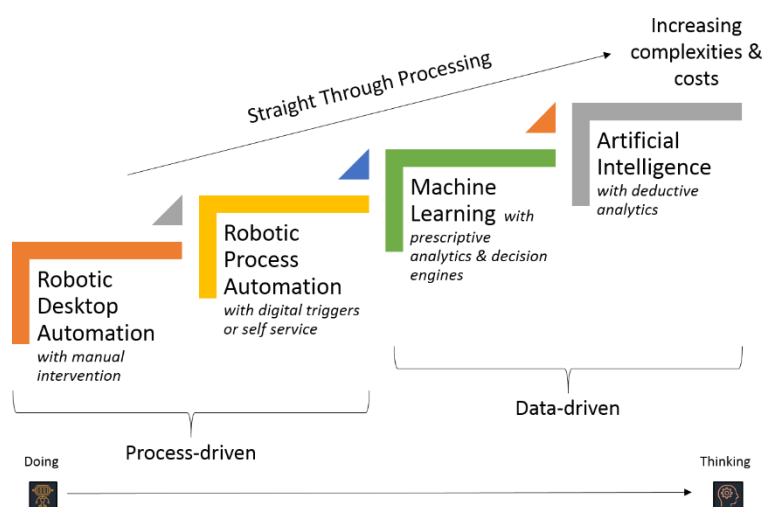


Ilustración 2.3: Evolución RPA-AI

Por otro lado, tal y como vemos en el siguiente flujo, se tiende a comparar RPA con Machine Learning en la forma de abordar un mismo problema y, por tanto, en los resultados de su resolución.

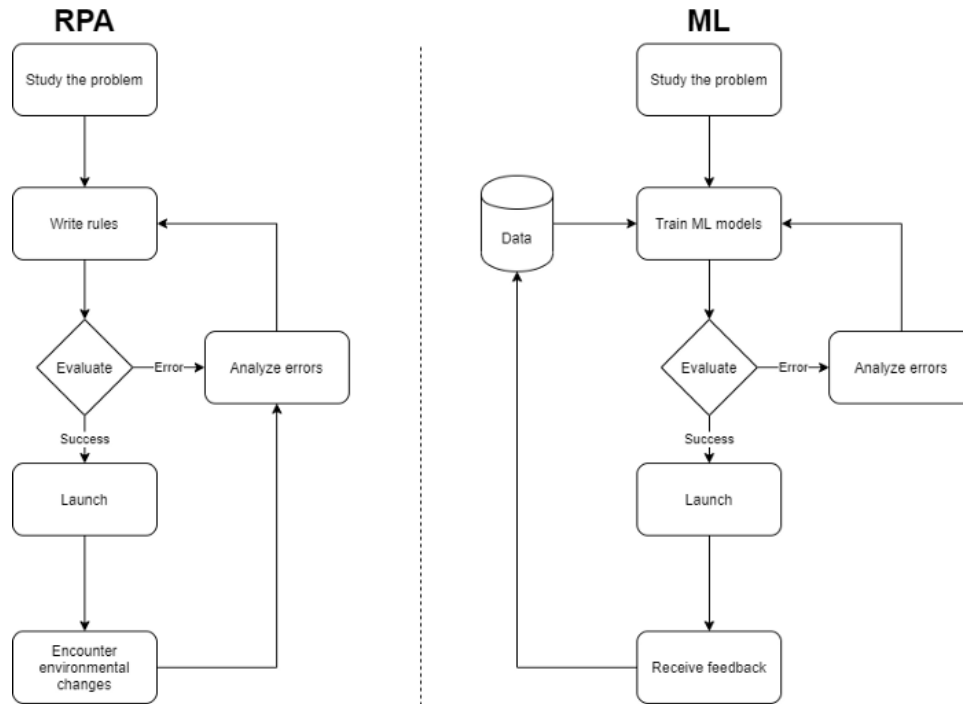


Ilustración 2.4: Resolución de problemas: RPA vs ML[14]

Sin embargo, a pesar de que la comparación es razonable, hay que tener en cuenta 2 puntos:

- La inteligencia artificial se puede definir como la habilidad de una máquina para replicar el comportamiento humano (o de un ser vivo). Esto incluye tanto lo relacionado con el pensamiento como lo relacionado con los patrones de comportamiento. Por tanto, RPA es, en realidad, una forma muy básica de inteligencia artificial, ya que consiste en replicar el comportamiento de un humano en un entorno digital.[14]
- RPA y el Machine Learning son dos técnicas complementarias para abordar un problema completo y complejo, no dos formas distintas de resolver un mismo problema. La consecuencia de comprender RPA y Machine Learning de esta forma es que permite a empresas que ya han hecho inversiones en inteligencia artificial para desarrollar esa rama de su negocio de forma totalmente independiente a las inversiones realizadas en desarrollar RPA, promover el trabajo conjunto de ambos departamentos para alcanzar una automatización inteligente sin necesidad de pasar por el costoso proceso de empezar con RPA e ir invirtiendo nuevamente en IA.[14]

La idea que se desprende de los dos puntos anteriores se puede ilustrar de la siguiente forma

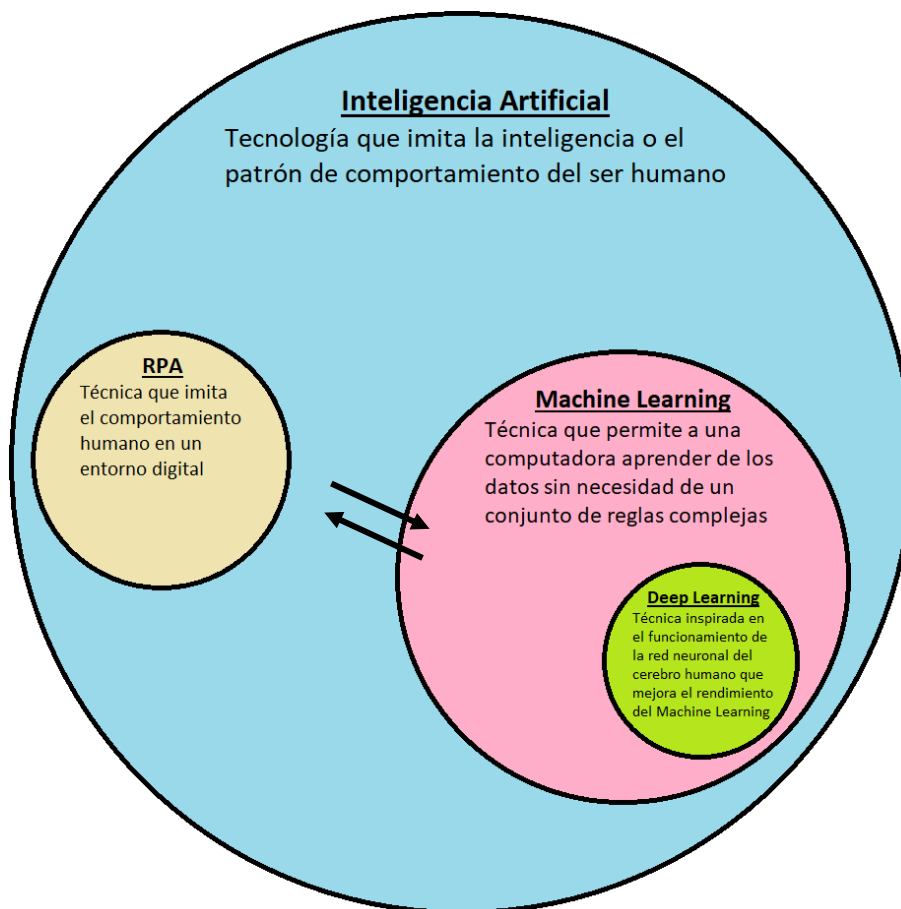


Ilustración 2.5: Relación entre RPA, IA & ML[14]

3 INTELIGENCIA ARTIFICIAL

Las máquinas podrán hacer cualquier cosa que hagan las personas, porque las personas no son más que máquinas.

-Marvin Minsky-

3.1 Breve historia

La inteligencia artificial surge como respuesta ante la necesidad de automatizar procesos propios de la inteligencia humana, y su nacimiento fue posible gracias a los enormes avances que se hicieron en el ámbito de la Computación a lo largo del S.XX. Se puede considerar que el punto de partida de esta área se sitúa en 1950, cuando Alan Turing publica su trabajo “Máquinas Computacionales e Inteligencia” donde se desarrolla el mecanismo del Test de Turing para determinar si una máquina puede ser considerada inteligente.

Un hito importante se dio pocos años después cuando Arthur L. Samuel, ingeniero de IBM, desarrolla un algoritmo basado en el Minimax de Claude Shannon que permite a su programa, un juego de damas, aprender y mejorar en cada partida a partir de los resultados obtenidos en partidas anteriores. Samuel llamó a este mecanismo de aprendizaje **machine-learning**.

En la primera década del S.XXI, y tras un periodo de lenta progresión y desánimo en el área, se ataca uno de los grandes problemas de la IA: la máquina debe comunicarse con el ser humano en su propio lenguaje natural para que se pueda considerar que se comporta de forma inteligente. De este problema nace otra de las grandes ramas de la IA, el **Procesamiento de Lenguaje Natural**.

Otro gran hito que se da durante esta década es la victoria del sistema Watson de IBM en el concurso televisivo “Jeopardy!” contra sus dos mejores concursantes. La dificultad radicaba en que el ordenador tenía que entender la pregunta y buscar la respuesta adecuada en un tiempo reducido. La capacidad del ordenador para aprender a medida que trabajaba, acumular información e interactuar en un lenguaje natural fue clave en el desarrollo y teorización del **Deep Learning**.

No obstante, en estos años, la rama que ha conseguido ganar más protagonismo y en la que se basan casi la totalidad de nuevas técnicas es el **Machine Learning** y sus algoritmos basados en **redes neuronales** artificiales. La perspectiva para la IA en los próximos años sitúa el 2025 como un año clave en el que los avances serán suficientes como para integrar IA con robótica o realidad aumentada.[15], [16]

3.2 Acotación de nuestro caso

En este trabajo estudiaremos la aplicación real de la inteligencia artificial a RPA a través de un ejemplo práctico. Como se explicará más adelante, el ejemplo consistirá en automatizar un proceso de gestión de facturas y presupuestos de una empresa ficticia. El adecuado cumplimiento de esta tarea requiere de un profundo estudio del proceso para analizar y determinar cada acción que lo compone, la posibilidad de automatización y la forma en que esta se llevará a cabo. Este estudio paso a paso del proceso se puede encontrar en la [sección 9.2](#) del Anexo y la solución de automatización propuesta en la [sección 9.3](#).

En cuanto a la aplicación de inteligencia artificial en la automatización del proceso, se implementará con el objetivo de solucionar 2 necesidades:

- a. *Clasificación*: Es necesario reconocer qué tipo de documento estamos tratando, particularmente si se trata de una factura, de un presupuesto o ninguna de las anteriores.
- b. *Reconocimiento de información*: Por otro lado, independientemente del tipo de documento, es necesario extraer determinada información como, por ejemplo, el montante del presupuesto o factura, la fecha de emisión, el nombre del proveedor, etc.

Como hemos visto, el campo de la inteligencia artificial comprende varias ramas. Sin embargo, para tratar este problema nos centraremos en 2 ramas: Deep Learning (más concretamente, redes neuronales convolucionales) y Procesamiento del Lenguaje Natural.

3.2.1 Redes neuronales artificiales

Las redes neuronales artificiales (RNAs) son modelos matemáticos que tratan de emular la estructura neuronal del cerebro humano. Su funcionamiento está basado en el funcionamiento de las redes de neuronas biológicas. Las neuronas del cerebro humano están compuestas de dendritas, soma y axón. Las dendritas se encargan de captar los impulsos nerviosos que emiten otras neuronas. Estos impulsos, se procesan en el soma y se transmiten a través del axón, que emite un impulso nervioso hacia las neuronas contiguas. Como veremos más adelante, el funcionamiento de una RNA es similar: cada neurona recibe una información del exterior o de otra neurona, la procesa en su núcleo y emite una salida a otra neurona o al exterior.

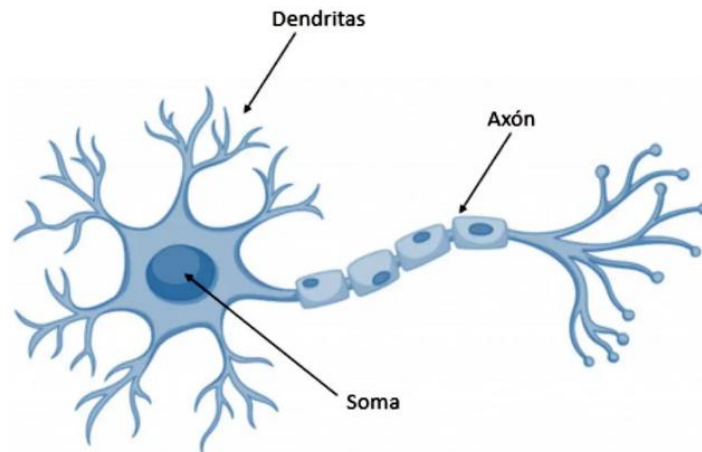


Ilustración 3.1: Representación de una neurona biológica

Las RNA se caracterizan por el aprendizaje a través de la experiencia. Teuvo Kohonen las definió en 1988 de la siguiente manera:

“Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico” (Kohonen, 1988).

Desde un punto de vista formal, una RNA puede definirse como un grafo dirigido con las siguientes propiedades:

1. A cada nodo j se le asocia una variable de estado x_j .
2. A cada conexión (i, j) de los nodos i y j se le asocia un peso $w_{ij} \in \mathbb{R}$.
3. En muchas ocasiones a cada nodo se le asocia un umbral de disparo θ_j .
4. Para cada nodo j se define una función $f_j(x_j, w_{ij}, \theta_j)$, que depende de los pesos de sus conexiones, del umbral y de los estados de los nodos i a él conectados. Esta función proporciona un nuevo estado del nodo.

Además:

- Un nodo se representa mediante una neurona.
- Una conexión se representa mediante una sinapsis.
- Una neurona de entrada es aquella que no tiene conexiones entrantes.
- Una neurona de salida es aquella que no tiene conexiones salientes.
- Las neuronas que no son de entrada o salida son neuronas ocultas.
- Una red es recurrente si el flujo de información puede retroalimentarse.

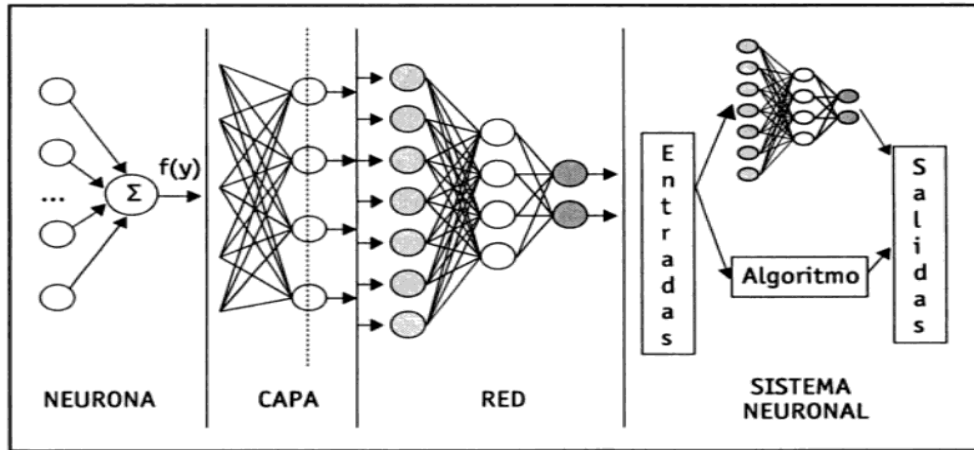


Ilustración 3.2: Estructura jerárquica de un sistema basado en RNA.[17]

3.2.1.1 Funcionamiento general de una RNA

El funcionamiento básico de una RNA consiste en tomar datos del exterior, procesarlos internamente y devolver una salida. Estos pasos se realizan bajo una determinada arquitectura; en redes *feedforward* las neuronas se organizan jerárquicamente formando capas, en redes *recurrentes* las neuronas se conectan arbitrariamente con otras neuronas, ya sean de la misma capa, de distinta o consigo mismas. Una capa es una agrupación de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino. En función de sus entradas y salidas, las capas se clasifican en:

1. **Capa de entrada:** Es la capa que recibe la información del exterior del sistema. En esta capa no se realiza ningún procesamiento.
2. **Capas ocultas:** Son aquellas cuyas entradas y salidas se encuentran dentro del sistema. Reciben información de la capa de entrada o de otras capas ocultas, la procesan y transmiten la salida a la siguiente capa.
3. **Capa de salida:** Es la capa que proporciona la respuesta de la red al exterior.

En función de sus capas, las RNA se distinguen en:

- **Redes monocapa:** Son aquellas redes constituidas con una única capa.
- **Redes multicapa:** Son aquellas redes constituidas con más de una capa.

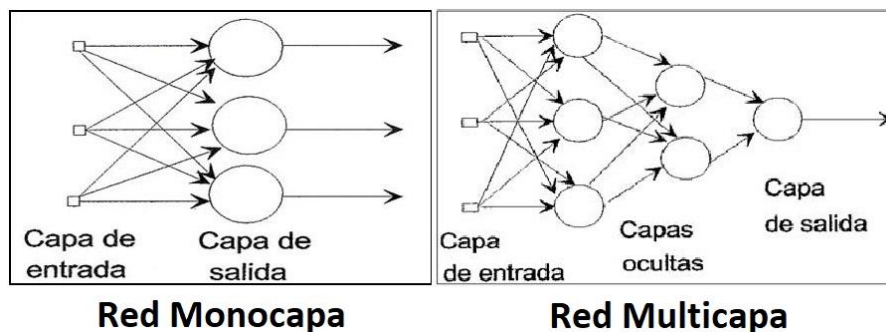


Ilustración 3.3: Esquema general de una red neuronal artificial

3.2.1.2 Procesamiento de los datos

Hemos dicho que la información se procesa en las capas ocultas pero ¿qué tipo de información recibe cada neurona y cómo la procesa? Independientemente del caso al que apliquemos el modelo de RNA, las entradas y salidas de cada neurona de la red son numéricas. Cada neurona realiza 3 operaciones que llamamos *función de entrada* o *regla de propagación*, *función de activación* y *función de salida*. [18]–[21]

La **función de entrada** o regla de propagación es la que proporciona el valor que se va a procesar en la neurona. Este valor es función de las entradas y los pesos. Las más comunes son:

- Sumatorio de entradas ponderadas:

$$\sum_{j=1}^n w_{ji} a_j$$

- Producto de entradas ponderadas:

$$\prod_{j=1}^n w_{ji} a_j$$

- Máximo de entradas ponderadas:

$$\max_j w_{ji} a_j, \quad j = 1, 2, \dots, n$$

La **función de activación** es la que procesa el valor de la regla de propagación. Su objetivo es mantener los valores producidos por cada neurona en un determinado rango, habitualmente entre 0 y 1 o bien entre -1 y 1. Además, impide que la red se comporte como una función lineal. Las más usuales son las siguientes:

Función Bipolar

$$\text{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x \leq 0 \end{cases}$$

Función Umbral

$$\text{umbral}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Función Sigmoide

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Función ReLU

$$\text{RL}(x) = \max(0, x) = \begin{cases} x & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Función SoftMax

Para \underline{X}_k un vector k – dimensional

$$\sigma_{SM}(x_j) = \frac{e^{x_j}}{\sum_{i=1}^k e^{x_i}}, \quad j = 1, \dots, k$$

La **función de salida** es la que transforma el valor de la función de activación en un valor de salida de la neurona. Habitualmente suele ser la función identidad, por lo que el valor de la función de activación suele ser el valor de salida.

Adicionalmente, cada neurona recibe un valor denominado *umbral* o *bias*. El umbral indica a partir de qué valor de la función de entrada la neurona produce una salida significativa. Por tanto, el umbral determina si la neurona se activa o no.

Así pues, el cálculo que realiza cada neurona es

$$a_i = f \left(h(w_{ji}, a_j) \right)$$

Donde:

- a_i = Salida proporcionada por la neurona i
- w_{ji} = Peso de la conexión entra las neuronas j, i .
- f = Función de activación
- $h(w_{ji}, a_j)$ = Función de entrada. Normalmente suele ser $\sum_{j=1}^n w_{ji} a_j$, el sumatorio del producto de las salidas y los pesos de todas las neuronas j que envían su salida a la neurona i .
- Para $j = 0$ se considera una entrada ficticia $a_0 = -1$ y un peso w_{0i} = umbral.

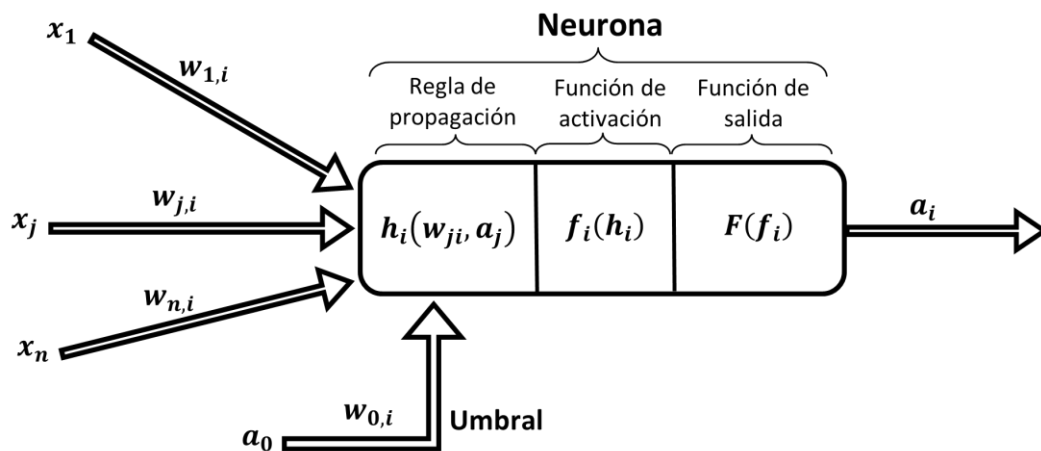


Ilustración 3.4: Esquema de una neurona artificial.

3.2.1.3 Fases de procesamiento de datos en una RNA: Entrenamiento y Ejecución

El **entrenamiento** de una red consiste en ajustar los pesos w_{ji} en función de la información de entrada. En esencia, durante el entrenamiento pueden ocurrir 3 tipos de cambios: destrucción de una conexión ($w_{ji} \neq 0 \rightarrow w_{ji} = 0$), modificación de una conexión ($w_{ji} \neq 0 \rightarrow w_{ji} \neq 0$) o creación de una conexión ($w_{ji} = 0 \rightarrow w_{ji} \neq 0$). La finalidad de entrenar una RNA es determinar un conjunto de pesos W que permitan a la red realizar con el mayor éxito posible el tipo de procesamiento requerido.

En la fase de entrenamiento de una RNA se distinguen 3 conceptos:

- *Paradigma de aprendizaje*: Es la información de la que dispone la red, en la que se podrá basar para construir los modelos.
- *Regla de aprendizaje*: Es el principio que gobierna el aprendizaje. Habitualmente consiste en la optimización de una función de error, que mide la eficacia de la red.
- *Algoritmo de aprendizaje*: Es el procedimiento numérico de ajuste de los pesos.

Basándonos en el **paradigma de aprendizaje**, podemos diferenciar 4 tipos: [22]

- *Supervisado*: Este tipo de aprendizaje se caracteriza porque está controlado por un agente externo que determina las respuestas que la red debería generar a partir los datos introducidos. Se confía en que las respuestas conocidas, las que el agente externo determina, son las correctas. Si la respuesta calculada no coincide con la respuesta determinada, la real, entonces se modifican los pesos de las conexiones. Este ajuste se realiza iterativamente hasta que la salida tiende a ser la deseada.
- *No supervisado*: Este tipo de aprendizaje consiste en introducir a la red un conjunto de datos de entrada sin las respuestas deseadas, por lo que la red debe agrupar patrones similares. Este tipo de redes tratan de estimar la función densidad de probabilidad $p(x)$ que describe la distribución de patrones, permitiendo así extraer rasgos o agrupar patrones según su similitud.
- *Híbrido*: La red contiene los dos tipos de aprendizaje descritos anteriormente en distintas capas.
- *Por refuerzo*: Se basa en no indicar durante el entrenamiento la salida exacta que se desea, sino en ofrecer una señal de refuerzo si la salida calculada, que se obtiene a partir de un motor que simule el comportamiento del sistema, se ajusta a la deseada (éxito = +1 o fracaso = -1), o sea, se indica lo bien o mal que lo está haciendo. A partir de esto los pesos se ajustan según un mecanismo de probabilidades.

Basándonos en el **algoritmo de aprendizaje** se distinguen 4 tipos:

- *Minimización del error*: Consiste en ajustar los pesos en función de la diferencia entre valores deseados y calculados, siendo

$$\Delta w_{ji} = \beta a_i (d_j - a_j)$$

Un ejemplo de algoritmos que usan este tipo de aprendizaje es la **regla Delta**, el **error mínimo cuadrado** o la **regla Delta generalizada**.

- *Estocástico* o de *Boltzmann*: Consiste en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.
- *Hebbiano*: Consiste en el ajuste de los pesos de las conexiones de acuerdo con la correlación de los valores de activación de las dos neuronas conectadas. Si ambas neuronas son activas la conexión se refuerza, en caso contrario la conexión se debilita.

- **Competitivo:** Consiste en activar solo un conjunto de neuronas, aquellas con el valor más cercano a la salida deseada. Por tanto, las neuronas compiten por activarse. El objetivo de este algoritmo es categorizar los datos de entrada. Así, las informaciones similares son clasificadas formando parte de la misma categoría y activando, por tanto, la misma neurona de salida.

Hay que tener en cuenta el problema del **sobreajuste** de la red al conjunto de datos de entrada. La red puede aprender *demasiado* sobre las características de los datos proporcionados y ofrecer resultados excelentes sobre ese mismo conjunto de datos, pero ofreciendo poca variabilidad para ajustarse a otro conjunto de datos desconocidos. Para evitar este problema, es aconsejable utilizar un conjunto de datos distinto al **conjunto de entrenamiento** que permita controlar el proceso de aprendizaje. Este conjunto se denomina **conjunto de validación**. El conjunto de validación se extrae como subconjunto del conjunto de entrenamiento, ya separado previamente del conjunto de prueba. Durante el aprendizaje la red va ajustando los pesos según los datos de entrenamiento y los evalúa frente al conjunto de validación para determinar el error que está cometiendo.

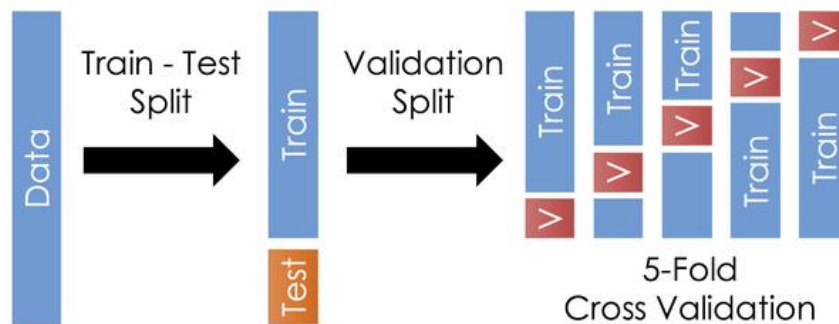


Ilustración 3.5: Ejemplo de separación de datos en entrenamiento, validación y test

La fase de entrenamiento se da por concluida cuando los pesos alcanzan valores estables, la variación es despreciable o se alcanza algún criterio de parada impuesto.

Una vez concluida la fase de entrenamiento, los pesos y la estructura de la red se fijan para evitar que sigan cambiando y se pasa a la fase de **ejecución**, donde usamos la red ya entrenada. Así, en esta fase se introducen en la red valores cuya respuesta se desconoce, que son procesados en función de los pesos y la estructura para obtener una salida: la *predicción* de la respuesta calculada por la red.

3.2.2 Redes neuronales convolucionales

Las redes neuronales convolucionales (CNNs por sus siglas en inglés) son un tipo de redes neuronales artificiales multicapa que esperan, explícitamente, imágenes como datos de entrada. Las CNN procesan las imágenes a través de una serie de filtros consecutivos, conformados por sus capas ocultas, para extraer patrones o características concretas de la imagen. Por ejemplo, una capa se encarga de reconocer líneas rectas, otras capas se encargan de reconocer tonos azules, etc. Es decir, cada capa va aprendiendo diferentes niveles de abstracción. Otro rasgo distintivo de este tipo de redes es que, a diferencia de las RNA, cada neurona de una capa no está conectada a todas las neuronas de la siguiente capa, sino solo a un grupo de ellas.[23]

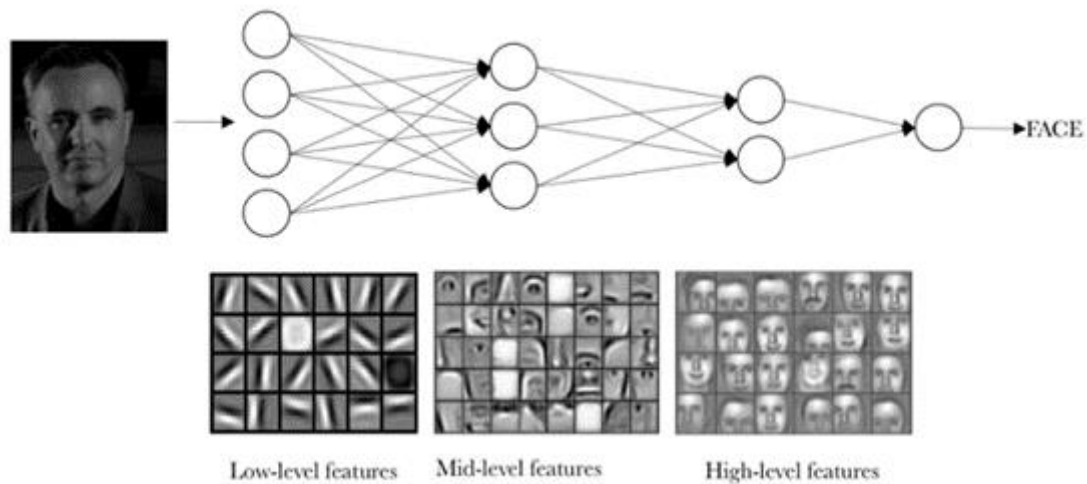


Ilustración 3.6: Capas de una CNN

3.2.2.1 Arquitectura de una CNN

La extracción de características de una imagen se realiza a través de una arquitectura basada en 2 tipos de capas alternadas: *convolutional layers* y *pooling layers*. Una vez extraída la información, esta se procesa a través de una serie de capas denominadas *fully connected layers* que finalizan con una capa de salida de tipo *softmax* para realizar la clasificación.

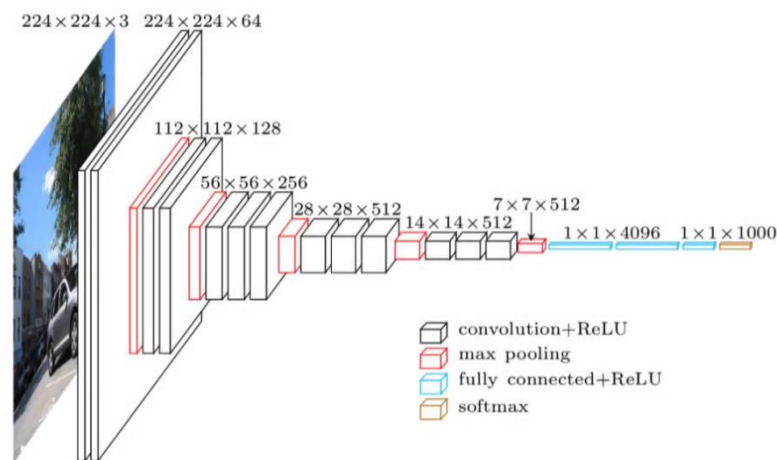


Ilustración 3.7: Arquitectura de una CNN basada en el modelo VGG16 [24]

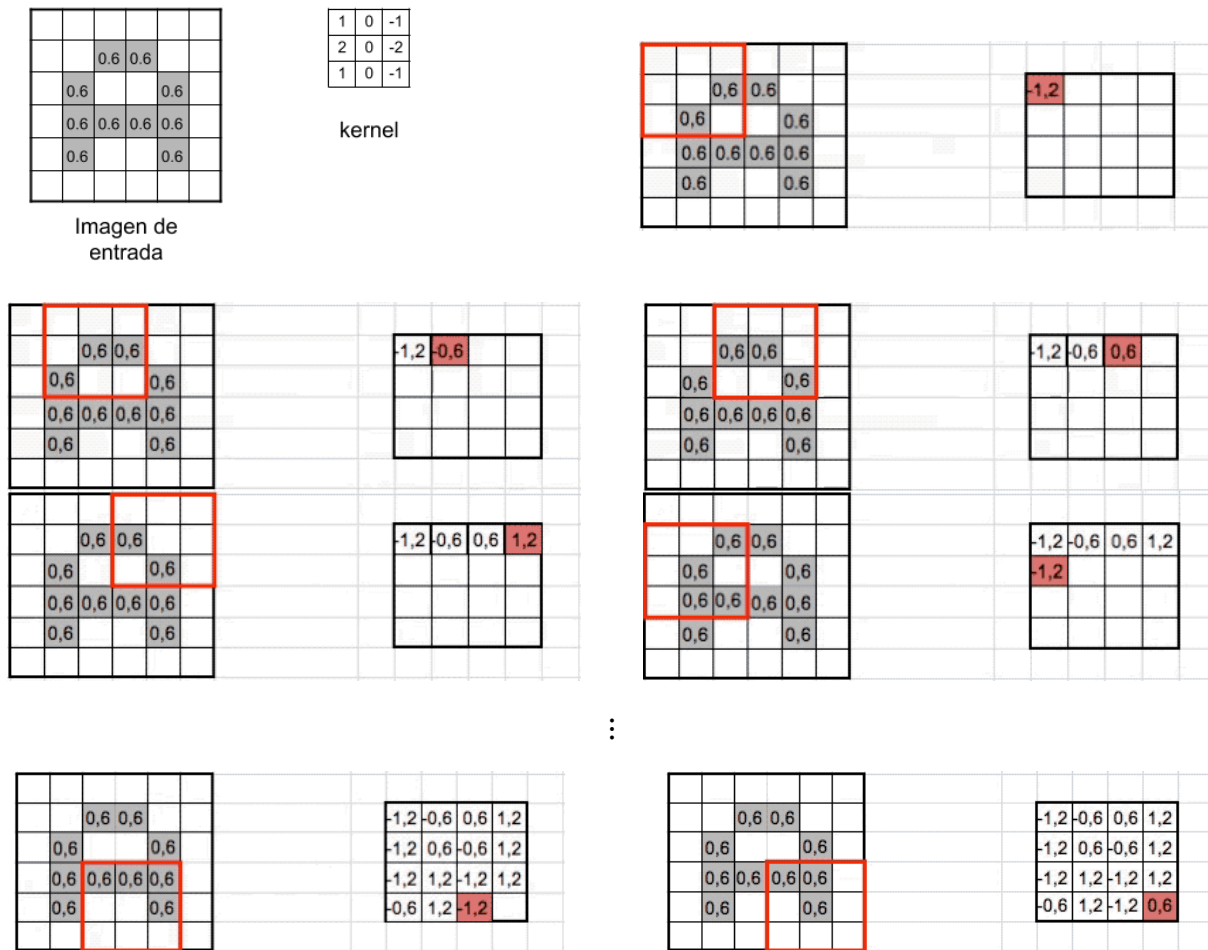
3.2.2.2 Convolutional layer

La función principal de las capas de convolución es detectar características o patrones en las imágenes. Este tipo de capas presentan dos características extremadamente interesantes: por un lado, una vez que aprenden una característica (como, por ejemplo, una línea vertical) en algún punto de la imagen, pueden reconocerla más tarde en cualquier otro punto; por otro lado, se preservan las relaciones espaciales entre capas. Esto es, si una capa aprende a detectar algún tipo de elemento básico, como líneas verticales, las capas posteriores pueden aprender a detectar elementos más complejos compuestos por los elementos básicos de las capas anteriores hasta llegar a patrones complejos.

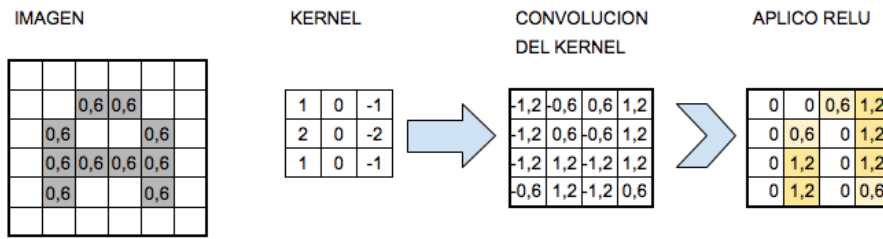
Pero, cuando decimos que la red procesa una imagen y extrae características, ¿a qué nos estamos refiriendo exactamente? Para un ordenador una imagen no es más que un tensor tridimensional con dos *ejes espaciales* de dimensión 1 y un *eje de canal* de dimensión 3 para imágenes a color (red, blue, green) o de dimensión 1 para imágenes en escala de grises. Por tanto, el “procesamiento” de la imagen es, en realidad, una serie de operaciones matriciales, concretamente operaciones de **convolución**.

La convolución consiste en una serie de productos y sumas entre una matriz constante, real y simétrica denominada *kernel* y una submatriz *M*, ventana de la matriz original de la imagen, con la misma dimensión que el kernel. Si desplazamos *M* por toda la matriz original, con un salto (o stride) de *k* columnas y filas, de izquierda a derecha y de arriba hacia abajo, estaremos generando una nueva matriz “filtrada”, que será la salida que servirá de entrada para la próxima capa.

Supongamos que tenemos una imagen de 6×6 píxeles en blanco y negro. Esto se traduce como un tensor $6 \times 6 \times 1$, lo que suponen $6 \times 6 = 36$ neuronas en la capa de entrada. Por otro lado, nuestro kernel es de dimensión 3×3 , por lo que cada neurona de esta capa está conectada con $3 \times 3 = 9$ neuronas de la siguiente capa. Entonces, aplicamos y desplazamos el kernel por cada neurona de entrada (stride = 1) para obtener como salida una matriz, a la que se le aplicará la función de activación oportuna (habitualmente ReLu) y que será la matriz de entrada de la siguiente capa. Visualmente se podría ejemplificar así. [25]



O sea,



Nótese que, al aplicar el kernel, siempre obtenemos una matriz de dimensiones menores a la matriz original. Si quisiéramos evitar esta reducción de dimensión, usaríamos una técnica denominada *padding*, que consiste en rodear la matriz original con tantas filas y columnas de 0 como fueran necesarias.[23], [25]

Es importante destacar que en una misma capa convolucional se pueden aplicar varios kernels simultáneamente. Un conjunto de kernels aplicados a una misma capa se conoce como *filtro*. Al aplicar un filtro obtenemos como salida un tensor de dimensiones $m \times n \times \#kernels$. Suponiendo que tuviéramos una imagen de 28×28 píxeles en escala de grises a la que aplicaríamos 32 kernels 5×5 , obtendríamos como salida 32 matrices 24×24 , por lo que necesitaríamos $24 \times 24 \times 32 = 18432$ neuronas en la siguiente capa. Por tanto, es necesario una operación para reducir la dimensión y los tiempos de procesamiento.

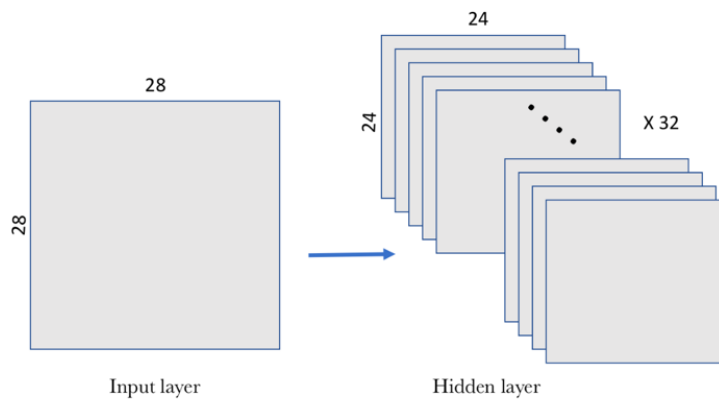


Ilustración 3.8: Ejemplo de capa con 32 filtros. [23]

3.2.2.3 Pooling Layer

La reducción de la dimensión se hace a través de las capas de pooling, que habitualmente se aplican justo tras una capa convolucional. Estas capas hacen una simplificación de la información extraída por la capa convolucional a través de un método de submuestreo, sin embargo, mantienen las características más relevantes detectadas por cada filtro. El método de submuestreo más utilizado por sus buenos resultados es el **max-pooling**.

Al igual que hacíamos para aplicar el kernel, el método max-pooling consiste en recorrer cada una de las k matrices de dimensión m de entrada con una matriz V de dimensión p y con $stride = p$, preservando el valor más alto dentro de nuestra matriz V . Al final obtenemos k matrices de dimensión $\frac{m}{p}$ que contiene los n valores más altos (los más relevantes) de las matrices de entrada.

En el ejemplo de un tensor $28 \times 28 \times 1$ al que aplicamos una convolución con 32 filtros de kernel 5×5 y un max-pooling 2×2 , habríamos reducido la cantidad de neuronas necesarias para procesar la información en la siguiente capa desde $24 \times 24 \times 32 = 18432$ hasta $12 \times 12 \times 32 = 4608$.

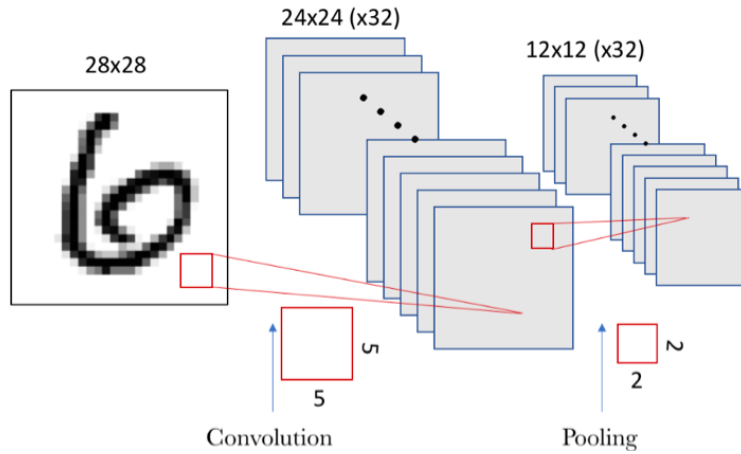


Ilustración 3.9: Submuestreo por max-pooling de dimensión 2. [23]

Este proceso de convolución y submuestreo se podría considerar como un ciclo. En un primer ciclo se detectan características básicas como rectas o curvas pero, como ya dijimos, cuantos más ciclos se realicen mayor será la complejidad de las características que se pueden extraer. Sin embargo, este proceso no es infinito. Al realizar submuestreos, las matrices de entrada de los ciclos posteriores serán cada vez de dimensión menor.

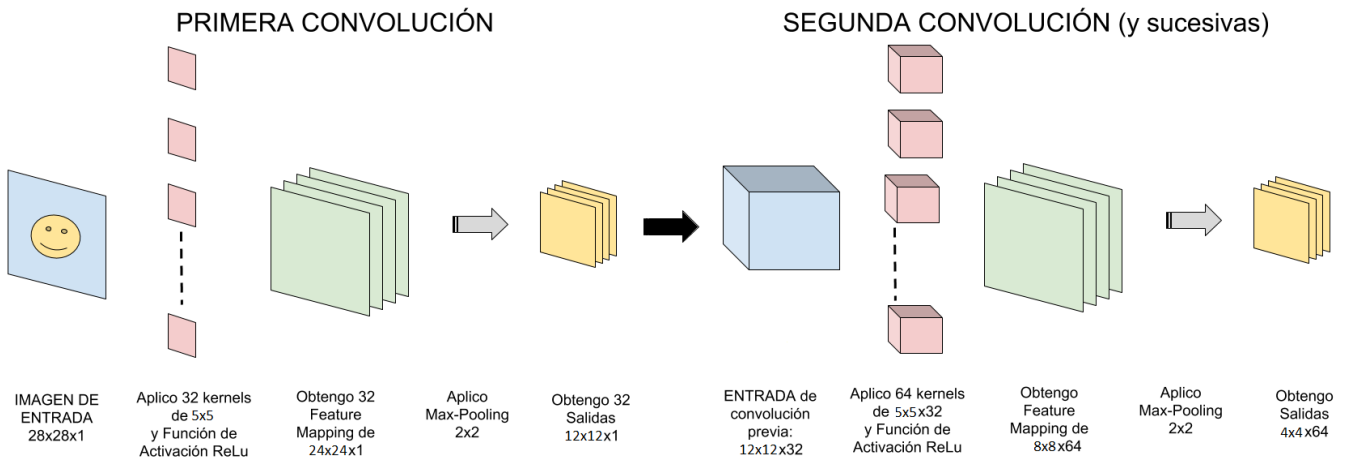


Ilustración 3.10: Ciclos de convoluciones en una CNN. [25]

En resumidas cuentas, en nuestro ejemplo solo podríamos hacer 2 ciclos.

Ciclo	Entrada	Kernel	Obtengo	Max-pooling	Salida del ciclo
1	$28 \times 28 \times 1$	$\Rightarrow 5 \times 5 \times 32$	$\Rightarrow 24 \times 24 \times 32$	$\Rightarrow 2 \times 2 \times 32$	$\Rightarrow 12 \times 12 \times 32$
2	$12 \times 12 \times 32$	$\Rightarrow 5 \times 5 \times 64$	$\Rightarrow 8 \times 8 \times 64$	$\Rightarrow 2 \times 2 \times 32$	$\Rightarrow 4 \times 4 \times 64$

En general, el número de ciclos depende del número de píxeles originales, del tamaño de los kernels, del padding y del stride. Si, por ejemplo, nuestra imagen de entrada hubiera tenido más píxeles, o nuestro kernel tuviera una dimensión menor, se podrían hacer más ciclos de convoluciones. Una vez finalizada esta etapa, procesamos los datos de salida con la siguiente etapa.

3.2.2.4 Fully Connected Layers

Este conjunto de capas forma una RNA, habitualmente un perceptrón multicapa. En este momento tenemos un tensor “tridimensional” con k matrices de dimensión $m \times n$. Para poder procesarlo en el perceptrón tenemos que reducir su dimensionalidad a 1. Este proceso se denomina “aplanamiento”. Para aplanar un tensor solo debemos disponer sus columnas o sus filas, de forma ordenada, en un vector columna.

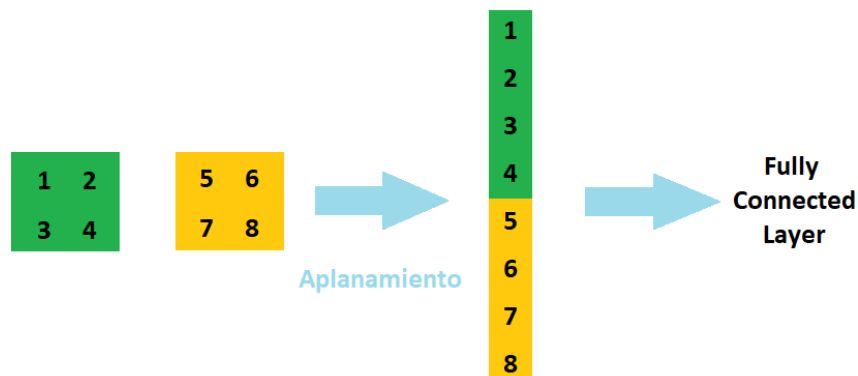


Ilustración 3.11: Proceso de aplanamiento

Una vez aplanado, conectamos con la primera capa oculta del perceptrón multicapa, que tendrá $m \times n \times k$ neuronas, y procesamos la información reduciendo su dimensionalidad capa tras capa hasta tener una capa de salida con s neuronas, una por cada categoría de nuestra clasificación. Habitualmente, en esta fase, se usa como función de activación la función Softmax porque es la que suele ofrecer mejores resultados así como una salida más intuitiva, ya que su salida se puede utilizar para representar una distribución de probabilidad sobre las s posibles categorías de clasificación.

Al final de esta fase obtendremos como salida el resultado de la clasificación calculada por la red convolucional.

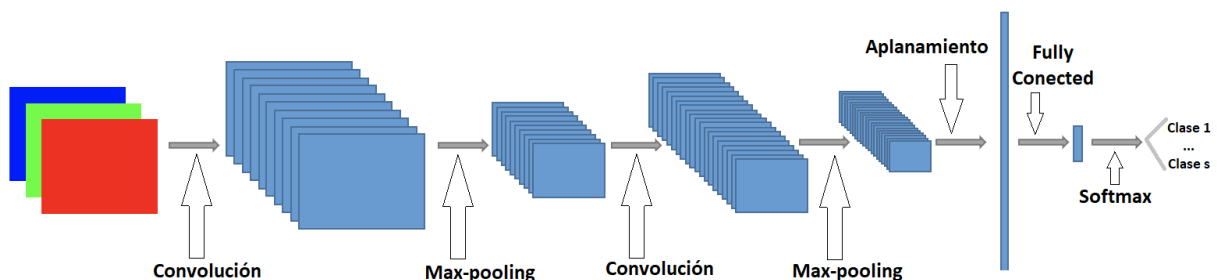


Ilustración 3.12: Proceso completo de una CNN

Para finalizar, diremos que las CNN tienen un proceso de aprendizaje similar al de una RNA con la ventaja de que la cantidad de pesos que se deben ir actualizando en el entrenamiento es considerablemente menor. Esto se debe a que los parámetros que se deben ajustar son los valores de los kernels, que tienen tamaños muy reducidos. Por ejemplo, supongamos una capa de 784 neuronas conectadas con otra capa de 4608 neuronas. En una RNA, todas las neuronas de una capa están conectadas con todas las neuronas de la siguiente capa, por lo que habría que actualizar $784 \times 4608 = 3612672$ pesos. En el caso de la CNN, si aplicamos 32 filtros con kernel 3×3 , solo se tendrían que actualizar $3 \times 3 \times 32 = 288$ pesos.

3.2.3 Procesamiento del lenguaje natural

Un lenguaje formal es aquel que está predefinido por reglas bien establecidas y que se rigen rigurosamente por estas como, por ejemplo, un lenguaje de programación computacional. Por el contrario, el lenguaje natural es un lenguaje que evoluciona constante y fluidamente para adaptarse a las necesidades comunicativas de los seres humanos o, expresado de otra forma, es el lenguaje que usamos las personas para comunicarnos cotidianamente. El lenguaje natural se caracteriza por las siguientes propiedades:[26]

- Se define a partir de una gramática G , sin embargo, como evoluciona y se enriquece progresivamente, también modifica la gramática que la define. Esto dificulta la formalización de la definición de G .
- Tiene un gran poder expresivo debido a la riqueza del componente semántico. Contiene sutilezas, figuras como la ironía o mensajes subyacentes sobreentendidos que dificultan aún más la formalización de G .

El procesamiento del lenguaje natural consiste en el uso del lenguaje natural para comunicarnos efectivamente con el ordenador, facilitar el desarrollo de software que realice tareas relacionadas con el lenguaje o desarrollar modelos que ayuden a comprender los mecanismos relacionados con el lenguaje.

3.2.3.1 Arquitectura general de un sistema de PLN

La arquitectura de un sistema de PLN se basa en la definición del lenguaje natural por niveles:

- *Nivel fonológico*: Trata de cómo las palabras se relacionan con los sonidos que representan.
- *Nivel Morfológico*: Trata de cómo las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas.
- *Nivel Sintáctico*: Trata de cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega en la oración y qué sintagmas son parte de otros sintagmas.

- *Nivel Semántico*: Trata del significado de las palabras y de cómo los significados se unen para dar sentido a una oración. También se refiere al significado independiente del contexto, es decir de la oración aislada.
- *Nivel Pragmático*: Trata de cómo las oraciones se usan en distintas situaciones y de cómo el uso afecta a la semántica de las oraciones. Se reconoce un subnivel recursivo-discursivo que trata de cómo el significado de una oración se ve afectado por las oraciones inmediatamente anteriores.

El funcionamiento es el siguiente:

1. EL usuario introduce en el ordenador una expresión.
2. El ordenador analiza las oraciones morfológica y sintácticamente para verificar la correcta estructura de las oraciones. Este análisis depende de un analizador lexicográfico y sintáctico denominados scanner y parser respectivamente.
3. El ordenador analiza las oraciones sintácticamente para conocer el significado de cada oración y asignarlo a expresiones lógicas de tipo verdadero/falso.
4. El ordenador realiza el análisis pragmático de la expresión teniendo en cuenta el orden de las oraciones.
5. De forma general, y dependiendo del objetivo que se pretenda conseguir con el sistema, a partir de los pasos anteriores el ordenador obtiene una expresión final, la ejecuta y devuelve el resultado al usuario.

3.2.3.2 Recuperación y extracción de información

Los campos de desarrollo y aplicación de PLN son muchos y variados. Recordemos que el problema que pretendemos resolver es la automatización de un proceso de gestión de facturas y presupuestos. Para ello será necesaria la extracción de información de los documentos que queremos procesar, por tanto, en esta sección nos centraremos en la recuperación y extracción de información de documentos.

Recuperación de información

La *recuperación de información* es el proceso de encontrar, en grandes bases de datos, documentos no estructurados o semiestructurados, relevantes y pertinentes que satisfagan una necesidad de información. Con datos no estructurados nos referimos a datos que no siguen un esquema claro y, por tanto, no están listos para procesar. Los datos semiestructurados se refieren a datos que se encuentran en documentos con marcas explícitas como, por ejemplo, HTML, XML o Markdown. Las estrategias de recuperación de información se desarrollan según modelos de acuerdo con dos posibles dimensiones: bases matemáticas y propiedades.[27]

En la dimensión de bases matemáticas, el texto es representado como conjuntos de palabras donde las coincidencias se logran en función de operaciones algebraicas booleanas, lo que lleva a modelos algebraicos en espacios vectoriales o matriciales, modelos probabilísticos o modelos basados en características a través de métodos de machine learning.

Por otro lado, la dimensión de propiedades está más relacionada con la forma en que se presentan las relaciones entre los términos del documento en el espacio vectorial: ortogonales e independientes o dependientes.

Extracción de información

Una vez obtenidos los documentos pasamos a la *extracción de información*. La extracción de información es el proceso de obtener las partes que nos interesan del texto para convertirla en información estructurada. Cada documento se procesa para encontrar entidades y relaciones con significado y contenido, definiendo antes los tipos de información semántica que van a ser extraídos. Las salidas que ofrecen los motores de extracción de información son un conjunto de marcos etiquetados cuyos campos se recogen en una tabla.

Hay cuatro tipos de elementos básicos que pueden ser extraídos:

- Entidades: elementos con significado propio.
- Atributos: características de las entidades
- Hechos: relacionan entidades
- Eventos: actividades u ocurrencias en las que participan las entidades

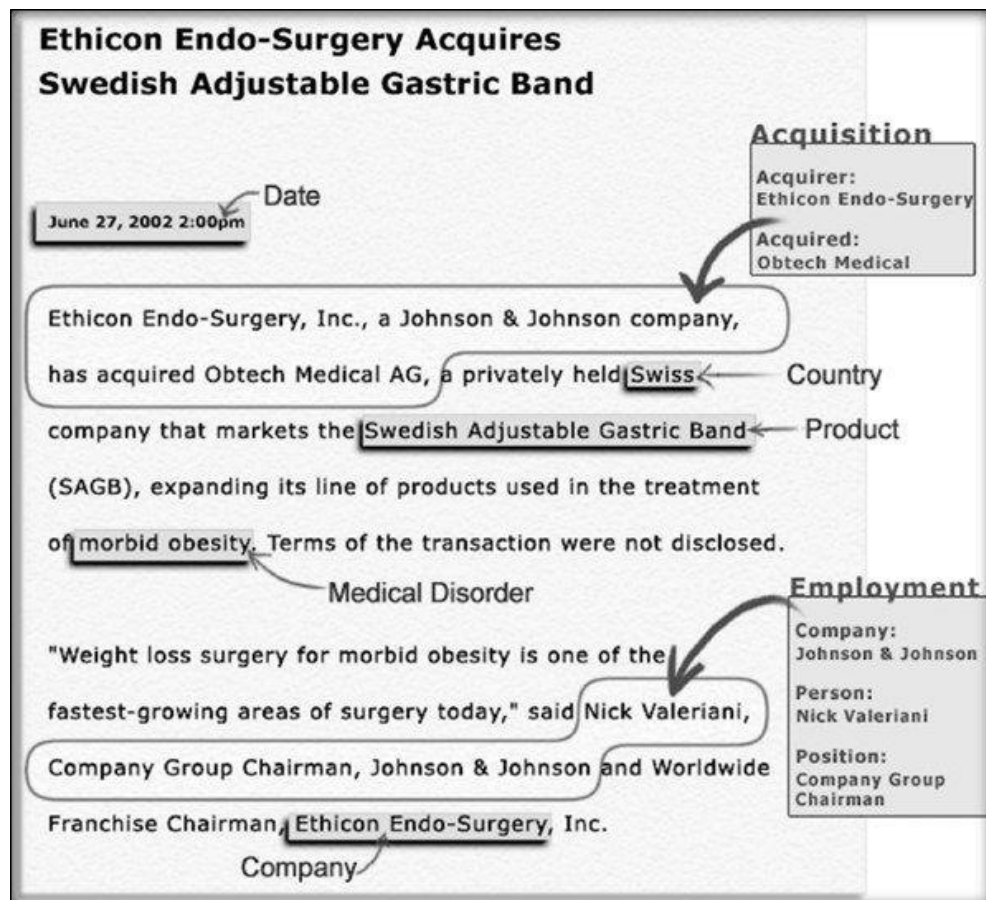


Ilustración 3.13: Ejemplo de documento etiquetado con extracción de información. [27]

3.2.4 Reconocimiento Óptico de Caracteres (OCR)

Para poder extraer información del texto de una imagen, el ordenador debe saber qué texto es el que contiene dicha imagen. Aquí es donde entra en juego el **reconocimiento óptico de caracteres** (OCR por sus siglas en inglés). Este proceso convierte una imagen de texto en un formato de texto que pueden leer las máquinas.

3.2.4.1 Funcionamiento general de un motor de OCR

Un motor de OCR funciona mediante los siguientes pasos:[28]

1. Analiza la imagen y clasifica las áreas claras como fondo y las oscuras como texto.
2. Preprocesa la imagen eliminando manchas, ruido, u otros problemas para mejorar su lectura y reconocimiento.
3. Segmenta la imagen para distinguir cada carácter de otros elementos que no interesan.
4. Reconoce el texto a través de dos posibles algoritmos: coincidencia de patrones o extracción de características.
5. Procesa el texto reconocido para transformarlo en texto procesable en un computador.

Preprocesamiento

En esta fase se pretende dejar la imagen libre de impurezas eliminando el ruido o cualquier imperfección que no sea un carácter. Para eliminar estas impurezas se usan distintos algoritmos:

- *Etiquetado*: para la división de la imagen en regiones de componentes conectadas.
- *Erosión – Expansión*: para la eliminación de pequeños grupos de píxeles.
- *Umbralizado de histograma*: para la eliminación de los objetos más brillantes u oscuros de la imagen.

Además, se estandariza la imagen a través de una binarización: las zonas con texto pasan a valer 1 y las zonas de fondo 0.

Segmentación

Una vez preprocesada, hay que segmentar las partes de la imagen donde los píxeles son adyacentes entre sí. Esta fase es fundamental, ya que permite el posterior reconocimiento de cada carácter individualmente. Además, durante esta fase es necesario reconocer la localización del texto en el documento para poder mantener la estructura renglón-palabra-letra en el mismo orden en el que están en la imagen original. La segmentación se realiza en el siguiente orden:

1. Primero se detectan las líneas (renglones) mediante unos cálculos matriciales respecto a los píxeles no nulos.
2. Luego se aíslan los caracteres a través de sumas de las componentes de la matriz anterior. Si la suma es 0 significa que hay un hueco blanco que representa la separación entre un carácter y otro. Los huecos de separación entre palabras se detectan porque son más anchos que el hueco de separación entre letras de una palabra.

Extracción de características

Una vez realizada la segmentación, se tiene una imagen normalizada en la que se encuentra la información que queremos extraer. La extracción de las características es una de las fases más difíciles en los sistemas de reconocimiento de caracteres, puesto que es difícil escoger un conjunto de características que sea óptimo. Para conocer un conjunto de características suficientemente bueno como para extraer información se recurre a algunas técnicas estadísticas basadas en transformaciones del espacio de representación muestral como, por ejemplo, el análisis de componentes principales o el análisis discriminante.

Una vez tenemos el conjunto de características que necesitamos (por ejemplo líneas, circuitos cerrados o intersecciones) se aplican otro tipo de técnicas estadísticas como KNN, Random Forest o redes neuronales para clasificar cada carácter.

The image shows an invoice from Zylker Dezigns. The text is segmented with red bounding boxes. The items are:

- 1: INVOICE 0.997
- 2: Zylker Dezigns 0.960
- 3: #INV-001654 0.992
- 4: 4606 Benson Park Drive 0.973
- 5: Balance Due 0.971
- 6: \$928.80
- 7: Balance Due 0.971
- 8: Rosedale, 0.993
- 9: \$928.80 0.998
- 10: Oklahoma 73086 0.991
- 11: U. S. A. 0.993
- 12: Bill To 0.990
- 13: Thelma O. Park 0.960
- 14: Invoice Date : 0.948
- 15: 13 July 2021 0.976
- 16: Terms: 0.997
- 17: 3781 Olen Thomas Drive 0.970
- 18: Net14 1.000
- 19: Due Date: 0.940
- 20: Wichita Falls, 0.990
- 21: 27 July 2021 0.995
- 22: TX 76301 1.000
- 23: Project Name : 0.994
- 24: Design Project 0.989
- 25: U. S. A. 0.996

Ilustración 3.14: Ejemplo de salida del motor PaddleOCR

La imagen anterior muestra la salida del motor PaddleOCR. Por una parte vemos cómo es capaz de reconocer la posición, dentro de la imagen, de cada conjunto de palabras y nos lo muestra encuadrándolos en un marco rojo. Como dijimos, primero se detectan los renglones, por este motivo las agrupaciones de palabras reconocidas son, como máximo, de una línea. Por otro lado, tenemos la lista de agrupaciones de palabras encontradas con el siguiente formato: orden de detección, conjunto encontrado, grado de confianza expresado en un rango de 0 a 1 (mayor confianza cuanto más cercano a 1).

4 PRESENTACIÓN DEL CASO DE ESTUDIO

El tiempo es nuestro activo más valioso
y, sin embargo, tendemos a
desperdiciarlo, matarlo y gastarlo en
lugar de invertirlo

-Jim Rohn-

4.1 Introducción

El objetivo de este trabajo es comparar y estudiar los resultados que se obtienen al automatizar un proceso con RPA pero sin IA y al hiperautomatizarlo. Para ello vamos a crear un caso ficticio de una empresa ficticia que solicita nuestro servicio de automatización y lo abordaremos de la forma más realista posible. Una vez planteado se propondrán y estudiarán dos posibles soluciones al caso:

- La primera solución empleará RPA atendida.
- La segunda solución empleará hiperautomatización.

Finalmente se extraerán conclusiones sobre la eficiencia de ambas soluciones desde un punto de vista económico a través de factores como el ahorro de costes o la reducción del tiempo de proceso.

4.1.1 Software empleado

Existen multitud de programas para automatizar, siendo habitualmente los más destacados UiPath, Blue Prism y Automation Anywhere. En un principio, para la realización de este estudio se consideró Microsoft Power Automate, la nueva herramienta de automatización lanzada por Microsoft que ha sido propuesta por Gartner como competidor directo de las herramientas antes mencionadas, en el cuadrante de empresas líderes en RPA [29]. El motivo era la extraordinaria integración que tiene esta herramienta con la suite office, el resto de apps de Microsoft y con Windows. Sin embargo, tras algunas pruebas, el software finalmente usado será Blue Prism por su versatilidad y su diseño visual e intuitivo, además de la potencia de automatización que ha demostrado tener.

Además, para la gestión de facturas usaremos Contoso Invoicing, un programa demo creado para el curso de Microsoft Power Automate que servirá a nuestro propósito.

Por último, usaremos Outlook para la gestión de emails, Excel para la elaboración de reportes y Sharepoint para la distribución de documentos.

Mostraremos a continuación algunas capturas de Blue Prism y de Contoso Invoicing para ilustrar, brevemente, las herramientas fundamentales con las que vamos a trabajar. Para más detalles, tanto de la automatización del proceso como del uso y funcionamiento de Blue Prism, revisar el [Anexo](#).

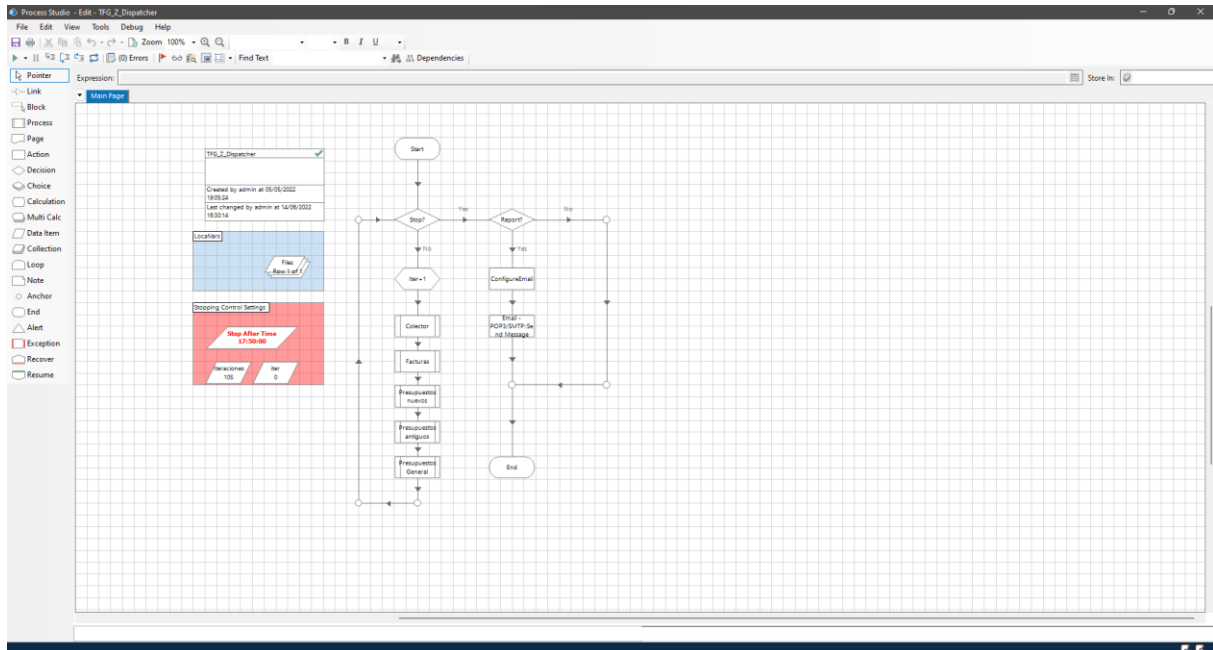


Ilustración 4.1: Ventana de creación de procesos de Blue Prism

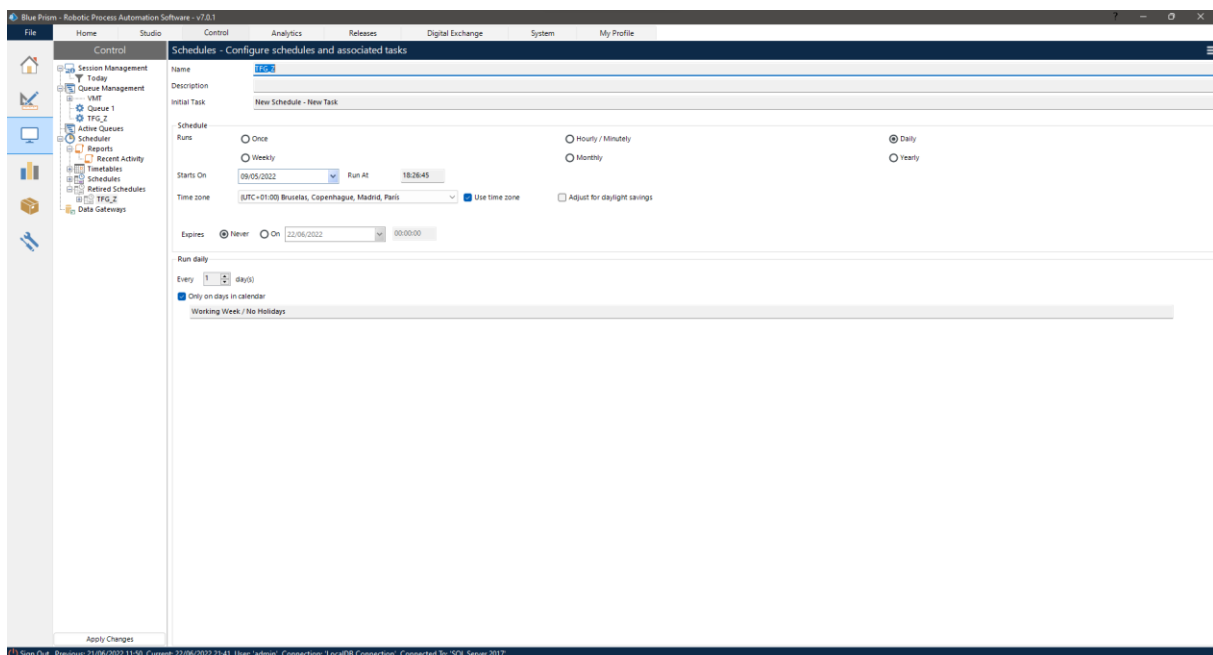


Ilustración 4.2: Ventana de temporizador de ejecución de procesos de Blue Prism

Las capturas anteriores corresponden al modo *Studio* y *Control* de Blue Prism respectivamente. En el modo *Studio* podemos desarrollar la automatización de un proceso como un flujo de trabajo. Cada componente del flujo es una acción, un cálculo o un subproceso completo. En el modo *Control* se pueden gestionar los robots creados, configurando su tiempo y horario de ejecución o las bases de datos internas a las que puede acceder.

Por otro lado, como habíamos mencionado anteriormente, Contoso Invoicing es un programa demo. A simple vista parece contener muchas opciones y configuraciones pero ninguna es realmente funcional salvo los apartados *Invoices*, donde se registran las facturas, y *Accounts*, donde se registran los nuevos proveedores.

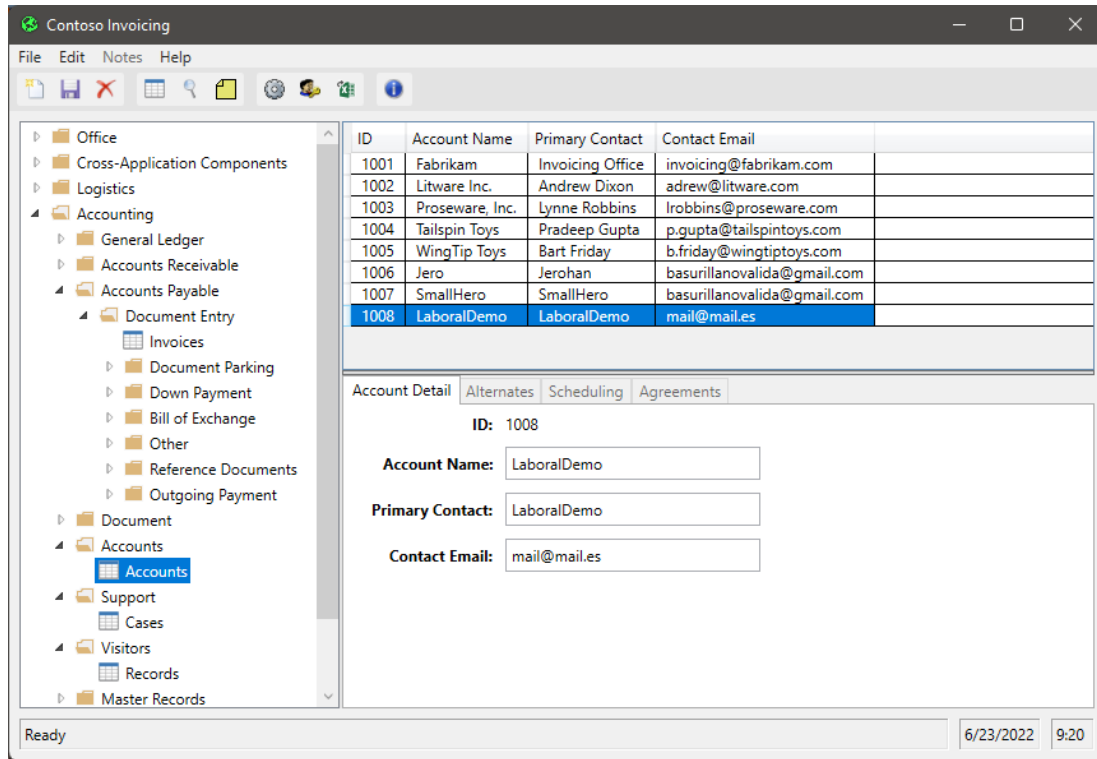


Ilustración 4.3: Ventana de gestión de proveedores de Contoso Invoicing

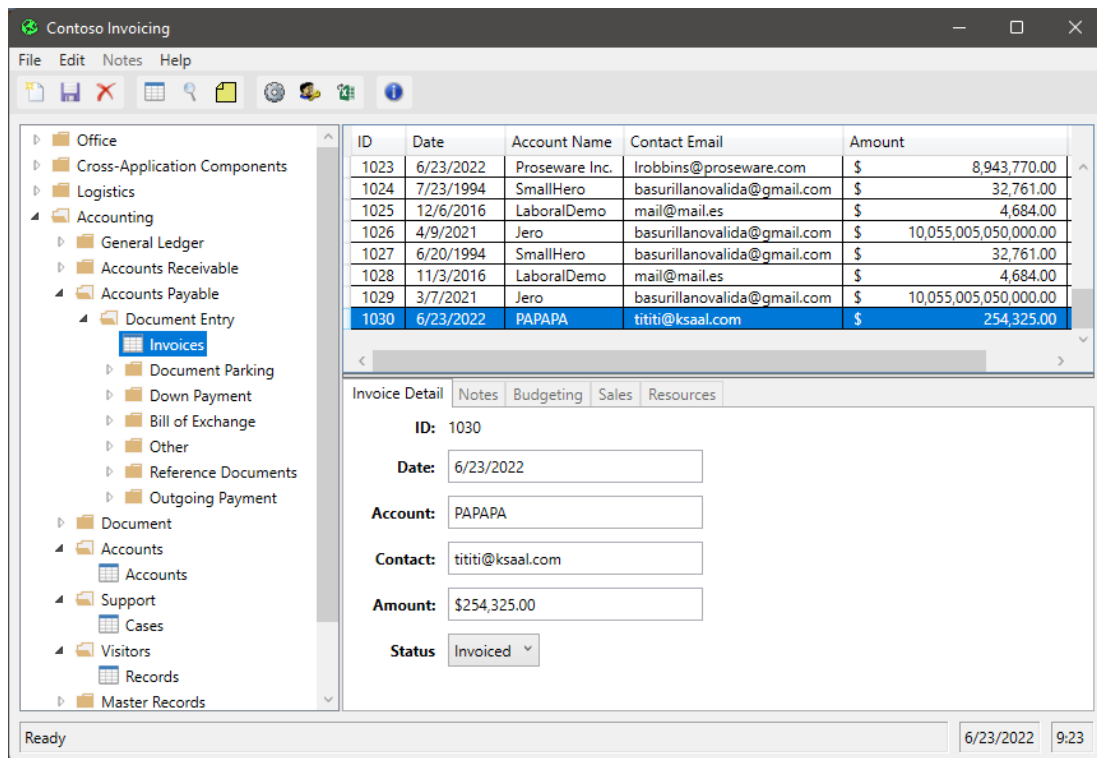


Ilustración 4.4: Ventana de gestión de facturas de Contoso Invoicing

4.2 Caso

Una empresa de grandes almacenes llamada Zeta solicita la automatización de un proceso de gestión de facturas y presupuestos. Entre sus planes a corto plazo está abrir una nueva sucursal, por lo que se espera un aumento de la carga de trabajo en esta área. Actualmente hay una persona encargada de revisar regularmente los emails que llegan a la cuenta del departamento de facturación. Cuando llegan correos nuevos de proveedores clasifica el archivo adjunto en facturas o presupuestos, sube los documentos a sus respectivas carpetas en Sharepoint y desecha aquellos correos que no sean ni una cosa ni otra.

Si se trata de una factura, introduce los datos de facturación en la sección *Invoices* de Contoso. En el caso de que el proveedor fuera nuevo, se crea antes un perfil en la sección Accounts de Contoso. Posteriormente se envía un email al proveedor para comunicar que se ha recibido y procesado adecuadamente su factura. Al final del día envía al departamento financiero un email con los datos de todas las facturas procesadas para su seguimiento contable.

Si se trata de un presupuesto, envía un email al responsable de aprobación para su estudio. Además, a lo largo de la jornada revisa regularmente el correo electrónico para verificar si hay respuestas del responsable de aprobación y reclasificar los presupuestos en “aprobados” o “denegados” según la respuesta recibida. Por último, envía un email al cliente comunicando la decisión respecto al presupuesto y, al final del día, envía un email con la relación de presupuestos aprobados, denegados y pendientes al departamento financiero para su seguimiento contable. Adicionalmente a las tareas descritas, verifica si existen presupuestos pendientes con más de 14 días de antigüedad para enviar un segundo aviso al responsable de aprobación con copia al jefe del departamento.

4.2.1 Abordaje del caso

Tal y como se haría en el ámbito profesional, se creará un informe con la presentación y el estudio detallado del proceso a automatizar. Se estructurará en un PDD (Process Definition Document) que recoge todos los detalles del proceso a automatizar, y un SDD (Solution Definition Document), donde se detallan las soluciones propuestas. Para no extender demasiado este trabajo con temas que están fuera del objetivo que se persigue, se presupondrán realizadas determinadas tareas necesarias en fases previas como, por ejemplo, el análisis del proceso AS-IS en la fase de Discovery, o el Process Mining para determinar las métricas del proceso. Tanto el PDD como el SDD se pueden encontrar en el Anexo, [sección 9.2](#) y [sección 9.3](#) respectivamente.

5 IMPLEMENTACIÓN DE IA EN RPA

Nunca vayas por el camino trazado,
porque conduce hacia donde otros ya
han estado

- Alexander Graham Bell -

5.1 Introducción

Nuestro caso particular trata el problema de clasificación de documentos en imágenes. En general, distintos tipos de documentos suelen tener distinto aspecto visual. Por ejemplo, sin pararnos a leer el contenido podríamos diferenciar de un solo vistazo entre un artículo de prensa y una factura. Sin embargo, el problema es, ciertamente, más complejo que la simple clasificación de imágenes ya que enfrenta dos inconvenientes:

- Existe una enorme variabilidad visual en documentos que pertenecen a la misma clase. Esto complica la clasificación según la disposición espacial e imposibilita la comparación con plantillas.
- Existen importantes similitudes visuales entre documentos de distintas categorías. Por ejemplo, sería difícil distinguir, sin leer el contenido, entre un cuestionario y un formulario, o entre una carta personal y un artículo de prensa.

Algunas soluciones para la resolución de este tipo de problemas suelen pasar por un preprocesamiento de imágenes para el reconocimiento óptico de caracteres (OCR), pero los resultados que ofrecen son ampliamente mejorables.

En los documentos estructurados, la disposición del texto y de las imágenes suelen reflejar información sobre la clase, por lo que documentos de una misma clase tienden a compartir algunas características. Con base en este planteamiento, se empleará una técnica denominada *Spatial Pyramid Matching*, que consiste en un análisis por regiones concatenado a un análisis holístico. De esta forma sería posible construir un modelo que extrajera características generales y locales del documento.[30]

5.2 Modelo de clasificación propuesto

La solución que se propone finalmente para la clasificación se estructura en 3 fases:[31]

1. Se cargan los pesos del modelo VGG16 preentrenado en el dataset ImageNet y, a partir de estos, se entrena el modelo holístico en el dataset RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing)[30]. La finalidad de esta fase es afrontar el problema de similitudes entre clases, mejorando la distinción entre tipos de documentos.
2. A partir de los pesos del modelo holístico se entrenan los modelos basados en regiones. Las regiones son: cabecera, pie, parte izquierda y parte derecha del documento. La finalidad de esta fase es afrontar el problema de la variabilidad visual en documentos del mismo tipo, mejorando el reconocimiento de patrones distintivos de cada clase.
3. A partir de las características extraídas por los 5 modelos anteriores, se crea y entrena una red neuronal cuya salida será la predicción de la clase del documento.

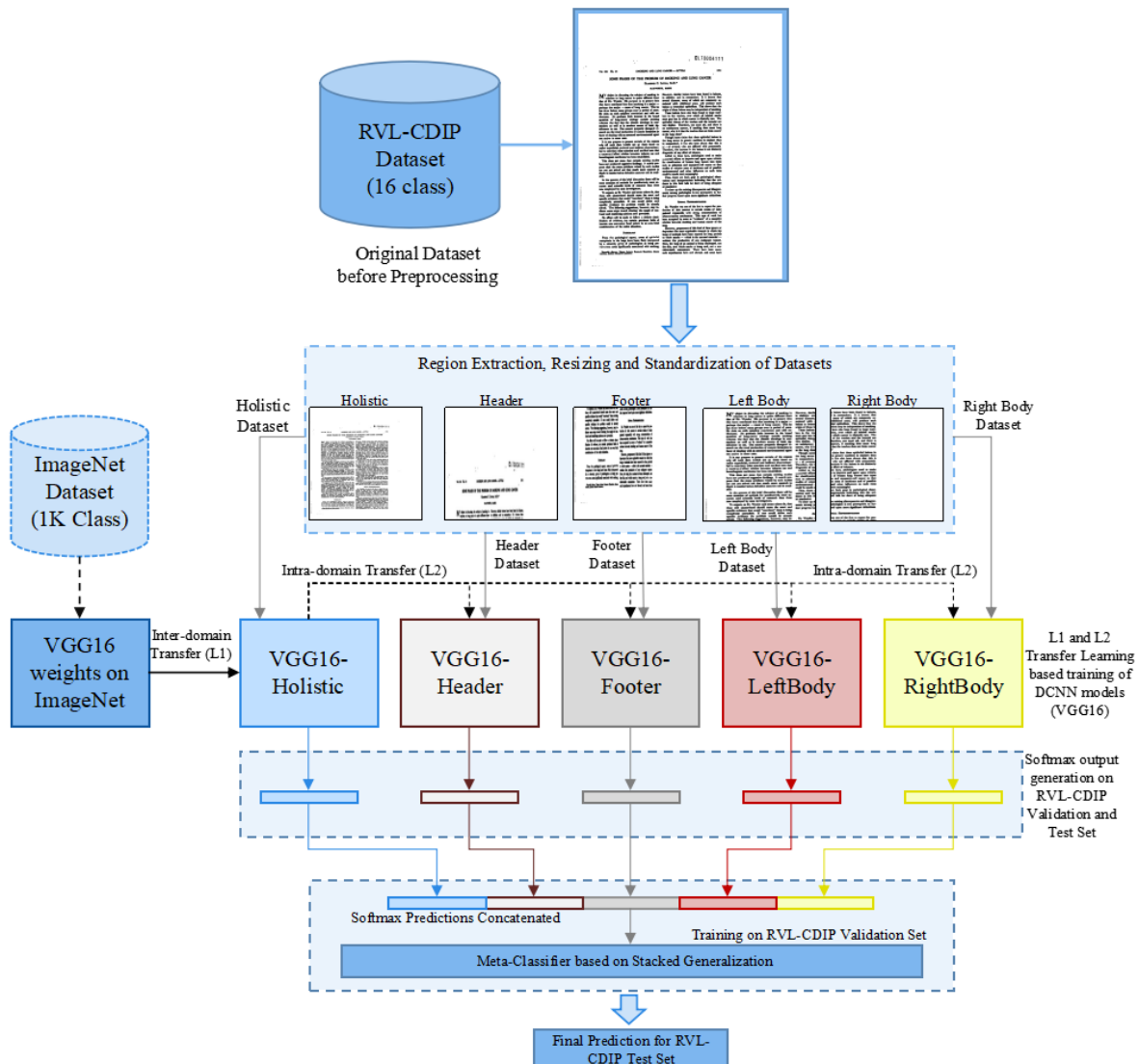


Ilustración 5.1: Estructura general del modelo propuesto [32]

5.3 Motor OCR propuesto

Una vez clasificado el documento, el siguiente objetivo es extraer la fecha de emisión, el montante, y el nombre del proveedor de la factura o presupuesto. Este es un paso de suma importancia porque, junto al resultado del paso anterior, nos permite potencialmente prescindir de la intervención humana en todo el proceso. Es decir, este paso junto al anterior nos permitiría la completa automatización del proceso, pudiendo usar los datos extraídos de los documentos en cualquier parte necesaria del flujo de automatización sin la necesidad de una persona. El enfoque consiste en la resolución de 3 tareas: [33]

- *Reconocimiento de texto*: extracción de palabras completas de una imagen.
- *Detección de texto*: extracción de las coordenadas de una palabra en una imagen.
- *Reconocimiento de información*: comprensión de la semántica del texto para una correcta extracción de información relevante.

Para llevar a cabo nuestro propósito usaremos el motor PaddleOCR en conjunto al modelo NLP LayoutLM.

PaddleOCR es un motor OCR multilingüe muy potente con una gran capacidad para el reconocimiento y detección de texto. La siguiente imagen muestra un resumen del pipeline y los modelos usados para el procesamiento de imágenes por PaddleOCR.

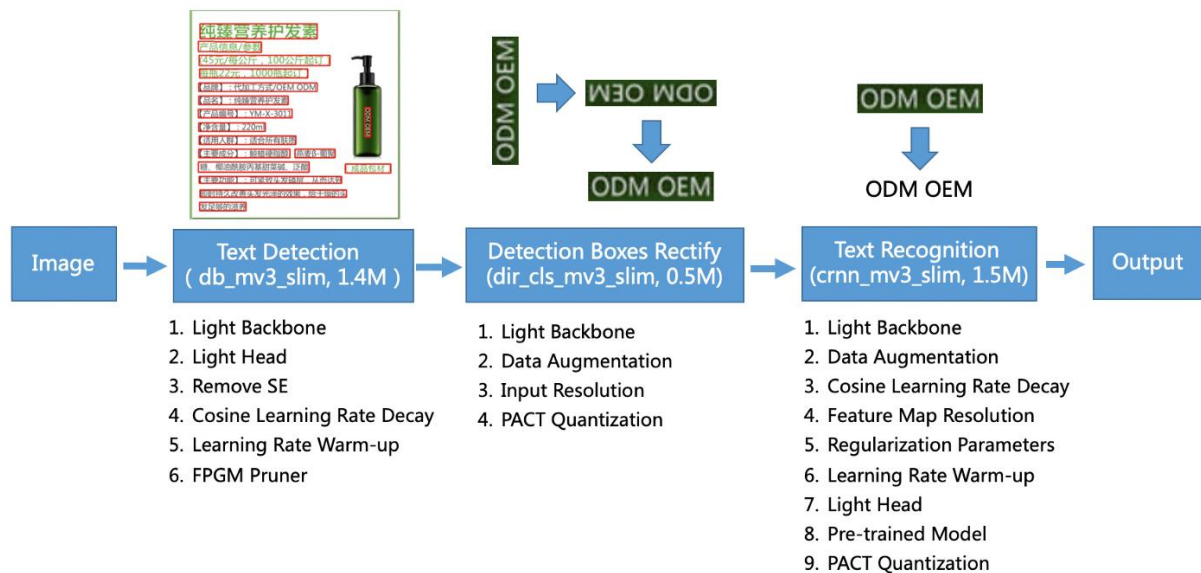


Ilustración 5.2: Resumen del pipeline de PaddleOCR [34]

La estructura de la imagen es la siguiente:

- Los recuadros azules indican la secuencia de actuación del motor. Como elemento de entrada se recibe una imagen. Esta imagen se procesa, en primer lugar, para detectar el texto que contiene. Luego pasa a otro modelo para detectar y rectificar la posición de las cajas de texto detectadas en el paso anterior. Por último, la procesa un modelo de reconocimiento de texto, del que se obtiene la salida final.
- En los recuadros azules, entre paréntesis, se muestra el modelo usado y el tamaño que este tiene en millones de parámetros. En el [proyecto de Github de PaddleOCR](#) se pueden encontrar los modelos actualizados.
- Debajo de cada recuadro se enumeran las estrategias seguidas para mejorar la eficiencia del modelo usado. En el paper oficial [34] se listan pero no ofrecen mayor explicación de en qué consisten.
- Sobre los recuadros azules se muestran imágenes ilustrativas del propósito de cada paso.

La salida que ofrece PaddleOCR se puede resumir en un dataframe con las posiciones de las cajas de las palabras y las palabras extraídas, que será la información de entrada para el modelo LayoutLM.

	x0	y0	x1	y1	x2	y2	x3	y3	text
1									
2	338.0	268.0	466.0	270.0	465.0	310.0	337.0	308.0	GREEN FIELD
3	270.0	307.0	541.0	309.0	541.0	332.0	270.0	330.0	530SEPACIFIC COAST HWY
4	290.0	331.0	518.0	333.0	518.0	358.0	290.0	356.0	LongBeach,CA90004
5	323.0	356.0	485.0	358.0	485.0	383.0	323.0	381.0	(562) 597-0906
6	175.0	406.0	348.0	409.0	348.0	431.0	175.0	428.0	Server:Fancis

Ilustración 5.3: Ejemplo de salida de PaddleOCR formateada en tabla

LayoutLM es un modelo multimodal preentrenado para el procesamiento de imágenes con fines de comprensión de documentos y extracción de información. LayoutLM puede llevar a cabo dos tareas: clasificación de documentos y etiquetado secuencial. Nosotros nos centraremos en el etiquetado secuencial, que permite extraer de las imágenes de los documentos pares clave-valor.

A pesar de que LayoutLM es un modelo ya entrenado, es necesario afinar sus pesos para nuestra tarea concreta, la extracción de información de facturas y presupuestos. Para entrenar este modelo se usará el dataset SROIE del ICDAR2019. [35]

Es necesario ajustar este dataset a nuestras necesidades. Para que LayoutLM efectúe correctamente el etiquetado secuencial necesita que cada caja contenga una única palabra, por lo que necesitamos aislar cada palabra en su propia caja. Para solucionar este problema se ha hecho lo siguiente:

1. Para cada caja devuelta se cuenta el número de palabras que contiene
2. Se divide la caja, a partir de sus coordenadas, en tantas partes iguales como palabras contiene.
3. Se cuenta el número de letras de cada palabra.
4. Se reencuadran las cajas a partir de los tamaños de cada palabra.

Una vez solucionado este problema y con el fin de restringir el entrenamiento a las categorías que nos interesan (fecha de emisión, montante, y nombre del proveedor), necesitamos etiquetar cada palabra para saber a qué categoría pertenece. Una vez etiquetadas ya tendríamos listo nuestro dataset de entrenamiento.

Una vez entrenado, nuestro modelo será capaz de clasificar cada palabra en su categoría correcta y devolverá un dataframe con cada palabra y la categoría a la que pertenece. Si agrupamos por categorías y en orden de aparición las palabras, podremos extraer la información que necesitamos.



```

1 GREEN B-COMPANY
2 FIELD B-COMPANY
3 530SEPACIFIC B-COMPANY
4 COAST B-COMPANY
5 HWY B-COMPANY
6 LongBeach,CA90004 0
7 (562) 597-0906 0
8 Server:Fancis 0
9 Station: 3 0
10 Order#: 69923 0
11 DineIn 0
12 Table:B11 0
13 Guests: 2 0
14 1 0

```

```

1 {
2   "company": "GREEN FIELD 530SEPACIFIC COAST HWY",
3   "date": "5/26/2016",
4   "total": "56.58"
5 }

```

Ilustración 5.4: Parsing de la salida a un JSON con la información requerida

5.4 Implementación en Blue Prism

La inteligencia artificial no es algo nativo de Blue Prism, pero sí han desarrollado algunos módulos que se integran con varias tecnologías para ofrecer este tipo de capacidades. Sin embargo, como cabría esperar, estos módulos de pago no están disponibles en la versión de aprendizaje (gratuita) del programa. Por este motivo, y aprovechando la versatilidad de Blue Prism, he tenido que dar un pequeño rodeo para poder integrar el nuevo apartado de inteligencia artificial en el proceso a automatizar.

Lo que se ha hecho es lo siguiente:

1. Se han entrenado los modelos propuestos en Python 3 a través de Jupyter Notebook.
2. Se han guardado las versiones finales de los modelos para usarlos en las predicciones.
3. Se ha creado un fichero que se usará de pipeline para ambas tareas (clasificación y extracción de información) de tal forma que solo es necesario mover el documento (factura o presupuesto) a la carpeta adecuada, recibir como parámetro el nombre del documento y ejecutar el código para obtener los datos necesarios.

4. Se mantiene abierto Jupyter Notebook en segundo plano continuamente. Así, la solución ha pasado por crear un subproceso en Blue Prism que mueve el fichero a la carpeta especificada, pasa el nombre del documento como dato de entrada y ejecuta el código para recuperar la salida obtenida.
5. La salida se almacena en la base de datos interna de Blue Prism y se continúa el resto del proceso con normalidad.

A continuación mostramos algunas capturas del proceso.

```

Out[27]: 'D:\\TFGRepository\\MyModel'

In [28]: for i in range(len(df)):
         for j in range(16):
             if df['class'][i]==j:
                 shutil.copy2('data/images/'+df['image'][i],'data/train/'+str(j)+'/')
             else:
                 continue

In [29]: #Renaming folder names for train set
class_name={0:'letter','1':'form','2':'email','3':'handwritten','4':'advertisement','5':'scientific'}
path = 'data/train'

i = 0
for j in class_name.keys():
    os.rename(path+'/'+j, path+'/'+class_name[j])
    i=i+1

In [31]: import glob
folders = glob.glob('data/train/*')

In [34]: #Audio(sound_file, autoplay=True)

In [35]: images_list=[]
for i in folders:
    for f in glob.glob(i+'/*.tif"):
        images_list.append(f)

In [42]: pip install --upgrade pip

Requirement already satisfied: pip in c:\users\jerohan\anaconda3\lib\site-packages (22.1.1)
Collecting pip
  Downloading pip-22.1.2-py3-none-any.whl (2.1 MB)
----- 2.1/2.1 MB 10.5 MB/s eta 0:00:00
Installing collected packages: pip
Attempting uninstall: pip
  Found existing installation: pip 22.1.1
  Uninstalling pip-22.1.1:
    Successfully uninstalled pip-22.1.1
  Successfully installed pip-22.1.2
Note: you may need to restart the kernel to use updated packages.

In [43]: pip install opencv-python
  
```

Ilustración 5.5: Preparando los datos de entrenamiento - creando la estructura de carpetas.

El set de imágenes RVL-CDIP viene estructurado de una forma muy extraña, por tanto, se crea una nueva estructura para agrupar las imágenes según su clase, manteniendo la separación train-test.

The screenshot shows a Jupyter Notebook titled 'Model - Jupyter Notebook' running on a local host. The notebook contains several code cells for data processing and model creation. The first cell, 'CONVERTING THE IMAGES OF TRAIN TEST AND VALIDATION TO BATCHES', uses ImageDataGenerator to create batches of images. The second cell, 'SOLVING WITH WHOLE IMAGE', downloads VGG16 weights and creates a model. The third cell, '#Adding custom Layers', adds a custom layer to the model. The fourth cell, '# creating the final model', creates the final model. The fifth cell, '# compile the model', compiles the model. The sixth cell, 'model.summary()', displays the model's architecture.

```

CONVERTING THE IMAGES OF TRAIN TEST AND VALIDATION TO BATCHES

In [68]: #https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/
train_datagen = ImageDataGenerator(rescale = 1./255)
test_datagen = ImageDataGenerator(rescale = 1./255)
validation_datagen = ImageDataGenerator(rescale = 1./255)

train_data = train_datagen.flow_from_directory('data/train/',target_size=(256, 256),batch_size=32)
test_data = test_datagen.flow_from_directory('data/test/',target_size=(256, 256),batch_size=32)
validation_data = validation_datagen.flow_from_directory('data/validation/',target_size=(256,256),batch_size=32)

Found 319837 images belonging to 16 classes.
Found 39997 images belonging to 16 classes.
Found 39995 images belonging to 16 classes.

SOLVING WITH WHOLE IMAGE

In [69]: # Parameters
#using vgg16 with weights of imagenet and input shape is 256,256,3
img_width, img_height = 256,256
model = VGG16(weights = "imagenet", include_top=False, input_shape = (img_width, img_height, 3),)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kerne
ls_notop.h5
58889256/58889256 [=====] - 25 0us/step

In [71]: #using transfer Learning
for layer in model.layers:
    layer.trainable=False
model.layers[-2].trainable=True
model.layers[-3].trainable=True

In [72]: #Adding custom Layers
x = model.output
x = Flatten()(x)
x = Dense(256, activation="relu")(x)
x = Dropout(0.5)(x)
x = Dense(128, activation="relu")(x)
predictions = Dense(16, activation="softmax")(x)

In [74]: # creating the final model
model_ho = Model(model.input, predictions)

In [76]: # compile the model
model_ho.compile(loss = "categorical_crossentropy", optimizer =Adam(learning_rate=0.0001), metrics=["accuracy"])

In [77]: model_ho.summary()

Model: "model"
_____
Layer (type)                 Output Shape          Param #
-----
input_1 (InputLayer)        [(None, 256, 256, 3)] 0
block1_conv1 (Conv2D)       (None, 256, 256, 64) 1792

```

Ilustración 5.6: Creación del modelo holístico

Esta imagen muestra la parte del proceso en que descargo los pesos del modelo preentrenado VGG16 y creo el modelo holístico.

```

In [80]: %time
fit_generator(
    train_data,
    steps_per_epoch = 319837/128,
    epochs=10,
    validation_data=validation_data,
    validation_steps=39995/128,
    callbacks=[reduce_learning,model_save,tensorboard])

<timed exec>:1: UserWarning: 'Model.fit_generator' is deprecated and will be removed in a future version. Please use 'Model.fit', which supports generators.

Epoch 1/10
2499/2498 [=====] - ETA: 0s - loss: 1.1423 - accuracy: 0.6588WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9638s 4s/step - loss: 1.1423 - accuracy: 0.6588 - val_loss: 0.7350 - val_accuracy:
0.7798 - lr: 1.0000e-04
Epoch 2/10
2499/2498 [=====] - ETA: 0s - loss: 0.7809 - accuracy: 0.7718WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9601s 4s/step - loss: 0.7809 - accuracy: 0.7718 - val_loss: 0.6355 - val_accuracy:
0.8116 - lr: 1.0000e-04
Epoch 3/10
2499/2498 [=====] - ETA: 0s - loss: 0.6825 - accuracy: 0.7997WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9631s 4s/step - loss: 0.6825 - accuracy: 0.7997 - val_loss: 0.5946 - val_accuracy:
0.8233 - lr: 1.0000e-04
Epoch 4/10
2499/2498 [=====] - ETA: 0s - loss: 0.6120 - accuracy: 0.8196WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9651s 4s/step - loss: 0.6120 - accuracy: 0.8196 - val_loss: 0.5439 - val_accuracy:
0.8377 - lr: 1.0000e-04
Epoch 5/10
2499/2498 [=====] - ETA: 0s - loss: 0.5617 - accuracy: 0.8343WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 10054s 4s/step - loss: 0.5617 - accuracy: 0.8343 - val_loss: 0.5441 - val_accurac
y: 0.8357 - lr: 1.0000e-04
Epoch 6/10
2499/2498 [=====] - ETA: 0s - loss: 0.5329 - accuracy: 0.8417WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 10068s 4s/step - loss: 0.5329 - accuracy: 0.8417 - val_loss: 0.4970 - val_accurac
y: 0.8526 - lr: 1.0000e-04
Epoch 7/10
2499/2498 [=====] - ETA: 0s - loss: 0.5065 - accuracy: 0.8506WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9801s 4s/step - loss: 0.5065 - accuracy: 0.8506 - val_loss: 0.4881 - val_accuracy:
0.8549 - lr: 1.0000e-04
Epoch 8/10
2499/2498 [=====] - ETA: 0s - loss: 0.4726 - accuracy: 0.8602WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9875s 4s/step - loss: 0.4726 - accuracy: 0.8602 - val_loss: 0.4892 - val_accuracy:
0.8595 - lr: 1.0000e-04
Epoch 9/10
2499/2498 [=====] - ETA: 0s - loss: 0.4507 - accuracy: 0.8654WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9568s 4s/step - loss: 0.4507 - accuracy: 0.8654 - val_loss: 0.5010 - val_accuracy:
0.8552 - lr: 1.0000e-04
Epoch 10/10
2499/2498 [=====] - ETA: 0s - loss: 0.3744 - accuracy: 0.8886WARNING:tensorflow:Can save best model on
ly with acc available, skipping.
2498/2498 [=====] - 9597s 4s/step - loss: 0.3744 - accuracy: 0.8886 - val_loss: 0.4357 - val_accuracy:
0.8730 - lr: 2.0000e-05
CPU times: total: 5d 22h 12min 9s
wall time: 1d 3h 4min 37s

```

Ilustración 5.7: Entrenamiento del modelo holístico

Esta imagen muestra el proceso de entrenamiento del modelo holístico. En los valores *CPU times total* y *Wall time* se aprecia el altísimo coste computacional que tuvo el entrenamiento de este modelo. El resto del proceso es prácticamente idéntico para los otros modelos.

Por otro lado, para conectar Blue Prism con Jupyter Notebook he creado un objeto en Blue Prism llamado “Modelo”. A este objeto le he enseñado a reconocer los elementos UI en Chrome que necesito para ejecutar el código correctamente. En la [sección 9.1.1](#) del Anexo se detalla la creación de objetos en Blue Prism y el reconocimiento de elementos UI.

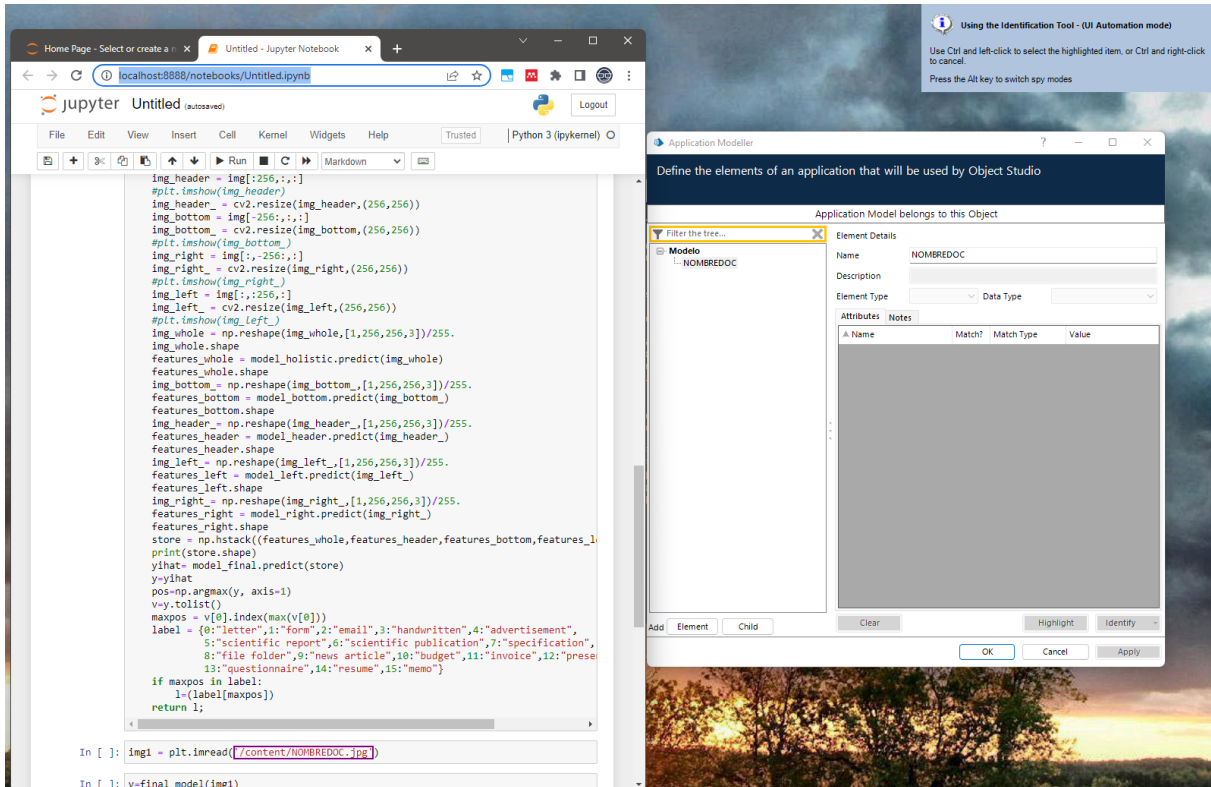


Ilustración 5.8: Reconocimiento del elemento donde BP debe insertar el nombre del documento

Al hacer click en *Identify*, Blue Prism nos permite resaltar los elementos que necesitamos que reconozca. Una vez detectado, configuramos algunas opciones para que la identificación sea unívoca, o sea, no pueda confundirse de elemento pero sea capaz de encontrarlo. Para verificar que efectivamente es capaz de reconocerlo, debemos pinchar sobre *Highlight*.

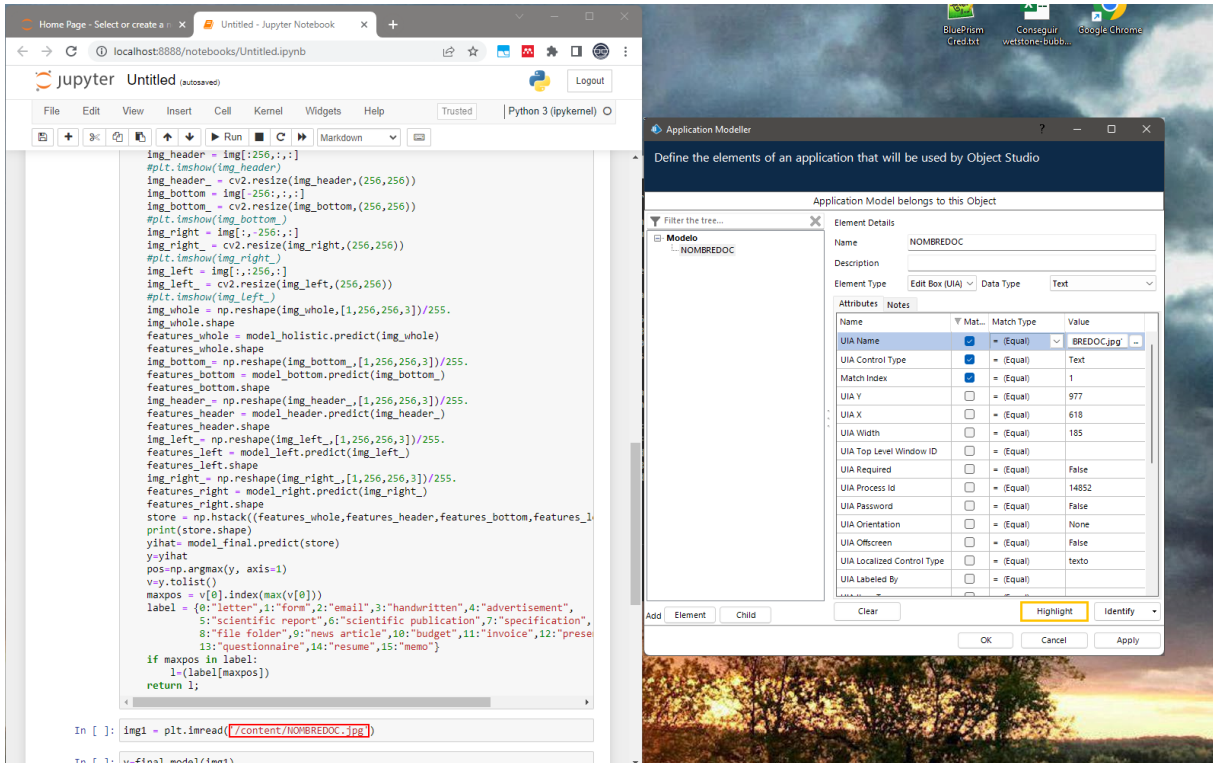
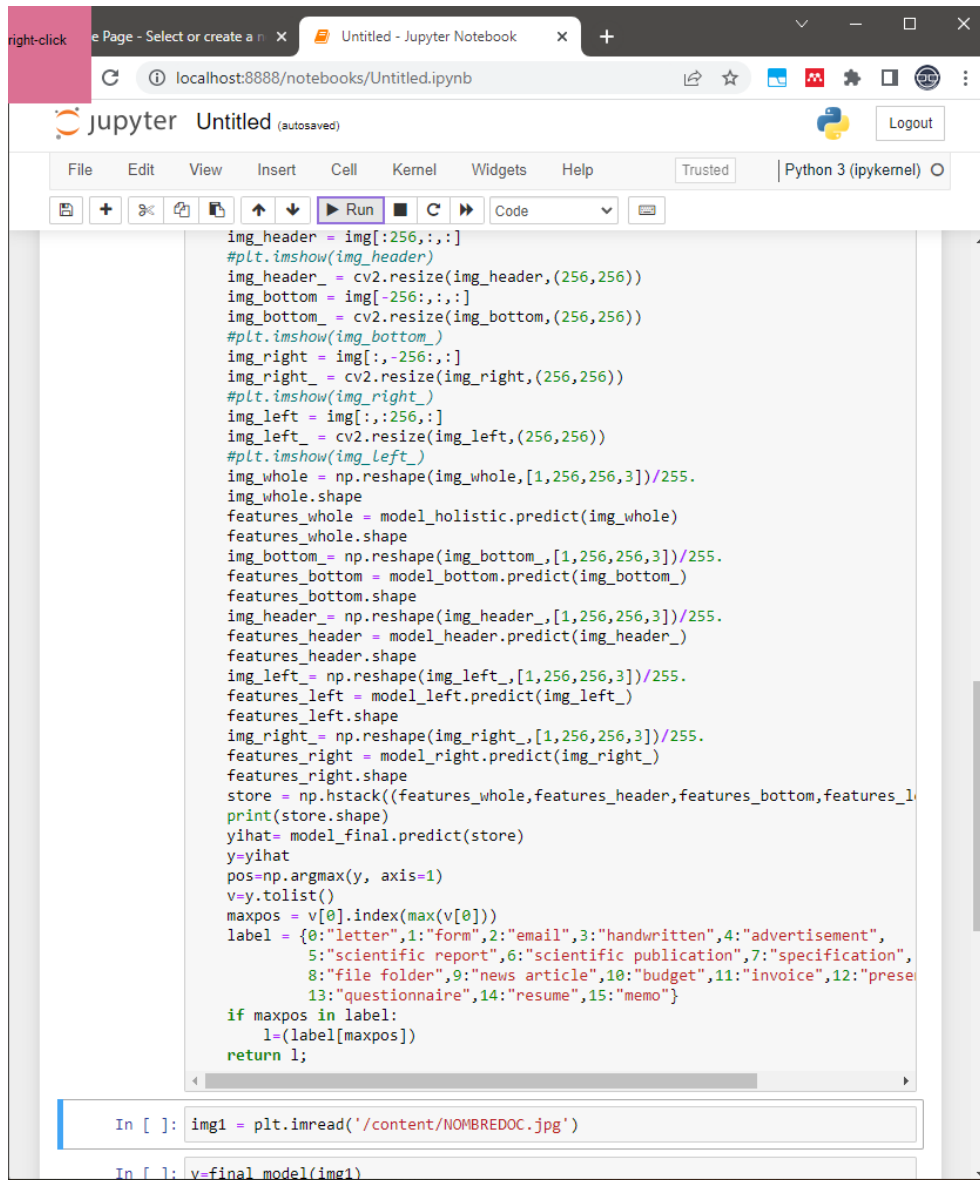


Ilustración 5.9: Blue Prism reconoce adecuadamente el elemento que necesitamos

El recuadro rojo indica que Blue Prism es capaz de reconocer el elemento y que, además, solo encuentra uno. Ahora, para poder ejecutar los chunks de código que necesitamos, solo tenemos que detectar el botón Run. De esta forma, lo único que debemos hacer es ir a los chunks que queramos, insertar la información que necesitamos y hacer click sobre el botón Run.



The image shows a Jupyter Notebook interface in a web browser. The browser address bar shows 'localhost:8888/notebooks/Untitled.ipynb'. The Jupyter interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with a highlighted 'Run' button. The main area contains a code cell with the following Python code:

```
img_header = img[:256,:,:]
#plt.imshow(img_header)
img_header_ = cv2.resize(img_header,(256,256))
img_bottom = img[-256,:,:]
img_bottom_ = cv2.resize(img_bottom,(256,256))
#plt.imshow(img_bottom_)
img_right = img[:, -256,:]
img_right_ = cv2.resize(img_right,(256,256))
#plt.imshow(img_right_)
img_left = img[:, :256,:]
img_left_ = cv2.resize(img_left,(256,256))
#plt.imshow(img_left_)
img_whole = np.reshape(img_whole,[1,256,256,3])/255.
img_whole.shape
features_whole = model_holistic.predict(img_whole)
features_whole.shape
img_bottom_ = np.reshape(img_bottom_,[1,256,256,3])/255.
features_bottom = model_bottom.predict(img_bottom_)
features_bottom.shape
img_header_ = np.reshape(img_header_,[1,256,256,3])/255.
features_header = model_header.predict(img_header_)
features_header.shape
img_left_ = np.reshape(img_left_,[1,256,256,3])/255.
features_left = model_left.predict(img_left_)
features_left.shape
img_right_ = np.reshape(img_right_,[1,256,256,3])/255.
features_right = model_right.predict(img_right_)
features_right.shape
store = np.hstack((features_whole,features_header,features_bottom,features_l
print(store.shape)
yihat= model_final.predict(store)
y=yihat
pos=np.argmax(y, axis=1)
v=y.tolist()
maxpos = v[0].index(max(v[0]))
label = {0:"letter",1:"form",2:"email",3:"handwritten",4:"advertisement",
5:"scientific report",6:"scientific publication",7:"specification",
8:"file folder",9:"news article",10:"budget",11:"invoice",12:"prese
13:"questionnaire",14:"resume",15:"memo"}
if maxpos in label:
l=(label[maxpos])
return l;
```

Below the code cell, there are two input prompts:

```
In [ ]: img1 = plt.imread('/content/NOMBREDOC.jpg')
```

```
In [ ]: v=final_model(img1)
```

Ilustración 5.10: Reconocimiento del botón Run

Para recuperar la información debemos reconocer la salida.

```

y=final_model(img1)
print(y)

In [13]:
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:54
1: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:44
32: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:42
67: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:6
6: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:14
8: The name tf.placeholder_with_default is deprecated. Please use tf.compat.v1.placeholder_with_defau
lt instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:37
33: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be remo
ved in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:19
0: The name tf.get_default_session is deprecated. Please use tf.compat.v1.get_default_session instea
d.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:19
7: The name tf.ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:20
3: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:20
7: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:21
6: The name tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized
instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:22
3: The name tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer ins
tead.

Loaded model from disk Holistic
Loaded model from disk Header
Loaded model from disk Bottom
Loaded model from disk Left
Loaded model from disk Right
WARNING:tensorflow:Large dropout rate: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate ins
tead of keep_prob. Please ensure that this is intended.
Loaded the final model from disk
(1, 80)
presentation

```

Ilustración 5.11: Salida de la clasificación de una imagen

A Blue Prism le resulta imposible reconocer como elemento único el resultado que necesitamos (en este caso sería “presentation”). Esto se debe a que toda la salida es un único elemento html `<pre>`.

```

<div data-mime-type="text/plain" class="jp-RenderedText jp-OutputArea-output">
  <pre> == $0
"WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:541: The name tf.placeholder is
deprecated. Please use tf.compat.v1.placeholder instead. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:4432: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:4267: The name tf.nn.max_pool is
deprecated. Please use tf.nn.max_pool2d instead. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:66: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph
instead. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:148: The name
tf.placeholder_with_default is deprecated. Please use tf.compat.v1.placeholder_with_default instead. WARNING:tensorflow:From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:3733: calling dropout (from tensorflow.python.ops.nn_ops) with
keep_prob is deprecated and will be removed in a future version. Instructions for updating: Please use `rate` instead of `keep_prob`. Rate
should be set to `rate = 1 - keep_prob`. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:190: The name tf.get_default_session is deprecated. Please use tf.compat.v1.get_default_session
instead. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:197: The name tf.ConfigProto is
deprecated. Please use tf.compat.v1.ConfigProto instead. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:203: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:207: The name tf.global_variables is
deprecated. Please use tf.compat.v1.global_variables instead. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:216: The name tf.is_variable_initialized is deprecated. Please use
tf.compat.v1.is_variable_initialized instead. WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/keras/backend/tensorflow_backend.py:223: The name tf.variables_initializer is deprecated. Please use
tf.compat.v1.variables_initializer instead. Loaded model from disk Holistic Loaded model from disk Header Loaded model from disk Bottom Loaded
model from disk Left Loaded model from disk Right WARNING:tensorflow:Large dropout rate: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout
rate instead of keep_prob. Please ensure that this is intended. Loaded the final model from disk (1, 80) presentation "
  </pre>
</div>

```

Ilustración 5.12: Formato html de la salida obtenida

No obstante, esto no supone realmente un problema, solo basta con almacenar toda la salida y hacer una búsqueda de las palabras *invoice* o *budget* posteriormente.

6 RESULTADOS

Los resultados se consiguen al explotar
oportunidades, no por resolver
problemas

-Peter Drucker-

6.1 Resultados de los modelos

En general los modelos han ofrecido buenos resultados. Esto es importante porque una mala clasificación o extracción de información errónea deriva posteriormente en un “business exception” que paraliza el proceso, gastando tiempo y recursos en encontrar y solucionar el error. A continuación se analizan por separado los resultados de los procesos de clasificación y de extracción de información.

6.1.1 Clasificación

El modelo de clasificación se estructuraba en 5 submodelos: uno basado en la imagen completa y otros 4 para las distintas regiones de la imagen. El dataset usado para el entrenamiento de estos modelos, como ya comentamos en la [sección 5.2](#), era RVL-CDIP. Este dataset se compone de 400.000 imágenes en escala de grises distribuidas en 16 clases, con 25.000 imágenes por clase. Viene ya separado en 320.000 imágenes de entrenamiento, 40.000 de validación y 40.000 de prueba. El tamaño de cada imagen ha sido redimensionado para que el ancho sea de 1.000 píxeles. Las 16 categorías de clasificación son:

- Carta
- Memorándum
- Presupuesto
- Presentación
- Formulario
- Manuscrito
- Factura
- Publicidad
- Correo electrónico
- Artículo de noticias
- Carpeta de archivos
- Publicación científica
- Cuestionario
- Curriculum
- Informe científico
- Folleto informativo

Para contrastar la efectividad de los modelos se ha usado como métrica la **precisión**. Los resultados arrojados del entrenamiento de cada modelo han sido los que se detallan a continuación:

Modelo holístico

Las especificaciones del modelo, cuyo código se mostró en la [sección 5.4](#), son:

```
In [77]: model_ho.summary()
Model: "model"
-----
```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 256, 256, 3)]	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0
flatten_1 (Flatten)	(None, 32768)	0
dense_3 (Dense)	(None, 256)	8388864
dropout_1 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32896
dense_5 (Dense)	(None, 16)	2064

```
-----
Total params: 23,138,512
Trainable params: 13,143,440
Non-trainable params: 9,995,072
-----
```

Ilustración 6.1: Especificaciones del modelo holístico

Se ha definido un batch size de 128 sobre un total de 320.000 imágenes del conjunto de entrenamiento, lo que produce un total de 2.500 iteraciones sobre 10 epochs o recorridos completos del dataset, como ya se vio en la [sección 5.4](#). La precisión alcanzada por el modelo holístico ha sido del 87.13%.

```
score
[0.44351495553057985, 0.871305910543131]
```

Ilustración 6.2: Precisión del modelo holístico sobre el dataset de evaluación

Modelos por partes

Para entrenar cada modelo por partes (Cabecera, pie, parte izquierda y parte derecha), se ha definido un batch size de 128 sobre un total de 320.000 imágenes del conjunto de entrenamiento, lo que produce un total de 2.500 iteraciones sobre 5 epochs o recorridos completos del dataset (10 epochs por error en la parte de la cabecera). No hay resultados de la precisión de cada modelo porque consideré que carecía de sentido evaluarlos por separado. Precisamente el objetivo de los modelos por partes es, en conjunto, mejorar los resultados del modelo holístico. A continuación se muestran los resultados del entrenamiento de cada modelo. Tomamos como referencia la precisión de la validación.

Cabecera

```
Epoch 1/10
2499/2498 [=====] - 1058s 424ms/step - loss: 0.8831 - acc: 0.7398 - val_loss: 0.7162 - val_acc: 0.7902
Epoch 2/10
2499/2498 [=====] - 1080s 432ms/step - loss: 0.7374 - acc: 0.7840 - val_loss: 0.6352 - val_acc: 0.8149
Epoch 3/10
2499/2498 [=====] - 1070s 428ms/step - loss: 0.6850 - acc: 0.8017 - val_loss: 0.6028 - val_acc: 0.8246
Epoch 4/10
2499/2498 [=====] - 1057s 423ms/step - loss: 0.6517 - acc: 0.8103 - val_loss: 0.5529 - val_acc: 0.8365
Epoch 5/10
2499/2498 [=====] - 1054s 422ms/step - loss: 0.5468 - acc: 0.8389 - val_loss: 0.5735 - val_acc: 0.8301
Epoch 6/10
2499/2498 [=====] - 1065s 426ms/step - loss: 0.5437 - acc: 0.8408 - val_loss: 0.5501 - val_acc: 0.8409
Epoch 7/10
2499/2498 [=====] - 1055s 422ms/step - loss: 0.5391 - acc: 0.8412 - val_loss: 0.5561 - val_acc: 0.8408
Epoch 8/10
2499/2498 [=====] - 1033s 413ms/step - loss: 0.5457 - acc: 0.8408 - val_loss: 0.5313 - val_acc: 0.8476
Epoch 9/10
2499/2498 [=====] - 1025s 410ms/step - loss: 0.4395 - acc: 0.8678 - val_loss: 0.5306 - val_acc: 0.8460
Epoch 10/10
2499/2498 [=====] - 1032s 413ms/step - loss: 0.4455 - acc: 0.8677 - val_loss: 0.5460 - val_acc: 0.8456
```

Ilustración 6.3: Entrenamiento del modelo por regiones: Cabecera

Se aprecia una buena mejora de la precisión durante los primeros 5 epochs. A partir de del 6º se sigue mejorando lentamente hasta el 8º, donde alcanza el máximo. A partir de aquí se podría decir que se está sobreajustando el modelo.

Pie

```

Epoch 1/5
2499/2498 [=====] - 1178s 471ms/step - loss: 0.5684 - acc: 0.8269 - val_loss: 0.6193 - val_acc: 0.8182
Epoch 2/5
2499/2498 [=====] - 1160s 464ms/step - loss: 0.5490 - acc: 0.8323 - val_loss: 0.6504 - val_acc: 0.8088
Epoch 3/5
2499/2498 [=====] - 1168s 467ms/step - loss: 0.4725 - acc: 0.8539 - val_loss: 0.6465 - val_acc: 0.8161
Epoch 4/5
2499/2498 [=====] - 1183s 473ms/step - loss: 0.4423 - acc: 0.8622 - val_loss: 0.6479 - val_acc: 0.8201
Epoch 5/5
2499/2498 [=====] - 1193s 477ms/step - loss: 0.4249 - acc: 0.8682 - val_loss: 0.6218 - val_acc: 0.8230

```

Ilustración 6.4: Entrenamiento del modelo por regiones: Pie

En 5 epochs apenas se consigue mejorar la precisión del modelo un 0.5%. Cabe la posibilidad de que el modelo siguiera mejorando, pero no compensaría la cantidad mejorada con el coste de entrenamiento.

Parte izquierda

```

Epoch 1/5
2499/2498 [=====] - 1031s 412ms/step - loss: 0.8722 - acc: 0.7422 - val_loss: 0.7143 - val_acc: 0.7872
Epoch 2/5
2499/2498 [=====] - 1028s 411ms/step - loss: 0.7344 - acc: 0.7852 - val_loss: 0.6487 - val_acc: 0.8065
Epoch 3/5
2499/2498 [=====] - 1033s 413ms/step - loss: 0.6898 - acc: 0.7995 - val_loss: 0.6136 - val_acc: 0.8189
Epoch 4/5
2499/2498 [=====] - 1029s 412ms/step - loss: 0.6627 - acc: 0.8072 - val_loss: 0.5893 - val_acc: 0.8235
Epoch 5/5
2499/2498 [=====] - 1044s 418ms/step - loss: 0.5591 - acc: 0.8354 - val_loss: 0.5959 - val_acc: 0.8302

```

Ilustración 6.5: Entrenamiento del modelo por regiones: parte izquierda

El entrenamiento de este modelo parece reflejar que tenía más margen de mejora. Si nos fijamos, hay un salto grande en la mejora de la precisión en el primer epoch, y otro salto algo más pequeño en el último. Quizás, con más epochs se podría haber mejorado más el modelo. No obstante, como ya se vio con el modelo holístico, el coste computacional es alto.

Parte derecha

```

Epoch 1/5
2499/2498 [=====] - 1039s 416ms/step - loss: 0.9517 - acc: 0.7122 - val_loss: 0.7467 - val_acc: 0.7702
Epoch 2/5
2499/2498 [=====] - 1054s 422ms/step - loss: 0.7858 - acc: 0.7638 - val_loss: 0.6907 - val_acc: 0.7915
Epoch 3/5
2499/2498 [=====] - 1060s 424ms/step - loss: 0.7385 - acc: 0.7791 - val_loss: 0.6515 - val_acc: 0.8030
Epoch 4/5
2499/2498 [=====] - 1065s 426ms/step - loss: 0.7051 - acc: 0.7880 - val_loss: 0.6396 - val_acc: 0.8110
Epoch 5/5
2499/2498 [=====] - 1061s 425ms/step - loss: 0.6024 - acc: 0.8172 - val_loss: 0.6220 - val_acc: 0.8178

```

Ilustración 6.6: Entrenamiento del modelo por regiones: parte derecha

Se aprecia un salto grande del primer al segundo epoch y otro algo menor del 2º al 3º. A partir de aquí la mejora parece ser lenta.

Red Neuronal Artificial y modelo final

Para la construcción de esta RNA se ha considerado como función de activación para las capas ocultas la función ReLU, y se ha acabado con la función Softmax para la capa de salida. En este caso se ha usado un batch size de 128 y solo 2 epochs. El motivo de usar una cantidad de epochs tan extremadamente baja es el coste computacional que tiene y la falta de tiempo y potencia para dejar el modelo más tiempo.

```

Train on 319837 samples, validate on 39995 samples
Epoch 1/2
319837/319837 [=====] - 9s 29us/step - loss: 0.4974 - acc: 0.8461 - val_loss: 0.3759 - val_acc: 0.9325
Epoch 2/2
319837/319837 [=====] - 9s 28us/step - loss: 0.2592 - acc: 0.9182 - val_loss: 0.4203 - val_acc: 0.9341

```

Ilustración 6.7: Entrenamiento del modelo final

No obstante parece ser que la mejoría que obtiene el modelo entre el primer y segundo epoch es extremadamente baja. Mejorar este modelo hubiera sido demasiado costoso.

```

score=model_final.evaluate(test_features,test_labels,batch_size=128)
print(score)

39996/39996 [=====] - 0s 11us/step
[0.4689435036187914, 0.9185668566886491]

```

Ilustración 6.8: Precisión del modelo final sobre el dataset de evaluación

La precisión alcanzada por el modelo final sobre el conjunto test ha sido del 91.86%.

6.1.2 Extracción de información

El modelo de extracción de información se basaba en el reconocimiento de caracteres con PaddleOCR y en la correcta asignación de etiquetas en formato BIO a través de un modelo de clasificación basado en LayoutML. El dataset usado para el entrenamiento de este modelo, como ya se comentó en la [sección 5.3](#), era SROIE del ICDAR2019. Este dataset se compone de 1.000 imágenes de recibos con anotaciones y fue creado expresamente para la competición ICDAR2019, que consistía en resolver las tareas propuesta en el punto 5.3.

El dataset para el entrenamiento y validación viene estructurado de la siguiente manera:

1. **Img:** contiene la imagen de cada recibo

The image shows a receipt from 'tan woon yann' (BOOK TA .K (TAMAN DAYA) SDN BHD) with a total amount of 9.00 RM. The receipt includes details like date (25/12/2018), cashier (MANIS), and a handwritten '9.00' in a circle. To the right, a 'Data Explorer' window shows a file list for the 'img' folder, listing files like 'X00016469612.jpg' through 'X51005396179.jpg'.

Ilustración 6.9: SROIE dataset: imagen de un recibo

2. **Box:** contiene la salida de PaddleOCR para cada recibo

X00016469612.txt (1.86 KiB) [↓](#) [🔍](#) [>](#)

```
72,25,326,25,326,64,72,64,TAN WOON YANN
50,82,440,82,440,121,50,121,BOOK TA .K(TAMAN DAYA) SDN BND
205,121,285,121,285,139,205,139,789417-W
110,144,383,144,383,163,110,163,NO.53 55,57 & 59, JALAN SAGU 18,
192,169,299,169,299,187,192,187,TAMAN DAYA,
162,193,334,193,334,211,162,211,81100 JOHOR BAHRU,
217,216,275,216,275,233,217,233,JOHOR.
50,342,279,342,279,359,50,359,DOCUMENT NO : TD01167104
50,372,96,372,96,390,50,390,DATE:
165,372,342,372,342,389,165,389,25/12/2018 8:13:39 PM
```

Data Explorer
Version 4 (1.02 GiB)

- ▼ SROIE2019
 - ▶ layoutlm-base-uncased
 - ▶ test
 - ▼ train
 - ▼ box
 - X00016469612.txt
 - X00016469619.txt
 - X00016469620.txt
 - X00016469622.txt
 - X00016469623.txt

Ilustración 6.10: SROIE dataset: OCR de un recibo

3. **Entities:** contiene la información, en formato JSON, que se pretende extraer de cada recibo.

X00016469612.txt (188 B) [↓](#) [🔍](#) [>](#)

```
{
  "company": "BOOK TA .K (TAMAN DAYA) SDN BHD",
  "date": "25/12/2018",
  "address": "NO.53 55,57 & 59, JALAN SAGU 18, TAMAN DAYA, 81100 JOHOR BAHRU, JOHOR.",
  "total": "9.00"
}
```

Data Explorer
Version 4 (1.02 GiB)

- ▼ SROIE2019
 - ▶ layoutlm-base-uncased
 - ▶ test
 - ▼ train
 - ▶ box
 - ▼ entities
 - X00016469612.txt
 - X00016469619.txt
 - X00016469620.txt
 - X00016469622.txt

Ilustración 6.11: SROIE dataset: información extraída de un recibo

Las especificaciones del modelo han sido

```
{
  "attention_probs_dropout_prob": 0.1,
  "finetuning_task": null,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "is_decoder": false,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "max_2d_position_embeddings": 1024,
  "num_attention_heads": 8,
  "num_hidden_layers": 12,
  "num_labels": 2,
  "output_attentions": false,
  "output_hidden_states": false,
  "output_past": true,
  "pruned_heads": {},
  "torchscript": false,
  "type_vocab_size": 2,
  "use_bfloat16": false,
  "vocab_size": 30522
}
```

Ilustración 6.12: Especificaciones del modelo

Para contrastar la efectividad del modelo se ha usado como métrica la **precisión**. Los resultados arrojados del entrenamiento han sido los siguientes:

```
Iteration: 70% 50/71 [00:41<00:17, 1.22it/s]
Iteration: 72% 51/71 [00:41<00:16, 1.21it/s]
Iteration: 73% 52/71 [00:42<00:15, 1.22it/s]
Iteration: 75% 53/71 [00:43<00:14, 1.21it/s]
Iteration: 76% 54/71 [00:44<00:13, 1.21it/s]
Iteration: 77% 55/71 [00:45<00:13, 1.22it/s]
Iteration: 79% 56/71 [00:46<00:12, 1.21it/s]
Iteration: 80% 57/71 [00:46<00:11, 1.21it/s]
Iteration: 82% 58/71 [00:47<00:10, 1.21it/s]
Iteration: 83% 59/71 [00:48<00:09, 1.21it/s]
Iteration: 85% 60/71 [00:49<00:09, 1.21it/s]
Iteration: 86% 61/71 [00:50<00:08, 1.21it/s]
Iteration: 87% 62/71 [00:51<00:07, 1.21it/s]
Iteration: 89% 63/71 [00:51<00:06, 1.21it/s]
Iteration: 90% 64/71 [00:52<00:05, 1.21it/s]
Iteration: 92% 65/71 [00:53<00:04, 1.21it/s]
Iteration: 93% 66/71 [00:54<00:04, 1.21it/s]
Iteration: 94% 67/71 [00:55<00:03, 1.21it/s]
Iteration: 96% 68/71 [00:56<00:02, 1.20it/s]
Iteration: 97% 69/71 [00:56<00:01, 1.21it/s]
Iteration: 99% 70/71 [00:57<00:00, 1.21it/s]
Iteration: 100% 71/71 [00:58<00:00, 1.21it/s]
Epoch: 100% 5/5 [04:44<00:00, 56.92s/it]
```

Ilustración 6.13: Parte final del proceso de entrenamiento del modelo de extracción de información

La precisión alcanzada por el modelo sobre el conjunto test ha sido del 91.39%

```
cat output/test_results.txt
```

```
f1 = 0.945092952875054
loss = 0.0345635545026097
precision = 0.9138795986622074
recall = 0.9785138764547896
```

Ilustración 6.14: Resultados del modelo de extracción de información sobre el conjunto de evaluación

6.2 Resultados de la automatización de procesos

Recordemos que el propósito de este proyecto consistía en determinar y comparar el impacto que supone automatizar un proceso empleando tanto RPA como RPA+IA. El objetivo fundamental de la automatización de un proceso es reducir los recursos necesarios para llevarlo a cabo, entendiendo por recursos tanto los humanos como los costes asociados. Con este objetivo en mente y tratando de buscar la manera más efectiva de medir la inversión de recursos en un proceso, se propone como métrica el Full Time Equivalent (FTE).

El FTE es un indicador que permite medir el rendimiento del trabajador según las horas trabajadas. Se calcula como el cociente de las horas efectivas de trabajo entre las horas de la jornada. Las horas efectivas de trabajo son la cantidad de tiempo que un empleado invierte en realizar un determinado trabajo o tarea. Llevado a un ámbito mensual, el cálculo sería

$$\text{FTE} = \frac{\text{Horas efectivas}}{160 \text{ horas de trabajo en un mes}}$$

Se ha decidido usar esta métrica porque permite calcular el número de recursos (trabajadores en este caso) que consume o son necesarios para desarrollar una tarea. Por ejemplo, supongamos que tenemos 3 trabajadores desempeñando una tarea. Cada trabajador invierte al mes 100 horas, 80 horas y 60 horas de sus jornadas respectivamente en llevar a cabo esta tarea. Eso supondría

$$\text{FTE} = \frac{100 + 80 + 60}{160} = 1.5$$

O sea, sería equivalente tener a un empleado a tiempo completo y a otro a media jornada para llevar a cabo esta tarea.

6.2.1 Proceso sin RPA

Tal como se refleja en el PDD (Process Definition Document) se reciben, de media, 700 emails al mes. Actualmente hay un trabajador encargado del proceso de gestión de facturas y presupuestos. Tarda, en promedio, unos 7 minutos en realizar el proceso completo. Por tanto,

$$\frac{7 \text{ min} \times 700 \text{ vol. medio mensual} \times 1 \text{ recurso}}{60 \text{ min en hora}} = 81,67 \text{ horas} \Rightarrow \text{FTE} = \frac{81,67}{160} = 0,5104$$

O, lo que es lo mismo, el trabajador dedica el 51,04% de su jornada, algo más de la mitad, a gestionar los documentos.

Hablando en términos económicos, suponiendo que el trabajador cobrara el SMI (14.000€ brutos anuales) y que el resto de su jornada lo dedicara a otras tareas, se podría decir que el coste de la gestión de este proceso es de

$$14000 \times 0,5104 = 7145.6\text{€ anuales}$$

6.2.2 Proceso con RPA atendida

Como los documentos recibidos son datos no estructurados y no empleamos IA en esta solución, Blue Prism es incapaz de discernir los tipos de documentos y extraer la información requerida. Por tanto, es necesario tener a un trabajador que realice la tarea de clasificación y extracción de información. Además, durante la ejecución del proceso automatizado, han ocurrido algunas *business exceptions*.

Una *business exception* es un error de naturaleza conocida y, por tanto, previsto pero que necesitan intervención humana, así que el tiempo empleado en solucionarlos se añadirá a las horas de trabajo efectivo del trabajador.

Dada la dimensión de este proyecto no han ocurrido errores de tipo *system exceptions*. Este tipo de errores suelen darse por motivos desconocidos ajenos al proceso y relacionados con el entorno de ejecución como, por ejemplo, desbordamiento de memoria o inaccesibilidad a algún recurso interno. Cuando este tipo de errores ocurren, el documento en proceso se vuelve a poner en cola para reintentar su gestión más tarde. Si después de tres reintentos el documento vuelve a arrojar este error, estamos ante una excepción final. En este caso el proceso se para y requiere de intervención humana para solucionar el problema. En cualquier caso, el efecto es el aumento en el tiempo de ejecución del robot y, en última instancia, la intervención humana.

Un buen proceso no debería contener más de un 2% de *system exceptions* finales, por este motivo se ha decidido aumentar unos 10 segundos extra al tiempo de ejecución media del robot y otros 10 segundos extra a las horas de trabajo efectivo del trabajador.

En esta solución el trabajador tarda, de media por documento, 1 minuto y 20 segundos en revisar el email, descargar el documento, leerlo y anotar en un Excel la información. Además, se han añadido 30 segundos al tiempo medio en concepto de business exceptions y system exceptions.

Esto supone

$$\frac{1,83 \text{ min} \times 700 \text{ vol. medio mensual} \times 1 \text{ recurso}}{60 \text{ min en hora}} = 21,39 \text{ horas} \Rightarrow \text{FTE} = \frac{21,39}{160} = 0,134$$

O, lo que es lo mismo, el trabajador dedica el 13,4% de su jornada en realizar su parte de la tarea.

Por otro lado, el robot ha tardado 63 minutos en gestionar 105 documentos. De media ha tardado 36 segundos por documento en realizar el resto del proceso. Se le suman los 10 segundos en concepto de system exceptions.

ID	Process	Resource	User	Status	Start Time	End Time
10...	TFG_Z_Dispatcher	WFGDESK	admin	Completed	13/06/2022 16:42:14	13/06/2022 17:45:17

Ilustración 6.15: Tiempo de ejecución del proceso

Esto supone

$$\frac{0,77 \text{ min} \times 700 \text{ vol. medio mensual} \times 1 \text{ recurso}}{60 \text{ min en hora}} = 8,94 \text{ horas} \Rightarrow \text{FTE} = \frac{8,94}{160} = 0,056$$

El robot está programado y licenciado para ser ejecutado 8 horas diarias. Como se puede apreciar en el cálculo anterior, el robot tarda casi 9 horas en realizar el trabajo de todo un mes. Este resultado deja ver que el proceso es ampliamente escalable. El robot podría gestionar volúmenes mucho mayores de trabajo.

Hablando en términos económicos, desde una perspectiva realista habría que decir que sería necesario comprar licencias y adquirir infraestructuras necesarias, que se sumarían al coste de desarrollo y mantenimiento del robot. Pero desde esa misma perspectiva también hay que mencionar que ninguna empresa se plantearía todo eso para automatizar, exclusivamente, un único proceso de 0,5104 FTE's. Habitualmente se automatizan más procesos y más complejos que justifican esta inversión.

Por esto, para realizar un cálculo económico acorde al tamaño y forma de este proyecto, solo se contará el coste estricto de desarrollo y mantenimiento del robot. Suponiendo que el robot ha tenido un coste de desarrollo de 120 horas a 25€ la hora, que se contrata una bolsa de 100 horas para mantenimiento a 40€ la hora y que seguimos con el mismo trabajador con SMI, el coste de tener este proceso automatizado sería de

$$14000 \times 0,134 + 120 \times 25 + 100 \times 40 = 1876 + 3000 + 4000 = 8876€ \text{ el 1er año}$$

$$14000 \times 0,134 + 100 \times 40 = 1876 + 4000 = 5876€ \text{ en sucesivos años}$$

El coste de implementación ya se amortiza a partir del tercer año, suponiendo un ahorro de 1269,2 € al año.

6.2.3 Proceso hiperautomatizado

En este caso, hipotéticamente, no es necesaria la intervención de un trabajador para realizar la clasificación y extracción de información necesaria. Esto podría ser así en caso de tener modelos infalibles, pero los errores ocurren y a veces las clasificaciones o la información extraída son erróneas, lo que propicia la ocurrencia de business exceptions. Estas excepciones deben ser gestionadas por un trabajador. La ventaja a largo plazo de esta solución es que, idealmente, el trabajador encargado de gestionar las excepciones puede actuar como agente externo para reentrenar y mejorar el modelo corrigiendo los errores. No obstante, esa posibilidad no se contemplará en este trabajo.

Durante la ejecución del proceso tuve que intervenir 24 veces en 105 documentos procesados para solucionar las excepciones ocurridas. De estas, 8 veces fueron por una mala clasificación; 10 por una mala extracción de información y 6 por ambas cosas.

Si se gestionan, de media 700 documentos mensuales, cabría esperar la intervención del trabajador en, aproximadamente 160 ocasiones. Cuando esto ocurre el trabajador debe eliminar el registro de la base de datos de Blue Prism y hacer la gestión completa manualmente, ya que Blue Prism no permite el acceso ni la modificación de los elementos en su base de datos.

Item Key	Priority	Status	Tags	Resource	Attempt	Created	Last Updated
5	0	Collector	Business Exception;Exception	WFGDESK	1		
3	0	Collector	Business Exception;Exception	WFGDESK	1		
4	0	Collector	Business Exception;Exception	WFGDESK	1		
2	0	Collector	Business Exception;Exception	WFGDESK	1		
1	0	Collector	Business Exception;Exception	WFGDESK	1		

- Refresh View
- Select All
- Unlock Selected Case(s)
- Mark Selected Case(s) with Exception
- Delete Worked Case(s) from Selection
- Force Retry

Ilustración 6.16: Eliminar documentos marcados como excepciones en la BBDD de Blue Prism

El trabajador tarda en eliminar el registro, meter los datos en Contoso (cuando es necesario), mover el archivo a la carpeta de SharePoint y enviar el email, de media, 2 minutos y 20 segundos.

$$\frac{2,33 \text{ min} \times 160 \text{ intervenciones} \times 1 \text{ recurso}}{60 \text{ min en hora}} = 6,21 \text{ horas} \Rightarrow \text{FTE} = \frac{6,21}{160} = 0,039$$

O, lo que es lo mismo, el trabajador dedica el 3,9% de su jornada en solucionar los errores del proceso.

Por otro lado, el robot ha tardado, de media, 52 segundos por documento en realizar el proceso. Se le suman los 10 segundos en concepto de system exceptions. Esto supone

$$\frac{1,03 \text{ min} \times 700 \text{ vol. medio mensual} \times 1 \text{ recurso}}{60 \text{ min en hora}} = 12,06 \text{ horas} \Rightarrow \text{FTE} = \frac{12,06}{160} = 0,075$$

El robot ahora tarda un poco más en completar el proceso, pero es comprensible teniendo en cuenta que ahora debe operar con un programa más, Jupyter Notebook en Chrome, y recuperar las salidas de los modelos.

Por otro lado, hablando en términos económicos y suponiendo que el robot ha tenido un coste de desarrollo de 160 horas a 25€ la hora, que se contrata una bolsa de 100 horas para mantenimiento a 50€ la hora y que seguimos con el mismo trabajador con SMI, el coste de tener este proceso automatizado sería de

$$14000 \times 0,039 + 160 \times 25 + 100 * 50 = 546 + 4000 + 5000 = 9546€ \text{ el } 1^{\text{er}} \text{ año}$$

$$14000 \times 0,039 + 100 * 50 = 546 + 5000 = 5546€ \text{ en sucesivos años}$$

El coste de implementación ya se amortiza a partir del tercer año, suponiendo un ahorro de 1599,6 € al año.

7 CONCLUSIONES

La vida es el arte de sacar conclusiones
suficientes a partir de datos
insuficientes.

-Samuel Butler-

El objetivo de este proyecto ha consistido en el desarrollo de un robot que automatice un proceso de gestión de facturas y presupuestos y el estudio de la integración de inteligencia artificial, concretamente, de técnicas de Deep Learning como las redes convolucionales y de Procesamiento del Lenguaje Natural, al robot desarrollado. Es por esto por lo que vamos a estructurar esta sección en 2 apartados: conclusiones acerca de los modelos planteados y conclusiones acerca de la implementación.


7.1 Respecto a los modelos planteados

El planteamiento original fue buscar un único modelo que hiciera el proceso de clasificación y de extracción de información. Al principio se contempló LayoutLM como opción, puesto que puede realizar ambas tareas. Sin embargo, en cuanto a clasificación de documentos, sus resultados no eran aceptables.

Posteriormente se encontró un novedoso modelo llamado **DocFormer** [36]. Este modelo prometía tener resultados extraordinarios tanto para clasificación como para extracción de información.

Results from the Paper

 Edit

 Ranked #1 on Document Image Classification on RVL-CDIP [→ Get a GitHub badge](#)





Task	Dataset	Model	Metric Name	Metric Value	Global Rank	Result	Benchmark
Document Image Classification	RVL-CDIP	DocFormer large	Accuracy	95.50%	# 6		Compare
			Parameters	536M	# 25		Compare
Document Image Classification	RVL-CDIP	DocFormerBASE	Accuracy	96.17%	# 1		Compare
			Parameters	183M	# 20		Compare

Ilustración 7.1: Métricas de DocFormer extraídas de <https://paperswithcode.com/paper/docformer-end-to-end-transformer-for-document>

El problema radica en que es tan novedoso que, a nivel práctico, apenas existe documentación de su implementación en Python o R y, en el momento de escribir este trabajo, aún se siguen añadiendo funcionalidades y desarrollando funciones.

```
Pretrain the model (Work in progress)  
A self-supervised training task can be run with  
  
# train a model from that tensor  
# docformer_ssl <- docformer_pretrain(doc_tensor, epochs=30)  
  
Train the model (work in progress)  
...followed by a supervised training task on some annotated documents...  
  
# docformer_model <- docformer_fit(doc_tensor, from_model=docformer_ssl, epochs=30)  
  
Predict with the model  
Predict with the headless model gives a document-layout embedding tensor of shape [ ,  
<max_position_embeddings>, <hidden_size> ]  
  
doc_embedding <- docformer_model(doc_tensor)
```

Ilustración 7.2: Implementación de Docformer en R en desarrollo
Fuente: <https://github.com/cregouby/docformer>

Por tanto, finalmente se decidió realizar el proceso con dos modelos separados, cada uno encargado de desarrollar una tarea particular.

7.1.1.1 Modelo de clasificación

La concatenación de modelos ha dado buenos resultados, un 91.86%, sin embargo, el coste computacional fue extraordinariamente alto. Primero comenzamos con el modelo holístico, que obtuvo una precisión del 87.13% tras más de 27 horas de entrenamiento. Una precisión alta pero insuficiente para una implementación eficaz en RPA. Para ganar ese 4.73% más de precisión se requirieron cerca de 40 horas de entrenamiento de los otros 4 modelos y de la RNA final.

Una posible mejora hubiera sido entrenar el modelo con varios datasets distintos o ampliar el número de *epochs* en el entrenamiento, pero en este último caso el coste de entrenamiento se hubiera disparado.

Por otro lado, se hizo un pequeño ensayo de implementación en Blue Prism del modelo holístico exclusivamente. Los resultados fueron inaceptables porque se generaban demasiadas business exceptions. Por este motivo era necesario mejorar la precisión.

Se ha podido comprobar que, a pesar de partir de un modelo preentrenado (VGG16 en nuestro caso) como base, resulta muy costoso entrenar una red convolucional para superar el 90% de precisión.

7.1.1.2 Modelo de extracción de información

Algo que me ha quedado claro después de horas de búsqueda de información y motores para el reconocimiento óptico de caracteres es que al OCR aún le queda un amplio recorrido por delante para ser realmente eficaz.

En un principio se consideró la opción de usar un motor OCR para extraer el texto de una imagen y luego, mediante algunos algoritmos y regex, tratar de extraer la información necesaria. Primero se estuvieron haciendo pruebas con Tesseract + magick en R, considerado uno de los mejores motores OCR actuales. Sobre documentos manuscritos los resultados eran, sencillamente, inútiles. Por tanto, restringí el ámbito de actuación a documentos escritos a máquina. Sobre una imagen de alta resolución con fondo perfectamente claro, los resultados eran buenos

Atlas
31 Cunningham Court
London, W91AE
United Kingdom

Billed To
Makera K.
Stockhorn Capital
Wimpole Street
London, W91AE
United Kingdom

Date of Issue
18/09/2018

Invoice number
qj 2219

Amount Due
£48,100.00

Due Date
18/09/2018

Purchase Order
qj 2219

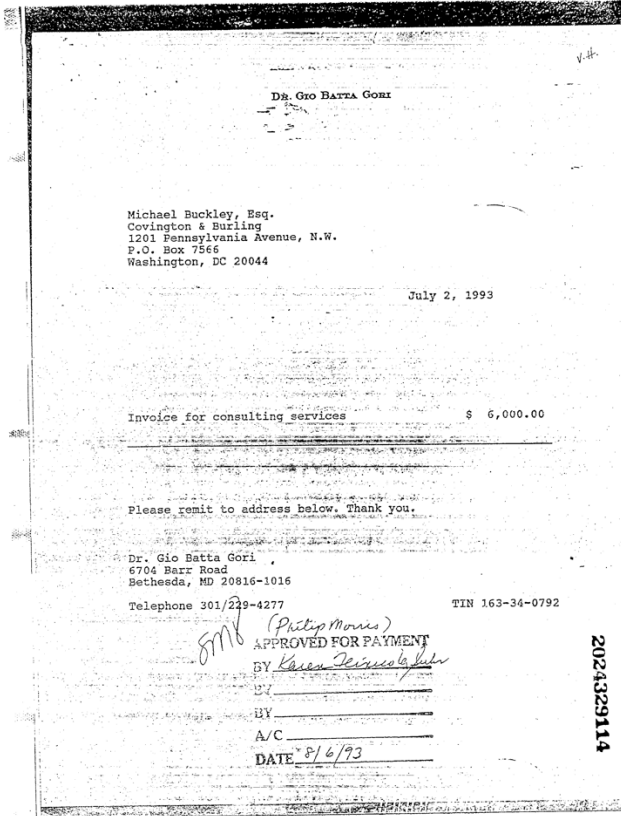
Item	Quantity	Cost	Price
Web Services	100	£ 200.00	£ 20,000.00
Development Services	80	£ 300.00	£ 24,000.00
		Subtotal	£ 44,000.00
		Tax: (20%)	£ 8,500.00
		Discount: (10%)	-£ 4,400.00
		Amount Due	£ 48,100.00

Terms and Conditions
Please make the payment by the due date. If not, we will add 0.2% each month to the total price.

info@helpdojo.com +1 (800) 123 4567 helpdojo.com
Company Reg. No. 2221qj2812jq

```
> ##### Tesseract + magick for OCR
> eng = tesseract("eng")
> text = tesseract::ocr("fact01.png", engine = eng)
> results = tesseract::ocr_data("fact01.png", engine = eng)
> cat(text)
Atlas
31 Cunningham Court
atlas a
United Kingdom
Billed To Date of Issue Invoice number Amount Due
Makera K. 18/09/2018 2219
"nt " £48,100.00
en Due Date Purchase Order
wimpole Street 18/09/2018 j2219
London, w91AE
United Kingdom
Item Quantity Cost Price
Web Services 100 £200.00 £ 20,000.00
Development Services 80 £ 300.00 £24,000.00
Subtotal £ 44,000.00
Tax: (20%) £ 8,500.00
Discount: (10%) -£4,400.00
Amount Due £ 48,100.00
Terms and Conditions
Please make the payment by the due date. If not, we will
info@helpdojo.com +1 (800) 1234567 _helpdojo.com
Company Reg. No. 2221qj2812jq
> results %>% head(20)
# A tibble: 20 x 3
  word confidence bbox
<chr> <dbl> <chr>
1 Atlas 96.1 847,67,887,81
2 31 93.8 716,98,732,111
3 Cunningham 93.8 740,97,839,115
4 Court 96.8 844,98,886,111
5 atlas 94.8 243,101,400,157
6 a 52.4 764,127,887,143
7 United 95.7 764,157,812,172
8 Kingdom 95.7 818,157,887,175
9 Billed 96.7 72,268,120,282
10 To 97.0 125,269,144,282
11 Date 96.9 272,269,310,282
12 of 96.9 315,268,332,282
13 Issue 96.7 338,269,381,282
14 Invoice 95.6 450,269,509,282
15 number 96.6 514,268,578,282
16 Amount 96.1 785,269,849,282
17 Due 96.7 856,269,886,282
18 Makera 92.2 74,302,127,314
19 K. 81.3 135,302,149,314
20 18/09/2018 96.5 273,300,366,314
>
```

Sin embargo, sobre imágenes con ruido o un fondo no perfectamente claro, los resultados no eran adecuados.



```
> eng = tesseract("eng")
> text = tesseract::ocr("fact03.jpg", engine = eng)
> results = tesseract::ocr_data("fact03.jpg", engine = eng)
> cat(text)
| a 7 Serie et RIT 4
a & i
i . : Sah GETIRE'D ' t
; - Di, Gro Barra Gorr |
Michael Buckley, Esq. i |
Covington & Burling a
1201 Pennsylvania Avenue, N.W. g
B.O, Box 7866 5
Washington, DC 20044 4
| ~ = ba duly 2, 1993 Es
| |
i! & ater f
I ie :
I invoice for consulting services " $ 6,000.00
a8 el wee age od '
Joh S02 ey ed oe ech acinar
| a Reis Se ant igi ect egati ar
i Please remit to address below. Thank you,
| +o Ede. Gio Batta Gori, a :
| 6704 Barr Road -
Bethesda, MD 20816-1026
Telephone 301/239-4277 PIN 163-34-0792
AY (Partie Mouv )
gf \b sodpovidD FOR PAYMED >
e sace by S
a BY tae estes Ia bed RG
atl eee = Sa
Be a &
estate BY o &
BE
| Ase _____ - BG
i = 8/6/93 Z|
\ - parele/4/73 ~ e
> results %>% head(20)
# A tibble: 20 x 3
  word confidence bbox
  <chr> <dbl> <chr>
1 | 65.6 27,55,28,62
2 a 0 76,45,147,65
3 7 26.1 285,50,287,52
4 Serie 10.3 329,50,382,60
5 et 7.97 389,42,436,62
6 RIT 48.9 473,48,522,56
7 4 80.9 724,44,731,61
8 a 15.9 29,72,38,85
9 & 24.1 678,67,687,80
10 i 28.5 725,63,731,81
11 i 51.9 35,91,38,97
12 . 0 93,98,95,100
13 : 81.8 183,89,185,91
14 Sah 3.76 313,91,341,93
15 GETIRE'D 0 405,88,452,91
16 æ" 42.6 666,73,671,85
17 t 0 724,74,727,90
18 ; 15.9 30,131,31,136
19 - 0 170,116,182,122
20 Di, 58.9 315,119,338,132
> |
```

Por este motivo se desechó este motor. Mas tarde se encontró PaddleOCR, un motor muy reciente y *entrenable* que, además de reconocer texto, permitía la extracción inteligente de información. Así que se adoptó un nuevo enfoque para esta parte del trabajo.

Un 91.39% de precisión en la extracción de información es un resultado extraordinario, dada la susceptibilidad del OCR al ruido de las imágenes. Por otro lado, el modelo ha sido entrenado en un único conjunto de datos con "tan solo" 1000 casos parecidos entre sí. Es probable que los resultados hubieran sido muy distintos si el modelo se hubiera entrenado con casos más variados y con una mayor cantidad.

7.2 Respeto a la implementación

El resultado **de implementar RPA en un proceso sin automatizar** ha sido el esperado: se ha liberado al trabajador de casi la totalidad del tiempo dedicado al proceso, por lo que se podría reubicar en otros proyectos o tareas. El robot implementado tiene capacidad para gestionar un volumen mucho mayor de trabajo, por tanto, la apertura de la nueva sucursal no supondría problema alguno. El coste de implementación es algo mayor al coste actual del proceso, pero el retorno de la inversión es breve.

No obstante, la falta de perfección en los modelos de clasificación y extracción de información evitan que, aunque se **automaticen tareas cognitivas al proceso**, se pueda liberar por completo al trabajador. Es cierto que, en comparación entre el caso sin IA y el caso con IA, el segundo es mucho más eficiente tanto a nivel económico como de liberación de recursos. El coste de desarrollo e implementación del caso con IA es mayor, pero el retorno de la inversión es prácticamente el mismo que en el caso sin IA y, a largo plazo, el beneficio es mayor.

Cabe destacar que, con mayores recursos computacionales o sin el plazo de entrega propio de un TFG, hubiera merecido la pena aumentar el entrenamiento de los modelos usados. Mejorar la precisión hubiera supuesto una mejoría en el ROI, la práctica liberación del trabajador o, asumiendo quizás el coste del error, la liberación completa del trabajador.

En cuanto al desarrollo en Blue Prism, a pesar de la falta de acceso a los módulos especializados para IA, se ha podido implementar de forma alternativa (aunque no ideal) la solución. No se ha podido comprobar, por esta misma falta de acceso, pero es altamente probable que los tiempos de ejecución del robot sean aún más reducidos si en vez de tener que realizar la clasificación a través del acceso por interfaz de usuario se realizara directamente por API. Además, también se reducirían los errores y excepciones, ya que al conectar por API se eliminan los casos en los que no extrae la información del resultado de los modelos del sitio correcto (aunque el modelo haya dado la respuesta acertada), o no encuentra en la UI el elemento que está buscando como referencia.

Como conclusión general extraemos la idea de que la inteligencia artificial tiene mucho potencial en el campo de RPA, permitiendo automatizar procesos end to end que, de otra forma, no podrían ser completamente automatizables o mejorando procesos ya automatizados.

Al principio de este trabajo definimos RPA como una amalgama de tecnologías integradas. Aunque aún queda mucho camino por delante y las técnicas de machine learning deben mejorar, es indudable que la inteligencia artificial pasará a formar parte regular de esta integración y que el futuro de RPA está en la hiperautomatización.

7.3 Comentarios personales

Quisiera destacar la satisfacción que ha supuesto trabajar en este proyecto. El desarrollo de este TFG me ha permitido y obligado a ir aprendiendo, sobre la marcha, los mecanismos de automatización de procesos a través de una herramienta muy extendida y valorada como es Blue Prism. Poder experimentar el éxito de ver cómo lo que has creado tras tantas horas de investigación, lecturas, videos, ensayo y error, funciona es una sensación única. Además, me ha permitido introducirme en el ámbito del OCR, que desconocía por completo antes de este trabajo, y que resulta ser una rama con proyectos de lo más desafiantes.

En cuanto a redes convolucionales, este trabajo me ha demostrado la profundidad y complejidad que este campo puede llegar a tener. Resulta tan fascinante como abrumadora la cantidad de estrategias que se pueden adoptar para abordar un mismo problema, y la dificultad que subyace tras cada una de ellas.

Este trabajo, sin duda alguna, ha sido una experiencia enriquecedora de aprendizaje y crecimiento.

8 BIBLIOGRAFÍA

- [1] S. Gundín, “RPA: Robotic Process Automation - Qué es y cómo nos ayuda • AuraQuantic.” <https://www.auraquantic.com/es/rpa--robotic--process--automation--que--es/> (accessed Feb. 12, 2022).
- [2] J. Welsh, “What the History of RPA Technology Says About its Future – GPA,” Mar. 28, 2019. <https://globalpayrollassociation.com/blogs/technology/what-the-history-of-rpa-technology-says-about-its-future> (accessed Feb. 14, 2022).
- [3] Google, “RPA - Explorar - Google Trends,” 2022. <https://trends.google.es/trends/explore?date=2015-01-01%202022-02-14&q=RPA> (accessed Feb. 14, 2022).
- [4] A. Haleem, M. Javaid, R. P. Singh, S. Rab, and R. Suman, “Hyperautomation for the enhancement of automation in industries,” *Sensors International*, vol. 2, p. 100124, Jan. 2021, doi: 10.1016/J.SINTL.2021.100124.
- [5] Gartner, “Gartner Top 10 Strategic Technology Trends For 2020,” 2019. <https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020> (accessed Feb. 14, 2022).
- [6] Google, “hyperautomation - Explorar - Google Trends,” 2022. <https://trends.google.es/trends/explore?date=today%205-y&q=hyperautomation> (accessed Feb. 14, 2022).
- [7] I. Gavilán, *Robots en la sombra RPA, robots conversacionales y otras formas de automatización cognitiva*. Madrid: Anaya Multimedia, 2021.
- [8] pfsTECH, “Maximiza la eficiencia de tu empresa con RPA • pfsTECH.” <https://pfstech.es/ebook-maximiza-la-eficiencia-de-tu-empresa-con-robotic-process-automation/> (accessed Feb. 12, 2022).
- [9] K. Soeny, G. Pandey, U. Gupta, A. Trivedi, M. Gupta, and G. Agarwal, “Attended robotic process automation of prescriptions’ digitization,” *Smart Health*, vol. 20, p. 100189, Apr. 2021, doi: 10.1016/J.SMHL.2021.100189.
- [10] A. Haleem, M. Javaid, R. P. Singh, S. Rab, and R. Suman, “Hyperautomation for the enhancement of automation in industries,” *Sensors International*, vol. 2, p. 100124, Jan. 2021, doi: 10.1016/J.SINTL.2021.100124.
- [11] W. A. Ansari, P. Diya, S. Patil, and S. Patil, “A Review on Robotic Process Automation - The Future of Business Organizations,” *SSRN Electronic Journal*, Apr. 2019, doi: 10.2139/ssrn.3372171.

- [12] R. T. Yarlagadda, "The RPA and AI Automation," 2018. [Online]. Available: www.ijcrt.orgwww.ijcrt.org365
- [13] "The Difference between Robotic Process Automation and Artificial Intelligence | by CFB Bots | Medium." <https://cfb-bots.medium.com/the-difference-between-robotic-process-automation-and-artificial-intelligence-4a71b4834788> (accessed Mar. 07, 2022).
- [14] T. Nguyen, "Intelligent Automation — A symbiotic relationship between RPA and Data Science | by Trung Nguyen | Towards Data Science," Apr. 06, 2020. <https://towardsdatascience.com/intelligent-automation-a-symbiotic-relationship-between-rpa-and-data-science-632af34555e3> (accessed Mar. 07, 2022).
- [15] F. Sancho Caparrini, "Breve Historia de la Inteligencia Artificial," Sep. 22, 2021. <http://www.cs.us.es/~fsancho/?e=221> (accessed Jun. 09, 2022).
- [16] www.elternativa.com, "Historia de la inteligencia artificial: origen y auge de la IA." <https://www.elternativa.com/blog-elternativa/historia-inteligencia-artificial> (accessed Jun. 09, 2022).
- [17] R. Florez López, *Las Redes Neuronales Artificiales : fundamentos teóricos y aplicaciones prácticas* /. Oleiros (La Coruña): Netbiblo, 2007.
- [18] R. López, "¿Qué es y cómo funciona «Deep Learning»?," May 07, 2014. <https://rubenlopezg.wordpress.com/2014/05/07/que-es-y-como-funciona-deep-learning/> (accessed Jun. 09, 2022).
- [19] M. H. Ortí, D. Toro, and P. A. Guinot Martínez, "Introducción a las redes neuronales aplicadas al control industrial," 2013.
- [20] F. J. Palacios Burgos, "Redes Neuronales con GNU/Linux," 2003. https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html (accessed Jun. 10, 2022).
- [21] P. Jurado Leal and M. J. Jiménez, "CONCEPTOS BÁSICOS SOBRE REDES NEURONALES," 2013. <http://grupo.us.es/gtocom/pid/pid10/RedesNeuronales.htm> (accessed Jun. 10, 2022).
- [22] "Características de las redes neuronales." <https://thales.cica.es/rd/Recursos/rd98/TecInfo/07/capitulo3.html> (accessed Jun. 10, 2022).
- [23] J. Torres i Viñals, "Deep Learning – Introducción práctica con Keras," 2018. https://torres.ai/deep-learning-inteligencia-artificial-keras/#Introduccion_a_las_redes_neuronales_convolucionales (accessed Jun. 10, 2022).
- [24] M. Sotaquirá, "¿Qué son las Redes Convolucionales? | Codificando Bits," Mar. 23, 2019. <https://www.codificandobits.com/blog/redes-convolucionales-introduccion/> (accessed Jun. 21, 2022).

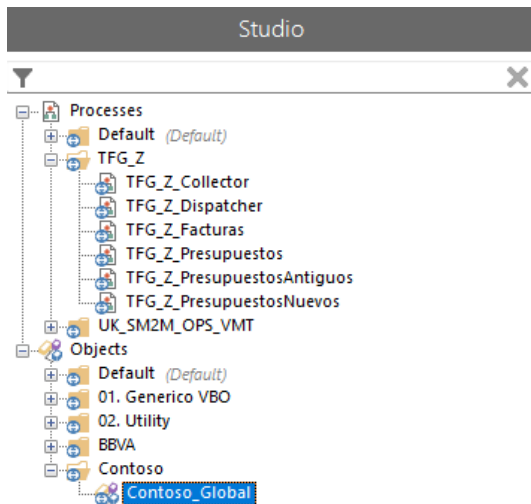
- [25] J. I. Bagnato, "Convolutional Neural Networks: La Teoría explicada en Español," Nov. 29, 2018. <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/> (accessed Jun. 11, 2022).
- [26] A. C. Vásquez, H. V. Huerta, J. P. Quispe, and A. M. Huayna, "Procesamiento de lenguaje natural," *Revista de investigación de Sistemas e Informática*, vol. 6, no. 2, pp. 45–54, Dec. 2009, Accessed: Jun. 12, 2022. [Online]. Available: <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/5923>
- [27] M. Hernández and J. M. Gómez, "Aplicaciones de Procesamiento de Lenguaje Natural," *Revista Politécnica*, vol. 32, pp. 87–96, Jul. 2013.
- [28] C. J. Fernández Sánchez and V. Sandonís Consuegra, "Reconocimiento óptico de caracteres (ocr)," *Universidad Carlo*, vol. 3, 2008.
- [29] Gartner, "Magic Quadrant for Robotic Process Automation 2021," Sep. 29, 2021. <https://www.gartner.com/doc/reprints?id=1-26QWQ67B&ct=210709&st=sb> (accessed Mar. 19, 2022).
- [30] A. W. Harley, A. Ufkes, and K. G. Derpanis, "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval," 2015. [Online]. Available: <http://scs.ryerson.ca/>
- [31] I. Chatterjee, "RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing)," Feb. 02, 2020. <https://medium.com/analytics-vidhya/rvl-cdip-ryerson-vision-lab-complex-document-information-processing-aa30b00a2b1e> (accessed Jun. 12, 2022).
- [32] A. Das, S. Roy, U. Bhattacharya, and S. K. Parui, "Document Image Classification with Intra-Domain Transfer Learning and Stacked Generalization of Deep Convolutional Neural Networks," Jan. 2018.
- [33] M. Ramina, "Beginner's guide to Extract Receipt's Information using Deep Learning (OCR & NLP Model)," Feb. 12, 2021. <https://medium.com/one9-tech/information-extraction-receipt-ocr-scan-deep-learning-1e68ce5a9ae7> (accessed Jun. 12, 2022).
- [34] Y. Du *et al.*, "PP-OCR: A Practical Ultra Lightweight OCR System," Sep. 2020, doi: 10.48550/arxiv.2009.09941.
- [35] Z. Huang *et al.*, "ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction," Mar. 2021, doi: 10.1109/ICDAR.2019.00244.
- [36] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, and R. Manmatha, "DocFormer: End-to-End Transformer for Document Understanding," Jun. 2021.

9 ANEXO

En este anexo se muestra el desarrollo en la creación de objetos y procesos a través de Blue Prism, así como su funcionamiento general. También se podrá encontrar el PDD, que recordemos que era el documento en el que se analiza el proceso que se quiere automatizar y el SDD, que era el documento en el que se detalla la solución propuesta.

9.1 Blue Prism

El objetivo de este apartado es mostrar brevemente y en líneas generales la forma de trabajar y automatizar procesos con Blue Prism para ilustrar el trabajo realizado.



En primer lugar, en Blue Prism trabajamos con objetos y con procesos.

Los objetos son instrumentos para interactuar con los distintos programas del entorno digital. En esencia, son la base de la construcción de procesos.

Los procesos son flujos de trabajo que siguen una determinada lógica secuencial. Se construyen a partir de objetos y variables. Los procesos se dividen en subprocesos y cada subproceso, a su vez, se divide en páginas

Ilustración 9.1: Ventana de procesos y objetos en Blue Prism

Al igual que sucedería en programación orientada a objetos, se podría decir que los objetos en Blue Prism se componen, aproximadamente, de atributos y métodos. El objeto `Contoso_Global`, que usamos para acceder y trabajar en Contoso, se construyó con métodos para inicializar y cerrar la aplicación, fijar la aplicación en un plano de trabajo, acceder a distintas opciones, extraer datos e insertar datos. Los atributos serían las variables empleadas.

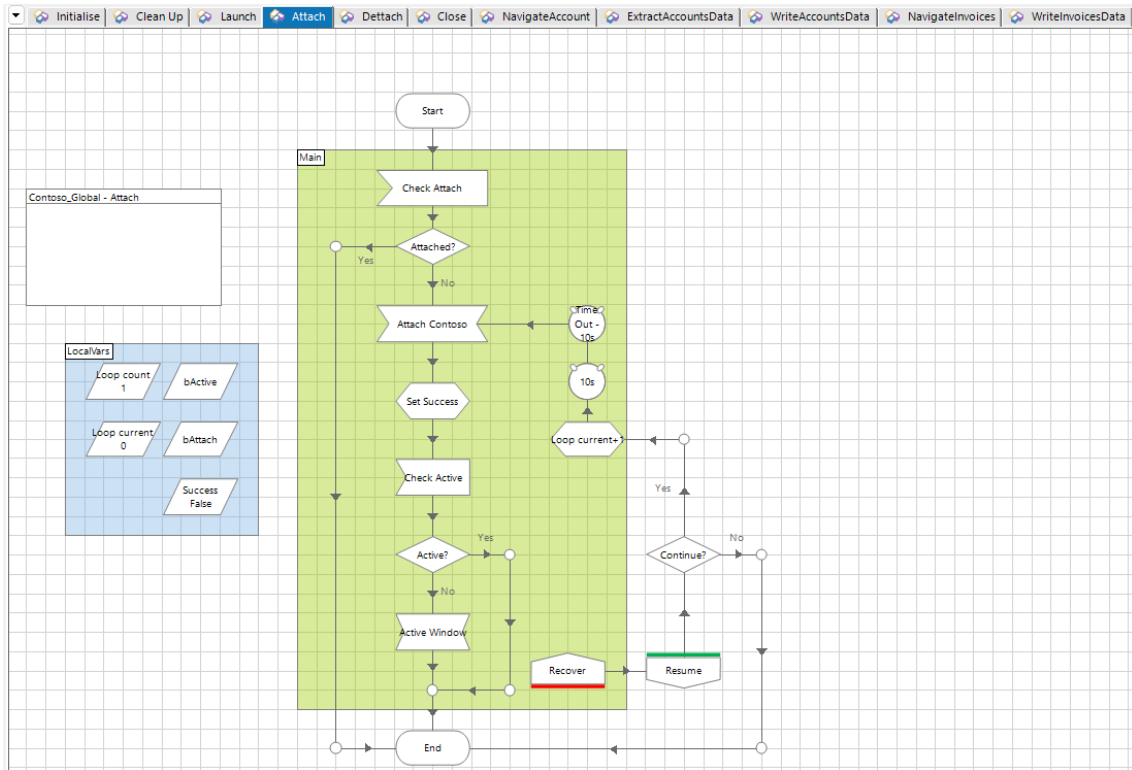


Ilustración 9.2: Método attach del objeto Contoso_Global.

9.1.1 Construcción de objetos en Blue Prism

Para construir un objeto ejecutaremos la opción *Application Modeller Wizard* que nos guiará paso a paso en la creación. En primer lugar, tras haberle dado un nombre adecuado a nuestro objeto, tendremos que determinar qué tipo de aplicación es la que queremos detectar. En nuestro caso es una *Windows Application*.

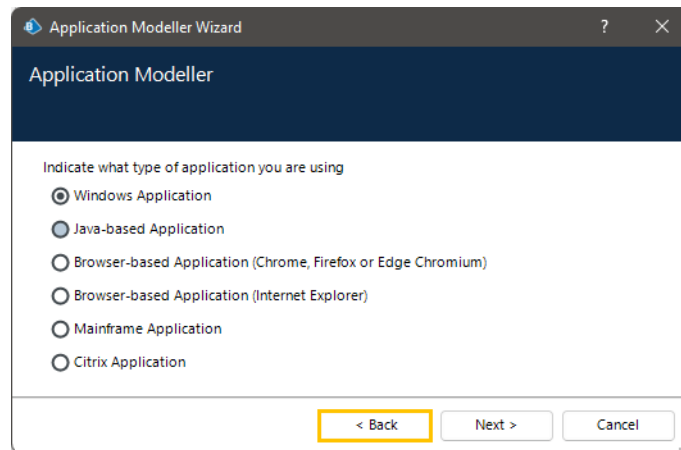


Ilustración 9.3: Ventana para elegir el tipo de aplicación en el Modeller

A continuación, seleccionamos el modo de ejecución de la aplicación: se lanza cuando se necesita, o siempre está en funcionamiento y solo necesitamos adherirnos a una instancia. En nuestro caso la lanzaremos cuando se requiera.

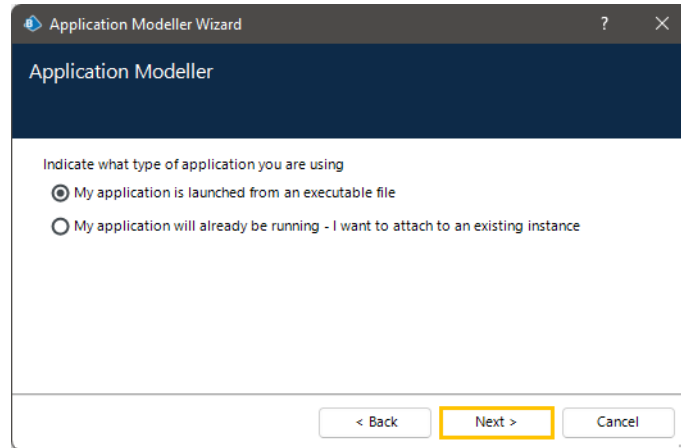


Ilustración 9.4: Ventana para elegir el modo de ejecución en el Modeller

Introducimos la ruta de acceso a la aplicación.

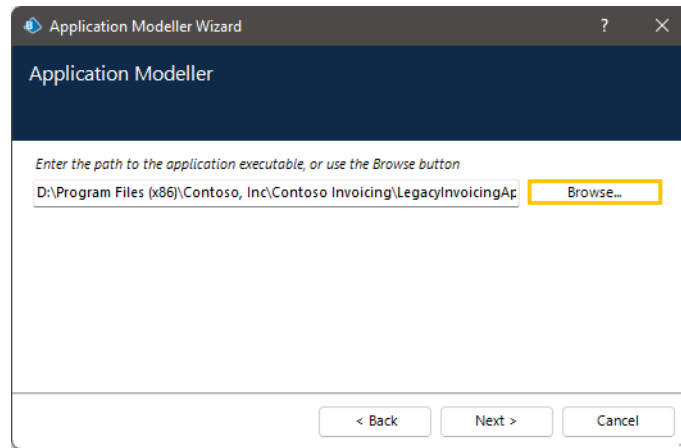


Ilustración 9.5: Ventana para insertar la ruta de acceso a la aplicación en Modeller

A continuación vienen una serie de pasos sin mayor importancia que no vamos a reflejar.

Una vez creado el objeto debemos enseñar a Blue Prism a reconocer los elementos del UI de Contoso que nos interesan para poder navegar por la aplicación, insertar datos o extraerlos. Para ello usaremos la opción *Application Modeller*. Esta opción nos ofrece una serie de herramientas para identificar unívocamente cada elemento del UI del programa.

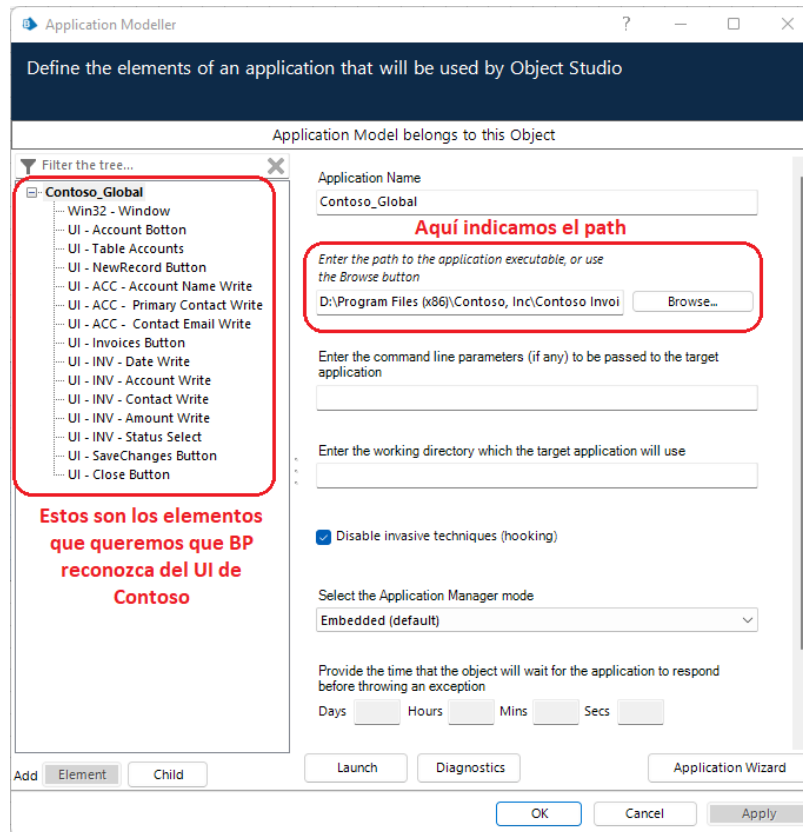


Ilustración 9.6: Application Modeller para Contoso

Para añadir un nuevo elemento, por ejemplo, la tabla de proveedores de la sección Accounts de Contoso, debemos hacer click sobre la opción *Add Element*, abrir el programa y seleccionar el elemento que nos interesa. Esto nos creará un elemento con multitud de opciones de identificación. Siempre hay que tratar de seleccionar las opciones mínimas imprescindibles para identificar nuestro elemento. Si se seleccionan muy pocas opciones se corre el riesgo de que exista más de un elemento identificable con esas opciones, pero si añadimos demasiadas puede ocurrir que un mínimo cambio impida el correcto reconocimiento de nuestro elemento y desemboque en *business exceptions* en nuestro proceso.

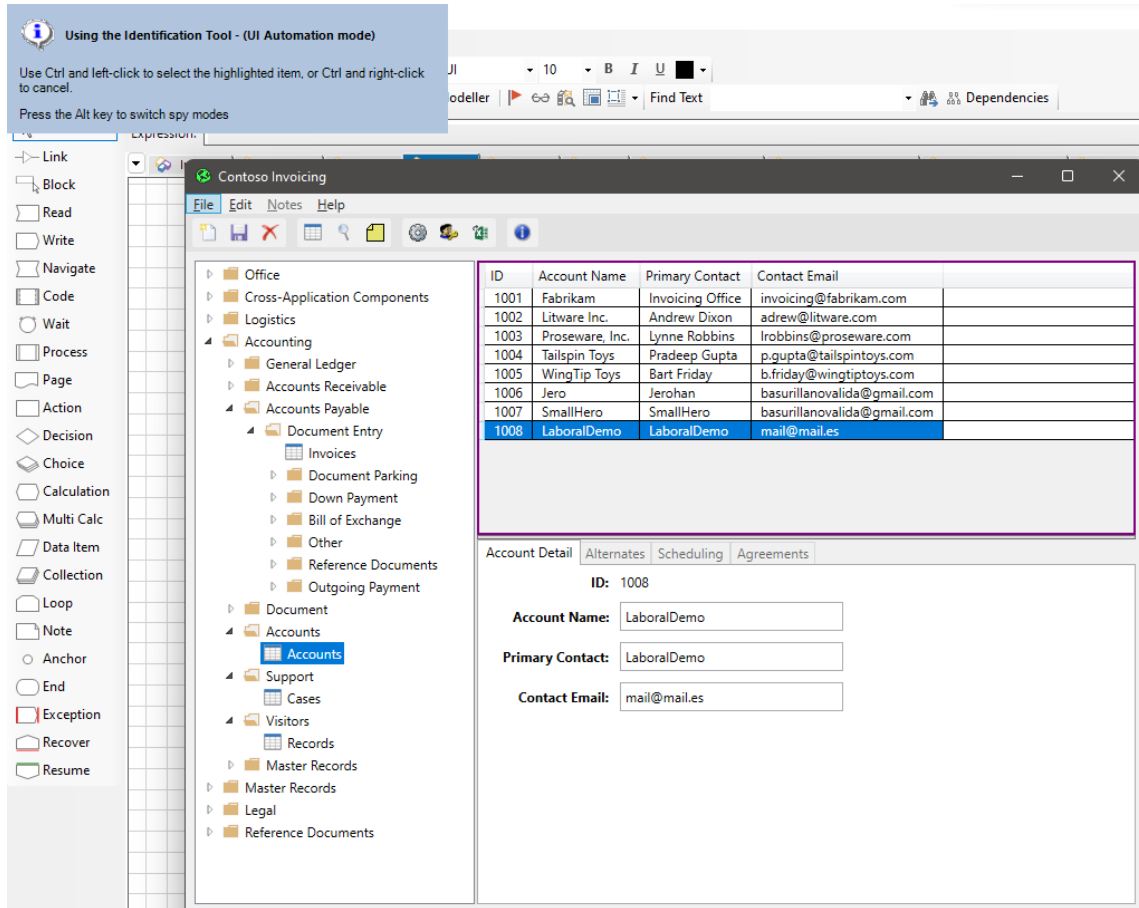


Ilustración 9.7: Detección de elementos en Modeller

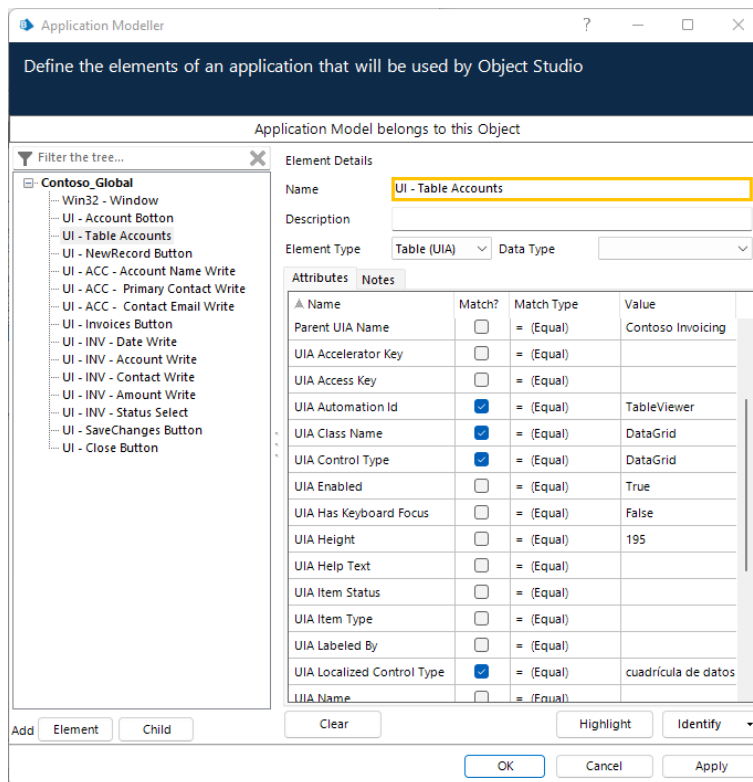


Ilustración 9.8: Configuración de opciones de identificación de elementos en Modeller

Una vez contruidos los objetos necesarios (Contoso_Global en este caso), se empieza a desarrollar el flujo del proceso.

9.1.2 Desarrollo de flujos de procesos

Una buena práctica a la hora de automatizar procesos consiste en crear la siguiente estructura:

- **Subproceso colector:** Es un subproceso encargado, exclusivamente, de obtener los datos con los que se va a trabajar e insertarlos en la base de datos interna de Blue Prism.
- **Proceso orquestador:** Es un proceso de alto nivel encargado de la lógica general del proceso. Es decir, es el proceso que irá llamando a los distintos subprocesos requeridos en el tiempo y orden adecuados.
- **Subprocesos:** Son las distintas tareas en las que se puede dividir un proceso completo.

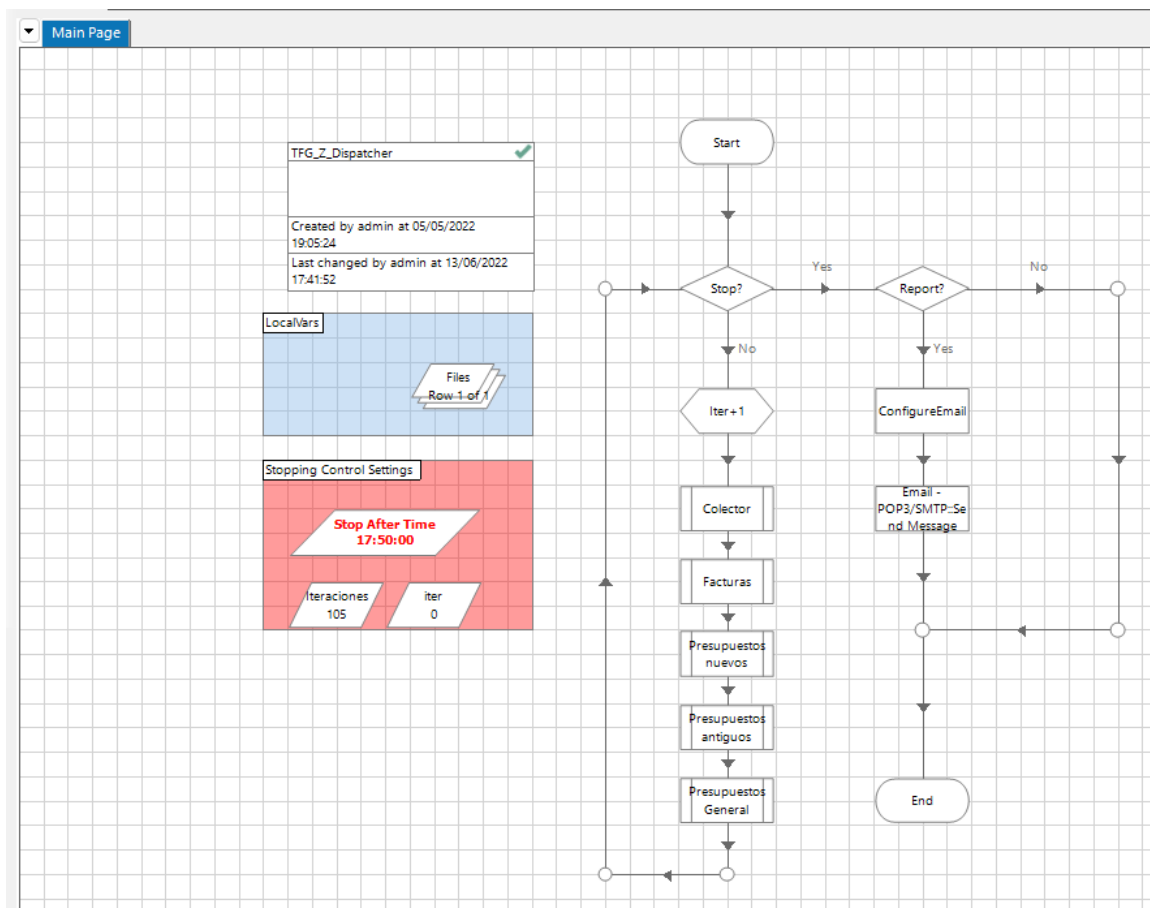


Ilustración 9.9: Vista general del proceso orquestador

Como se aprecia en la imagen, el desarrollo de flujos de procesos y la construcción de objetos en Blue Prism es muy visual. Consiste en elaborar un flujo de trabajo en el que se van

llamando, en la secuencia indicada, a los subprocesos, objetos o acciones necesarias. Para construir este flujo de trabajo usamos las herramientas de flujo que nos facilita Blue Prism.

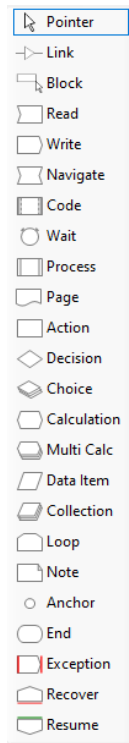


Ilustración 9.10: Herramientas de flujo de Blue Prism

Vamos a ilustrar el procedimiento de creación de un proceso a través del método `WriteAccountsData` del objeto `Contoso_Global`. Aunque usemos un método para el ejemplo (por ser mucho más breve y sencillo) el sistema de construcción de un proceso es idéntico.

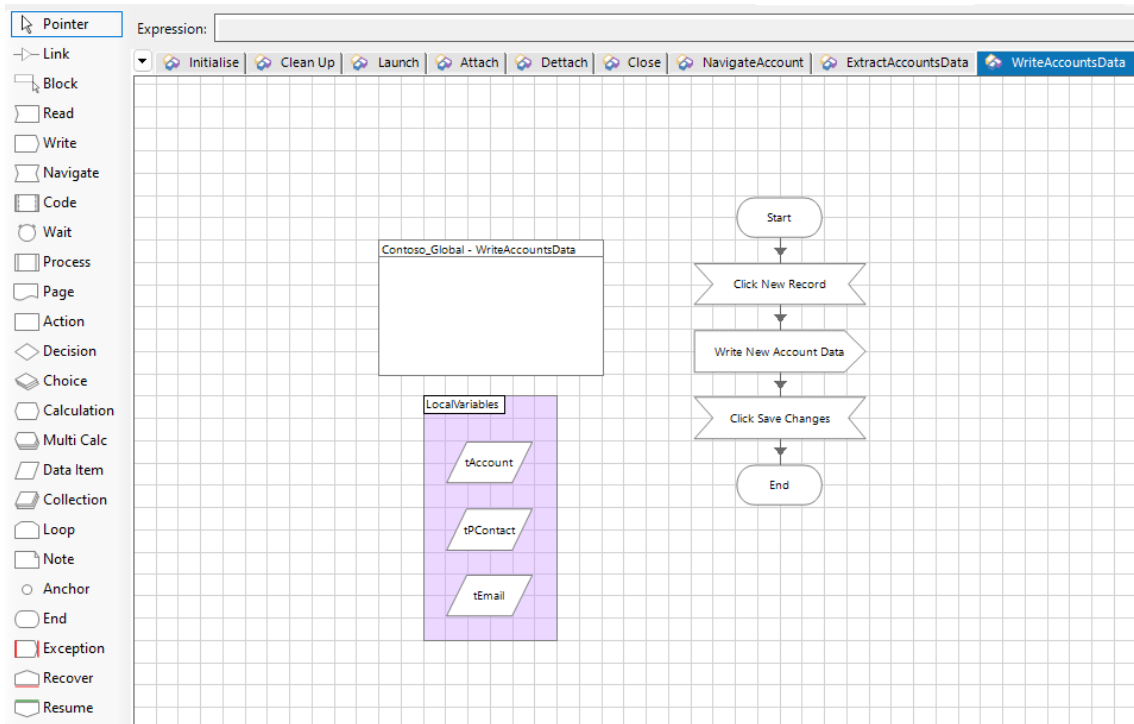


Ilustración 9.11: Método `WriteAccountsData` del objeto `Contoso_Global`

La imagen anterior muestra los pasos del método y las variables usadas. El método consiste en hacer click en el botón *New Record* de la aplicación Contoso, escribir los datos del nuevo proveedor y hacer click en *guardar cambios*.

Si miramos las propiedades de *Start*, se aprecia que este método recibe como parámetros de entrada 3 datos: *ContosoAccounts.Supplier*, *ContosoAccounts.Email* y *ContosoAccounts.Contact*, que son guardados en las variables *tAccount*, *tEmail* y *tPContact* respectivamente.

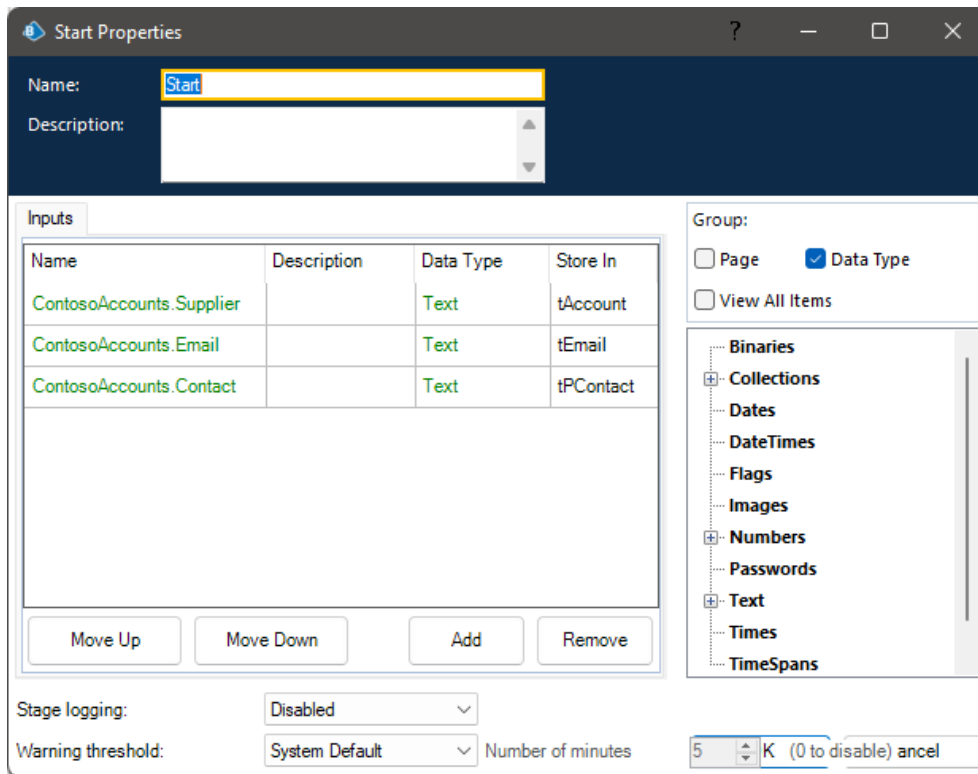


Ilustración 9.12: Propiedades de Start en el método WriteAccountsData

Estos datos los recibe desde el subproceso que llama al método de este objeto.

El primer paso de este método es hacer click en el botón *New Record* de Contoso. Para conseguir este propósito se inserta un elemento de flujo del tipo *Navigate*, seleccionamos el elemento de UI de Contoso al que queremos navegar (que ya habíamos reconocido en la creación del objeto *Contoso_Global*), y seleccionamos la acción que queremos ejecutar (un click en este caso). Adicionalmente se ha configurado un breve tiempo de espera (0.25 segundos) entre cada paso para evitar que Blue Prism ejecute las acciones más rápido de lo que la aplicación Contoso puede procesar, evitando posibles problemas y errores.

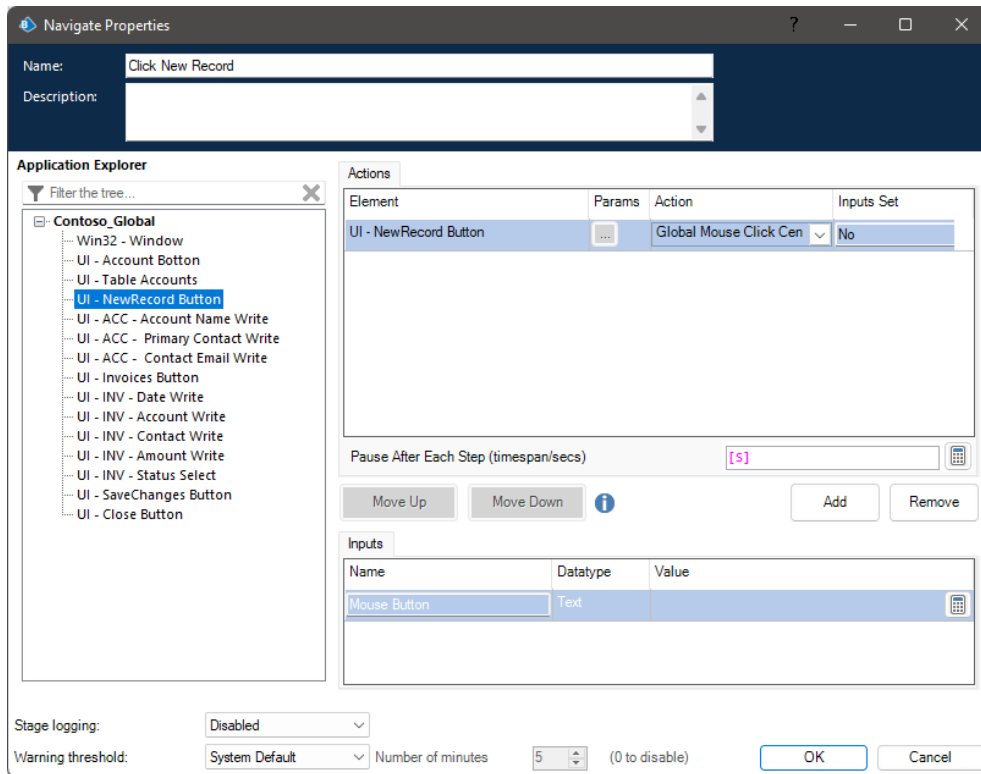


Ilustración 9.13: Configuración para hacer click en New Record

A continuación introducimos los datos del proveedor en la tabla de proveedores de Contoso. Para ello insertamos un elemento de flujo de tipo *write*, seleccionamos los elementos UI donde queremos insertar los datos y las variables que los contienen.

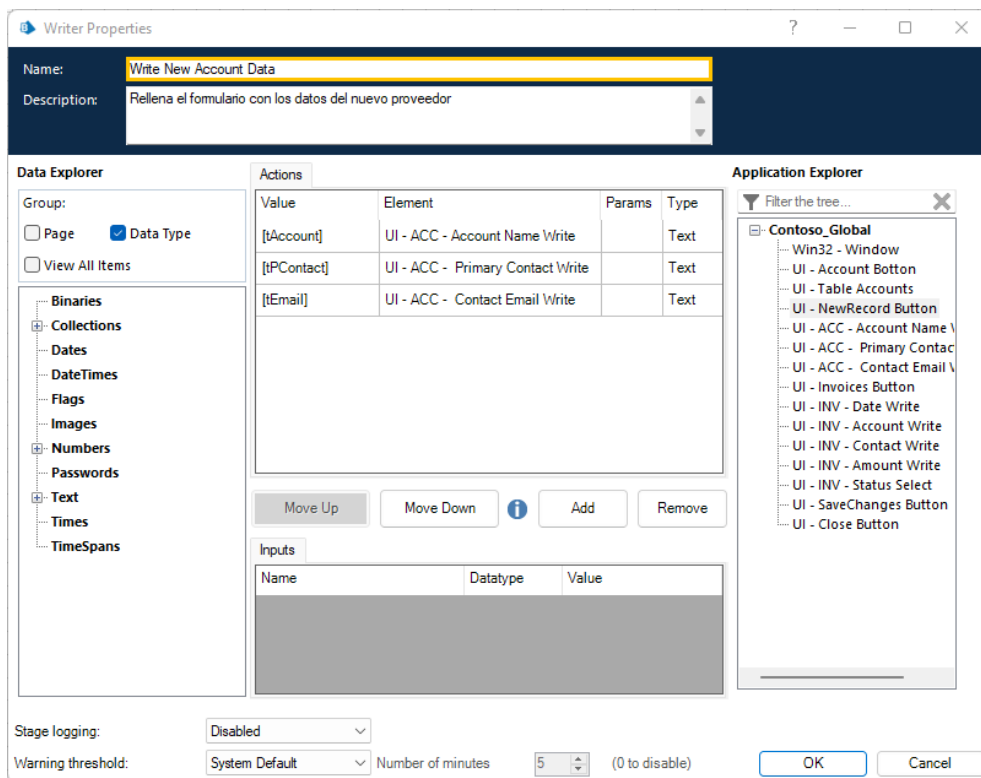


Ilustración 9.14: Configuración para introducir datos en la tabla de Proveedores de Contoso

Por último, guardamos los cambios haciendo click en el botón *Save Changes* de Contoso, cuyo procedimiento es idéntico al del primer paso.

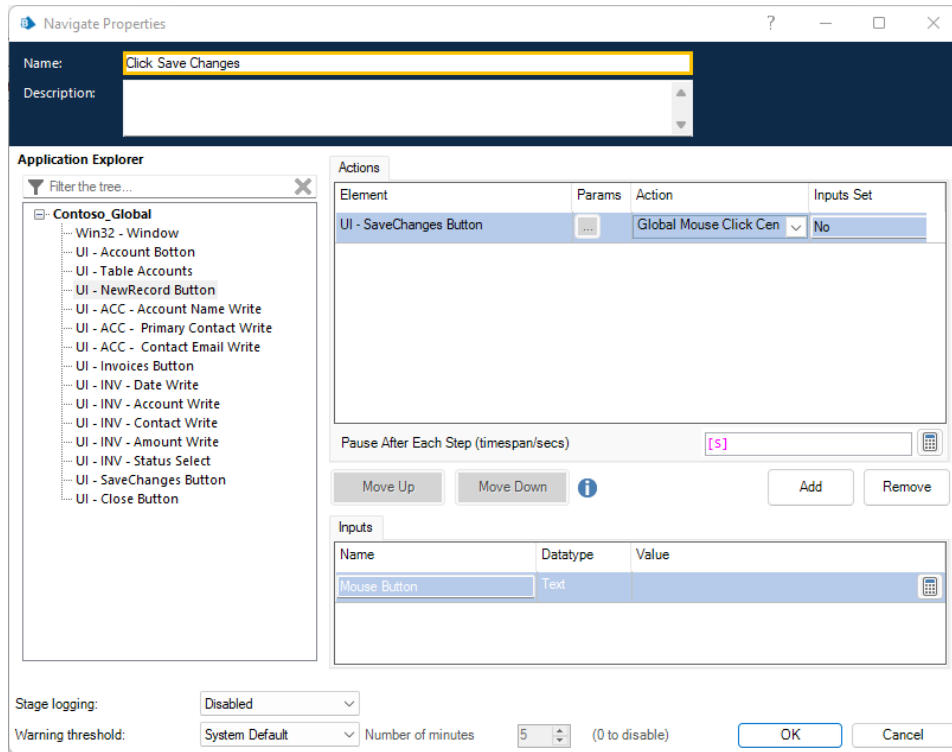


Ilustración 9.15: Configuración para hacer click en *Save Changes*

Si revisamos las propiedades de *End* podemos ver que este método no ofrece información de salida, o sea, no devuelve ningún dato al subproceso que ha hecho la llamada.

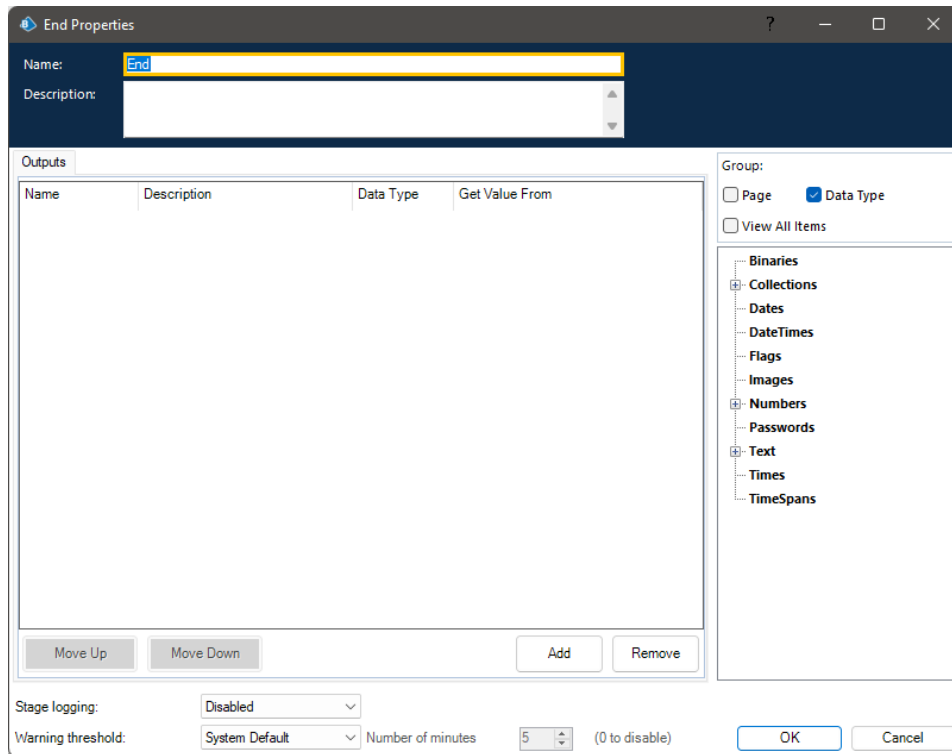


Ilustración 9.16: Propiedades de *End* del método *WriteAccountsData*

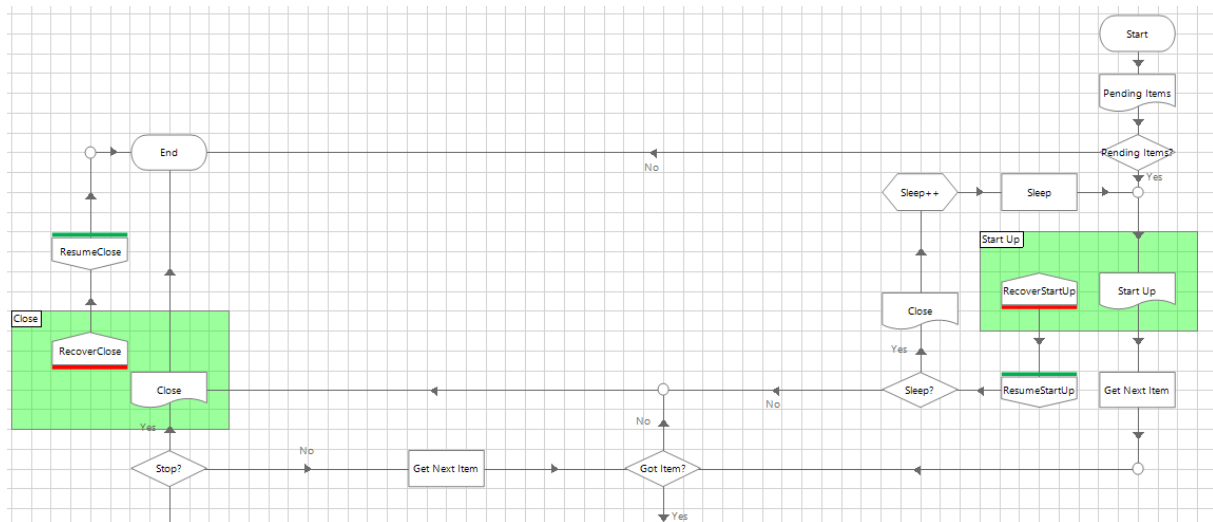


Ilustración 9.18: Primera tarea del subprocesso Facturas

Los elementos de flujo de tipo *Recover-Resume* nos permiten capturar posibles errores o excepciones en la ejecución del proceso. Es importante introducir este tipo de elementos de flujo para que el proceso no se paralice ante un error.

2. Una vez extraído un elemento de la cola se realiza la siguiente secuencia
 - a. Ir a la pestaña de cuentas
 - b. Extraer los datos de las cuentas existentes
 - c. Renombrar las columnas de los datos extraídos para usarlos de entrada para el método *WriteAccountsData*.
 - d. Extraer los nombres de proveedores de los datos extraídos de Contoso para compararlos con el nombre del proveedor del elemento extraído de la cola.
 - e. Verificar si el proveedor ya existe: en caso negativo se registran los datos nuevos.
 - f. Navegar a la pestaña de facturación y escribir los datos de la factura
 - g. Mover los documentos a la carpeta de SharePoint
 - h. Enviar el email correspondiente
 - i. Escribir los datos en el reporte que se enviará más tarde.

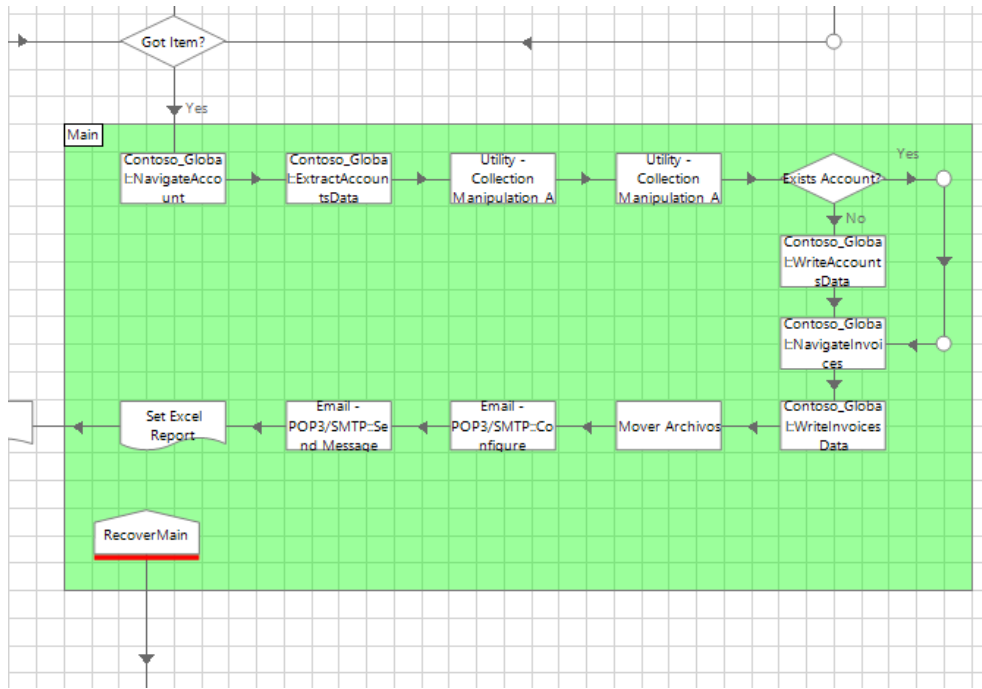


Ilustración 9.19: Segunda tarea del subproceso Facturas

- Si no ha habido errores en los pasos anteriores, el documento se marca como completado y se itera al siguiente elemento en cola. Si ha habido algún error, el documento se marca como excepción. El subproceso se repite hasta que dejan de existir elementos en cola. Entonces se pasa al siguiente subproceso.

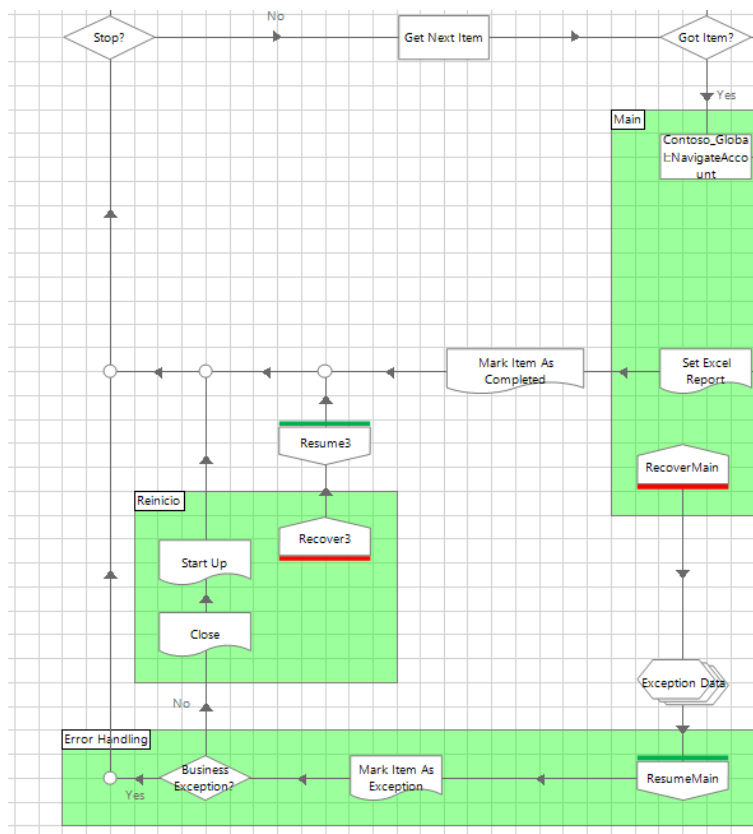


Ilustración 9.20: Tercera tarea del subproceso Facturas

La siguiente imagen muestra, como ejemplo ilustrativo, la página para detectar si hay elementos en cola. Cuando el proceso *Facturas* pasa por el elemento de flujo *Pending Items*, realiza una llamada a esta página, que ejecuta su subproceso y devuelve una salida.

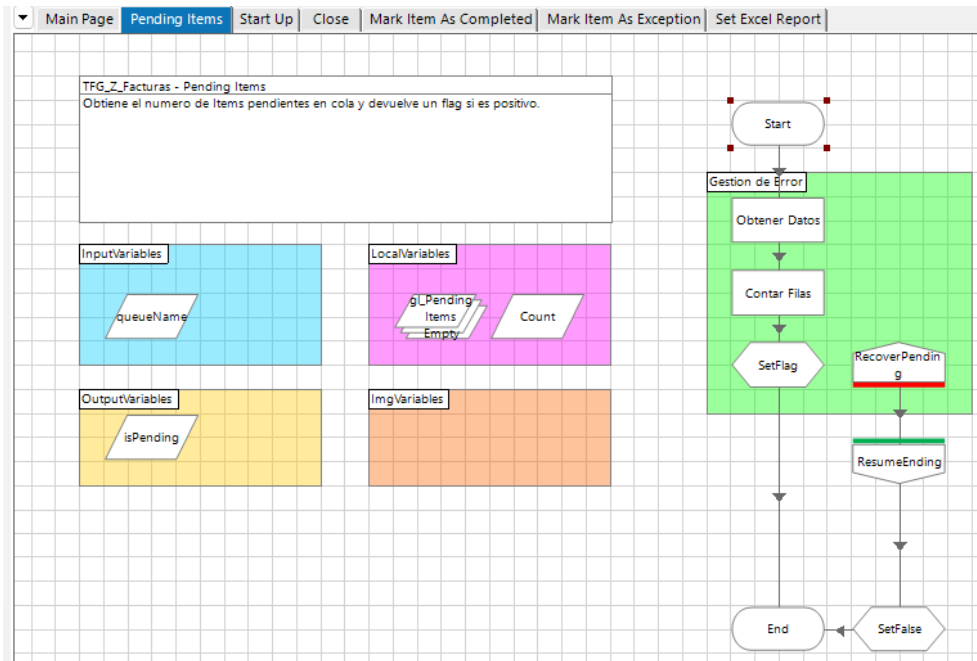


Ilustración 9.21: Página para detectar elementos en cola.

A través del elemento *Start* configuramos como dato de entrada el nombre de la base de datos que hay que revisar.

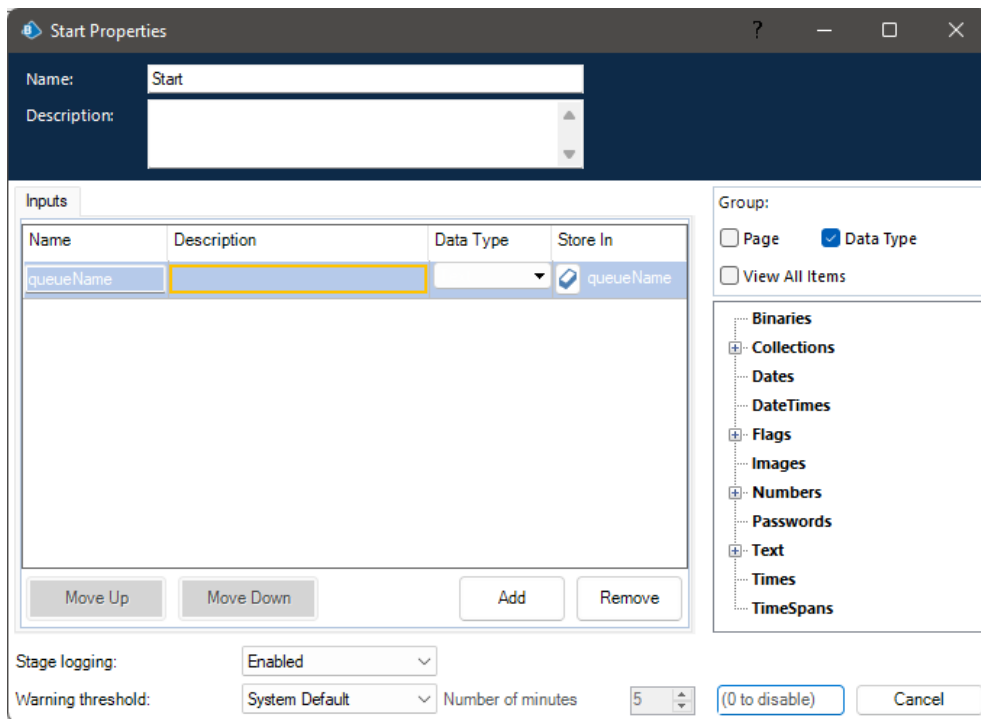


Ilustración 9.22: Propiedades de Start de la página Pending Items

Se obtienen los datos de la base de datos que tienen como tag identificativo “Factura” y se almacenan en la variable *gl_Pending Items*.

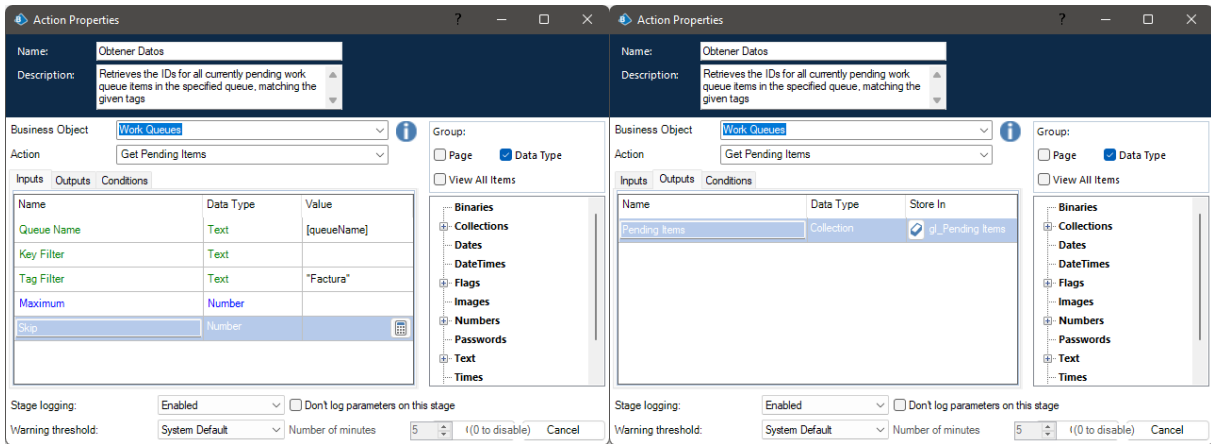


Ilustración 9.23: Ventana de propiedades de la acción Obtener Datos: Inputs y Outputs

Posteriormente se cuentan las filas y se comprueba si la suma es distinta de 0, en cuyo caso se modifica la variable lógica de salida *IspPending* a *True*.

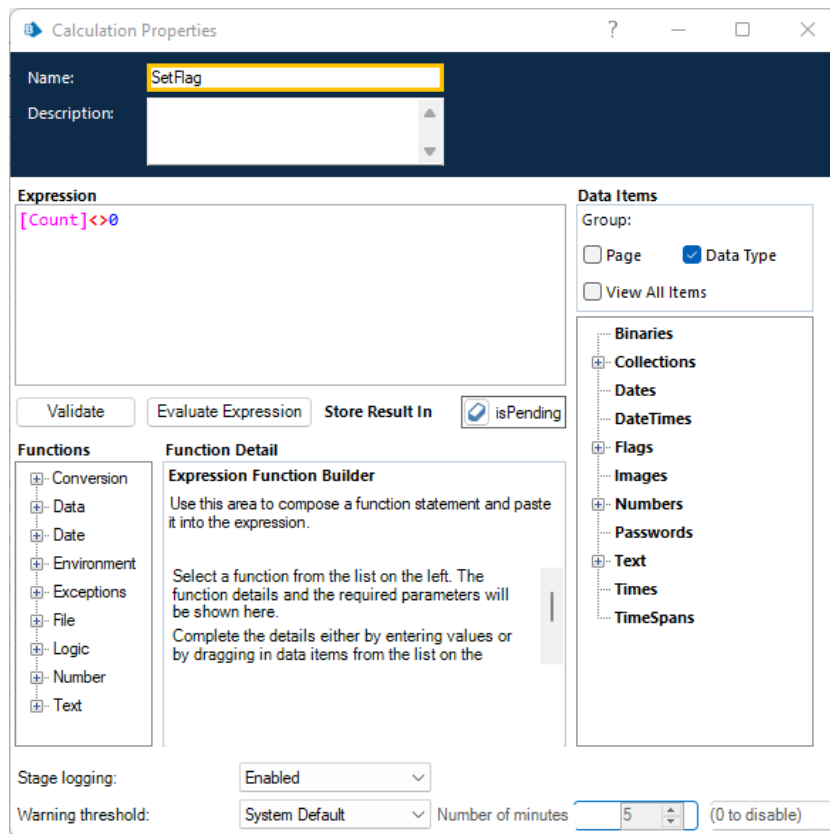


Ilustración 9.24: Ventana de verificación de elementos en cola

9.1.3 Módulos

Si nos fijamos en la ilustración 9.23, observaremos la opción *Business Object: Work Queue*. La acción de extraer elementos de la base de datos se realiza a través de un método llamado *Get Pending Items* del objeto *Work Queue*. Este objeto es, en realidad, un **módulo**. Los módulos en Blue Prism son muy importantes, porque te permite importar acciones y objetos complejos que otras personas (o la propia Blue Prism) han desarrollado. De no haber podido acceder a este módulo, hubiese tenido que desarrollar mi propio objeto con algún método para acceder a los elementos de la cola.

9.2 Process Definition Document (PDD)

El propósito de este documento es proporcionar una descripción detallada del proceso de **Clasificación y control de facturas y presupuestos** que actualmente se está llevando a cabo por los usuarios del departamento de **Facturación y Contabilidad**, y crear una solución a alto nivel para robotizar, incluyendo la programación, la ejecución, el coste de ejecución y el esfuerzo.

El desarrollo de este documento se llevará a cabo en 2 fases:

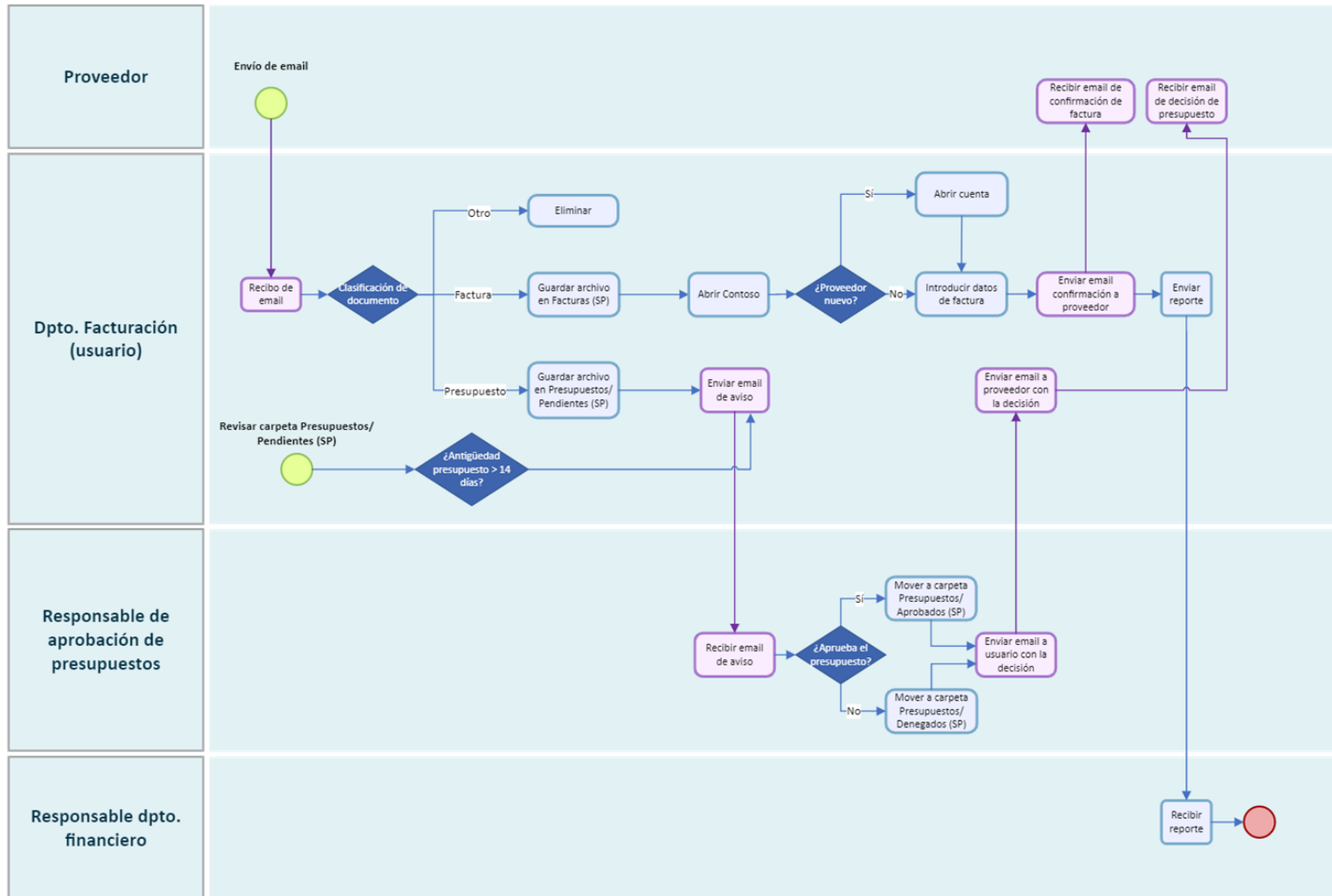
1. **Fase 1:** Se compone de las dos primeras secciones, donde se realiza una primera aproximación al proceso. Estas secciones determinarán si el proceso puede ser robotizado o no.
2. **Fase 2:** Se compone de las secciones 3 y 4, siempre que el proceso sea automatizable. Estas secciones proveen un nivel de detalle suficiente, acerca del proceso, como para poder desarrollar y construir una solución en RPA.

9.2.1 Descripción general del proceso

El proceso de **Clasificación y control de facturas y presupuestos** consiste en cribar la documentación que se recibe en un email específico, clasificando adecuadamente las facturas y los presupuestos. Posteriormente se almacena la información de las facturas en Contoso Invoice y se confirma por email al proveedor la recepción y procesamiento. Si fuera necesario, se crea un nuevo perfil de proveedor. Los presupuestos recibidos son reenviados al responsable de aprobación y los procesados se clasifican según hayan sido aprobados o rechazados. Luego se comunica por email al proveedor la decisión tomada. Al final del día se envía un reporte por email a los responsables con los datos de todas las facturas y presupuestos procesados.

9.2.1.1 Diagrama de alto nivel

Un diagrama de alto nivel es una representación de la secuencia de los pasos más significativos de un proceso. A continuación se muestra el diagrama de alto nivel del proceso que se quiere automatizar.



Comentarios

Los colores indican:

Morado: Comunicación entre departamentos o entre proveedor y usuario.

Azul: Acciones propias del proceso.

9.2.1.2 Objetivos

- 1. Liberar recursos:** Este proceso ocupa la mayor parte de la jornada laboral del trabajador. El objetivo es reducir al mínimo la dedicación de tiempo del trabajador a estas gestiones.
- 2. Aumentar capacidad de procesamiento:** Se planea la apertura de un nuevo almacén, que conllevará el aumento de facturas y presupuestos recibidos y procesados. Se necesita mejorar la capacidad de procesamiento de facturas y presupuestos.

9.2.2 Análisis técnico

A continuación se detalla, en tablas, las métricas del proceso que queremos automatizar.

Tiempo de ejecución manual y horario

Ejecución					
Periodicidad	Diario	Semanal	Mensual	Anual	Bajo petición
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Preguntas	Respuestas
¿La solución se ejecutará en días específicos de la semana?	De lunes a viernes.
¿Hay días en los que la solución no debe ejecutarse?	Días festivos nacionales

Horario	
Hora de comienzo del proceso	09:00
Hora de fin del proceso	18:00
Frecuencia de ejecución en el periodo	Proceso iterativo

Volúmenes & métricas del proceso

Volumen	Cantidad mensual
Mínimo	350
Medio	700
Máximo	1500
Máximo histórico	1650

Preguntas	Respuestas	
¿Existe algún periodo en que se pueda anticipar una alta carga de trabajo?	Sí <input checked="" type="checkbox"/>	No <input type="checkbox"/>
¿Cuándo se dan?	<ul style="list-style-type: none"> • Primera semana de septiembre • Primera semana de octubre • Primera semana de diciembre • Primera semana de junio 	

Métricas	
Tiempo medio de operación (usuario)	7 minutos
Número de recursos que consume el proceso	1
FTE's (1 fte = 160h/mes)	0,65625

Nota: el cálculo del FTE es $\frac{7 \text{ min} \times 700 \text{ vol.medio mensual} \times 1 \text{ recurso}}{60 \text{ min en hora}} = 81,67 \text{ horas} \Rightarrow \frac{81,67}{160} = 0,5104$

Requisitos y criticidad del acuerdo de nivel de servicio (ANS)

ANS	Consecuencias de incumplimiento
Tiempo de respuesta para presupuestos menor a 20 días.	Los precios y descuentos estipulados en el presupuesto podrían no mantenerse.

Existe un acuerdo con los proveedores para mantener el precio de los productos y los descuentos aplicados durante 20 días. Si transcurrido ese tiempo no reciben respuesta, el presupuesto se da por rechazado y se pierden todos los descuentos pactados, así como los precios de venta, que podrían verse actualizados.

Criticidad para el negocio por	Baja	Media	Alta
Retraso en la ejecución del robot	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Robot no disponible	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Retraso en el lanzamiento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sobrecarga de ejecuciones	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

9.2.3 Requisitos de sistema

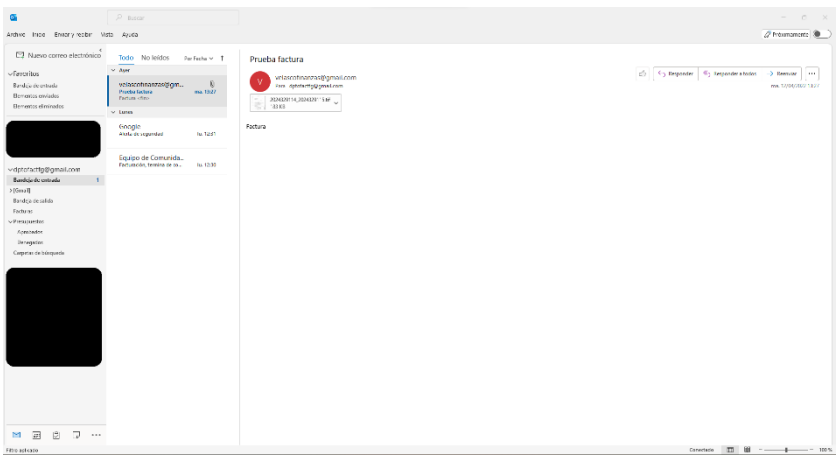
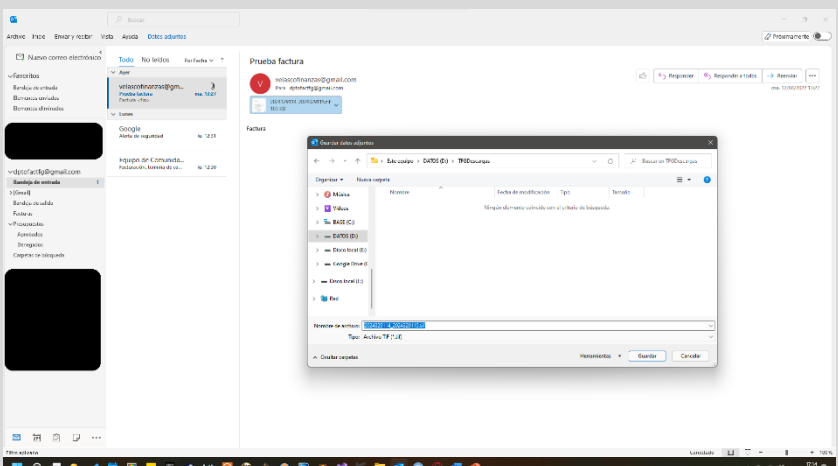
Las aplicaciones que se usan actualmente en el proceso (sin automatizar) son:

Aplicaciones involucradas

Nombre y versión	Lenguaje	Tecnología	Disponibilidad horaria	Comentarios
Outlook 360	Español	Email	09:00 – 18:00	Hay configuradas varias cuentas
SharePoint Web Version	Español	Cloud Storage	09:00 – 18:00	Contiene las subcarpetas para Facturas, Presupuestos y otra carpeta que no se usará en este proceso
Contoso Invoicing V1.0.15.0	Inglés	Gestor de facturación	09:00 – 18:00	

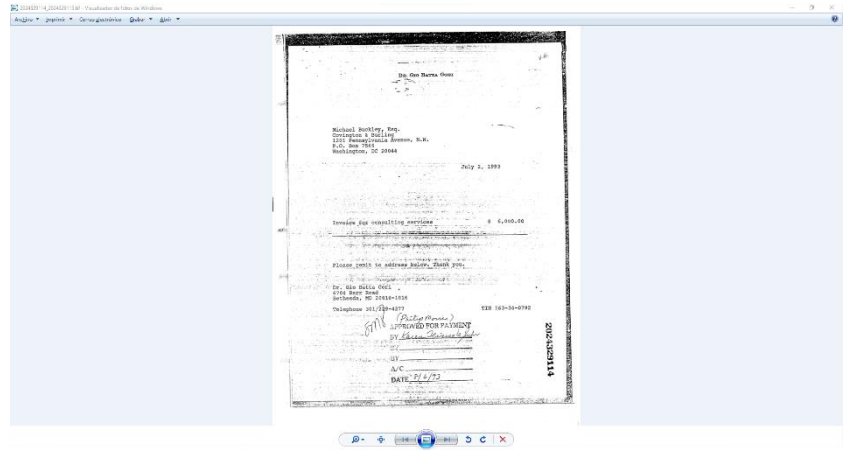
Adobe Acrobat Reader DC V2022.001.20117 64 bits	Español	Lector PDF	09:00 – 18:00
Excel 360	Español	Hoja de cálculo	09:00 – 18:00
Windows 11 64 bits	Español	Sistema Operativo	09:00 – 18:00

9.2.4 Paso a Paso del proceso

Paso	Acción	Screenshots
1	Abrir Outlook y verificar correos no leídos de proveedores	
2	Descargar archivo adjunto y guardarlo en la ruta "D:\TFGDescargas"	

3

Abrir archivo y clasificarlo

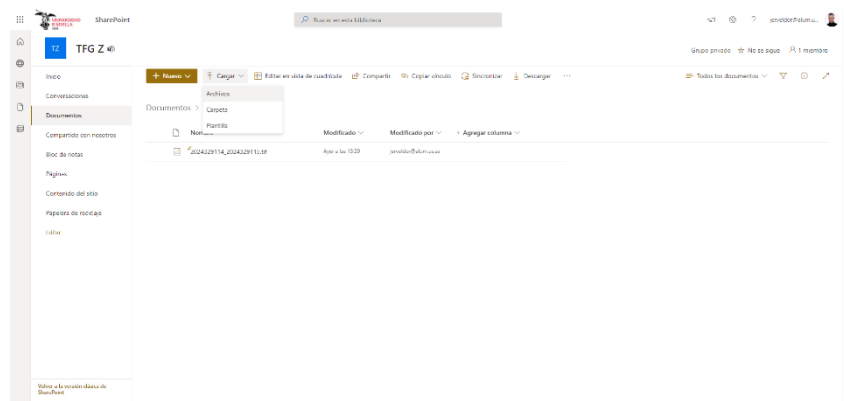


4

Si es una factura

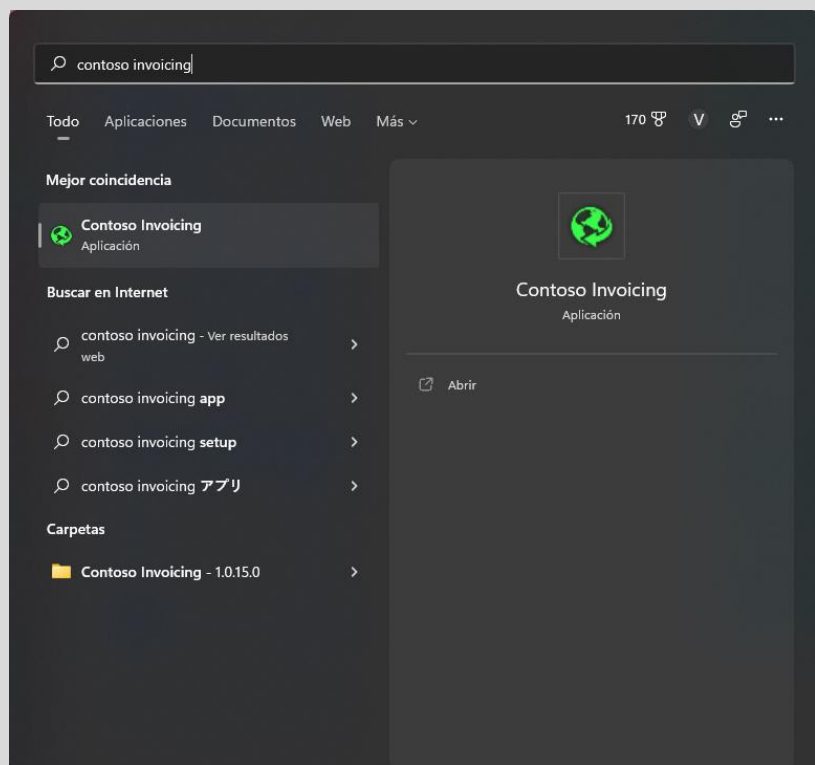
4.1

Subir el archivo a la carpeta Facturas de Sharepoint



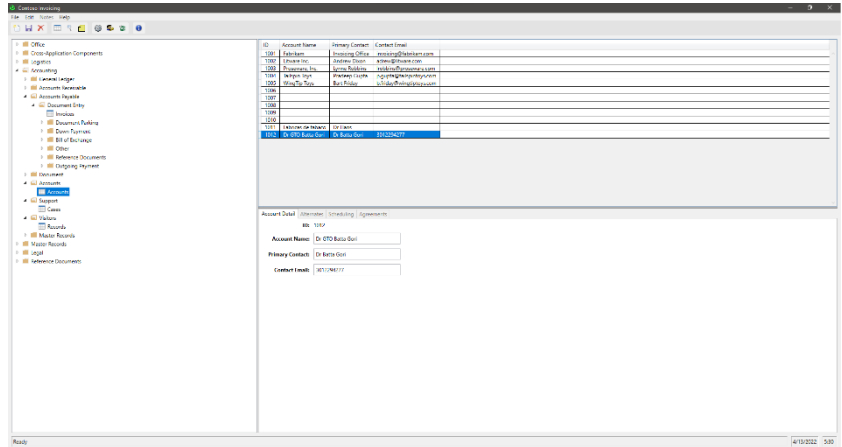
4.2

Abrir Contoso Invoicing



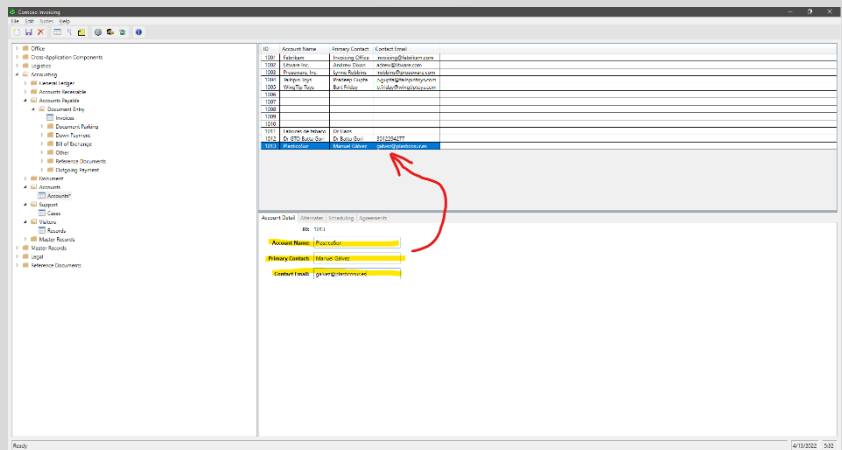
4.3

Comprobar si el proveedor tiene cuenta en Contoso



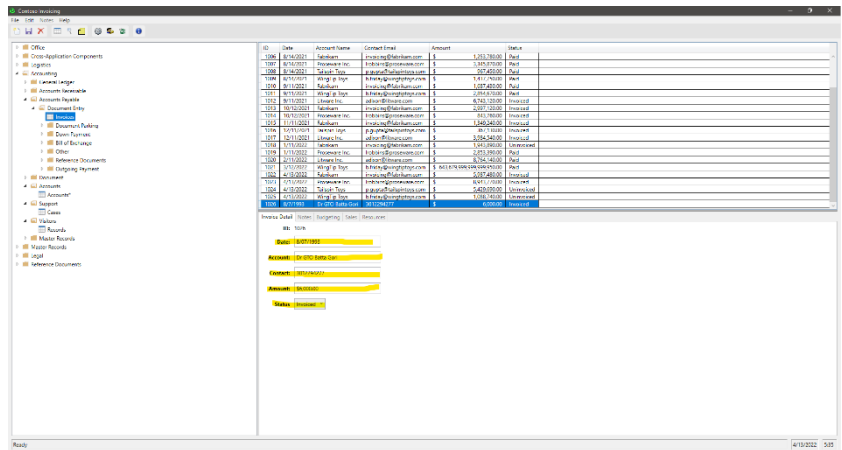
4.3.1

Si no tiene cuenta, crear una nueva

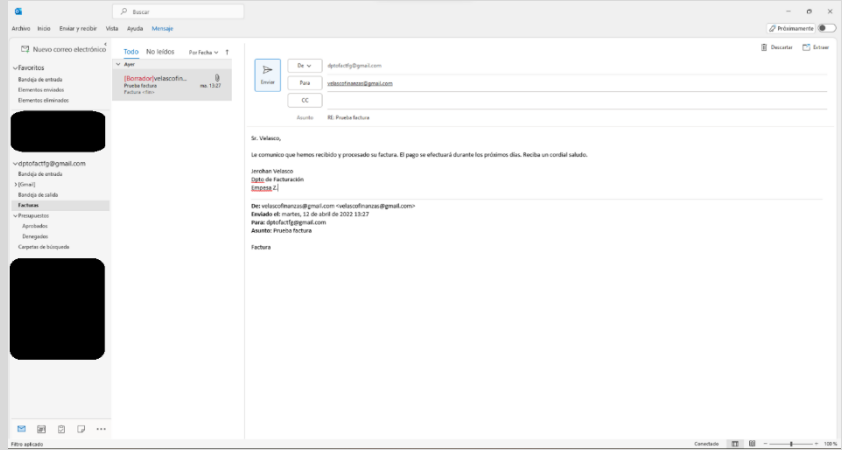


4.4

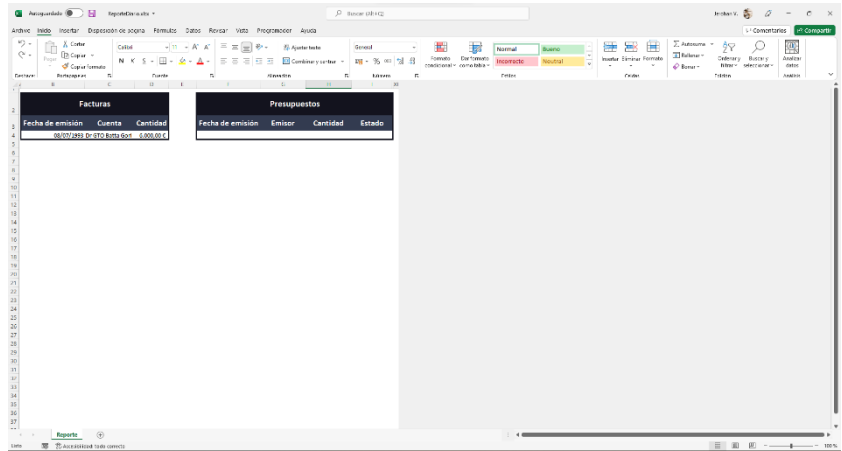
Meter los datos de la factura en Contoso



4.5 Volver a Outlook y responder al proveedor confirmando la recepción y procesamiento de la factura

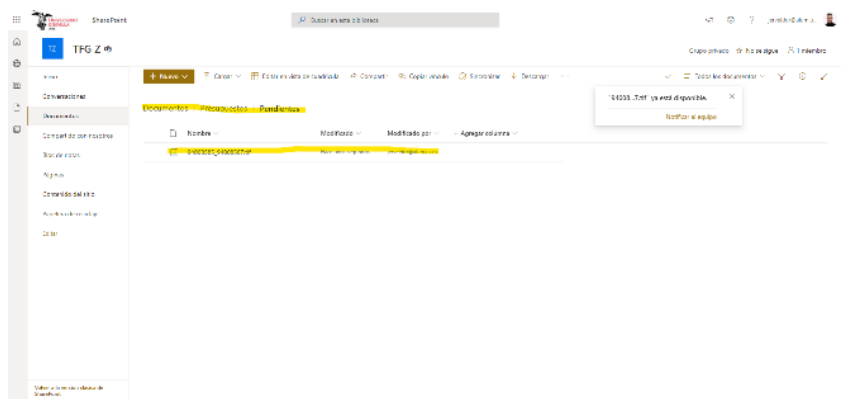


4.6 Abrir el documento "D:\TFGDescargas\ReporteDiario.xlsx" y anotar los datos de la factura en el reporte



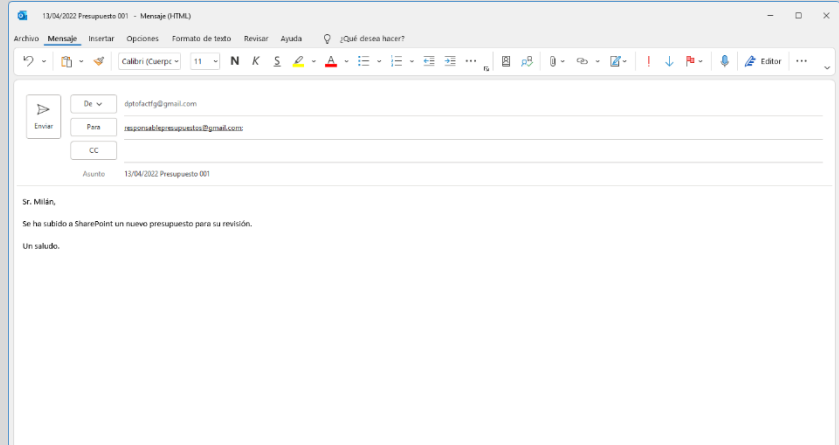
5 Si es presupuesto

5.1 Subir el archivo a la carpeta Presupuestos/ Pendientes de Sharepoint



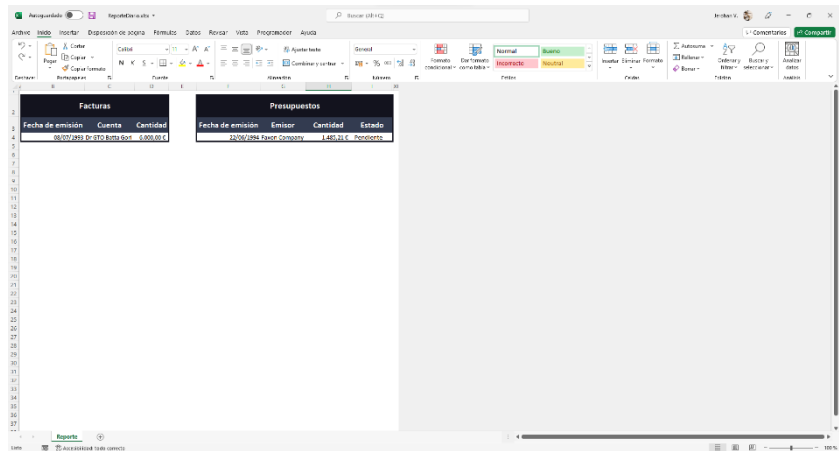
5.2

Enviar un email al responsable de aprobación de presupuestos con un aviso



5.3

Abrir el documento "D:\TFGDescargas\ReporteDiario.xlsx" y anotar los datos del presupuesto en el reporte. Estado: Pendiente

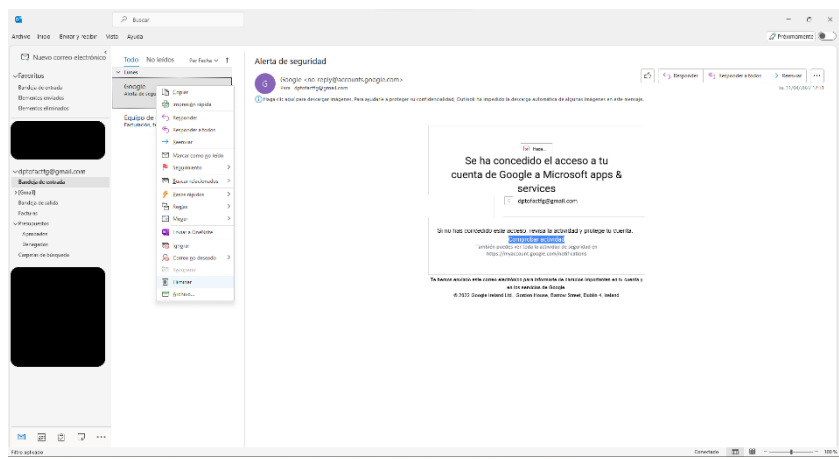


6

Si no es factura ni presupuesto

6.1

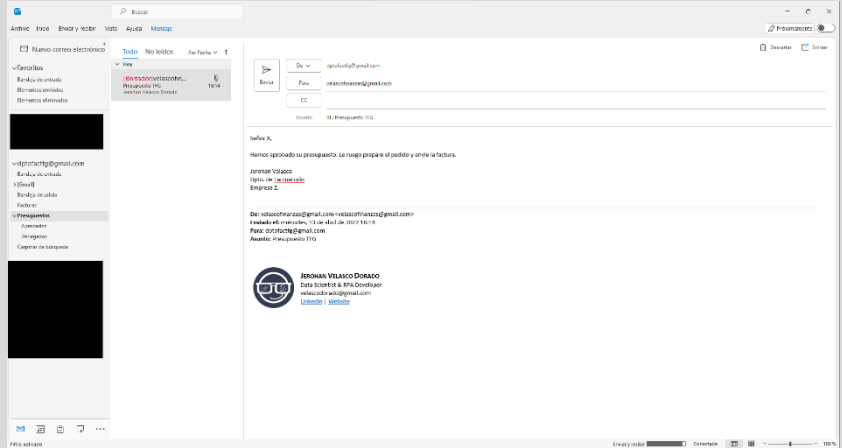
Se elimina el email



Paso	Acción	Screenshots
1	Abrir Outlook y verificar respuestas del responsable de aprobación de presupuestos	
2	Si se ha aprobado	
2.1	Mover el archivo a la carpeta Presupuestos/Aprobados	
3	Si se ha denegado	
3.1	Mover el archivo a la carpeta Presupuestos/Denegados	

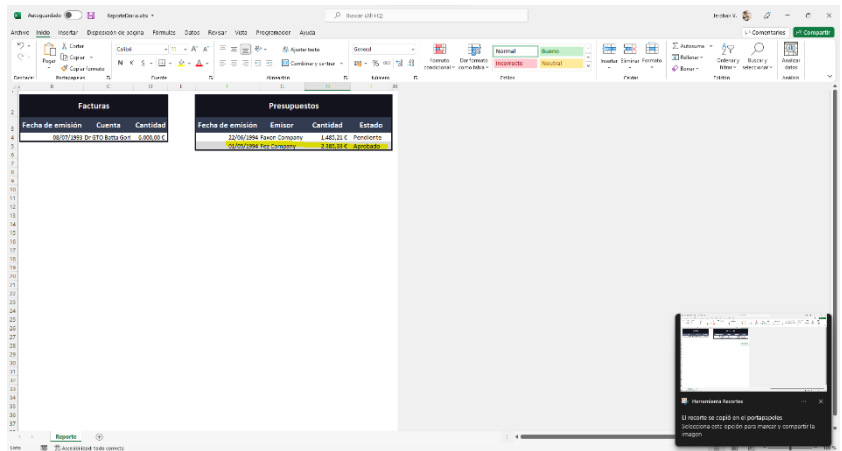
4

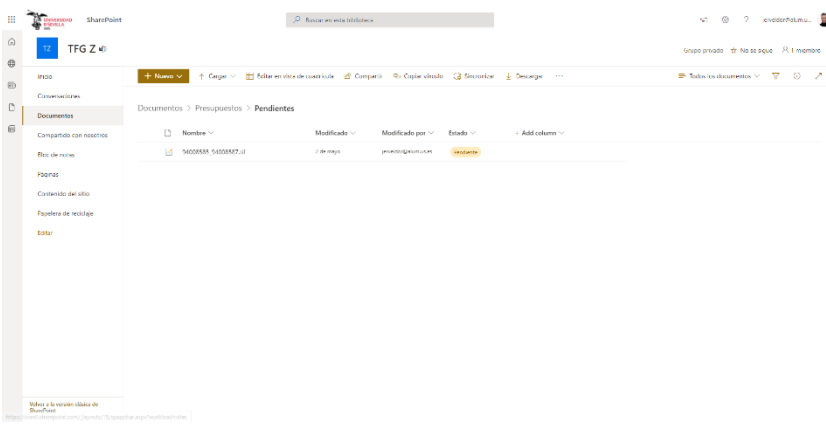
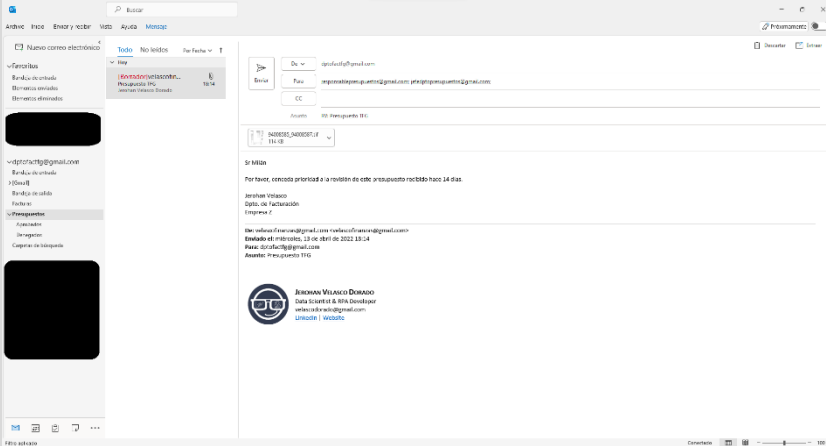
En cualquier caso, comunicar la decisión al proveedor respondiendo a su email original



5

Abrir el documento "D:\TFGDescargas\ReporteDiario.xlsx" y anotar los datos del presupuesto en el reporte, con el estado adecuado.



Paso	Acción	Screenshots
1	Verificar antigüedad de los presupuestos en la carpeta Presupuestos\Pendientes	
2	Si existiera algún presupuesto con antigüedad mayor a 14 días, se envía un mensaje al responsable de aprobación de presupuestos con copia al jefe del departamento	

Nota: Al final del día se envía el archivo “D:\TFGDescargas\ReporteDiario.xlsx” al responsable del dpto. financiero por email.

Tratamiento de errores

En general se suelen dar 3 tipos de errores:

- Fallo de la aplicación:** Si cualquiera de las aplicaciones que se deben usar falla, se reintenta más tarde. Si el error persiste se llama al servicio técnico correspondiente.
- Fallo de internet:** Si la conexión a internet falla, se avisa al encargado de redes y sistemas. Si el fallo es ajeno a la propia empresa, se reintenta la conexión más tarde.
- Archivo corrupto:** Si el archivo que envía el proveedor está corrupto o es ilegible, se le envía un email de aviso para que reenvíe su documento.

9.3 Solution Definition Document (SDD)

Nota: Sólo se hará un SDD a modo de ejemplo para la solución “solo RPA”. El motivo es que el SDD apenas tiene variaciones entre una solución y otra y no aportaría nada nuevo ni interesante.

9.3.1 Descripción de la solución

En la primera solución se propone una robotización parcial del proceso. Puesto que no existe un formato predefinido y claramente estructurado de factura y/o presupuesto, es necesario que el usuario identifique y clasifique manualmente cada email recibido y descargue su archivo adjunto. Además, deberá introducir en una hoja de cálculo los siguientes datos:

- **ID:** Identificador único de entrada.
- **Tipo de documento:** factura o presupuesto.
- **Nombre de documento:** nombre del archivo descargado.
- **Fecha:** fecha de emisión del documento.
- **Proveedor:** nombre del proveedor que envió el documento.
- **Email:** email del proveedor que envió el documento.
- **Contacto:** nombre del contacto de la empresa proveedora.
- **Cantidad:** total de la factura o del presupuesto.
- **Estado:** pendiente o pagada para facturas. Pendiente, aprobado o denegado para presupuestos.

El resto del proceso se dividirá en 5 subprocesos que se podrán ejecutar de forma independiente o como parte del proceso completo.

- Un subproceso para recolectar los datos.
- Un subproceso para gestionar las facturas.
- Un subproceso para gestionar los presupuestos enviados por parte del proveedor.
- Un subproceso para gestionar los presupuestos tras la respuesta del responsable.
- Un subproceso para gestionar los presupuestos con más de 14 días de antigüedad.

El robot será ejecutado cada 2 minutos para asegurar que no se acumulan demasiados archivos para procesar. Además, se ejecutará de lunes a viernes de 09:00 a 18:00 salvo festivos. Los detalles de cada subproceso se explicarán más adelante.

9.3.2 Alcance del proceso

Dentro del alcance del proceso se encuentra:

1. Reubicación de los documentos a sus carpetas correctas en SharePoint.
2. Extracción de datos de la hoja de cálculo.
3. Registro de cuenta nueva de proveedor, si fuera necesario, y grabación de datos de la factura en Contoso.
4. Envíos de emails.
5. Elaboración y envío del reporte.

Fuera del alcance del proceso se encuentra la identificación y clasificación del tipo de documento, así como la extracción de sus datos.

9.3.3 Flujo de datos y aplicaciones

Entrada

Entrada	Formato	Aplicación	Frecuencia de tiempo
Hoja de cálculo	Excel	Microsoft Office Excel 360	Diaria

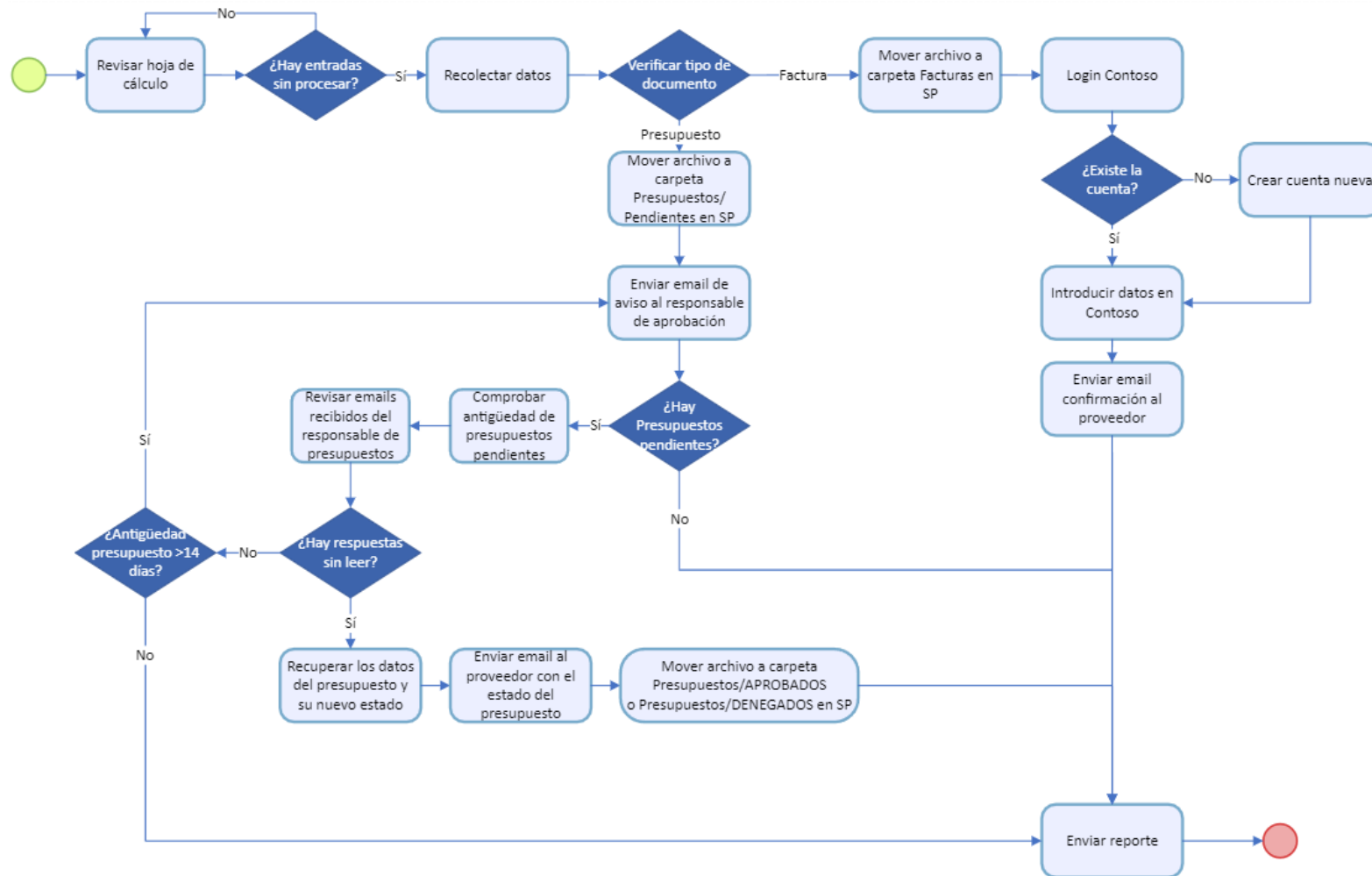
Salida

Salida	Formato	Ubicación	Frecuencia de Tiempo
Reporte con los resultados de la ejecución	Excel	Envío por mail	Diaria

Aplicaciones con las que se interactúa

Aplicación	Interfaz	Clave Requerida / URL	Ente de actuación
Microsoft Office Outlook 360	Windows	No requiere clave	Humano
Microsoft Office Excel 360	Windows	No requiere clave	Humano/Robot
SharePoint Web Version	Opera	https://uses0.sharepoint.com/sites/TFGZ/Documentos%20compartidos/Forms/AllItems.aspx	Robot
Contoso Invoicing Version 1.0.15.0	Windows	No requiere clave	Robot

9.3.4 FlowChart



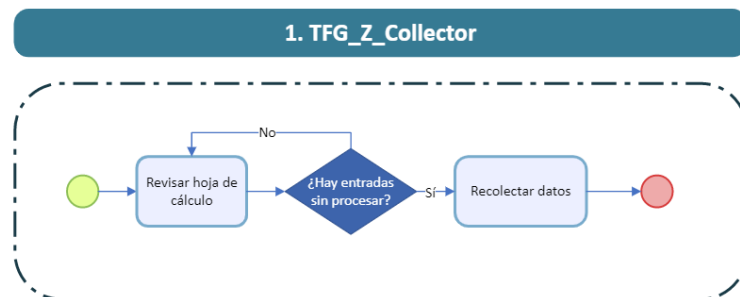
9.3.4.1 Subproceso 1: Colector de datos

Este proceso consiste en la recolección y almacenamiento de datos en la base de datos propia de Blue Prism a partir de la hoja de datos *DatosDiarios.xlsx*.

ID	Tipo_de_documento	Nombre_de_documento	Fecha_de_emision	Proveedor	Email	Contacto	Cantidad	Estado
1	Factura	fact01.jpg	06/10/1994	SmallHero	a@a.com	SmallHero	€ 327,61	Pendiente
2	Factura	fact02.png	10/24/2016	LaboralDemo	mail@mail.es	LaboralDemo	€ 46,84	Pendiente
3	Presupuesto	Pres01.png	11/10/2020	EmpresaPrueba	asd@asd.es	Ara	€ 211.750,00	Pendiente
4	Factura	fact03.jpg	02/25/2021	Jero	aja@hmm.com	Jerohan	€ 100.550.050.500,00	Pendiente
5	Presupuesto	Pres03.jpg	05/15/2021	PacoResource	eso@aquello.es	Paco	€ 25.632,05	Pendiente

Este subproceso se encargará de:

1. Abrir el documento *D:\TFGDescargas\DatosDiarios.xlsx*.
2. Leer todos los datos contenidos en la tabla.
3. Procesarlos para almacenarlos en la base de datos de Blue Prism.

Workflow

9.3.4.2 Subproceso 2: Tratamiento de facturas

Este subproceso está basado en la robotización del procesamiento de las facturas. Una vez recolectados los datos este subproceso se encargará de:

1. Filtrar las facturas de la base de datos.
2. Mover el documento desde la carpeta de descarga hasta la carpeta *Facturas* en Sharepoint.
3. Abrir el programa Contoso Invoices.
4. Verificar si existe el proveedor en la base de datos de proveedores.
 - a) En caso negativo, abrir una nueva cuenta con los datos recuperados de la hoja de cálculo.
5. Insertar los datos de la factura: fecha de emisión, nombre del proveedor, email de contacto, cantidad total facturada y estado de la factura.
6. Enviar email al proveedor con el siguiente texto y formato:

Para: [Email del proveedor]

Asunto: Notificación_Factura_[Fecha/De/Emisión]

Mensaje:

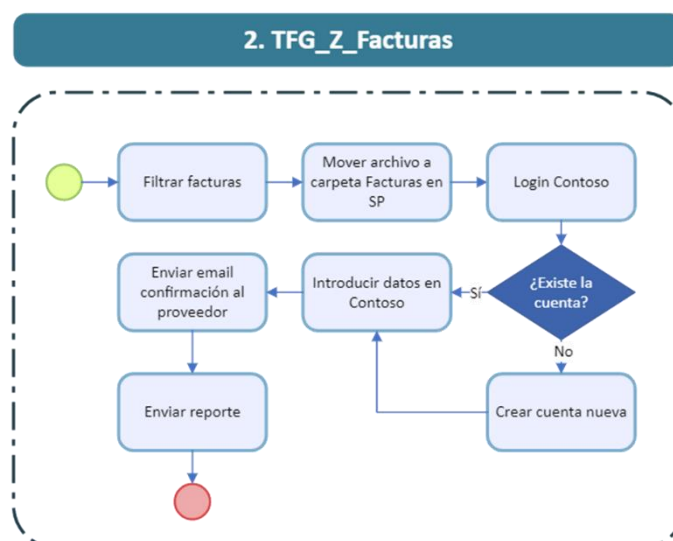
Notificación EmpresaZ

Estimado [Nombre de proveedor], le notificamos que la factura emitida el día [Fecha de emisión] con un montante total de [Cantidad total facturada] ha sido recibida y procesada. En un plazo máximo de 3 días se procederá al pago.

Jerohan Velasco Dorado

Dpto de facturación EmpresaZ

7. Enviar reporte

Workflow

9.3.4.3 Subproceso 3: Tratamiento de presupuestos nuevos

Este subproceso está basado en la robotización del procesamiento de los presupuestos en la fase de recepción. Una vez recolectados los datos este subproceso se encargará de:

1. Filtrar los presupuestos nuevos de la base de datos.
2. Mover el documento desde la carpeta de descarga hasta la carpeta *Presupuestos/Pendientes* en Sharepoint.
3. Enviar un email al responsable de aprobaciones de presupuestos con el siguiente texto y formato:

Para: responsablepresupuestos@gmail.com

Asunto:

Presupuesto_[NombreFichero]_[Fecha/De/Emisión]_[NombreDeProveedor]_[Cantidad]

Mensaje:

Sr Milán,

Se ha subido un nuevo presupuesto a SharePoint para su revisión.

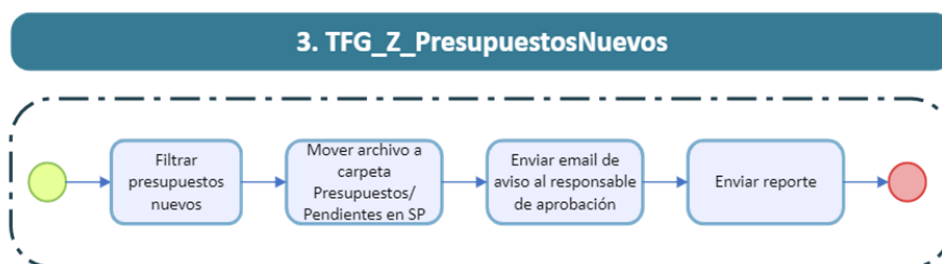
Un saludo.

Jerohan Velasco Dorado

Dpto de facturación EmpresaZ

4. Enviar reporte

WorkFlow



9.3.4.4 Subproceso 4: Tratamiento de presupuestos antiguos

Este subproceso está basado en la robotización del procesamiento de los presupuestos con más de 14 días de antigüedad. Una vez procesados los presupuestos nuevos, este subproceso se encargará de:

1. Verificar la antigüedad de los presupuestos en fase de revisión.
2. Si existieran documentos con más de 14 días de antigüedad, se enviará un email al responsable de aprobación de presupuestos con copia al jefe de departamento.

Para: responsablepresupuestos@gmail.com, jefepresupuestos@gmail.com

Asunto:

Presupuesto_PENDIENTE_[NombreFichero]_[Fecha/De/Emisión]_[NombreDeProveedor]_[Cantidad]

Mensaje:

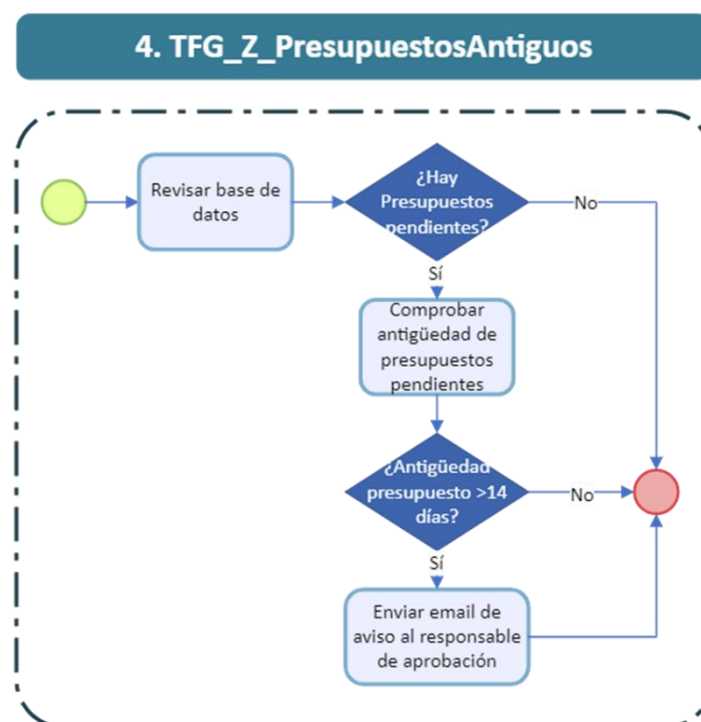
Sr Milán,

En la carpeta *Presupuestos / Pendientes* de SharePoint se encuentra el presupuesto descrito, con más de 14 días de antigüedad, sin revisar. Por favor, conceda la máxima prioridad a este presupuesto. Quedo a la espera de su decisión para comunicárselo al proveedor a la mayor brevedad.

Un saludo.

Jerohan Velasco Dorado

Dpto de facturación EmpresaZ

Workflow

9.3.4.5 Subproceso 5: Tratamiento de presupuestos tras la respuesta

Este subproceso está basado en la robotización del procesamiento de los presupuestos tras la decisión del responsable de aprobación. El cuerpo del mensaje de respuesta deberá contener únicamente la decisión tomada: Aprobado o Denegado.

Una vez procesados los presupuestos nuevos y los posibles antiguos, este subproceso:

1. Comprobará si existen respuestas por parte del responsable de presupuestos para alguno de los presupuestos pendientes.
2. En caso afirmativo, moverá el documento desde la carpeta *Presupuestos/Pendientes* hasta la carpeta correspondiente: *Presupuestos/APROBADOS* o *Presupuestos/DENEGADOS* en SP.
3. Enviar email al proveedor con el siguiente texto y formato:

Para: [Email del proveedor]

Asunto: Presupuesto_[Fecha/De/Emisión]_[Cantidad]

Mensaje:

Notificación EmpresaZ

Estimado [Nombre de proveedor], le notificamos que el presupuesto emitido el día [Fecha de emisión] con un montante total de [Cantidad total facturada] ha sido [Estado].

{Si ha sido aprobado} En un plazo máximo de 7 días se procederá a realizar el pedido.

{Si ha sido denegado} No obstante, mantendremos su contacto en nuestra base de datos por si necesitáramos de sus productos en un futuro.

Jerohan Velasco Dorado

Dpto de facturación EmpresaZ

4. Enviar reporte

Workflow