



***Facultad  
de  
Ciencias***

**TECNOLOGÍAS RPA EN EL SECTOR  
LOGÍSTICO  
(RPA technologies in logistics sector)**

Trabajo de Fin de Grado  
para acceder al

**GRADO EN INGENIERÍA INFORMÁTICA**

Autor: Alberto de Castro Manzano  
Director: Marta Elena Zorrilla Pantaleón

Febrero – 2020



## Índice de contenidos

Índice de contenidos .....	3
Índice de tablas .....	4
Índice de figuras .....	4
Resumen.....	5
Abstract .....	5
1. Introducción .....	6
1.1 Antecedentes y objetivos .....	6
1.2 Metodología y plan de trabajo.....	6
2. Tecnologías RPA .....	8
2.1 Herramientas RPA .....	9
2.2 Criterios para la evaluación y selección de herramientas.....	14
2.3 Evaluación de herramientas: UiPath vs TagUI .....	16
3. Especificación de requisitos .....	22
3.1 Asignación de vehículo a pedidos recibidos por formulario web .....	22
3.2 Formato y carga de fichero de facturación .....	23
4. Diseño e implementación .....	25
4.1 Asignación de vehículo a pedidos recibidos por formulario web .....	27
4.2 Formato y carga de fichero de facturación .....	31
5. Pruebas.....	34
6. Conclusiones y trabajos futuros.....	36
6.1 Conclusiones.....	36
6.2 Trabajos futuros .....	36
Referencias.....	37

## Índice de tablas

Tabla 2.1: Comparación entre TagUI y UiPath. ....	21
Tabla 5.1: Casos de pruebas de la implementación.....	34

## Índice de figuras

Figura 2.1: Mercado de servicios y software RPA (2016-2022) (HFS Research, 2018). ....	8
Figura 2.2: Aplicaciones de RPA en diferentes sectores. ....	9
Figura 2.3: Cuadrante Mágico de Gartner RPA 2019 (Gartner, 2019). ....	11
Figura 2.4: Ranking 2018 (EGHAM, 2019). ....	12
Figura 2.5: Lista de herramientas open source RPA (AIMultiple, 2020). ....	12
Figura 2.6: Criterios de elección de herramientas RPA (Kappagantula, 2019). ....	15
Figura 2.7: Lectura y tratamiento archivo CSV.....	17
Figura 2.8: Código para leer y rellenar formulario.....	18
Figura 2.9: Código para extraer datos de una página web. ....	19
Figura 4.1: Organigrama para completar asignación de vehículo a pedidos recibidos por formulario web.....	25
Figura 4.2: Organigrama para completar formato y carga de fichero de facturación. ....	25
Figura 4.3: Secuencias creadas en UiPath para ambos casos de uso.....	26
Figura 4.4: Visualización de la codificación de la secuencia “ObtenerCorreo”.....	27
Figura 4.5: Ejemplo de descomprensión archivo en UiPath. ....	28
Figura 4.6: Obteniendo coordenadas del punto de recogida. ....	29
Figura 4.7: Configuración de conexión con la base de datos. ....	29
Figura 4.8: Pseudocódigo del algoritmo para conocer el vehículo más cercano. ....	30
Figura 4.9: Obtención de valores de una columna.....	30
Figura 4.10: Ejecución de la inserción del json.....	31
Figura 4.11: Script que inserta el json. ....	31
Figura 4.12: Comprido y envío de CMR.....	31
Figura 4.13 Código que comprueba las fechas.....	32
Figura 4.14: Código que comprueba que la suma de ciertas celdas es igual al total.....	32
Figura 4.15: Código que comprueba si el número de facturación es texto. ....	32
Figura 4.16: Código que comprueba si la fila tiene errores de formato. ....	32
Figura 4.17: Ejemplo de la importación de una macro a Excel. ....	33
Figura 4.18: Código para subir los archivos o crear un log. ....	33

## Resumen

Actualmente la mayoría de las empresas están inmersas en programas de mejora continua con objeto de incrementar la eficiencia, productividad, calidad y seguridad de los procesos operacionales de la organización, por ejemplo, mediante avances en la digitalización y automatización de todos los eslabones de la cadena de negocio que sean susceptibles de ser sustituidos por robots software.

Este trabajo se centra en estudiar, analizar y comparar distintas tecnologías RPA (Robotic Process Automation) para la construcción de robots software. En primer lugar, se describen estas tecnologías de automatización, así como las herramientas open source y comerciales más relevantes para su implementación, incluyendo un análisis comparativo entre UiPath y TagUI en base a sus funcionalidades e idoneidad para una serie de casos de uso.

Por último, con la tecnología seleccionada, UiPath, se desarrollan soluciones para dos casos de uso orientados al sector del transporte. En particular, ambos están dirigidos hacia la automatización de las tareas que realiza el auxiliar de transporte como son la recepción de pedidos, asignación de conductores y generación de la carta de porte o CMR y la verificación y carga de archivos de facturación a la base de datos.

Palabras clave: RPA, robot software, automatización de procesos, sector logístico

## Abstract

Modern companies are increasingly focussing their efforts on continuous improvement processes in order to boost their efficiency, productivity, quality and security, for example improving the degree of digitalization and automation for all the tasks within the business chain that could be replaced by software robots.

The aim of this project is to study, analyse and compare RPA (Robotic Process Automation) technologies for the development of software robots. First, these automation technologies are described, next, the most relevant open source and commercial tools for their implementation are listed, including a comparative analysis between UiPath and TagUI based on their functionalities and suitability for several use cases.

The selected technology, UiPath, is then used to develop solutions for two use cases from the logistics sector. Specifically, the goal is the automation of tasks executed by a transport assistant such as the reception of orders, the assignment of drivers, the generation of waybills or CMR, and the verification and upload of billing files to the database.

Keywords: RPA, software robot, process automation, logistics sector

## 1. Introducción

En este capítulo se realiza una introducción al concepto de automatización robótica de procesos, más conocido como RPA (Robotic Process Automation), ámbito sobre el que trata este trabajo fin de grado (TFG). Asimismo, se describen los antecedentes que promovieron la realización de este trabajo y los objetivos del mismo.

### 1.1 Antecedentes y objetivos

Este TFG se enmarca dentro del proyecto de I+D llamado ROBOTICO, que tiene como objeto estudiar, analizar y comparar distintas tecnologías de software que permitan la construcción de robots software, así como el desarrollo de un prototipo que automatice una selección de casos de uso orientados al sector del transporte.

Estas tecnologías se conocen por sus siglas RPA y su fin último es la sustitución de procesos o acciones que realiza un ser humano manualmente sobre un sistema informático por robots software (AIMultiple, 2019). Por ejemplo, RPA permite recopilar la información contable de una empresa, agregar esos datos y hacer un informe de manera automática, permitiendo así a los empleados dedicarse a otros menesteres.

Fagor Telecom (en adelante, Fagor) como empresa interesada en este proyecto, pretende estudiar y analizar las posibilidades que permite las tecnologías RPA, su ámbito de aplicación, el coste económico y temporal, además de su impacto en la productividad. Para así poder tomar decisiones sobre las tareas a ser automatizadas bajo este paradigma y aplicarlas en un futuro, mejorando la competitividad.

Esta empresa cuenta con distintos productos, entre ellos está FlotasNet, que permite la gestión y control remoto de flotas de vehículos. En relación a este, Fagor ha seleccionado dos casos de uso para validar la relación coste/beneficio del uso de las tecnologías RPA.

En concreto, el objeto de este TFG es:

- Primeramente, estudiar, analizar y comparar software RPA a través del estudio de manuales y la implementación de casos de uso sencillos.
- Y a continuación, la implementación de los dos casos de uso propuestos por Fagor con la tecnología RPA seleccionada. Uno relacionado con la recepción de pedidos y asignación de conductor y generación de la carta de porte o CMR, y otro, asociado con la verificación y carga de archivos de facturación en una base de datos.

### 1.2 Metodología y plan de trabajo

Como se ha mencionado, este proyecto se concibió dentro de un proyecto de I+D y, por tanto, no estaba sujeto a resultados finales bien establecidos, la empresa no tenía conocimientos en la materia ni expertos en este ámbito.

El trabajo ha sido realizado por el autor de este TFG tratando de conseguir los requisitos especificados. Para ello se diferencian dos apartados en este trabajo, uno de investigación de las tecnologías y otro de implementación de dos casos de uso. La estrategia seguida para acometer el proyecto fue en primer lugar, investigar sobre tecnologías RPA al tiempo que se implementaban casos de uso sencillos, con el fin de ir profundizando en las capacidades y posibilidades que ofrecen estas herramientas. Esta actividad se acometió durante los meses de octubre y noviembre. Seguidamente se entabló una reunión con Fagor en la que se

especificaron los casos de uso que para ellos tenía interés, los cuales se abordaron a continuación. Por último, en la primera semana de febrero, se les mostró el estudio e implementación realizadas para su evaluación, y en su caso, adopción en la empresa.

## 2. Tecnologías RPA

Hoy en día las empresas intentan aplicar las nuevas tecnologías con el objetivo de reducir costes y obtener mejores resultados. Aquí es donde RPA entra en juego. Como se dijo anteriormente, esta tiene por objeto automatizar tareas repetitivas, de tal manera que un robot software realice el papel de un empleado y así el empleado utilice ese tiempo en tareas de mayor envergadura. También se suelen utilizar para mejorar la experiencia del usuario.

Principalmente la aplicación de RPA proporciona los siguientes beneficios:

- Mayor calidad y precisión en las tareas, principalmente debido a la eliminación de errores humanos.
- Reducción de costes, fundamentalmente por la reducción de horas de empleado (FTE) o su dedicación a tareas de mayor valor.
- Aumento de productividad: los robots trabajan con mayor velocidad que los humanos y, además, se pueden desplegar en paralelo en función de la carga de trabajo.
- Mejora medible, dado que se registran los datos de todas las actividades y se podrá analizar e incluso mejorar los procesos tras su análisis.
- Desarrollo de nuevas funcionalidades con desarrollos no invasivos y de bajo coste.

Con objeto de poder valorar el peso de esta tecnología actualmente, la Figura 2.1, muestra el crecimiento anual en investigación en el Mercado de servicios y software RPA (2016-2022) (HFS Research, 2018). Se puede observar cómo del 2016 al 2017 se incrementó la inversión en esta tecnología casi un 90%, comenzando en 612 millones de dólares y a día de hoy alcanza los 2933 millones de dólares. Lo que indica que el uso de esta tecnología ha aumentado considerablemente, pero no solo su uso en servicios, sino que también se observa en la gráfica el dinero invertido en investigar y mejorar el software RPA.

### RPA Services and Software Market 2016-2022

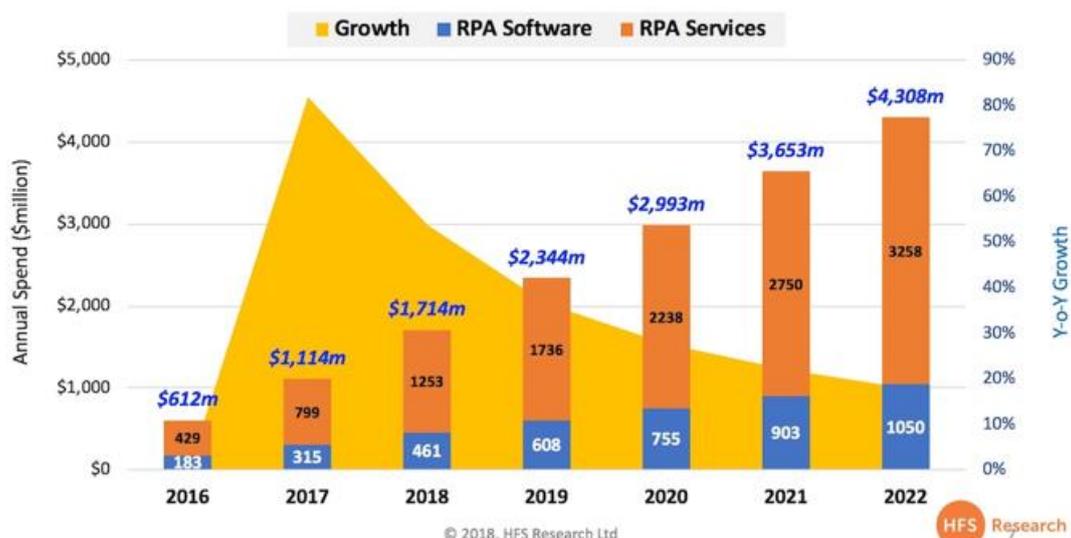


Figura 2.1: Mercado de servicios y software RPA (2016-2022) (HFS Research, 2018).

Según las consultoras más importantes en tecnologías de la información, como es Gartner (EGHAM, 2019), el mercado seguirá creciendo a medida que las organizaciones buscan la forma de aumentar su eficiencia operativa. Para el 2022 se estima un crecimiento del 100% tomando

como referencia el año 2019 (ver Figura 2.1), lo que confirmaría la previsión de estas grandes consultoras.

Como se desprende de la Figura 2.2, es una tecnología que está en auge, debido a la gran variedad de aplicaciones que tiene para diferentes sectores. A modo de ejemplo se enumeran los siguientes casos.

Las compañías de servicios financieros podrían usar RPA para diferentes tareas como los pagos con moneda extranjera, automatización de la apertura y cierre de cuentas, o auditando peticiones y procesos de reclamaciones del seguro y así poder detectar el uso fraudulento de estos. De cara al consumidor, RPA puede ayudar a las compañías financieras a ofrecer un servicio al cliente automático, incluyendo la verificación de las firmas electrónicas, así como la subida de documentos y validación de información en procesos automáticos de confirmación o negación.

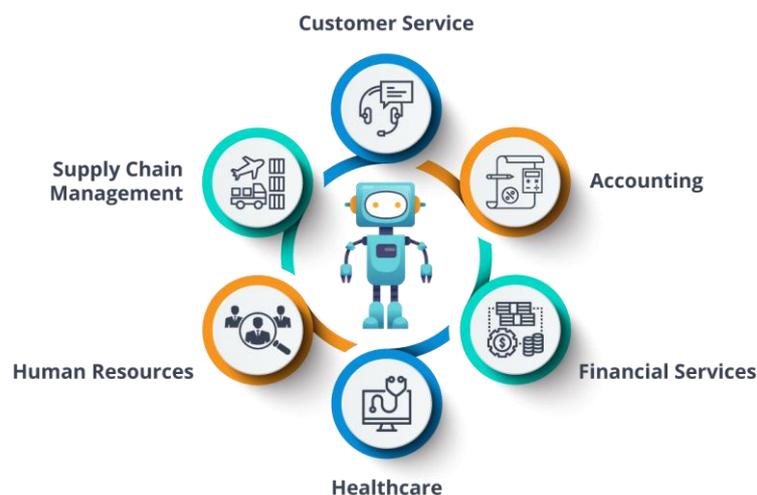


Figura 2.2: Aplicaciones de RPA en diferentes sectores.

En el área de recursos humanos, RPA puede automatizar los procesos de altas y bajas de los empleados, además de la actualización de información del empleado y las hojas de horas de trabajo de este.

También se puede aplicar al área de las organizaciones médicas, para el manejo de los expedientes del paciente, el procesamiento de reclamaciones, realización de facturas, además de la creación de informes y análisis de estos (DevTools, s.f.).

Otros de los posibles usos de RPA es en el sector de la logística, donde permite automatizar las tareas de pedidos y pagos, o el seguimiento de los pedidos cuando estos están dirigiéndose a su destino.

Estos son solo algunos casos de aplicación de las tecnologías RPA los cuales permiten ver y valorar su potencial y utilidad.

## 2.1 Herramientas RPA

Antes de entrar en detalles sobre las herramientas que hay en el mercado, primero debemos describir los diferentes tipos de herramientas RPA que existen.

Los predecesores de las herramientas RPA fueron las automatizaciones que los empleados realizaban a procesos simples con la ayuda de Excel 97 o versiones posteriores con la utilización

de macros. El problema que tenía este tipo de automatización era su baja escalabilidad y seguridad, aun así, la productividad aumentaba.

Tras esto aparecieron las primeras herramientas RPA, que fueron los robots (*bots*) programables. Para la utilización de estas herramientas era necesario que los programadores entendiesen todas las acciones que se necesitaban para finalizar el proceso, y así poder programarlos. Estas herramientas interactuaban con otros sistemas y requerían de una entrada de datos específica.

Otro tipo de herramientas RPA que han surgido recientemente son las llamadas '*self-learning*' o herramientas de auto aprendizaje. Su función es observar a los empleados en acción y entender los procesos que ellos realizan. Cuando tienen la seguridad de lo que tienen que hacer en cada momento, realizan la acción, en el caso de que haya una nueva entrada, la cual aún no tienen conocimiento de ella, preguntan al humano para saber que han de hacer (AIMultiple, 2019).

Por último, se encuentran las herramientas RPA, llamadas cognitivas o *bots* inteligentes. Estas herramientas son una versión mejorada de los '*self-learning bots*', con funcionalidades como el *Machine Learning*, reconocimiento de imágenes y la incorporación del *Natural Language Processing*. Estas herramientas pueden trabajar con diferentes tipos de datos, ya sean datos estructurados o no estructurados, apoyando de manera efectiva en los procesos de toma de decisiones, gracias al aumento de datos digitalizados e información procesada (Kappagantula, 2019).

Como se puede concluir, las tecnologías dentro de este ámbito están en constante mejora y actualización, integrando las tecnologías más innovadoras, consiguiendo así una mayor capacidad de automatización.

Después de la clasificación de las herramientas, pasamos a enumerar los *frameworks* que existen en el mercado. Para ello nos apoyaremos en el estudio realizado por la (Gartner, 2019). En la Figura 2.3 se muestra el Cuadrante mágico de Gartner para las herramientas RPA (Gartner, 2019). Este gráfico está formado por dos ejes, el eje "x" define la categoría integridad de visión, lo que indica la capacidad de poder aprovechar el momento actual del mercado y generar un valor. Y luego está el eje "y", que define la capacidad de ejecución, donde el proveedor tiene la habilidad de ejecutar su visión del mercado de manera exitosa. (TIC, 2016)



Figura 2.3: Cuadrante Mágico de Gartner RPA 2019 (Gartner, 2019).

El gráfico clasifica las herramientas en cuatro grupos: en los “niche players” se encontrarían herramientas que no tienen la suficiente capacidad de visión y ejecución comparándoles con las demás. También están los visionarios, capaces de anticiparse a las necesidades del mercado, pero no son capaces de ejecutar su visión completa debido a la escasez de medios. Los aspirantes o “challengers” ofrecen buenas funcionalidades, pero se centran en menos productos.

Pero el que realmente nos interesa de estos grupos son las herramientas que pertenecen a los líderes, estas marcas tienen una alta capacidad de anticiparse al mercado y habilidad para llevar a cabo su ejecución. Por lo que crean productos que abarcan la resolución de muchos problemas, y además son capaces de adaptarse a la demanda del mercado.

2017 Rank	2018 Rank	Company	2017 Revenue	2018 Revenue	2017-2018 Growth (%)	2018 Market Share (%)
5	1	UiPath	15.7	114.8	629.5	13.6
1	2	Automation Anywhere	74.0	108.4	46.5	12.8
3	3	Blue Prism	34.6	71.0	105.0	8.4
2	4	NICE	36.0	61.5	70.6	7.3
4	5	Pegasystems	28.9	41.0	41.9	4.8
8	6	Kofax	10.4	37.0	256.6	4.4
11	7	NTT-AT	4.9	28.5	480.9	3.4
6	8	EdgeVerve Systems	15.7	20.5	30.1	2.4
7	9	OpenConnect	15.2	16.0	5.3	1.9
9	10	HelpSystems	10.2	13.7	34.3	1.6
		Others	273.0	333.8	22.2	39.4
		<b>Total</b>	<b>518.8</b>	<b>846.2</b>	<b>63.1</b>	<b>100.0</b>

Figura 2.4: Ranking 2018 (EGHAM, 2019).

En la Figura 2.4 se muestra el crecimiento de ingresos conseguido por estas herramientas en los años 2017 y 2018, así como su posición y cuota de mercado.

El top tres de herramientas RPA son: UiPath, Automation Anywhere y Blue Prism. Todas estas herramientas son comerciales y de pago.

Al tratarse de un trabajo de investigación sobre las herramientas RPA, también debemos tener en cuenta herramientas *open source* y analizarlas.

RPA Software (clickable links for download)	Category	Fortune 500 users	Number of contributors on github (as of 10/Aug/2019)	Last commit (as of 10/Aug/2019)
Argos Labs	14 day free trial	Softbank, SK Telecom, Toyota and others	4	12/06/2019
AssistEdge RPA	Freemium			
Automagica	Open source			
Automation Anywhere Community Edition	Freemium	Numerous (25 new Fortune 500 customers won in H1/17)		
Robocorp	Open source			
Robot Framework Foundation	Open source		82	9/08/2019
TagUI - AI Singapore	Open source		9	30/06/2019
Taskt	Open source		4	3/08/2019
UiPath Community Edition	Freemium	Numerous including 8 out of top 10 in Fortune 500		
VisualCron	45 day free trial			
Workfusion RPA Express	Freemium	Numerous		

Figura 2.5: Lista de herramientas open source RPA (AiMultiple, 2020).

En la Figura 2.5 aparecen dos de las herramientas mencionadas, UiPath, Automation Anywhere, debido a que tienen una versión Community Edition. En concreto las cuatro herramientas open source de la lista son: Robot Framework Foundation, TagUi-AI Singapore, Taskt y Automagica. Las dos primeras son las herramientas que más recientemente sus versiones han sido actualizadas, por lo tanto, nos centraremos en ellas dos a la hora de analizarlas.

A continuación, se introduce brevemente las 5 herramientas seleccionadas, con el fin de resumir qué objetivo y qué capacidades ofrecen dichos *frameworks*.

## Automation Anywhere

Esta empresa fue fundada como Tethys Solutions, pero en 2010 fue nombrada Automation Anywhere.

Esta herramienta es capaz de automatizar el negocio entero o partes de él, dependiendo de lo que quiera el cliente. Usa una plataforma escalable, lo que le permite crear *bots* de manera rápida y más adelante aumentar el número según la demanda.

No es necesario cambiar el proceso de negocio, los sistemas o las aplicaciones, ya que, se adapta al entorno. No es necesario un alto conocimiento de programación, gracias a su facilidad *drag and drop*, permitiendo programar sin habilidades de programación.

Otra de las características es la monitorización centralizada, permitiendo el manejo y control de todas las funciones desde un mismo lugar. Además, es seguro y puedes acceder a servicios de auditoría proporcionados por Automation Anywhere, para ayudarte con la automatización de los procesos e incluso encontrar nuevas posibles ideas de automatización.

Esta herramienta es de pago, aunque tiene una versión Community Edition, lo que permite ser utilizada por pequeñas empresas, pequeños equipos de desarrolladores y estudiantes.

## Blue Prism

Blue Prism fue fundada por un grupo de expertos en automatización de procesos en 2001, siendo de las primeras empresas en entrar en el mercado de las RPA.

Esta herramienta ayuda a agilizar y rentabilizar los procesos de oficina manuales y que están basados en reglas, a través de la automatización. Al igual que la anterior herramienta tiene una función de arrastrar y soltar que facilita la automatización de los procesos comerciales. De manera que la implementación puede ser realizada en un periodo entre 4 y 6 semanas. Tiene monitorización para el control de los *bots* y ofrece *feedback* en tiempo real. Blue Prism tiene una alta seguridad, para garantizar la integridad de los datos, el registro de auditoría es almacenado de manera que no haya posibilidad de manipulación. (Guru99, 2020)

Blue Prism no ofrece ninguna versión gratuita de manera indefinida, solo permite una versión de prueba de dos semanas.

## Robot Framework

La primera versión de este software fue desarrollada por Nokia Networks en 2005, pero en el 2008 aparece una nueva versión, la 2.0. Esta versión se lanzó como una herramienta open source, y lo continúa siendo en las nuevas versiones.

Esta herramienta permite la automatización de tests, pruebas de aceptación y RPA. Toda la documentación, el código fuente y otros artefactos se encuentran en GitHub. Tiene una sintaxis de texto simple y se puede utilizar mediante Python o Java. No depende de ninguna aplicación, por lo que es independiente. Consta de múltiples bibliotecas y funciones, con la posibilidad de que pueden usarse en proyectos separados. La ejecución de los test se han de realizar por la línea de comando, los resultados y los logs son guardados en HTML y la salida en XML. (RobotFramework, 2020)

Como se trata de una herramienta de testing, se descarta por no adecuarse a nuestros objetivos.

## TagUI-AI Singapore

TagUI es una herramienta de automatización de procesos mediante líneas de comando. La empresa AI Singapore es la encargada del mantenimiento de esta. Esta herramienta trata de automatizar las tareas repetitivas, que se tienen en páginas webs o en el escritorio de manera que el ordenador pueda realizarlas por ti, bajo una estructura y unas condiciones.

Las características de esta herramienta son varias, pero las que destacan son:

- Automatización web, gracias a una extensión de Chrome es capaz de grabar las acciones web o las acciones que no se ven del buscador.
- Automatización visual del escritorio y de páginas web.
- Cross plataforma, disponible en Windows, Mac OS y Linux.
- Más de 20 lenguajes en los que poder escribir los scripts.
- Integración de R y Python para la aplicación de inteligencia artificial y *machine learning*. (TagUI, 2019)
- Fácil de instalar.

## UiPath

UiPath es una empresa que fue fundada en el 2005 en Rumanía por Daniel Dines and Marius Tîrcă y que ahora se dedica a desarrollar una plataforma para la automatización robótica de procesos.

Esta plataforma es capaz de automatizar cualquier tarea de escritorio o web, ya sea en una máquina virtual o en otro entorno. Todo esto a través de arrastrar y soltar las actividades que se quieran ejecutar. Por lo que no es necesario habilidades de programación para realizar la implementación, es posible utilizar tanto las librerías predefinidas, como compartir y reutilizar automatizaciones realizadas por otros usuarios, ahorrando tiempo y coste de desarrollo.

Destaca su capacidad de extracción de datos con exactitud sobre cualquier aplicación ya sea documentos, páginas webs o del mismo escritorio.

Proporciona asimismo mecanismos para controlar y solucionar errores y excepciones, de manera fácil y amigable.

Es capaz de ejecutar varios robots de manera simultánea. Estos son controlados de forma remota a través de una aplicación web o de escritorio.

Esta herramienta es de pago, aunque tiene una versión Community Edition, lo que permite ser utilizada para pequeños proyectos y estudiantes. Esta versión tiene menos características que la versión de pago, pero suficientes para implementar ciertas tareas básicas de oficina.

## 2.2 Criterios para la evaluación y selección de herramientas

Tras conocer cómo se encuentra el mercado de las herramientas RPA, es necesario establecer un criterio de selección de herramientas para lo cual nos apoyamos en las características recogidas en la Figura 2.6, y así elegir la herramienta adecuada dependiendo de los requisitos del cliente.

Una de estas es la **escalabilidad**, muy importante por la necesidad de una herramienta RPA a adaptarse y responder a las necesidades del negocio, actualizaciones, excepciones o el aumento de operaciones. Cuanto más alta sea la escalabilidad, más capacidad tiene la herramienta para ejecutar diferentes flujos de trabajo al mismo tiempo.

Otro de los parámetros a tener en cuenta es la **seguridad**, la herramienta es un software que va a manejar datos privados. Por lo tanto, la herramienta ha de tener ciertas medidas de seguridad que garanticen que el sistema no pueda volverse vulnerable a ataques externos, mal uso de datos confidenciales y de esta manera evitar problemas de privacidad, problemas que perjudiquen al valor de la empresa.



Figura 2.6: Criterios de elección de herramientas RPA (Kappagantula, 2019).

Otro aspecto importante es la **tecnología** sobre la que funciona dicha herramienta ya que conviene que la herramienta tiene que ser independiente a la plataforma en la que funcione y ser soportada por cualquier tipo de plataforma.

A toda empresa le importa el coste que tendrá dicha inversión, en este coste se incluyen, la configuración del sistema, tarifas de la licencia, y costes de mantenimiento. Por lo que será totalmente necesario que la empresa realice un estudio sobre los beneficios que otorgará dicha herramienta (Kappagantula, 2019).

Por otra parte, una herramienta RPA tiene que ser fácil de usar, y que no requiera muchos conocimientos de programación para democratizar su uso. Por lo que disponer de una **GUI amigable y sencilla** puede ser de gran importancia a la hora de la elección de la herramienta a utilizar.

Otro aspecto a tener en cuenta es si la herramienta está extendida en el sector, esto ayudará a que el proveedor tenga más **experiencia en las necesidades y adopciones del mismo** y lo que facilita la implementación y despliegue del software en la empresa.

El **mantenimiento y soporte** es otro de los parámetros importantes. Es siempre bueno tener proveedores de confianza y asegurarte que den un buen servicio post venta. Un buen soporte ayudará a conseguir mejores implementaciones, mejor capacitación y certificación y mejor relación proveedor-cliente.

No se tiene que dejar de lado el manejo de excepciones, con estas herramientas se busca la automatización dejando de lado la parte manual. Por lo tanto, si los procesos a llevar a cabo son procesos con predisposición a tener errores o excepciones, se ha de buscar una herramienta que se tenga más capacidad de manejar estas excepciones de manera automática sin necesidad del usuario. (10XDS, 2019)

Una vez explicadas las características que se deben tener en cuenta a la hora de escoger una herramienta RPA, es el momento de escoger dos herramientas, una *open source* y otra comercial, para su evaluación. Para ello se implementarán diferentes casos de uso y de esta experiencia se determinará la herramienta con la que implementar los casos de uso de Fagor.

## 2.3 Evaluación de herramientas: UiPath vs TagUI

Las dos herramientas seleccionadas para un estudio más exhaustiva son UiPath y TagUI.

El motivo de la selección de UiPath de entre las herramientas comerciales, fue la gran capacidad para interactuar con las aplicaciones de escritorio, la extracción de datos estructurados y no estructurados y la posibilidad de utilizar la versión Community Edition de manera gratuita.

Este hecho fue el más decisivo y por el que se relegaron las otras dos herramientas comerciales.

En cuanto a la decisión de escoger TagUI, el motivo principal fue la capacidad de esta herramienta de interactuar con entornos web. Como se mencionó previamente, Robot Framework es una herramienta de automatización de tests, por lo que no se ajustaba a nuestros objetivos.

Para su comparación se implementaron algunas automatizaciones simples, con el fin de obtener la experiencia de usuario enfrentándonos a problemas básicos, así como poder realizar un estudio a fondo en cuanto a su funcionalidad.

Pero antes de esto va a describir el procedimiento que se ha seguido para instalar dichas herramientas.

En UiPath se solicitó una sesión Community, por correo electrónico. Recibes un enlace, el cual te dirige a una página y comienza a descargar el ejecutable. Solamente es necesario ejecutarlo y se instalará la herramienta UiPath.

En el caso de TagUI es necesario acceder a GitHub y descargarse el zip correspondiente, siguiendo las instrucciones que aparecen en uno de los archivos dentro del zip. Como se trata de una herramienta que funciona a través de línea de comandos, facilita mucho la programación con ella la definición de una variable de entorno.

Ninguna de las dos herramientas tiene un proceso costoso o tedioso de instalación, permitiendo su instalación a cualquier persona que tenga unas nociones básicas de informática.

### Casos de uso

El primer caso de uso especificado aborda la lectura de un archivo csv con pedidos de un cliente, con el cual se extraerán los datos y rellenará uno o varios formularios en la Web en función del

archivo csv que se trate. En este caso, se utiliza un formulario de Google como ejemplo. Además, se pondrán ciertas condiciones a la hora de rellenar los formularios, como buscar en un base de datos para saber si hay stock o existe el producto.

Para el caso de UiPath, la herramienta permite realizar la lectura del archivo csv (véase Figura 2.7), simplemente arrastrando la actividad Leer CSV sobre la secuencia a realizar. En esta actividad es necesario especificar la dirección donde está localizado el archivo a leer, el método de separación entre datos y la variable donde guardar la tabla de datos.

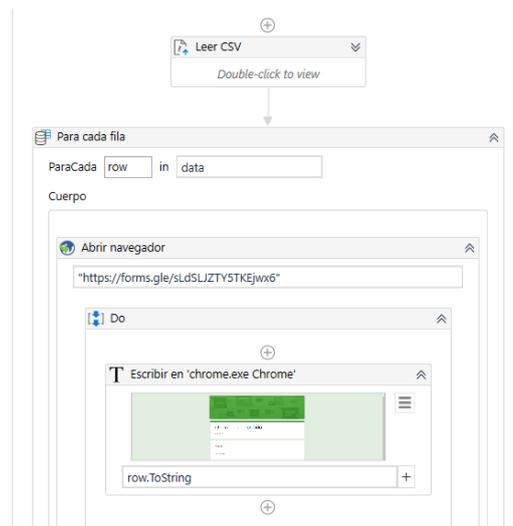


Figura 2.7: Lectura y tratamiento archivo CSV.

La parte relativa a rellenar el formulario es sencilla, debido a la opción “Grabar”, permitiendo hacer una grabación de tu interacción con la página web. Haciendo el uso de clics en pantalla y utilización de variables.

Para acceder a los datos de una base de datos, existe una actividad que permite realizar la configuración de la conexión con diferentes tipos de gestores (SqlServer, MySql, Oracle, etc.). Tras esto, la plataforma es capaz de realizar todo tipos de consultas (*queries*) sobre la base de datos configurada. Dejando la posibilidad de comprobar mediante condiciones “if” si el *input* de los formularios sigue ciertas reglas.

Para este caso en TagUI, la herramienta permite realizar la carga del archivo, utilizando una función específica del programa (Figura 2.8), estas funciones han sido encontradas en el sitio web: <https://www.rpa-sg.org/category/TagUI/>

```

1 //Leer un archivo csv y pasarlo a arrays
2 array_productos = []
3 array_cantidad = []
4 csv_file = 'C:/TagUI/tagui/prueba9.csv'
5 //csv_file = 'prueba10.csv'
6 load '+csv_file+' to lines
7
8 array_lines = lines.split('\n')
9
10 for (n=0; n<array_lines.length; n++)
11 {
12     items = array_lines[n].split(';')
13     array_productos[n] = items[1].trim()
14     array_cantidad[n] = items[2].trim()
15 }
16
17 array_errores = []
18 contador=0
19 for (n=1; n<array_lines.length; n++)
20 {
21     error=0
22
23     https://forms.gle/sLdSLJZTY5TKEjwx6
24     producto= array_productos[n]
25     cantidad= array_cantidad[n]
26     if (cantidad < 500 && producto != "")
27     {
28         enter //input[@name="entry.2005620554"] as '+producto+'
29         enter //input[@name="entry.1166974658"] as '+cantidad+'
30     }
31 }

```

Figura 2.8: Código para leer y rellenar formulario.

Como se puede observar, es necesario tener conocimientos de programación ya que utiliza el lenguaje JavaScript. El problema aparece a la hora de realizar la programación con la web, debido a que se ha de ir inspeccionado los elementos con los que se quiere interactuar, en la imagen Figura 2.8 esta corresponde con una entrada de input.

Otra opción de interactuar con el formulario es a través del formato:

*“comando+nombreElemento” →enter Your answer as ‘+producto+’*

El problema que surge aquí es que no es capaz de realizar distinciones con esta nomenclatura en cuanto hay dos inputs con el mismo nombre, de tal manera que solo rellena el primer elemento.

En relación a la utilización de la base de datos, solo se ha conseguido utilizar MySQL, aunque se puede realizar con otros gestores de bases de datos (Soh, 2018).

Después de implementar el caso de uso con las dos herramientas llegamos a la conclusión, que cualquiera de las dos herramientas cumple los objetivos deseados. En UiPath es algo más fácil debido a su interfaz gráfica lo que permite realizar esta tarea con pocos conocimientos de programación. Con TagUi es necesario tener experiencia con JavaScript y saber cómo funcionan los comandos, siendo algo más complicado si no tienes conocimientos y experiencia en programación.

El segundo caso de uso que se propuso consistía en mover archivos de un directorio a otro, devolviendo por pantalla un mensaje con el número de archivos que van a ser desplazados.

En el caso de UiPath se han de crear diferentes variables, una para guardar el nombre de los archivos en el directorio y otra para guardar el path del directorio. Tras esto se utiliza la actividad “Para cada” sobre el array de Strings, y dentro del cuerpo de esta actividad se añade la actividad “Mover Archivo” con el respectivo path donde se quieran mover los archivos. Añadiéndose al final de la secuencia la actividad Bandeja de mensajes, esta actividad muestra por pantalla el objeto deseado, en este caso el número de archivos movidos.

En cambio, con la herramienta TagUi no ha sido posible implementar esta tarea, debido a que esta herramienta no tiene ninguna función que permita interactuar con elementos del escritorio.

Por lo tanto, TagUi no es capaz de interactuar con los sistemas de ficheros, para poder mover o copiar archivos de un directorio a otro. Dejando a UiPath como única solución para esta tarea, además, resulta fácil de implementar.

El tercer caso de uso consistió en extraer datos de una página web que tiene un catálogo de libros y crear un archivo csv donde se almacenen los datos extraídos. Los datos que se quieren extraer son el nombre del libro y la URL.

La implementación con UiPath, no reviste ninguna complejidad, pero tiene un problema. La extracción de datos es Ad hoc, lo que quiere decir que tienes que realizar este proceso para cada página de la que quieras sacar información. Se ha de abrir la página de la que se quiere extraer y luego ejecutar la extracción. Los datos se guardarán en una tabla de datos, que luego se llevarán a Excel mediante una actividad llamada "Crear CSV".

En TagUI, en cambio la implementación es más complicada, es necesario identificar el patrón que siguen los elementos de la lista y poder recorrer elemento por elemento obteniendo los datos necesarios.

En el caso de que se quiera extraer datos de una página que mantenga la misma estructura, es posible reutilizar el script para próximas automatizaciones. En la figura 2.9 se puede ver el código usado para realizar esta tarea.

```
num_books = count('//td/a//span[@itemprop="name"]')
echo "num_books = " + num_books

csv_header = '"Sno", "Title", "URL"\r\n'
dump csv_header to test1.csv

for (n=1; n<=num_books; n++)
{
    read (//td/a//span[@itemprop="name"]['+n+']) to book
    echo "book = " + book

    read (//td/a/@href)['+n+'] to url2
    echo "url2 = " + url2

    record = '"' + n + '", "' + book + '", "' + url2 + '"'
    write record to test1.csv
}
```

Figura 2.9: Código para extraer datos de una página web.

La lectura del correo es una tarea recurrente en tareas de oficina, por lo que otro caso de uso que se especificó fue leer los mensajes recibidos a un correo electrónico, de manera que sean guardados en un directorio del ordenador, almacenando los archivos adjuntos en caso de tenerlos.

Respecto a UiPath, la herramienta tiene una actividad que permite devolver una lista de correos electrónicos y otra que permite ir mensaje por mensaje guardando los mensajes en el directorio deseado.

En cuanto a TagUI, ha sido posible la lectura del último correo recibido, además de guardar los archivos adjuntos en el caso de que los hubiera. Por lo que no se ha podido completar la tarea en su totalidad, dejando sin realizar la lectura completa del buzón de entrada. Se ha buscado información sobre este tema o similares en Internet, pero al tratarse de un framework con pocos

usuarios no se ha obtenido ninguna posible solución al mismo. Esta herramienta no ofrece soporte, la ayuda en foros de la comunidad es escasa.

Por último, se propuso la extracción de datos de un PDF, sin importar si los datos a extraer aparecieran en formato texto o que el texto estuviera contenido en una imagen.

Uno de los desarrolladores de TagUI explica en uno de sus comentarios, que esta funcionalidad aún no ha sido implementada y que no se cree que vaya a implementarse. Pero esta herramienta es capaz de simular en un documento abierto en la web la pulsación de teclas, permitiendo así copiar texto del documento al archivo que se desee.

UiPath, por defecto, no contiene los paquetes necesarios para realizar esta tarea, pero como se trata de una herramienta con una librería extensa, se puede descargar dichos paquetes. Se necesitarían dos actividades diferentes, una que solo leyese texto del archivo PDF, y la otra que con la utilización de un motor OCR (reconocimiento óptico de caracteres) sea capaz de interpretar el texto de las imágenes.

A continuación, se resumen las características más relevantes de estas herramientas y la experimentación realizada en forma de Tabla 2.1, para facilitar su comparación.

<u>Características</u>	<u>UiPath</u>	<u>TagUI</u>
Interfaz Gráfica.	Si.	No.
Open Source.	No.	Si.
Simultaneidad de Robots.	La versión community permite 2, mediante UiPath Orchestrator.	No creo que haya límite, lo que te permita tu dispositivo.
Formato del archivo donde se guarda el proceso.	.xaml	.js
Lenguaje de programación.	Se utiliza VB.net o #C	JavaScript
Conocimientos de programación.	No es necesario. Gracias a su <i>drag and drop</i> .	Es necesario tener un conocimiento medio.
Instalación.	Fácil, desde un enlace al correo.	Fácil, desde github.
Soporte de ayuda.	Si, cantidad de foros con diferentes soluciones. En el caso de tener la versión comercial se tendría servicio al cliente.	Si, uno de los desarrolladores va respondiendo a dudas cuando tiene tiempo libre. Escasas soluciones a problemas.
Machine Learning.	Si, a través de Python	Si, a través de Python
Gestión de paquetes.	Si, se pueden añadir diferentes paquetes y dependencias al proyecto.	Si, es posible agregar las funciones de Sikuli <sup>1</sup> .
Funcionalidades		
Extracción de datos Web.	Tareas sencillas ad hoc mediante clics. Realizando una búsqueda de patrones en los datos a extraer.	Ad hoc pero permite extracción de datos de páginas con la misma estructura. Tarea no tan intuitiva, es necesario buscar el selector de cada elemento.
Extracción de datos Excel.	Sencilla, función implementada, mete los datos en una tabla.	Media, se carga el archivo y se va introduciendo en array cada columna.
Interacción con la Web.	Si, de manera visible, tiene diferentes extensiones de navegadores, e invisible ejecutando también en segundo plano.	Si, se pueden grabar las acciones visibles y las que no lo son, tiene una extensión de Chrome.
Interacción con el escritorio.	Si, permite grabar directamente las acciones a realizar.	Si, pero no es fácil de implementar esta tarea, es necesario buscar la imagen en la cual quieres hacer clic.
Interacción con aplicaciones de escritorio.	Si.	No.
Cargar y escribir Excel	Si.	Si.
Rellenar formularios	Fácil, hacer clics en los inputs y llamar a las funciones para introducir los valores de las variables.	Fácil, pero es necesario el selector de cada elemento, tener los datos que se quieren rellenar en variables.
Integración con otros lenguajes.	Se puede invocar código de Python.	Se puede hacer uso de Python y R.
Leer correo	Permite de forma fácil obtener los mensajes sin necesidad de interactuar con la web.	Se puede realizar, pero de manera más costosa, es necesario ir mensaje por mensaje.

Tabla 2.1: Comparación entre TagUI y UiPath.

### 3. Especificación de requisitos

Esta sección recoge los requisitos de los casos de uso propuestos por Fagor. La reunión para la recogida de dichos requisitos fue el 18 de noviembre de 2019, los participantes de esta reunión fueron: Alberto de Castro; Marta Zorrilla; Rosa Martínez y José Ramón Grau. Los dos últimos participantes como representantes de Fagor.

Fagor propone dos casos de uso habituales, en los cuales sería interesante aplicar las tecnologías RPA para poder automatizar la ejecución de estas tareas.

#### 3.1 Asignación de vehículo a pedidos recibidos por formulario web

Fagor describe este caso de uso como la cumplimentación de CMR/Carta de Porte, desde mail o mensaje telefónico. Ejemplo. En Transportes Pepe los clientes dejan los encargos vía telefónica, de 8:00 a 15:00, tienen un operador que recoge las llamadas y completa los campos del formulario CMR (origen, destino, nº de bultos, peso, etc.) con esos datos busca en Flotasnet (el sistema de gestión de flotas) un vehículo que esté libre y por la zona de origen y completa el formulario con la matrícula y el nombre del conductor.

Para automatizar el proceso y pasar a ser 24x7, se ha propuesto la siguiente solución:

Se quiere incorporar una funcionalidad que facilite a las empresas solicitar un transporte a cualquier hora del día. Para ello, se proporcionará un formulario web donde se recoja la información que debe rellenar el solicitante. A continuación, se conectará al software FlotasNet (vía consulta a BD sqlserver) para que, de acuerdo al origen, destino y tipo de mercancía (frigorífico, peligrosas, etc.) especificada, se asigne un conductor y la matrícula de un camión disponible. El resultado conformará parcialmente el documento CMR que deben llevar los camioneros. Para facilitar la confección de este documento, la información recogida se almacenará en FlotasNet en formato JSON, asociado a la matrícula del camión y fecha-hora de la solicitud.

El solicitante después de enviar los datos del pedido recibirá la carta CMR por correo electrónico. La carta se enviará sin los datos de facturación ya que se desconocen, los demás datos serán obtenidos del formulario y la base de datos. Todas las acciones que se desencadenan se anotan en un log para trazabilidad y mejora del proceso. En caso de fallo, además se registra el motivo y se envía correo electrónico al responsable y notificación al solicitante de que llame por tfno. por problemas técnicos.

Los datos que se han de recoger en el formulario son:

1. Remitente
  - a. Nombre
  - b. Dirección (tipo vía, vía, localidad y CP)
  - c. País (proporcionar lista desplegable)
2. Destinatario
  - a. Nombre.

---

<sup>1</sup> Sikuli: Es una herramienta que permite la automatización a través de interactuar con los elementos de la interfaz.

- b. Dirección (tipo vía, vía, localidad y CP).
  - c. País (proporcionar lista desplegable).
3. Lugar entrega de la mercancía.
4. Lugar de carga de la mercancía.
5. Tipo de mercancía.
  - a. Estándar.
  - b. Frio.
  - c. Ultracongelado.
  - d. Peligrosa.
6. Documentos anexos:
  - a. Archivo (fichero).
  - b. Descripción.
7. Número de bultos.
8. Peso bruto en Kg.
9. Volumen en m<sup>3</sup>.
10. Instrucciones del remitente.
11. Forma de pago, elegir entre:
  - a. Tarjeta.
  - b. Transferencia.
  - c. Metálico.

El formulario CMR ha sido proporcionado por Fagor en formato Word.

El esquema de la base de datos Flotasnet y un script con la creación y relleno de dicha base de datos, ha sido proporcionado por Fagor. Se simulará la interacción con Flotasnet a través del uso de una base de datos con el mismo esquema que Flotasnet. En un futuro se hará uso de un microservicio que contemple las reglas de asignación para un vehículo disponible y adecuado a la mercancía a transportar.

### 3.2 Formato y carga de fichero de facturación

Este caso tiene por objeto automatizar el proceso de revisión de los ficheros excel (o pdf incluso) de facturación de las expediciones que realizan los transportistas que llegan vía correo electrónico. La tarea se realiza ahora manualmente y consiste en revisar los datos de las Excel para comprobar que están todas las columnas rellenas y que la información que contienen es coherente y mediante un pequeño script subirlos a la base de datos.

La solución propuesta para automatizar este proceso se describe a continuación.

Se define una palabra clave para identificar los correos que incluyen archivos de facturación, p.ej. “e-Facturación”.

Los correos que en el asunto lleven “e-Facturación” y sean dirigidos a la cuenta de correo electrónico “xxx” se procesan de acuerdo a:

1. Se verifica que el fichero Excel tiene las columnas definidas según la plantilla que se ha proporcionado por Fagor.
2. Se establece la comprobación del contenido de cada columna de acuerdo a las reglas que se han especificado:
  - a. Que no haya NULL.
  - b. Formatos correctos (textos, números y fechas).
  - c. Que la columna Total sea la suma de las seis columnas anteriores.

3. Si consecuencia del proceso de verificación se encuentran errores, se recogerán los problemas encontrados en un fichero de log que se dejará en un directorio XXX y con nombre YYY (nombre del fichero y fecha de comprobación) y se notificará para su revisión al correo electrónico ZZZ.
4. En caso de que el fichero se pueda cargar, se ejecutará el jar `subidaArchivosExcel.jar` que simula el jar ejecutado por FAGOR para la subida de archivos a la base de datos.

## 4. Diseño e implementación

Conociendo los requisitos de los casos de uso, se procedió a su diseño e implementación.

Se comienza explicando el diseño del primer caso de uso definido en el apartado 3.1. En la Figura 4.1 se puede observar las tareas que se han de realizar y en qué orden, así como las diferentes aplicaciones con las que la herramienta RPA tiene que interactuar.

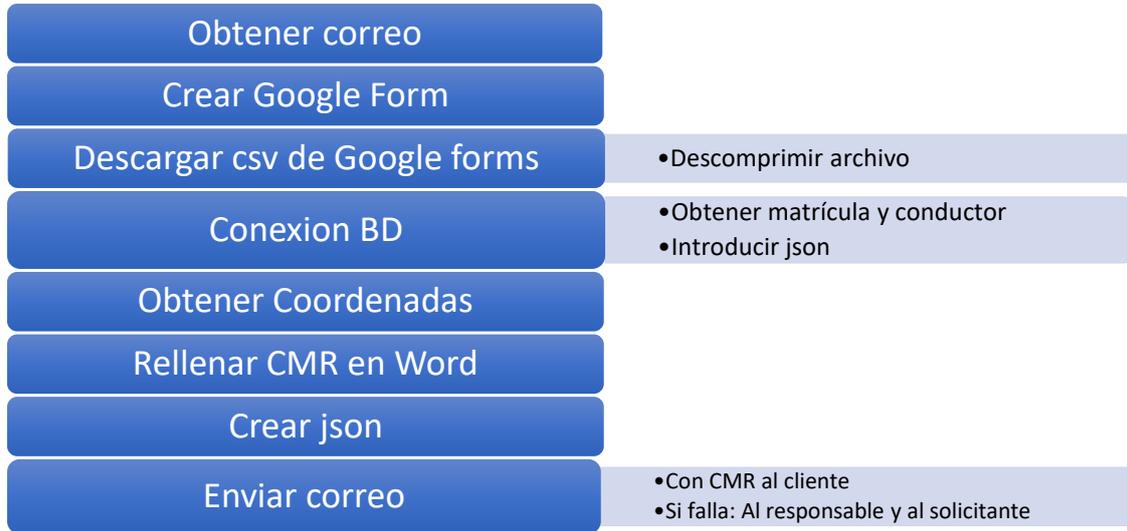


Figura 4.1: Organigrama para completar asignación de vehículo a pedidos recibidos por formulario web.

En este caso, la herramienta tiene que interactuar con la base de datos SQLServer, para acceder a los datos necesarios sobre el vehículo y conductor que han de realizar el pedido. Además de introducir los datos del pedido en formato json.

La interacción con la carta de porte será mediante el programa Microsoft Word, se ha de rellenar con los datos del formulario y los obtenidos de la base de datos.

Además, deberá interactuar con Google Chrome, con el objeto de descargarse el archivo csv del formulario web y obtener las coordenadas del lugar de origen de esta URL: <https://www.coordenadas-gps.com/>.

A continuación, se muestra el diseño del proceso asociado al (véase Figura 4.2) segundo caso de uso descrito en el apartado 3.2.

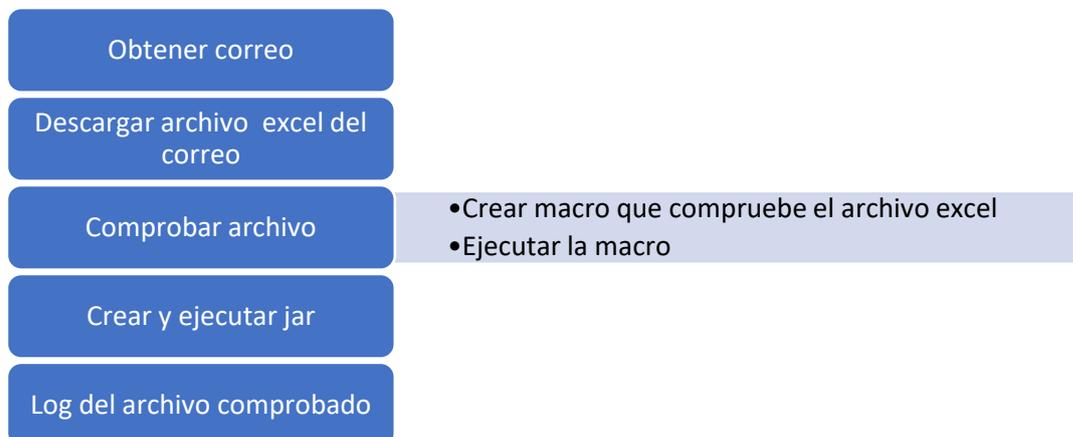


Figura 4.2: Organigrama para completar formato y carga de fichero de facturación.

Para la realización de este caso, es necesario que la herramienta interactúe con el programa Microsoft Excel para poder comprobar que el formato del documento es correcto. Esto se realizará mediante la ejecución de una macro, que ha sido previamente programada.

La herramienta tendrá que ejecutar un Jar, que simule la carga del archivo a la base de datos.

Con la información recopilada sobre los elementos con lo que la herramienta RPA ha de interactuar, es necesario seleccionar cuál de las dos herramientas vamos a utilizar para llevar a cabo la implementación de estas tareas.

Se decidió la utilización de UiPath debido a la gran capacidad de interacción con aplicaciones de escritorio y posibilidad de grabar procesos, descartando TagUI por sus limitaciones para abordar estas tareas. En este caso es obligatorio la interacción con Word y Excel, acción que UiPath permite realizar.

La implementación de los dos casos de uso en UiPath se ha realizado a través de secuencias. En la Figura 4.3 se puede observar la estructura que tiene el conjunto de las secuencias programadas para cada caso de uso.

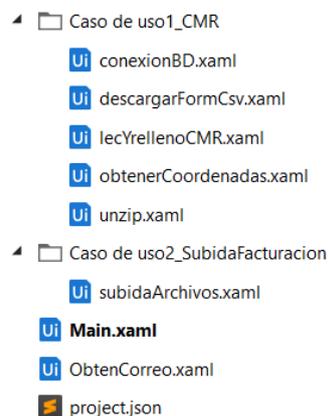


Figura 4.3: Secuencias creadas en UiPath para ambos casos de uso.

Las secuencias “Main.xaml” y “ObtenCorreo.xaml” pertenecen a los dos procesos. El primero, tiene por objeto la ejecución del proceso de manera indefinida. Este hace uso de un bucle de manera que ejecute “ObtenCorreo.xaml” cada 10 segundos.

Esta secuencia lee la bandeja de entrada del correo electrónico deseado, con la función *GetIMAPMailMessages*. La primera vez que se realiza esta petición es necesario activar en la cuenta de correo los permisos para interactuar con aplicaciones externas. Esta actividad guarda todos los mensajes no leídos en una variable *List<MailMessages>* que es tratada mediante un bucle *for each*. Dentro de este bucle para cada mensaje de la lista se comprobará el asunto, y si contiene “Formulario Pedido” y el valor de una variable que indica que es la primera vez que se descarga el csv es “0”, entonces se invocan a las secuencias: “descargarFormCsv”, “unzip” y “lecYrellenoCMR”. Estas serán explicadas más adelante. Además, es necesario borrar los archivos descargados y ya procesados. Por ello, la última tarea a realizar es poner el valor de la variable a “1” para que solo realice la descarga una sola vez, ya que en el archivo csv contiene todos los pedidos del formulario, y evitar de este modo que se procese dos veces el mismo pedido.

Si no se encontrase en el asunto del mensaje “Formulario Pedido”, utilizando otra condición se busca si el asunto contiene “e-Facturación”. Si contiene este String, entonces los archivos

adjuntos se descargan y se guarda el nombre del archivo en una variable y en otra el path donde se encuentra el archivo. Como se van a realizar cambios sobre el archivo para comprobar su formato, se realiza una copia del archivo, nombrándolo “pruebaAuto.xlsm”. Tras esto se invoca a la secuencia “subidaArchivos.xaml” que tiene como parámetros el path y el nombre del archivo.

En la Figura 4.4 se puede observar cómo queda codificado en UiPath.

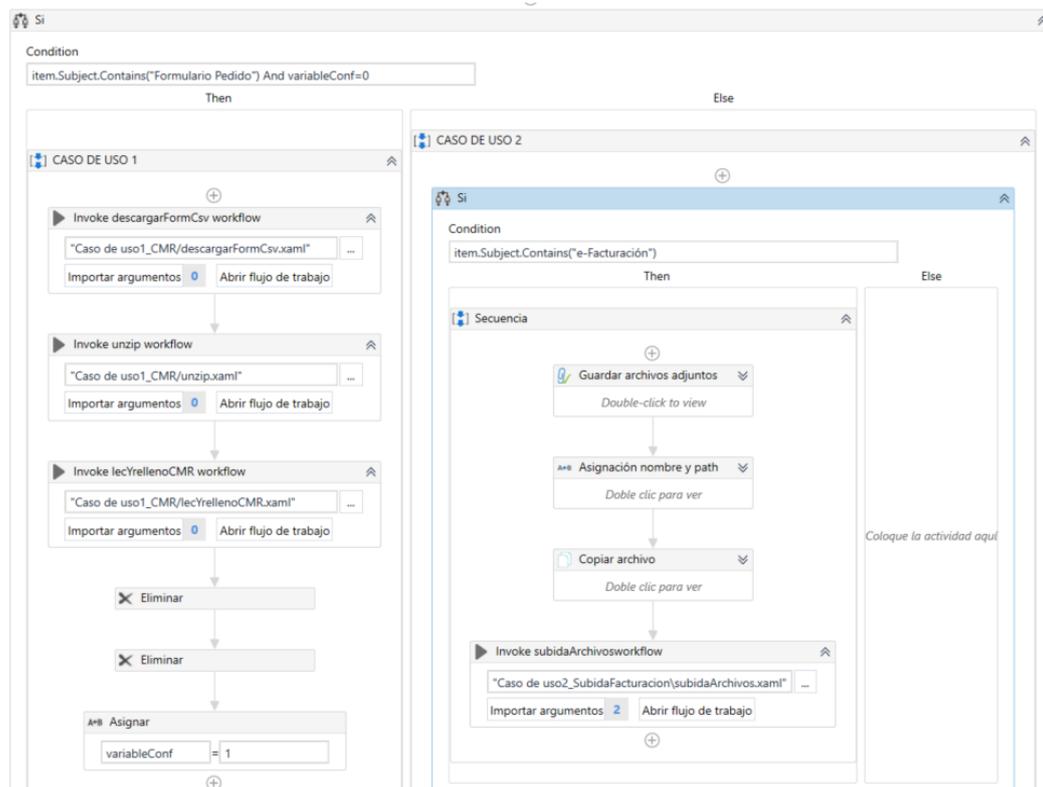


Figura 4.4: Visualización de la codificación de la secuencia “ObtenerCorreo”.

Vamos a realizar un análisis de las secuencias del primer y segundo caso de uso independientemente, siguiendo su orden de ejecución en el proceso. También se incluirán todos los artefactos que han sido necesarios para la implementación.

## 4.1 Asignación de vehículo a pedidos recibidos por formulario web

Lo primero que se va a realizar es la descarga del csv con los datos del formulario, por lo tanto, el primer artefacto a crear es un formulario Web. Se va a utilizar Google Forms, ya que, permite guardar todas las respuestas en un documento online y descargar dichas respuestas en un archivo csv.

El formulario sigue ciertas reglas, es necesario introducir un correo electrónico para poder comunicarse con el cliente; además, los campos tienen que cumplir ciertas expresiones regulares para comprobar si los datos están siendo bien introducidos.

El formulario creado se encuentra en: <https://forms.gle/xGuXM2cENCwj8PLm9>. Una vez creado y listo para empezar a recibir peticiones, es el momento de programar la secuencia “descargarFormCsv”, que se encarga de descargar el archivo csv con las respuestas del formulario. Para ello se utiliza una de las funciones de UiPath que se encarga de grabar el

proceso que se realiza con el ratón por el escritorio. En este caso se graba cómo se accede a la página de respuestas del formulario y se hará clics en diferentes puntos de las interfaces dirigidas a la descarga del archivo y borrar las respuestas que han sido ya descargadas. Y, por último, cerrar la página.

La siguiente secuencia (véase Figura 4.5) es la que se encarga de descomprimir el archivo descargado, en este caso se ha utilizado la invocación del método *ExtractToDirectory* asignándole el path donde se encuentra el archivo comprimido y el path donde se quiere descomprimir.

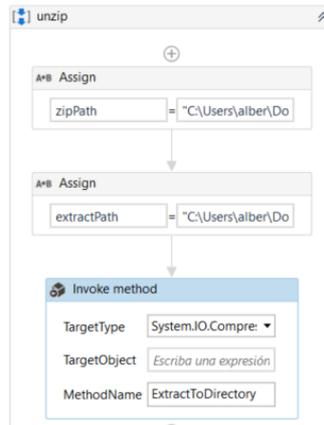


Figura 4.5: Ejemplo de descompresión archivo en UiPath.

Antes de entrar en detalles con la secuencia “IecYrellenoCMR” es necesario explicar dos secuencias que son utilizadas dentro de esta.

Para poder encontrar el vehículo más cercano al punto de origen son necesarias las coordenadas donde se encuentra posicionado. La base de datos ofrece las últimas posiciones de los vehículos. Con el objeto de conocer las coordenadas del punto de recogida del pedido, se crea la secuencia “obtenerCoordenadas”, obteniendo la latitud y longitud de dicha posición a través de una página web. Tiene como parámetro de entrada el nombre de la calle donde se ha de recoger el pedido y devuelve como parámetro de salida las coordenadas de la localización. Al igual que se realizó anteriormente con la descarga del archivo csv, va a ser necesario interactuar con la página web: <https://www.coordenadas-gps.com/> que nos ofrece la localidad a la que pertenece esa geoposición.

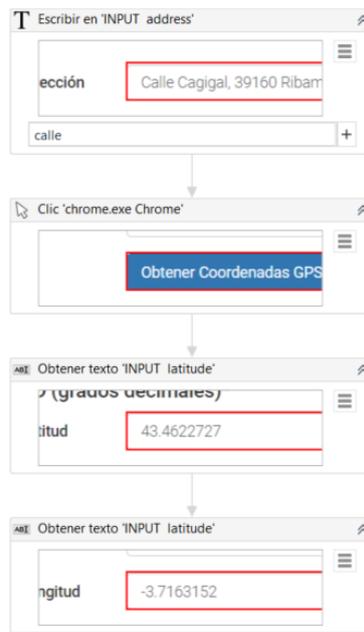


Figura 4.6: Obteniendo coordenadas del punto de recogida.

En la Figura 4.6 se puede observar la utilización del parámetro de entrada ‘calle’ para buscar las coordenadas del punto de recogida, después se guarda la latitud y longitud en variables. El valor absoluto de estos valores va a ser sumado y devuelto como parámetro de salida de la secuencia.

La siguiente tarea a realizar es comparar la localización obtenida anteriormente con la de los vehículos de la base de datos. Y para ello es necesario acceder a la base de datos mediante la secuencia “conexionBD”. Los dos parámetros de entrada son la localización de recogida y el país de origen. Tiene como salida el nombre del vehículo y el nombre del conductor.

La configuración de conexión con la base de datos se realiza como se observa en Figura 4.8.

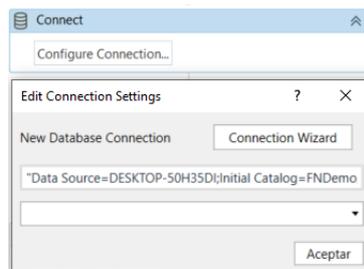


Figura 4.7: Configuración de conexión con la base de datos.

Tras la conexión ya se puede realizar la *query* que nos devuelva los vehículos de la base de datos. Aquí entra en juego el parámetro de entrada país, permitiendo acotar la búsqueda a todos los vehículos que se encuentren en ese país. La consulta que se realiza es la siguiente:

```
'select Localidad,Id_Conductor,Id_Vehiculo,matricula,conductor,Latitud,Longitud from Flota where Localidad like @pais And Id_Conductor Is Not Null'
```

Además, se aplica otra restricción que es que el vehículo tenga conductor. El resultado de dicha consulta se guarda en una tabla de datos que luego va a ser utilizada para determinar que vehículo está más cercano.

Para conocer qué vehículo es el más cercano se recorren todos los vehículos mediante un bucle *for each*, el pseudocódigo se puede ver en la Figura 4.8.

```

for each row in respuestaQuery
  obtiene latitud y longitud
  se suma el valor absoluto de estos valores
  distancia=valor absoluto de la resta del origen-localizacion del vehiculo
  if distanciaMenor>distancia
    distanciaMenor=distancia
  se guardan los valores del conductor y matricula en las variables
fin if

```

Figura 4.8: Pseudocódigo del algoritmo para conocer el vehículo más cercano.

Después de ejecutarse el algoritmo en la aplicación, en los parámetros de salida tendremos el conductor y la matrícula del vehículo para ser utilizadas cuando se rellene la carta de porte. Por último, se ha de cerrar la conexión con la base de datos, dando por finalizada la secuencia.

Conociendo ya la función de estas dos secuencias podemos describir la secuencia “IecYrellenoCMR”. Se empieza leyendo el archivo de csv con las respuestas del formulario descargadas y para cada fila del archivo se van a realizar ciertas actividades. Para poder utilizar la carta de porte de manera segura, se realiza una copia de la original, de manera que nunca sea modificada. Con objeto de utilizar la función remplazar texto de Word es necesario añadir en cada apartado del documento (celda a rellenar) una palabra clave que luego será remplazada por la función. Por ejemplo: “NOMBREEMITENTE” en el apartado de remitente.

Pero antes de realizar esta tarea se ha de guardar todos los datos del formulario en sus correspondientes variables, en la Figura 4.9 se observa la actividad para realizarlo.



Figura 4.9: Obtención de valores de una columna.

Teniendo todos los valores en las variables, solo faltan de conocer el conductor y la matrícula, por lo que es necesario invocar a las secuencias anteriormente explicadas. Ahora ya se puede empezar a rellenar el CMR utilizando la actividad *WordApplicationScope*, que su objetivo es abrir el documento Word deseado y realizar actividades sobre él. En este caso se remplazará las palabras claves por los datos que se obtuvieron, dejando el archivo CMR listo para ser enviado al cliente.

Otra tarea que se necesita es la inserción de estos datos en formato json, por lo tanto, es necesario crear un String con este formato y los respectivos datos que ha de tener. Este sería un pequeño ejemplo del String a construir:

```
"{"+""Matricula""+v+t+matricula+t+c+""Conductor""+v+t+conductor+t+c+""fecha""+v+t+fecha+...+"}"
```

El valor de la variable ‘v’ es “.”, el de ‘t’ es “”””” y el de ‘c’ es “,”.

El String será guardado en una variable para poder hacer uso de él y poder ir guardando todos los datos en un archivo json llamado “pedidos.json”. Además de para ser utilizado cuando se realice el *insert* en la base de datos. Como el objetivo es la inserción de los datos a la base de datos se va a utilizar la función OPENJSON de SqlServer mediante la ejecución de una *query* (Microsoft, 2019). Se van a crear dos variables con el fin de guardar el script que se necesita ejecutar para la inserción (véase la Figura 4.10) de los datos. El formato del script que se ha de guardar en las variables es el que se ve en la Figura 4.11.



Figura 4.10: Ejecución de la inserción del json.

```

select * from Pedido
declare @json varchar(max)=json
INSERT INTO Pedido (Matricula,...)
Select Matricula,...
FROM OPENJSON(@json)
WITH (Matricula varchar(20),
      conductor varchar(20),
      .
      .
      descripcionDoc varchar(40)
)

```

Figura 4.11: Script que inserta el json.

Para finalizar el caso de uso nos falta enviar el correo electrónico con la carta CMR sin los datos de facturación, para ello es necesario comprimir el archivo en un zip y adjuntarlo al mail que va a ser enviado, véase en la figura 4.12.

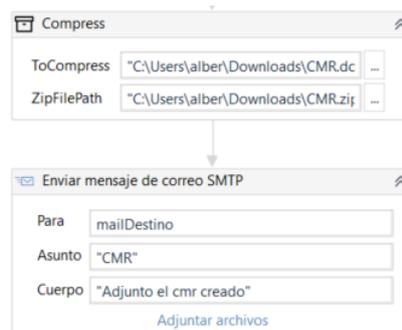


Figura 4.12: Comprido y envío de CMR.

Con esta última tarea se da por finalizada todas las tareas y secuencias que han sido necesarias para realizar el primer caso de uso. En el apartado 5. Pruebas se podrá comprobar el funcionamiento de este.

## 4.2 Formato y carga de fichero de facturación

Este caso de uso utiliza la secuencia “ObtenCorreo.xml”, previamente descrita y que se encarga de comprobar la llegada de correos que contengan en el asunto la palabra “e-Facturación” y que se descarguen sus archivos adjuntos. A continuación, se procede a la comprobación de los ficheros descargados para lo cual se tiene que programar un macro en Excel. UiPath nos planteó un problema inicialmente, dado que cuando abre el documento Excel no permite registrarle la macro, por ello se tuvo que crear un archivo con extensión “.bas” (fichero con código en lenguaje de programación BASIC).

En este archivo se crean las variables que va a utilizarse en la verificación y añade columnas con las que se va a comprobar las diferentes reglas que se han de cumplir. Lo primero que se comprueba es si las fechas tiene un formato correcto (ver código asociado en la Figura 4.13) donde la variable ‘numero’ es el valor obtenido de aplicar la función *CONTARA(A:A)*. Esta función devuelve el número de filas que tienen datos de la tabla, para poder aplicar la comprobación de las fechas a toda la columna.

```

For Contador = 2 To numero
    s = "AI" & Contador
    e = "D" & Contador
    Range(s).Value = IsDate(Range(e).Value)
    s = "AJ" & Contador
    e = "G" & Contador
    Range(s).Value = IsDate(Range(e).Value)
    s = "AK" & Contador
    e = "AF" & Contador
    Range(s).Value = IsDate(Range(e).Value)
    s = "AL" & Contador
    e = "AG" & Contador
    Range(s).Value = IsDate(Range(e).Value)
Next

```

Figura 4.13 Código que comprueba las fechas.

Para saber si el total es la suma de las demás celdas se aplica la expresión:

`=SI(IGUAL(SUMA(Y2:AD2);AE2);"VERDADERO";"FALSO")`

Es necesario aplicar esa expresión a todas las celdas de la columna y se hace mediante el código de la Figura 4.15 donde se puede ver que se usa una variable para crear el rango donde se tiene que aplicar.

```

e = "AM2:AM" & numero
Range("AM2").Select
ActiveCell.FormulaR1C1 = _
"=IF(EXACT(SUM(RC[-14]:RC[-9]),RC[-8]),""VERDADERO"", ""FALSO"")"
Selection.AutoFill Destination:=Range(e), Type:=xlFillDefault
Range(e).Select

```

Figura 4.14: Código que comprueba que la suma de ciertas celdas es igual al total.

También es necesario comprobar si ciertas celdas son texto, para ello se utiliza la expresión `EsTexto` de Excel como se puede ver en la Figura 4.15.

```

str = "AN2:AN" & numero
Range("AN2").Select
ActiveCell.FormulaR1C1 = "=ISTEXT(R2C2:R1001C2)"
Range("AN2").Select
Selection.AutoFill Destination:=Range(str), Type:=xlFillDefault
Range(str).Select

```

Figura 4.15: Código que comprueba si el número de facturación es texto.

Para finalizar la macro se añade una columna que sirva para saber si la fila de datos no cumple alguna de las reglas de formato y si es así, no se podría subir el archivo. Véase Figura 4.16.

```

str = "A02:A0" & numero
Range("A02").Select
ActiveCell.FormulaR1C1 = "=COUNTIF(RC[-6]:RC[-1],""FALSO"")>0"
Selection.AutoFill Destination:=Range(str), Type:=xlFillDefault
Range(str).Select

```

Figura 4.16: Código que comprueba si la fila tiene errores de formato.

Con todo esto ya estaría creado el archivo que contiene la macro. Ahora se debe incorporar en el Excel cada vez que abra un archivo, para esto se utilizó la herramienta de grabación. Una vez abierto el documento que se quiere comprobar con la actividad `ExcelApplicationScope` se realiza la grabación del proceso para importar el fichero "aa.bas". En la Figura 4.17 se ve una parte de esta automatización.

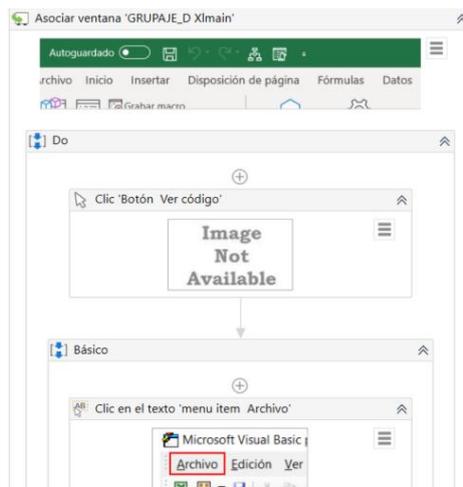


Figura 4.17: Ejemplo de la importación de una macro a Excel.

El proceso en este punto está listo para ejecutar la macro a través de la actividad *ExecuteMacro* pasándole el nombre de la misma, “AutomatizacionSubida”. Tras ejecutarse la macro se necesita leer la columna que confirma si no se puede subir el archivo. Y para ello se obtiene el nombre de la hoja con la actividad *ExcelGetWorkbookSheet*. Con el nombre de la hoja podemos leer la columna “AO2” gracias a la actividad *ExcelReadColumn* que guarda los valores en un variable de tipo *IEnumerable<Object>*.

A continuación, se recorre dicha columna comprobando si su valor es “True”, y en ese caso se guarda en una variable ‘log’ diciendo que en esa fila ha ocurrido un problema, y pasa a la siguiente fila.

Por último, se invoca el jar creado a través de Eclipse que guarda en un archivo llamado “logArchivosSubidos.txt” el nombre del archivo y la fecha de comprobación.

Después de recorrer toda la columna se comprueba si la variable log tiene algún valor, en el caso de no tenerlo se ejecutaría el jar creado anteriormente y en caso contrario, se crea un archivo de log con los problemas que se tuvieron en la comprobación, como se puede ver en la Figura 4.18.

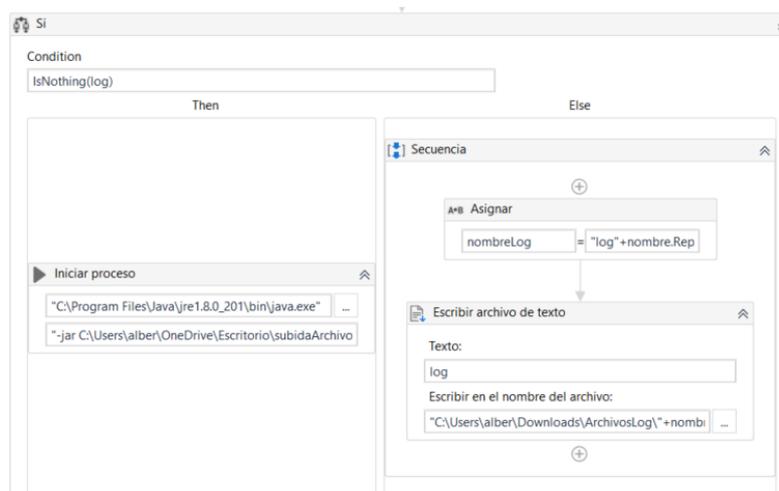


Figura 4.18: Código para subir los archivos o crear un log.

Con esto se finalizaría la implementación del segundo caso de uso, dejando por finalizado este apartado de diseño e implementación.

## 5. Pruebas

Cualquier implementación software no está libre de fallos, por eso es necesario realizar pruebas con el fin de minimizar su número. En este caso, se van a realizar pruebas de ejecución, ya que, lo que nos interesa es saber si cumple con su tarea y en qué tiempo se realizan.

Las primeras pruebas fueron realizadas por el desarrollador, los resultados de dichas pruebas se muestran en la Tabla 5.1. Para la ejecución de las mismas se ha utilizado un dispositivo distinto al ordenador que aloja los robots software.

Para comprender la tabla se va a explicar uno de los casos. Por ejemplo, en el caso de prueba 4 han sido rellenados cuatro formularios de pedido y al mismo tiempo se han enviado tres archivos de facturación. Mientras se trataban, se ha rellenado un formulario más y cuatro archivos de facturación más. En cuanto a los archivos de facturación se han ido alternando archivos con el formato correcto e incorrecto. Durante la ejecución del proceso no ha ocurrido ningún fallo con una duración de cuatro minutos y cincuenta segundos.

Prueba	Respuestas de formularios	Archivos de facturación	Fallos	Tiempo
1	Dos	Uno de 20 filas, otro de 720	No	2:20 min
2	Tres y uno	Dos y uno	Si	-
3	Tres	Uno y uno	No	3 min
4	Cuatro y uno	Tres y cuatro	No	4:50 min
5	Uno, uno y uno	Dos, tres y dos	No	5:10 min

Tabla 5.1: Casos de pruebas de la implementación.

En todos los casos de prueba se han comprobado los archivos que se debían crear, modificar o borrar, además de comprobar el estado de la base de datos tras la ejecución.

En la segunda prueba hubo un problema, ya que se colapsó cuando llegaron los CMR al correo del ordenador. Esto no debería haber pasado si en los formularios se hubiera puesto otro correo a quien se debían de enviar el CMR y no al correo que tenía configurado en el dispositivo que se ejecuta el software.

Pero estas pruebas no son del todo objetivas, la mejor manera de comprobar el funcionamiento es que las lleven a cabo terceras personas, puesto que es más fácil ver los errores de otra persona o pensar de forma diferente al programador y considerar casos de uso que él ha obviado. Además, nos proporciona la experiencia de usuario, tratándose de un servicio para el cliente, esta característica es de vital importancia.

Se ha dejado ejecutando el software durante dos horas y se les ha pedido a dos personas con conocimientos básicos de informática que vayan rellenando formularios con un periodo variable. Una de esas personas lo ha realizado a través de un *smartphone* y la otra persona con un ordenador. La parte de los archivos de facturación han sido enviados por la persona desarrolladora del software, alternando archivos con formato correctos e incorrectos a una media de cuatro archivos cada cinco minutos.

Después del transcurso de las dos horas se recibieron un total de quince formularios y unos 90 archivos de facturación, tras mirar los correspondientes archivos y la base de datos se confirma que ha pasado la prueba de estas terceras personas.

Aun así, se necesitaría más afluencia de usuarios interactuando con el sistema, con el objeto de saber hasta qué nivel de concurrencia funcionaría el software desarrollado en la máquina en la que se ha desplegado (ordenador personal Intel i7-8550U).

## 6. Conclusiones y trabajos futuros

En este apartado se aborda las conclusiones y posibles trabajos futuros tras la realización de este proyecto.

### 6.1 Conclusiones

Este proyecto ha tenido por objeto el estudio de las tecnologías RPA, los *frameworks* existentes y su campo de aplicación con objeto de poder emplear estas tecnologías en el sector del transporte. En concreto se perseguía automatizar las tareas de recepción de pedidos, cumplimentación de cartas de porte y subida de archivos de facturación sin la necesidad de un operario.

Se ha de señalar que este TFG ha logrado satisfacer todos los requisitos especificados por el cliente, la empresa Fagor.

Dentro de este proyecto se diferencian dos apartados: primero, el trabajo de estudio e investigación y segundo, la implementación de dos casos de uso comunes en el sector del transporte. En la primera parte se ha abordado la búsqueda y el estudio de las herramientas RPA, en concreto se ha profundizado en el uso de TagUI y UiPath. La segunda parte, se trata de un desarrollo software en el que se han seguido todas las fases del ciclo de vida, construyendo un programa funcional y de fácil mantenimiento.

Este trabajo me ha servido para concienciarme de la gran utilidad de las herramientas RPA en tareas de oficina, permitiendo reducir los costes y el tiempo de ejecutar dichas tareas, a través de un robot software, evitando los errores que podría llegar a cometer un humano y mejorando la productividad de las empresas. Además, me ha permitido comprender que muchos puestos de trabajo de oficina están en peligro de extinción, ya que, estas herramientas siguen creciendo y mejorando para suplir a estos trabajadores, por lo que la formación continua persiguiendo la especialización debe inculcarse en el ámbito profesional. Las posibilidades de aplicación de las RPA no parecen tener límite.

### 6.2 Trabajos futuros

El desarrollo de este proyecto ha logrado cumplir con los requisitos iniciales planteados. Pero todo desarrollo software siempre tiene margen de extensión y mejora. Por ello como líneas de trabajo futuro se exponen las siguientes:

- Mejora del rendimiento: la continua adición de la macro a todos los archivos de Excel toma un tiempo notable en la ejecución, por lo que la solución a esto podría mejorar el número de ficheros procesados por día.
- Implementación del microservicio sobre Flotasnet que devuelva el conductor y la matrícula sin necesidad de interactuar con la base de datos.
- La utilización de UiPath Orchestrator para realizar un seguimiento de los procesos en ejecución, permitiendo estructurarlos y controlarlos desde el móvil o desde la web.
- Comprobación de archivos de facturación enviados en PDF, utilizando OCR.
- Por otro lado, sería interesante hacer un estudio más profundo sobre las herramientas Taskt y Automagica, estas dos herramientas son *open source* y fueron descubiertas tras empezar a implementar los casos de uso con UiPath. Automagica, realiza la implementación en la nube y Taskt tiene prestaciones similares a UiPath, una de las prestaciones importantes es una interfaz gráfica fácil de usar.

## Referencias

- 10XDS. (4 de 6 de 2019). *10XDS*. Recuperado el 28 de 11 de 2019, de <https://www.10xds.com/blog/choosing-the-right-rpa-tool/>
- AIMultiple. (14 de Octubre de 2019). *AIMultiple*. Obtenido de AIMultiple: <https://blog.aimultiple.com/rpa/>
- AIMultiple. (22 de 10 de 2019). *AIMultiple*. Obtenido de <https://blog.aimultiple.com/robotic-process-automation-rpa-vendors-comparison/#rpa-tools>
- AIMultiple. (1 de 1 de 2020). *AIMultiple*. Recuperado el 26 de 1 de 2020, de <https://blog.aimultiple.com/open-source-rpa/>
- DevTools. (s.f.). *DevTools*. Recuperado el 28 de 11 de 2019, de <http://www.devtools.in/html/rpa.html>
- EGHAM. (24 de 6 de 2019). *Gartner*. Obtenido de <https://www.gartner.com/en/newsroom/press-releases/2019-06-24-gartner-says-worldwide-robotic-process-automation-sof>
- Gartner. (2019). *UiPath*. Recuperado el 28 de 11 de 2019, de <https://www.uipath.com/es/company/rpa-analyst-reports/gartner-magic-quadrant-robotic-process-automation>
- Guru99. (15 de 1 de 2020). *Guru 99*. Obtenido de <https://www.guru99.com/robotics-process-automation-tools.html>
- HFS Research. (30 de 11 de 2018). *Horses for Sources*. Obtenido de [https://www.horsesforsources.com/RPA-forecast-2016-2022\\_120118](https://www.horsesforsources.com/RPA-forecast-2016-2022_120118)
- Kappagantula, S. (27 de 11 de 2019). *Edureka*. Recuperado el 28 de 11 de 2019, de <https://www.edureka.co/blog/rpa-tools-list-and-comparison/#Types%20of%20RPA%20Tools>
- Microsoft. (14 de 05 de 2019). *Microsoft | Documentacion de SQL*. Obtenido de <https://docs.microsoft.com/es-es/sql/relational-databases/json/json-data-sql-server?view=sql-server-ver15>
- RobotFramework. (10 de 1 de 2020). *Robot Framework*. Recuperado el 21 de 1 de 2020, de <https://robotframework.org/>
- Soh, K. (31 de 3 de 2018). *GitHub-TagUI*. Recuperado el 26 de 1 de 2020, de <https://github.com/kelaberetiv/TagUI/issues/112>
- TagUI. (16 de 8 de 2019). *GitHub/TagUI*. Obtenido de <https://github.com/kelaberetiv/TagUI#set-up>
- TIC, S. P. (7 de Noviembre de 2016). *Solo pienso en TIC*. Obtenido de <http://www.solopiensoentic.com/cuadrante-magico-de-gartner/>
- UiPath. (20 de 1 de 2020). *UiPath*. Obtenido de <https://www.uipath.com/>