

---

This is the **published version** of the bachelor thesis:

Torres Rodríguez, Aleix; Bolta Torrell, Helena, dir. Automatització del filtratge i distribució de taules de dades amb RPA. 2022. (958 Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/264112>

under the terms of the  license

# AUTOMATITZACIÓ DEL FILTRATGE I DISTRIBUCIÓ DE TAULES DE DADES AMB RPA

Aleix Torres Rodríguez

**Resum**— Aquest article recull el desenvolupament de l'automatització d'un procés de negoci real dins de l'empresa Boehringer Ingelheim. Aquesta automatització s'ha dut a terme amb RPA o Robotic Process Automation, una tecnologia que serveix per a automatitzar processos de negoci repetitius i avorrits. En concret, el projecte s'ha realitzat amb UiPath, l'eina que és capaç d'automatitzar qualsevol cosa que una persona fa amb un ordinador. El projecte consisteix en llegir dades d'un Excel i per cada fila, filtrar a partir d'aquestes dades, la informació que es mostra en una taula a una pàgina web. Aquesta taula s'exporta en format PDF i s'envia per correu electrònic a la persona indicada a l'Excel anteriorment comentat. Aquest procés de filtrar, exportar i enviar per correu es repeteix per cada fila de l'excel. El resultat obtingut és l'esperat, és a dir, un robot que automatitza al cent per cent tot aquest procés i allibera a un empleat de fer tot aquest treball de forma mensual, de forma que pugui dedicar el seu temps a altres tasques més importants i creatives.

**Paraules clau**—màxim 2 línies. RPA, robot, UiPath, Robotic Enterprise Framework, Orchestrator, Queue, Asset

**Abstract**— This article covers the development of the automation of a real business process within the Boehringer Ingelheim company. This automation has been done with RPA or Robotic Process Automation, a technology used to automate repetitive and tedious business processes. Specifically, the project has been done with UiPath, the tool that allows to automate anything a person does with a computer. The project consists of reading data from an Excel and for each row, filtering based on this data, the information displayed in a table on a web page. This table is exported in PDF format and e-mailed to the person listed in the Excel previously mentioned. This process of filtering, exporting, and e-mailing is repeated for each row in the excel. The result is the expected, that is, a robot that fully automates this whole process and relieves an employee of doing all this work on a monthly basis, so that he can dedicate his time to other creative or important tasks.

**Index Terms**—RPA, robot, UiPath, Robotic Enterprise Framework, Orchestrator, Queue, Asset

## 1 INTRODUCCIÓ

### 1.1 Context del treball

La proposta d'aquest treball sorgeix arran de la necessitat de varis usuaris que duen a terme una tasca molt repetitiva i consumidora de temps de forma mensual que es pot automatitzar. Aquest cas d'ús s'ha implementat dins l'equip RPA CoE de l'empresa Boehringer Ingelheim.

RPA és una tecnologia que va aparèixer fa més de 20 anys però no va ser fins el 2016 que realment es va donar a conèixer.

El projecte s'ha dut a terme amb UiPath, un llenguatge de programació de l'estil "Drag and drop", és a dir, arrastres d'activitats, per exemple, una activitat de clic o d'escriure i la poses en una part de codi. També utilitza .Net per a tot el tema de variables i funcions.

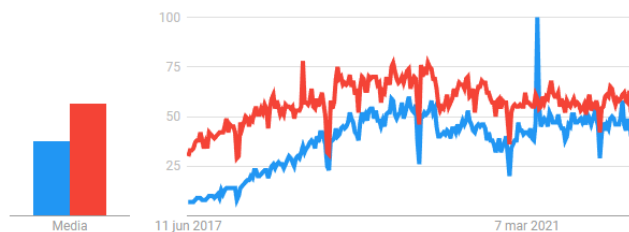


Fig. 1: Cerques a Google dels termes UiPath (Blau) i RPA (Vermell) [1]

Aquest procés de negoci a automatitzar, consisteix en llegir dades de tres excel·lents diferents. Cadascun d'aquests Excel correspon a un conjunt de persones amb càrrecs diferents dins de la empresa. Un cop llegides, es guardaran a una plataforma que ofereix el proveïdor del software UiPath anomenada Orchestrator. En aquesta plataforma es

- E-mail de contacte: 1493910@uab.cat
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Helana Boltà (Ciències de la computació)
- Curs 2021/22

guardaran només les dades necessàries de cada fila, per tant, prèviament es filtraran les dades rebudes com a entrada. Per cada fila de cada Excel s'ha de filtrar (a partir de les dades guardades al Orchestrator) la informació que es mostra en una taula. Aquesta taula es troba a una pàgina web de la empresa. Per accedir-hi, el robot haurà de obrir el navegador, navegar a la pàgina i iniciar sessió al sistema. Un cop dins, en funció de si la fila que s'està utilitzant per a filtrar la taula corresponia a un fitxer Excel o un altre, la forma de filtrar la taula canviarà.

Per a filtrar la taula, el robot escriurà a tres buscadors una informació i seleccionarà el filtre. Al finalitzar el filtre, la taula s'actualitza. Per a extreure la taula i poder-la enviar a la persona corresponent en funció de la fila de l'Excel, es farà clic dret a la taula i s'exportarà com a PDF. Per cada persona (per cada fila) a la que s'ha d'enviar un PDF diferent, el robot també crearà, dins d'un servidor, una carpeta per cada persona on hi guardarà el PDF corresponent.

Un cop guardat el PDF, s'enviarà un email utilitzant un template i adjuntant el PDF.

Cal recalcar que aquest robot imita el comportament humà, no s'automatitza mitjançant APIs, sinó amb un llenguatge específic (UiPath). Una de les dificultats més grans d'aquest llenguatge és marcar bé on ha de fer clic el robot, on ha d'escriure, etc. Això es fa analitzant el codi de la web que s'està automatitzant, ja que el robot busca selectors. Per tant, una tasca senzilla com fer un clic, es converteix en una de les més complicades al haver d'escollir els selectors més robustos i que no canviïn amb futures actualitzacions de la web mirant el codi HTML de la pàgina, entenent-lo i escollint els selectors adequats. [2]

## 1.2 Objectius

L'objectiu del projecte és la creació d'un robot que faci la tasca anteriorment descrita de forma funcional i autònoma. Per assolir-lo, s'han definit els següents objectius:

- Fer la presa de requeriments mitjançant entrevistes amb l'usuari.
- Dissenyar del codi a partir de la informació recopilada a les entrevistes.
- Documentar tot el projecte amb documents necessaris com el Process Definition Document, Robot Sheet, Monitoring Documentation i Robot Operational Manual.
- Desenvolupar el codi del robot que automatitza el procés de negoci.
- Test a l'entorn de desenvolupament i UAT o User Acceptance Testing, on l'usuari accepta el robot si compleix lo establert o no.
- Desplegament del robot a l'entorn de QA (Quality

Assuarnc) i Producció.

- Monitoratge de les primeres execucions.
- Anàlisi dels beneficis aportats a gràcies al robot.

## 2 PLANIFICACIÓ

El projecte consta de nou fases, cada una d'una durada aproximada d'entre una i dos setmanes, excepte la fase de desenvolupament, que dura un mes. Cada columna de la Figura 2 representa una setmana i cada fila un dels objectius del projecte.

Algunes tasques es poden fer en paral·lel. Per exemple el disseny de codi és una part de la documentació, per tant, quan es treballa en el disseny del codi, també es treballa en la documentació del projecte.

La fase de documentació comença abans del desenvolupament i es finalitza un cop acabat el robot, ja que hi ha petits canvis que poden ocórrer al llarg del desenvolupament i degut també a que hi ha certes parts de la documentació que no es pot saber sense haver fet el robot.

Les dues últimes setmanes de desenvolupament es solapen amb les dues setmanes de test ja que a mesura que es va desenvolupant els diferents mòduls del robot, es van testejant. Tot i que al final del codi es fan varis casos de prova.

El desplegament a l'entorn de Producció i el monitoratge de de les primeres execucions també es solapa perquè van de la mà. Un cop es fa el desplegament, ja comença la fase de Hypercare, on es controla d'aprop què fa el robot quan s'executa. Per detectar ràpidament possibles falles.

	Setmana 1	Setmana 2	Setmana 3	Setmana 4	Setmana 5	Setmana 6	Setmana 7	Setmana 8	Setmana 9
Presa de requeriments mitjançant entrevistes amb el client	■								
Disseny del codi a partir de la informació recopilada		■							
Documentació						■			
Desenvolupar codi			■	■	■	■			
Test a l'entorn de desenvolupament							■		
Fer UAT (User Acceptance Testing)								■	
Desplegament i test a l'entorn de QA (Quality Assurance)									■
Desplegament a l'entorn de producció									■
Monitoratge de les primeres execucions (fase de Hypercare) i llançament del projecte									■

Fig. 2: Diagrama de Gantt de la planificació del projecte.

## 3 METODOLOGIA

La metodologia utilitzada es la Cascada degut a que els desenvolupaments de les automatitzacions de processos solen durar poc temps. Entre tres i cinc setmanes, i com per definició, la metodologia en cascada és apta per a projectes petits amb entregables que són fàcils de definir des del començament [3], la metodologia utilitzada comunament a RPA i també en aquest treball, és la Cascada.

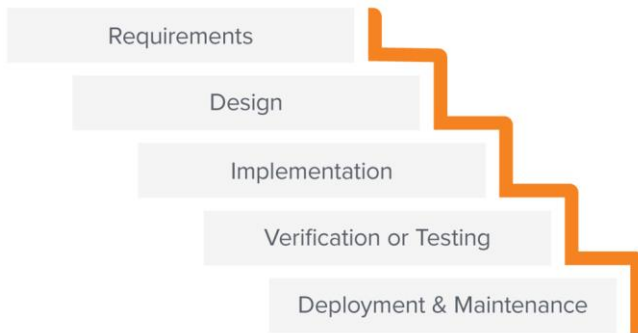


Fig. 3: Representació de la metodologia Cascada.

Les tasques que engloba la metodologia Cascada es poden trobar desengranades i amb el temps previst que requerirà cada tasca al diagrama de Gantt mostrat anteriorment (Fig. 2).

## 4 ESTAT DE L'ART

Pel disseny i implementació del codi s'han seguit una sèrie de bones pràctiques que té l'equip d'RPA de Boehringer Ingelheim.

Per al mòdul d'enviament de correus electrònics s'ha utilitzat una llibreria creada també per l'equip d'RPA de Boehringer Ingelheim.

No s'ha utilitzat cap altre projecte com a inspiració. Tot el disseny i desenvolupament surt del coneixement i experiència prèvia de portar més d'un any treballant en aquest sector.

## 5 DESENVOLUPAMENT

En aquest apartat s'explica tot el desenvolupament del projecte, que engloba desde la primera reunió amb l'usuari, fins la fase de monitoratge.

### 5.1 Presa de requeriments

La presa de requeriments va consistir en dos entrevistes amb el client. La primera sessió es va dur a terme a través de Microsoft Teams. El client, compartint la seva pantalla, va mostrar tot el procés que en el seu moment feia de forma manual, explicant tots els passos i resolent els dubtes que van sorgir. La presa de requeriment contempla des de la ubicació dels fitxers d'entrada, fins la URL de la pàgina on es troba la taula, passant per com es filtra la taula, com descarregar la taula en format PDF i l'enviament del document per correu.

Aquesta primera sessió es va gravar per a no oblidar cap pas, facilitar la creació de la documentació, evitar contactar al client i per a veure de forma fàcil i visual quins són els passos a seguir durant el desenvolupament del robot.

Posterior a aquesta primera entrevista, es va comprovar que el compte d'usuari utilitzat per a desenvolupar el

robot tenia els permisos necessaris. Entre altres, es va mirar que tingués accessos a les carpetes compartides on estan els fitxers d'entrada, es va comprovar que es tenia accés a la pàgina web, cosa que no era així, i es va haver de crear una conta a la pàgina web per un ús exclusiu del robot.

Després de fer aquestes comprovacions, vam tenir una altra reunió en la que vam tractar temes que no s'havien definit completament encara com la freqüència d'execució del robot i a quina hora i zona horària o quin contingut ha de tenir del correu que s'envia als destinataris. També es va comentar quins son els següents passos a seguir tant per la banda del client com per la del desenvolupador.

### 5.2 Disseny del codi

En aquest apartat es comenten els diagrames creats per al disseny del codi. Primer es mostra el diagrama d'alt nivell anomenats "As is" que fa referència a com és el procés actualment, és a dir, què fa el client actualment de forma manual i també veurem el diagrama "To be", és a dir, com serà el procés que farà el robot.

#### 5.2.1 As is

El diagrama d'alt nivell del procés consta de 5 passos. Anar al la taula, comprovar si les dades de la taula estan actualitzades i si ho estan filtrar la taula, descarregar-la en format PDF i enviar-la al destinatari. Aquests últims passos, corresponents als passos 3 i 4 es repeteixen fins que s'hagin utilitzat totes les dades dels fitxers d'entrada.

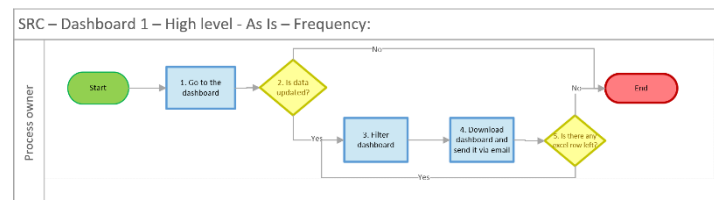


Fig. 4: Diagrama "As is". Representa què fa l'usuari manualment.

#### 5.2.2 To be

El diagrama mostra quins passos ha de fer el robot. La única diferència a alt nivell és que si les dades de la taula s'han actualitzat, aleshores el robot llegirà els fitxers d'entrada. Això es fa així per a ser més eficients, ja que en cas de que les dades no s'hagin actualitzat, no es llegiran els fitxers d'entrada, cosa que millora el temps d'execució.

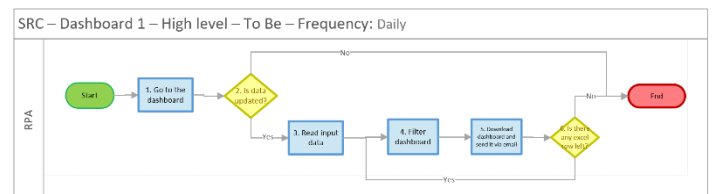


Fig. 5: Diagrama "To be". Representa què farà el robot.

### 5.3 Documentació

La documentació requerida per a poder començar el desenvolupament del codi és el Process Design Document (Document de disseny del procés) que conté entre d'altres:

- Contactes clau com el desenvolupador, l'analista i el propietari del procés.
- Requeriments mínims per a poder automatitzar el procés.
- Aplicacions utilitzades durant el procés.
- Diagrames "As is" i "To be" d'alt nivell i detallats.
- Passos del procés explicats amb un títol, una breu descripció, detalls del pas en concret i una captura de pantalla senyalant què ha de fer el robot. Per exemple, on ha de fer clic.
- Errors possibles i gestió d'excepcions. Aquest punt és molt important degut a que un error que no es té en compte pot fer que el robot es pari i finalitzi la execució sense haver acabat de fer el que havia de fer o sense avisar a ningú de què ha passat. Gestionant les excepcions, es controla que les parts crítiques no facin que el robot es pari.
- Informes generats. El robot genera un informe en format Excel amb totes les transaccions (files dels Excels) que el robot ha processat. En aquest informe, s'afegeix per cada transacció un camp que indica si s'ha processat correctament o no. I en cas de fallada, s'afegeix un missatge que conté una ruta on es guarda una captura de pantalla que fa el robot de l'error per a que es pugui analitzar de forma ràpida i visual què ha pogut passar.

La documentació restant, és a dir Solution Definition Document, Robot Sheet i Monitoring Documentation es finalitzen un cop s'acaba el desenvolupament.

A continuació s'explica què conté cada document:

- Solution Definition Document: Normalment el prepara el desenvolupador i ajuda els usuaris finals a entendre com implementar la solució automatitzada
- Robot Sheet: és un excel que conté tots els detalls tècnics del robot. Com el nom i valor del variables globals, el nom del procés, la versió del codi, contactes rellevants, etc. És a dir, conté tota la informació necessària per a poder fer el desplegament del robot a QA i Producció.
- Monitoring Documentation: Conté la informació necessària per a saber com monitoritzar el robot un cop està a Producció.

### 5.4 Desenvolupament del codi

En aquest apartat s'expliquen les eines utilitzades per al desenvolupament i el desenvolupament del robot en sí.

#### 5.4.1 Eines

El desenvolupament s'ha dut a terme en una màquina on l'entorn i les configuracions de les aplicacions és pràcticament idèntic a producció.

Per a l'automatització d'aquest procés s'ha utilitzat UiPath Studio, un software de RPA en el que s'arrosseguen i col·loquen activitats que fan accions. Per exemple, una activitat de "Clic" o de "Escriptura" o de "Obrir Excel".

Juntament amb UiPath Studio s'ha fet ús de la plataforma que també ofereix UiPath anomenada UiPath Orchestrator, que és un panell centralitzat d'administració de robots disponible a una pàgina web, on fàcilment es pot implementar, protegir i fer la gestió dels robots a escala.

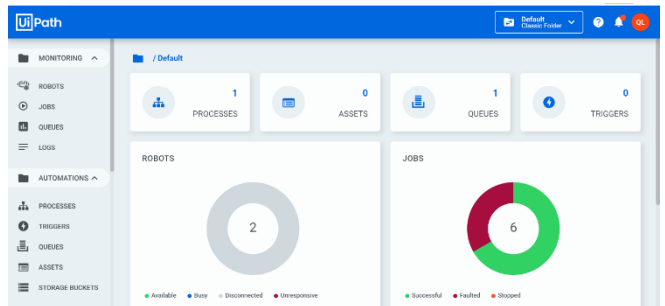


Fig. 6: Plataforma UiPath Orchestrator

#### 5.4.1 Codi

Per al desenvolupament del codi del robot, s'ha utilitzat el framework "Robotic Enterprise Framework" que ofereix

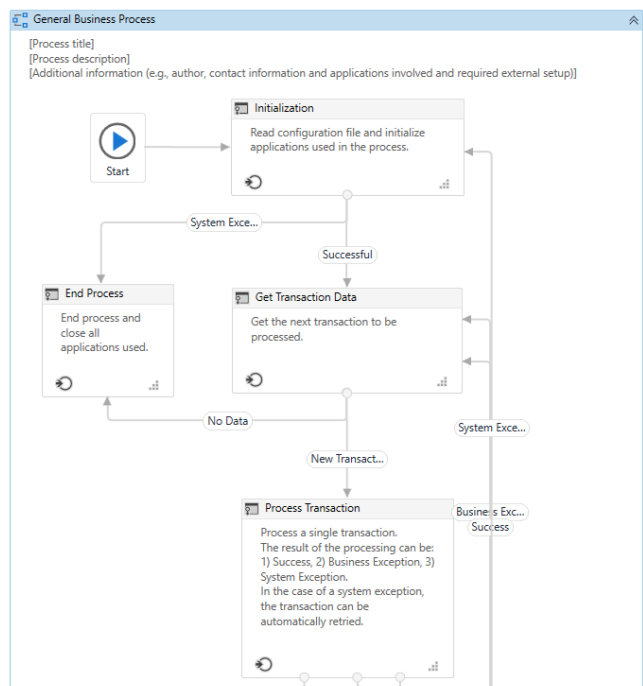


Fig. 7: Representació visual del RE Framework utilitzat.

UiPath i que proporciona l'esquelet principal que la majoria de robots comparteixen. Aquest esquelet consta de dels mòduls "Initialization", "Get Transaction Data", "Process Transaction" i "End Process". A continuació s'explica què inclou cada mòdul per a aquest procés en concret.

- **Initialization:** Bloc que llegeix el fitxer de configuració, inicialitza les aplicacions utilitzades durant el procés i emplena la "Queue".

Les "Queues" o cues, són llistes d'elements que els robots processaran. En aquest projecte, els elements de la cua són les files dels excels. Cada element de la cua conté el nom, correu i identificador de l'usuari que ha de rebre l'informe amb la taula, i les dades necessàries per filtrar la taula per a aquest usuari en concret.

El fitxer de configuració conté tots els detalls necessaris sobre com el Robot realitza el procés. Per a aquest projecte, el contingut és el nom de la cua, els noms dels fitxers d'entrada, la ubicació on es guarden captures de pantalla que fa el robot quan es troba amb un error i els noms dels "Assets" utilitzats.

Els "Assets" solen representar variables o credencials compartides que es poden utilitzar en diferents projectes (robots). Poden emmagatzemar informació específica per a que els robots hi puguin accedir fàcilment.

Aquests variables es troben a la plataforma UiPath Orchestrator, que s'accedeix a través del navegador web. Això permet poder canviar els valors dels Assets de forma ràpida i senzilla quan sigui necessari.

En aquest bloc, el primer que es fa és inicialitzar tots els paràmetres llegint el fitxer de configuració prèviament omplert. Tot seguit es guarden en una variable tots els Assets i es crida a una funció que s'encarrega de tancar totes les aplicacions que el robot fa servir. Això es fa per a assegurar que el robot sempre començarà a fer el procés des del mateix estat, és a dir, amb totes les aplicacions tancades.

A continuació s'ha programat la funció que inicialitza les aplicacions usades, és a dir, el navegador web "Microsoft Edge".

En aquesta funció, un cop obert el navegador, es navega a la URL de la pàgina web i es comprova si ja s'ha iniciat sessió de forma automàtica o no. Si no, procedeix a iniciar sessió entrant com a usuari i contrasenya les credencials guardades com a Assets al Orchestrator.

Després de fer varies proves, es va veure que en funció de quantes dades ha de carregar la pàgina, el robot arribava al temps màxim que espera entre acció i acció. Quan això passa, el robot falla, ja que

interpreta que el següent element que està buscant no apareix degut algun error. Aquest temps màxim és de 30 segons i s'ha augmentat a 1 minut. També es va implementar un sistema que reinventa una activitat fins que no apareix un element prèviament seleccionat [4]. Aquest element és un text que apareix després de la pàgina d'inici de sessió.

Si al cap de tres intents, encara no apareix aquest text, aleshores, s'assumeix que algun error per part de la pàgina ho està provocant i l'execució es para, després d'entrar al bloc de "End Process", que finalitza la execució tancant totes les aplicacions obertes

Un cop iniciada la sessió, es comprova que les dades de la taula han estat actualitzades. Per a fer això el robot compara el mes en el que s'han actualitzat les dades per últim cop amb el valor d'un Asset que conté quan va ser l'últim cop que es van actualitzar les dades. Si aquest valor és diferent, vol dir que s'han actualitzat i que per tant, com s'ha vist en el diagrama d'alt nivell, s'ha de continuar amb tot el procés.

La següent funció que s'ha desenvolupat és la que omple la cua amb les dades que es faran servir per filtrar la taula.

Per a fer això s'han llegit els tres fitxers d'entrada que són tres excels, on per cada fila, tenim un usuari i les respectives dades que s'utilitzaran per a filtrar la taula.

Un cop es tenen els fitxers llegits, s'eliminen les files duplicades per a evitar que els usuaris rebin el mateix correu varies vegades. No hauria d'haver files iguals, però s'ha implementat per a controlar el cas en que per un error humà, dues files són iguals.

Un cop filtrades les dades d'entrada, les utilitzem per a omplir la cua. Això implica que a la plataforma Orchestrator, podem veure tots els elements que hi ha a la cua, que correspon a quantes vegades es repetirà el procés de filtrar la taula.

- **Get Transaction Data:** És el bloc o mòdul de recuperació de dades, que s'utilitza per obtenir la transacció de la cua. Un cop el robot ha inicialitzat correctament totes les aplicacions que necessita per a un procés i ha omplert la cua, en aquest mòdul, es retorna un per un, cada element de la cua per a ser processat en el següent mòdul.

La funcionalitat preconstruïda del Framework te una opció que permet aturar, desde l'Orchestrator el procés encara que encara hi hagi elements per processar a la cua. Això esdevé útil en situacions en què s'observa que el robot no està fent el que s'espera que faci i es vol aturar el procés sense pararlo de cop en mig de la transacció que s'està

processant actualment. Es sol utilitzar quan el robot està a l'entorn de QA o Producció i es detecta un error.

- **Process Transaction:** Aquest bloc sempre s'executa tantes vegades com elements té la cua. I cada execució amb un element diferent fins que no queda cap element a la cua.

En funció de si l'element actual pertany a un fitxer d'entrada o un altre, la forma de filtrar la taula variarà una mica.

S'han programat aquestes tres variants de filtratge per separat:

- Cas 1: l'usuari a qui se li enviaren les dades és un cap.
- Cas 2: l'usuari a qui se li enviaren les dades és un gerent.
- Cas 3: l'usuari a qui se li enviaren les dades és un delegat.

En aquest mòdul s'utilitzen tres dades per a filtrar la taula. El filtre conté tres buscadors, on el robot, per cada buscador, comprova si el text que ja està introduït és el mateix que el es vol escriure. Si és diferent, introdueix el text i selecciona (fent clic) la opció que coincideix exactament amb el text que s'ha escrit.

Un cop aplicats els tres filtres, el robot identifica la taula i fa clic dret sobre ella. Això desplega un menú que ofereix la possibilitat de seleccionar "Export" i després "Visualization to PDF...". Al fer clic aquí, s'obra Pop Up en el que es seleccionen unes propietats concretes que fan que la taula es vegi ajustada amb les mesures correctes i finalment s'exporta la taula, és a dir, es descarrega com a PDF.

El robot està pendent a la descarrega [5]. Quan aquesta finalitza, mou el fitxer a una carpeta compartida [6] on es guarden tots els informes (o taules exportades com a PDF) que es van generant per a ser enviats més endavant.

Un cop el PDF està guardat a la carpeta corresponent, s'envia un correu a la persona adequada amb el PDF adjunt i un missatge personalitzat. Per a personalitzar el missatge i poder utilitzar variables per a canviar el nom del receptor del correu per exemple, s'ha transformat el missatge de tipus "String" a un codi HTML. L'activitat que envia correus es capaç de transformar les variables dins d'un codi HTML a String amb els valors de les variables, enlloc del nom en sí de la variable.

Com ja s'ha comentat, tot aquest procés de filtrar,

descarregar el PDF, guardar-lo i enviar-lo es fa per cada element de la cua, és a dir, cada fila dels Excel.

- **End Process:** Aquest bloc s'executa quan el robot falla al bloc "Initialization" o quan ja ha acabat tot el treball a fer, és a dir, quan ja no queden més items a la cua (queue) de l'Orchestrator.

Quan s'executa el bloc, el robot tanca sessió de la pàgina web i finalitza. Tot seguit tanca totes les aplicacions i en cas de trobar-se amb qualsevol error durant el tancament de les aplicacions obertes, llavors s'executa un funció que mata les aplicacions, com si finalitzessis el procés des de l'administrador de tasques de Windows.

La funció de tancar totes les aplicacions, primer tanca sessió en la pàgina web i tot seguit tanca l'aplicació del navegador web.

Com s'ha comentat, en cas de fallada, es crida la funció que mata les aplicacions. Bàsicament, mata el navegador web i finalment, es para l'execució.

## 5.5 Test i UAT (User Acceptance Testing)

Es va dissenyar un test plan que consta dels següents casos:

- El robot no troba els fitxers d'entrada.
- Els fitxers d'entrada estan buits.
- El navegador inicia sessió directament sense que el robot ho faci degut a que la sessió s'ha quedat oberta.
- El robot no troba algun element de la pàgina com els buscadors del filtre o certs elements als que el robot fa clic.
- El robot no troba el PDF descarregat.
- Cas en que el robot ha de processar 200 elements. Per comprovar que amb un gran nombre de elements funciona correctament.
- Enviament correcte de correus amb els PDFs pertinents.

Després d'aquests tests, la UAT va ser realitzada. La UAT és una reunió amb l'usuari per a que confirmi que els resultats obtinguts al executar el robot són els esperats. En aquesta sessió s'han fet els tests d'acceptació amb 2 files a cadascun dels fitxers d'entrada per a que l'usuari vegi que els 3 filtres funcionen.

L'usuari va donar el vistiplau per a continuar amb el desplegament a l'entorn de qualitat (QA).

## 5.6 Desplegament a QA i Producció

A l'entorn de QA (Quality Assurance) s'han reproduït els mateixos casos de prova per a comprovar que el robot actua correctament a aquest entorn. Com l'entorn de QA és igual que el de Producció, un cop els casos de prova donen resultats correctes, es pot desplegar el robot a producció. És important tornar a testejar en aquest entorn, ja que sol haver diferències entre l'entorn de desenvolupament i el de QA, i tot i que el robot funcioni a l'entorn de desenvolupament, pot haver un clic, per exemple, que no funcioni a QA.

El que s'ha intentat és que s'executi almenys un 80% del codi. És possible que al canviar de màquines on el robot s'executa, algun element que el robot busca canviï, per tant, s'ha d'intentar que el màxim de codi s'executi per a comprovar que efectivament el robot troba tots els elements que busca.

## 5.7 Monitoratge (Hypercare)

Aquesta fase consisteix en controlar que el robot s'executa bé a producció les primeres vegades amb dades reals.

El robot s'executa una vegada cada mes. Per tant, el dia d'execució (que varia en funció de si les dades de la taula estan actualitzades) es va monitoritzar el robot cada 30 minuts. El robot es va executar correctament però es va veure que es podia optimitzar la part d'inici de sessió [7], de tal forma que es va tornar a l'entorn de desenvolupament, es va crear un inici de sessió formal per a fer canvis als robots anomenat "Change Request" i es va millorar la funcionalitat fent els selectors més robustos i unes comprovacions prèvies que abans no existien. Un cop millorada la funcionalitat, es va desplegar a QA, es va testejar la nova funcionalitat a QA i tot seguit es va tornar a desplegar a Producció un altre cop amb la nova versió del codi.

## 6 RESULTATS

Aquest robot realment s'executa per dos taules diferents. El procés es repeteix exactament igual per les dos taules. Aquestes taules estan en URLs diferents. Per tant, el robot primer fa totes les transaccions a una taula, i després a l'altra.

Tenint en compte això i gràcies a les estimacions proporcionades per l'usuari, obtenim les següents dades:

- Mitja del nombre de persones que fan el procés per obtenir la seva taula: 280.
- Mitjana de consultes mensuals de cada usuari per comprovar si s'han actualitzat les dades de la taula: 1.8 consultes/mes
- Mitja consultes per treure informació: 1.2 consultes
- Temps per iniciar sessió a la pàgina web: 2 minuts

- Esperar que la pàgina carregui totes les dades i comprovar que les dades s'han actualitzat: 4 minuts
- Seleccionar la primera taula, esperar càrrega, filtratge i descàrrega de dades: 7 minuts
- Seleccionar la segona taula, esperar càrrega, filtratge i descàrrega de dades: 5 minuts
- Mitjana mensual de temps de resolució de dubtes respecte a càrrega de les taules per part de l'equip encarregat: 72 minuts = 1.2 h/mes

A partir d'aquestes dades obtenim:

Hores estalviades anuals = 280 persones \* (1.8 consultes comprovació dades actualitzades \* 6 minuts per comprovar-ho + 1.2 consultes per treure informació de la taula \* 14 minuts d'inici sessió, taula 1 i taula 2) + 72 mins resolució dubtes = 7800 mins = 130h \* 12 mesos = 1560h

Les hores anuals estalviades per aquest robot són 1560.

Si el procés d'entrar a la pàgina web, iniciar sessió, esperar que carregui, filtrar la taula i descarregar el PDF el fa el robot obtenim els següents resultats:

El robot triga 27,5 segons en fer cada transacció, és a dir, iniciar sessió, filtratge, descàrrega i enviament del PDF.

Si de mitja hi ha 280 transaccions, el robot triga  $280 * 27,5 / 3600 = 2,13$  hores cada execució. Tenint en compte que el robot s'executa dues vegades al mes (una per cada taula), obtenim que el robot triga  $2,13h * 2 = 4,26$  hores al mes, que són 51,12 hores anuals.

## 7 CONCLUSIÓ

Com podem veure a l'apartat dels resultats, la automatització d'aquest procés ha resultat un èxit ja que anualment estalvia 1560 hores que ara poden ser empleades per altres tasques més importants. Per a donar una referència sobre si són moltes o poques hores, només s'automatitzen processos que estalviïn un mínim de 300 hores. És a dir, que supera per 5 vegades el mínim.

A més, s'ha de tenir en compte que al delegar aquesta tasca a un robot, evitem errors que poden provenir de l'equivocació humana a l'hora de filtrar la taula. Així que no només s'han estalviat hores, sinó que també s'eviten errors provocats pels humans.

L'objectiu de la tecnologia Robotic Process Automation és automatitzar tasques repetitives i avorrides per a que les persones puguin dedicar el seu temps a tasques més importants i creatives. Així que podem dir, que aquest projecte s'ha completat exitosament ja que s'ha aconseguit l'objectiu principal d'RPA.



## AGRAÏMENTS

A la tutora d'aquest Treball de Final de Grau per part de la UAB, Helena Boltà, per entendre l'objectiu que tenia a l'hora de fer aquest projecte i la llibertat i confiança donada.

Al tutor per part de Boehringer Ingelheim, Pedro Rodríguez, per les facilitats donades per a tirar endavant el projecte i ajudar quan ha estat necessari.

## BIBLIOGRAFIA

[1] Creixement dels termes de cerca de "UiPath" i "RPA" proporcionat per Google Trends [en línia]. Recuperat de [https://trends.google.es/trends/explore?date=today%205-y&q=%2Fg%2F1f50xjkr9,%2Fg%2F1c3p\\_5fs0](https://trends.google.es/trends/explore?date=today%205-y&q=%2Fg%2F1f50xjkr9,%2Fg%2F1c3p_5fs0)

[2] Fòrum oficial de UiPath [en línia] Recuperat de <https://forum.uipath.com/t/is-there-a-best-practice-for-creating-selectors-so-they-continue-to-work-when-a-website-is-updated/246707>

[3] Definició de la metodologia [en línia]. Recuperat de <https://www.lucidchart.com/blog/es/pros-y-contras-de-la-metodologia-de-cascada#:~:text=La%20cascada%20se%20basa%20en,de%20definir%20desde%20el%20inicio>

[4] Documentació oficial de UiPath. Com funciona la activitat "Retry Scope" [en línia]. Recuperat de <https://docs.uipath.com/activities/docs/retry-scope>

[5] Fòrum oficial de UiPath. Com controlar una descarrega de fitxer [en línia]. Recuperat de <https://forum.uipath.com/t/wait-for-download-does-not-execute-activity-to-initiate-download-block/287963>

[6] Fòrum oficial de UiPath. Moure un fitxer a una carpeta compartida [en línia]. Recuperat de <https://forum.uipath.com/t/how-to-move-files-within-a-directory-to-a-directory-in-a-shared-cloud/54128>

[7] Fòrum oficial de UiPath. El robot no troba el selector en un clic [en línia]. Recuperat de <https://forum.uipath.com/t/getting-ui-element-or-selector-not-found-exception/194885>