



**UNIVERSIDAD
DE ANTIOQUIA**

**Automatización de la descarga de información
transaccional en la Mesa de Dinero del Grupo
Bancolombia.**

Autor

Frank Kevin Castro Cangrejo

Universidad de Antioquia

Facultad de Ingeniería

Departamento de Ingeniería Electrónica y de
Telecomunicaciones

Medellín, Colombia

2020



Automatización de la descarga de información transaccional en la Mesa de Dinero del
Grupo Bancolombia.

Frank Kevin Castro Cangrejo

Informe de práctica como requisito para optar al título de:
Ingeniero electrónico

Asesores:

Sebastián Isaza Ramírez
Asesor Universidad de Antioquia
Laura María Barrientos Tamayo
Asesora Bancolombia

Universidad de Antioquia
Facultad de Ingeniería
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Medellín, Colombia
2020

Tabla de contenido

Resumen	3
Introducción	4
Objetivos	5
Marco teórico.....	6
Metodología.....	11
Resultados y análisis	17
Conclusiones	24
Referencias bibliográficas	25

Resumen

En la Mesa de Dinero del Grupo Bancolombia se realizaba una descarga de información desde unos aplicativos, la cual se preparaba luego para ser usada en procesos de vigilancia. Este trabajo era repetitivo y requería que varias personas destinaran parte de su tiempo de trabajo para realizarlo.

Con este proyecto se logró automatizar tanto el proceso de descarga, como el de preparación de la información por medio de un asistente robótico, dicho asistente usa la tecnología RPA. En el proceso de desarrollo se presentaron varios retos que fueron solucionados en el transcurso del tiempo y se usaron varios lenguajes de programación para construir el asistente. Una vez terminado el proyecto, se entregaron los ejecutables del software desarrollado al área encargada de descargar la información.

Introducción

Dentro del grupo Bancolombia existe una gerencia que se encarga de realizar procesos analíticos de vigilancia y monitoreo a las transacciones comerciales realizadas en la Mesa de Dinero para verificar el debido cumplimiento de las normas éticas y legales de la organización. La información de esas transacciones queda registrada en aplicativos, estos pueden ser propios del banco, como también existen aplicativos externos al banco, es decir manejado por terceros.

Diferentes personas estaban encargadas de desplazarse hasta las oficinas donde se encuentra la Mesa de Dinero para descargar la información desde los diferentes aplicativos donde quedaba almacenada. Esta información se compone de tablas que se descargan con diferentes formatos de archivos de Excel. Luego de descargada, se le hacían ciertas adecuaciones de formato y por último la guardaban en una carpeta específica para cada archivo.

El trabajo de descarga era igual todos los días y requería que varias personas sacaran espacio de su tiempo de trabajo para realizarlo. Las adecuaciones de la información siempre se realizaban en los mismos campos de las tablas, por tanto, tenía las características requeridas para realizarse automáticamente. Con base en estas características del proceso, surgió la idea de automatizarlo para ahorrarle tiempo al equipo de la gerencia que estaba encargado de obtener la información.

Se usó el framework de Automation Anywhere para desarrollar un asistente automático haciendo uso de la tecnología RPA (Robot Process Automation). Al momento de iniciar el proyecto ya se contaba con la licencia de Automation Anywhere y los permisos necesarios por parte del banco para configurar el asistente robótico. El asistente se programó para ejecutar varias tareas en dos computadores, cada tarea es realizada por un bot y se encarga de descargar de uno a varios archivos. De un aplicativo de la Mesa de Dinero que es fuente de información se descargan dos o más documentos de Excel. Se requirió crear un archivo de Excel por cada computador para almacenar usuarios y contraseñas de cada aplicativo, como también variables útiles tales como fechas, nombre de la carpeta de destino del archivo de Excel, entre otros datos.

En la automatización también se usó programación en Visual Basic para apoyar al asistente robótico cuando tiene que realizar adecuaciones a una tabla de Excel. Se configuró un registro para que almacene los fallos presentados en la ejecución de cada bot, con la recolección de estos datos se obtuvo información que se

empleó para mejorar el desempeño de los bots. Los archivos de registro de fallos también serán útiles en futuros mantenimientos del software RPA.

Se redactó una documentación con información para describir la manera en que opera cada bot, además de los archivos necesarios para que funcione correctamente e instrucciones para su mantenimiento. En la documentación también se incluyó la hora en la que cada bot debe ejecutarse, además se mencionó la ubicación del código fuente y demás requisitos para el correcto funcionamiento de todo el asistente. Para asegurar una correcta comprensión del código, este se dejó debidamente comentado.

Luego se entregó el código fuente y la documentación del asistente robótico al Centro de Excelencia Operativa, que es área del banco encargada de ejecutar y monitorear las automatizaciones realizadas en el framework de Automation Anywhere.

En este informe se presenta una descripción del proceso realizado para implementar el asistente robótico encargado de la descarga de información de las mesas de dinero, los resultados del software desarrollado y se analiza su desempeño durante la etapa de seguimiento y evaluación, mencionando tanto sus limitaciones como también sus ventajas.

Objetivos

Objetivo general: Desarrollar una aplicación de automatización del proceso de descarga diaria de información, por medio de la implementación de RPA para optimizar el tiempo de la Vicepresidencia de Cumplimiento en la ejecución de sus procesos analíticos de vigilancia y monitoreo dentro de las mesas de dinero.

Objetivos específicos:

- Analizar el proceso que se va a automatizar haciendo uso de la información suministrada por la persona que estaba encargada, para conocer los requisitos funcionales que debe tener la aplicación.
- Desarrollar la aplicación de RPA usando el framework de Automation Anywhere para descargar la información de la Mesa de Dinero.
- Almacenar los fallos presentados en ejecución mediante un archivo de texto para facilitar las labores de mantenimiento del software.

- Construir una automatización robusta, basada en la información obtenida por la experiencia de los analistas en el manejo de la plataforma, para que el software reconozca la mayor cantidad de escenarios posibles presentados por las aplicaciones que opera.

Marco Teórico

Dentro del Grupo Bancolombia existe un espacio físico llamado Mesa de Dinero, dicho lugar es un centro de contacto para el cierre de transacciones comerciales. Su propósito principal es llegar a la solución de los problemas temporales de empresas del sector financiero, donde ciertas personas llamadas “Traders” actúan como un puente para la negociación entre oferentes y demandantes de dinero. La Mesa de Dinero está compuesta por varias mesas más pequeñas que manejan diferentes mercados financieros, tales como: divisas, capitales, deuda pública, etc. [1]

Todas las transacciones realizadas en la Mesa de Dinero quedan registradas en tablas y se almacenan en diferentes aplicaciones. Algunas aplicaciones son propias del banco y almacenan las operaciones que el banco realizó en cualquier mercado financiero a nombre propio y por parte de sus clientes. Otras aplicaciones no pertenecen al banco, entre sus propietarios están la Bolsa de Valores de Colombia y el Banco de la República. En estos aplicativos externos se almacenan todas las operaciones realizadas en el mercado, es decir están incluidas las operaciones propias de Bancolombia, las operaciones realizadas por Bancolombia a nombre de terceros y las transacciones realizadas en las demás entidades financieras.

En el Grupo Bancolombia se encuentra una vicepresidencia encargada de vigilar el debido cumplimiento de las normas jurídicas y de ética propias de la empresa. En el desempeño de su misión, la Vicepresidencia realiza seguimiento a las operaciones realizadas por los Traders, las cuales se concretan en las mesas de dinero. Para llevar a cabo el monitoreo de estas transacciones se descarga la información desde los aplicativos de la Mesa de Dinero.

La información descargada se almacena en un servidor que se accede por medio de una ubicación de red, que es compartida por las personas que se encargan de hacer los monitoreos. Este servidor está compuesto por diferentes carpetas y cada archivo que se descarga tiene una carpeta específica asignada. Toda la

información que se descarga está compuesta por archivos de Excel, que manejan los siguientes formatos: csv, xls, xlsb yxlsx. Csv es el formato de archivos de texto que se usa para almacenar tablas, donde los campos de la tabla están separados por comas y los registros se diferencian con un salto de línea. Xls es un formato que contiene libros de Excel 97 a Excel 2003. Xlsb es un formato que contiene libros de Excel 2010 y Excel 2007. Y Xlsx es un formato de archivo basado en XML predeterminado de Excel 2010 y Excel 2007. [2]

- **Herramientas usadas en el proyecto:**

Dentro de los archivos descargados hay datos de tipo fecha, texto y número. La información descargada necesita adaptarse a los requisitos técnicos que exige el software empleado para los monitoreos. En algunos casos se almacena el archivo de Excel tal y como se descargó, no requiere ningún ajuste, en otros casos es necesario cambiar el formato de xls a xlsx. También, a veces es necesario cambiar una variable de tipo numérica a texto y viceversa, o eliminar alguna columna. Una vez aplicado los respectivos ajustes a la información, se convierte en insumo para las pruebas de monitoreo. Para lograr adaptar los archivos descargados al formato correcto y aplicar los ajustes antes mencionados, se usa una herramienta que Excel provee para realizar automatizaciones, esta herramienta recibe el nombre de macro. Una macro es un conjunto de comandos que se almacena en un lugar especial de Excel de manera que están siempre disponibles cuando se necesiten ejecutar. Estos comandos se escriben en el lenguaje de programación llamado VBA, que en inglés significa Visual Basic for Applications. Excel posee un editor de VBA con herramientas de depuración, con la cual también se pueden construir interfaces gráficas [3]. Xlsm es el formato de archivo basado en XML y habilitado para macros para Excel 2016, Excel 2013, Excel 2010 y Excel 2007. Almacena código de macros de VBA u hojas de macros de Excel [2].

Otra herramienta útil para el proyecto es VBS. Este es un lenguaje de programación de scripts basado en Visual Basic y sirve para interactuar con el sistema operativo de Windows.

El proyecto de automatización requiere de un asistente robótico, el asistente está compuesto a su vez por dos instancias que reciben el nombre de "Runners". Cada Runner está instalado en un computador diferente, con su respectivo usuario de Windows. En cada instancia, el asistente contiene tareas más pequeñas, que

reciben el nombre de bots. Al espacio de tiempo tomado para la programación y evaluación se le conoce como etapa de configuración del asistente robótico.

- **Tecnología empleada para la implementación del asistente robótico:**

El asistente robótico usa la tecnología RPA. La automatización robótica de procesos, RPA por sus siglas en inglés, permite configurar un software informático que hace posible que un bot emule e integre las acciones de una interacción humana en sistemas digitales para ejecutar un proceso, en nuestro caso, el proceso es el conjunto de pasos necesarios para que la información esté lista para sus usuarios. Los bots emplean la interfaz de usuario para capturar datos y manipular las aplicaciones del mismo modo que los humanos. Estos bots realizan interpretaciones, activan respuestas y se comunican con otros sistemas para operar en una amplia gama de tareas repetitivas. Y tienen las ventajas de ser menos costosos que un empleado y están disponibles todo el tiempo [4].

- **Arquitectura de la aplicación de desarrollo de RPA:**

Automation Anywhere es un framework de desarrollo de software RPA, posee una interfaz de desarrollo gráfica para la implementación de bots. Dentro del entorno de desarrollo existe un panel de comandos, un administrador de variables y el editor del código fuente. El desarrollo de un bot tiene dos etapas, la etapa de configuración y la etapa de producción. En ambas fases, los bots son administrados desde el Control Room que es una interfaz basada en un servidor de Windows, para acceder a él se requiere conexión a la intranet del banco. El Control Room proporciona una vista unificada de todo el entorno de automatización y una plataforma para la administración centralizada. Además de implementar y ejecutar, también se encarga de monitorear el funcionamiento de los bots. Al ser un administrador, puede definir roles personalizados, además de establecer privilegios y permisos para el conjunto completo de objetos y funciones de Control Room, que incluye la administración de usuarios, licencias, programación de bots, paneles y registros de auditoría [5].

En Automation Anywhere los bots reciben el nombre de Taskbot. También, se pueden usar otro tipo de módulos de programación conocidos como metabots, que tienen características similares a una función en el contexto de otros lenguajes de programación. Los metabots tiene la ventaja de agregar modularidad en el proceso de creación de un taskbot y además son reutilizables, lo que simplifica el código fuente [6].

En la figura 1 se observa la ventana principal de Automation Anywhere. En la parte 1 se muestran dos pestañas, una de automatización y otra de administración. En la pestaña de automatización se puede observar los diferentes tipos de softwares que se pueden desarrollar, en nuestro caso, se hizo uso de los metabots y taskbots. La pestaña de administración permite el acceso al Control Room, a un repositorio con los bots y metabots desarrollados en la empresa por otras áreas. En la parte 2 de la figura se observan los bots desarrollados con un resumen de algunas propiedades y el resultado de la última ejecución. Y en la parte 3 se observa las configuraciones que se pueden hacer sobre los bots. En el proyecto se dejó la configuración de los bots que viene por defecto en el framework.

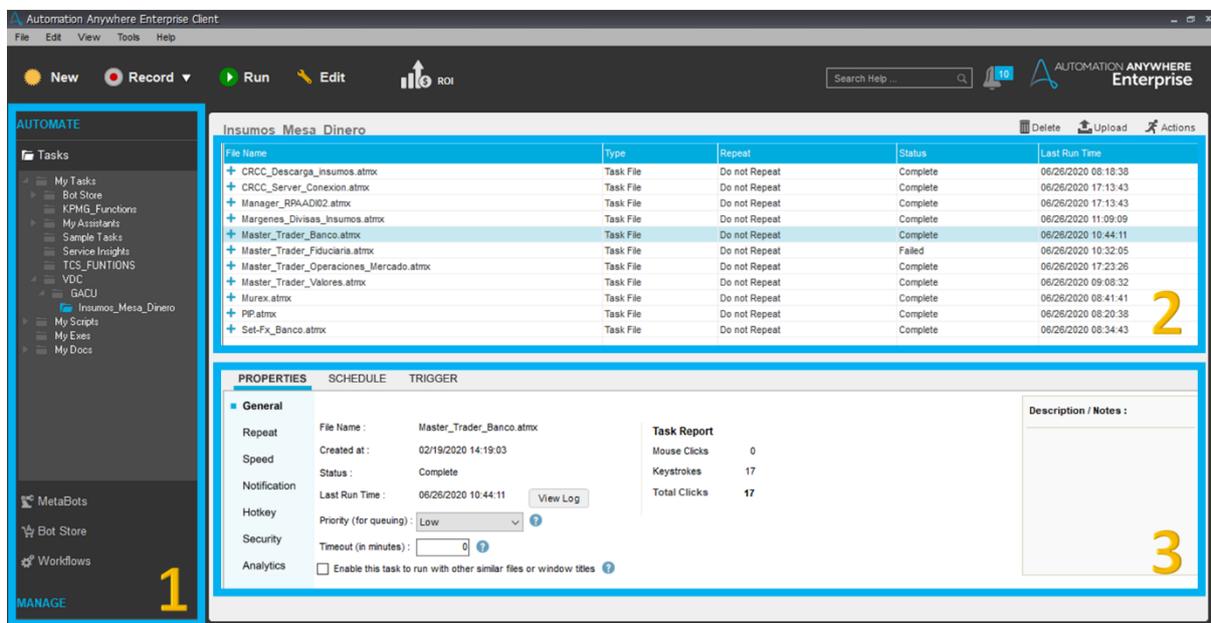


Figura 1. Ventana principal de Automation Anywhere.

El lenguaje de programación de Automation Anywhere está compuesto por diversos comandos, en la figura 2 se puede observar la lista de todos ellos. Y en la tabla 1 se realiza una descripción breve los principales comandos.

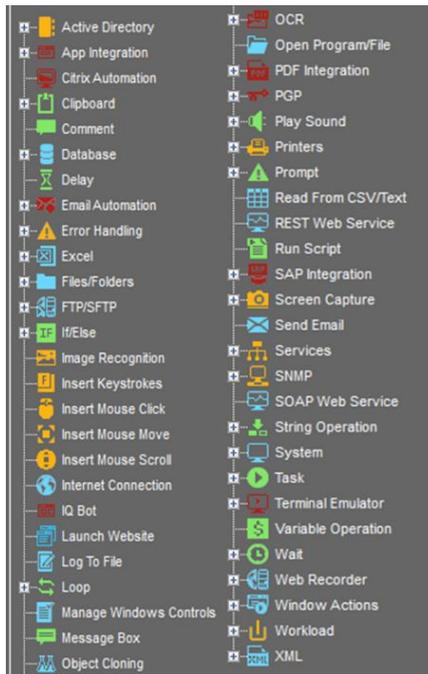


Figura 2. Comandos para desarrollar el software RPA en Automation Anywhere.

Tabla 1. Descripción de los principales comandos de Automation Anywhere.

Comando	Función
Delay	Detiene la ejecución del bot durante un periodo de tiempo estipulado por el programador.
Error handling	Son dos instrucciones que delimitan un bloque de código, en caso de presentarse un error dentro del bloque se elige dentro de una lista de opciones la acción a realizar.
Excel	Es una serie de comandos: abre y cierra archivos de Excel, ejecuta macros y realiza modificaciones a los datos del archivo.
Files/Folders	Realiza acciones en carpetas y archivos, tales como: Crear, copiar, mover, eliminar, etc.
If/Else	Condicionales para la toma de decisiones.
Insert Keystroke	Hace uso del teclado.
Insert Mouse Click	Hace uso del mouse.
Loop	Para hacer iteraciones sobre un bloque de código.
Manage Windows controls	Realiza una acción determinada sobre algún elemento de una ventana.
Object Cloning	Este es uno de los comandos más usados, extrae las propiedades de un objeto que esté en la interfaz gráfica y las almacena. En tiempo de ejecución del bot, esta instrucción busca en la ventana de windows el elemento guardado durante la configuración, eligiendo al que mejor coincida con las propiedades guardadas y realiza la acción que fue determinada realizarse sobre él.
Open Program/File	Abre un programa o un archivo.
Run Script	Ejecuta scripts de Java o Visual Basic.
Task	Ejecuta un bot desde el bot que está en ejecución, también, detiene o pausa el bot actual.
Variable Operation	Realiza asignaciones o modificaciones a las variables.
Wait	Espera hasta que determinado evento ocurra.
Windows Actions	Maximiza, minimiza, cierra y reescala las dimensiones de una ventana.

El Cedex o Centro de Excelencia Operativa es el área del banco que administra las herramientas del Control Room, a esa sección se le entrega los ejecutables y la documentación del asistente, lo cual sucede en la última etapa del proyecto que es el paso a producción.

Metodología

El proyecto tuvo 3 etapas para el desarrollo completo del asistente robótico:

- **Primera etapa:**

La primera etapa del proyecto consistió en conocer el framework de desarrollo, entender la necesidad que se quería cubrir con el desarrollo del asistente y establecer cuáles eran los mejores parámetros de diseño y estructura del código fuente.

En los primeros días estudié un documento donde se resumía las principales herramientas del framework de Automation. Después dediqué una semana a familiarizarme con el entorno de desarrollo, para ello desarrollé algunos bots usando unas cuantas instrucciones con el propósito de conocer los alcances y limitaciones de cada comando, e hice unas pruebas de ejecución. También me apoyé en la documentación de referencia que está publicada en la página oficial de Automation Anywhere para entender la funcionalidad de cada comando, e investigaba en foros de programación cuando se me presentaba algún error de ejecución. Con esto quedaron establecidas las bases conceptuales y prácticas para un primer desarrollo. El conocimiento de las herramientas y las habilidades de programación de RPAs fue mejorando con el tiempo.

Después la asesora externa, que estuvo acompañándome durante el periodo de práctica, me contó acerca del proyecto de automatización que querían desarrollar. Me entregó una descripción de la descarga de algunos archivos, también me mencionó los aplicativos de mesas de dinero que se querían impactar con el asistente robótico y la información que necesitaban descargar. Después me presentó en diferentes encuentros el proceso de descarga manual para algunas aplicaciones, también mencionó algunas recomendaciones para que las tuviera en cuenta al momento de hacer la automatización, en la mayoría de tareas se siguen los siguiente pasos: Abrir la aplicación, ingresar credenciales de acceso (algunas aplicaciones no necesitan credenciales de acceso), navegación en los

menús de la aplicación, ingresar la fecha del día del cual se va a descargar de la información (en algunos casos solo existen los datos del mismo día, de lo contrario la fecha que se ingresa es la del último día hábil), descargar el archivo de Excel, hacerle los respectivos ajustes (algunos solo se les coloca el nombre y se dejan tal y como están) y almacenarlo en la carpeta especificada. A parte de mi asesora, otros empleados del banco me dieron a conocer el proceso de descarga que realizaban manualmente, junto con algunas recomendaciones acerca de eventos que ocasionalmente ocurrían en las aplicaciones de donde se extrae la información. Conociendo el proceso empecé a investigar la mejor manera de automatizarlo, además, junto con mi asesora externa establecimos un cronograma de actividades que incluía objetivos por cumplir y reuniones de seguimiento semanales.

Al principio se tenían muy pocos patrones establecidos para la estructura de la programación del asistente, por lo que después de hacer varias pruebas se fueron consolidado los patrones de diseño. En el diseño se tuvo en cuenta los siguientes aspectos: el lugar del almacenamiento de variables, el manejo de errores y la determinación de qué parte de la automatización se haría en Excel.

- **Segunda etapa:**

En la segunda etapa se desarrolló el código fuente de los bots y se construyeron los archivos y demás herramientas externas usados por el asistente robótico. Estos elementos externos ayudan a hacer estable la programación en Automation Anywhere, disminuyen la longitud de los bots en términos de líneas de código y aportan modularidad y escalabilidad al proyecto.

El proceso de programación se hacía en varios pasos. El primero de estos era crear todas las variables que se iban a usar. En el paso siguiente se cargaban desde un archivo externo los valores que almacenarían algunas variables y también se inicializaban el resto de las variables que lo requerían. Luego se empezaba a colocar unas pocas líneas de código y se ejecutaba el bot para verificar que funcionara correctamente, en caso de que se presentara un error se corregía el comando que había fallado y nuevamente se ejecutaba. Se repetía una y otra vez el proceso de ensayo y error hasta que terminara satisfactoriamente la ejecución de lo que se había programado, después, se adicionaban más comandos y nuevamente se ejecutaba el bot para evaluar su funcionamiento. Así sucesivamente se iba construyendo el software hasta completarlo. En el proceso se

agregaban comentarios y se escribía en la documentación algún aspecto que fuera necesario para la comprensión del código en el futuro.

La programación en Automation es muy intuitiva, porque los bloques de construcción del código fuente son gráficos y corresponden a acciones que son realizadas por una persona. En la figura 3 se presenta la ventana de configuración de bots, la cual se encuentra dividida en 3 secciones. En la primera sección están los componentes básicos del programa que son los comandos y los metabots. En la parte 2 de la figura número 3 se encuentra el editor de instrucciones del bot en desarrollo, en esta parte se escribe el código fuente del software RPA. En la parte 3 de la misma figura están todas las variables usadas por el bot, por lo general se emplean para almacenar cadenas de texto; también se puede acceder a una lista de variables propias de Automation Anywhere, que almacenan entre otros los siguientes datos: fecha y hora del sistema, texto del portapapeles e información del sistema.

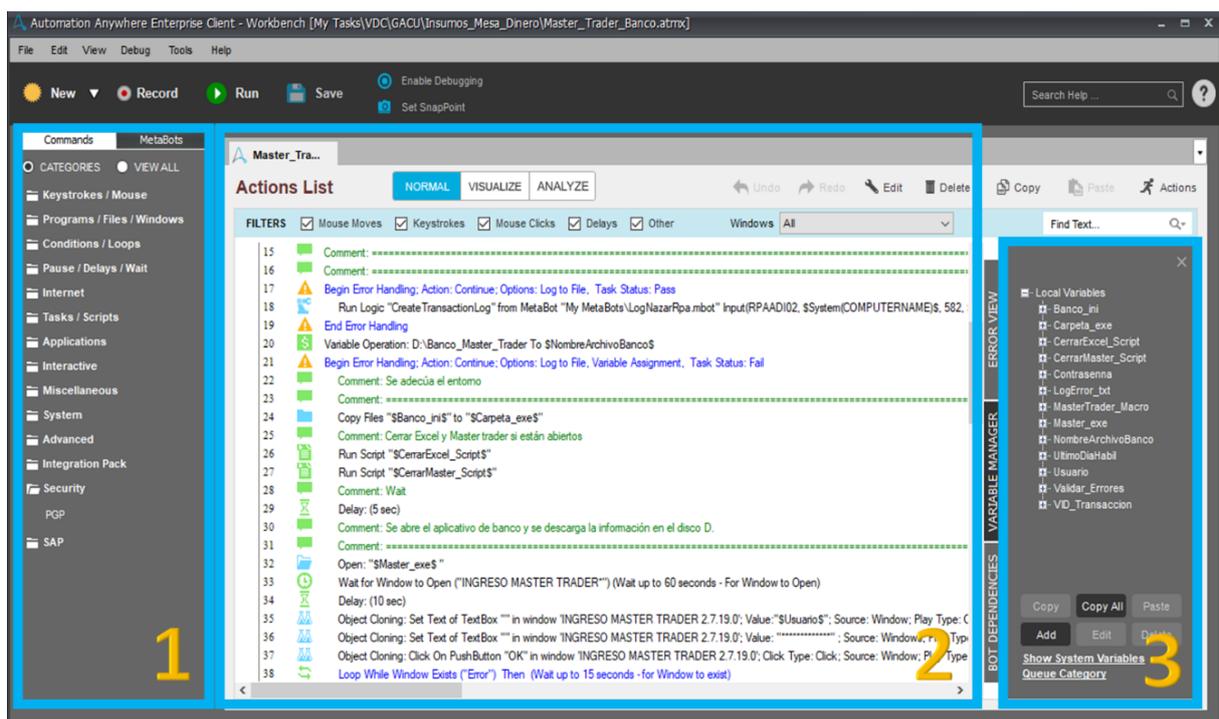


Figura 3. Ventana de configuración de bots de Automation Anywhere.

Los primeros bots tuvieron varias actualizaciones para enfrentar diferentes retos que se presentaban en el transcurso del proyecto o para aplicar herramientas más

eficientes que iba encontrando en el transcurso del tiempo. En los siguientes párrafos se mencionan algunos de esos retos:

- Uno de los primeros problemas que tuve que enfrentar fue relacionado con el manejo de los datos de entrada de los bots, ya que preferiblemente estos debían estar en un archivo externo al código fuente del bot. Unos datos son constantes, pero otros contienen el nombre del archivo a almacenar, por lo que incluyen en su estructura la fecha que corresponde al día en que se creó la información. Las variables que contienen datos de fecha se deben actualizar diariamente, de ese modo las variables cada día son diferentes, además se manejan diferentes formatos de fecha y se debía tener en cuenta que la información se descarga en su mayor parte el día hábil después de generarse, esto implica que el bot debe descargar el lunes la información del viernes o en caso tal que sea festivo la información del viernes se descarga el martes. Así que un formato txt no era el más conveniente para almacenar los datos de entrada del software RPA y, por otro lado, hacer la construcción de las variables de nombres de archivos en el lenguaje de Automation Anywhere ocupaba muchas líneas de código. La mejor opción para almacenar las variables fue escribirlas dentro de un archivo de Excel, que tiene varias ventajas frente a un archivo txt.
- Para descargar la información fue necesario calcular la fecha del último día hábil, el mayor desafío era diferenciar entre un fin de semana normal, un fin de semana con festivo y la semana santa. Hice un primer desarrollo en Automation Anywhere, pero ocupaba mucho espacio en el editor de código fuente y era necesario grabar en variables del software RPA las fechas de los días festivos. La solución fue usar una lista con la fecha de los días hábiles. Con este archivo se puede obtener la fecha del último día hábil en un solo comando de Automation Anywhere.
- Otro reto que se presentó durante el proyecto fue verificar que al finalizar la ejecución del asistente la información realmente quedara descargada, pues en algunos casos es posible que un bot finalice exitosamente su trabajo y no obtenga ningún dato. Para solucionar esto, se implementó un bot que tiene la función de administrar la ejecución de los demás bots, encargándose de verificar que toda la información quedó almacenada, en caso contrario vuelve a ejecutar el bot encargado de descargar el archivo faltante.

- El comando de clonación de objetos a veces presentaba en ejecución problemas al interactuar con la interfaz gráfica de cierta aplicación, esto se debía a la variación de alguna propiedad del elemento de la interfaz que se iba a accionar. Para solucionar este problema se seguía alguna de las siguientes alternativas: usar el comando de manejo de controles de ventanas o usar los atajos de Windows para acceder a los menús, pues, en varios bots la manera más conveniente de acceder a algún elemento de la interfaz de usuario fue por medio de teclas especiales como Tab, Alt, Enter y Ctrl; como último recurso cuando las dos alternativas mencionadas no funcionaban, se insertaban las coordenadas (x,y) del elemento con el cual se quería interactuar, no era la mejor solución porque esta es inestable ante un cambio de la posición de la ventana en la pantalla, pero algunas veces fue la única manera de programar lo que se quería realizar. Para que esta última opción fuera segura, se adecuaba todo para que las coordenadas no cambiaran.
- La mayoría de los comandos de Automation Anywhere usan el nombre de la ventana de Windows que van a intervenir como parámetro de entrada, pero en algunas ocasiones la ventana que se debía manipular poseía un nombre que contenía una fecha en su estructura. Se encontró solución usando el comodín (*) para reemplazar la parte del nombre de la ventana que variaba el tiempo.
- Los bots requerían muchos comandos y se hacían difíciles de entender debido a que en el principio del proyecto desconocía la programación de VBA en Excel. VBA disminuyó en gran medida la cantidad de líneas de código en las nuevas versiones de los bots, ya que parte de la automatización se realizó con macros de Excel.

En las versiones iniciales no había manejo de errores, ni un archivo donde quedara registrados los fallos durante la ejecución del asistente, estas funciones se agregaron a medida que transcurría el tiempo. Así se fueron agregando nuevas funcionalidades a cada bot

En los primeros dos meses de la práctica la programación del asistente se hizo de manera presencial en las oficinas de la Mesa de Dinero. Durante este periodo el equipo de tecnología de Bancolombia le realizó mantenimiento a un computador del asistente, en el proceso actualizaron la versión de Microsoft Excel porque era anterior a la versión oficial que se usa en el banco y se cambió la configuración de

la interfaz gráfica de una aplicación. Estos cambios no fueron reconocidos por el software RPA, por tanto, fue necesario actualizar la automatización que cada bot realizaba sobre Excel y empezar de nuevo el bot que interactuaba con la aplicación de la Mesa de Dinero actualizada.

En los primeros dos meses quedaron programados casi todos los bots del asistente, y durante seis semanas, que fue el tiempo en que empezó la cuarentena, hasta que tuve acceso por medio de escritorio remoto a los computadores del banco, el software desarrollado funcionó sin supervisión. En tres ocasiones asistí de manera presencial para realizar ajustes a los bots. Durante este periodo se presentaron varias fallas en todos los procesos del asistente.

Luego de estas seis semanas, de manera remota continué con el proyecto hasta su finalización. De todos los bots desarrollé nuevas versiones que solucionaban los problemas presentados durante la contingencia por la pandemia.

- **Tercera etapa:**

En la última etapa se efectuó la evaluación del desempeño del asistente, donde se verificaba que cada bot cumpliera con los siguientes requisitos: No debía fallar ninguna línea de código y el software debía estar en capacidad de dar un correcto manejo a las ventanas que aparecían ocasionalmente. En los primeros dos meses se construyó una bitácora donde anotaba los resultados de cada ejecución. Con la información recolectada hacía las debidas correcciones a las instrucciones del bot. Cada vez que se corregía un error el bot resultaba más estable. Después, durante las seis semanas que no tuve acceso a los computadores de la Mesa de Dinero, el asistente dejaba registrada en unos archivos de texto toda la información que necesitaba para mejorar el funcionamiento del software RPA, estos archivos contenían la descripción de los errores que se presentaban en cada bot. Luego, cuando tuve el acceso de manera remota a los equipos de la Mesa de Dinero, usé la información recolectada para hacer nuevas versiones de los bots. La información recolectada en casi cuatro meses, la de la bitácora más la de los archivos de errores, resultó muy útil para programar la manera de manipular la mayor cantidad de escenarios posibles presentados por las aplicaciones de mesas de dinero. Cuando se logró llegar a la estabilidad en cada tarea se le agregó una sección de código para el monitoreo en tiempo de producción realizado parte del Cedex. Durante el tiempo de evaluación se pudo determinar el periodo de duración de las

contraseñas en los aplicativos y se encontró que en la mayoría de los casos se debe cambiar mensualmente.

- **Fin del proceso de configuración y paso a producción del asistente robótico:**

El proceso de programación para cada tarea tenía las tres etapas antes mencionadas, pero en cada bot se manejaban tiempos diferentes. Mientras que algún bot estaba en etapa de diseño, había otros en etapa de desarrollo y otros en evaluación. Una vez terminado el desarrollo del asistente, quedó listo para su paso a producción. Durante el proceso de paso a producción se cargaron los ejecutables del software RPA al servidor del Control Room, se entregó la documentación al Cedex, se gestionaron los respectivos permisos de ejecución y se llenaron varios documentos con información del asistente robótico. Después hice una presentación al equipo de trabajo de gerencia a la que pertenecí, informando los resultados del proyecto. En el mes de marzo se pasó la responsabilidad de la descarga de la información de mesas de dinero a esta gerencia. Por tanto, fue necesario enseñar el proceso que fue automatizado a los nuevos encargados de este. Luego, expliqué el funcionamiento del asistente robótico a las personas destinadas de supervisarlos, de esta manera quedó oficializada la entrega del proyecto al Grupo Bancolombia.

Resultados y análisis

El asistente desarrollado está compuesto por 17 bots, dos de estos son administradores de la ejecución de los 15 bots restantes. En un computador se ejecutan 7 bots, que descargan 11 archivos de 3 aplicaciones de la Mesa de Dinero. En el otro computador se ejecutan 10 bots, que descargan 13 archivos de 7 aplicaciones. En la tabla 2 se detalla la información antes mencionada. Los bots administradores están programados para empezar a la 7:30 am. A esta hora descargan la información del día anterior y en la tarde descargan la información que se generó en ese mismo día. Los archivos que se descargan en la tarde solo están disponibles por un día. Los bots encargados de descargar esta información volátil tienen una sección de código encargada de enviar un correo a la persona encargada del proceso, notificando el resultado de la ejecución.

Tabla 2. Información general del asistente robótico para la descarga de insumos de la Mesa de Dinero.

Usuario Windows	Bots administradores	Hora de ejecución	Bots	N° insumos
RPAADI01	Manager_RPAADI01	5:00:00 p. m.	BVC X-Stream	3
		7:30:00 a. m.	Set-fx A	4
			Bot Liquidaciones	1
			Bot Liquidaciones acciones	1
			Bot Órdenes acciones	1
			Bot Órdenes renta fija	1
RPAADI02	Manager_RPAADI02	7:30:00 a. m.	Set-fx B	4
			Master Trader 1	1
			Master Trader 2	1
			Master Trader 3	1
			Bot_App_Int	2
			CRCC	2
			PIP	1
		3:00 pm, 5:00 pm	Master Trader Op Mercado	1
5:00:00 p. m.	CRCC conexión cámara	0		

- **Descripción del software RPA:**

Los parámetros de diseño que se aplicaron durante el desarrollo del asistente fueron los siguientes:

- Los datos de entrada que usa el asistente robótico que están sujetos a cambiar en el tiempo, como la contraseña o el nombre de usuario, se obtienen desde un archivo de Excel al empezar la ejecución del bot y se almacenan en variables. Esto es muy útil cuando se debe actualizar alguna contraseña de un aplicativo de la Mesa de Dinero, ya que en tal caso no es necesario modificar la programación de Automation Anywhere. Unos datos que se obtienen desde el archivo de Excel son los nombres de los archivos descargados, estos tienen en general una estructura compuesta por un texto seguido por la fecha del día en que la información se creó, debido a esto, se organizó el archivo de tal manera que siempre que se abre actualiza los datos que almacena para que esté sincronizado con la fecha de descarga. En una hoja de cálculo de Excel se colocó una lista con las fechas de los días hábiles de los próximos 20 años y en otra hoja se escribe la fecha del día hábil actual, la fecha del día hábil anterior y la fecha del día hábil siguiente, las tres se escriben en varios formatos, según se necesite. En este

archivo también se incluyó la ruta de las aplicaciones usadas en la Mesa de Dinero, de donde se descarga la información y todos los demás datos que necesite el software RPA.

- El asistente se implementó en dos computadores, pues no todas las aplicaciones de mesas de dinero están en un mismo computador. Adicionalmente, todo el proceso se dividió en varios bots, de ese modo fue más sencillo el proceso de programación. También, con el código separado en varios archivos es más comprensible por parte de otro programador, cuando se requiera hacerle mantenimiento.
- Los procesos que requerían manejo de Excel se decidió implementarlos en macros. Las macros tienen la ventaja de ser más estables y rápidas que Automation Anywhere cuando se manejan archivos de Excel. Otra ventaja de las macros es que disminuyeron las líneas de código de los bots. Las macros se encargan de tareas como:
 - Cambiar formato del archivo, por ejemplo: de xls axlsx.
 - Cambiar el formato de una columna de tipo numérica a texto.
 - Eliminar columnas y registros de una tabla.
 - Enviar correo electrónico por medio de un módulo que se conecta al servidor de correo electrónico de Outlook. Si en algún momento se requiere cambiar el destinatario del correo se puede cambiar directamente en el código de Excel sin intervenir en el archivo de Automation Anywhere del bot.
- Después de la sección de lectura de datos desde el archivo de Excel, todo el código fuente quedó dentro de bloques de manejo de errores, para dejar un registro del error en caso de presentarse. El registro contiene los siguientes datos acerca del error: el tiempo en que se presentó, la línea de código donde se detuvo el bot y su descripción.
- Se desarrolló un bot en cada computador que cumple con la función de gestionar la ejecución de las demás tareas. Este bot cuenta con la capacidad de repetir aquellas tareas que no se completaron satisfactoriamente, o cuando el insumo no pase la prueba de validación donde se establece que la descarga fue exitosa.
- Cuando se finaliza el proceso de descarga de la información, este es verificado por medio de una macro de Excel. En ella se establecieron una serie de reglas para certificar que cada archivo cumple las características para ser insumo. Ahora se detallará en que consiste la validación de la

información. Para algunos archivos descargados basta sólo con verificar que sí existe. En otros archivos es importante contar la cantidad de registros debido al siguiente motivo: Hay un par de archivos que se descargan de una misma aplicación de mesas de dinero el día hábil siguiente en que se crea la información, pero esta aplicación necesitan de otro programa para descargar los datos desde un servidor el mismo día en que se realizaron las transacciones comerciales, cuando este programa no se conecta el mismo día en que se realizaron las transacciones, la información no se descarga del servidor, por tanto, al día siguiente cuando se descargan los dos archivos, estos aparecen vacíos. En otros archivos se verifica el orden y el nombre de los campos de la tabla. En caso de que los archivos fueron descargados, pero no cumplieron con las reglas antes mencionadas, son eliminados para no causar errores en el software que los usa para hacer los monitoreos.

- La información se almacena en carpetas separadas por cada nombre de aplicativo de mesas de dinero. A su vez, en algunos casos estas carpetas se dividen en carpetas organizadas por año, y varias de estas carpetas se dividen en carpetas mensuales. Antes de descargar la información se verifica donde sea necesario, que la carpeta en la cual se va a almacenar ya existe, si no, esta se crea al inicio de la ejecución del bot.
- El código fuente de un bot tiene generalmente tres secciones principales. En la primera parte se cargan datos en las variables del bot desde un archivo de Excel. En la siguiente sección se encuentra la parte principal del programa que a su vez está compuesta por las siguientes subpartes: la primera es la adecuación del entorno que cierra ventanas o programas que puedan interferir con el proceso de descarga, luego están las instrucciones de la respectiva navegación dentro de la aplicación de mesas de dinero, después, en la parte final, en la mayoría de bots se llama a alguna macro que gestiona el proceso de almacenamiento de la información descargada y finalmente se cierra la aplicación de donde esta se obtiene. Toda esta última porción de código mencionado está encerrada dentro de un bloque de manejo de errores para escribir en un archivo de texto la causa del error, si este se presenta. En la parte final del código fuente está la implementación de un metabot encargado del proceso de monitoreo de la ejecución del bot.

El código fuente del asistente quedó debidamente comentado, tanto el código de las macros como el de los bots de Automation Anywhere.

En el momento de iniciar el proyecto existían varias funciones escritas en VBA para realizar adecuaciones a la información cuando se descargaba manualmente. Estas funciones se integraron en las diferentes macros creadas para el asistente robótico, donde se adicionaron otra serie de pasos necesarios para preparar la información para su uso posterior.

- **Proceso de ejecución del asistente robótico:**

La ejecución del software RPA desarrollado se hace desde el servidor del Control Room, por lo tanto, es necesario que cada computador tenga conexión a este servidor. Para llevar a cabo el proceso de obtención de la información de mesas de dinero se ejecutan solamente los dos bots administradores. La orden de inicio de cada uno de estos bots se puede programar para realizarse automáticamente en el momento indicado o se puede hacer manualmente. En la figura 4 se observa la ventana de ejecución que despliega Automation Anywhere en la parte inferior izquierda de la pantalla. Esta ventana contiene el nombre del bot, la línea de código y el comando que se está ejecutando, también, tiene dos controles para pausar o detener el proceso. La ventana de Automation Anywhere estará activa desde que empiece a ejecutarse el primer bot a las 7:30 am hasta cuando termina la ejecución del último bot luego de las 5:00 pm, en ambos computadores se maneja el mismo horario.



Figura 4. Ventana de ejecución de Automation Anywhere.

- **Monitoreo de las ejecuciones del software desarrollado:**

Para el monitoreo del asistente en la etapa de producción se usa un metabot, el cual se encarga de agregar registros en una base de datos que es monitoreada por el Cedex. La información insertada en la base de datos contiene reportes de la ejecución del bot, tales como: Tarea finalizada exitosamente y ocurrió un error. Los errores que pueden ocurrir son los siguientes: error no controlado o error de programación, datos ingresados no válidos, la aplicación de donde se descarga la información no responde, clave fallida, y clave próxima a vencerse. Con la base de datos se puede hacer seguimiento a las ejecuciones del asistente e informar a tiempo acerca de errores de ejecución a las personas que quedaron encargadas del asistente.

- **Descripción de la documentación del asistente robótico:**

Se escribió una documentación con información general del asistente y con información específica de cada bot. Esta documentación incluye un archivo con la descripción de la configuración general del proyecto, que describe una serie de pasos que se deben seguir en cada computador para que el asistente se ejecute correctamente. Estos pasos son para realizar las siguientes acciones: Habilitar Excel para que ejecute macros y envíe correos electrónicos, habilitar la opción de Auto-Login de Automation Anywhere, entre otras configuraciones específicas de las aplicaciones que se intervienen con la automatización. El Auto-Login es una función de Automation Anywhere que permite ejecutar el asistente aun cuando Windows esté bloqueado. Para habilitar esta funcionalidad, hay que almacenar las credenciales de acceso de Windows y permitir que se realice el inicio de sesión automáticamente. En general, la documentación contiene las tres siguientes partes: datos técnicos de la aplicación que se interviene y del proceso de descarga de datos, requisitos de funcionamiento del asistente robótico e información para mantenimiento. En los requisitos de funcionamiento se menciona la ruta donde debe estar el código fuente y la ubicación de los archivos usados por el software RPA. La información para mantenimiento contiene una imagen con todos los datos que se extraen de Excel y la celda donde estos se ubica dentro del archivo que los contiene; además se incluye las ubicaciones en Windows de los archivos de datos externos y del registro de errores de cada bot.

- **Limitaciones del software desarrollado:**

El asistente no funcionará si se presenta cualquiera de las siguientes situaciones:

- Cuando suceda una actualización de las aplicaciones de mesas de dinero que modifique las propiedades de la interfaz gráfica.
- En caso de haber un cambio en el formato de almacenamiento de la información dentro de los aplicativos de mesas de dinero.
- Cuando alguno de los computadores esté apagado.
- Cuando no haya conexión a la red del banco.
- Cuando no haya conexión con el servidor del Control Room.
- Cuando expiren las contraseñas de acceso a los aplicativos de mesas de dinero.
- Cuando se acabe el espacio de almacenamiento del servidor de Windows donde se guarda la información descargada.

Cuando cualquiera de los primeros dos escenarios antes mencionados se presenta, el asistente por sí solo no puede solucionar nada, es necesario actualizar el software RPA. En cualquiera de las siguientes tres situaciones de la lista, ni siquiera puede empezar a ejecutarse el asistente. Y cualquiera de los dos últimos eventos mencionados se pueden evitar.

Cuando el sistema operativo de Windows se demora procesando las ordenes un tiempo mayor en comparación al tiempo que normalmente se tarda, el software RPA no funciona como se esperaría y la mayoría de bots fallan en el proceso de descarga de la información.

- **Resultados cuantitativos:**

El trabajo que el asistente robótico está encargado de realizar estaba a cargo de 5 personas, donde el tiempo de descarga promedio era de tres minutos por cada archivo. Si multiplicamos este tiempo por el total de archivos descargados, que son 24, obtenemos el total de 1 hora y 12 minutos de tiempo empleado diariamente para descargar la información de mesas de dinero por parte del personal humano de la empresa. El asistente ejecuta dos procesos de descarga de información en paralelo, uno en cada computador, un proceso tiene una duración de 50 min en promedio y el otro de 40 min en promedio. Como ambos procesos se realizan a la vez, el tiempo de ejecución del asistente robótico tiene una duración de 50 min, 2 minutos en promedio por cada insumo.

Conclusiones

En este proyecto se desarrolló un asistente robótico usando la tecnología RPA, por medio del framework de Automation Anywhere. Con este asistente se automatizó la descarga, adecuación y validación de la información de mesas de dinero del Grupo Bancolombia, para su posterior uso como insumo de los procesos analíticos de vigilancia y monitoreo realizados por la Vicepresidencia de Cumplimiento.

Según [7], Bancolombia había implementado hasta el 2018 cientos de bots en el primer año que llevaba haciendo uso de Automation Anywhere. El asistente robótico implementado en este proyecto es parte del desarrollo del plan trazado por la organización en el año 2018 de expandir la fuerza de trabajo digital y combinar las capacidades humanas, RPA, cognitivas y analíticas para dar a los clientes una experiencia de clase mundial.

Estas son las principales ventajas adquiridas con la puesta a producción del software RPA: no se requiere que alguna persona se desplace hasta el sitio de la Mesa de Dinero para hacer el proceso manualmente, se ahorra tiempo por parte de la gerencia encargada de los monitoreos y se evitan errores humanos que podrían presentarse en el proceso.

En el desarrollo del proyecto hice uso del lenguaje de programación de Visual Basic, el cual no conocía y aprendí durante el desarrollo de la práctica académica. También usé herramientas como macros y scripts que utilizan el lenguaje de programación antes mencionado. Encontré que las macros resultan útiles para automatizar procesos relacionado con Excel y los scripts son adecuados cuando se requiere interactuar con comandos el sistema operativo de Windows. Además, estudié y aprendí la tecnología de RPA para automatizar procesos por medio de la implementación de bots.

La parte más difícil en el desarrollo de un bot es cuando en el proceso de programación de la interacción con una aplicación, algún elemento de la interfaz gráfica de esta no es reconocido por el framework de programación, pues hay que buscar exhaustivamente varios métodos para continuar con el proceso de implementación del código.

Los bots se fueron robusteciendo a medida que adquiría más experiencia en el manejo de las aplicaciones de la Mesa de Dinero, del sistema operativo de Windows y de las herramientas de desarrollo provistas por Automation Anywhere;

esto se sumó a la información recibida por parte de las personas que realizaban el proceso manualmente.

Las principales recomendaciones para hacer un asistente robótico son las siguientes:

- Realizar una lista con las variables a usar.
- Observar el comportamiento de la aplicación con la que se va a interactuar, por lo menos por un mes. Así se cubren la mayor cantidad de escenarios que se pueden presentar.
- Llevar un registro de todos los errores que se presentan durante el desarrollo del software RPA.
- Usar diferentes herramientas en la configuración de la automatización, tales como scripts y macros de VBA, los cuales fueron usados en este proyecto. Las herramientas que se van a usar deben ser compatibles y reconocidas por el framework de desarrollo de software RPA.
- Separar el software en varios módulos, para hacer más fácil el proceso de desarrollo y evaluación del código fuente.

Para el software de RPA desarrollado se pueden considerar las siguientes oportunidades de mejora:

- Notificar por medio de correo electrónico a los administradores del asistente robótico, acerca del próximo vencimiento de las contraseñas de los aplicativos con dos días de anticipación.
- Hacer uso de módulos ya implementado por otras áreas del banco para disminuir la longitud del código fuente.
- Guardar una copia de la información descargada en el disco local cuando la ubicación de red compartida quede sin espacio de almacenamiento.

Referencias Bibliográficas

[1] ¿Qué es una mesa de dinero? [En línea]. Jannier Duque Navarro. 30/06/2018. [Citado en junio de 2020]. Disponible en: <https://www.abcfinanzas.com/administracion-financiera/que-es-una-mesa-de-dinero>

[2] Formatos de archivo que admite Excel. [En línea]. (s.n.). (s.f.). [Citado en junio de 2020]. Disponible en: <https://support.microsoft.com/es-es/office/formatos-de-archivo-que-admite-excel-0943ff2c-6014-4e8d-acea-b83d51d46247?ui=es-es&rs=es-es&ad=es>

[3] ¿QUÉ ES UNA MACRO DE EXCEL? [En línea]. Moisés Ortiz. (s.f.). [Citado en junio de 2020]. Disponible en: <https://exceltotal.com/que-es-una-macro-de-excel/>[4] Automatización robótica de procesos (RPA). [En línea]. (s.n.). (s.f.). [Citado en junio de 2020]. Disponible en: <https://www.uipath.com/es/rpa/automatizacion-robotica-de-procesos>

[5] Descripción general de Enterprise Control Room. [En línea]. (s.n.). (s.f.). [Citado en junio de 2020]. Disponible en: <https://docs.automationanywhere.com/bundle/enterprise-v11.3/page/enterprise/topics/control-room/getting-started/control-room-overview.html>

[6] Automation Anywhere – MetaBots Basics. [En línea]. (s.n.). 2020. [Citado en junio de 2020]. Disponible en: <https://rpaknowledge.com/automation-anywhere-metabots-basics/>

[7] Bancolombia automation anywhere case study. [En línea]. (s.n.). Julio de 2018. [Citado en junio de 2020]. Disponible en: <https://resources.automationanywhere.com/case-studies/bancolombia>