PISTE: Herramienta de automatización de Bancolombia para el seguimiento de los casos de créditos de consumo.

Johan Alexis Berrío Arenas

Informe final de prácticas académicas para optar al título de Ingeniero Electrónico

Tutores

Felipe Cabarcas Jaramillo
Asesor interno
Profesor Facultad de Ingeniería

Lina Marcela Peña Gallego
Asesora externa
Dueña de Producto, Bancolombia

Universidad de Antioquia

Facultad de ingeniería

Departamento de Ingeniería Electrónica y Telecomunicaciones

Medellín, Colombia

2022

TABLA DE CONTENIDO

Resumen

En modelos de negocio como los bancarios, que suelen manejar una gran cantidad de datos entre información de clientes, transacciones y demás, es una necesidad de las entidades contar con herramientas que permitan tener acceso y control a dichos datos. La herramienta que se diseñó para la extracción de información y seguimiento automáticos de los créditos de consumo generados por la entidad financiera Bancolombia permite a sus analistas el acceso a una nueva base de datos que recopila cada movimiento del proceso respecto a cada cliente gestionado desde diciembre de 2021 por las empresas contratadas por el Banco y que, hasta la fecha, eran los únicos con un acceso global a toda la gestión.

PISTE (como fue nombrada la herramienta desarrollada) ha gestionado a la fecha más de 60000 casos de créditos de consumo de clientes de Bancolombia, lo que permitió examinar falencias, observar errores comunes en el proceso y tomar decisiones enfocadas en detectar y corregir los casos en que no se llevaba el debido proceso de cara a la respuesta al cliente. Por otro lado, PISTE permitió hacer un sondeo de qué aliados del Banco estaban cometiendo más infracciones respecto a la venta de créditos de consumo de forma telefónica y, usar dicha información para realizar los respectivos seguimientos.

Introducción

Los robots informáticos (también conocidos como Bots) son programas que se encargan de automatizar tareas repetitivas como si se tratase de un ser humano y de una forma más rápida, estas pueden ir desde enviar cantidades masivas de correos hasta la descarga de información desde páginas web. El uso de estos robots se ha vuelto muy recurrente en las empresas pues libera mano de obra y acelera los procesos.

El *web scraping* es una técnica utilizada para acceder a sitios web y realizar distintas operaciones sobre estos de forma automatizada, es decir, mediante el uso de Bots. Una de las acciones que se puede realizar mediante este tipo de herramientas es la extracción de datos para su posterior análisis.

En ambientes con flujos de datos muy altos, como lo son los bancos, es necesario llevar control sobre esta información debido a su sensibilidad e importancia. Sin embargo, teniendo en cuenta su cuantía, este proceso puede tomar mucho tiempo y ser repetitivo, si se realiza de forma manual. Por otro lado, la privacidad en información de los usuarios es el deber ser de una entidad financiera como Bancolombia. Debido a su gran expansión mercantil, la empresa terceriza parte de sus procesos con aliados con los que comparten informes constantemente, pero que no pueden acceder a la red interna donde se encuentran las bases confidenciales y deben apoyarse en herramientas como Bizagi que, entre otras cosas, sirve de puente seguro para la transmisión de datos. En casos como el anterior entran en juego los recursos de análisis y procesamiento de datos como los ofrecidos por Python.

El robot de extracción de datos será una herramienta interna del banco que permita el ingreso automático a Bizagi y la descarga diaria de los lotes de información referentes a los créditos de consumo de sus usuarios. Además, procesa y filtra dichos informes y los cruza, con el fin de generar una base de datos diaria que permita al personal bancario conocer de forma rápida y estructurada el estado de consumo.

Actualmente el banco cuenta con varios procesos automáticos que ingresan a Bizagi a extraer información, pero, debido a la complejidad de la plataforma, cada proceso es muy diferente a los demás. Por otro lado, es la primera vez que se busca realizar un seguimiento, por medio de software, sobre esta información; actualmente, quien esté interesado debe realizar el proceso manual de descarga y filtrado que puede tardar en promedio unos 10 minutos, pero suele hacerse sobre clientes específicos, no sobre las bases y no se guarda un registro de esto. Haciendo una aproximación y teniendo en cuenta que el robot de extracción se espera que procese más de 500.000 registros de entre 30 y 50 tablas diariamente, hacerlo de forma manual no es viable en ningún caso, lo que supondrá un gran beneficio en cuanto a disponibilidad de información se refiere.

1. Objetivos

1.1. Objetivo general

Desarrollar una herramienta en Python que, combinando los conceptos de *scraping* y bases de datos, permita la automatización del proceso de búsqueda, procesamiento y almacenamiento de información desde la aplicación Bizagi para el monitoreo de casos sobre créditos de consumo

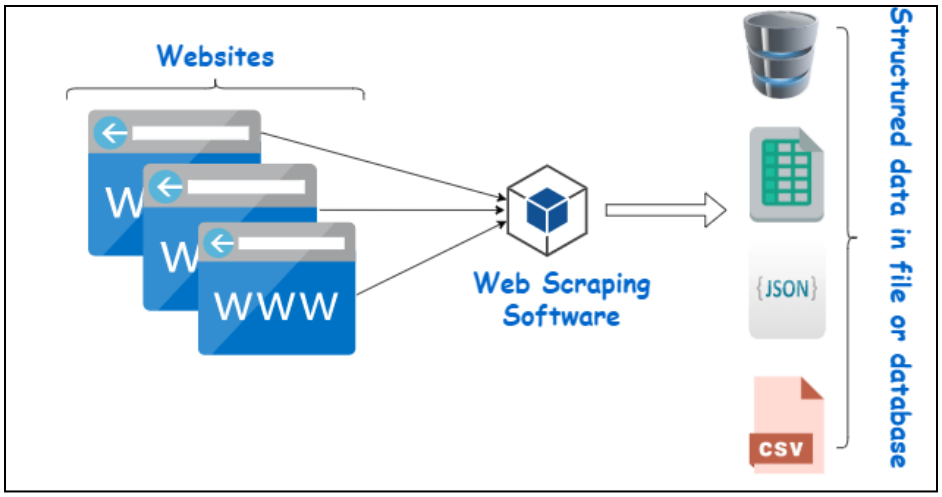
1.2. Objetivos específicos

- Desarrollar un *Bot* que, por medio de *scraping*, ingrese a Bizagi y genere tablas de créditos de consumo de las bases de datos para realizar un seguimiento de estos
- Crear una herramienta que genere un informe, a partir de las tablas obtenidas, de las solicitudes de usuarios que no se hayan validado, para la supervisión de los tiempos de respuesta
- Generar una base de datos con los informes de cada usuario en Bizagi que permita, mediante *scraping* de la página web, obtener las respuestas de todas las grabaciones de pagarés
- Crear una tabla en la base de datos de banco y cargar la información ya procesada para generar un histórico de todos los casos validados y culminados
- Validar los resultados que genera la herramienta comparándolos con los resultados que se realizan manualmente

2. Marco teórico

Web scraping

Por definición, el web scraping es el proceso de recolectar datos estructurados de la web de forma automática; existen muchas necesidades por las cuales se busca aplicar este desarrollo entre las cuales se encuentran el monitoreo de precios, estudios de mercado, recolección de datos, entre otros. Su utilidad en este proyecto radica en la posibilidad de ingresar a páginas web y descargar información, en este caso, los datos sobre los procesos de crédito que los aliados el banco ingresan en la aplicación Bizagi.



Gráfica 1. Metodología de web scraping.
Disponible: https://rpubs.com/andres1111/700136

La gráfica 1 explica el proceso general que desarrolla esta metodología de búsqueda de información en la web: El software desarrollado ingresa a las páginas necesarias y descarga los datos que se requieren, esto suele entregarlo en forma de bases de datos estructuradas en alguno de sus formatos (.xlsx, .csv, .json, entre otros).

Selenium WebDriver

El lenguaje de programación en que se implementará la totalidad del proyecto es Python. Este, gracias a su gran versatilidad a la hora de trabajar con grandes cantidades de datos y el hecho de ser un lenguaje multipropósito, lo hacen el candidato idóneo para esta tarea. Por el lado del web scraping se tiene a Selenium, una librería que, junto a WebDriver, permite el control de los

navegadores web a nivel del sistema operativo permitiendo la automatización de las acciones en el buscador.

Pandas

Para el procesamiento de los datos se usa a Pandas, la librería por excelencia en cuanto a análisis y manipulación de datos en Python se refiere; este paquete permite usar estructuras de datos (Dataframes) que facilitan las operaciones de filtrado, unión y cruce de tablas de datos.

Pandas es imprescindible en este proyecto también por su capacidad de lectura y escritura en formatos de archivos como .xlsx de Microsoft Excel pues las tablas en Bizagi son subidas de esa forma.

Hadoop

Hadoop es una estructura de código abierto que permite el almacenamiento y procesamiento de grandes cantidades de datos en clústeres de hardware comercial. Es altamente utilizado en temas de Big Data y analítica pues permite acceder de forma sencilla y rápida a las bases de datos que se almacenan mediante esta herramienta gracias a su enorme poder de cómputo.

Cloudera Impala

Motor que permite integrar SQL (Structured Query Language) a Hadoop para realizar consultas sobre los datos almacenados de una forma más nativa, es decir, sin necesidad de hacer transformaciones o movimientos extra sobre la data y por ende aumentando la velocidad y fluidez de procesamiento

SAS Studio

SAS se ha convertido en un enorme conjunto de programas que, entre otras cosas, trabajan en los campos de la automatización, gestión de datos y computación estadística [3]. Su posibilidad de trabajar con una interfaz de Impala permite que el acceso y almacenamiento de datos sea aún más sencillo, a la vez que, al tener acceso a servidores locales, puede cargar de forma automática información a bases de datos en la nube.

3. Desarrollo

Este proyecto se desarrolló bajo la metodología de cascada, donde el flujo de ejecución de las actividades es de forma secuencial y cada una depende de la anterior, es decir, la fase de búsqueda información general en la web da paso al primer filtrado de datos y del mismo modo, después de este se puede realizar una segunda búsqueda más específica para al final cruzar toda la información y terminar generando y almacenando la base de datos del día.

Datos: el conjunto de datos transaccionales utilizados son propiedad privada del BANCOLOMBIA y no puede revelarse información sensible referente a los clientes, ni de detalles de valores históricos de ventas. Por ende, se opta por ocultar información específica como nombres o identificadores reales.

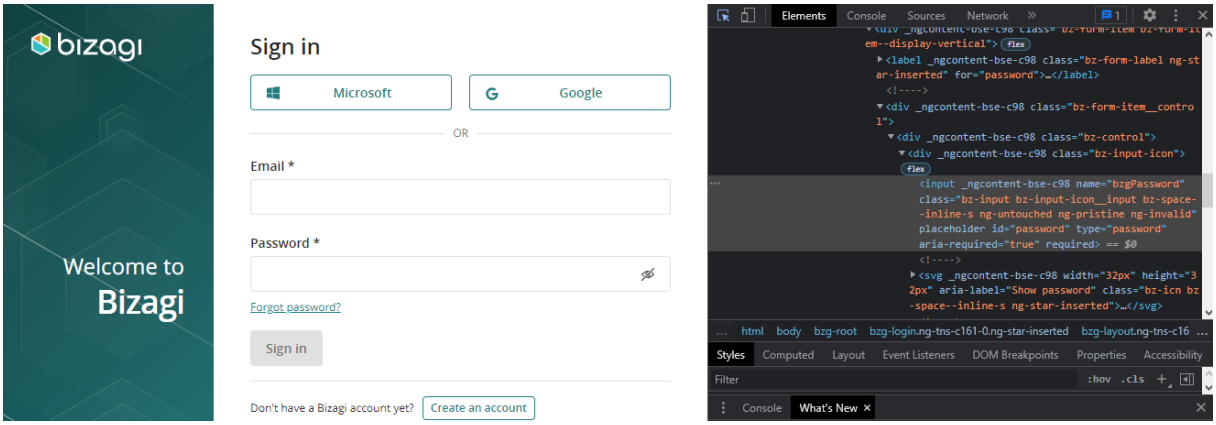
A continuación, se muestran las diferentes metodologías en la resolución del problema, tanto antes como después de iniciar la práctica profesional.

Inicialmente, sí el personal del Banco quería conocer el estado actual del proceso de asignación de un crédito de consumo o una tarjeta de crédito a un cliente, era necesario hacer una búsqueda inicial en el aplicativo Bizagi pues allí es donde converge la información que las empresas tercerizadas comparten con Bancolombia. Los datos son compartidos en hojas de Excel de forma diaria, es decir, dichas hojas no son específicas por caso si no que contiene todo lo procesado en el día. Teniendo en cuenta lo anterior, era necesario realizar un filtro por el caso específico y, a su vez, realizar una segunda consulta en Bizagi con ciertos criterios para llegar al objetivo inicial y obtener la información que se deseaba.

Mediante PISTE, todo el proceso se automatizó y se hizo de la siguiente forma:

Búsqueda general en Bizagi

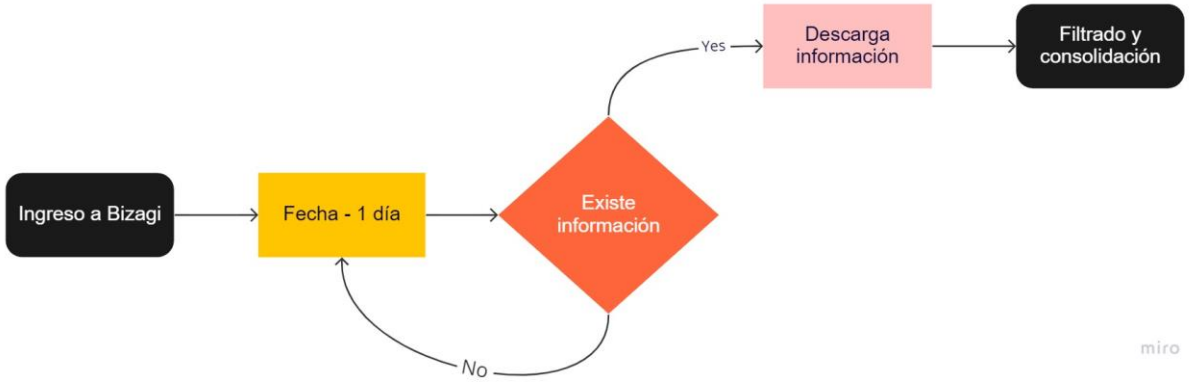
Todos los días, a las 8 A.M. el BOT que se desarrolló entra a Bizagi y descarga mediante web scraping todos los radicados que fueron procesados el día anterior. Para ello, se tuvo en cuenta que no se realizan ventas fines de semana ni días festivos y que, además, puede que no se haya gestionado información de un modelo de negocio, pero de otro sí. En este caso se hizo uso de la librería *Time* de Python para realizar las búsquedas mediante deltas de tiempo y evitar reprocesamientos o pérdidas de información. Una gran ventaja de la forma en que se estructuró la búsqueda es que no sólo se ha limitado la búsqueda a iniciar siempre en el día inmediatamente anterior, esta se puede configurar para consultar cualquiera fecha en el pasado. Lo anterior siempre depende del tiempo que permanezca almacenada a información en Bizagi pues esta herramienta es usada como una base de datos temporal debido al enorme flujo de datos que soporta diariamente.



Gráfica 2. Método de revisión del código fuente de Bizagi.

Cuando se detectaron datos, el robot los descarga y, como se mencionó que las hojas de Excel contienen datos que no son relevantes para el proceso, debe realizarse un filtro en cada una de estas y generar un consolidado también en Excel que se nombró como "requerimientos generales". Todo el proceso de construcción de tablas se realizó con Pandas y sus DataFrames que son estructuras de datos bidimensionales que se asemejan a hojas de cálculo y que tienen muchas funciones de gran utilidad para consultar, filtrar y organizar información.

El proceso se resume de la siguiente manera:



Gráfica 3. Proceso general para la extracción de requerimientos generales desde Bizagi.

Búsqueda caso por caso en Bizagi

Con el informe que se generó en la primera etapa, el RPA debe ingresar de nuevo a Bizagi y realizar una búsqueda más exhaustiva haciendo uso del controlador Selenium, caso por caso, de cada gestión y a su vez, cada caso debe ser filtrado con diferentes criterios dependiendo del modelo de negocio al que pertenezca, en total, fueron 3 modelos de negocio analizados y añadidos a la búsqueda.

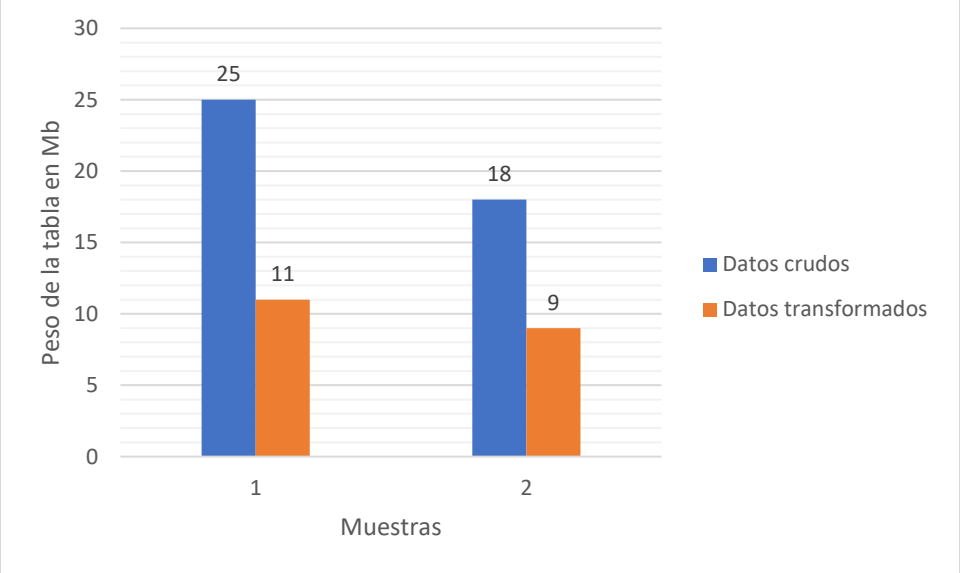
Este informe, entre otras cosas, muestra el estado actual de aprobación de los créditos para el día consultado, las causales de rechazo y las observaciones. Los analistas han hecho uso de esta información para detectar posibles oportunidades de mejora en los modelos.

Base de datos

Se creó una base de datos que contiene los informes procesados diariamente por el robot, para ello se utilizó Impala y Hadoop.

Mediante consultas SQL implementadas en Impala, se logró crear una tabla con la estructura y tipos de datos necesarios para posterior carga de información. Cabe resaltar que, para efectos de optimización en el almacenamiento, se aplicaron funciones de transformación de datos con el fin de reservar la menor cantidad de memoria posible sin incurrir en pérdidas de información.

Se realizó una comparación entre la ocupación de memoria de dos cargas de datos crudos y optimizadas teniendo en cuenta que en ambos se tiene la misma información, pero almacenada en tipos de datos diferentes en algunos casos.



Gráfica 4. Comparación de ocupación de memoria para datos crudos y procesados.

Automatización

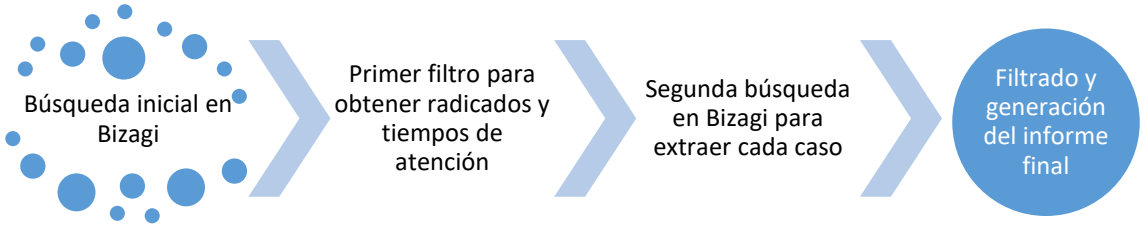
Aunque ya se mencionó que el proceso de extracción de datos se ejecuta de forma automática, el almacenamiento en la base de datos no se puede incluir en esa etapa pues no se cuenta con una integración entre Python e Impala que permita realizar el procesamiento de optimización para el conjunto de datos final que se busca almacenar en la nube. Teniendo en cuenta que era necesaria una herramienta intermediaria, se optó por SAS Studio y su posibilidad de

combinación el formato de impala, de esta forma, el proceso de automatización y almacenamiento final se dividió en varias aplicaciones de la siguiente manera:

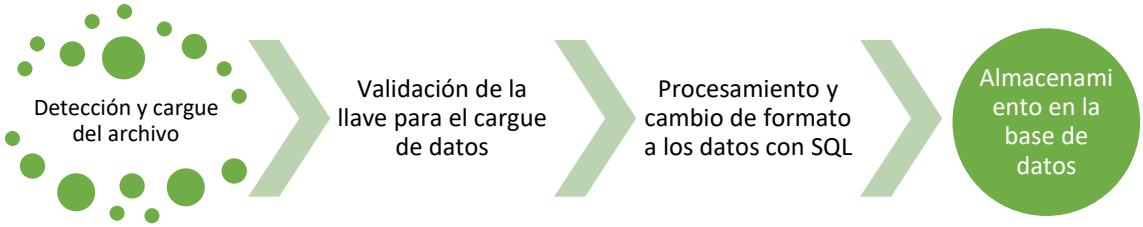
1. Python entrega en una ruta específica el set de datos en Excel resultante de las etapas de web scraping y filtrado
2. SAS Studio busca el archivo, lo carga y define los tipos de datos que cree convenientes
3. Mediante la integración de SAS Studio con Impala y, haciendo uso de SQL, se transforman los datos a su formato final y se añade la información a la base de datos

Los pasos 2 y 3 se ejecutan automáticamente pues fueron calendarizados, es decir, SAS permite definir intervalos de tiempo para realizar operaciones. Por otro lado, y con el fin de tener en cuenta los días en que no se generan informes (festivos, domingos, etc.), se definió una clave que, mediante Pandas, se agrega en la primera fila del conjunto de datos e indica si el archivo debe ser tenido en cuenta por SAS o no, de esta forma se evitaron problemas en SAS cuando intentaba procesar archivos inexistentes. En otras palabras, siempre se genera un archivo en Excel, aunque no se haya obtenido información de Bizagi para ser cargado en SAS, pero, se indicó cuando debe ser tenido en cuenta y cuando no.

El proceso completo de la automatización se muestra a continuación y se divide en dos partes, una que se realiza en Python y otra en SAS Studio.



Gráfica 5. Proceso de automatización realizado en Python.



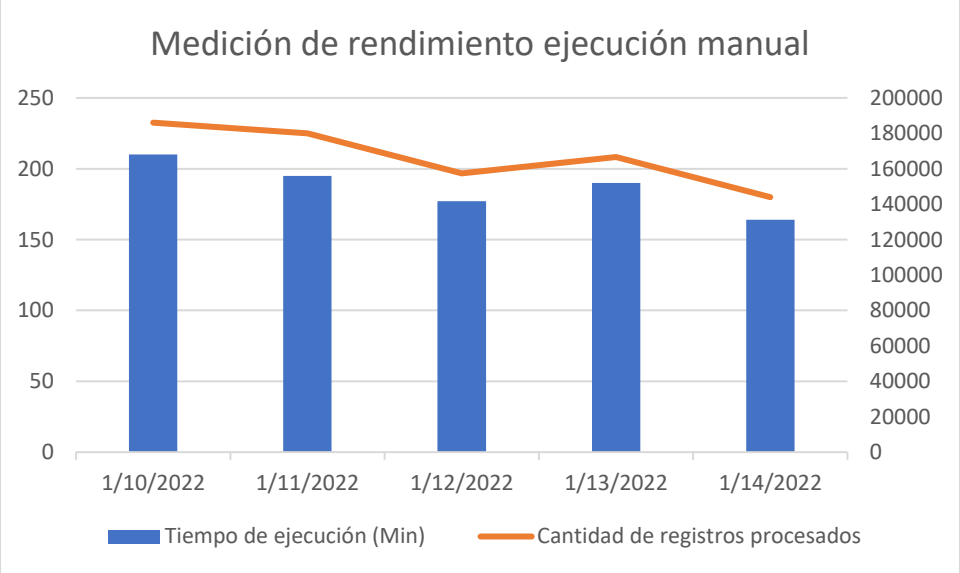
Gráfica 6. Proceso de automatización realizado en SAS Studio.

4. Resultados y análisis

En esta sección se muestran los resultados de PISTE, la herramienta para la extracción de datos de procesos de venta de créditos de consumo y tarjetas de crédito. Es importante volver a enfatizar que no es posible revelar información sensible de la compañía, por lo cual, se opta por mostrar todo de forma general para conjuntos de datos anónimos.

Cabe resaltar que, como este proceso no se venía ejecutando periódicamente de forma manual, se realizaron algunas ejecuciones de esta forma para tener una base de comparación. Aun así, teniendo en cuenta la cantidad de archivos que la herramienta procesa diariamente, el trabajo manual se hizo con menor cantidad de datos.

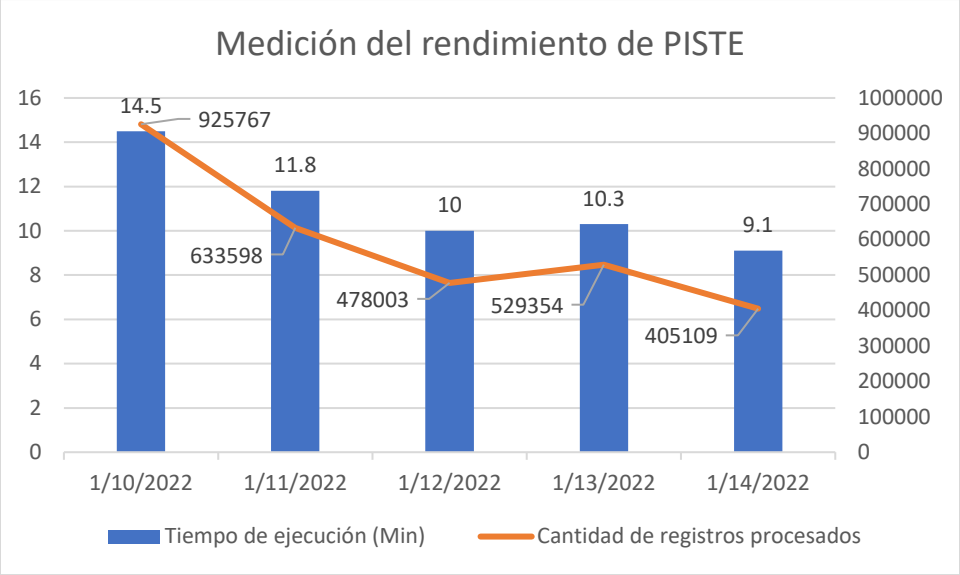
Inicialmente se realizaron las pruebas 5 días consecutivos que fueron desde enero 10 de 2022 hasta enero 14 de 2022, fechas en que se destinó parte del tiempo a validar también que los datos obtenidos tanto mediante la herramienta como de forma manual fuesen correctos.



Gráfica 7. Registros procesados y tiempo de ejecución manual.

La gráfica anterior presenta los resultados de las 5 ejecuciones manuales. Como se mencionó anteriormente, estas se realizaron con una porción de la cantidad total de datos. En promedio, de forma manual se logran procesar 166.000 datos en 3 horas, teniendo en cuenta que la cantidad total registros puede subir casi hasta el millón, no es nada optimo hacerlo de esta forma pues se consumiría gran parte por no decir que toda la jornada laboral.

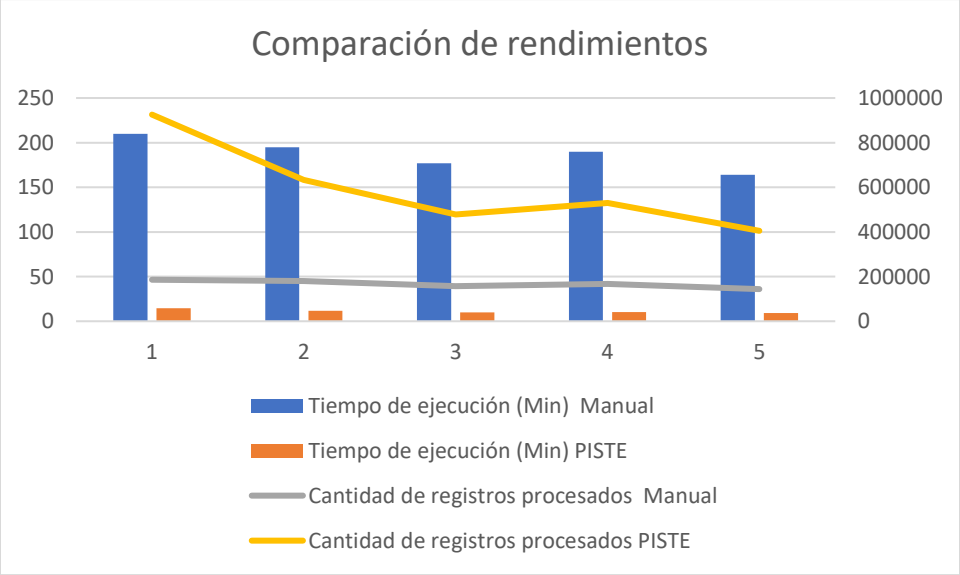
De forma simultánea, el Bot desarrollado se encargaba de hacer la ejecución automática y los resultados de los 5 días se presentan a continuación.



Gráfica 8. Registros procesados y tiempo de ejecución automática.

Se observa que la cantidad de archivos procesados aumenta en gran medida, se pasó de unos 166.000 a un promedio de 600.000, cifra que varía según el día pues a medida que se adentra en la semana, las peticiones de créditos de clientes tienden a disminuir respecto a los primeros días.

Por otro lado, y donde se observa la enorme diferencia y el poder de procesamiento automático es en el tiempo de ejecución que disminuye de 180 minutos a sólo 11, lo que equivale a una reducción de más del 94% del tiempo que tarda de forma manual.



Gráfica 9. Comparación de rendimiento entre ejecución manual y automática.

La comparación observada en la gráfica anterior permite ver de forma clara la gran diferencia respecto a los dos tipos de ejecución, tanto en la cantidad de registros procesados como el tiempo que se tarda en procesarlos. Indiscutiblemente el RPA cumple con su objetivo de automatizar correctamente el proceso y lo hace en tiempo récord. Gracias a esto, permite al personal Banco destinar capacidades a la realización de otras labores a la par que el registro en la base de datos permite el acceso y visualización de información antes transparente para el Banco.

Al finalizar el proceso de filtrado y cuando se tienen listos los datos para la su carga en la base de datos, estos presentan una estructura similar a la que se muestra en la tabla a continuación:

Tabla 1. Ejemplo de estructura de información consolidada luego del proceso en Python.

num_doc	tipo_doc	f_gestion	f_solucion	estado_agen	codase	…	Columna 31
1020564873	1	1/8/2022	1/14/2022	exitoso	32465	…	Fila 1
1017567654	1	1/7/2022	1/14/2022	exitoso	32465	…	Fila 2
21487569	2	1/5/2022	1/14/2022	no exitoso	32460	…	Fila 3
…	…	…	…	…	…	…	…
1021563025	1	12/30/2021	1/14/2022	exitoso	32458	…	Fila 2371

Donde las 31 columnas contienen toda la información referente al proceso; allí se puede encontrar tanto datos sobre el usuario que solicita el crédito, así como, por tratarse de un proceso tercerizado, información sobre el asesor que realizó el proceso, la empresa contratada y el estado actual de la gestión. Estos datos permiten, como se buscaba inicialmente, tener una visión total sobre la forma en que se está llevando a cabo el procedimiento para la venta de un crédito de consumo, conocer las oportunidades de mejora del negocio y realizar las modificaciones que sean necesarias a partir de la información suministrada.

Ahora pues, al usar SAS Studio para cargar la tabla y darles formato a los datos, se obtiene en dicha herramienta un resultado similar al que se muestra a continuación, donde es notorio el tratamiento que reciben los datos referentes a fechas:

Tabla 2. Ejemplo de estructura de información consolidada luego del proceso en SAS.

1	1689	8	03JAN1991
1	1005	2	04FEB1995
1	1406	0	16JUN1998
1	846	0	21JUL1998
2	912	0	24AUG1997
2	1198	0	13OCT1998
2	2447	6	23MAR1992
2	912	2	12JAN1995

Se observa que existen diferencias notorias respecto al tipo de datos que se hallan en Excel, específicamente con las fechas. Se asignó un formato *datetime* a sus fechas y esto cambia su estructura respecto a otros programas ya que, si se carga el dato crudo, muchas veces se generan valores incongruentes y que no muestran relación alguna con la realidad. La tabla generada en SAS Studio es la que finalmente se agrega a la base de datos del banco y puede ser consultada por los usuarios que cuenten con el acceso a esta información.

5. Conclusiones

Para reconocer el impacto de los resultados obtenidos con el proyecto realizado, es una buena opción remitirse a los objetivos propuestos inicialmente. La finalidad general del desarrollo era poder monitorear mediante un Bot codificado en Python, el proceso detrás de los créditos de consumo que vende Bancolombia a través de sus aliados y que comparten parcialmente mediante Bizagi; la herramienta que se desarrolló cumple a cabalidad esta tarea y ahora permite a los analistas del banco acceder a todas la etapas del proceso desde una sola base, tarea que antes ocupaba mucha capacidad pues era necesario remitirse a distintas bases para obtener la información. Además de la reducción de tiempo que conlleva hallar todo en un solo lugar y que los datos se actualicen de forma automática, es importante resaltar que esto abre la puerta a estudios del modelo y de negocio referentes al proceso, cuestión de la que antes era más complicada por la falta de información.

Otro aspecto importante para tener en cuenta es la escalabilidad, aunque la herramienta se desarrolló para un proceso bancario específico, no está confinada a sólo generar información de créditos de consumo; gracias a las buenas prácticas de programación que se utilizaron, PISTE puede adaptarse para extraer datos desde Bizagi de distintos modelos, eso sí, luego de un estudio profundo sobre la información necesaria y aún más importante, los filtros que permiten llegar a esta.

Ya se habló de la importancia que tiene la consolidación de la información mediante PISTE y la reducción de tiempos que aporta, pero hay algo muy importante para tener en cuenta y es que, gracias al nuevo conocimiento sobre el paso a paso de la gestión de venta de créditos de consumo, los analistas tienen la posibilidad de detectar con mayor facilidad fraudes y malas prácticas por parte de las empresas aliadas y/o sus asesores de venta.

Finalmente, se logró culminar cada uno los objetivos planteados de forma satisfactoria, desde el desarrollo del robot para la ejecución de web scraping en la página de Bizagi hasta la generación de tablas y almacenamiento en la base de Bancolombia, optimizando los tiempos de búsqueda de datos, proporcionando información que anteriormente no estaba consolidada y mejorando la seguridad de la gestión y el tratamiento de datos para el proceso venta de créditos de consumo de la Organización.

6. Referencias bibliográficas

Chapagain, A. (2019). Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, Regex, and others. Packt Publishing Ltd.

Gunawan, R., Rahmatulloh, A., Darmawan, I., & Firdaus, F. (2019, March). Comparison of web scraping techniques: regular expression, HTML DOM and Xpath. In International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018) Comparison (Vol. 2, pp. 283-287).

Impala. (2021). Retrieved 3 February 2022, from https://impala.apache.org/overview.html

Kornacker, M., Behm, A., Bittorf, V., Bobrovytsky, T., Ching, C., Choi, A., ... & Yoder, M. (2015, January). Impala: A Modern, Open-Source SQL Engine for Hadoop. In Cidr (Vol. 1, p. 9).

McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. Python for high performance and scientific computing, 14(9), 1-9.

Riley, C. (2008). Getting SAS® to Play Nice With Others: Connecting SAS® to Microsoft SQL Server Using an-ODBC Connection. In SAS Global Forum 2008 Paper 135-2008.

Russell, J. (2013). Cloudera Impala. " O'Reilly Media, Inc.".

SQL Tutorial. Retrieved 6 February 2022, from https://www.w3schools.com/sql/default.asp

¿Qué es Hadoop?. (2022). Retrieved 3 February 2022, from https://www.sas.com/es_co/insights/big-data/hadoop.html#hadoopworld