# Visualización de principales cuentas e indicadores financieros de la superintendencia de Guatemala, El Salvador, Panamá y Colombia usando web scraping.


Autor:

Julian David Valencia Diaz

Asesores:

Jesus Francisco Vargas Bonilla

Maria Victoria Cardenas Londoño

Universidad de Antioquia

Facultad de ingeniería

Ingeniería electrónica


Medellín, Antioquia

2023

| | |
|---|---|
| **Cita** | (Valencia Diaz, 2023) |
| **Referencia**<br><br>**Estilo APA 7 (2020)** | Valencia Diaz, J. D. (2023). Visualización de principales cuentas e indicadores financieros de la superintendencia de Guatemala, El Salvador, Panamá y Colombia usando web scraping [Semestre de industria]. Universidad de Antioquia, Seleccione ciudad UdeA. |



Centro de documentación ingeniería, CENDOI

**Repositorio Institucional:** http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

**Rector:** John Jairo Arboleda Céspedes.

**Decano/Director:** Julio César Saldarriaga.

**Jefe departamento:** Augusto Enrique Salazar Jiménez.

**Resumen**

La vicepresidencia financiera de Bancolombia enfrentaba desafíos en cuanto al acceso a los estados financieros de los bancos del sistema financiero colombiano y de los países en los que el Grupo Bancolombia operaba. Además, no contaba con una base de datos consolidada que integrara todas las entidades, cuentas y periodos de tiempo relevantes. Para abordar esta situación, se implementó un proceso de Extracción, Transformación y Carga (*ETL*) en *Python* para cada país.

Este proceso de *ETL* involucró la extracción de información de las superintendencias financieras correspondientes, haciendo uso de técnicas como el *web scraping*. A continuación, se utilizó librerías como *pandas* y *numpy* para la transformación y limpieza de datos, lo que resultó en la creación de bases de datos consolidadas para cada país.

Una vez completada la consolidación de los datos, se utilizó la herramienta *Power BI* para generar visualizaciones detalladas de las principales cuentas e indicadores utilizados en el análisis de competencia del banco. Esto permitió tomar decisiones oportunas en relación al comportamiento del mercado y proporcionó una visión clara de la situación financiera en cada país.

**Introducción**

El grupo Bancolombia, entidad financiera en Colombia, en orden con su ideología y misión, está en constante evolución y transformación digital. Para dicha evolución, Bancolombia dispone de diferentes equipos con diferentes enfoques, la presente propuesta se orienta en función de servir a la vicepresidencia de control financiero, donde una de sus funciones es analizar el desempeño financiero de la corporación, sus compañías y negocios, y entregar recomendaciones.

Es importante al analizar el desempeño financiero de la corporación revisar como ha sido el comportamiento respecto al mercado nacional, esta información se encuentra alojada en la página web de la superintendencia financiera de cada país la cual es encargada de promover la estabilidad del Sistema Financiero, la integridad y transparencia del mercado de valores y velar por la protección de los derechos de los consumidores financieros [7]. Se encuentra información acerca del balance (activos, pasivos y patrimonio), pérdidas y ganancias (PyG) y algunos indicadores, que permiten obtener información muy completa de todas las entidades financieras en un gran periodo de tiempo.

Desde la vicepresidencia de control financiero se han construido bases históricas de forma manual, lo cual es un proceso de mucho tiempo, posibles errores y no ha sido posible abarcar todas las entidades financieras por la gran cantidad de información que hay.

Se requiere automatizar el proceso de extracción, transformación y carga (ETL) de datos alojados en las superintendencias financieras de Guatemala, El Salvador, Panamá y Colombia. Estos países son de interés para el Grupo Bancolombia, ya que tiene presencia en ellos. El objetivo es obtener una base de datos histórica y generar visualizaciones de las principales cuentas e indicadores. Estos recursos serán de gran utilidad para la toma de decisiones en la vicepresidencia de control financiero.

## Objetivos

### Objetivo General:

- Realizar el proceso de ETL y visualización con la información de los estados financieros suministrada por la página web de las superintendencias de Guatemala, El Salvador, Panamá y Colombia, realizando *web scraping,* construyendo un *dashboard* para cada país de los principales indicadores y cuentas financieras, que serán de utilidad para la toma de decisiones de la vicepresidencia financiera.

### Objetivos específicos:

- Desarrollar un análisis de la estructura de la página web de cada superintendencia financiera para definir el método de *web scraping* más eficiente que se implementará.

- Almacenar en *onedrive* la información histórica de la superintendencia financiera de Guatemala, El Salvador, Panamá y Colombia, construyendo un *web scraping* que automatice el proceso de extracción de información para ser conservada en la nube.

- Crear base de datos histórica para cada país, transformando y concatenando todos los archivos históricos que se han obtenido para explorar y analizar los datos de manera rápida y eficiente.

- Construir visualizaciones en *Power BI* de las principales cuentas e indicadores de cada país, realizando un análisis exploratorio de datos y por medio de objetos visuales resumir y presentar información efectiva y eficiente para proporcionar una vista de ¿cómo va el banco? y facilitar la toma de decisiones.

**Marco Teórico**

- **ETL:**
  ETL es un tipo de integración de datos que hace referencia a los tres pasos (extraer, transformar, cargar) que se utilizan para mezclar datos de múltiples fuentes. Se utiliza a menudo para construir un almacén de datos. Durante este proceso, los datos se toman (extraen) de un sistema de origen, se convierten (transforman) en un formato que se puede almacenar y se almacenan (cargan) en un *data warehouse* u otro sistema. Extraer, cargar, transformar (ELT) es un enfoque alterno pero relacionado diseñado para canalizar el procesamiento a la base de datos para mejorar el desempeño.[6]

- **Web Scraping:**
  *Web Scraping* es un método automático para obtener grandes cantidades de datos de sitios web. La mayoría de estos datos son datos no estructurados en formato *HTML* que luego se convierten en datos estructurados en una hoja de cálculo o una base de datos para que puedan usarse en varias aplicaciones. [1]

- **Requests**: La librería *requests* nos permite enviar solicitudes *HTTP* con *Python* sin necesidad de tanta labor manual, haciendo que la integración con los servicios web sea mucho más fácil. No es necesario agregar manualmente consultas a las *URLs* o de convertir información a formularios para realizar una solicitud *POST*. Todo esto es logrado gracias a la buena integración de urllib3 en *Requests*. [8]

- **Selenium**: *Selenium* es una herramienta diseñada para ayudarle a ejecutar pruebas automatizadas en aplicaciones web. Está disponible en varios lenguajes de programación.
  Aunque no es su propósito principal, *Selenium* también se usa en *Python* para *web scraping*, porque puede acceder a contenido renderizado en *JavaScript* (lo que las herramientas de *scraping* normales como *BeautifulSoup* no pueden hacer).
  *Selenium* también es útil cuando necesita interactuar con la página de alguna manera antes de recopilar los datos, como hacer clic en botones o completar campos. [10]

- **Selenium Undetected Chromedriver:**
  *Selenium Undetected Chromedriver* es una versión optimizada del *ChromeDriver* estándar diseñado para eludir los mecanismos de detección de la mayoría de las soluciones anti-bot como *DataDome*, *Perimeterx* y *Cloudflare*.
  El *Selenium ChromeDriver* estándar filtra mucha información que los sistemas anti-bot pueden usar para determinar si se trata de un navegador/raspador automático o de un usuario real que visita el sitio web.
  *Selenium Undetected ChromeDriver* fortalece el *Selenium ChromeDriver* estándar parcheando la gran mayoría de las formas en que los sistemas anti-bot pueden usar para detectar su bot/*scraper* de *Selenium*. [9]

- **Pandas:**
  *Pandas* es una librería de *Python* especializada en el manejo y análisis de estructuras de datos.
  Las principales características de esta librería son:

  - Define nuevas estructuras de datos basadas en los *arrays* de la librería *NumPy* pero con nuevas funcionalidades.
  - Permite leer y escribir fácilmente ficheros en formato *CSV*, *Excel* y bases de datos *SQL*.
  - Permite acceder a los datos mediante índices o nombres para filas y columnas.
  - Ofrece métodos para reordenar, dividir y combinar conjuntos de datos.
  - Permite trabajar con series temporales.
  - Realiza todas estas operaciones de manera muy eficiente. [11]

- **Numpy:**
  *NumPy* es una librería de *Python* especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos.
  Incorpora una nueva clase de objetos llamados *arrays* que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación.
  La ventaja de *Numpy* frente a las listas predefinidas en *Python* es que el procesamiento de los *arrays* se realiza mucho más rápido (hasta 50 veces más)

que las listas, lo cual la hace ideal para el procesamiento de vectores y matrices de grandes dimensiones. [12]

- **XML:**
  El lenguaje de marcado extensible (*XML*) permite definir y almacenar datos de forma compartible. *XML* admite el intercambio de información entre sistemas de computación, como sitios web, bases de datos y aplicaciones de terceros. Las reglas predefinidas facilitan la transmisión de datos como archivos *XML* a través de cualquier red, ya que el destinatario puede usar esas reglas para leer los datos de forma precisa y eficiente. [3]

- **XPATH:**
  El lenguaje de vía de acceso *XML* (*XPath*) se utiliza para identificar de forma exclusiva o resolver partes de un documento *XML*. Puede utilizarse una expresión *XPath* para buscar en un documento *XML* y extraer información de sus nodos, que son cualquier parte del documento como, por ejemplo, un elemento o atributo. Se puede utilizar *XPath* sólo o con *XSLT*.[4]

- **Matriz de datos:**
  La matriz de datos es la herramienta principal que permite el registro de los valores de las diferentes variables con un ordenamiento de la información fácilmente visible, a partir del cual ejecutar los diferentes análisis. Ya sea con el fin de realizar operaciones estadísticas o para someterlos al tratamiento necesario, en función de nuestros objetivos, la matriz permite explotar al máximo el uso de los datos. [5]

- **Power BI:**
  *Power BI* es un conjunto de herramientas que pone el conocimiento al alcance de todos y nos brinda acceder a nuestros datos de forma segura y rápida, generando grandes beneficios para nosotros y para nuestra empresa. Es un sistema predictivo, inteligente y de gran apoyo, capaz de traducir los datos (simples o complejos) en gráficas, paneles o informes por sus cualidades como la capacidad gráfica de presentación de la información, o la integración de *Power Query*: el motor de extracción, transformación y carga (*ETL*) incluido en *Excel*. [2]

**Metodología**

La metodología a implementar en los cuatros países fue la siguiente:


Figura 1. Metodología

1. **Exploración página Web:** Como primer paso, resultó crucial familiarizarse con la estructura de las páginas web relevantes, evaluar su seguridad en términos de acceso a la información y determinar los datos de interés que debían descargarse. Además, fue necesario establecer el período de tiempo a descargar para la creación de las bases de datos. Durante esta etapa, se identificaron los siguientes aspectos:

| **País** | **Seguridad** | **Período de información a descargar** |
|---|---|---|
| Colombia | Fácil de acceder | 1995 - período actual |
| Panamá | Fácil de acceder | 2008 - período actual |
| El Salvador | Fácil de acceder | 2008 – período actual |
| Guatemala | Difícil de acceder, contiene captcha al ingresar | 2008 – período actual |

2. **Web Scraping:** Una vez que se comprendió la estructura de las páginas web, resultó crucial determinar qué librerías de *Python* utilizar para extraer toda esa información y automatizar el proceso. Para este propósito, se emplearon dos bibliotecas populares de *web scraping*: *Requests* y *Selenium*. Estas

herramientas permitieron acceder a los datos necesarios y facilitaron la automatización de la extracción de información de las páginas web.

Se utilizó la librería *Requests* para acceder a la información de las páginas web mediante peticiones *HTTP*. Esta biblioteca permitió establecer comunicación rápida con los servidores de las páginas web y obtener cualquier información necesaria. Específicamente, se optó por utilizar *Requests* en los casos de Colombia, Panamá y El Salvador, dado que las páginas web de estos países no tenían ningún tipo de seguridad adicional. La comunicación con los servidores fue inmediata, lo que permitió descargar la información de forma rápida y eficiente.

Por otro lado, utilizamos la librería *Selenium* para simular el comportamiento de una persona en la página web, especialmente para realizar *web scraping* en Guatemala.

Implementamos dos enfoques para superar el desafío de los captchas en la página web:

1- En primer lugar, al encontrarnos con el captcha al acceder a la página web, optamos por actualizar repetidamente la página hasta lograr ingresar. Esta solución se basó en la suposición de que algunos captchas se basan en un puntaje asignado a una persona al interactuar con varias páginas web. Al actualizar la página múltiples veces, el puntaje aumentaba gradualmente, lo que nos permitía finalmente ingresar a la página.

2- Dado que la primera opción resultó ineficiente, buscamos otras bibliotecas que interactuaran con *Selenium*, como *Undetected ChromeDriver*. Esta librería fortaleció nuestra capacidad para no ser detectados como un *scraper*, permitiéndonos acceder a la página web de manera inmediata y obtener la información de forma rápida.

Estos enfoques nos permitieron superar los desafíos de los captchas y acceder a la información requerida en la superintendencia financiera de Guatemala.

3. **Transformación y carga de datos:**
   Para optimizar el manejo de la información descargada y almacenada en *OneDrive*, fue necesario emplear librerías como *Pandas* y *NumPy* para transformar y unificar los datos en un formato matricial.
   Dado que se disponía de información de múltiples períodos de tiempo, se identificaron diversas variaciones en la presentación de los datos de un año a otro. Específicamente, se encontraron 6 formatos distintos para Colombia, 4 para Panamá, 5 para El Salvador y 2 para Guatemala.

4. **Detectar principales cuentas e indicadores:** Después de obtener las bases de datos de todos los países, se llevó a cabo la selección de las cuentas de balance, PyG (Pérdidas y Ganancias) y los indicadores pertinentes. Para lograr una estandarización completa, se tuvieron en cuenta los informes de competencia de cada país. Este proceso permitió uniformizar todas las cuentas necesarias para la construcción de los tableros en Power BI.
   Todos los tableros contarán con la siguiente información:

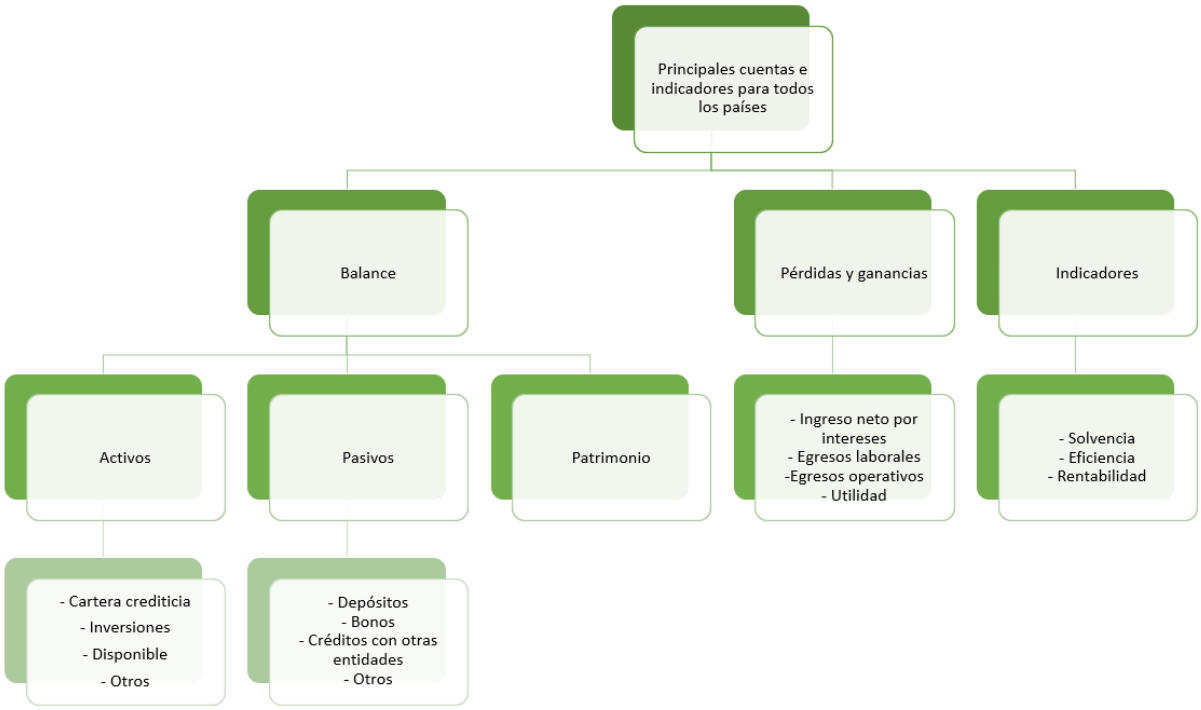


Figura 2. Principales cuentas e indicadores

5. **Visualización**: Después de eso, se procedió a desarrollar un esquema de la estructura del Power BI para cada país. Se brindó la libertad de presentar varias propuestas que incluyeran la siguiente información: un resumen del sistema bancario del país, una visión general de las cuentas a observar y un desglose completo de todas las cuentas. Finalmente, se presentaron varias opciones y se seleccionó el siguiente formato que consta de tres estructuras:

   - General
   - Resumen
   - Detalle

   La vista general se construyó pensando en ofrecer una estructura que permitiera visualizar información clave de manera rápida y eficiente. Por lo tanto, se mostraron los siguientes elementos:

   - Una sección mostró la cantidad de bancos en el país, diferenciando entre bancos nacionales y extranjeros.
   - Se clasificaron los bancos en grandes, medianos y pequeños mediante una segmentación, facilitando la selección de los bancos de mayor interés.
   - Se presentó una tabla que mostró la participación de cada banco en términos de activos con respecto al total del sistema financiero.
   - Se utilizaron botones que permitieron interactuar con las otras vistas.

   La estructura proporcionó una visión general completa y organizada, permitiéndote acceder rápidamente a la información relevante sobre los bancos, su participación en el sistema y los aspectos financieros.

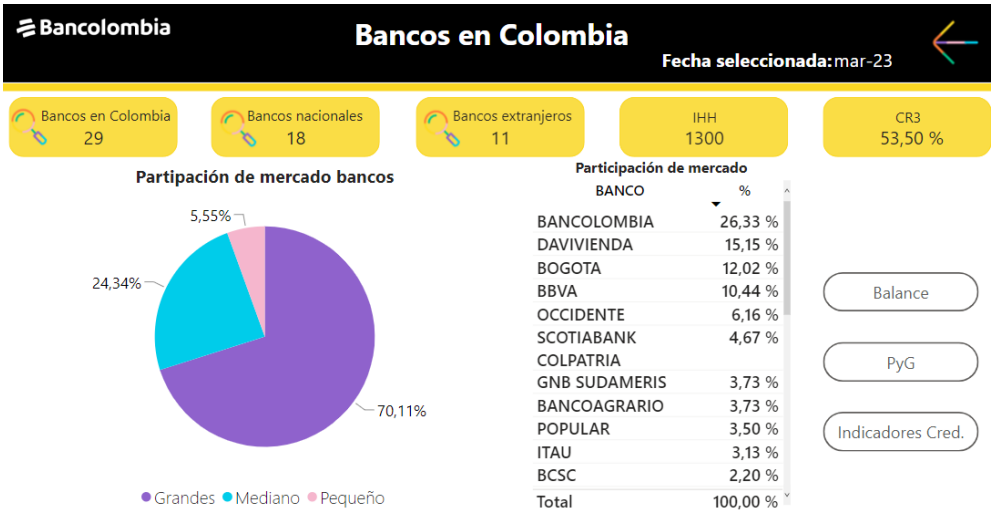


Figura 3. Vista general

En la vista resumen, se presentó una tabla detallada que mostraba la información relevante de las cuentas e indicadores a revisar. Se implementaron filtros para seleccionar el año, mes y banco específico para visualizar el Balance, el Estado de Resultados o los indicadores.

La tabla proporcionó datos como saldo actual, variación mensual y variación anual para todas las cuentas. Además, se incluyó una gráfica que representaba los últimos 12 meses, según la selección del mes, permitiendo apreciar las variaciones mensuales y anuales.

Adicionalmente, se añadieron botones que permitieron acceder al detalle de activos y pasivos, brindando una navegación más detallada y completa en la información financiera.


Figura 4. Vista resumen

En la vista detalle, se priorizó mostrar la máxima información posible para cada cuenta. Por lo tanto, se construyó una hoja por cada cuenta con información que permitió visualizar la participación de cada banco, la variación anual y la composición de dicha cuenta en términos de saldo y porcentaje. Esta visualización detallada brindó una comprensión más completa y precisa de cada cuenta.

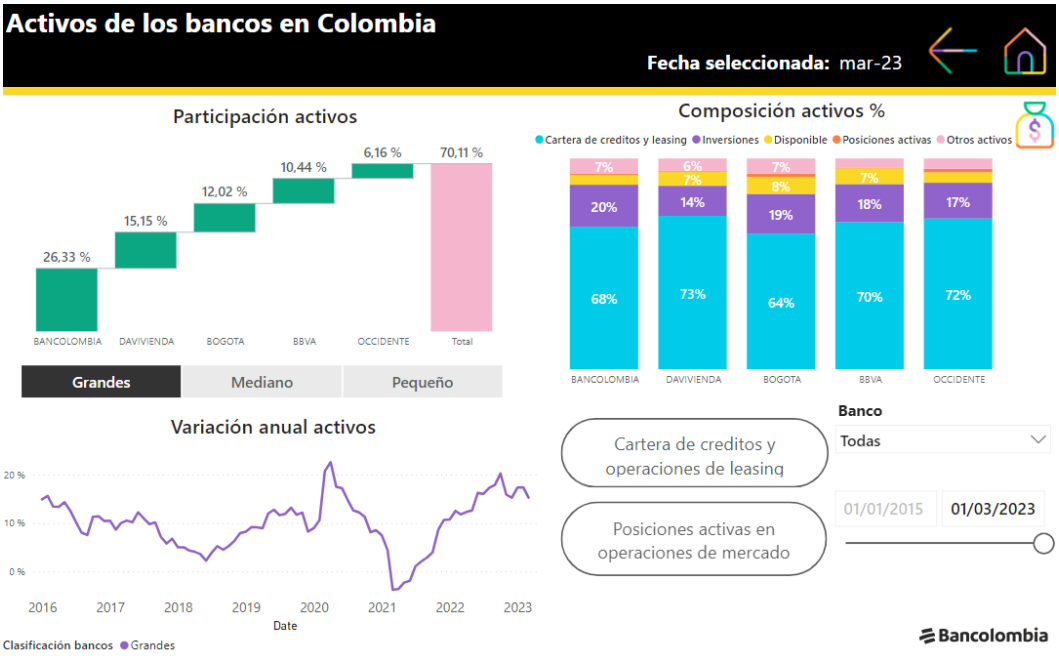


Figura 5. Vista detalle

6. **Validación de los resultados:** Se solicitó a los equipos de Guatemala, Panamá, El Salvador y Colombia que proporcionaran informes de competencia para comparar datos y verificar la funcionalidad adecuada de los programas desarrollados con Python. Posteriormente, se llevaron a cabo reuniones con dichos equipos, durante las cuales se compartió el código y los tableros generados, así como una explicación sobre su funcionamiento, posibles errores y cómo corregirlos.

   Fue gratificante observar cómo los equipos de diferentes regiones nos informaron que les tomaba aproximadamente tres semanas generar un informe similar, pero limitado a los bancos principales. En contraste, les proporcionamos un código de Python y un tablero de Power BI que les ahorra todo ese trabajo en tan solo 15 minutos, ofreciéndoles una cantidad mucho mayor de información disponible.

**Conclusiones**

- El análisis de la estructura de las páginas web nos permitió aprovechar las ventajas y desventajas de las bibliotecas de *web scraping*, como *Requests* y *Selenium*. Se recomendó el uso de Selenium para aquellas páginas que cuentan con algún tipo de seguridad, mientras que para aquellas sin estas medidas se recomendó el uso de *Requests* debido a su velocidad. De esta manera, se garantizó un funcionamiento eficiente en los *scrapers*, adaptándolos a las necesidades específicas de cada página.

- Almacenar la información en servicios en la nube como *OneDrive* o en un repositorio similar brinda una capa adicional de seguridad. Esto se debe a que, en caso de que ocurra un error y se elimine accidentalmente algún dato importante, es posible acceder a esa información gracias a las funcionalidades de versionamiento y copias de seguridad que ofrecen estas aplicaciones. Estas características permiten restaurar y recuperar datos anteriores, brindando tranquilidad y garantizando la disponibilidad de la información en caso de incidentes inesperados.

- La automatización implementada para transformar los diversos formatos históricos disponibles en cada país resultó en un ahorro significativo de tiempo. Este enfoque evitó que una persona tuviera que realizar la

laboriosa tarea de combinar información de múltiples archivos de Excel manualmente, lo que no solo consumía tiempo, sino que también aumentaba el riesgo de cometer errores al copiar y pegar los datos. La automatización permitió procesar los datos de manera eficiente y precisa, liberando recursos humanos para tareas más estratégicas y reduciendo la posibilidad de errores inherentes a las tareas manuales.

- La construcción de visualizaciones para todos los países y la recopilación de información de todos los bancos nos brindó una visión integral del comportamiento del mercado financiero. Esto nos permitió comprender cómo se estructuran los balances, los estados de resultados y los indicadores clave. Esta información resultó fundamental para tomar decisiones oportunas y fundamentadas en el contexto financiero. Las visualizaciones proporcionaron una representación clara y accesible de los datos, lo que facilitó el análisis y la identificación de tendencias y patrones relevantes. Gracias a esto, se logró obtener una visión estratégica y precisa del mercado, proporcionando una base sólida para la toma de decisiones.

**Referencias Bibliográficas**


[1] ¿Qué es el web scraping? (s.f.). Obtenido de ciberseguridad.com:https://ciberseguridad.com/guias/recursos/web-scraping/

[2] ¿Qué es Power BI? (s.f.). Obtenido de www2.deloitte.com: https://www2.deloitte.com/es/es/pages/technology/articles/que-es-power- bi.html

[3] ¿Qué es XML? (s.f.). Obtenido de aws.amazon.com: https://aws.amazon.com/es/whatis/xml/#:~:text=Un%20archivo%20de%20lenguaje%20de,a%20otros%20archivos%20de%20texto.

[4] Descripción general de XPath. (s.f.). Obtenido de www.ibm.com: https://www.ibm.com/docs/es/baw/19.x?topic=expressions-xpath-overview

[5] La matriz de análisis de datos, un aliado para empresas data-driven. (s.f.).

Obtenido de powerdata.es: https://blog.powerdata.es/el-valor-de-la-gestion- de-datos/la-matriz-de-analisis-de-datos-un-aliado-para-empresas-data-driven

[6] SAS Institute Inc. (s.f.). ETL. Obtenido de www.sas.com: https://www.sas.com/es_co/insights/data-management/what-is-etl.html

[7] Nuestra Entidad. (s.f.). Obtenido de www.superfinanciera.gov.co: https://www.superfinanciera.gov.co/inicio/nuestra-entidad-20483

[8] SOLICITUDES HTTP EN PYTHON CON REQUESTS. (s.f.). Obtenido de unipython.com: https://unipython.com/solicitudes-http-en-python-con-requests/

[9] Selenium Undetected Chromedriver (s.f.). Obtenido de scraperops.io.: Bypass Anti-Bots With Ease: https://scrapeops.io/selenium-web-scraping playbook/python-selenium-undetected-chromedriver/

[10] Selenium (1 de Julio de 2021). Freecodecamp.org. Obtenido de https://www.freecodecamp.org/espanol/news/como-codificar-un-scraping-bot-con-selenium-y-python/#:~:text=Selenium%20es%20una%20herramienta%20dise%C3%B1ada,en%20varios%20lenguajes%20de%20programaci%C3%B3n.

[11] La librería Pandas , Alberca, A. S. (14 de Junio de 2022). Obtenido de aprendeconalf.es: https://aprendeconalf.es/docencia/python/manual/pandas/

[12] La librería Numpy , Alberca, A. S. (12 de mayo de 2022). Obtenido de aprendeconalf.es: https://aprendeconalf.es/docencia/python/manual/numpy/