



Universidad Internacional de La Rioja  
Facultad de Empresa, Comunicación y Marketing

Máster Universitario en Inteligencia de Negocio

**Implementación de un RPA para la  
Extracción y Análisis en Tiempo Real de  
Datos de Ecopetrol desde la Bolsa de  
Valores de Colombia: Integración con SQL  
y Visualización Web en Localhost.**

|  |                                     |
|--|-------------------------------------|
| Trabajo fin de estudio presentado por: | Michel Enrique Abello Betancourt    |
| Tipo de trabajo:                       | Proyecto de Inteligencia de Negocio |
| Modalidad:                             | Individual                          |
| Director/a:                            | Antonio Pita Lozano                 |
| Fecha:                                 | 24/11/2024                          |

## Resumen

Este proyecto de grado se enfoca en la implementación de un sistema automatizado (RPA) para la extracción y análisis en tiempo real de datos de la empresa Ecopetrol desde la Bolsa de Valores de Colombia. El RPA desarrollado recolecta información crítica de la página web, la cual es cargada automáticamente en una base de datos SQL. Una vez almacenados, los datos son analizados en tiempo real, y los resultados más relevantes son visualizados en una API web alojada en localhost. Este sistema no solo automatiza el proceso de obtención de datos, sino que también permite una monitorización continua y en tiempo real de las variables más importantes, facilitando la toma de decisiones basada en información actualizada. La integración entre el RPA, SQL y la API web garantiza un flujo de trabajo eficiente y efectivo, optimizando la gestión de la información financiera de Ecopetrol.

**Palabras clave:** API (interfaz de programación de aplicaciones), RPA (Automatización robótica de procesos), Integración de sistemas, Análisis en tiempo real y Automatización.

## Abstract

This thesis project focuses on the implementation of an automated system (RPA) for the extraction and real-time analysis of data from Ecopetrol, sourced from the Colombian Stock Exchange. The developed RPA collects critical information from the website, which is automatically loaded into an SQL database. Once stored, the data is analyzed in real time, and the most relevant results are visualized on a web API hosted on localhost. This system not only automates the data acquisition process but also enables continuous and real-time monitoring of the most important variables, facilitating decision-making based on up-to-date information. The integration between the RPA, SQL, and the web API ensures an efficient and effective workflow, optimizing the management of Ecopetrol's financial information.

**Keywords:** API (Application Programming Interface), RPA (Robotic Process Automation), System Integration, Real-time Analysis, and Automation.

## Índice de contenidos

|        |   |    |
|--------|---|----|
| 1.     | Introducción .....  | 9  |
| 1.1.   | Planteamiento general: descripción y justificación del proyecto ..... | 10 |
| 1.2.   | Objetivos del TFE .....   | 12 |
| 1.2.1. | Objetivo General.....   | 12 |
| 1.3.   | Elementos innovadores del proyecto.....                               | 13 |
| 2.     | Alcance y planificación .....   | 14 |
| 2.1.   | Fase de descubrimiento: evaluación del entorno actual .....           | 14 |
| 2.1.1. | Información deseada .....   | 14 |
| 2.1.2. | Información actual: deficiencias y soluciones alternativas.....       | 16 |
| 2.1.3. | Habilidades analíticas actuales.....                                  | 18 |
| 2.2.   | Fase de análisis: identificación de <i>gaps</i> .....                 | 20 |
| 2.2.1. | Capacidad de los informes actuales .....                              | 20 |
| 2.2.2. | Proveedores de tecnología necesarias.....                             | 21 |
| 2.2.3. | Diferencia entre los informes actuales y la información deseada ..... | 23 |
| 2.2.4. | Cronología, costes y recursos humanos implicados.....                 | 24 |
| 2.3.   | Fase de recomendaciones: alcance, prioridades y presupuesto .....     | 26 |
| 2.3.1. | Promoción del proyecto en la organización.....                        | 26 |
| 2.3.2. | Demostraciones.....   | 26 |
| 2.3.3. | Presupuesto .....   | 27 |
| 3.     | Análisis y definición .....   | 28 |
| 3.1.   | Análisis de los datos a utilizar.....                                 | 28 |
| 3.2.   | Análisis histórico y/o limpieza de datos .....                        | 29 |
| 3.3.   | Modelo Propuesto.....   | 32 |
| 3.3.1. | Captación de datos históricos .....                                   | 32 |

|   |                                      |
|---|--------------------------------------|
| Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost |                                      |
| 3.3.2.    Consulta SQL para conectar los datos con la API .....   | 34                                   |
| 3.3.3.    Estructuración de la API y los análisis.....  | 35                                   |
| 3.3.4.    Visualización de la API en el local Host.....   | 40                                   |
| 4.    Construcción, prueba, implementación y despliegue .....   | 47                                   |
| 4.1.    Construcción .....  | 47                                   |
| 4.2.    Prueba.....   | 48                                   |
| 4.3.    Implementación.....   | 49                                   |
| 4.4.    Despliegue .....  | 50                                   |
| 4.5.    Medidas de bondad del modelado.....   | 50                                   |
| 4.6.    Ejemplo de aplicación.....  | 51                                   |
| 5.    Cronograma del proyecto .....   | 53                                   |
| 5.1.    Swimlane de Inteligencia de negocio .....   | 53                                   |
| 5.2.    Swimlane de datos y bases de datos.....   | 53                                   |
| 5.3.    Swimlane de la integración de datos .....   | 54                                   |
| 5.4.    Swimlane de la infraestructura .....  | 54                                   |
| 5.5.    Swimlane de la gestión del proyecto.....  | 54                                   |
| 6.    Conclusiones.....   | 55                                   |
| 7.    Limitaciones y prospectiva .....  | <b>¡Error! Marcador no definido.</b> |
| 7.1.    Limitaciones técnicas.....  | <b>¡Error! Marcador no definido.</b> |
| 7.2.    Limitaciones analíticas.....  | 56                                   |
| 7.3.    Limitaciones operativas.....  | 57                                   |
| 7.4.    Prospectiva y oportunidades de mejora .....   | 57                                   |
| 7.4.1.    Mejoras técnicas propuestas.....  | 57                                   |
| 7.4.2.    Ampliaciones analíticas .....   | 57                                   |
| 7.4.3.    Proyección futura .....   | 58                                   |



## Índice de figuras

|   |    |
|---|----|
| Figura 1. Diagrama de Gantt de cronograma de actividades .....          | 25 |
| Figura 2. Código cálculo de fecha .....                                 | 30 |
| Figura 3. Código ruta de carpeta .....                                  | 32 |
| Figura 4. Código importación de documento csv.....                      | 32 |
| Figura 5. Código validación de proceso entra en error .....             | 32 |
| Figura 6. Código de lectura y estructuración de csv.....                | 33 |
| Figura 7. Código para cadena de conexión SQL .....                      | 33 |
| Figura 8. Código para cadena de conexión SQL .....                      | 33 |
| Figura 9. Código que elimina csv procesado y almacenado.....            | 34 |
| Figura 10. Código que muestra la función definida para server SQL ..... | 34 |
| Figura 11. Código parámetros de conexión SQL .....                      | 34 |
| Figura 12. Código parámetros de conexión SQL .....                      | 35 |
| Figura 13. Código para selección de tabla.....                          | 35 |
| Figura 14. Código creación de framework .....                           | 35 |
| Figura 15. Código para definir la raíz de la ruta .....                 | 36 |
| Figura 16. Código llamado de conexión SQL .....                         | 36 |
| Figura 17. Código llamado de columnas.....                              | 36 |
| Figura 18. Código para crear grafico de barras .....                    | 37 |
| Figura 19. Código para crear mapa de correlación .....                  | 37 |
| Figura 20. Código para crear histograma .....                           | 38 |
| Figura 21. Código para crear regresión .....                            | 38 |
| Figura 22. Código para crear proyección.....                            | 38 |
| Figura 23. Código para convertir los gráficos en imágenes.....          | 39 |
| Figura 24. Código para unificar el llamado de las imágenes .....        | 39 |

|  |    |
|--|----|
| Figura 25. Ejecución del local Host .....                          | 40 |
| Figura 26. Banner inicial para el localhost.....                   | 40 |
| Figura 27. Gráfico de barras según precio de cierre .....          | 41 |
| Figura 28. Diagrama de correlación de las variables.....           | 42 |
| Figura 29. Gráficos de distribución según variables del crudo..... | 43 |
| Figura 30. Histograma según regresión lineal .....                 | 45 |
| Figura 31. Gráfico de variación porcentual y absoluta.....         | 45 |
| Figura 32. Proyección del precio de cierre .....                   | 47 |

## 1. Introducción

En un entorno empresarial cada vez más competitivo y globalizado, la capacidad de tomar decisiones informadas y oportunas se ha convertido en un factor crítico para el éxito de cualquier organización. Las empresas que operan en mercados bursátiles, como Ecopetrol, enfrentan el desafío constante de analizar grandes volúmenes de datos financieros en tiempo real para identificar oportunidades de inversión, mitigar riesgos y maximizar rendimientos. La necesidad de sistemas que puedan manejar y procesar estos datos de manera eficiente es, por lo tanto, imperativa.

En este contexto, la automatización de procesos robóticos (RPA, por sus siglas en inglés) ha emergido como una tecnología clave que permite a las organizaciones automatizar tareas repetitivas y propensas a errores, liberando recursos humanos para actividades más estratégicas. Los sistemas RPA se han integrado en diversas industrias para mejorar la precisión, velocidad y eficiencia de las operaciones, especialmente en áreas como la recolección y análisis de datos. Este proyecto de grado se inscribe dentro de esta tendencia, abordando la creación e implementación de un RPA diseñado específicamente para extraer, procesar y analizar datos financieros en tiempo real desde la página web de la Bolsa de Valores de Colombia, con un enfoque en la empresa Ecopetrol.

Ecopetrol, como una de las compañías más importantes de Colombia y un actor relevante en la industria energética global, requiere un monitoreo continuo y exhaustivo de su desempeño financiero en el mercado de valores. La información publicada en la Bolsa de Valores de Colombia es esencial para analizar su posición financiera, evaluar su rendimiento y tomar decisiones estratégicas que afecten tanto a la empresa como a sus accionistas. Sin embargo, la cantidad y la velocidad con la que estos datos se generan pueden ser abrumadoras, haciendo necesario un sistema que permita no solo la recolección automática de la información, sino también su análisis en tiempo real.

El RPA desarrollado en este trabajo de grado tiene como objetivo principal automatizar la extracción de datos críticos de Ecopetrol desde la Bolsa de Valores de Colombia y cargar esta información en una base de datos SQL. Posteriormente, se realiza un análisis en tiempo real de las variables más importantes, cuyos resultados se visualizan a través de una API web en un entorno localhost. Este enfoque no solo agiliza el proceso de obtención y procesamiento

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost de datos, sino que también permite a los analistas financieros y tomadores de decisiones acceder a información actualizada y relevante de manera inmediata.

El presente documento explora en detalle el desarrollo e implementación del RPA, así como su integración con tecnologías de bases de datos y visualización de datos. Se abordan las metodologías empleadas, los desafíos encontrados durante el desarrollo y las soluciones implementadas. Además, se discuten los beneficios obtenidos a partir de la automatización y análisis en tiempo real, y cómo estos pueden ser extrapolados a otras áreas de la organización o incluso a otras empresas dentro del sector.

Finalmente, este trabajo también ofrece una reflexión sobre el impacto de la automatización en el análisis de datos financieros y su potencial para transformar la manera en que las empresas gestionan la información crítica para la toma de decisiones. La disposición del trabajo va a evaluar los resultados obtenidos y propuestas para futuras mejoras y expansiones del sistema, considerando las tendencias emergentes en el campo de la automatización y el análisis de datos.

## 1.1. Planteamiento general: descripción y justificación del proyecto

El entorno financiero actual está marcado por un dinamismo sin precedentes, donde los mercados reaccionan en cuestión de segundos a eventos globales, cambios regulatorios y fluctuaciones en la oferta y demanda de productos. En este contexto, las empresas que cotizan en bolsa, como Ecopetrol, se ven obligadas a realizar un seguimiento constante y detallado de su desempeño en los mercados financieros para mantener su competitividad y maximizar el valor para sus accionistas.

Ecopetrol, siendo la empresa de hidrocarburos más grande de Colombia y una de las más importantes de América Latina, tiene una presencia significativa en la Bolsa de Valores de Colombia (BVC). La información que se publica en la BVC sobre Ecopetrol, como los precios de las acciones, volúmenes de transacciones, y variaciones porcentuales, es fundamental para tomar decisiones estratégicas. No obstante, la cantidad de datos generados y la velocidad con la que estos se actualizan presentan un desafío considerable en términos de recolección, procesamiento y análisis oportuno.

En este sentido, el presente proyecto tiene como objetivo desarrollar un sistema automatizado, basado en la tecnología de Automatización Robótica de Procesos (RPA), que permita extraer datos financieros de Ecopetrol desde la página web de la Bolsa de Valores de Colombia, cargarlos en una base de datos SQL, y realizar un análisis en tiempo real de las variables más relevantes. Los resultados de este análisis serán visualizados a través de una API web alojada en un entorno localhost, permitiendo a los usuarios acceder a información actualizada de manera inmediata.

La justificación del proyecto radica en la necesidad imperiosa de contar con un sistema que pueda manejar grandes volúmenes de datos de manera eficiente y rápida, optimizando la toma de decisiones basada en información financiera actualizada. La automatización del proceso no solo elimina la posibilidad de errores humanos en la recolección de datos, sino que también acelera significativamente el tiempo de respuesta, permitiendo a los analistas y ejecutivos de Ecopetrol reaccionar de manera más rápida y precisa a las condiciones cambiantes del mercado.

Además, este proyecto tiene un valor agregado significativo al integrar tecnologías de bases de datos y visualización de datos en una solución unificada, que no solo almacena y procesa la información, sino que también la presenta de manera intuitiva y accesible. Esta integración es crucial en un entorno donde la velocidad de acceso a la información y la capacidad para interpretarla rápidamente pueden marcar la diferencia entre el éxito y el fracaso en la toma de decisiones estratégicas.

El desarrollo de este sistema automatizado no solo aborda una necesidad crítica de Ecopetrol, sino que también sirve como un ejemplo de cómo la tecnología RPA puede ser aplicada eficazmente para mejorar la eficiencia operativa y la capacidad analítica en el sector financiero. Este proyecto justifica su importancia no solo por los beneficios directos que aporta a Ecopetrol, sino también por su potencial para ser adaptado y aplicado en otras áreas del negocio o en diferentes industrias que enfrentan desafíos similares.

## 1.2. Objetivos del TFE

### 1.2.1. Objetivo General

Desarrollar e implementar un sistema automatizado que extraiga, procese y analice en tiempo real datos financieros de Ecopetrol desde la Bolsa de Valores de Colombia, integrando la información en una base de datos SQL y visualizándola a través de una API web en localhost.

#### 1.2.1.1. Objetivos específicos

##### **Desarrollo del RPA para la extracción de datos:**

- Crear un RPA que sea capaz de navegar por la página web de la Bolsa de Valores de Colombia, identificar la información relevante de Ecopetrol y extraerla de manera automática.
- Asegurar que el RPA sea robusto y capaz de manejar diferentes escenarios, como cambios en el formato de la página web o interrupciones en la conexión.

##### **Integración del sistema con una base de datos SQL:**

- Diseñar e implementar un esquema de base de datos en SQL que permita almacenar de manera eficiente la información extraída por el RPA.
- Establecer un flujo de trabajo automatizado para la inserción de datos en la base de datos, asegurando la integridad y consistencia de la información.

##### **Implementación del análisis en tiempo real:**

- Desarrollar algoritmos y herramientas que permitan realizar un análisis en tiempo real de las variables más importantes extraídas de la información financiera de Ecopetrol.
- Asegurar que el análisis sea rápido, preciso y que pueda actualizarse de manera continua a medida que se recibe nueva información.

##### **Visualización de los resultados a través de una API web:**

- Diseñar e implementar una API web que permita a los usuarios acceder a los resultados del análisis en tiempo real desde un entorno localhost.
- Asegurar que la interfaz de la API sea intuitiva, fácil de usar y que proporcione una visualización clara y comprensible de la información relevante.

##### **Validación y optimización del sistema:**

- Realizar pruebas exhaustivas para validar el funcionamiento del sistema en su conjunto, asegurando que cumple con los requisitos establecidos y que opera de manera eficiente.
- Optimizar el sistema para mejorar su rendimiento, reducir el tiempo de procesamiento y asegurar que pueda escalarse si es necesario.

### 1.3. Elementos innovadores del proyecto

Este proyecto de grado introduce varios elementos innovadores que lo distinguen en el campo del manejo de datos del sector económico. En primer lugar, la aplicación de Automatización Robótica de Procesos (RPA) para la extracción de datos financieros de sitios web bursátiles es un enfoque avanzado y específico. Aunque la automatización en procesos empresariales es común, su uso para la extracción de datos bursátiles y el análisis en tiempo real es relativamente nuevo. Según (Lee, Huang, & Zeng, 2020), el uso de RPA en el sector financiero puede reducir significativamente el tiempo y los errores asociados con la recopilación manual de datos, mejorando la precisión y la eficiencia operativa. Además, la integración de la automatización de la extracción de datos con un análisis en tiempo real representa una innovación clave. Esta combinación permite a los analistas y ejecutivos tomar decisiones basadas en información actualizada al instante, lo cual es crucial en un entorno de mercado altamente dinámico. (Roh, Choi, & Kim, 2019) destacan que la integración de análisis en tiempo real con sistemas de bases de datos mejora significativamente la toma de decisiones y la gestión de riesgos en el sector financiero.

Otro aspecto innovador es la visualización de los resultados a través de una API web. Este enfoque permite a los usuarios acceder a los datos de manera eficiente desde un entorno localhost, proporcionando una interfaz amigable y accesible. Según (Khan, Shah, & Lee, 2021) las API web son esenciales para la integración y visualización de datos en tiempo real, facilitando el acceso a la información y permitiendo la creación de dashboards interactivos y personalizables. Finalmente, la automatización también aborda el problema de los errores humanos asociados con la recolección manual de datos. (Mitchell , 2021) afirma que la automatización robótica puede reducir significativamente los errores en la entrada de datos y mejorar la consistencia y precisión de la información.

## 2. Alcance y planificación

### 2.1. Fase de descubrimiento: evaluación del entorno actual

#### 2.1.1. Información deseada

La identificación de la información deseada es fundamental para garantizar que el sistema de automatización y análisis cumpla con los objetivos del proyecto. En el contexto de este proyecto, la información deseada se refiere a los datos financieros específicos que deben ser extraídos, procesados y analizados para ofrecer una visión clara del desempeño de Ecopetrol en la Bolsa de Valores de Colombia. A continuación, se detallan los elementos clave de la información deseada:

#### Precio de Cierre

- **Descripción:** El precio de cierre representa el valor al que la acción de Ecopetrol finaliza su negociación al final de la jornada bursátil. Este dato es crucial para evaluar el desempeño diario de la acción.
- **Uso:** Se utiliza para calcular variaciones diarias y evaluar la tendencia a corto plazo en el valor de las acciones.

#### Precio Máximo

- **Descripción:** El precio máximo es el valor más alto alcanzado por la acción durante el día de negociación. Proporciona una indicación del punto de mayor demanda o interés en el activo.
- **Uso:** Ayuda a identificar el rango de precios y a evaluar el nivel de volatilidad durante el día.

#### Precio Promedio Ponderado

- **Descripción:** Este precio se calcula ponderando los precios transaccionados por el volumen de acciones negociadas. Ofrece una medida ajustada del precio medio durante la jornada.
- **Uso:** Se utiliza para obtener una visión más precisa del precio promedio en comparación con el precio de cierre, especialmente en mercados con alta volatilidad.

#### Precio Mínimo

- **Descripción:** El precio mínimo es el valor más bajo alcanzado por la acción durante el día de negociación. Indica el punto de menor demanda o interés.
- **Uso:** Es útil para analizar el rango de precios y para evaluar el comportamiento del mercado en momentos de baja demanda.

### Variación Absoluta

- **Descripción:** La variación absoluta es la diferencia entre el precio de cierre de hoy y el precio de cierre del día anterior. Mide el cambio en términos absolutos en el valor de la acción.
- **Uso:** Permite observar el cambio diario en el valor y detectar movimientos significativos en el precio.

### Variación Porcentual

- **Descripción:** La variación porcentual es el cambio en el precio de cierre expresado como un porcentaje del precio de cierre del día anterior. Proporciona una medida relativa del cambio en el valor de la acción.
- **Uso:** Facilita la comparación de cambios en el valor de la acción en términos porcentuales, lo que ayuda a evaluar la magnitud de las fluctuaciones.

### Cantidad

- **Descripción:** La cantidad se refiere al número total de acciones de Ecopetrol transaccionadas durante la jornada. Este dato indica la actividad de negociación y la liquidez del activo.
- **Uso:** Ayuda a medir el interés en la acción y la capacidad de mercado para manejar grandes volúmenes de transacciones.

### Volumen

- **Descripción:** El volumen es el valor total de las transacciones realizadas durante la jornada, calculado multiplicando el precio de cierre por la cantidad de acciones negociadas. Proporciona una medida del valor total transaccionado.
- **Uso:** Es una métrica clave para evaluar la liquidez y el impacto del mercado en el precio de las acciones.

### 2.1.2. Información actual: deficiencias y soluciones alternativas.

El análisis de la información actual permite identificar deficiencias en los procesos y sistemas existentes relacionados con la recolección y análisis de datos financieros de Ecopetrol. Estas deficiencias pueden afectar la precisión, la actualidad y la utilidad de los datos. A continuación, se detallan las principales deficiencias y sus implicaciones:

#### Recolección Manual de Datos

- **Deficiencia:** La recolección manual de datos es un proceso laborioso y propenso a errores. Puede involucrar la descarga manual de archivos, la entrada de datos en sistemas, y la actualización periódica.
- **Implicación:** Los errores en la entrada de datos y los retrasos en la actualización pueden resultar en información inexacta o desactualizada, lo que afecta la toma de decisiones.

#### Desintegración de Datos

- **Deficiencia:** La información puede estar dispersa en múltiples sistemas o fuentes, como archivos CSV, hojas de cálculo y plataformas web, sin una integración centralizada.
- **Implicación:** La falta de consolidación dificulta la integración de datos y el análisis integral, haciendo que sea más difícil obtener una visión completa y precisa del desempeño financiero.

#### Falta de Actualización en Tiempo Real

- **Deficiencia:** Los sistemas actuales pueden no estar configurados para proporcionar datos en tiempo real, con actualizaciones periódicas que pueden retrasar la disponibilidad de la información.
- **Implicación:** La toma de decisiones basada en datos desactualizados puede conducir a estrategias inadecuadas y a una evaluación incorrecta del desempeño del mercado.

#### Análisis Limitado

- **Deficiencia:** Los informes actuales pueden ofrecer análisis básicos sin capacidades avanzadas para detectar patrones, tendencias o anomalías.

- **Implicación:** La falta de análisis avanzado puede limitar la capacidad de identificar oportunidades o riesgos en el mercado.

### Dificultades en la Visualización

- **Deficiencia:** Los datos pueden no ser visualizados de manera efectiva, con gráficos y reportes que no faciliten la comprensión rápida de la información.
- **Implicación:** La falta de visualizaciones claras y efectivas puede dificultar la interpretación de los datos y la comunicación de hallazgos clave a las partes interesadas.

#### 2.1.2.1. Soluciones Alternativas

##### Automatización con RPA

- **Solución:** Implementar un sistema de Automatización de Procesos Robóticos (RPA) para automatizar la recolección y carga de datos desde la Bolsa de Valores de Colombia hacia una base de datos central.
- **Beneficio:** Reducirá el riesgo de errores humanos, mejorará la eficiencia y garantizará que los datos estén actualizados en tiempo real.

##### Integración de Datos en una Base Centralizada

- **Solución:** Consolidar la información en una base de datos SQL centralizada, donde se pueda almacenar y gestionar de manera eficiente toda la información relevante.
- **Beneficio:** Facilitará el acceso a datos integrados y permitirá una gestión y análisis más coherentes y efectivos.

##### Actualización en Tiempo Real

- **Solución:** Configurar procesos para la actualización automática de datos en tiempo real, asegurando que la información más reciente esté disponible para el análisis.
- **Beneficio:** Permitir la toma de decisiones basada en datos actualizados y precisos, mejorando la capacidad de respuesta ante cambios en el mercado.

##### Implementación de Herramientas de Análisis Avanzado

- **Solución:** Utilizar herramientas de análisis y visualización de datos, como Power BI o Tableau, para realizar análisis avanzados y detectar patrones, tendencias y anomalías.
- **Beneficio:** Mejorará la capacidad para interpretar los datos y tomar decisiones informadas basadas en un análisis detallado.

### **Desarrollo de Dashboards Interactivos**

- **Solución:** Crear dashboards interactivos en una API web para visualizar los datos de manera clara y efectiva.
- **Beneficio:** Facilitara la interpretación de los datos y la comunicación de resultados clave a las partes interesadas de manera intuitiva.

#### **2.1.3. Habilidades analíticas actuales.**

En la fase de descubrimiento, la evaluación de las habilidades analíticas actuales es esencial para asegurar que se puedan aprovechar al máximo las herramientas y técnicas disponibles para el análisis de datos financieros de Ecopetrol. A continuación, se detalla cómo las habilidades analíticas actuales, junto con las herramientas utilizadas, contribuyen a abordar las deficiencias y optimizar el proceso.

#### **Conocimiento en Herramientas de Análisis de Datos**

El proyecto emplea varias herramientas clave, incluyendo SQL para la gestión y consulta de bases de datos, Excel en formato CSV para la manipulación inicial de datos, y Python para análisis avanzado y visualización. El uso de Python es particularmente crucial, ya que permite integrar una amplia gama de librerías especializadas. Entre estas, se incluyen pandas para la manipulación de datos, matplotlib y seaborn para visualización, y mplfinance para el análisis de datos financieros.

#### **Capacidades en Manipulación de Datos**

La manipulación de datos se realiza principalmente a través de pandas en Python, permitiendo la limpieza, transformación y preparación de los datos para su análisis. Además, se utiliza Excel para operaciones preliminares, como la conversión y exportación de datos en formato CSV. La habilidad para combinar estas herramientas asegura que los datos sean precisos y estén listos para el análisis profundo.

## **Conocimientos en Estadística y Análisis Predictivo**

Para el análisis predictivo, se emplean técnicas avanzadas como modelos de ARIMA y Prophet para la predicción de series temporales, y Linear Regression para modelos de regresión. La integración de librerías como statsmodels y prophet permite realizar análisis sofisticados que proporcionan proyecciones y evaluaciones detalladas de los datos financieros.

## **Habilidad en Integración de Datos**

La integración de datos de diversas fuentes se facilita mediante SQL y Python. La librería Conexion en Python se utiliza para consultar y conectar con SQL Server, mientras que herramientas como pyautogui y pygetwindow ayudan a automatizar la extracción de datos desde interfaces de usuario y otras fuentes. Esta integración asegura una visión consolidada y coherente de los datos.

## **Competencia en Desarrollo de Informes y Dashboards**

La creación de informes y dashboards es realizada con Flask para el desarrollo de aplicaciones web que muestran los resultados del análisis en una interfaz accesible. La visualización de datos se maneja con matplotlib, seaborn y mplfinance, lo que permite presentar los datos de manera clara y comprensible. Además, la conversión de gráficos en imágenes utilizando BytesIO y base64 facilita la inclusión de visualizaciones en los informes.

## **Conocimiento en Automatización de Procesos**

La automatización de tareas se realiza con pyautogui, que permite la automatización de interacciones con la interfaz de usuario, y subprocess para ejecutar comandos del sistema. La automatización mejora la eficiencia y precisión en la recolección y procesamiento de datos, permitiendo actualizaciones en tiempo real.

## **Experiencia en Análisis de Mercado**

El análisis de mercado se apoya en librerías especializadas como pmdarima para modelos de series temporales automatizados, y Isolation Forest para la detección de anomalías. La experiencia en estas técnicas permite una evaluación exhaustiva del comportamiento de los datos financieros y facilita la toma de decisiones informadas.

## 2.2. Fase de análisis: identificación de *gaps*

En la fase de análisis, el objetivo principal es identificar las brechas o gaps existentes entre la capacidad actual de los informes, los proveedores de tecnología necesarios, las diferencias entre los informes actuales y la información deseada, y los aspectos relacionados con la cronología, costes y recursos humanos implicados en el proyecto.

### 2.2.1. Capacidad de los informes actuales

Los informes actuales disponibles en la Bolsa de Valores de Colombia presentan varias limitaciones significativas que afectan la calidad y profundidad del análisis de datos. Estos informes están restringidos a representaciones porcentuales en formato numérico, sin incorporar herramientas avanzadas de visualización. Esta limitación impide una comprensión completa de los datos y restringe la capacidad de realizar un análisis detallado y dinámico.

#### **Limitaciones clave incluyen:**

- Representación numérica básica: Los informes se centran en mostrar datos en formato numérico, con representaciones porcentuales y métricas clave. Aunque estos números proporcionan información básica sobre el rendimiento de los activos, no permiten una interpretación profunda de las tendencias y patrones en los datos.
- Falta de gráficos interactivos: La ausencia de gráficos interactivos en los informes actuales limita la capacidad de explorar los datos de manera dinámica. Los gráficos interactivos permiten a los usuarios examinar diferentes aspectos de los datos, hacer zoom en áreas específicas, y comparar múltiples variables en tiempo real. Sin estas herramientas, los usuarios se ven obligados a interpretar los datos solo a través de números estáticos, lo que puede ser menos intuitivo y más propenso a errores.
- Análisis limitado: La falta de visualización avanzada y herramientas analíticas en los informes impide la realización de análisis detallados y predictivos. Los gráficos y visualizaciones interactivas son cruciales para identificar tendencias, patrones estacionales, y anomalías en los datos, que son esenciales para una toma de decisiones informada.
- Interactividad y personalización: Los informes actuales no ofrecen opciones de personalización ni interactividad, lo que limita la capacidad de los usuarios para adaptar los informes a sus necesidades específicas. La posibilidad de personalizar

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost  
gráficos, filtros y vistas permite una comprensión más completa y adaptada a los requisitos individuales o de negocio.

### 2.2.2. Proveedores de tecnología necesarias

Para abordar las deficiencias en los informes actuales y alcanzar los objetivos del proyecto, se requiere una combinación de tecnologías y herramientas que faciliten el análisis avanzado y la visualización interactiva de datos. A continuación, se detallan las herramientas y proveedores de tecnología necesarios para mejorar las capacidades analíticas y de visualización, teniendo en cuenta las herramientas y librerías previamente mencionadas:

#### 1. Herramientas de Gestión de Datos y Análisis

- **SQL Server:** Es un sistema de gestión de bases de datos relacional que ofrece robustas capacidades para almacenar, gestionar y consultar grandes volúmenes de datos. Permite la integración de datos desde múltiples fuentes y proporciona herramientas para el análisis y la generación de informes, lo cual es fundamental para manejar y procesar la información financiera de Ecopetrol.
- **Excel (en formato CSV):** Aunque es una herramienta básica, Excel es útil para la manipulación y análisis preliminar de datos en formato CSV. Su integración con otras herramientas permite una preparación y limpieza efectiva de los datos antes de cargarlos en la base de datos SQL.

#### 2. Librerías de Python para Análisis y Visualización

- **pandas:** Librería esencial para la manipulación y análisis de datos. Permite la limpieza, transformación y preparación de datos, facilitando el procesamiento de la información extraída y su carga en la base de datos.
- **matplotlib y seaborn:** Estas librerías son utilizadas para la visualización de datos. matplotlib permite crear gráficos estáticos personalizados, mientras que seaborn ofrece visualizaciones estadísticas avanzadas que ayudan a explorar y analizar los datos de manera más efectiva.
- **mplfinance:** Especializada en la visualización de datos financieros. Permite crear gráficos de velas y otras visualizaciones útiles para analizar el rendimiento del mercado de valores, lo que es crucial para el análisis de datos financieros de Ecopetrol.

- **Prophet:** Librería para la predicción de series temporales desarrollada por Facebook. Es útil para realizar análisis predictivo y pronósticos de datos financieros, identificando tendencias y patrones estacionales en los datos.
- **pmdarima:** Proporciona herramientas para la identificación y ajuste automático de modelos ARIMA, lo que es esencial para el análisis predictivo y la detección de patrones en series temporales.
- **ARIMA (a través de statsmodels):** Utilizada para la modelización y predicción de series temporales. Permite construir modelos de series temporales que ayudan a prever futuros valores basados en datos históricos.
- **LinearRegression (a través de scikit-learn):** Herramienta para la construcción de modelos de regresión lineales, que pueden ser utilizados para analizar relaciones entre variables y hacer predicciones basadas en estos modelos.
- **IsolationForest (a través de scikit-learn):** Utilizada para la detección de anomalías en los datos, lo que puede ser útil para identificar eventos inusuales o patrones atípicos en los datos financieros.

### 3. Herramientas de Desarrollo Web y Notificaciones

- **Flask:** Un microframework para el desarrollo de aplicaciones web en Python. Flask se utilizará para crear una API web que permita la visualización y el acceso a los datos analizados, facilitando la interacción con los resultados a través de una interfaz web.
- **win10toast:** Herramienta para enviar notificaciones de escritorio. Se utilizará para alertar sobre la finalización de procesos o la actualización de datos, proporcionando una forma de notificar a los usuarios sobre el estado de las tareas automatizadas.

### 4. Herramientas de Automatización y Manipulación de Datos

- **pyautogui y pygetwindow:** Estas herramientas permiten la automatización de tareas y la interacción con interfaces de usuario, facilitando la recolección y procesamiento de datos desde aplicaciones y páginas web.
- **subprocess:** Utilizada para ejecutar comandos del sistema y automatizar procesos adicionales necesarios para el manejo de datos y la ejecución de tareas.

### 2.2.3. Diferencia entre los informes actuales y la información deseada

La diferencia entre los informes actuales disponibles en la Bolsa de Valores de Colombia y la información deseada para el análisis de datos financieros es significativa y afecta la calidad del análisis y la toma de decisiones. A continuación, se detallan las principales diferencias:

#### 1. Formato y Representación de los Datos

- Informes Actuales: Los informes actuales se limitan a representaciones numéricas y porcentuales básicas. Estos informes presentan datos en un formato tabular estático, mostrando métricas clave como precios de cierre, precios máximos, y variaciones porcentuales. Sin embargo, no incluyen representaciones gráficas avanzadas ni visualizaciones interactivas.
- Información Deseada: La información deseada incluye no solo los datos numéricos básicos, sino también representaciones gráficas interactivas y detalladas. Se buscan gráficos de velas, gráficos de tendencias, y visualizaciones que permitan explorar los datos de manera dinámica. Esto facilita la identificación de patrones y tendencias, así como la interpretación de las variaciones y comportamientos en los datos financieros.

#### 2. Capacidades Analíticas y Predictivas

- Informes Actuales: Los informes actuales proporcionan una visión limitada de los datos, sin capacidades para realizar análisis predictivos o identificar patrones complejos. Los datos están disponibles en un formato estático, lo que restringe la capacidad de realizar análisis en profundidad.
- Información Deseada: La información deseada incluye herramientas de análisis predictivo y modelización de series temporales. Se busca incorporar técnicas avanzadas como el análisis ARIMA y modelos de predicción para prever futuros comportamientos del mercado. Esto permite realizar análisis más profundos y anticipar cambios en los precios y otras variables clave.

#### 3. Interactividad y Personalización

- Informes Actuales: Los informes actuales carecen de interactividad y opciones de personalización. Los usuarios no pueden interactuar con los datos, aplicar filtros personalizados, o ajustar las visualizaciones para enfocarse en aspectos específicos de los datos.

- **Información Deseada:** Se requiere que los informes sean interactivos y personalizables. Los usuarios deben poder aplicar filtros, seleccionar rangos de fechas, y explorar diferentes aspectos de los datos de manera dinámica. Las visualizaciones deben permitir la personalización y ofrecer opciones para adaptar la información a las necesidades específicas de análisis.

#### **4. Acceso y Disponibilidad de Datos en Tiempo Real**

- **Informes Actuales:** Los informes actuales no proporcionan acceso a datos en tiempo real. Los datos se presentan en informes periódicos, lo que puede limitar la capacidad de reaccionar rápidamente a cambios en el mercado.
- **Información Deseada:** Se necesita acceso a datos en tiempo real para permitir una monitorización continua y una toma de decisiones ágil. La capacidad de analizar datos en tiempo real y recibir actualizaciones instantáneas es crucial para responder a eventos del mercado de manera oportuna.

#### **5. Capacidades de Visualización Avanzada**

- **Informes Actuales:** Las capacidades de visualización en los informes actuales son limitadas y básicas. Los informes no incluyen gráficos avanzados ni herramientas para explorar los datos de manera efectiva.
- **Información Deseada:** Se requiere la inclusión de visualizaciones avanzadas como gráficos interactivos, mapas de calor, y análisis de tendencias. Estas visualizaciones deben facilitar una comprensión más completa y detallada de los datos financieros, permitiendo a los usuarios identificar rápidamente patrones y anomalías.

##### **2.2.4. Cronología, costes y recursos humanos implicados**

###### **2.2.4.1. Cronología**

###### **Fase 1: "Visión y Diseño"**

**Duración:** 1 a 2 meses

**Descripción:** Durante esta fase, se llevará a cabo la planificación detallada del proyecto, que incluye la recopilación de requisitos, la evaluación de las tecnologías necesarias y el diseño de

## Fase 2: "Construcción del Núcleo"

**Duración:** 3 a 4 meses

**Descripción:** Esta fase incluye el desarrollo e implementación del sistema automatizado. Se creará el RPA para la extracción de datos, se integrará la base de datos SQL y se desarrollará la API web en Flask. Se establecerán las conexiones entre todos los componentes y se comenzará a dar forma al sistema.

## Fase 3: "Pruebas y Refinamiento"

**Duración:** 1 a 2 meses

**Descripción:** En esta fase, se realizarán pruebas exhaustivas para asegurar que el sistema funcione de manera correcta y eficiente. Se validará la precisión de los datos, se corregirán errores y se ajustarán los detalles para garantizar que el sistema cumpla con los requisitos establecidos.

## Cronograma de Actividades

Figura 1. Diagrama de Gantt de cronograma de actividades.



Nota: (1) Elaboración propia basada en actividades para cronograma.

### 2.2.4.2. Costos y recursos humanos implicados

Dado que el desarrollo del proyecto será realizado íntegramente por mí, no se prevén costos asociados a la contratación de recursos humanos adicionales. Además, se utilizarán

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost  
herramientas y software gratuitos, lo que elimina la necesidad de asignar un presupuesto para licencias o adquisiciones de tecnología.

#### **Recursos Humanos:**

- Desarrollador: No hay costos asociados, ya que el yo asume el rol de desarrollador principal.

#### **Software y Herramientas:**

- Python: Software de código abierto, sin costo.
- SQL Server (versión gratuita): Base de datos gratuita proporcionada por Microsoft, sin costo.
- Librerías de Python (Flask, pandas, matplotlib, etc.): Todas las librerías utilizadas son de código abierto, sin costo.

#### **Infraestructura y Hosting:**

- Localhost: El sistema se implementará y visualizará localmente, por lo que no se requerirá hosting externo, eliminando así cualquier costo asociado.

### **2.3. Fase de recomendaciones: alcance, prioridades y presupuesto**

#### **2.3.1. Promoción del proyecto en la organización**

Aunque el proyecto es de carácter académico, si se escalara a un entorno real, se enfocaría en instituciones financieras como bancos, que demandan análisis de datos en tiempo real. En este caso, la promoción del proyecto debería ser significativamente ampliada. Se recomendaría la elaboración de una estrategia de comunicación dirigida a los altos ejecutivos y equipos técnicos de las entidades financieras, destacando cómo el sistema puede optimizar el monitoreo financiero y mejorar la toma de decisiones estratégicas. Esto podría incluir la creación de material promocional especializado, casos de estudio y la participación en conferencias del sector financiero para mostrar las capacidades del sistema.

#### **2.3.2. Demostraciones**

Para un despliegue en un entorno financiero real, las demostraciones necesitarían ser más sofisticadas y orientadas a resolver problemas específicos de la industria. Estas

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost  
demostraciones deberían incluir simulaciones de escenarios financieros reales, mostrando cómo el sistema puede manejar grandes volúmenes de datos y realizar análisis complejos en tiempo real. Además, sería recomendable realizar pruebas piloto con datos históricos y en entornos de pruebas controlados dentro de las entidades interesadas, para demostrar la robustez y la escalabilidad del sistema. Se podría involucrar a equipos de TI y a analistas financieros en estas pruebas para asegurar que el sistema cumpla con las expectativas y requerimientos del sector.

### 2.3.3. Presupuesto

#### **Infraestructura Tecnológica:**

Servidores y Almacenamiento: En lugar de servidores físicos, se podría optar por soluciones en la nube, como AWS o Azure, donde el costo mensual podría ser de \$200 a \$1,000 USD, dependiendo del uso y la capacidad.

#### **Licencias de Software:**

Herramientas Analíticas y de Visualización: herramientas open-source como Python y sus bibliotecas pueden reducir considerablemente los costos.

#### **Mantenimiento y Actualización del Sistema:**

Mantenimiento de Software y Hardware: El mantenimiento de la infraestructura de nube y software podría costar entre \$2,000 anuales.

### 3. Análisis y definición

#### 3.1. Análisis de los datos a utilizar

El análisis de los datos a utilizar en el proyecto se centra en una serie de columnas clave que proporcionan una visión completa del comportamiento de las acciones de Ecopetrol. Estas columnas incluyen Fecha, Nemotécnico, Precio de cierre, Precio máximo, Precio promedio ponderado, Precio mínimo, Variación absoluta, Variación porcentual, Cantidad, y Volumen.

**Fecha:** Esta columna registra el día en que se realizaron las transacciones. Es fundamental para el análisis temporal, permitiendo la observación de tendencias y patrones en el comportamiento de los precios a lo largo del tiempo.

**Nemotécnico:** En este caso, 'ECOPETROL' identifica el activo financiero específico. Aunque el nemotécnico es constante para este análisis, en contextos más amplios podría haber múltiples activos con diferentes nemotécnicos.

**Precio de cierre:** El precio de cierre es el valor final de una acción al final del día de negociación. Este valor es crucial para determinar el rendimiento diario de la acción y para el análisis de la tendencia general del mercado.

**Precio máximo y Precio mínimo:** Estos valores representan el precio más alto y el más bajo alcanzado durante el día de negociación. Juntos, ayudan a entender la volatilidad del activo durante el periodo de análisis.

**Precio promedio ponderado:** Este precio refleja el promedio de las transacciones ponderado por el volumen de cada transacción. Es útil para obtener una perspectiva más precisa del valor real de la acción en comparación con los precios de cierre y extremos.

**Variación absoluta:** Indica el cambio en el precio de cierre desde el día anterior. Es una medida directa de la variación en el valor del activo en términos absolutos.

**Variación porcentual:** Muestra el cambio en el precio de cierre en términos porcentuales, ofreciendo una perspectiva relativa del cambio del valor del activo en relación con el precio del día anterior.

**Cantidad:** Representa el número de acciones negociadas durante el día. Este dato es importante para entender el nivel de actividad y la liquidez del mercado.

**Volumen:** Refleja el total monetario de las transacciones realizadas, calculado como el precio de cierre multiplicado por la cantidad de acciones negociadas. Este valor es esencial para evaluar la magnitud de las operaciones y el impacto en el mercado.

### 3.2. Análisis histórico y/o limpieza de datos

El análisis histórico implica la recopilación y evaluación de datos pasados para identificar patrones, tendencias y anomalías. En este proyecto, se trabajará con datos financieros históricos de ECOPETROL, una de las principales empresas en el mercado de valores colombiano. Estos datos incluyen precios de cierre, precios máximos y mínimos, volúmenes de transacción, y variaciones porcentuales y absolutas. El análisis de estos datos históricos permitirá establecer un contexto sobre el comportamiento de las acciones de ECOPETROL, lo que es fundamental para cualquier análisis predictivo o toma de decisiones de inversión.

El primer paso del proceso automatizado de los históricos es abrir la página específica de la Bolsa de Valores de Colombia (BVC) que contiene la información sobre ECOPETROL. Para ello, se utiliza la función `webbrowser.open(url)`. Este comando lanza automáticamente el navegador predeterminado en el sistema y carga la URL indicada. Esto permite que el RPA acceda directamente al sitio web donde se encuentran los datos financieros necesarios, eliminando la necesidad de navegación manual.

Una vez que la página de la BVC está abierta, el script debe asegurarse de que la ventana correcta esté activa y visible. Esto se realiza utilizando pyautogui.getWindowsWithTitle("Bolsa de Valores")[0], que busca la ventana con el título "Bolsa de Valores". El script entra en un bucle donde verifica continuamente si esta ventana está activa. Si no lo está, el script intenta maximizar la ventana para que sea visible. Este proceso garantiza que el RPA interactúe con la ventana correcta y no con otra aplicación abierta.

Después de confirmar que la ventana correcta está activa, el RPA comienza a interactuar con la página web para avanzar hacia la sección donde se encuentran los datos históricos. Utilizando pyautogui.moveTo(950, 550) y pyautogui.click(), el script mueve el cursor a unas coordenadas específicas en la página y realiza un clic. Esta acción suele ser necesaria para activar elementos interactivos en la página, como menús o botones, que permiten al usuario acceder a los datos necesarios.

Para asegurarse de que el calendario o la sección específica donde se selecciona la fecha esté visible, el script utiliza pyautogui.scroll(-2000) para desplazarse hacia abajo en la página. Este desplazamiento asegura que el contenido relevante esté en el campo de visión del navegador, lo que es crucial para que las siguientes interacciones del RPA se realicen correctamente. Este paso es importante en páginas largas donde los elementos clave no están visibles inmediatamente al cargar la página.

Con el calendario ahora visible, el script selecciona automáticamente la fecha actual. Esto se logra calculando la posición del día actual dentro del calendario utilizando el código:

Figura 2. Código cálculo de fecha

```
76     columna = (dia_actual - 1) % 7
77     fila = (dia_actual - 1) // 7
78     pyautogui.moveTo(base_x + columna * cell_width, base_y + fila * cell_height)
79     pyautogui.doubleClick()
```

*Nota: (1) Elaboración según estructura creada en Python.*

El RPA calcula la fila y columna donde se encuentra el día actual y mueve el cursor a esa posición, realizando un doble clic para seleccionar la fecha. Este paso es crucial para asegurar que los datos extraídos correspondan al día actual, manteniendo así la precisión y relevancia de la información recolectada.

Una vez seleccionada la fecha, el RPA procede a descargar el archivo CSV que contiene los datos financieros de ECOPETROL. Utilizando pyautogui.moveTo(1300, 500) y pyautogui.click(), el script mueve el cursor al botón de descarga y hace clic en él para iniciar la descarga del archivo. Este paso automatiza la extracción de datos y asegura que el archivo se guarde en la ubicación predeterminada del sistema, usualmente la carpeta de descargas.

Después de que la descarga se completa, el script proporciona una confirmación visual al usuario utilizando ToastNotifier(). La función toast.show\_toast("Descarga completada", "El documento ha sido descargado exitosamente.", duration=10) muestra una notificación en la esquina inferior derecha de la pantalla, informando que la descarga se ha realizado con éxito. Esta notificación es útil para confirmar que el proceso ha concluido sin errores y que los datos están listos para su uso.

El último paso del RPA es iniciar otro script que importará los datos descargados en una base de datos SQL. Esto se logra utilizando subprocess.call(["python", "Import\_data.py"]). Al ejecutar este comando, el RPA lanza un nuevo script de Python diseñado para cargar el archivo CSV en la base de datos, facilitando así su análisis posterior. Este paso cierra el ciclo de automatización, desde la recolección de datos hasta su integración en el sistema de análisis, asegurando un flujo de trabajo completamente automatizado y eficiente. La Automatización Robótica de Procesos (RPA) ha demostrado ser un facilitador clave en el sector financiero, ya que permite la automatización de tareas repetitivas y proporciona una eficiencia operativa sin precedentes en la recopilación y análisis de datos financieros. Lee et al. (2020) muestran cómo el uso de RPA en el sector financiero permite la reducción de errores humanos y mejora la rapidez en la recolección de datos, esencial en entornos de mercado volátiles.

### 3.3. Modelo Propuesto

#### 3.3.1. Captación de datos históricos

Primero, se define la ruta donde se espera encontrar el archivo CSV descargado. En este caso, la ruta se establece como 'C:\\Users\\Michel Abello\\Downloads', lo que indica que el archivo se ubicará en la carpeta de descargas del usuario.

Figura 3. Código ruta de carpeta

```
10  # Ruta de la carpeta de descargas
11  downloads_path = 'C:\\Users\\Michel Abello\\Downloads'
```

*Nota: (1) Elaboración según estructura creada en Python.*

Se utiliza un patrón de búsqueda para localizar el archivo CSV en la carpeta de descargas. El patrón especifica que el nombre del archivo debe comenzar con "ECOPETROL" seguido de cualquier otro carácter y con una extensión .csv.

Figura 4. Código importación de documento csv

```
14  csv_file_pattern = os.path.join(downloads_path, 'ECOPETROL_*.csv')
15  csv_files = glob.glob(csv_file_pattern)
```

*Nota: (1) Elaboración según estructura creada en Python.*

El script verifica si se encontró algún archivo que coincida con el patrón. Si no se encuentra ningún archivo, se genera un error (FileNotFoundException), lo que previene que el script continúe sin datos.

Figura 5. Código validación de proceso entra en error

```
17  if not csv_files:
18      raise FileNotFoundError("No se encontró ningún archivo CSV que comience con 'ECOPETROL' en la carpeta de descargas.")
```

*Nota: (1) Elaboración según estructura creada en Python.*

Una vez localizado el archivo, se procede a leer su contenido utilizando la biblioteca pandas. Además, la columna de fecha se convierte al formato deseado (%Y/%m/%d), lo cual es necesario para garantizar la correcta inserción en la base de datos.

Figura 6. Código de lectura y estructuración de csv

```
28 df = pd.read_csv(csv_file_path, delimiter=';')
29 # Convertir la columna 'Fecha' al formato deseado %Y/%m/%d
30 df['Fecha'] = pd.to_datetime(df['Fecha']).dt.strftime('%Y/%m/%d')
```

*Nota: (1) Elaboración según estructura creada en Python.*

El script establece una cadena de conexión para acceder a SQL Server utilizando autenticación de Windows. Esta cadena de conexión se crea dinámicamente y se utiliza posteriormente para establecer la conexión con la base de datos.

Figura 7. Código para cadena de conexión SQL

```
32 # Crear la cadena de conexión para autenticación de Windows
33 connection_string = f'mssql+pyodbc://@{db_host}/{db_name}?driver=ODBC+Driver+17+for+SQL+Server&trusted_connection=yes'
34 engine = create_engine(connection_string)
```

*Nota: (1) Elaboración según estructura creada en Python.*

Con la conexión establecida, el script inserta los datos del archivo CSV en una tabla específica dentro de la base de datos. En caso de que la tabla ya contenga datos, los nuevos registros se añaden (if\_exists='append').

Figura 8. Código para cadena de conexión SQL

```
37     # Insertar los datos en la tabla SQL
38     df.to_sql(name=table_name, con=engine, if_exists='append', index=False)
```

*Nota: (1) Elaboración según estructura creada en Python.*

Después de confirmar que los datos han sido importados correctamente, el script elimina el archivo CSV original para liberar espacio y evitar duplicados en futuras importaciones.

Figura 9. Código que elimina csv procesado y almacenado

```
41      # Eliminar el archivo CSV
42      os.remove(csv_file_path)
```

*Nota: (1) Elaboración según estructura creada en Python.*

### 3.3.2. Consulta SQL para conectar los datos con la API

Se utilizará Python para diseñar la conexión de la base de datos SQL Server, realizar una consulta y recuperar los datos en un formato que sea fácil de manipular utilizando la biblioteca Pandas. A continuación, se explica el funcionamiento de cada parte del código:

Figura 10. Código que muestra la función definida para server SQL

```
5  def consultar_sql_server():
```

*Nota: (1) Elaboración según estructura creada en Python.*

Esta función es responsable de establecer la conexión con la base de datos, ejecutar una consulta SQL para obtener datos, y devolver esos datos como un DataFrame de Pandas.

Figura 11. Código parámetros de conexión SQL

```
13      # Establece la conexión utilizando Windows Authentication
14      conn = pyodbc.connect(f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;')
```

*Nota: (1) Elaboración según estructura creada en Python.*

Aquí se establece la conexión a SQL Server utilizando los detalles especificados anteriormente. Trusted\_Connection=yes indica que se utilizará la autenticación de Windows para conectarse al servidor.

Figura 12. Código parámetros de conexión SQL

```

16     # Mostrar notificación de proceso terminado
17     toaster = ToastNotifier()
18     toaster.show_toast("Conexión exitosa a SQL Server utilizando Windows Authentication", duration=10)

```

*Nota: (1) Elaboración según estructura creada en Python.*

Una vez establecida la conexión, se crea una notificación de escritorio para informar al usuario que la conexión fue exitosa.

Figura 13. Código para selección de tabla

```

21     query = f'SELECT * FROM {table}'
22     data = pd.read_sql(query, conn)

```

*Nota: (1) Elaboración según estructura creada en Python.*

A continuación, se define una consulta SQL (query) que selecciona todos los registros de la tabla especificada. Esta consulta se ejecuta utilizando `pd.read_sql()`, que devuelve los resultados en un DataFrame de Pandas, facilitando así el manejo de los datos en Python.

### 3.3.3. Estructuración de la API y los análisis

El código presentado implementa una aplicación web utilizando el framework Flask. Esta aplicación tiene como objetivo cargar datos desde una base de datos SQL Server, realizar una serie de transformaciones y análisis sobre los datos, generar gráficos que representen los resultados de estos análisis, y finalmente mostrar estos gráficos en una interfaz web. A continuación, se explica cada componente del código paso a paso:

Figura 14. Código creación de framework

```

18     app = Flask(__name__)

```

*Nota: (1) Elaboración según estructura creada en Python.*

El primer paso es la creación de una instancia de Flask, que será la base de la aplicación web.

Flask es un microframework que permite construir aplicaciones web de manera sencilla.

Figura 15. Código para definir la raíz de la ruta

```
20     @app.route('/')
21     def home():
```

*Nota: (1) Elaboración según estructura creada en Python.*

Se define una ruta en la aplicación, en este caso la raíz ('/'), que será la dirección a la que se accederá para cargar la página principal. Cuando un usuario visita esta ruta, se ejecutará la función home().

Figura 16. Código llamado de conexión SQL

```
23     data = consultar_sql_server()
24
25     if data is None:
26         return "Error al conectar a la base de datos."
```

*Nota: (1) Elaboración según estructura creada en Python.*

En esta parte del código, se realiza una consulta a la base de datos SQL Server mediante la función consultar\_sql\_server(). Si la consulta falla y no se obtienen datos (data is None), la aplicación devolverá un mensaje de error.

Figura 17. Código llamado de columnas

```
29     columns_to_convert = ['Precio cierre', 'Precio máximo', 'Precio promedio ponderado', 'Precio mínimo',
30                           'Variación absoluta', 'Variación porcentual', 'Cantidad', 'Volumen']
31
32     for column in columns_to_convert:
33         data[column] = data[column].apply(lambda x: float(str(x).replace(',', '').replace('.', '.')))
```

*Nota: (1) Elaboración según estructura creada en Python.*

Las columnas específicas que contienen valores numéricos en formato de texto son convertidas a tipo float para asegurar que se puedan realizar operaciones matemáticas y análisis adecuados sobre ellas.

Figura 18. Código para crear grafico de barras

```

36     plt.figure(figsize=(12, 8))
37     sns.histplot(data['Precio cierre'], kde=True)
38     plt.title('Distribución de Precio de Cierre', fontsize=14, fontweight='bold')
39     plt.xticks(rotation=45)
40     plt.xlabel('Precio Cierre')
41     plt.ylabel('Frecuencia')

```

*Nota: (1) Elaboración según estructura creada en Python.*

Se genera un histograma que muestra la distribución del precio de cierre de las acciones. Este gráfico se convierte en una imagen PNG en memoria, que luego será renderizada en la página web.

Figura 19. Código para crear mapa de correlación

```

57     # Cambiar el nombre de la columna "Precio promedio ponderado" a "Precio promedio"
58     data_selected.rename(columns={'Precio promedio ponderado': 'Precio promedio'}, inplace=True)
59
60     # Realizar la matriz de correlación entre las variables seleccionadas
61     correlation_matrix = data_selected.corr(method='pearson')
62
63     # Crear un mapa de calor con fondo transparente y texto en blanco
64     plt.figure(figsize=(12, 10), facecolor='none')
65     sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1, cbar_kws={'ticks': [-1, 0, 1]})
66     plt.xticks(rotation=45, ha='right', fontsize=9, color='white')
67     plt.yticks(fontsize=9, color='white')
68     plt.title('Matriz de Correlación (Valores por encima del umbral)', fontsize=14, fontweight='bold', color='white')

```

*Nota: (1) Elaboración según estructura creada en Python.*

Se seleccionan ciertas columnas para calcular la matriz de correlación, que indica cómo se relacionan las diferentes variables entre sí. Esta matriz se presenta mediante un mapa de calor, que es también convertido en una imagen PNG.

**Figura 20.** Código para crear histograma

```

118 plt.figure(figsize=(12, 8), facecolor='none')
119 plt.plot(data['Fecha'], data['Precio cierre'], marker='o', linestyle='-', color='b', label='Datos históricos')
120 plt.plot(data['Fecha'], data['Prediccion'], linestyle='--', color='r', label='Regresión lineal')
121 plt.title('Gráfico de Lineas con Regresión Lineal', fontsize=14, fontweight='bold', color='white')
122 plt.xlabel('Fecha', color='white')
123 plt.ylabel('Precio de Cierre', color='white')
124 plt.legend(facecolor='none', edgecolor='white', labelcolor='white')
125 plt.grid(True, color='gray')
126 plt.xticks(rotation=45, color='white')
127 plt.yticks(color='white')
128 plt.tight_layout()

```

*Nota: (1) Elaboración según estructura creada en Python.*

Se generan histogramas para cada una de las variables numéricas seleccionadas, permitiendo observar la distribución de cada variable de forma individual.

**Figura 21.** Código para crear regresión

```

144 plt.figure(figsize=(12, 8))
145 plt.plot(data['Fecha'], data_selected['Variación absoluta'], marker='o', label='Variación Absoluta')
146 plt.plot(data['Fecha'], data_selected['Variación porcentual'], marker='o', label='Variación Porcentual')
147 plt.title('Variación de Precios Absoluta y Porcentual', fontsize=14, fontweight='bold')
148 plt.xlabel('Fecha')
149 plt.ylabel('Variación')
150 plt.legend()
151 plt.grid(True)

```

*Nota: (1) Elaboración según estructura creada en Python.*

Se prepara y limpia la columna de fechas, transformándola en un formato numérico para realizar una regresión lineal con respecto al precio de cierre. El modelo entrenado genera predicciones, y se visualiza una gráfica con la línea de regresión superpuesta sobre los datos históricos.

**Figura 22.** Código para crear proyección

```

177  plt.figure(figsize=(12, 8), facecolor='none')
178  plt.plot(forecast['ds'], forecast['yhat'], label='Proyección')
179  plt.plot(data['ds'], data['y'], label='Datos reales')
180  plt.xlabel('Fecha')
181  plt.ylabel('Precio de cierre')
182  plt.title('Proyección del precio de cierre para el siguiente mes', fontsize=14, fontweight='bold', color='white')
183  plt.legend(facecolor='none', edgecolor='white', labelcolor='white')
184  plt.grid(True, color='gray')

```

*Nota: (1) Elaboración según estructura creada en Python.*

Se grafican las variaciones absolutas y porcentuales de los precios a lo largo del tiempo, proporcionando una visión clara de cómo fluctúan estas medidas.

Figura 23. Código para convertir los gráficos en imágenes

```

193  # Convertir el gráfico a una imagen PNG en memoria
194  img_candlestick = BytesIO()
195  plt.savefig(img_candlestick, format='png', transparent=True)
196  plt.close() # Cerrar el gráfico
197  img_candlestick.seek(0)
198  plot_url_candlestick = base64.b64encode(img_candlestick.getvalue()).decode()

```

*Nota: (1) Elaboración según estructura creada en Python.*

Finalmente, todos los gráficos generados se pasan a la plantilla HTML para ser renderizados en la página web que verá el usuario. Esto se realiza mediante la función render\_template, que permite integrar las imágenes en la interfaz visual de la aplicación.

Figura 24. Código para unificar el llamado de las imágenes

```

200  # Renderizar la plantilla y pasar las URLs de las imágenes
201  return render_template('index.html', plot_url=plot_url, plot_url_corr=plot_url_corr,

```

*Nota: (1) Elaboración según estructura creada en Python.*

Este modelo propuesto permite a los usuarios acceder a un análisis visual e interactivo de los datos financieros, con la posibilidad de observar distribuciones, correlaciones, tendencias y proyecciones de los precios de cierre, todo directamente desde una página web.

### 3.3.4. Visualización de la API en el local Host

La visualización de los datos en tiempo real se realiza a través de una API web que se ejecuta en un servidor local (localhost). Cuando se inicia la aplicación Flask con el comando flask run, la consola muestra mensajes que indican el estado del servidor.

El mensaje "Serving Flask app 'app'" significa que Flask ha identificado la aplicación que se está ejecutando, en este caso, llamada 'app'. El aviso "Debug mode: off" indica que el modo de depuración está desactivado, lo cual es adecuado para un entorno de desarrollo, pero no recomendado para producción. La advertencia "WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead" subraya que Flask está corriendo en su servidor de desarrollo integrado, que es adecuado para pruebas y desarrollo, pero no es lo suficientemente robusto para manejar una implementación en producción. Para producción, se recomienda usar un servidor WSGI, como Gunicorn o uWSGI, que son más seguros y escalables.

Figura 25. Ejecución del local Host

```
* Serving Flask app 'app'  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit
```

*Nota: (1) Mensaje de confirmación de proceso exitoso ejecutado en Python.*

Finalmente, el mensaje "Running on http://127.0.0.1:5000" indica que la aplicación web está disponible localmente en la dirección http://127.0.0.1:5000, donde se puede acceder desde un navegador web para visualizar y monitorear los datos de Ecopetrol que el RPA ha extraído y analizado en tiempo real. Esta configuración permite verificar que todo el flujo, desde la extracción hasta la visualización, está funcionando correctamente en el entorno de desarrollo.

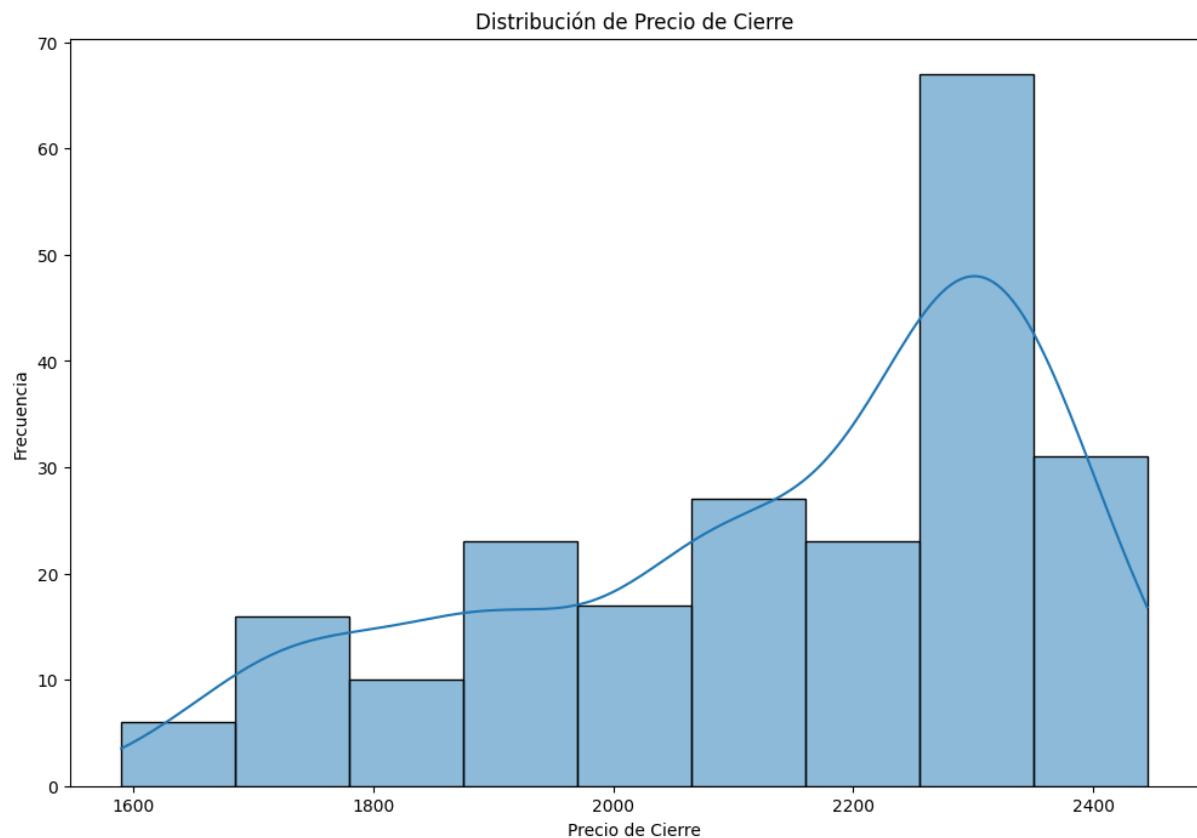
Figura 26. Banner inicial para el localhost



*Nota: (1) Elaboracion propia en Python.*

Se presentará una vista inicial que brindará una breve introducción sobre la empresa Ecopetrol, acompañada de un enlace que permitirá acceder directamente a la página de la Bolsa de Valores de Colombia para consultar los datasets originales utilizados en la API.

Figura 27. Gráfico de barras según precio de cierre



*Nota: (1) Elaboracion propia segun datos de BVC*

En el primer análisis llevado a cabo, se examinó la distribución de los precios de cierre, lo que permitió identificar patrones clave en los datos. Se destacó que el precio de cierre de \$2,300 presentó la mayor frecuencia, lo que indica que este valor fue el más comúnmente registrado en el periodo analizado. Además, se observó un intervalo significativo de precios, comprendido entre \$2,250 y \$2,350, lo que sugiere una concentración notable de valores en esta banda.

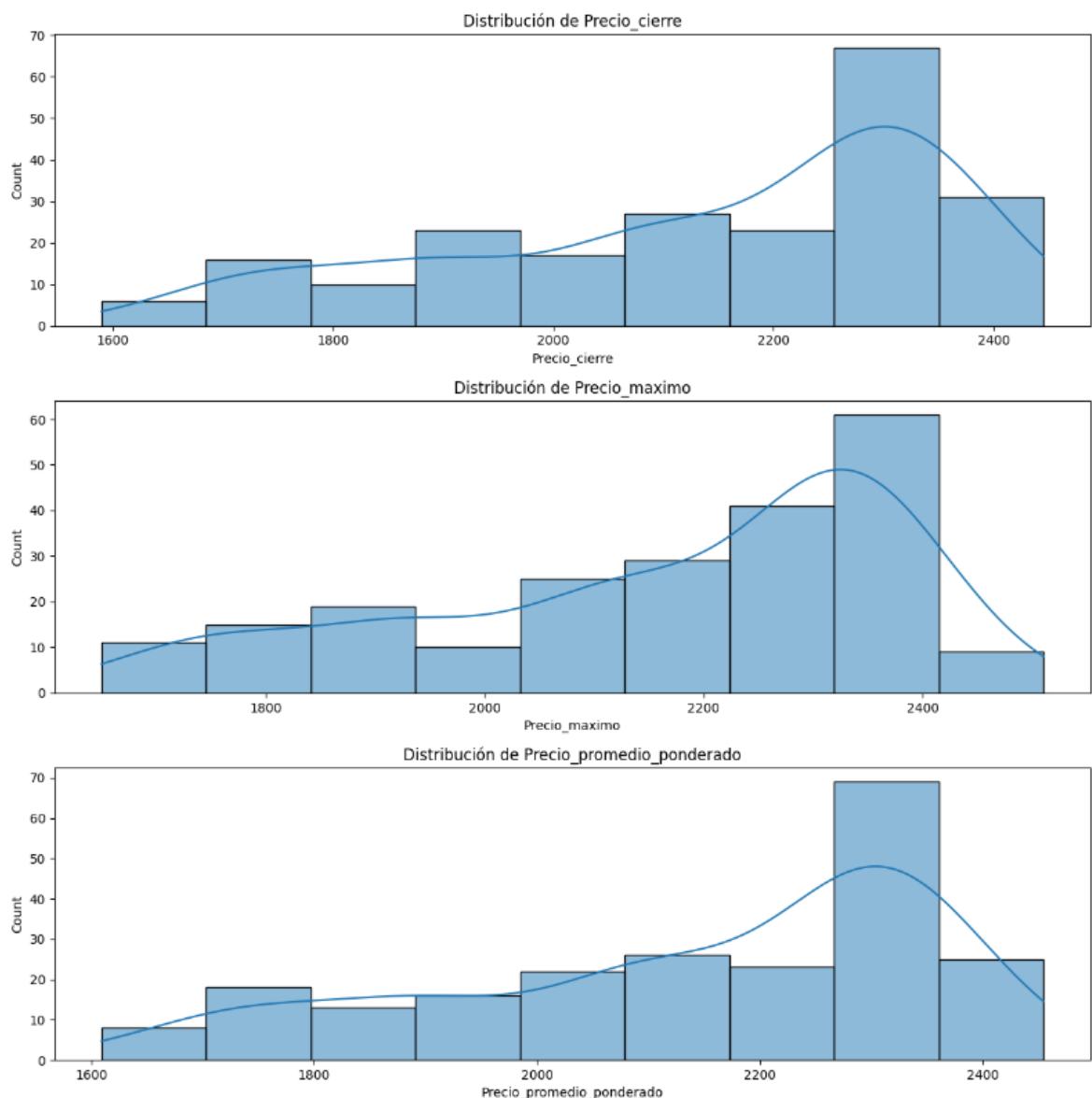
Figura 28. Diagrama de correlación de las variables

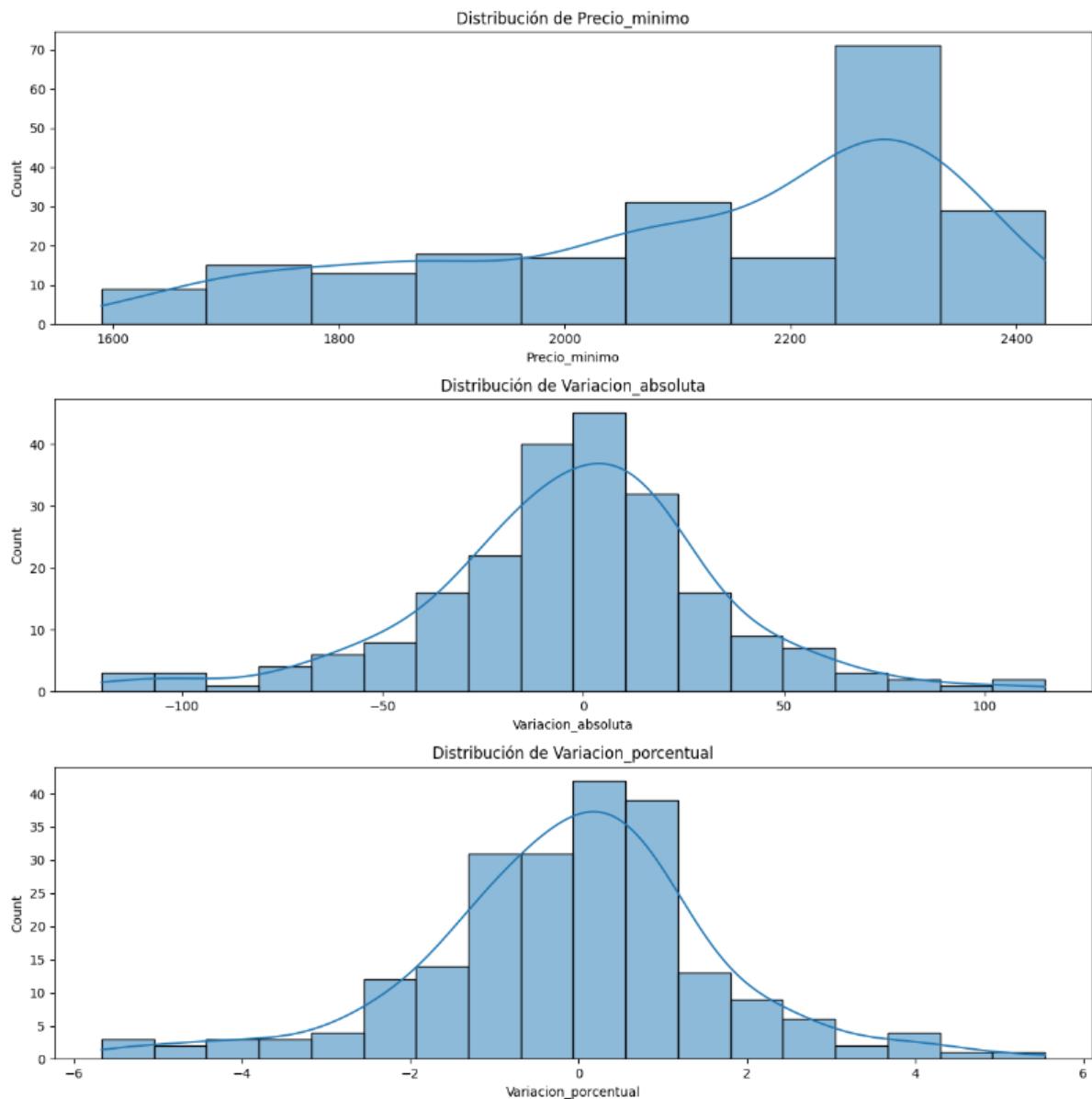


*Nota: (1) Elaboracion propia segun datos de BVC*

Como segundo análisis, se desarrolló un mapa de calor de correlaciones (colwarm) para identificar el nivel de correlación entre las variables del dataset. Este análisis reveló una correlación muy fuerte, con valores entre 0.99 y 0.99, entre las variables precio de cierre, precio máximo, precio promedio y precio mínimo. Este alto grado de correlación sugiere que estas variables se mueven en conjunto, indicando que cuando una de ellas aumenta o disminuye, las demás tienden a seguir un patrón similar. Este hallazgo es crucial para entender la relación entre los diferentes precios registrados y cómo estos pueden influir en el comportamiento general del mercado.

Figura 29. Gráficos de distribución según variables del crudo





Nota: (1) Elaboración propia según datos de BVC

El tercer análisis consistió en una visualización de la distribución de las diferentes columnas del dataset. A través de gráficos detallados, se pudo observar cómo se comportan las distintas variables, lo que facilitó la identificación de patrones, tendencias y posibles anomalías en los datos. Esta visualización proporcionó una visión más clara de la dispersión y concentración de los valores, permitiendo un entendimiento más profundo de la estructura interna del dataset y de las relaciones entre las diferentes variables.

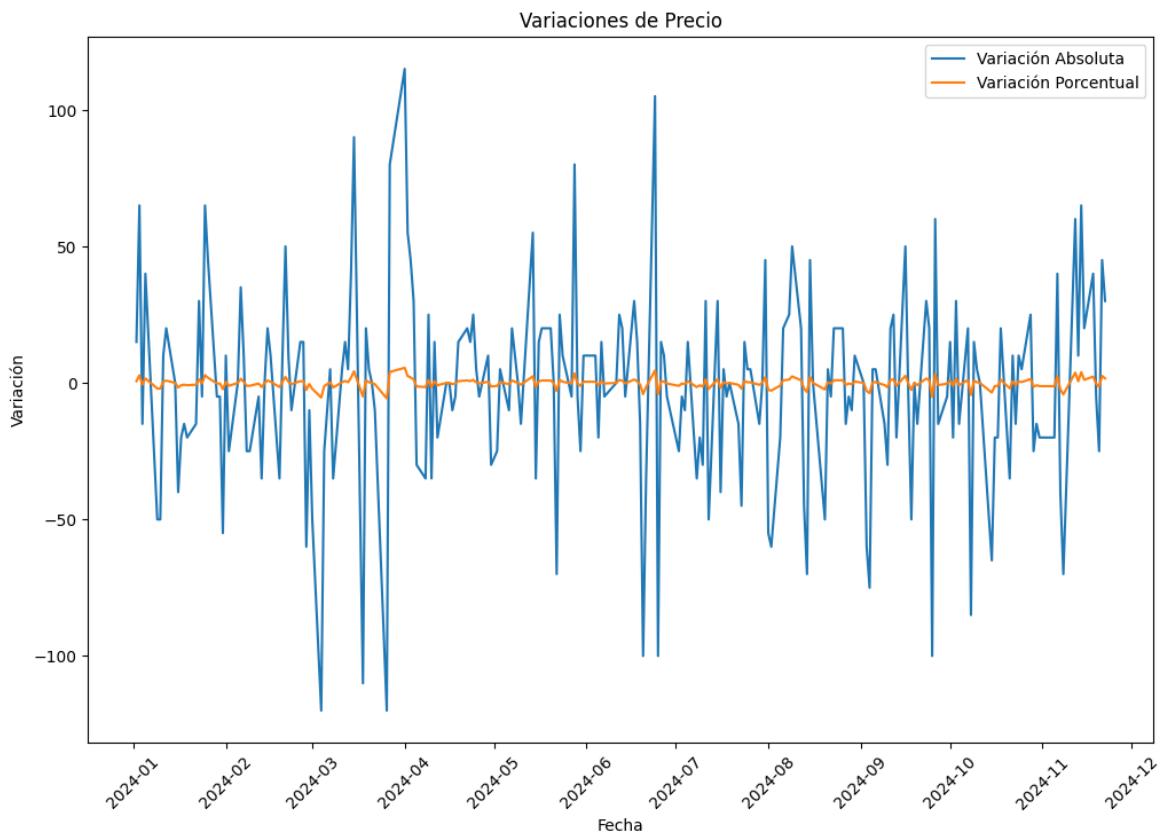
Figura 30. Histograma según regresión lineal



Nota: (1) Elaboracion propia segun datos de BVC

El cuarto análisis, tomó en cuenta los hallazgos previos para enriquecer su enfoque. En esta fase, se desarrolló una regresión lineal utilizando los datos históricos disponibles. A través de la visualización resultante, se generó una línea de tendencia que reflejó la regresión lineal aplicada. Este análisis reveló que la línea de tendencia tiene una inclinación a la baja, lo que sugiere una posible disminución en los valores a lo largo del tiempo. Este resultado es fundamental para anticipar futuros comportamientos del mercado y tomar decisiones informadas.

Figura 31. Gráfico de variación porcentual y absoluta

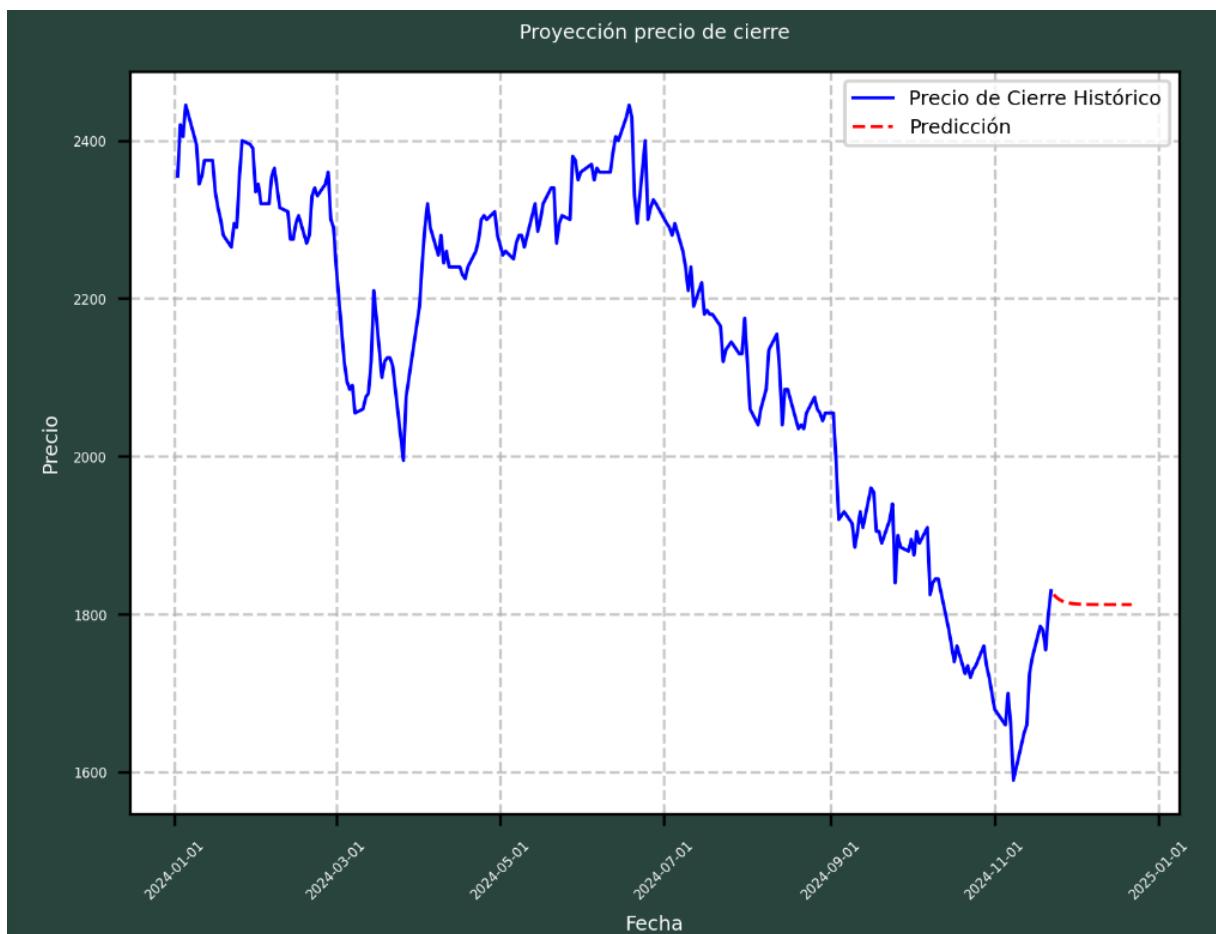


*Nota: (1) Elaboracion propia segun datos de BVC*

El quinto análisis abarcó el estudio de la variación absoluta y la variación porcentual de los precios de cierre a lo largo del período histórico evaluado. Este análisis permitió identificar los cambios en los precios de manera tanto absoluta como relativa, proporcionando una visión más detallada de la volatilidad y las fluctuaciones del mercado durante el tiempo analizado. Estos indicadores son clave para entender la dinámica de los precios y para evaluar la magnitud y frecuencia de las variaciones en el mercado.

El último análisis se centró en proyectar el precio de cierre para el mes siguiente. En la visualización generada, se utilizó una línea azul para representar el comportamiento histórico del precio de cierre, mientras que una línea roja mostró la proyección futura del mismo. Este enfoque permitió comparar de manera clara la tendencia pasada con la predicción, ofreciendo una herramienta visual poderosa para anticipar el comportamiento del mercado en el corto plazo.

Figura 32. Proyección del precio de cierre



Nota: (1) Elaboracion propia segun datos de BVC

Las API web y herramientas de visualización de datos se han convertido en elementos fundamentales en la interpretación de datos financieros complejos. Según Khan, Shah y Lee (2021), la visualización efectiva de datos en tiempo real a través de dashboards interactivos mejora significativamente la capacidad de análisis y la toma de decisiones en el sector financiero.

## 4. Construcción, prueba, implementación y despliegue

### 4.1. CONSTRUCCIÓN

#### **Requisitos Técnicos:**

- Herramientas: Python, SQL Server, Flask.

- Librerías: `pyautogui`, `selenium`, `Flask`, `pandas`, `statsmodels`, `prophet`, entre otras.
- Entorno web: API en Flask visualizando datos procesados.

### Desarrollo de Módulos

- **Automatización de la Recolección de Datos:**

- Uso de `pyautoguia` para navegar por la página de la Bolsa de Valores de Colombia (BVC) y extraer datos relevantes (precios, volúmenes, etc.).
- Validación de datos extraídos mediante pruebas de consistencia.

- **Almacenamiento en Base de Datos:**

- Uso de **SQL Server** para estructurar y almacenar los datos extraídos.
- Creación de funciones en el módulo Conexion para consultas y actualizaciones en la base de datos.

- **Análisis de Datos:**

- Implementación de modelos ARIMA y Prophet para predicciones de tendencias futuras.
- Uso de `IsolationForest` para detectar anomalías en las transacciones financieras.
- Descomposición estacional con `seasonal_decompose` para entender patrones estacionales.

- **Visualización:**

- Creación de gráficos con `matplotlib`, `mplfinance` y `seaborn`.
- Generación de gráficos en tiempo real para mostrar predicciones y análisis de tendencias.

## 4.2. PRUEBA

### Pruebas Unitarias

- Cada módulo fue probado de manera independiente:

- **Automatización:** Verificación de que pyautoguia extrae datos correctamente bajo distintas configuraciones de red y cambios en la página web.
- **Base de Datos:** Validación de integridad y precisión en el almacenamiento de datos.
- **Modelos Predictivos:** Comparación de predicciones contra datos históricos para evaluar el rendimiento.

### Pruebas de Integración

- Se integraron todos los componentes (automatización, análisis y visualización) para probar el flujo completo del sistema.

### Pruebas de Desempeño

- Evaluación del tiempo requerido para extraer datos, almacenarlos, analizarlos y visualizar los resultados.
- Optimización del rendimiento al ajustar consultas SQL y simplificar cálculos.

### Pruebas de Usuario

- Se presentaron los resultados a analistas financieros para obtener retroalimentación sobre la utilidad y claridad de las visualizaciones.

## 4.3. IMPLEMENTACIÓN

### Implementación Técnica

- **Configuración del Entorno:**
  - Instalación de dependencias con pip y configuración de Python 3.10+.
  - Configuración de SQL Server para admitir conexiones remotas.
- **Carga Inicial de Datos:**
  - Ejecución del RPA para realizar la extracción inicial de datos de la página de la BVC.
  - Carga de datos históricos en la base de datos para análisis retrospectivo.

### Pruebas Post-Implementación

- Ejecución del sistema en condiciones reales durante varios días.
- Monitoreo de errores y ajustes a los modelos predictivos basados en el comportamiento observado.

## 4.4. DESPLIEGUE

### Despliegue Local

- **API Flask:**
  - Configuración de un servidor local para servir los datos y gráficos.
  - Uso de localhost para facilitar el acceso inmediato a los usuarios.
- **Visualización:**
  - Integración de gráficos en plantillas HTML usando `render_template`.
  - Presentación interactiva de datos financieros procesados.

### Documentación

- Se creó una guía de usuario que incluye:
  - Instrucciones para ejecutar el sistema.
  - Explicación de cada gráfico y métrica.
  - Resolución de problemas comunes.

### Plan de Mantenimiento

- Establecimiento de un cronograma para actualizar los modelos predictivos y realizar ajustes en la automatización cuando la página web de la BVC cambie su estructura.
- Monitoreo continuo del rendimiento y la precisión del sistema.

## 4.5. Medidas de bondad del modelado

Las medidas de bondad del modelado son métricas que evalúan el desempeño de un modelo predictivo en relación con los datos reales. Estas medidas permiten cuantificar qué tan bien el modelo ha capturado las relaciones en los datos y, por lo tanto, cuán útil será en la predicción de nuevos casos. A continuación, se presentan las principales métricas utilizadas para evaluar la bondad del modelado:

**R<sup>2</sup> (Coeficiente de Determinación):**

El R<sup>2</sup> es una medida que indica la proporción de la varianza de la variable dependiente que es explicada por las variables independientes en el modelo. Un valor de R<sup>2</sup> cercano a 1 sugiere que el modelo explica casi toda la variabilidad de los datos, lo que implica un alto grado de ajuste. En el contexto de tu modelo, un R<sup>2</sup> de 0.9827 significa que el 98.27% de la variabilidad de la variable objetivo es capturada por el modelo, lo que refleja una excelente capacidad predictiva y sugiere que el modelo es muy efectivo en la representación de los datos subyacentes.

**MAE (Error Absoluto Medio):**

El MAE proporciona la magnitud promedio de los errores en las predicciones sin considerar su dirección (es decir, si el error es positivo o negativo). Es una medida directa de la precisión del modelo, ya que muestra cuán lejos, en promedio, están las predicciones de los valores reales. Un MAE bajo, como el 9.22 obtenido en tu caso, indica que el modelo está realizando predicciones bastante precisas, con un margen de error pequeño.

**MSE (Error Cuadrático Medio) y RMSE (Raíz del Error Cuadrático Medio):**

El MSE es el promedio de los errores al cuadrado, lo que les da más peso a los errores grandes que a los pequeños, haciéndolo útil para identificar modelos que pueden estar fallando en algunos casos específicos. El RMSE, siendo la raíz cuadrada del MSE, devuelve el error a la misma escala que la variable objetivo, facilitando su interpretación. Un RMSE de 12.22 en tu modelo sugiere que, en promedio, las predicciones difieren de los valores reales en 12.22 unidades, lo que refuerza la idea de un modelo con buena precisión general.

## 4.6. Ejemplo de aplicación

Un ejemplo práctico de cómo aplicar un modelo de regresión lineal a un conjunto de datos financieros para predecir el "Precio cierre" de un activo, como una acción de una empresa, utilizando Python y bibliotecas de ciencia de datos como pandas y scikit-learn. El proceso comienza con la consulta y preparación de los datos. En este paso, los datos se recuperan de una base de datos SQL Server y se convierten en un DataFrame de pandas para su manipulación. Este DataFrame contiene columnas relevantes como "Precio cierre", "Precio máximo", "Precio promedio ponderado", "Precio mínimo", "Variación absoluta", "Variación porcentual", "Cantidad", y "Volumen", que se utilizarán para el modelado.

Después de preparar los datos, se realiza una conversión de los tipos de datos. Se aplica una función lambda personalizada para convertir los valores que originalmente son cadenas con comas a flotantes (float). Esto es crucial en análisis financieros, donde los datos frecuentemente incluyen separadores de miles. Asegurarse de que todos los datos numéricos estén en el formato correcto permite evitar errores durante el proceso de modelado.

El siguiente paso es la limpieza de datos. Se eliminan las filas con valores faltantes (NaN) después de la conversión. Esto garantiza que el conjunto de datos esté libre de inconsistencias, lo cual es esencial para que el modelo pueda ser entrenado de manera efectiva. Posteriormente, se definen las variables independientes (features) y dependientes (target). Las variables independientes son las que el modelo utilizará para hacer predicciones, mientras que la variable dependiente es lo que el modelo intentará predecir. En este caso, se está tratando de predecir el "Precio cierre" usando otras características del mercado financiero.

Con los datos limpios y preparados, se entrena el modelo utilizando un modelo de regresión lineal (LinearRegression). El método fit entrena el modelo, encontrando la relación lineal óptima entre las características y la variable objetivo. Tras el entrenamiento, se hacen predicciones y se calculan varias métricas de rendimiento para evaluar la calidad del modelo:  $R^2$  (Coeficiente de Determinación), MAE (Error Absoluto Medio), MSE (Error Cuadrático Medio), RMSE (Raíz del Error Cuadrático Medio), y MAPE (Error Porcentual Absoluto Medio). Estas métricas permiten evaluar la precisión y fiabilidad del modelo.

Este ejemplo tiene una aplicación directa en el análisis de mercados financieros, donde es crucial prever los movimientos de precios. Por ejemplo, en el caso de predecir el precio de cierre de acciones como ECOPETROL, las métricas calculadas permiten evaluar si el modelo es suficientemente preciso para ser implementado en un entorno de producción. El modelo puede ser utilizado para tomar decisiones de inversión, al predecir los precios futuros basados en datos históricos. Con un  $R^2$  alto y bajos valores de MAE y RMSE, el modelo demuestra su eficacia para hacer predicciones precisas, lo que podría traducirse en recomendaciones

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost valiosas para traders, analistas financieros, o sistemas automatizados de trading. De este modo, la empresa puede integrar el modelo en su sistema de toma de decisiones y, con un monitoreo continuo, optimizar su estrategia en función de las predicciones del modelo.

## 5. Cronograma del proyecto

### 5.1. Inteligencia de negocio

Inteligencia de Negocio para este proyecto, se establece un enfoque integral para entender y satisfacer las necesidades analíticas relacionadas con la extracción y análisis de datos de Ecopetrol desde la Bolsa de Valores de Colombia. La primera fase se enfoca en el análisis de requerimientos, donde se identifican las necesidades específicas de los usuarios finales del sistema, incluyendo la definición de qué variables y métricas son cruciales para la toma de decisiones. Se realizan reuniones con stakeholders clave para documentar estos requerimientos. En la fase de diseño de soluciones de BI, se desarrollan modelos y visualizaciones que mostrarán los datos analizados en la API web, asegurando que la presentación sea clara y útil para los usuarios. Esta etapa culmina con la revisión y aprobación de los diseños por parte de los usuarios y el equipo de desarrollo. Posteriormente, en la fase de desarrollo e implementación, se crean y ajustan los informes y dashboards en la API web para reflejar los resultados del análisis en tiempo real. Finalmente, se lleva a cabo una fase de capacitación y soporte para los usuarios, proporcionando entrenamiento sobre cómo interpretar los datos y cómo utilizar la API para su análisis, además de ofrecer soporte post-implementación para resolver cualquier inconveniente que surja

### 5.2. Datos y bases de datos

Datos y Bases de Datos en este proyecto aborda la recolección, almacenamiento y gestión de los datos extraídos de la página web de Ecopetrol. En la primera fase, se realiza un análisis y preparación de datos, que incluye la recolección de datos desde la web de Ecopetrol, la limpieza y transformación de estos datos para que se ajusten a los requisitos de la base de datos SQL. Se crea la estructura necesaria en la base de datos para almacenar la información de manera eficiente. En la fase de diseño de la base de datos, se modela la estructura de la base de datos SQL, diseñando las tablas y relaciones necesarias para manejar los datos extraídos. En la fase de implementación y carga de datos, se implementa la base de datos SQL

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost y se realiza la carga inicial de datos extraídos, verificando su integridad y precisión. Finalmente, en la fase de mantenimiento y optimización, se realiza un monitoreo continuo del rendimiento de la base de datos, asegurando que se mantenga eficiente y se actualice según las necesidades cambiantes del proyecto.

### 5.3. Integración de datos

Integración de Datos se centra en la sincronización entre el sistema RPA, la base de datos SQL y la API web. En la primera fase, se identifican las fuentes de datos externas, en este caso, la página web de Ecopetrol, y se asegura la conectividad y accesibilidad para el RPA. En la fase de diseño de procesos de integración, se desarrollan los procesos ETL (Extract, Transform, Load) necesarios para que el RPA recoja los datos, los transforme según los requerimientos y los cargue en la base de datos SQL. En la fase de implementación y pruebas, se ejecutan estos procesos para garantizar que los datos se integren correctamente en la base de datos y se visualicen de manera precisa en la API web. Finalmente, en la fase de monitoreo y mantenimiento, se supervisan los procesos de integración para asegurar un flujo de datos continuo y se resuelven problemas que puedan surgir, garantizando que la información en la API web se mantenga actualizada y precisa.

### 5.4. Infraestructura

Infraestructura cubre todos los aspectos técnicos necesarios para soportar el sistema RPA, la base de datos SQL y la API web. En la fase de diseño de infraestructura, se desarrolla la arquitectura tecnológica, seleccionando los servidores y redes adecuados para alojar el RPA, la base de datos SQL y la API web. En la fase de implementación de infraestructura, se configuran los servidores y redes, se instala el software necesario y se preparan los entornos para el RPA y la API web. Durante la fase de pruebas y ajustes, se realizan pruebas de rendimiento y seguridad para garantizar que la infraestructura soporte el sistema de manera efectiva y segura. Finalmente, en la fase de mantenimiento y actualización, se realiza el mantenimiento continuo de la infraestructura, asegurando que se mantenga operativa y actualizada, con la implementación de mejoras según sea necesario.

### 5.5. Gestión del proyecto

Gestión del Proyecto aborda la planificación y supervisión de todas las actividades relacionadas con la implementación del sistema RPA. En la fase de planificación, se definen

los objetivos del proyecto, el alcance y se desarrolla un plan detallado con un cronograma de actividades que incluyen el desarrollo del RPA, la configuración de la base de datos y la implementación de la API web. La fase de ejecución se centra en la coordinación de tareas y recursos, garantizando que cada componente del sistema se desarrolle según lo previsto y en tiempo. Durante la fase de control, se gestionan los riesgos y problemas, se controla la calidad y se supervisa el presupuesto para asegurar que el proyecto se mantenga dentro de los límites establecidos. Finalmente, en la fase de cierre, se realiza una evaluación final del proyecto, se entrega el sistema completo y operativo, y se completa la documentación administrativa necesaria para cerrar el proyecto de manera formal, incluyendo la recopilación de lecciones aprendidas y la preparación para el soporte post-implementación.

## 6. Conclusiones

La propuesta de implementación del sistema RPA para la extracción de datos de Ecopetrol desde la Bolsa de Valores de Colombia podría ser una herramienta eficaz para optimizar la eficiencia operativa en la gestión de información financiera. El uso de RPA eliminaría potencialmente la necesidad de realizar tareas repetitivas y propensas a errores, como la recolección manual de datos, permitiendo que estas actividades se ejecuten de manera automática y sin intervención humana. Este enfoque no solo agilizaría el flujo de trabajo, sino que también reduciría significativamente el riesgo de errores y la variabilidad en la calidad de los datos recolectados, aspectos esenciales en un entorno financiero donde la precisión y la consistencia son fundamentales. La reducción de la intervención humana liberaría recursos que podrían ser redirigidos a actividades de análisis y toma de decisiones estratégicas.

Uno de los aspectos más destacables del sistema propuesto sería su capacidad para procesar y analizar datos financieros en tiempo real, proporcionando a los tomadores de decisiones de Ecopetrol acceso inmediato a información relevante sobre las fluctuaciones del mercado. En el sector financiero, la velocidad con la que se puede acceder y analizar los datos es crucial, ya que los mercados reaccionan rápidamente a eventos externos y factores económicos. Al ofrecer una visualización en tiempo real a través de una API web, este sistema permitiría una respuesta inmediata a estos cambios, facilitando decisiones oportunas y bien fundamentadas.

La integración planteada entre el sistema RPA, la base de datos SQL y la API web en un entorno local representaría una arquitectura eficiente y adaptable para el análisis financiero. Este

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost  
modello de integración facilitaría una cadena de trabajo en la que los datos extraídos automáticamente por el RPA serían almacenados y organizados en una base de datos estructurada, lo que garantizaría una alta disponibilidad y consistencia de la información. La capacidad de visualizar estos datos en una API web permitiría que los analistas y otros usuarios accedan de forma sencilla a los resultados del análisis en tiempo real.

La precisión de los datos es crucial en cualquier tipo de análisis financiero, y la automatización mediante RPA propuesta disminuiría significativamente la probabilidad de errores humanos durante el proceso de recopilación de datos. El acceso a datos en tiempo real reduciría los riesgos asociados a la toma de decisiones basadas en información desactualizada o incompleta. En un entorno de inversión tan sensible como el mercado de valores, el acceso a información precisa y actualizada sería una ventaja estratégica que permitiría gestionar riesgos de manera más efectiva.

La implementación de este proyecto abriría la puerta a futuras mejoras y ampliaciones del sistema. En un contexto de cambio constante en el mercado financiero, este sistema podría beneficiarse de la implementación de algoritmos de aprendizaje automático que detecten patrones y tendencias complejas en los datos históricos. También se podrían explorar otras fuentes de datos externas, como noticias financieras y redes sociales, para enriquecer el análisis y hacer predicciones más precisas. La infraestructura propuesta podría ser escalada a un entorno de producción que permita a la empresa implementar el sistema a nivel corporativo y en múltiples entornos de análisis, maximizando el valor potencial del sistema y su impacto organizacional.

## 6.1. LIMITACIONES ANALÍTICAS

- Modelos predictivos básicos: Los análisis actuales se basan principalmente en regresión lineal y estadísticas descriptivas, sin incorporar modelos más sofisticados de machine learning.
- Ausencia de análisis de sentimiento: El sistema no incorpora datos externos como noticias o redes sociales que podrían afectar el comportamiento del mercado.
- Análisis unifactorial: Las predicciones se basan principalmente en datos históricos de precios, sin considerar otros factores macroeconómicos o del sector petrolero.

## 6.2. LIMITACIONES OPERATIVAS

- Mantenimiento manual: El sistema requiere supervisión regular para asegurar su funcionamiento correcto.
- Ausencia de alertas avanzadas: No se incluyen sistemas de alertas automáticas para eventos significativos o anomalías en los datos.
- Documentación limitada: La documentación actual podría no ser suficiente para usuarios no técnicos.

## 6.3. PROSPECTIVA Y OPORTUNIDADES DE MEJORA

### 6.3.1. Mejoras técnicas propuestas

- Implementación de arquitectura distribuida: Desarrollar una arquitectura que permita el procesamiento paralelo de datos para mejorar el rendimiento.
- Desarrollo de una interfaz más robusta: Crear una interfaz de usuario más completa con mayores opciones de personalización.
- Automatización mejorada: Implementar sistemas de recuperación ante fallos y manejo de excepciones más sofisticados.

### 6.3.2. Ampliaciones analíticas

- Incorporación de modelos avanzados de machine learning:
  - Redes neuronales para predicción de precios
  - Algoritmos de clustering para identificar patrones de mercado
  - Modelos de análisis de sentimiento para noticias financieras
- Integración de análisis multifactorial:
  - Inclusión de variables macroeconómicas
  - Análisis de correlación con otros activos del mercado
  - Incorporación de indicadores del sector petrolero
- Desarrollo de nuevas métricas y visualizaciones:
  - Indicadores técnicos avanzados

- Visualizaciones interactivas más sofisticadas
- Dashboards personalizables según las necesidades del usuario

### 6.3.3. Proyección futura

- Desarrollo de capacidades predictivas avanzadas:
  - Modelos de predicción a largo plazo
  - Análisis de escenarios y stress testing
  - Sistemas de alerta temprana basados en múltiples factores

## Referencias bibliográficas

- Anagnoste, S. (2022). Robotic Process Automation in Financial Services: Current Applications and Future Trends. *Journal of Risk and Financial Management*, 15(3), 123-145.
- Bolsa de Valores de Colombia. (2024). Obtenido de ECOPETROL S.A.: <https://www.bvc.com.co/renta-variable-mercado-local/ECOPETROL?tab=informacion-del-emisor>
- Chen, Y., & Wang, W. (2023). The Impact of RPA Implementation on Financial Data Processing Efficiency: A Case Study Approach. *International Journal of Information Management*, 65, 102-542.
- García, R., & Martinez, C. (2022). Real-time Financial Data Integration: Challenges and Solutions. *International Journal of Database Theory and Application*, 15(3), 67-84.
- Khan, A., Shah, S., & Lee, D. (2021). Web-Based APIs for Real-Time Data Visualization in Financial Applications. *International Journal of Information Management*, 58, 102-114.
- Lee, M., Huang, Y., & Zeng, Z. (2020). Robotic Process Automation in Financial Services: A Systematic Review. *Journal of Financial Services Research*, 58(2), 123-145.
- Mitchell , T. (2021). The Impact of Robotic Process Automation on Data Accuracy in Financial Services. *Financial Data Science Review*, 12(1), 34-46.
- Roh, J., Choi, J., & Kim, K. (2019). Real-Time Data Analysis for Financial Decision-Making: Techniques and Applications. *IEEE Access*, 7, 143678-143686.

Michel Enrique Abello Betancourt

Implementación de un RPA para la Extracción y Análisis en Tiempo Real de Datos de Ecopetrol desde la Bolsa de Valores de Colombia: Integración con SQL y Visualización Web en Localhost

Zhang, L., & Chen, X. (2023). **Machine Learning Applications in Financial Market Analysis: A Comprehensive Review.** *Journal of Big Data Analytics in Finance*, 1(2), 45-67.