

# Extracción, Enriquecimiento y Análisis Predictivo de Datos Financieros de MercadoLibre (MELI)

---

## Resumen

Este trabajo implementa una solución automatizada para la extracción, enriquecimiento y análisis de datos históricos del precio de las acciones de MercadoLibre S.A. (MELI), utilizando Web Scraping desde Yahoo Finance. Se procesan los datos para generar indicadores financieros (KPIs) como volatilidad, tasa de variación y retorno acumulado. Además, se entrena un modelo predictivo con Random Forest y se realiza un análisis de series temporales con técnicas estadísticas como la descomposición estacional, el test ADF y el modelo ARIMA. Todo el flujo se automatiza con GitHub Actions y se registra mediante archivos de log. Esta solución integra herramientas de ciencia de datos aplicadas al análisis financiero.

## Introducción

En la actualidad, el análisis de datos financieros es clave para comprender la dinámica de los mercados. Si bien existen plataformas como Yahoo Finance que ofrecen información bursátil gratuita, su acceso está limitado para automatizaciones, ya que no cuentan con una API abierta. Ante esta limitación, se propone una solución basada en Web Scraping, análisis exploratorio, modelado de series temporales, y automatización con CI/CD, utilizando Python como lenguaje principal. El enfoque es práctico, modular y orientado a su aplicación en escenarios reales de análisis de precios de acciones.

## Metodología

La arquitectura del proyecto sigue un enfoque modular, dividiendo el sistema en las siguientes etapas:

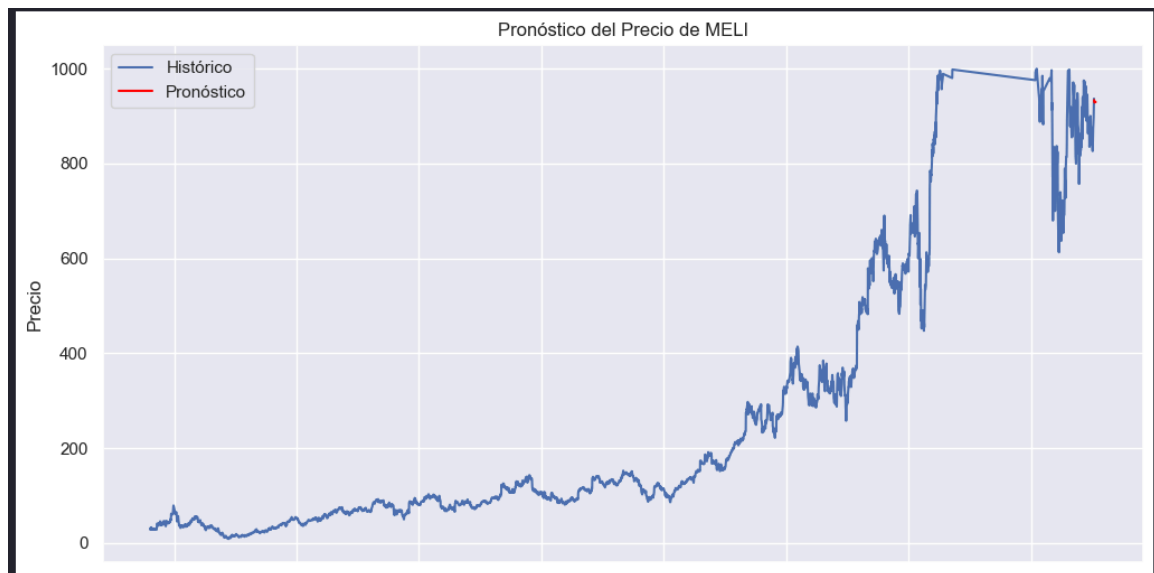
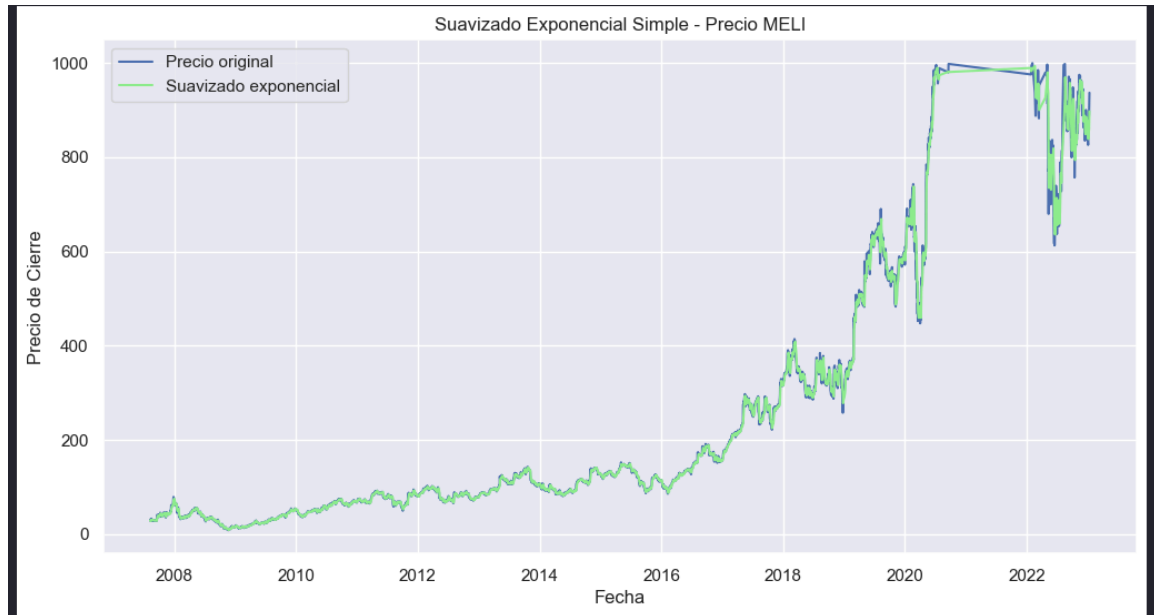
1. **Recolección de Datos (Collector):** Se accede a la tabla de precios históricos de Yahoo Finance mediante la librería requests con cabeceras simulando un navegador. El HTML es procesado con BeautifulSoup para extraer los datos, que luego son convertidos a un DataFrame de pandas. Finalmente, se guardan los datos originales como meli\_data.csv y también como tabla SQLite en historical\_meli.db.
2. **Enriquecimiento de Datos (Enricher):** Los datos recolectados son transformados para generar nuevos indicadores o KPIs, incluyendo:

- Volatilidad: desviación estándar móvil de 5 días.
  - Tasa de variación diaria.
  - Media móvil de 7 días.
  - Retorno acumulado.
  - Desviación total histórica.
3. Modelado Predictivo (Modeller): El modelo de predicción usa un RandomForestRegressor de scikit-learn y se entrena utilizando tanto los KPIs como variables temporales (año, mes, día, día de la semana). El modelo es serializado y guardado como modelo\_dolar.pkl. A partir de la última fila disponible, se predice el valor de cierre del siguiente día hábil.
4. Análisis de Series Temporales: Con la ayuda de statsmodels, se realiza:
- Descomposición estacional (tendencia, estacionalidad, ruido).
  - Prueba de Dickey-Fuller aumentada (ADF) para evaluar la estacionariedad.
  - Gráficas de autocorrelación (ACF) y autocorrelación parcial (PACF).
  - Modelado ARIMA para pronóstico a corto plazo (10 días), visualizado junto a los datos históricos.
5. Registro de Eventos (Logger): Se creó un sistema de logging personalizado que guarda los eventos en archivos .log dentro de la carpeta logs/, con información de clase, función y mensaje, lo que facilita la depuración y trazabilidad.
6. Automatización con GitHub Actions: Se definió un flujo de trabajo (update\_data.yml) que:
- Se activa automáticamente con cada push a main.
  - Crea un entorno virtual.
  - Instala todas las dependencias del proyecto vía setup.py.
  - Ejecuta el script principal main.py.
  - Si se generan cambios (datos, logs), se realiza automáticamente un commit y push.

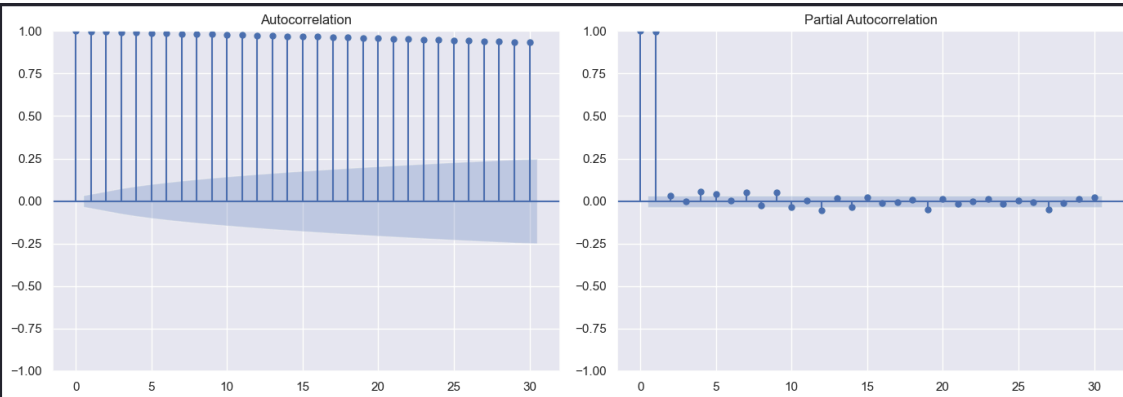
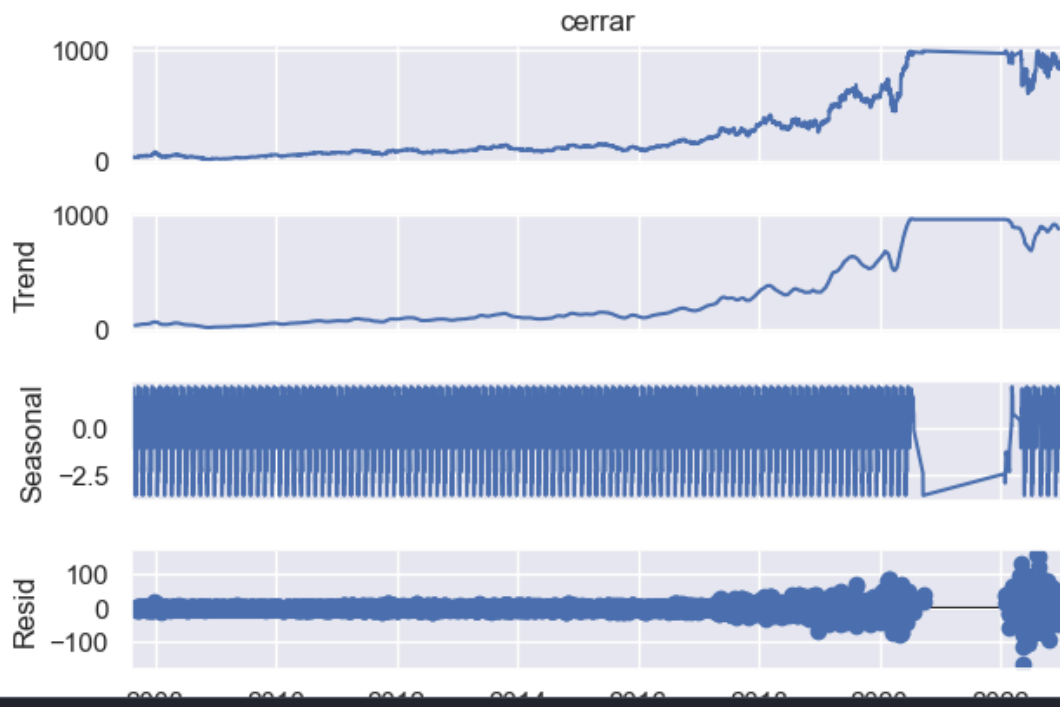
## Resultados

- Se obtuvieron más de 1.000 registros históricos de la acción MELI.
- Se generaron y visualizaron indicadores que permiten análisis más profundos del comportamiento del activo.
- Se entrenó un modelo de predicción con buen desempeño (MSE y  $R^2$  reportados en logs).
- El sistema completo se puede ejecutar localmente o de forma automatizada vía GitHub Actions.

## Graficas



## Descomposición de la Serie Temporal



## **Conclusiones**

El proyecto demuestra que es posible construir un pipeline completo y automatizado para el análisis financiero de activos bursátiles sin depender de APIs comerciales. Se integra scraping web, procesamiento de datos, modelado predictivo y series temporales, todo bajo una arquitectura modular y trazable. La solución es adaptable a otros activos o fuentes de datos, y su integración con CI/CD garantiza una ejecución continua y actualizada.