

Grado en Ingeniería de Sistemas de Inteligencia Artificial

Redes neuronales

Laboratorio 4: Implementación de un LSTM con las
acciones de TSLA



Universidad
Francisco de Vitoria
UFV Madrid

Jacobo Martín

Índice

1. Introducción.....	3
2. Cuestiones	3
2.1. Resultados de la Práctica 1	3
2.2. Resultados de la Práctica 2	6
2.3. Resultados de la Práctica 3	8
3. Bibliografía.....	11

1. Introducción

El presente trabajo documenta el desarrollo de un modelo predictivo basado en redes LSTM para la serie temporal del precio de cierre de las acciones de Tesla (TSLA). Se abordan las etapas clave, desde la obtención y preparación de los datos, entrenamiento y evaluación del modelo recurrente. Se analizan los resultados obtenidos, las métricas de error y se visualiza el rendimiento predictivo del modelo en el conjunto de prueba, destacando tanto sus capacidades como sus limitaciones predictivas.

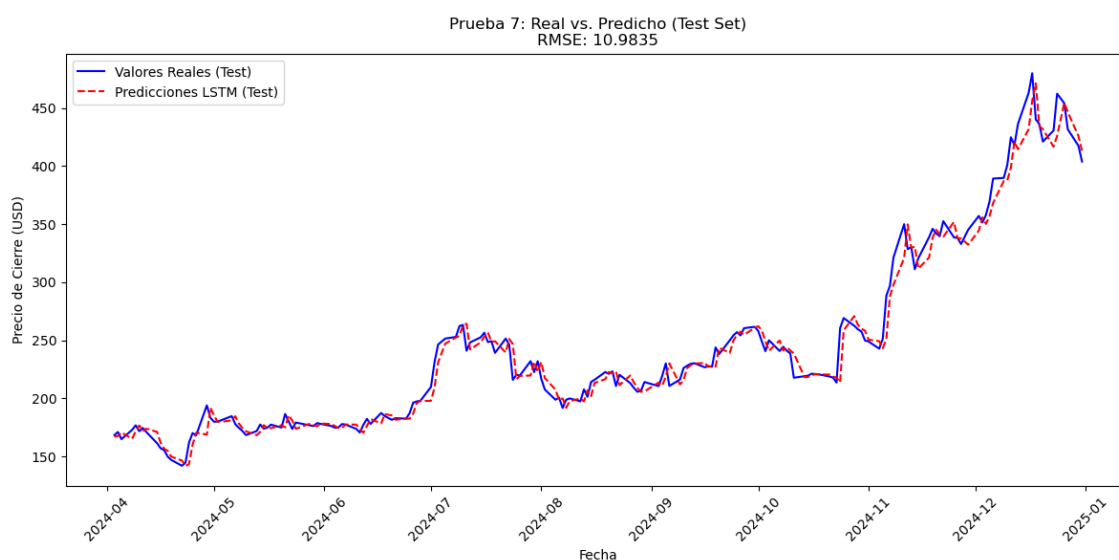


Figura 1. Ejemplo de la predicción del LSTM.

2. Cuestiones

2.1. Resultados de la Práctica 1

Análisis Gráfico de la Serie Temporal

Al observar la *Figura 2* del histórico de precios de cierre de TSLA desde 2020 hasta la fecha actual, podemos notar varias características clave:



Figura 2. Se han utilizado los datos de los últimos 5 años como dataset.

1. Tendencia:

- **No Estacionaria y Cambiante:** La serie es claramente **no estacionaria**. No tiene una media o varianza constante a lo largo del tiempo.
- **Múltiples Fases:** Se observa una fuerte tendencia alcista inicial muy pronunciada durante 2020 y 2021, alcanzando un pico significativo. A esto le sigue una corrección/caída importante en 2022. Posteriormente, parece entrar en una fase de mayor movimiento lateral, pero con fluctuaciones considerables (subidas y bajadas), sin una tendencia dominante clara y sostenida como la inicial. Hacia el final de los datos más recientes, parece haber una tendencia bajista.
- **Complejidad:** La tendencia general no es lineal; es compleja y parece cambiar de régimen a lo largo del periodo.

2. Estacionalidad:

- **No Evidente:** A simple vista, no se aprecian patrones claros y repetitivos que ocurran en intervalos fijos anuales, trimestrales o mensuales. Los picos y valles no parecen alinearse consistentemente con épocas específicas del año a lo largo de los diferentes años. La dinámica parece estar más dominada por la tendencia general y eventos externos que por una estacionalidad intrínseca fuerte.

3. Ruido/Volatilidad:

- **Alta Volatilidad:** La característica más destacada es la **alta volatilidad**. La línea de precios se caracteriza por tener fluctuaciones diarias/semanales significativas alrededor de la tendencia general. Hubo periodos de

volatilidad extrema, especialmente durante la gran subida y la posterior caída.

- **Irregularidad:** Las fluctuaciones no parecen seguir un patrón regular; son bastante erráticas, lo que sugiere la influencia de noticias, sentimiento del mercado y otros factores impredecibles.

Influencia de las Características en el Diseño del Modelo LSTM:

Estas características tienen implicaciones directas en cómo diseñamos y evaluamos el modelo LSTM:

1. Tendencia No Estacionaria:

- **Escalado:** Es **crucial** escalar los datos. Las LSTMs funcionan mejor con datos en un rango pequeño (ej., 0 a 1). El escalado evita que los valores absolutos grandes dominen el proceso de aprendizaje y ayuda a manejar diferentes magnitudes de precios a lo largo del tiempo.
- **Predicción a Largo Plazo:** La naturaleza cambiante de la tendencia hace que la extrapolación a largo sea propensa a errores grandes si la tendencia cambia.

2. Ausencia de Estacionalidad:

- **Diseño:** No necesitamos incorporar explícitamente características de estacionalidad o usar variantes de modelos específicos para estacionalidad. La LSTM, con una `window_size` adecuada, es capaz de capturar patrones temporales con precisión.

3. Alta Volatilidad/Ruido:

- **Window_size:** Es un hiperparámetro clave. Una ventana más larga puede ayudar a suavizar el ruido y capturar tendencias subyacentes, pero puede ser lenta para reaccionar a cambios rápidos. Una ventana corta es más reactiva pero más sensible al ruido. Los mejores resultados se obtuvieron con ventanas de tamaño 20 a 60.
- **Regularización:** El dropout es esencial para evitar que el modelo memorice el ruido específico del conjunto de entrenamiento y mejore su capacidad de generalización. Las tasas óptimas fueron 0.1 y 0.2.
- **Complejidad del Modelo:** Un modelo demasiado complejo podría sobreajustarse al ruido. Uno demasiado simple no capturarán patrones adecuadamente.
- **Predicción Iterativa:** La alta volatilidad es la principal razón por la que las predicciones iterativas a largo plazo se vuelven lisas o divergen. Esto significa que el modelo es más adecuado para predicciones a corto plazo.

2.2. Resultados de la Práctica 2

El parámetro **window_size** en una LSTM define cuántos pasos temporales del pasado se usan como entrada y afecta directamente la capacidad del modelo para capturar dependencias temporales. Un tamaño mayor permite aprender patrones a largo plazo pero incrementa la complejidad, el riesgo de sobreajuste y el costo computacional. Si es demasiado corto, puede no proporcionar suficiente contexto, lo que limita la capacidad predictiva. Además, tamaños grandes pueden incluir ruido irrelevante, afectando la generalización, mientras que tamaños pequeños pueden hacer que el modelo aprenda solo patrones locales.

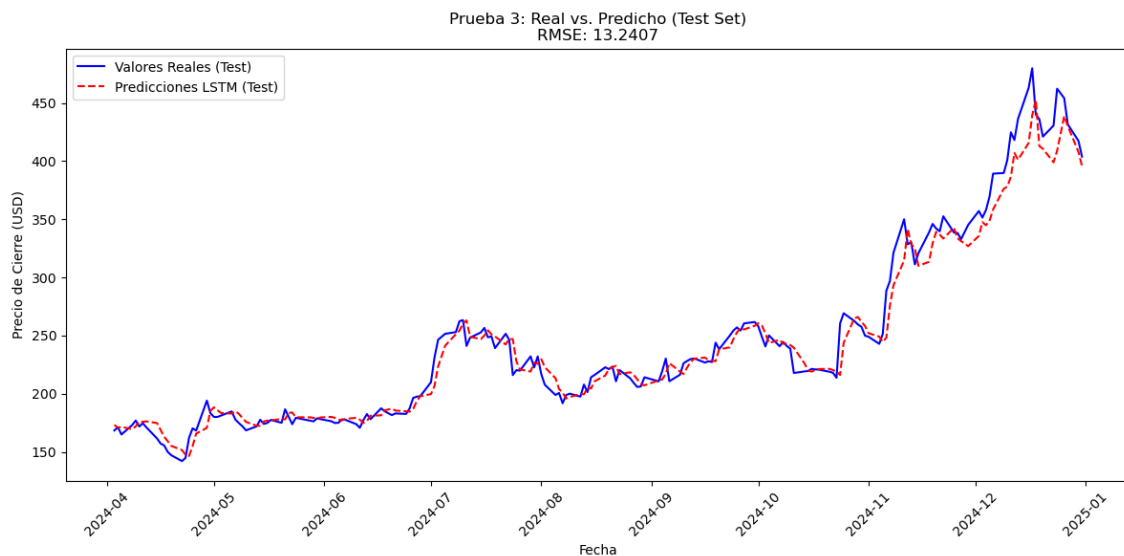


Figura 3. Predicción con un Window Size de 60.

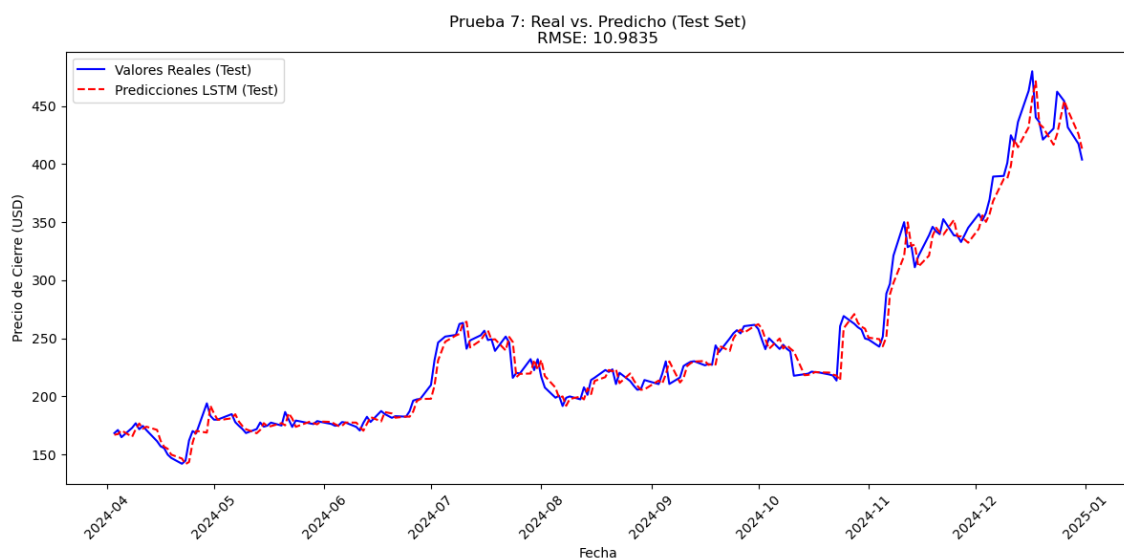


Figura 4. Predicción con un Window Size de 40.

Se realizó un proceso de búsqueda automatizada de hiperparámetros mediante GridSearch con el objetivo de optimizar el rendimiento del modelo LSTM. A continuación, se presentan los resultados obtenidos para las configuraciones evaluadas.

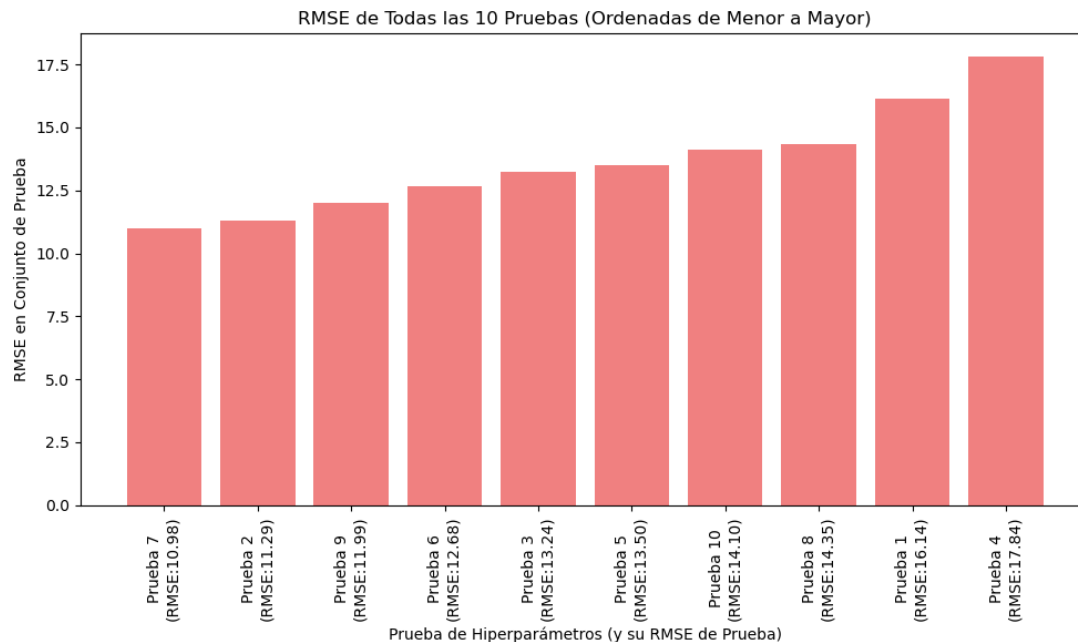


Figura 5. Estas son 10 de todas las pruebas que se hicieron.

Para saber cómo afecta el tamaño de la ventana se realizó un gráfico de barras que muestra el coeficiente de correlación de Pearson entre cada hiperparámetro y el RMSE, de modo que valores positivos indican que, al aumentar ese parámetro, el RMSE tiende a subir.

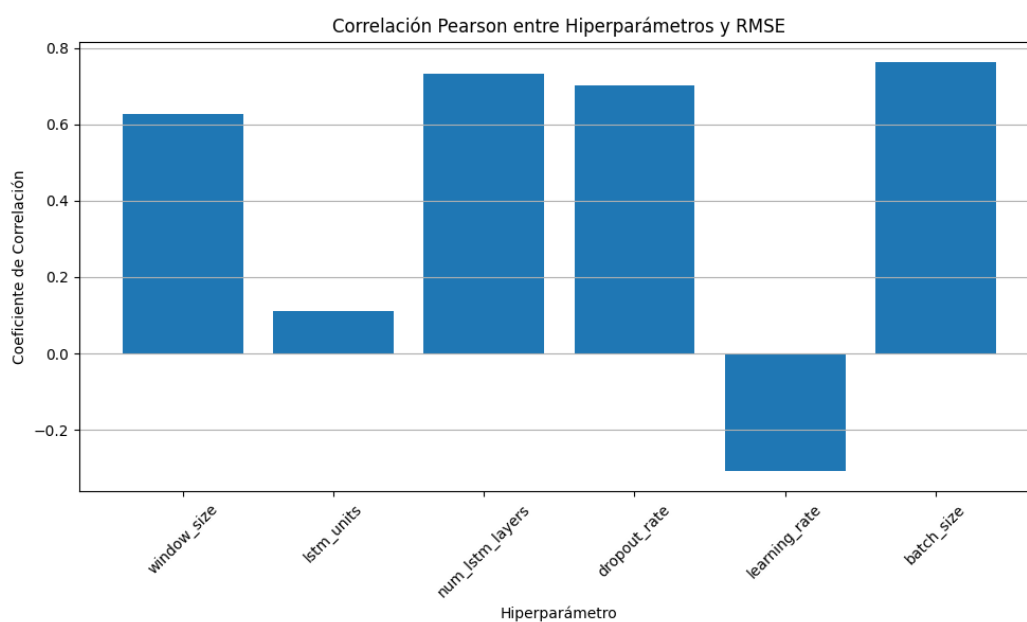


Figura 6. Un window_size mayor indica un aumento del error.

2.3. Resultados de la Práctica 3

Se realizó una predicción a múltiples pasos temporales. En este escenario, se observa que la precisión del modelo tiende a degradarse a medida que aumenta el horizonte de predicción. Esto se debe a la acumulación de errores en cada paso futuro, especialmente cuando el modelo utiliza sus propias predicciones anteriores como entrada en lugar de datos reales, lo que incrementa la incertidumbre y reduce la fiabilidad de las predicciones más lejanas.

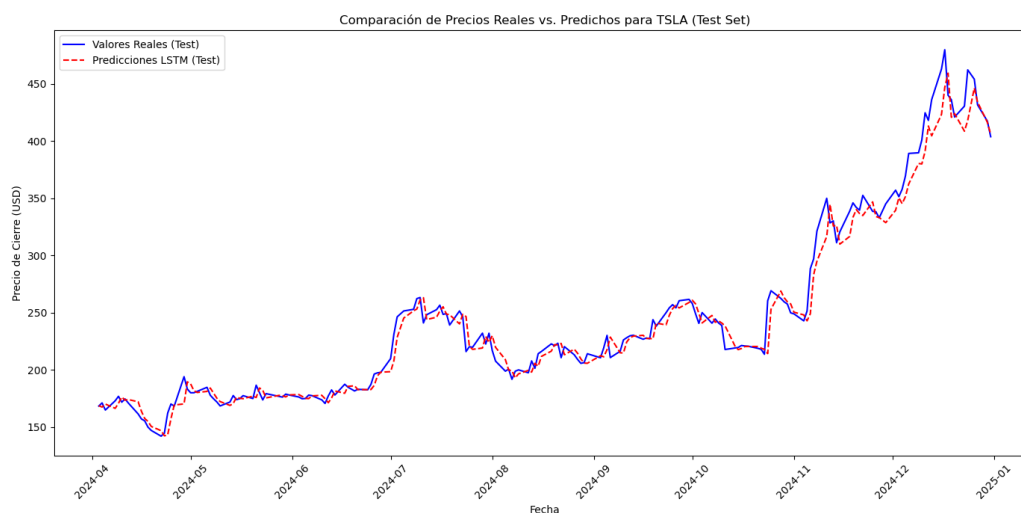


Figura 7. Predicción a futuro realizada con RMSE : 11.9014.

A continuación, se muestra la pérdida de entrenamiento y validación del modelo.

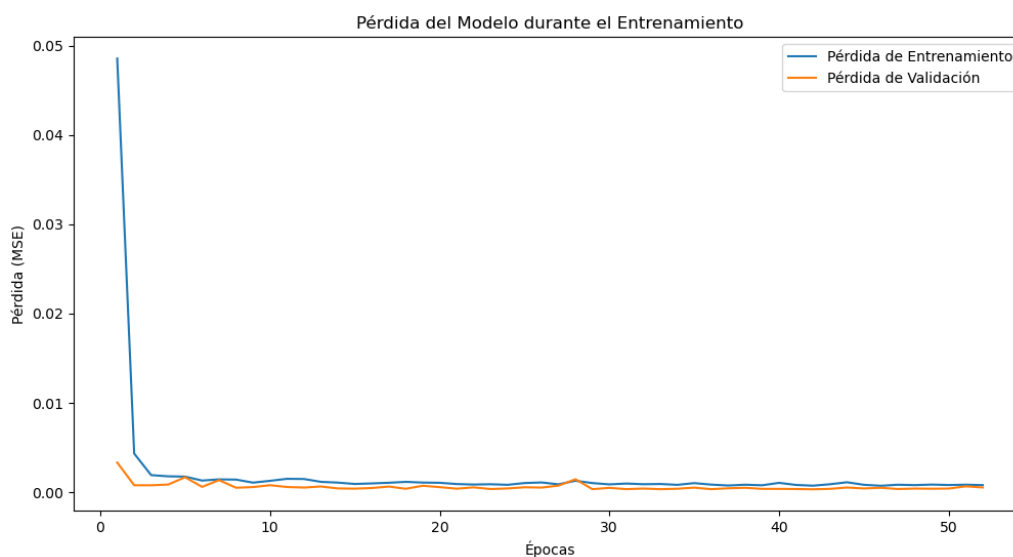


Figura 8. Indica que el modelo está correctamente entrenado.

Para comprobar como se degradaba la capacidad de predicción, se aumentó la cantidad de días a predecir y estos fueron los resultados:

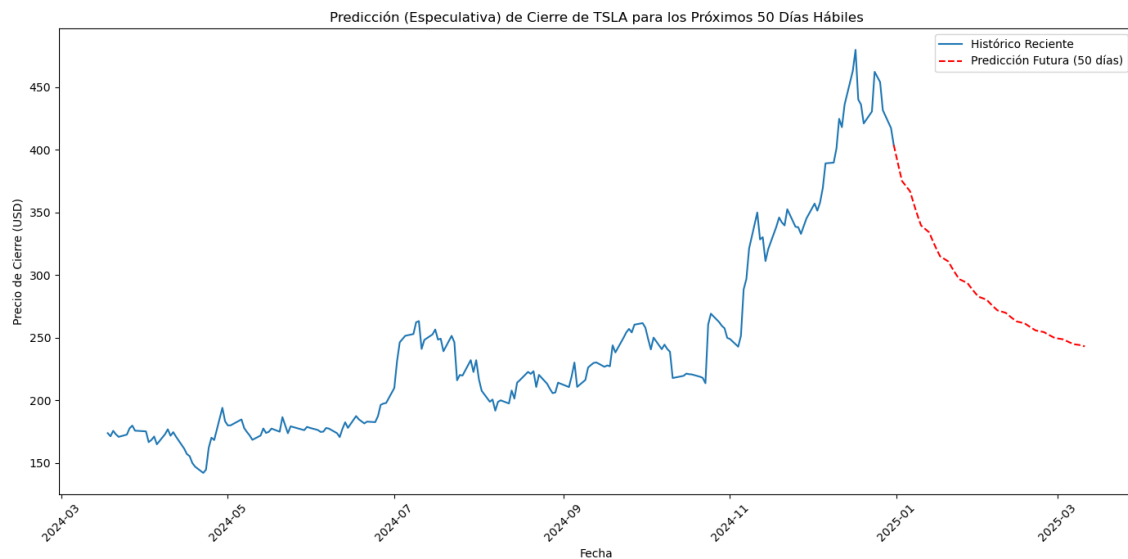


Figura 9. Predicción extendida 50 días.

Se puede observar que tiende a suavizarse lo que indica que el modelo pierde precisión por lo que decidí añadirle un 2% de ruido y comparar con datos reales como se ven en la *Figura 10* y *Figura 11*.

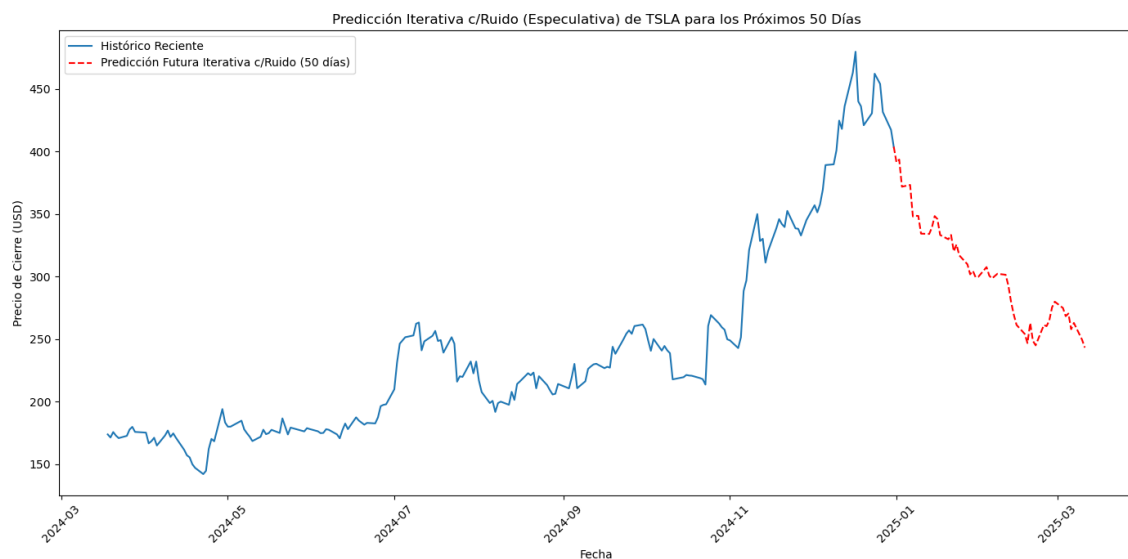


Figura 10. Predicción extendida con un 2% de ruido añadido manualmente.

Se puede observar que, al introducir ruido, las predicciones vuelven a asemejarse a los datos reales de la *Figura 11*.

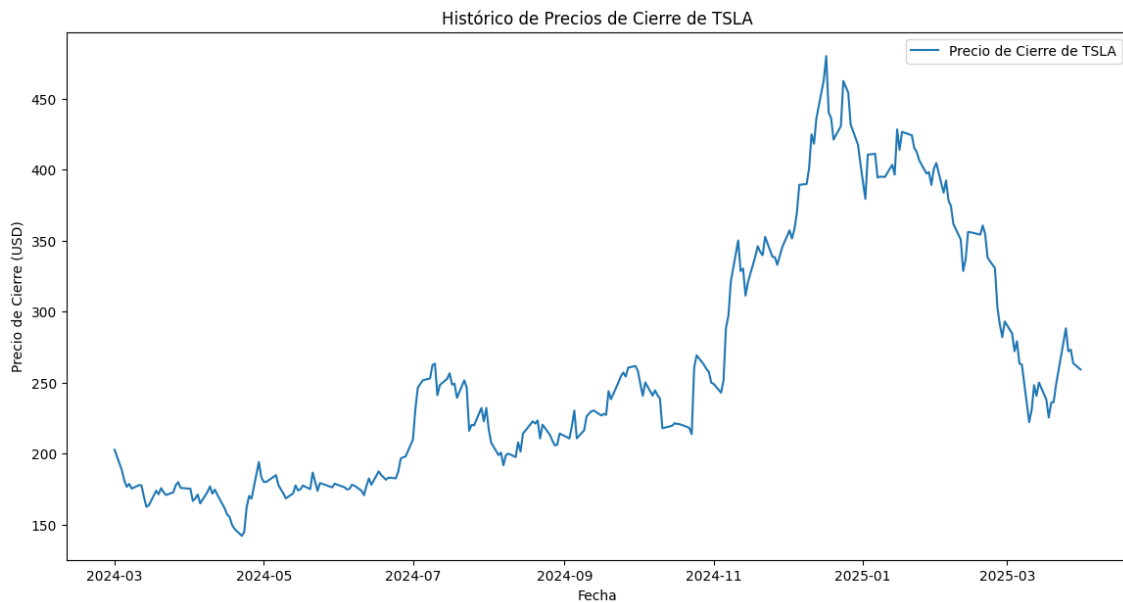


Figura 11. Datos reales.

En segundo lugar, se propondrán diferentes estrategias para mejorar las predicciones a largo plazo:

1. **Modelos Multi-Step Directos:**

- **Estrategia:** En lugar de predecir $t+1$ y luego $t+2$, etc., entrenar un modelo que tome una secuencia de entrada y prediga directamente *toda* la secuencia futura de una sola vez.
- **Ventaja:** Evita la acumulación de errores del método iterativo. El modelo aprende directamente la estructura de dependencia para todo el horizonte.
- **Implementación:** Requiere modificar `create_sequences`, la capa de salida del `LSTMModel`, y reentrenar todo el modelo.

2. **Arquitecturas Más Avanzadas:**

- **Encoder-Decoder con o sin Atención:** Arquitecturas diseñadas para mapear una secuencia de entrada a una secuencia de salida de longitud potencialmente diferente. Puede ser más robusto para dependencias largas.
- **Transformers:** Usan auto-atención para ponderar la importancia de todos los puntos pasados al predecir el futuro. Son computacionalmente más costosos, pero pueden capturar patrones complejos.

3. **Incorporar Más Características:**

- **Estrategia:** El precio de cierre por sí solo tiene información limitada. Añadir otras variables puede dar más contexto al modelo.

- **Ejemplos:** Volumen de transacciones, precios Open/High/Low, Indicadores Técnicos, datos macroeconómicos o análisis de sentimiento.

4. Modelos Híbridos:

- **Estrategia:** Combinar las fortalezas de diferentes tipos de modelos.
- **Ejemplos:**
 - CNN-LSTM: Usar una CNN para extraer características locales de la secuencia de entrada antes de pasarla a la LSTM.
 - LSTM-ARIMA: Usar ARIMA para modelar la tendencia/estacionalidad lineal y LSTM para los residuos no lineales.

5. Enfoques Probabilísticos:

- **Estrategia:** En lugar de predecir un único valor, predecir una distribución de probabilidad para cada paso futuro. Esto captura la incertidumbre inherente.
- **Ejemplos:** LSTMs Bayesianas, DeepAR.

3. Bibliografía

Contenido de la Asignatura de Redes Neuronales. (s.f.). UFV.

OpenAI. (s.f.). *ChatGPT*. <https://chatgpt.com>.