

# Taller: Análisis de Sentimiento en Tweets con RNN, LSTM y BiLSTM + Atención

## 1 Objetivo

Los estudiantes implementarán modelos de redes neuronales recurrentes (RNN, LSTM y BiLSTM con mecanismo de atención) para resolver un problema de análisis de sentimiento. Deberán estructurar un repositorio organizado, con código modular y documentación clara, donde comparen el desempeño de los modelos y reflexionen sobre su aplicabilidad.

## 2 Problemática

Dado un conjunto de datos con tweets etiquetados con sentimiento, se requiere predecir si el sentimiento expresado es positivo o negativo. Esta tarea es clave en aplicaciones como monitoreo de redes sociales, análisis de opinión pública y sistemas de recomendación.

## 3 Datos Utilizados

Este taller utilizará un conjunto de datos de tweets en inglés, disponibles públicamente. El dataset será descargado directamente desde una URL utilizando herramientas de Python, por lo que **no es necesario subir archivos manualmente** ni incluir los datos en el repositorio.

### 3.1 Origen del Dataset

El dataset será cargado desde el siguiente recurso:

- URL: [https://raw.githubusercontent.com/dd2405/Twitter\\_Sentiment\\_Analysis/master/train.csv](https://raw.githubusercontent.com/dd2405/Twitter_Sentiment_Analysis/master/train.csv)
- Contiene más de 30,000 tweets con etiquetas de sentimiento (0 = negativo, 1 = positivo).

### 3.2 Carga recomendada

```
import pandas as pd
import tensorflow as tf

url = "https://raw.githubusercontent.com/dD2405/
      Twitter_Sentiment_Analysis/master/train.csv"
csv_path = tf.keras.utils.get_file("twitter_sentiment.csv",
                                     url)

df = pd.read_csv(csv_path)
df[['tweet', 'label']].head()
```

## 4 Estructura del Taller

### 4.1 Introducción a la problemática

- Aplicaciones del análisis de sentimiento en redes sociales.
- Importancia de modelar secuencias y contexto lingüístico.

### 4.2 Exploración y preprocesamiento del dataset

- Limpieza del texto: eliminación de menciones, hashtags, símbolos y URLs.
- Tokenización.
- División en conjuntos de entrenamiento y prueba.

### 4.3 Implementación de modelos

Escoja uno de los siguientes:

- RNN básico.
- LSTM estándar.
- BiLSTM con mecanismo de atención.

### 4.4 Comparación de resultados

- Evaluar con métricas como accuracy, precision, recall y F1-score.

## 4.5 Estructuración del repositorio

```
mi_proyecto_sentimiento/  
  notebooks/  
    01_exploracion.ipynb  
    02_entrenamiento_RNN.ipynb  
    03_entrenamiento_LSTM.ipynb  
    04_BiLSTM_atencion.ipynb  
    README.md  
  
  src/  
    data_loader.py  
    model_rnn.py  
    model_lstm.py  
    model_bilstm_attention.py  
    train.py  
    evaluate.py  
    utils.py  
  
  models/  
    rnn_model.h5  
    lstm_model.h5  
    bilstm_attention_model.h5  
    README.md  
  
requirements.txt  
.gitignore  
README.md
```

El archivo `README.md` a nivel raíz debe contener:

- Requisitos y dependencias para ejecutar el proyecto.
- Instrucciones para entrenamiento, evaluación y predicción.
- Breve explicación de cada modelo y su arquitectura.
- Resultados obtenidos con cada modelo y análisis comparativo.

**Nota:** Se espera que el código sea ejecutable de forma reproducible siguiendo las instrucciones del `README.md`.