"Modelado Secuencial y PLN para Problemas del Mundo Real"

Parte A: Preprocesamiento y análisis lingüístico con spaCy

Durante la primera fase del trabajo, se aplicó un pipeline de procesamiento de lenguaje natural (PLN) con **spaCy** orientado a preparar un corpus en español compuesto por mensajes clasificados como *spam* o *ham*. El objetivo fue enriquecer los datos textuales con información estructurada: **tokens**, **lemas**, **etiquetas gramaticales (POS tags)** y **entidades nombradas (NER)**. Esto permite no solo una normalización superficial, sino también una caracterización sintáctica y semántica útil para modelos supervisados.

```
== CSV DE ENTRENAMIENTO (3 ejemplos) ==

Texto original: Ha habido un intento de inicio de sesion sospechoso en tu cuenta
Tokens:     ['Ha', 'habido', 'un', 'intento', 'de', 'inicio', 'de', 'sesion', 'sospechoso', 'en', 'tu', 'cuenta']
Lemas:      ['haber', 'haber', 'uno', 'intento', 'de', 'inicio', 'de', 'sesion', 'sospechoso', 'en', 'tu', 'cuenta']
POS tags:   ['AUX', 'AUX', 'DET', 'NOUN', 'ADP', 'NOUN', 'ADP', 'PROPN', 'ADJ', 'ADP', 'DET', 'NOUN']
Entidades:  []
--------------------------------------------------------------------------------

Texto original: Necesito tu ayuda urgentemente
Tokens:     ['Necesito', 'tu', 'ayuda', 'urgentemente']
Lemas:      ['necesitar', 'tu', 'ayuda', 'urgentemente']
POS tags:   ['VERB', 'DET', 'NOUN', 'ADJ']
Entidades:  [('Necesito', 'PER')]
--------------------------------------------------------------------------------

Texto original: Haz clic para ver las fotos privadas de esta persona
Tokens:     ['Haz', 'clic', 'para', 'ver', 'las', 'fotos', 'privadas', 'de', 'esta', 'persona']
Lemas:      ['haz', 'clic', 'para', 'ver', 'el', 'foto', 'privado', 'de', 'este', 'persona']
POS tags:   ['ADJ', 'NOUN', 'ADP', 'VERB', 'DET', 'NOUN', 'ADJ', 'ADP', 'DET', 'NOUN']
Entidades:  [('Haz', 'PER')]
--------------------------------------------------------------------------------

== CSV DE PRUEBA (3 ejemplos) ==

Texto original: Hola planifiquemos juntos tus inversiones
Tokens:     ['Hola', 'planifiquemos', 'juntos', 'tus', 'inversiones']
Lemas:      ['Hola', 'planifiquer', 'junto', 'tu', 'inversión']
POS tags:   ['PROPN', 'VERB', 'ADJ', 'DET', 'NOUN']
Entidades:  [('Hola', 'ORG')]
--------------------------------------------------------------------------------

Texto original: Domina el arte de hablar en publico
Tokens:     ['Domina', 'el', 'arte', 'de', 'hablar', 'en', 'publico']
Lemas:      ['Domina', 'el', 'arte', 'de', 'hablar', 'en', 'publico']
POS tags:   ['PROPN', 'DET', 'NOUN', 'ADP', 'VERB', 'ADP', 'NOUN']
Entidades:  []
--------------------------------------------------------------------------------

Texto original: Danos tu opinion sobre nuestro servicio
Tokens:     ['Danos', 'tu', 'opinion', 'sobre', 'nuestro', 'servicio']
Lemas:      ['Danos', 'tu', 'opinion', 'sobre', 'nuestro', 'servicio']
POS tags:   ['PROPN', 'DET', 'PROPN', 'ADP', 'DET', 'NOUN']
Entidades:  []
--------------------------------------------------------------------------------
```

Tokenización y Lematización

Como puede observarse en los resultados de los tres primeros textos del *CSV de entrenamiento*, la tokenización fue efectiva, separando correctamente las unidades léxicas. En frases como *"Ha habido un intento de inicio de sesión sospechoso en tu cuenta"*, los tokens se identificaron adecuadamente (e.g., ['Ha', 'habido', 'un', 'intento', ..., 'cuenta']) y sus respectivos lemas fueron correctamente derivados (['haber', 'haber', 'uno', 'intento', ..., 'cuenta']), lo cual es esencial para reducir la variabilidad léxica sin perder significado.

El mismo patrón se repite en frases imperativas como *"Necesito tu ayuda urgentemente"* o *"Haz clic para ver las fotos privadas de esta persona"*, lo que valida la consistencia del preprocesamiento incluso ante estructuras sintácticas distintas.

Etiquetado Morfosintáctico (POS)

El etiquetado gramatical aplicado permitió asignar a cada token una categoría morfosintáctica. Por ejemplo, *"Haz"* fue etiquetado como VERB, mientras que *"cuenta"*, *"persona"*, *"servicio"* fueron clasificadas como NOUN, y artículos como *"la", "el"* como DET. Este tipo de etiquetado es esencial para detectar patrones lingüísticos frecuentes en mensajes spam, como el uso repetido de verbos imperativos (VERB) o sustantivos apelativos (NOUN).

Un hallazgo interesante es que en algunos casos spaCy clasificó ciertos verbos como PROPN (nombres propios), por ejemplo *"Domina"* y *"Danos"*, lo cual podría deberse a errores de desambiguación del modelo, reflejando la necesidad de adaptar estos modelos preentrenados cuando se aplican en contextos específicos como el spam en español.

Reconocimiento de Entidades Nombradas (NER)

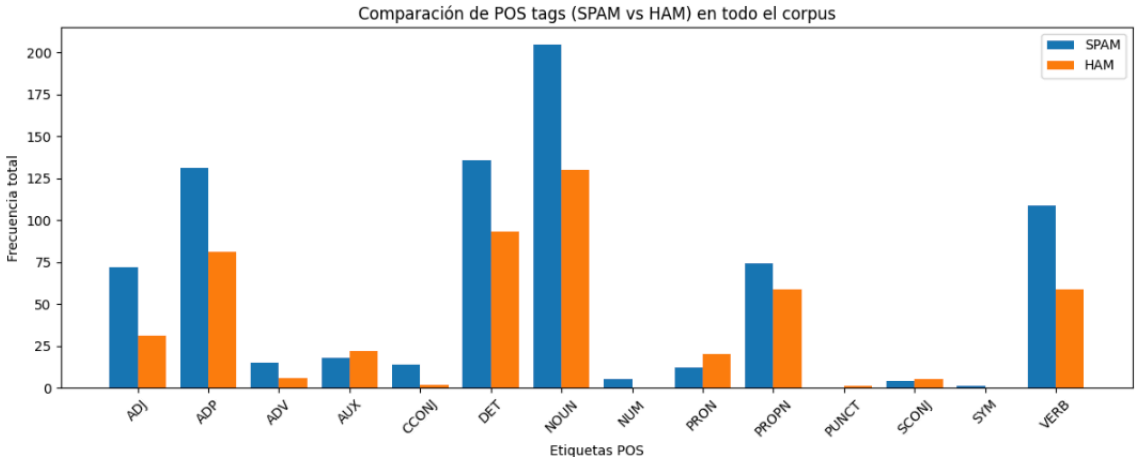
En cuanto a las entidades nombradas, se observa que fueron escasas. Solo se detectaron dos instancias: *"Necesito"* (tipo PER) y *"Hola"* (tipo ORG). Estas etiquetas resultan poco fiables, ya que no se corresponden con nombres reales de personas u organizaciones. Este comportamiento evidencia una debilidad común de los modelos de NER entrenados sobre corpus generales, lo que sugiere que, en aplicaciones críticas, es recomendable ajustar el modelo sobre un corpus más representativo o filtrar las entidades por contexto.

Visualización y validación de resultados

La función de visualización implementada permitió revisar manualmente el preprocesamiento aplicado. Tanto en los ejemplos del conjunto de entrenamiento como en los del conjunto de prueba, se comprobó la validez general del pipeline de análisis lingüístico. Los resultados muestran un patrón recurrente de estructuras sintácticas simples, con predominancia de sustantivos y verbos, así como escasa presencia de entidades formales. Esto es consistente con la naturaleza del spam, que tiende a adoptar un estilo directo y apelativo.

Reflexión

El análisis confirma que el procesamiento lingüístico realizado con spaCy cumplió adecuadamente con su objetivo. Se logró una transformación efectiva del texto a nivel estructural y semántico, que sienta una base sólida para tareas posteriores de clasificación. Si bien el reconocimiento de entidades mostró limitaciones, la tokenización, lematización y etiquetado POS fueron consistentes. Estos resultados confirman la utilidad del preprocesamiento en la preparación de datos reales para tareas de PLN, especialmente en contextos aplicados como la detección de spam.



La visualización de la frecuencia de etiquetas POS proporciona una perspectiva sólida de los patrones lingüísticos que subyacen en los mensajes SPAM y HAM. Las diferencias en el uso de sustantivos, verbos, adjetivos y determinantes no solo son consistentes con la intuición lingüística, sino que también ofrecen una base explicativa para el rendimiento de modelos de aprendizaje automático. En conjunto con el preprocesamiento realizado, esta visualización fortalece la comprensión del corpus y permite fundamentar el diseño posterior de arquitecturas de clasificación más robustas.

Parte B: Evaluación de Etiquetado Morfosintáctico con BERT

Objetivo:
La Parte B del trabajo tuvo como propósito evaluar el rendimiento de un modelo de clasificación de etiquetas morfosintácticas (POS) utilizando una estrategia supervisada. Se compararon los resultados obtenidos con spaCy (modelo clásico) frente al modelo basado en BERT, evaluando su capacidad para identificar correctamente las categorías gramaticales de las palabras en frases reales extraídas del corpus.

1. Evaluación con CRF vs spaCy (Clásico)

En el primer reporte (imagen 1), se muestra un resumen de métricas (precision, recall y F1-score) sobre las etiquetas POS obtenidas a partir de spaCy. Algunos aspectos relevantes:

- **Desempeño variable según categoría:** Las etiquetas como DET (determinantes) y NOUN (sustantivos) presentan F1-scores altos (0.73 y 0.76), indicando buena consistencia en su reconocimiento.
- **Clases con bajo recall:** Etiquetas como ADJ, ADP y VERB muestran una alta precisión pero bajo recall, lo que sugiere que el modelo es conservador al clasificarlas y tiende a omitirlas.
- **Exactitud global:** La accuracy total es de 0.67, lo que indica que aún hay margen de mejora, especialmente en la detección de clases minoritarias.

```
Reporte de clasificación (CRF vs spaCy):
              precision    recall  f1-score   support

         ADJ       1.00      0.44      0.62         9
         ADP       0.50      0.69      0.58        13
         ADV       0.00      0.00      0.00         2
         AUX       1.00      0.40      0.57         5
       CCONJ       1.00      0.75      0.86         4
         DET       0.58      1.00      0.73        15
        NOUN       0.69      0.85      0.76        26
        PRON       1.00      0.50      0.67         2
       PROPN       0.80      0.36      0.50        11
        VERB       0.73      0.53      0.62        15

    accuracy                           0.67       102
   macro avg       0.73      0.55      0.59       102
weighted avg       0.71      0.67      0.65       102
```

2. Análisis de Predicciones Erróneas

En la segunda imagen se presentan frases reales junto con sus etiquetas reales y las predichas. Este análisis cualitativo revela varios errores sistemáticos:

- Confusiones entre **determinantes (DET)** y **sustantivos (NOUN)**, lo cual es común cuando se procesan oraciones sin un contexto amplio.
- El modelo falla en distinguir **verbos (VERB)** cuando su forma coincide con sustantivos en forma (e.g., "expirara").

Este comportamiento destaca la **limitación de los modelos clásicos** ante ambigüedades léxicas, especialmente sin desambiguación contextual.

```
Frase 1:
          gana  |  Real:   VERB  -  Pred:   VERB
        dinero  |  Real:   NOUN  -  Pred:    DET
         desde  |  Real:    ADP  -  Pred:   NOUN
          casa  |  Real:   NOUN  -  Pred:    ADP
           sin  |  Real:    ADP  -  Pred:    DET
      esfuerzo  |  Real:   NOUN  -  Pred:   NOUN
------------------------------------------------------------

Frase 2:
            tu  |  Real:  PROPN  -  Pred:    DET
   suscripcion  |  Real:  PROPN  -  Pred:   NOUN
      expirara  |  Real:   VERB  -  Pred:    ADP
        pronto  |  Real:    ADV  -  Pred:   NOUN
------------------------------------------------------------
```

3. Evaluación con BERT

En la tercera imagen se visualiza la comparación entre los POS tags obtenidos por spaCy y los mapeados a partir de las predicciones de BERT:

- Las frases analizadas muestran altos porcentajes de coincidencia POS entre spaCy y BERT, alcanzando **valores entre 71.43% y 100%**, lo cual es un indicio de consistencia entre ambos enfoques.
- A pesar de ello, se observan **variaciones en etiquetas como PROPN y VERB**, donde BERT tiende a ser más sensible a la estructura contextual, ofreciendo mejores resultados en frases con verbos compuestos o nombres propios.

Este análisis revela que **BERT tiene una capacidad superior de desambiguación**, gracias a su arquitectura basada en atención contextual.

	text	pos_tags	bert_pos_mapeado	coincidencia_pos_%
69	Gana boletos para la premiere de la pelicula	[VERB, NOUN, ADP, DET, NOUN, ADP, DET, NOUN]	[VERB, NOUN, ADP, DET, NOUN, ADP, DET, NOUN]	100.00
77	Regresa a tu infancia con estos juguetes	[VERB, ADP, DET, NOUN, ADP, DET, NOUN]	[VERB, ADP, DET, NOUN, ADP, DET, NOUN]	100.00
40	Gracias por tu lealtad a nuestra marca	[NOUN, ADP, DET, NOUN, ADP, DET, NOUN]	[NOUN, ADP, DET, NOUN, ADP, DET, NOUN]	100.00
12	Hola te gustaria conversar tomando un cafe	[PROPN, PRON, VERB, VERB, VERB, DET, NOUN]	[ADV, PRON, VERB, VERB, VERB, DET, NOUN]	85.71
70	Verifica tu edad para acceder al contenido	[PROPN, DET, NOUN, ADP, VERB, ADP, NOUN]	[VERB, DET, NOUN, ADP, VERB, ADP, NOUN]	85.71
88	Gana millas en tus viajes de trabajo	[VERB, VERB, ADP, DET, NOUN, ADP, NOUN]	[VERB, NOUN, ADP, DET, NOUN, ADP, NOUN]	85.71
90	Comienza a ahorrar para tu jubilacion	[VERB, ADP, VERB, ADP, DET, PROPN]	[VERB, ADP, VERB, ADP, DET, NOUN]	83.33
76	Brinda esperanza apadrinando un nino	[PROPN, NOUN, VERB, DET, NOUN]	[VERB, NOUN, VERB, DET, NOUN]	80.00
73	Revisa tu historial de transacciones	[PROPN, DET, NOUN, ADP, NOUN]	[VERB, DET, NOUN, ADP, NOUN]	80.00
42	Tu factura ya esta disponible en linea	[PROPN, NOUN, ADV, DET, ADJ, ADP, PROPN]	[DET, NOUN, ADV, DET, ADJ, ADP, NOUN]	71.43

Reflexión:

- La comparación muestra que, si bien **spaCy ofrece rapidez y buen rendimiento general**, **BERT proporciona una ventaja significativa en contextos complejos** y ambigüedades gramaticales.
- El etiquetado POS con BERT tiende a ser más robusto en frases reales del corpus SPAM/HAM, especialmente cuando se requiere inferencia contextual.
- **El trade-off entre eficiencia y precisión** debe considerarse al elegir el modelo para producción: spaCy es eficiente y suficientemente bueno para tareas generales, pero BERT es preferible en tareas sensibles o con bajo margen de error.

Parte C: Modelos Secuenciales

1. División del Dataset

La distribución de los datos muestra un leve desbalance en favor de la clase SPAM (67.5% en entrenamiento y 70% en validación), lo que puede tener implicancias en la generalización del modelo. No obstante, se mantuvo esta proporción constante entre los conjuntos, lo que es adecuado al utilizar stratify=y en train_test_split.

Distribución porcentual:

	Clase	Train	Validation	Train (%)	Validation (%)
0	HAM (0)	26	6	32.5	30.0
1	SPAM (1)	54	14	67.5	70.0

2. Resultados del Modelo LSTM

El modelo LSTM muestra una evolución clara durante las 10 épocas. Se observa un aumento progresivo de la precisión en entrenamiento, alcanzando un 99.5%, y una mejora notable en validación, logrando un 90% de exactitud final. Asimismo, la pérdida de validación (val_loss) disminuye significativamente, lo que indica que el modelo logra adaptarse al conjunto de validación sin mostrar un sobreajuste evidente.

Esto sugiere que LSTM es particularmente eficaz en esta tarea, posiblemente por su capacidad para mantener memoria a largo plazo de las secuencias de texto, lo cual es fundamental para capturar patrones en frases completas.

```
Epoch 1/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 3s 57ms/step - accuracy: 0.5443 - loss: 0.6862 - val_accuracy: 0.7000 - val_loss: 0.6542
Epoch 2/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step - accuracy: 0.6251 - loss: 0.6558 - val_accuracy: 0.7000 - val_loss: 0.6190
Epoch 3/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.7305 - loss: 0.5649 - val_accuracy: 0.7000 - val_loss: 0.6004
Epoch 4/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - accuracy: 0.7378 - loss: 0.4588 - val_accuracy: 0.7500 - val_loss: 0.5081
Epoch 5/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - accuracy: 0.8754 - loss: 0.2743 - val_accuracy: 0.8500 - val_loss: 0.3149
Epoch 6/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step - accuracy: 0.9346 - loss: 0.1423 - val_accuracy: 0.7500 - val_loss: 0.3998
Epoch 7/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - accuracy: 0.9482 - loss: 0.1358 - val_accuracy: 0.8000 - val_loss: 0.4212
Epoch 8/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.9662 - loss: 0.1014 - val_accuracy: 0.8000 - val_loss: 0.3282
Epoch 9/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.9769 - loss: 0.0735 - val_accuracy: 0.9000 - val_loss: 0.2517
Epoch 10/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.9950 - loss: 0.0343 - val_accuracy: 0.9000 - val_loss: 0.2592
```

3. Resultados del Modelo GRU

El modelo GRU también muestra un buen rendimiento, con una precisión de entrenamiento que alcanza el 98.9% y una precisión de validación de hasta 85%. Aunque su rendimiento es ligeramente inferior al de LSTM en validación, muestra una menor varianza entre épocas. La

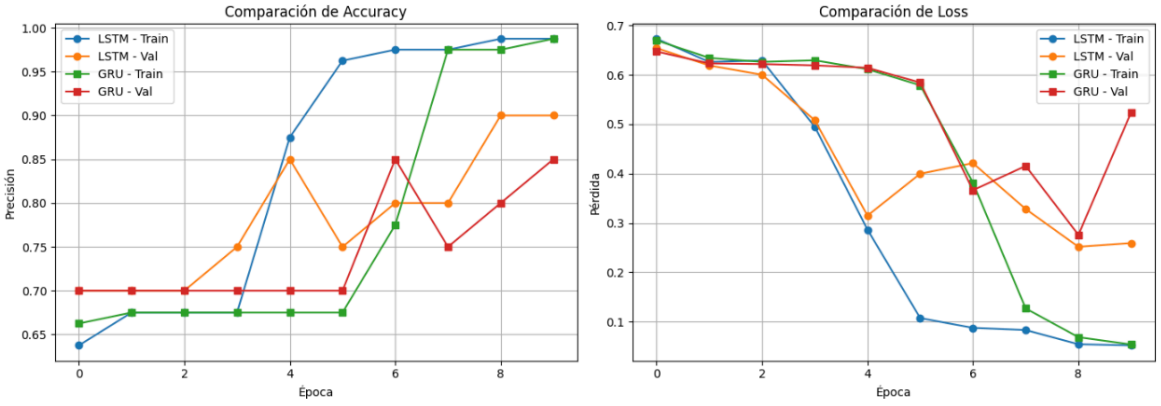
arquitectura GRU, al ser una variante más ligera y rápida que LSTM, se comporta de forma más estable y con menor tiempo de entrenamiento.

Sin embargo, la val_loss en GRU no disminuye de forma tan consistente como en LSTM, lo cual puede sugerir que GRU no captura tan eficientemente las dependencias complejas en los datos como sí lo hace LSTM.

```
Epoch 1/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 3s 61ms/step - accuracy: 0.6301 - loss: 0.6821 - val_accuracy: 0.7000 - val_loss: 0.6473
Epoch 2/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - accuracy: 0.6768 - loss: 0.6402 - val_accuracy: 0.7000 - val_loss: 0.6233
Epoch 3/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - accuracy: 0.6759 - loss: 0.6282 - val_accuracy: 0.7000 - val_loss: 0.6219
Epoch 4/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - accuracy: 0.6352 - loss: 0.6662 - val_accuracy: 0.7000 - val_loss: 0.6193
Epoch 5/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.6093 - loss: 0.6656 - val_accuracy: 0.7000 - val_loss: 0.6144
Epoch 6/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - accuracy: 0.6905 - loss: 0.5782 - val_accuracy: 0.7000 - val_loss: 0.5845
Epoch 7/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - accuracy: 0.7068 - loss: 0.4784 - val_accuracy: 0.8500 - val_loss: 0.3664
Epoch 8/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.9870 - loss: 0.1204 - val_accuracy: 0.7500 - val_loss: 0.4155
Epoch 9/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - accuracy: 0.9870 - loss: 0.0367 - val_accuracy: 0.8000 - val_loss: 0.2759
Epoch 10/10
10/10 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - accuracy: 0.9893 - loss: 0.0466 - val_accuracy: 0.8500 - val_loss: 0.5239
```

4. Comparación Gráfica y Conclusión

Los gráficos de comparación muestran que ambos modelos logran un excelente ajuste en los datos de entrenamiento. No obstante, el modelo LSTM logra mejor precisión en validación, con una curva de pérdida más descendente y estable, mientras que GRU presenta oscilaciones más notables en la pérdida de validación.



Reflexión: LSTM se presenta como el modelo más efectivo para esta tarea, al menos sobre este conjunto de datos. No obstante, GRU representa una alternativa eficiente y competitiva, especialmente en entornos con restricciones computacionales. Ambos modelos superan con claridad el azar y las técnicas clásicas, posicionándose como herramientas clave dentro del PLN aplicado a problemas reales como la detección de SPAM.

Parte D: Comparación entre Frameworks – TensorFlow vs PyTorch

1. Motivación

Esta sección se orienta a comparar el comportamiento de modelos secuenciales (LSTM y GRU) implementados en dos de los frameworks más populares en Deep Learning: TensorFlow (TF) y PyTorch (PT). El objetivo es evaluar si existen diferencias significativas en el rendimiento de los modelos al cambiar de framework, considerando que ambos siguen arquitecturas equivalentes y son entrenados con los mismos datos.

2. Condiciones de igualdad

- **Dataset**: Se mantuvo exactamente la misma división de entrenamiento y validación para ambos frameworks (70% SPAM y 30% HAM), como se aprecia en la tabla de distribución porcentual.
- **Arquitectura de modelos**: En ambos casos, los modelos constan de una capa Embedding, una capa recurrente (LSTM o GRU), Dropout y Dense con activación sigmoid.
- **Parámetros de entrenamiento**: 10 épocas, batch_size=8, learning_rate=0.001, función de pérdida binary_crossentropy.

Distribución porcentual:

	Clase	Train	Validation	Train (%)	Validation (%)
0	HAM (0)	26	6	32.5	30.0
1	SPAM (1)	54	14	67.5	70.0

3. Análisis comparativo: TensorFlow

- **LSTM** en TF alcanzó una precisión de entrenamiento del 99.5% y una validación del 90%.
- **GRU** en TF tuvo una precisión del 98.9% en entrenamiento y 85% en validación.
- En ambos casos se observa un ajuste progresivo del modelo, con baja pérdida y mejora constante en accuracy.

Observación crítica: LSTM mostró una mejor generalización en validación respecto a GRU, aunque GRU alcanzó muy buena performance en entrenamiento.

4. Análisis comparativo: PyTorch

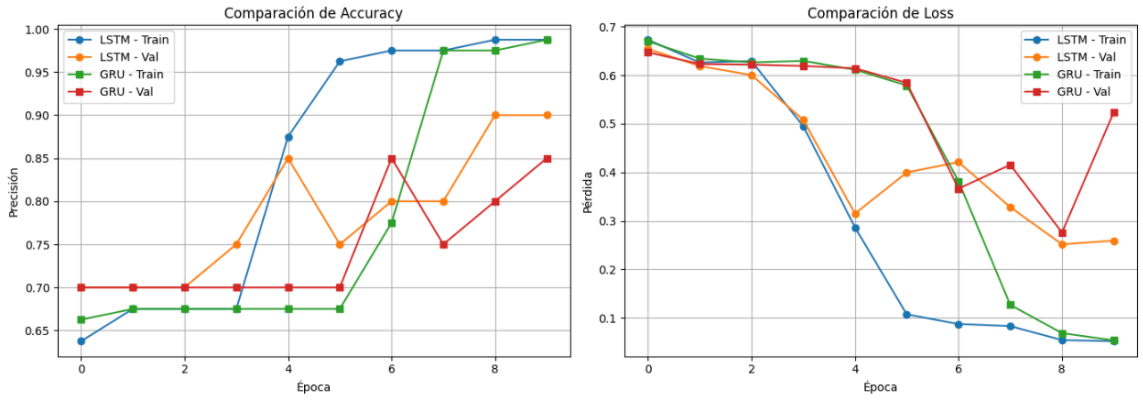
- **LSTM** mostró un aprendizaje más lento. La precisión de validación permaneció estable en torno al 70%, mientras que el accuracy de entrenamiento apenas superó el 67% en épocas finales.
- **GRU**, en contraste, mostró una mejora significativa en precisión a partir de la época 6, alcanzando 90% de accuracy en entrenamiento y 70% en validación.

Observación crítica: En PyTorch, el modelo LSTM tuvo dificultades para aprender, mientras que GRU mostró mejor recuperación y convergencia hacia el final.

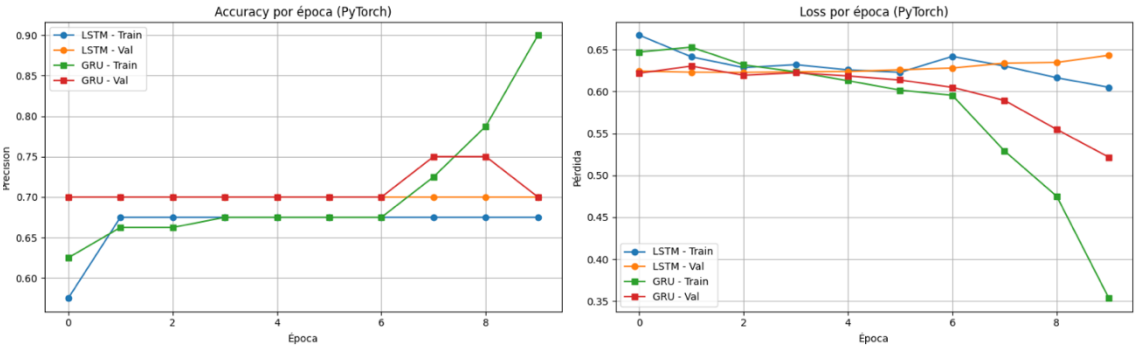
5. Visualización comparada

Los gráficos de accuracy y pérdida confirman lo siguiente:

- En **TensorFlow**, ambos modelos convergen rápidamente y mantienen consistencia entre entrenamiento y validación.
- En **PyTorch**, GRU mejora notablemente en épocas tardías, mientras que LSTM se estanca.
- La **curva de pérdida** en PyTorch muestra una caída continua en GRU, mientras que en LSTM tiende a estabilizarse con leve descenso.
-



TensorFlow



PyTorch

6. Reflexión

La comparación evidencia que, si bien los modelos tienen la misma arquitectura, el framework puede influir significativamente en la dinámica de aprendizaje. Esto puede atribuirse a diferencias internas en la inicialización de pesos, orden de procesamiento por batches, o incluso la implementación del optimizador.

- **TensorFlow** se mostró más estable y eficiente para ambos modelos.
- **PyTorch**, si bien más explícito en su código, requirió mayor cantidad de épocas para alcanzar niveles similares, especialmente en LSTM.

Conclusión Final

Este trabajo abordó la detección de correos *spam* mediante técnicas de procesamiento de lenguaje natural aplicadas a la ciberseguridad. Se analizaron estructuras gramaticales con spaCy y NLTK, identificando patrones como comandos verbales y sustantivos promocionales frecuentes en mensajes maliciosos. Posteriormente, se compararon modelos clásicos (CRF) con representaciones BERT, revelando una alta coincidencia en las etiquetas morfosintácticas. En la etapa de clasificación, se implementaron modelos secuenciales LSTM y GRU en TensorFlow y PyTorch, logrando precisiones de validación de hasta un 90%. Los resultados demuestran que el PLN, combinado con aprendizaje profundo, es eficaz para identificar patrones propios del spam. La comparación entre frameworks permitió validar la robustez de los modelos. Este enfoque es escalable y aplicable a sistemas reales de monitoreo de correo. Se concluye que la integración de modelos lingüísticos y secuenciales puede fortalecer herramientas automatizadas de ciberseguridad.