CLASIFICACIÓN AUTOMÁTICA DE MATERIAS EN REPOSITORIOS INSTITUCIONALES MEDIANTE APRENDIZAJE SUPERVISADO Y REPRESENTACIONES VECTORIALES MULTILINGÜES:

Un estudio de caso en SEDICI

Carlos Javier Nusch*
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
carlosnusch@sedici.unlp.edu.ar
https://orcid.org/0000-0003-1715-4228

Leticia Cecilia Cagnina
Universidad Nacional de San Luis;
LIDIC, Argentina
lcagnina@unsl.edu.ar
https://orcid.org/0000-0001-7825-2927

Silvia B. Peloche
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
silvia@sedici.unlp.edu.ar
https://orcid.org/0009-0009-1930-6521

Gonzalo Villarreal
Universidad Nacional de La Plata,
PREBI-SEDICI, y CESGI, Comisión de
Investigaciones Científicas, Argentina
gonzalo@prebi.unlp.edu.ar
https://orcid.org/0000-0002-3602-8211

Ariel Lira
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
alira@sedici.unlp.edu
https://orcid.org/0000-0003-3647-3101

Leandro Antonelli
Universidad Nacional de La Plata, LIFIA y
CAETI, Facultad de Tecnología Informática,
Universidad Abierta Interamericana,
Argentina
lanto@lifia.info.unlp.edu.ar
https://orcid.org/0000-0003-1388-0337

Lucas Eduardo Folegotto
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
lucas@prebi.unlp.edu
https://orcid.org/0009-0004-0912-7638

Marcelo Luis Errecalde
Universidad Nacional de San Luis;
LIDIC, Argentina
merreca@unsl.edu.ar
https://orcid.org/0000-0001-5605-8963

Marisa Raquel De Giusti
Universidad Nacional de La Plata, Argentina
marisadegiusti@gmail.com
https://orcid.org/ 0000-0003-2422-6322

DOI: *10.22477/xiv.biredial.417*
EJE TEMÁICO: *Infraestructura tecnológica*

* Las contribuciones de los autores se declaran siguiendo la taxonomía TaDiRAH (Taxonomy of Digital Research Activities in the Humanities). Carlos Javier Nusch fue responsable del desarrollo del código, el procesamiento del lenguaje natural, el análisis de datos y la redacción del artículo (programación, modelado, codificación de texto, análisis, limpieza, evaluación y enriquecimiento). Lucas Eduardo Folegotto estuvo a cargo del diseño, edición y mejora de las imágenes utilizadas en el trabajo, en colaboración con Carlos Javier Nusch. Leticia Cagnina, Silvia Peloche, Ariel Lira, Gonzalo Villarreal, Leandro Antonelli, Marcelo Errecalde y Marisa De Giusti participaron en tareas de revisión y corrección del contenido.

RESUMEN

Presentación del problema: El presente trabajo aborda la tarea de clasificación automática por materias para los contenidos albergados en el repositorio institucional SEDICI. A partir de un corpus de 126.081 ítems se propone ahora un enfoque supervisado de clasificación multilabel que permita predecir las materias asignadas a los ítems del repositorio a partir de sus resúmenes y palabras clave. **Materiales y metodología:** Los ítems incluyen resúmenes textuales, palabras clave y etiquetas temáticas. Se realizó un análisis de cobertura de etiquetas para obtener un subconjunto óptimo de clases que concentren la mayor parte de los ejemplos en el corpus. Luego se aplicaron distintas técnicas de representación del texto, incluyendo vectorizaciones clásicas por n-gramas (TF-IDF y frecuencia de términos) y modelos de embeddings multilingües (SBERT y LaBSE). Sobre estas representaciones se entrenaron varios clasificadores multilabel, como regresión logística, máquinas de soporte vectorial, Random Forest, Multinomial Naive Bayes y clasificadores por descenso de gradiente. La evaluación se realizó mediante métricas específicas para clasificación multilabel, incluyendo F1-score micro y macro. **Resultados:** Se observó que la combinación de Frecuencia de Término - Frecuencia Inversa de Documento (TF-IDF) con Máquinas de Soporte Vectorial Lineal (Linear SVC) ofreció un rendimiento destacado entre los enfoques clásicos, alcanzando los mayores valores de F1 macro y F1 micro en ambas configuraciones del conjunto de etiquetas. Los modelos basados en embeddings, especialmente LaBSE y SBERT combinados con Linear SVC, demostraron también un desempeño competitivo, superando en varios casos a los métodos clásicos, aunque a costa de mayores tiempos de entrenamiento. El Clasificador Lineal entrenado con Descenso de Gradiente Estocástico (SGD) se posicionó como una alternativa eficiente y escalable, con tiempos reducidos y métricas satisfactorias. La reducción del espacio de etiquetas de 61 a 37 materias permitió mejorar globalmente la precisión y reducir la complejidad computacional. **Conclusiones:** Este estudio se propuso demostrar la viabilidad de aplicar modelos supervisados para la clasificación automática de materias en grandes volúmenes de datos textuales en repositorios institucionales. La metodología propuesta es replicable y puede adaptarse a otros contextos documentales con estructuras temáticas similares, y podría contribuir a mejorar la eficiencia y calidad del proceso de curaduría de datos y materiales en repositorios institucionales.

Palabras-clave: Repositorios Institucionales, clasificación multilabel, aprendizaje automático, mapeo temático, SBERT, LaBSE, TF-IDF, curaduría de metadatos.

ABSTRACT

Problem Statement: This work addresses the task of automatic subject classification for the contents hosted in the SEDICI institutional repository. Using a corpus of 126,081 items, a supervised multi-label classification approach is proposed to predict the subjects assigned to the repository items based on their abstracts and keywords. **Materials and Methodology:** The items include textual abstracts, keywords, and subject tags. A tag coverage analysis was performed to obtain an optimal subset of classes that concentrate the majority of examples in the corpus. Various text representation techniques were then applied, including classic n-gram vectorization (TF-IDF and term frequency) and multilingual embedding models (SBERT and LaBSE). Several multi-label classifiers were trained on these representations, such as Logistic Regression, Support Vector Machines, Random Forest, Multinomial Naive Bayes, and Stochastic Gradient Descent classifiers. Evaluation was performed using metrics specific to multi-label classification, including micro and macro F1-score. **Results:** It was observed that the combination of Term Frequency-Inverse Document Frequency (TF-IDF) with Linear Support Vector Classifiers (Linear SVC) offered outstanding performance among the classic approaches, achieving the highest macro-F1 and micro-F1 scores in both tag set configurations. Embedding-based models, especially LaBSE and SBERT combined with Linear SVC, also demonstrated competitive performance, surpassing classical methods in several cases, albeit at the cost of longer training times. The Linear Classifier trained with Stochastic Gradient Descent (SGD) positioned itself as an efficient and scalable alternative, with reduced processing times and satisfactory metrics. Reducing the label space from 61 to 37 subjects allowed for an overall improvement in accuracy and reduced computational complexity. **Conclusions:** This study aimed to demonstrate the feasibility of applying supervised models for automatic subject

classification of large volumes of textual data in institutional repositories. The proposed methodology is replicable and can be adapted to other documentary contexts with similar thematic structures, and could contribute to improving the efficiency and quality of the data and material curation process in institutional repositories.

Keywords: Institutional Repositories, multilabel classification, machine learning, subject indexing, SBERT, LaBSE, TF-IDF, metadata curation.

INTRODUCCIÓN

En trabajos anteriores (Nusch et al., 2025), se exploró la detección automática del idioma en resúmenes de ítems del repositorio institucional SEDICI, empleando tanto bibliotecas de clasificación de idioma con un enfoque de tipo zero-shot como modelos entrenados específicamente para aquella tarea (mBERT, SBERT y XLM-RoBERTa). Aquel estudio permitió no solo validar la calidad de la catalogación manual, sino también corregir en lote errores frecuentes en el campo de idioma, aportando herramientas automáticas para las tareas de curaduría que se desarrollan a diario en repositorios académicos.

El presente trabajo toma como punto de partida el mismo corpus[1] y las prácticas de procesamiento desarrolladas, pero propone una tarea de naturaleza diferente: la clasificación automática de las materias asignadas a los ítems del repositorio. A diferencia de la detección de idioma, que consiste en una tarea de clasificación single-label (una sola clase por instancia) en la que un resumen puede pertenecer a un solo idioma, la asignación de materias supone una tarea de clasificación multilabel, en la que cada ítem puede tener asociadas múltiples materias simultáneamente; un artículo, por ejemplo, puede pertenecer tanto Ciencias Informáticas como a Medicina simultáneamente o inclusive a más áreas del conocimiento[2].

El desafío que se presenta, entonces, es entrenar modelos capaces de predecir correctamente una o varias materias por ítem, utilizando como entrada los resúmenes y/o las palabras clave asociadas. Esto quiere decir que clasificamos los ítems del repositorio basándonos en la información que los describe y asignándoles pertenencia a una o más materias de acuerdo a la información contenida tanto en el campo de palabras clave como en el campo de resumen.

Para poder utilizar métodos de clasificación basados en algoritmos de aprendizaje automático se requiere representar un texto de forma numérica. Para ello se utilizan representaciones vectoriales. Una representación vectorial convierte un texto en una secuencia de números. En los enfoques más simples, como el modelo de *bolsa de palabras* (en inglés *Bag*

[1] El conjunto de datos incluía información de 126.081 ítems, todos los presentes en el repositorio hasta el 7 de abril de 2022.

[2] Por materia entendemos el tema o contenido temático de un documento, tal como se representa mediante un encabezamiento de materia, es decir, un término normalizado y controlado utilizado para describir de manera uniforme los temas tratados. Estos encabezamientos se seleccionan a partir de listas específicas, como la Library of Congress Subject Headings (LCSH), una lista ordenada alfabéticamente de términos temáticos. En SEDICI se utiliza una lista propia de encabezamientos basada en la estructura académica de la Universidad Nacional de La Plata, en particular en las carreras y disciplinas de sus facultades.

of Words o *BoW*), se cuenta cuántas veces aparece cada palabra en un texto. Por ejemplo, en el modelo de Bolsa de Palabras (Bag of Words, BoW), si un resumen contiene las palabras *inteligencia*, *artificial* y *textos*, y otro contiene *inteligencia* y *robots*, los vectores podrían verse del siguiente modo:

Tabla 1 - Ejemplo simplificado del modelo *Bolsa de Palabras*, donde cada resumen es representado por la cantidad de veces que aparecen determinadas palabras. Esta técnica no considera el significado ni el orden de las palabras, solo su frecuencia.

Palabra	Resumen 1	Resumen 2
inteligencia	1	1
artificial	1	0
textos	1	0
robots	0	1

Fuente: Elaboración propia (2025).

Este tipo de representación de textos es bastante básico y plano puesto que no distingue qué palabras son más importantes para una buena clasificación. Por eso se suele usar TF-IDF, que asigna mayor peso a las palabras que son frecuentes en un documento pero poco comunes en el resto. Así, si *inteligencia* aparece en muchos ítems, pero *procesamiento* aparece solo en algunos, TF-IDF ayuda a destacar *procesamiento* por ser más informativa.

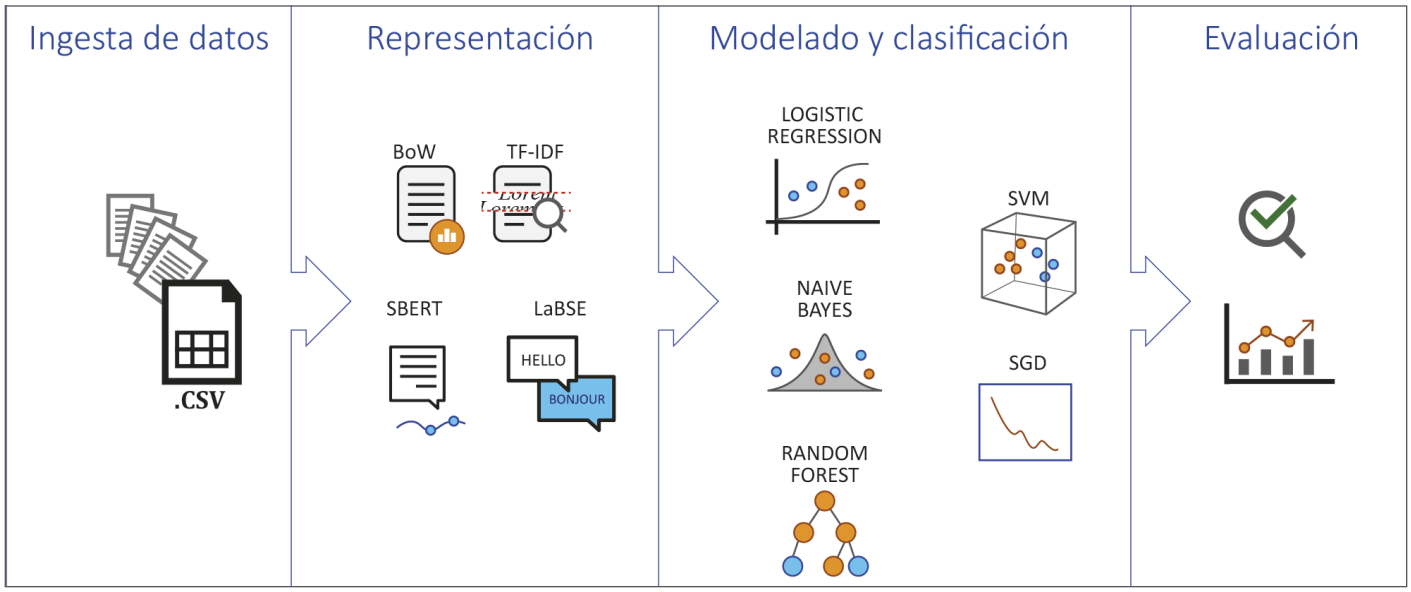
Pero se puede inclusive utilizar modos de representación de palabras u oraciones mucho más complejos y ricos. En lugar de contar palabras, se pueden usar modelos más sofisticados llamados embeddings. Los embeddings intentan representar el significado del texto. Por ejemplo, si un ítem usa *procesamiento de textos* y otro dice *análisis de documentos*, el modelo puede aprender que están hablando de cosas similares, aunque no usen exactamente las mismas palabras. También se utiliza un tipo de embeddings llamado embeddings contextuales que tienen la capacidad de detectar matices entre palabras vecinas: por ejemplo, la palabra *banco* tendrá representaciones distintas si aparece en el contexto de *plaza* o en el de *dinero*. Modelos como SBERT o LaBSE generan estas representaciones según el contexto, lo que permite comparar textos por su sentido, resolviendo problemas de homonimia o sinonimia, incluso si usan palabras distintas para un mismo concepto. SBERT está entrenado para tareas de similitud semántica entre frases (por ejemplo, ver si dos resúmenes significan lo mismo) y funciona muy bien en pocos idiomas. LaBSE está optimizado para funcionar bien en muchos idiomas a la vez (más de 100), y es especialmente útil en contextos multilingües como el de SEDICI, donde hay textos en español, inglés, portugués y otros idiomas.

En este trabajo se experimentó con estos distintos enfoques de representación del texto: desde técnicas clásicas basadas en n-gramas (TF-IDF, o frecuencia de palabras) hasta

embeddings multilingües generados por modelos como Sentence-BERT (SBERT) (Reimers & Gurevych, 2019) y Language-Agnostic BERT Sentence Embedding (LaBSE) (Feng et al., 2022). Sobre estas representaciones, se entrenaron diferentes clasificadores multilabel, incluyendo regresión logística, máquinas de soporte vectorial, bosques aleatorios y variantes de descenso de gradiente.

Este nuevo estudio busca ampliar el repertorio de herramientas automáticas disponibles para los repositorios institucionales y ofrecer soluciones replicables y eficientes para tareas críticas de descripción temática de los documentos. Como el dataset presenta grandes diferencias en la proporción en la que están distribuidas las etiquetas de materias (algunas como Ciencias Informáticas presentan ejemplos del orden de decenas de miles, mientras que otras, como Derechos Humanos, unos doscientos) se examinan tanto la precisión general de los modelos como su capacidad para reconocer materias minoritarias, a fin de garantizar una clasificación robusta incluso en contextos de desbalance de clases, algo habitual en colecciones documentales amplias y heterogéneas.

Figura 1 - Flujo de trabajo de la tarea de clasificación dividido por etapas, algoritmos y técnicas utilizadas.



Fuente: Elaboración propia (2025).

ANÁLISIS DE COBERTURA ACUMULADA DE ETIQUETAS

Como paso previo al entrenamiento de modelos de clasificación multilabel, se realizó un análisis de distribución de etiquetas con la idea de identificar un subconjunto óptimo de clases que concentren la mayor parte de los ejemplos en el corpus. Este procedimiento se conoce como análisis de cobertura acumulada de etiquetas (*cumulative label coverage analysis*), una

técnica comúnmente utilizada para abordar el desbalance de clases en tareas de aprendizaje automático. El procedimiento consistió en:

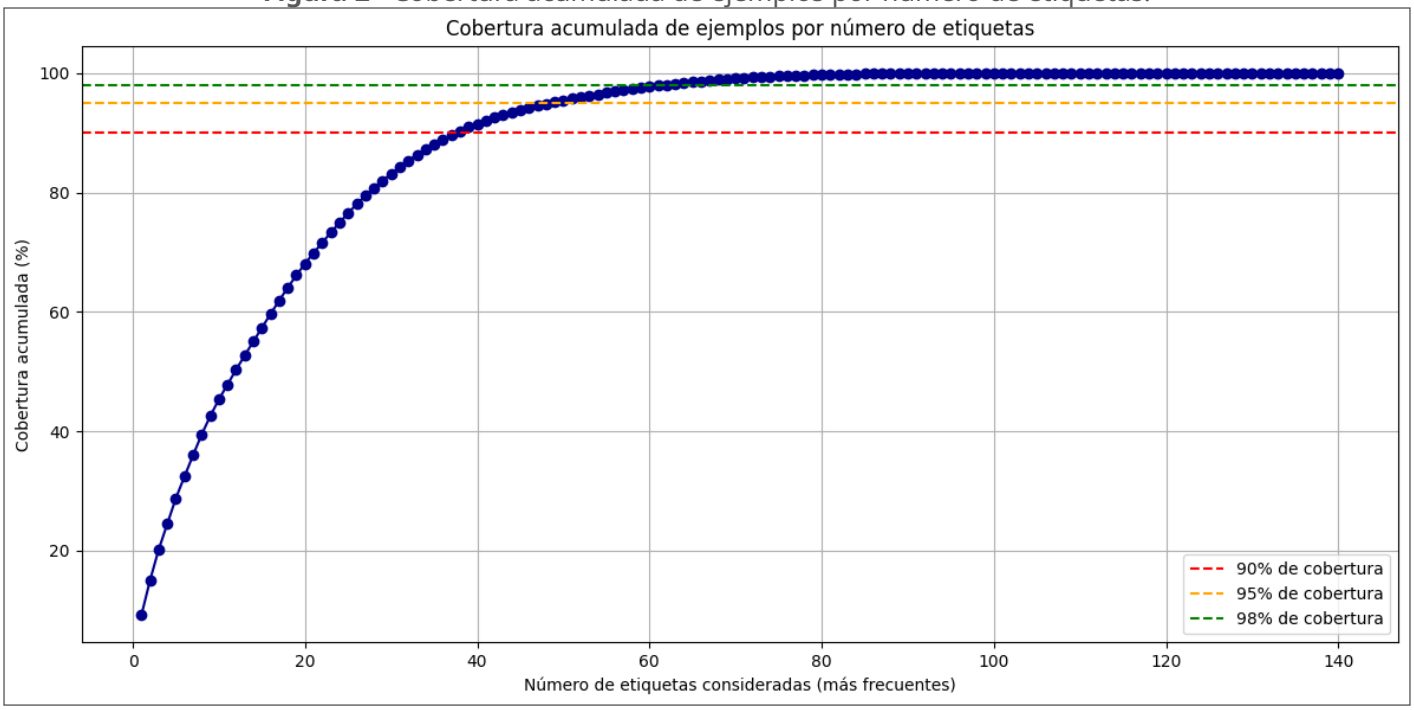
1. Extraer todas las etiquetas (materias) presentes en las columnas temáticas del dataset, normalizándolas.
2. Calcular la frecuencia absoluta de cada etiqueta y ordenarlas de forma descendente según su número de apariciones.
3. Calcular la frecuencia acumulada y el porcentaje acumulado de cobertura, es decir, la proporción del total de ejemplos que queda cubierta al considerar progresivamente las etiquetas más frecuentes.

En este caso particular, se identificó que:

- Las 37 etiquetas más frecuentes cubrían el 90% de los más de 126 mil registros del dataset.
- Las 48 etiquetas más frecuentes alcanzaban una cobertura del 95% del dataset
- Y con 61 etiquetas, se lograba cubrir el 98% del dataset.

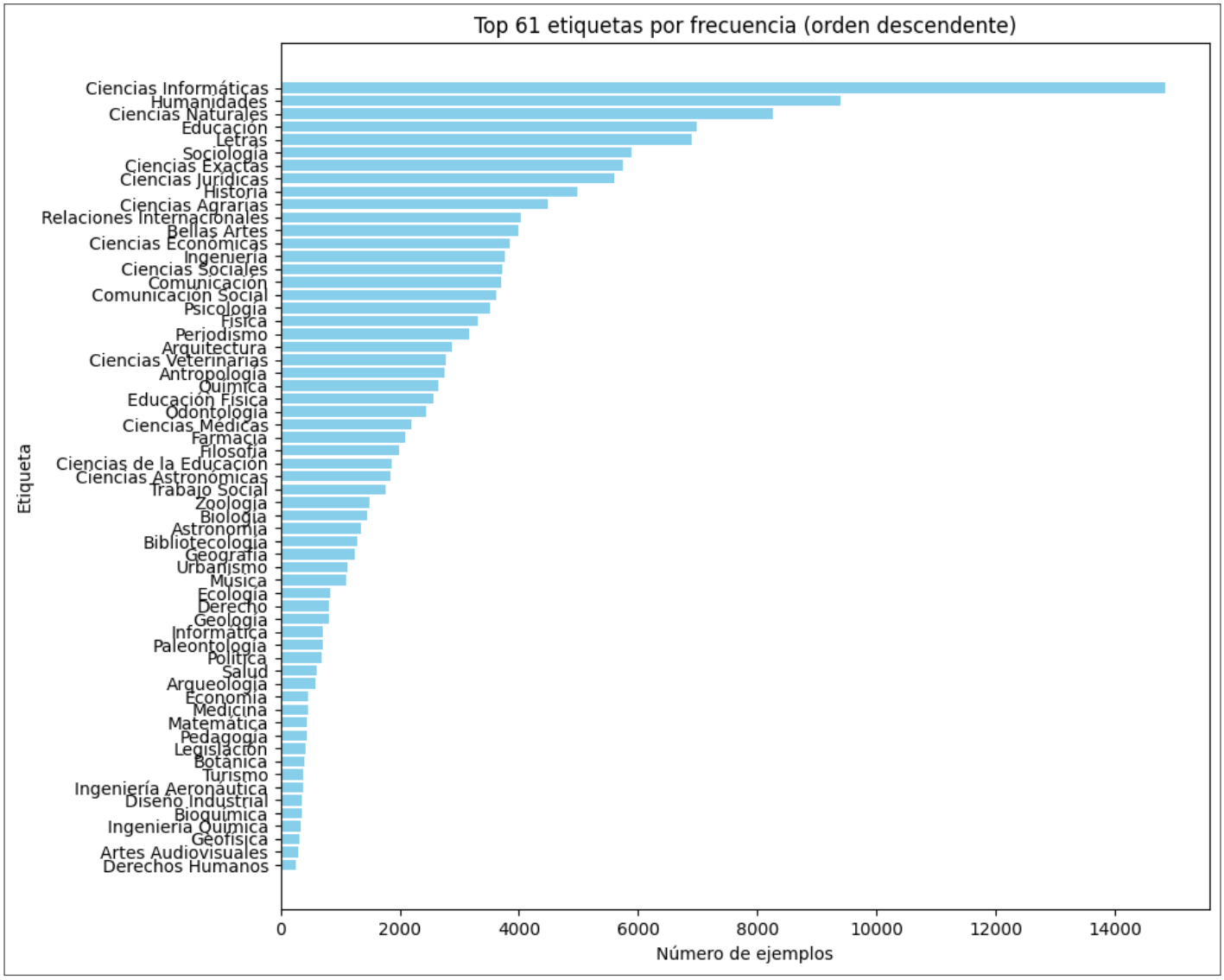
Esto significa que de las 140 etiquetas diferentes presentes en el corpus, más de la mitad tienen una frecuencia tan baja que su impacto en la cobertura general es mínimo. Este análisis justifica la posibilidad de reducir el número de etiquetas consideradas durante el entrenamiento. La selección de este subconjunto optimizado puede contribuir a mejorar la eficiencia del modelo, reducir la complejidad del espacio de salida y facilitar la curación de datos.

Figura 2 - Cobertura acumulada de ejemplos por número de etiquetas.



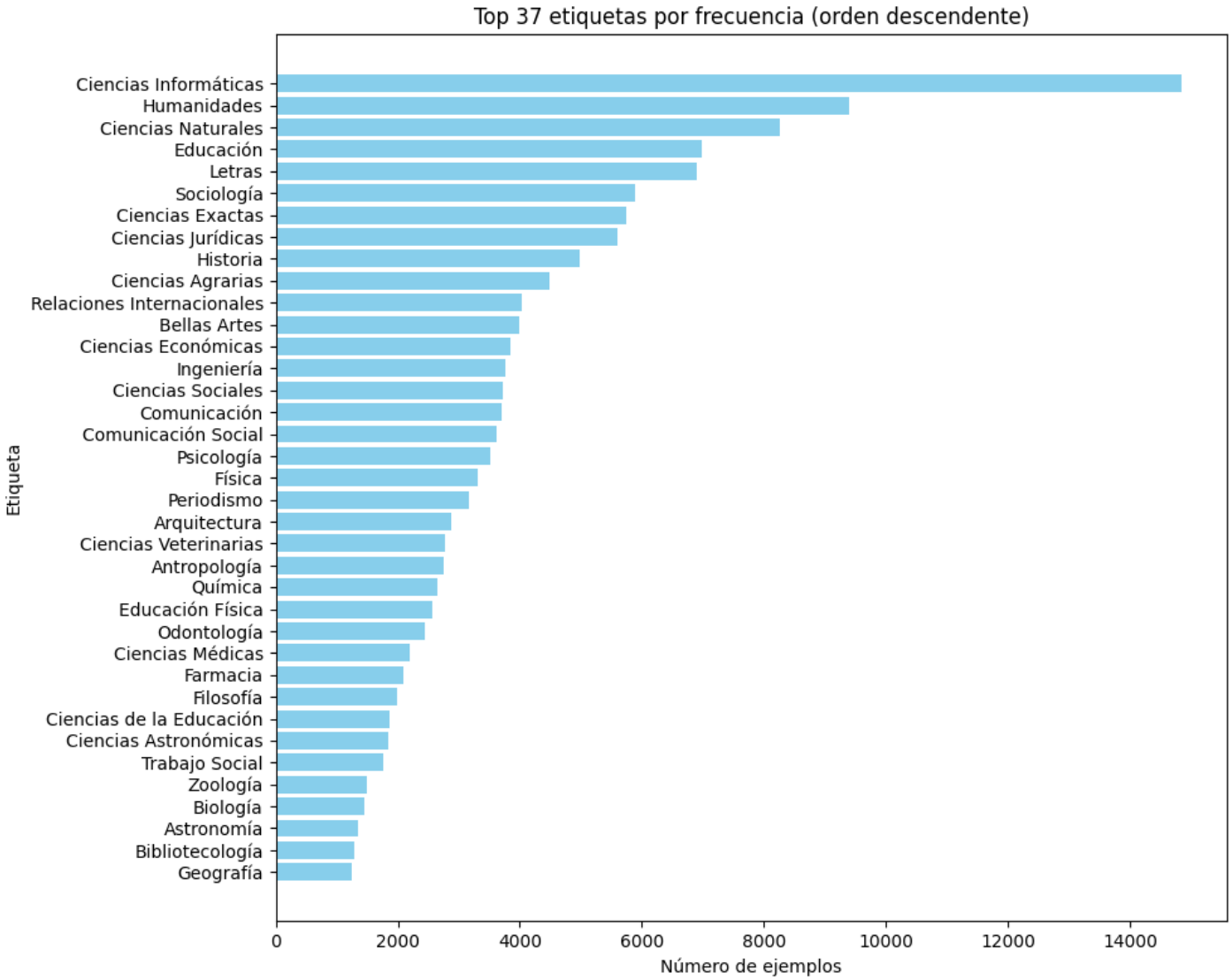
Fuente: Elaboración propia (2025).

Figura 3 - Gráfico de barras con la distribución de las 61 etiquetas más relevantes.



Fuente: Elaboración propia (2025).

Figura 4 - Gráfico de barras con la distribución de las 61 etiquetas más relevantes.



Fuente: Elaboración propia (2025).

SELECCIÓN Y PREPROCESAMIENTO DE ETIQUETAS

A partir del dataset original exportado desde el repositorio institucional SEDICI, se procedió a una primera fase de limpieza y preparación de las etiquetas temáticas, necesarias para abordar la tarea de clasificación multilabel. El archivo original contenía múltiples columnas de metadatos Dublin Core adaptados al propio esquema del repositorio, algunas de ellas asociadas a materias, tales como sedici.subject.materias, sedici.subject.materias[], sedici.subject.materias[es] y sedici.subject.other[es], todas ellas susceptibles de contener información relevante. Estas columnas fueron inspeccionadas de manera conjunta para extraer, normalizar y consolidar las etiquetas de materia asociadas a cada ítem.

Cada campo podía contener múltiples materias, concatenadas mediante el delimitador || y estructuradas según el esquema *Etiqueta::URI::ID*. Con el objetivo de obtener únicamente las etiquetas léxicas necesarias para el proceso de clasificación, se implementó un procedimiento de parsing que extrajo la parte anterior al primer delimitador ::, descartando los identificadores o URIs internos.

El procesamiento se llevó a cabo con dos variantes (una con 61 materias y otra con 37) tanto para optimizar el rendimiento del modelo y como para evaluar su comportamiento frente a distintas configuraciones del espacio de etiquetas. En el segundo caso, la reducción de cantidad de materias respondió a la necesidad de disminuir la dispersión, mejorar la representatividad de las clases y asegurar una cantidad mínima de ejemplos por clase que permitiera un entrenamiento supervisado efectivo.

Finalmente, se eliminaron los ítems que, tras este filtrado, no conservaban ninguna de las etiquetas seleccionadas.

DIVISIÓN DEL CONJUNTO DE DATOS

Con el conjunto ya filtrado y las etiquetas seleccionadas, se procedió a dividir cada dataset en tres subconjuntos independientes para las fases de entrenamiento, validación y prueba.

Dado que la tarea a resolver es de clasificación multilabel, se utilizó la clase *MultiLabelBinarizer* de *scikit-learn* para transformar las etiquetas de cada ítem en vectores binarios que indican la presencia o ausencia de cada una de las 60 materias. Esta representación fue utilizada para garantizar una distribución uniforme de ejemplos al momento de la partición, aunque no se aplicó estratificación explícita.

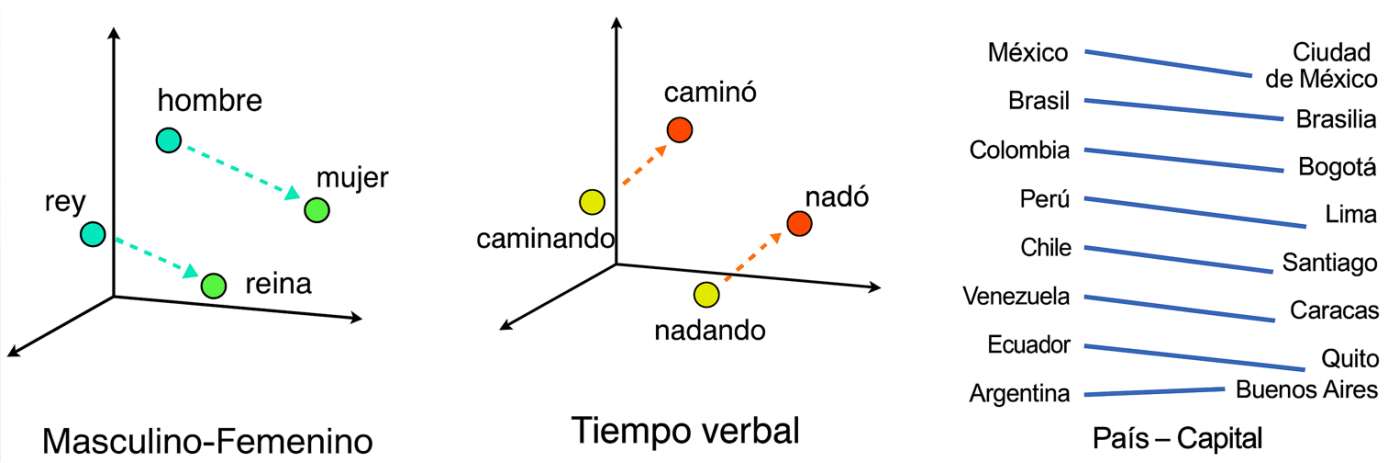
La división se llevó a cabo en dos etapas consecutivas: en primer lugar, se separó el 20% del total de datos para conformar el conjunto de prueba. Posteriormente, del 80% restante se extrajo un 10% para validación y el 90% restante se utilizó como conjunto de entrenamiento. Esta configuración dio lugar a una distribución final de:

- Dataset con 61 etiquetas:
  - Entrenamiento: 86.999 ejemplos (70%)
  - Validación: 12.429 ejemplos (10%)
  - Prueba: 24.858 ejemplos (20%)
- Dataset con 37 etiquetas:
  - Entrenamiento: 82.099 ejemplos (70%)
  - Validación: 11.729 ejemplos (10%)
  - Prueba: 23.457 ejemplos (20%)

METODOLOGÍA: REPRESENTACIÓN Y CLASIFICACIÓN DE TEXTOS

Con el conjunto de datos ya dividido y binarizado, se llevaron a cabo diversos experimentos de clasificación multilabel sobre los resúmenes y palabras clave de los ítems, con el objetivo de predecir automáticamente sus materias temáticas. Para ello, se implementó una arquitectura de entrenamiento basada en la biblioteca *scikit-learn*, que combinó múltiples representaciones vectoriales del texto con diferentes clasificadores multilabel, encapsulados en un enfoque *One vs Rest* (también conocido como *OvR* o *One-vs-All*). Como se explicó anteriormente, una representación vectorial convierte un texto en una secuencia de números que pueden ser entendidos por un algoritmo de aprendizaje automático. Por ejemplo, una frase cualquiera puede representarse como un vector de frecuencias de palabras (cuántas veces aparece cada palabra), o mediante técnicas más complejas como los embeddings, donde las palabras con significados similares se ubican cerca unas de otras en un espacio numérico, y es posible establecer relaciones de similitud entre términos utilizando medidas de distancia.

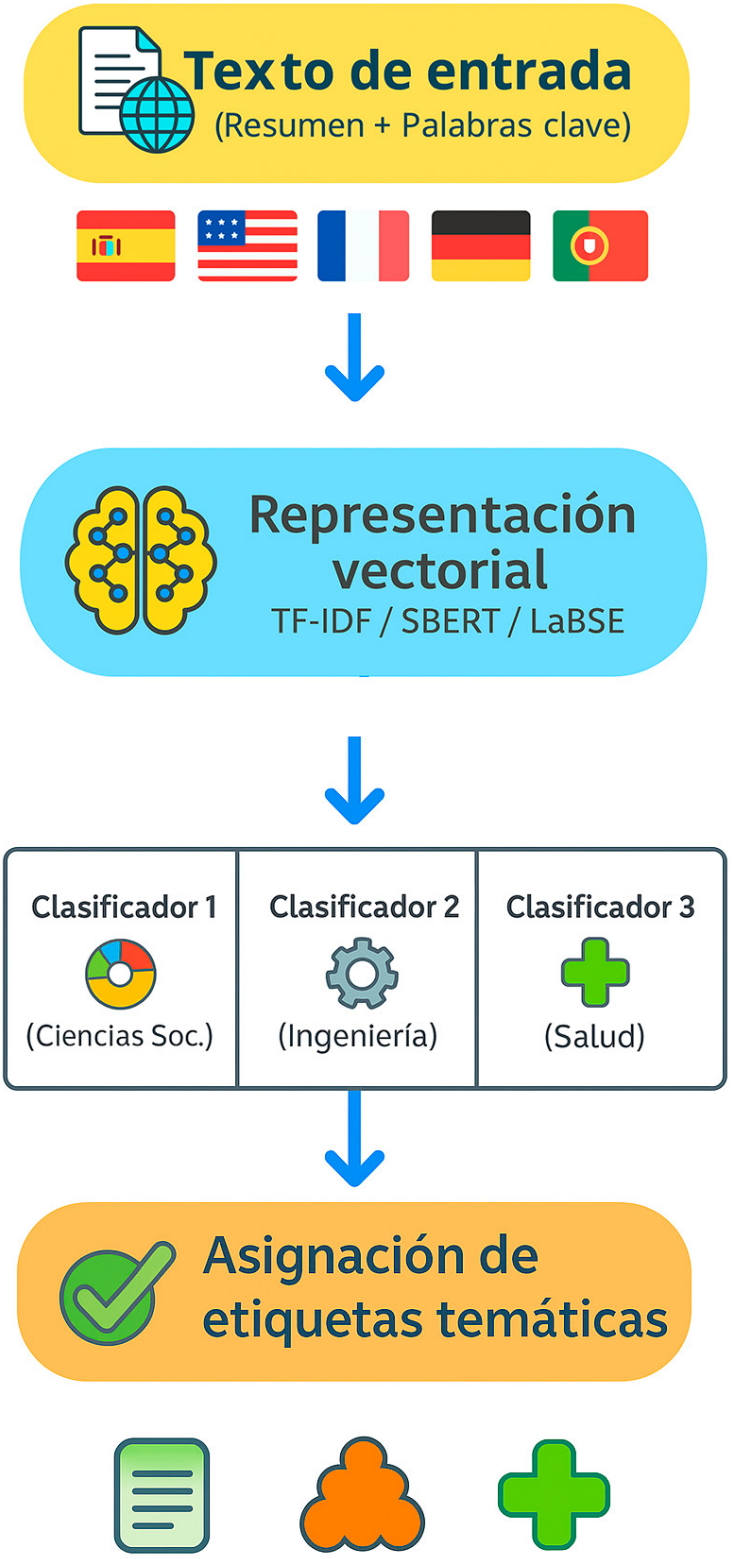
Figura 5 - Distribución aproximada de palabras en un espacio vectorial tridimensional. Las palabras con significados similares tienden a agruparse en la representación espacial, permitiendo al modelo captar relaciones semánticas más allá de la coincidencia textual. En el caso de LABSE y SBERT la representación se realiza a nivel de sentencia.



Fuente: Elaboración propia (2025).

La estrategia *One vs Rest* permite entrenar un clasificador independiente para cada etiqueta: uno por uno, decide si esa etiqueta debe asignarse o no a cada ejemplo. Por ejemplo, si se necesita determinar si un resumen pertenece a *Ciencias Sociales*, se entrena un modelo que responde sí o no para esa etiqueta, sin considerar las demás. Este proceso se repite por separado para *Ciencias Naturales*, *Ingeniería*, etc. Así, un mismo texto puede recibir varias etiquetas si los clasificadores correspondientes lo indican. Esta estrategia resulta muy útil cuando se utilizan algoritmos diseñados originalmente para clasificación binaria, como la regresión logística o las máquinas de vectores de soporte (SVM). Las tareas se organizaron en cuatro bloques experimentales, según el tipo de representación utilizada.

Figura 6 - Este diagrama muestra cómo texto multilingüe (resumen + palabras clave) es convertido en una representación numérica, y luego evaluado por un clasificador diferente para cada materia temática. Cada clasificador decide de forma independiente si asigna o no su etiqueta.



Fuente: Elaboración propia (2025).

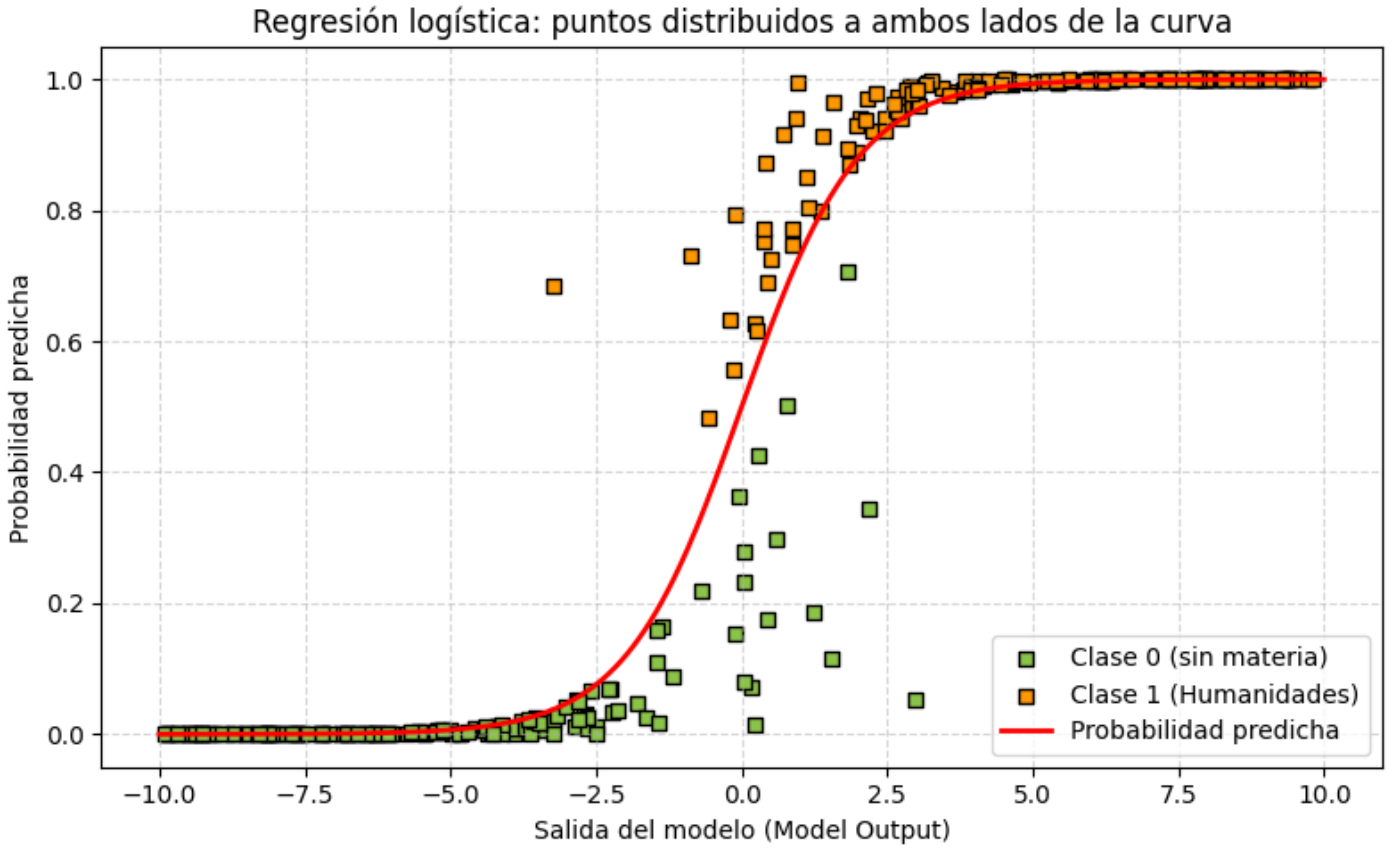
BREVE DESCRIPCIÓN DE LOS CLASIFICADORES UTILIZADOS

Para evaluar el rendimiento de las distintas representaciones vectoriales en la tarea de clasificación se utilizaron cinco familias de algoritmos de aprendizaje automático supervisado. Cada uno de ellos aporta enfoques diferentes para el modelado de la relación entre entradas y etiquetas, y presenta ventajas según el tipo de datos y representación empleada.

REGRESIÓN LOGÍSTICA

La regresión logística (Jerome Hastie, 2003; Pedregosa et al., 2011) es un modelo lineal que estima la probabilidad de que una observación pertenezca a una determinada clase, utilizando la función sigmoide para proyectar la salida a un rango entre 0 y 1 (por ejemplo, si un resumen del repositorio debe ser etiquetado con “Ciencias Sociales” o no). Si el resultado está cerca de 0, el modelo cree que no corresponde asignar la materia. Si el resultado está cerca de 1, el modelo cree que sí corresponde. En su versión básica, es un clasificador binario, aunque puede extenderse a problemas multilabel mediante la estrategia *One-vs-Rest*. Es particularmente eficaz cuando las clases son linealmente separables. Es decir, si en un plano es posible trazar una línea recta que separe perfectamente los resúmenes que deben etiquetarse como *Educación* de los que no.

Figura 7. Gráfico con ejemplo de Regresión Logística: cada punto representa un resumen clasificado por el modelo. La curva roja muestra la función sigmoide que estima la probabilidad de pertenecer a una materia. Los puntos naranjas (clase 1, pertenecientes a la materia Humanidades) aparecen principalmente por encima de la curva, mientras que los verdes (clase 0) se ubican por debajo.

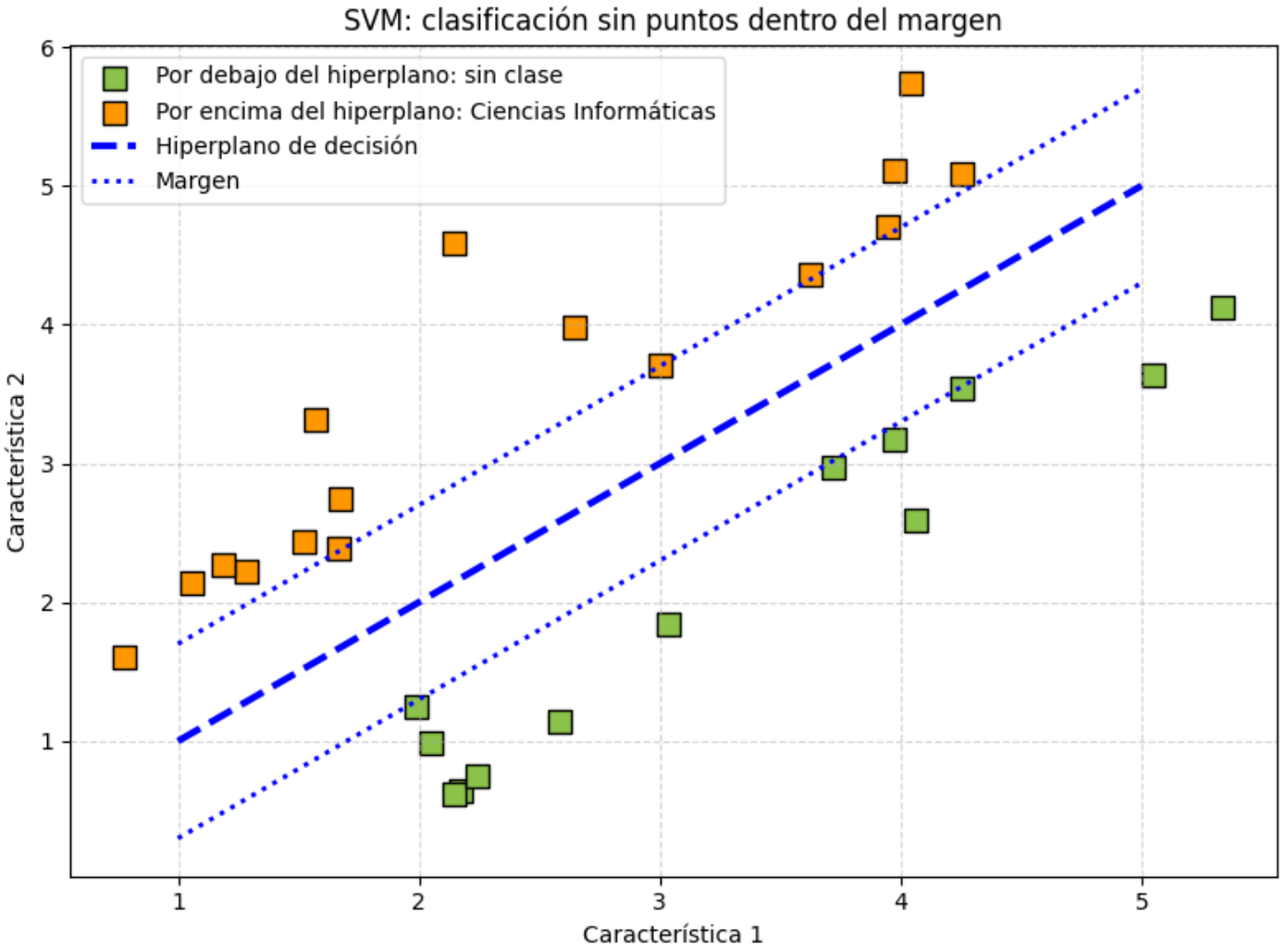


Fuente: Elaboración propia (2025).

Las máquinas de soporte vectorial (Fan et al., 2008) son clasificadores que buscan encontrar el hiperplano que maximiza el margen entre clases en el espacio de características. En su versión lineal son apropiadas para problemas de alta dimensionalidad como los que resultan de la vectorización de texto. Una de sus principales fortalezas es su capacidad para generalizar bien en contextos donde los datos no están claramente separados.

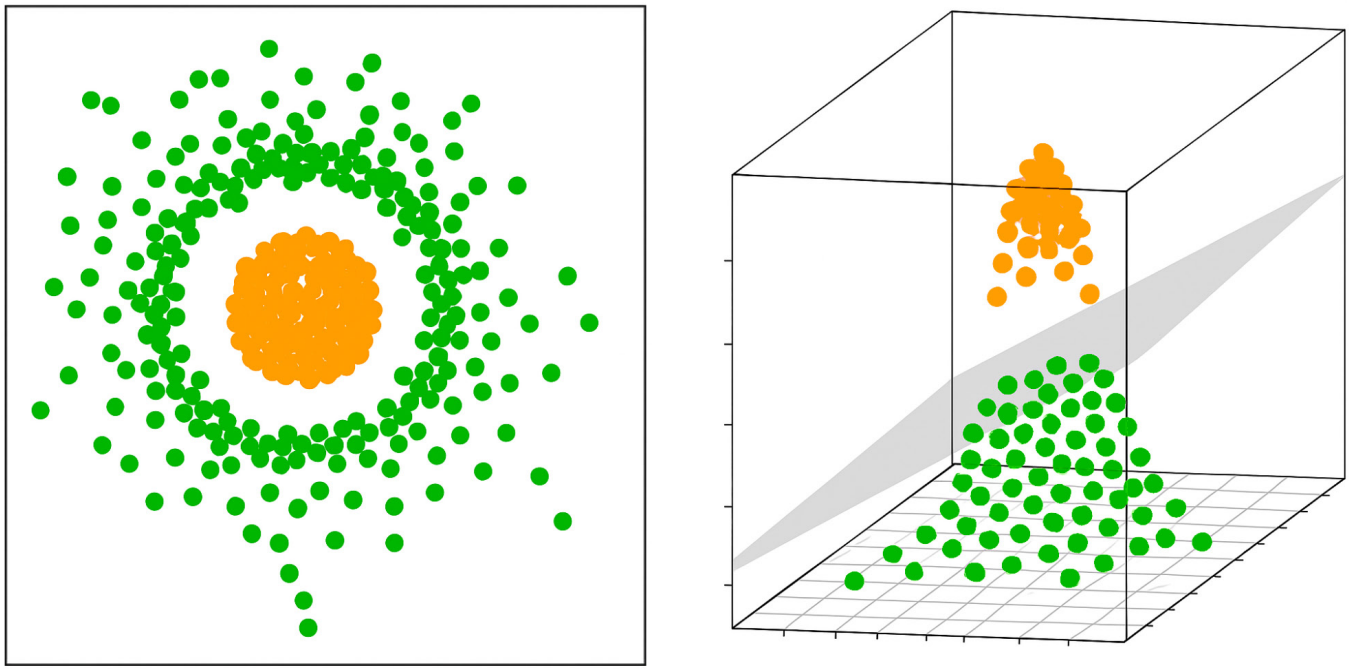
Por ejemplo, si se tienen fichas de resúmenes con dos valores: cuántas veces aparece el término *datos* y cuántas veces *educación*. Cada resumen se ubica como un punto en el plano. La SVM traza una línea (o hiperplano) entre las dos clases (por ejemplo: *Educación* vs *Sin materia*) y maximiza el espacio libre a ambos lados de la línea. Es decir, deja la mayor *zona de seguridad* posible entre las clases. Esto permite a la SVM generalizar bien, incluso si hay cierto ruido o los puntos no están perfectamente separados. Por eso, se usa mucho en clasificación de textos donde hay muchas variables y no todo es blanco o negro.

Figura 8 - Gráfico con ejemplo máquinas de soporte vectorial (SVM). El hiperplano de decisión (línea azul discontinua) separa claramente las dos clases, y las líneas punteadas marcan el margen de seguridad. Los cuadrados verdes representan ítems que no reciben ninguna etiqueta, y los naranjas corresponden a documentos clasificados en Ciencias Informáticas.



Fuente: Elaboración propia (2025).

Figura 9. Gráfico con ejemplo máquinas de soporte vectorial (SVM) en el que se transforma el espacio para que lo que no se puede separar con una línea recta en dos dimensiones en el gráfico, se pueda separar con una recta proyectada como un plano sumando una dimensión extra (Amat Rodrigo, 2020).



Fuente: Elaboración propia (2025).

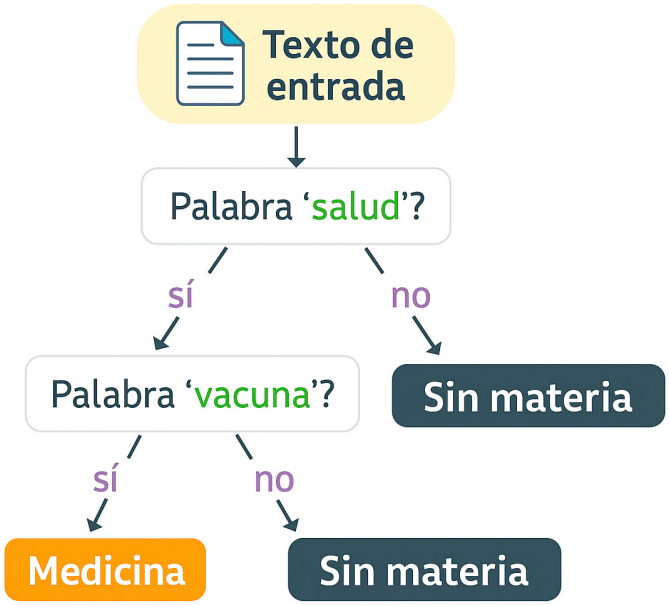
BOSQUES ALEATORIOS (RANDOM FOREST)

Los bosques aleatorios (Breiman, 2001) son un método de ensamblado basado en árboles de decisión que consiste en una colección de árboles entrenados sobre subconjuntos aleatorios del conjunto de entrenamiento y de las características, y cuya predicción final se basa en el voto mayoritario de los árboles. Esta técnica reduce la varianza del modelo base (el árbol de decisión) y mejora la generalización con datos heterogéneos y no lineales. Cada árbol de decisión dentro del bosque actúa como un pequeño cuestionario jerárquico, que hace preguntas simples una tras otra hasta llegar a una conclusión. Estas preguntas se basan en atributos del documento, por ejemplo:

1. ¿Contiene la palabra *salud* más de 3 veces?
2. ¿Tiene la palabra *sistema*?

A medida que se responde a estas preguntas, el árbol guía el ítem por una rama diferente hasta llegar a una hoja, que representa la predicción de ese árbol.

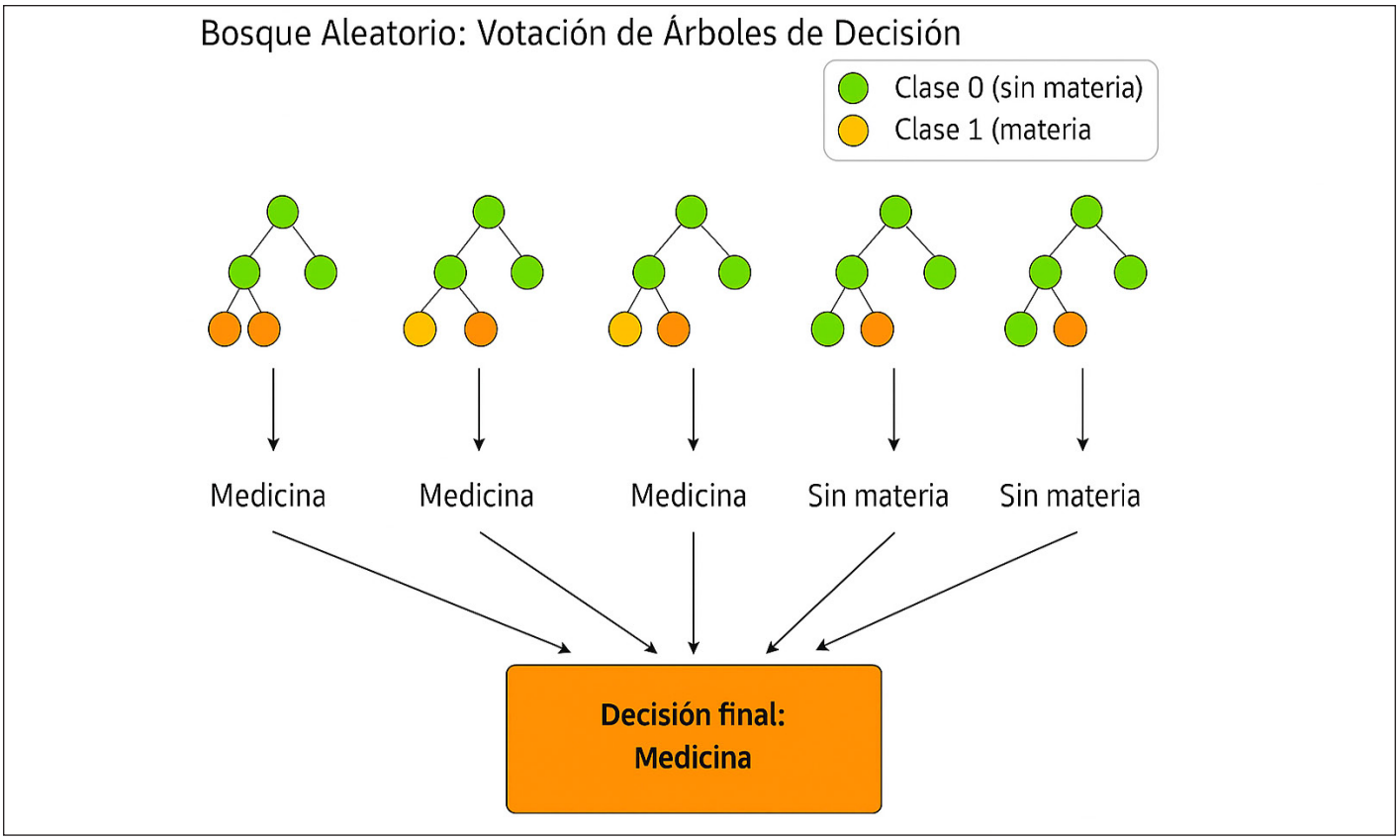
Figura 10 - Gráfico con ejemplo de funcionamiento de un árbol de decisión en el que la pertenencia a una clase se define por las palabras que contiene cada ítem.



Fuente: Elaboración propia (2025).

Un bosque aleatorio funciona como un equipo de catalogadores automáticos. Cada árbol da su opinión sobre a qué materia debería asignarse un documento, basándose en una muestra distinta del conjunto de datos y variables. Una vez que todos han *votado*, se toma la decisión por mayoría: si la mayoría de los árboles vota por *Clase 1* (por ejemplo, *Ciencias de la Información*), entonces el sistema asigna esa materia. Esta estrategia de combinación de decisiones reduce errores individuales y mejora la capacidad del modelo para generalizar cuando los datos son diversos, poco estructurados o no se separan claramente.

Figura 11 - Gráfico con ejemplo de bosques aleatorios y los resultados obtenidos por votación.



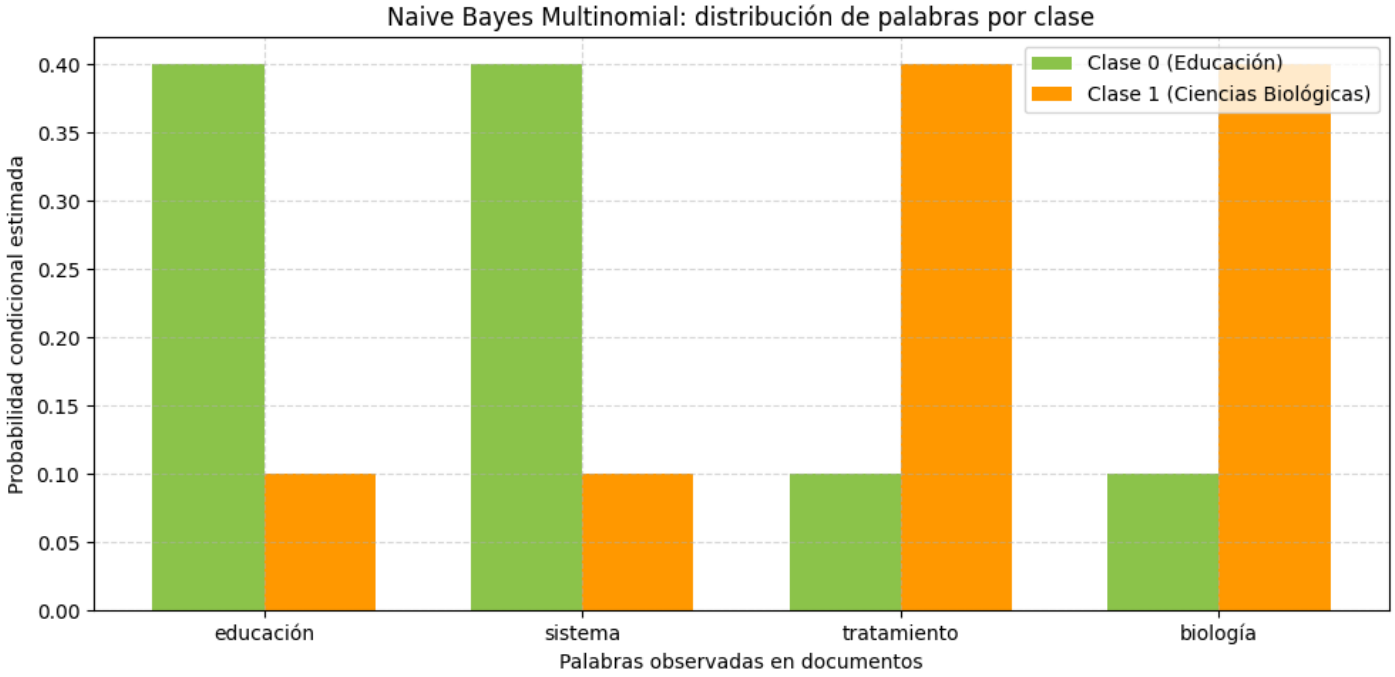
Fuente: Elaboración propia (2025).

NAIVE BAYES MULTINOMIAL

Se trata de un modelo probabilístico (Rennie et al., 2003) especialmente adecuado para datos discretos como los generados por técnicas de *bag-of-words* o *n-gramas*. Este modelo asume que las características (en este caso, palabras o n-gramas) son condicionalmente independientes entre sí dada cada clase, y estima la probabilidad de cada etiqueta en función de la frecuencia de aparición de las palabras en los documentos. Su simplicidad y eficiencia lo convierten en una opción atractiva para tareas de clasificación de texto. Sin embargo, como su rendimiento suele verse limitado en contextos multilabel con alta correlación entre etiquetas, o cuando se utilizan representaciones semánticas densas como los embeddings, no se utilizó para clasificar esas representaciones, sólo para los casos de Bow y TF-IDF.

En nuestro caso, Naive Bayes Multinomial funciona como un sistema que cuenta cuántas veces aparece cada palabra en los documentos de cada clase (por ejemplo, materias). Luego, calcula la probabilidad de que un nuevo documento pertenezca a cada clase en función de las palabras que contiene. Por ejemplo: si un documento contiene *educación* y *sistema*, y esas palabras aparecen con frecuencia en documentos etiquetados como *Educación*, el modelo probablemente asigne esa clase. Si en cambio contiene *tratamiento* y *biología*, podría ser clasificado como *Ciencias Biológicas*.

Figura 12. Gráfico con ejemplos de distribución de palabras por clase en Naive Bayes Multinomial. El modelo estima la probabilidad de cada clase en función de la frecuencia con que aparecen ciertas palabras en los documentos previamente clasificados. En este ejemplo, *educación* y *sistema* son más frecuentes en la Clase *Educación*, mientras que *tratamiento* y *biología* predominan en la Clase *Ciencias Biológicas*.



Fuente: Elaboración propia (2025).

VARIANTES DEL DESCENSO DE GRADIENTE ESTOCÁSTICO (SGD)

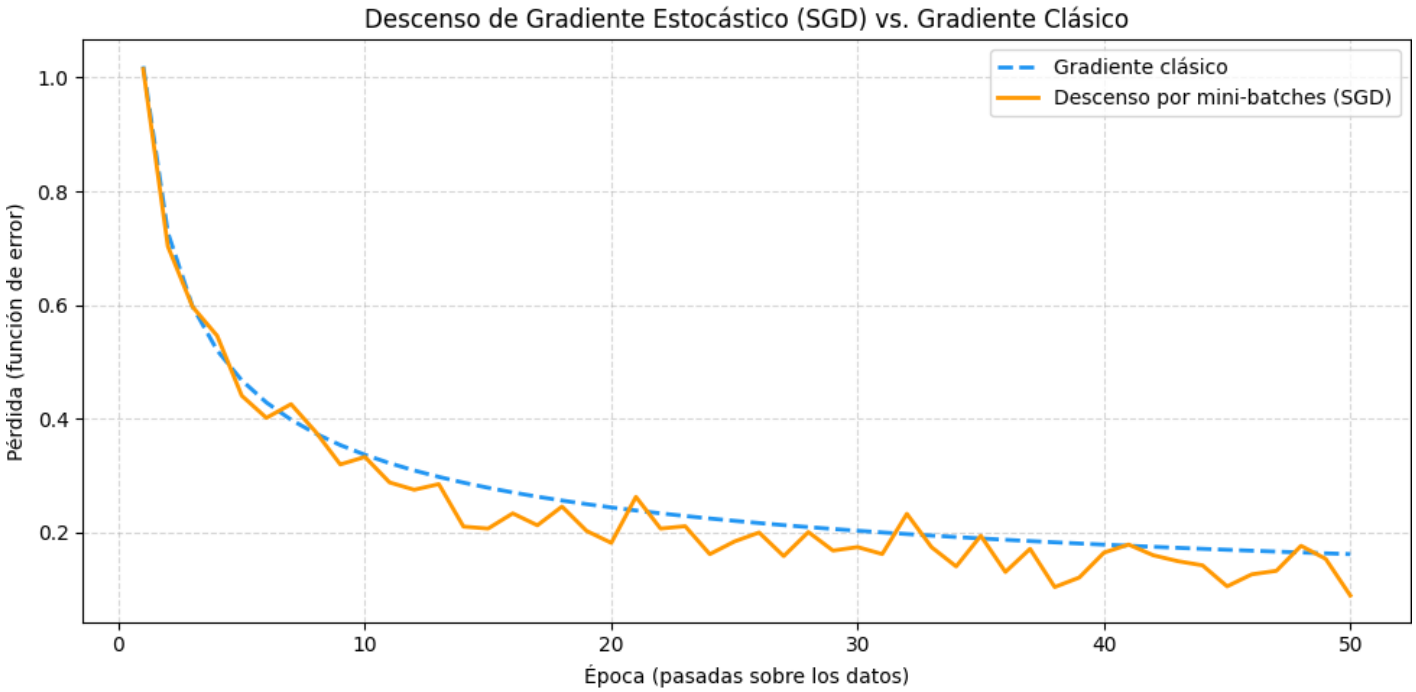
Los clasificadores que emplean descenso de gradiente estocástico (SGD, por sus siglas en inglés) (Bottou, 2010) optimizan una función de pérdida de manera iterativa ajustando los pesos del modelo a partir de ejemplos individuales o pequeños lotes de datos (mini-batches). El modelo aprende paso a paso, ajustando sus parámetros cada vez que ve un nuevo ejemplo o un pequeño grupo (mini-batch). Esto permite hacer actualizaciones rápidas y frecuentes, lo que lo hace muy útil para colecciones grandes, como un repositorio con miles de resúmenes. Se llama *descenso de gradiente* porque se basa en operaciones que buscan reducir un error paso a paso, como si uno descendiera por una montaña buscando el punto más bajo. Y se denomina *estocástico* porque no mira todos los datos juntos, sino que aprende a partir de ejemplos individuales o pequeños grupos.

A diferencia del descenso de gradiente clásico, que calcula el gradiente utilizando todo el conjunto de entrenamiento, el enfoque estocástico actualiza los parámetros con base en una muestra mucho más pequeña, lo que permite realizar actualizaciones más rápidas y fre-

cuentes. Esta característica lo convierte en una técnica especialmente eficiente desde el punto de vista computacional y altamente escalable[3].

El método clásico leería todos los resúmenes y haría estadísticas globales, es muy preciso pero también lento y costoso. Con la estrategia SGD se van leyendo resúmenes uno por uno, y se ajustan los criterios de clasificación a medida que se avanza. Es más rápido y flexible: aprende paso a paso, sin esperar a tener toda la información ya que se adapta en tiempo real. El modelo va aprendiendo de cada ejemplo y después de muchas iteraciones, el modelo aprende a reconocer que ciertos términos están asociados a ciertas materias.

Figura 13. Gráfico con ejemplo de descenso de gradiente clásico (línea azul suave) y el SGD (línea naranja con saltos).



Fuente: Elaboración propia (2025).

REPRESENTACIONES UTILIZADAS

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

Se utilizó el vectorizador *Term Frequency-Inverse Document Frequency* con un máximo de 5000 características, considerando n-gramas de 1 a 2 palabras. Esta representación fue aplicada sobre todas las columnas de texto relevantes (abstract, dc.subject, etc.).

[3] Las abreviaciones utilizadas para los clasificadores corresponden a las clases implementadas en la biblioteca scikit-learn (Pedregosa et al., 2011): LogReg hace referencia a Logistic Regression, clasificador de regresión logística; RandomForest corresponde a Random Forest Classifier, basado en bosques aleatorios; LinearSVC refiere a máquina de soporte vectorial lineal; MultinomialNB alude al clasificador Naive Bayes para datos discretos; y SGDClassifier al clasificador que entrena modelos lineales utilizando descenso de gradiente estocástico.

FRECUENCIA DE N GRAMAS

Se repitió la misma arquitectura, pero reemplazando TF-IDF por una representación basada en frecuencias absolutas, también con 5000 n-gramas máximo. Esta variante permite observar el impacto de la normalización TF-IDF frente al conteo puro en tareas multilabel.

EMBEDDINGS SBERT

Se utilizó el modelo SBERT (distiluse-base-multilingual-cased-v1), con codificación de oraciones directamente en vectores densos (embeddings). Estos vectores fueron generados en GPU para cada columna textual relevante y luego concatenados. No se incluyó el clasificador Multinomial Naive Bayes (MultinomialNB) en esta etapa, ya que este algoritmo, por diseño, está optimizado para trabajar con entradas basadas en conteos discretos no negativos (como las matrices generadas mediante Bag of Words o TF-IDF). Los embeddings densos, en cambio, son vectores continuos que pueden contener valores negativos y no representan frecuencias, lo que viola las suposiciones fundamentales del modelo multinomial.

EMBEDDINGS LABSE

La última etapa de clasificación consistió en una réplica de la arquitectura anterior, utilizando el modelo LaBSE (Language-Agnostic BERT Sentence Embedding), entrenado sobre más de 100 idiomas, como vectorizador semántico universal. También se ejecutó en GPU y sin Naive Bayes.

RESULTADOS OBTENIDOS Y ANÁLISIS COMPARATIVO

Se trabajó en el entorno Google Colab con la siguiente configuración de hardware: Intel(R) Xeon(R) CPU @ 2.20GHz, con 6 núcleos físicos y 12 núcleos lógicos, acompañado por una memoria RAM de 89,63 GB. Además, el sistema contaba con una GPU NVIDIA A100-SXM-4-40GB. Para cada combinación de representación y clasificador, se reportaron las siguientes métricas:

- Accuracy: mide el porcentaje de instancias cuya predicción coincide exactamente con el conjunto completo de etiquetas reales. Es una métrica muy exigente en contextos multilabel, ya que requiere que todas las etiquetas asignadas sean correctas y que no falte ninguna. Por este motivo, tiende a penalizar fuertemente los errores parciales y suele ofrecer valores conservadores en comparación con otras métricas.

- F1 macro: corresponde al promedio del F1-score[4] calculado individualmente para cada clase, sin ponderar por la frecuencia de las etiquetas. Esto significa que trata a todas las etiquetas por igual, independientemente de cuán frecuentes o infrecuentes sean. Es útil para evaluar el rendimiento del modelo sobre etiquetas minoritarias, que podrían ser ignoradas si solo se consideraran métricas globales[5].
- F1 micro: calcula el F1-score considerando la suma global de verdaderos positivos, falsos positivos y falsos negativos sobre todas las etiquetas y ejemplos. Es una métrica que favorece el rendimiento sobre las clases más frecuentes, ya que acumula todos los aciertos y errores sin distinguir entre etiquetas. Proporciona una visión agregada del desempeño del sistema[6].
- Tiempo de entrenamiento: indica la cantidad total de tiempo que requirió el modelo para ser entrenado sobre el conjunto de datos. Esta métrica permite analizar la eficiencia computacional de cada combinación de vectorizador y clasificador, y resulta relevante al comparar enfoques clásicos frente a representaciones densas como embeddings, que suelen implicar mayores costos de cómputo.

DATASET FILTRADO CON 61 ETIQUETAS

Los resultados obtenidos sobre el conjunto de datos con 61 etiquetas se resumen en la siguiente tabla.

Tabla 2 - Comparación del rendimiento de los clasificadores según el vectorizador utilizado para el dataset filtrado con 61 etiquetas.

Vectorizador	Clasificador	Accuracy	F1_macro	F1_micro	Tiempo
tfidf	LogReg	0.3804006758	0.4204315051	0.5690942851	0h 7m 0s
tfidf	RandomForest	0.1848097192	0.1945616363	0.3270184228	0h 22m 14s
tfidf	**LinearSVC**	**0.4353528039**	**0.5159955184**	**0.6174616126**	**0h 8m 27s**
tfidf	MultinomialNB	0.132673586	0.1122102976	0.2813951371	0h 0m 57s

[4] El F1-score es la media armónica entre la precisión (precision) y la exhaustividad (recall). La precisión mide la proporción de etiquetas predichas correctamente sobre el total de etiquetas asignadas por el modelo, mientras que la exhaustividad indica la proporción de etiquetas correctamente predichas sobre el total de etiquetas reales. El F1-score penaliza tanto los falsos positivos como los falsos negativos, lo que lo hace especialmente útil en contextos de desbalance de clases.

[5] El F1 macro se calcula obteniendo el F1-score de cada etiqueta por separado (como si cada una fuera una tarea independiente), y luego haciendo el promedio. Como no tiene en cuenta cuán frecuente es cada etiqueta, da el mismo peso a todas. Es útil para evaluar si el modelo está funcionando bien incluso con las etiquetas menos representadas.

[6] El F1 micro suma todos los verdaderos positivos, falsos positivos y falsos negativos de todas las etiquetas, y calcula un único F1-score a partir de esos totales. Este enfoque da más peso a las etiquetas frecuentes, ya que considera el desempeño global en lugar del desempeño por clase. Si el modelo clasifica muy bien las etiquetas frecuentes, el F1 micro será alto, incluso si ignora o falla en las etiquetas menos frecuentes.

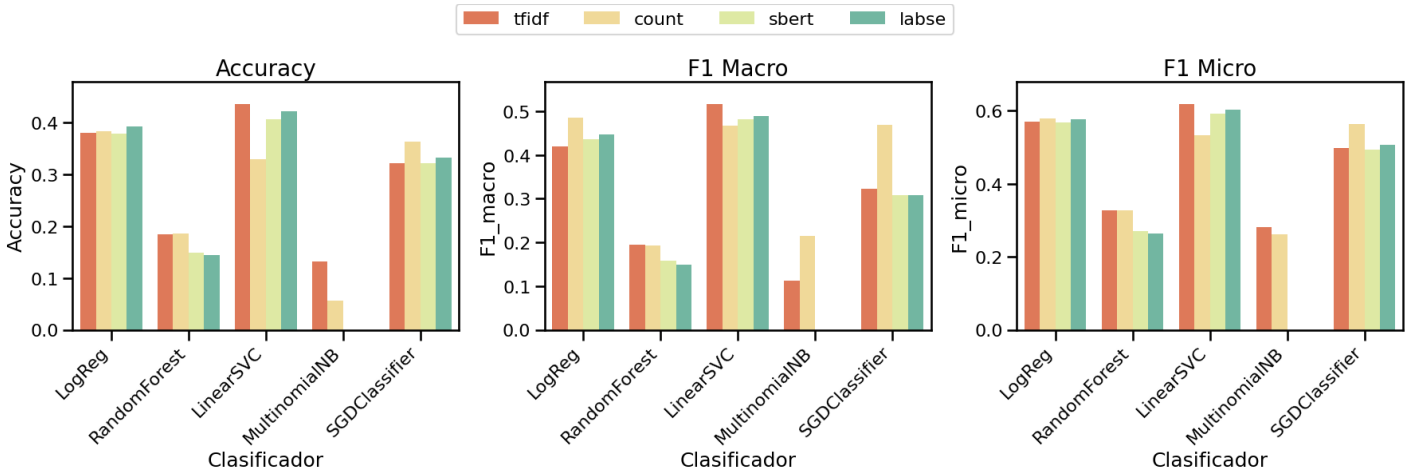
Vectorizador	Clasificador	Accuracy	F1_macro	F1_micro	Tiempo
tfidf	SGDClassifier	0.32110387	0.3233739802	0.4973750905	0h 1m 23s
count	**LogReg**	**0.3833775847**	**0.4852449442**	**0.5781777278**	**0h 35m 46s**
count	RandomForest	0.1856142892	0.1931265884	0.3277030549	0h 21m 17s
count	LinearSVC	0.3287472846	0.4669723875	0.5323772273	1h 17m 22s
count	MultinomialNB	0.056561268	0.21486296	0.2612229362	0h 0m 56s
count	SGDClassifier	0.3633437927	0.4685925754	0.5643639369	0h 2m 53s
sbert	LogReg	0.3792742779	0.4353536134	0.5668771873	4h 6m 34s
sbert	RandomForest	0.1487649851	0.157917617	0.2701765789	0h 26m 16s
sbert	**LinearSVC**	**0.4069514844**	**0.4817192664**	**0.5922658896**	**2h 45m 56s**
sbert	SGDClassifier	0.3218279829	0.3076033934	0.4942959002	0h 19m 42s
labse	LogReg	0.3919060262	0.4469154693	0.5759392486	6h 17m 19s
labse	RandomForest	0.1441789364	0.1482616591	0.2642721088	0h 36m 34s
labse	**LinearSVC**	**0.4220773996**	**0.4883060378**	**0.6035484253**	**4h 17m 1s**
labse	SGDClassifier	0.3317241934	0.3078241741	0.5068319728	0h 33m 31s

Fuente: Elaboración propia (2025).

En una primera comparación de resultados, la combinación de TF-IDF con Linear SVC se destacó como la opción más equilibrada entre los métodos clásicos, alcanzando los mejores valores tanto de F1 macro (0.51) como de F1 micro (0.61), con un tiempo de entrenamiento razonable y un rendimiento general robusto. Por otro lado, los modelos basados en embeddings —en particular, LaBSE con LinearSVC y SBERT con *LinearSVC*— mostraron un desempeño competitivo, con valores de F1 micro de 0.60 y 0.59 respectivamente aunque a costa de tiempos de entrenamiento considerablemente más altos.

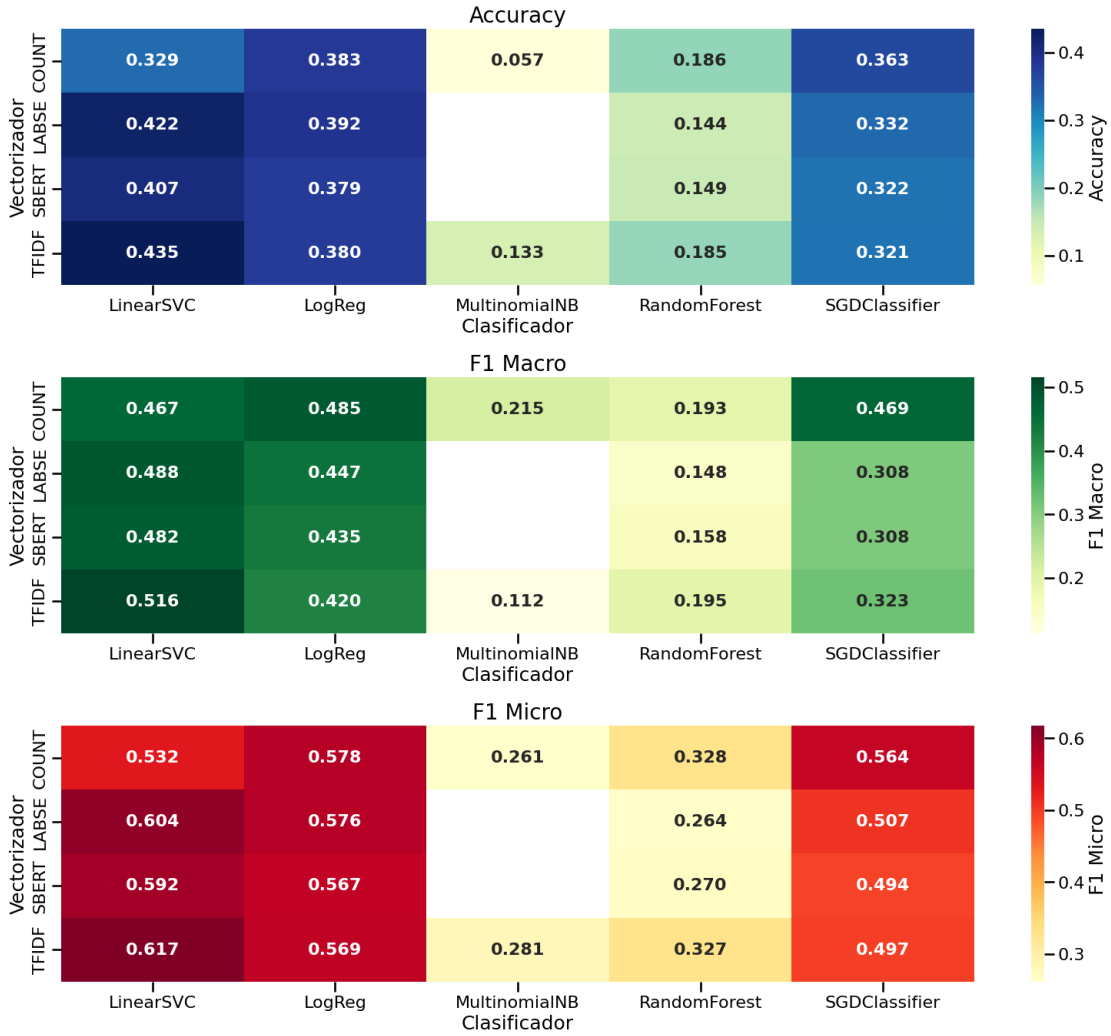
El clasificador Logistic Regression presentó un rendimiento aceptable al ser combinado con representaciones TF-IDF y Frecuencias absolutas, aunque sin alcanzar los niveles de precisión de las máquinas de soporte vectorial. En contraste, Random Forest y Naive Bayes Multinomial obtuvieron resultados notablemente inferiores en esta tarea multilabel, lo cual era previsible dada la alta dimensionalidad de las representaciones y la correlación entre etiquetas, que tienden a afectar negativamente a estos modelos. Finalmente, el *SGDClassifier* demostró ser una alternativa eficiente en términos de tiempo de entrenamiento, con resultados aceptables especialmente cuando se empleó en conjunto con frecuencias absolutas.

Figura 14 - Comparación del rendimiento de los clasificadores según el vectorizador para el dataset filtrado con 61 etiquetas.



Fuente: Elaboración propia (2025).

Figura 15. Mapas de calor del rendimiento por combinación de vectorizador y clasificador para el dataset filtrado con 61 etiquetas. Las celdas de colores más oscuros indican un mejor desempeño del clasificador combinado con el tipo de representación correspondiente.



Fuente: Elaboración propia (2025).

DATASET FILTRADO CON 37 ETIQUETAS

Los resultados obtenidos al aplicar distintas combinaciones de representaciones y clasificadores sobre el conjunto de datos filtrado con 37 etiquetas se resumen en la Tabla 3. Esta configuración, al reducir la cantidad de etiquetas, permite al sistema concentrarse en un subconjunto más representativo y frecuente, lo que suele reflejarse en un desempeño general más robusto y tiempos de entrenamiento más eficientes en algunos modelos.

Tabla 3. Comparación del rendimiento de los clasificadores según el vectorizador utilizado para el dataset filtrado con 37 etiquetas.

Vectorizador	Clasificador	Accuracy	F1_macro	F1_micro	Tiempo_train
tfidf	LogReg	0.414101799	0.5401501122	0.599964708	0h 4m 49s
tfidf	RandomForest	0.2001875693	0.2579565107	0.349822725	0h 17m 45s
tfidf	**LinearSVC**	**0.4716514622**	**0.6134127905**	**0.6470494924**	**0h 5m 52s**
tfidf	MultinomialNB	0.150481712	0.1808297026	0.3057156815	0h 0m 51s
tfidf	SGDClassifier	0.3552732543	0.4494409885	0.5368069989	0h 1m 8s
count	**LogReg**	**0.4195583596**	**0.577373627**	**0.605096023**	**0h 25m 42s**
count	RandomForest	0.2028305908	0.2594344438	0.3517893315	0h 17m 41s
count	LinearSVC	0.3637991304	0.5463004395	0.5599136623	0h 50m 52s
count	MultinomialNB	0.06795123199	0.2995391975	0.2886872792	0h 0m 50s
count	SGDClassifier	0.4086452383	0.5658741826	0.592598216	0h 2m 11s
sbert	LogReg	0.4245886265	0.5499867177	0.6063224537	2h 46m 10s
sbert	RandomForest	0.1649757013	0.2105144278	0.2961491264	0h 19m 8s
sbert	**LinearSVC**	**0.4474379743**	**0.5677293155**	**0.626302521**	**2h 2m 3s**
sbert	SGDClassifier	0.362520249	0.4416031875	0.5421461315	0h 15m 23s
labse	LogReg	0.4238212976	0.5516241022	0.6072237658	4h 6m 40s
labse	RandomForest	0.1538068036	0.1921932834	0.2834413671	0h 28m 33s
labse	**LinearSVC**	**0.4545144514**	**0.5689937902**	**0.6313212326**	**3h 3m 21s**
labse	SGDClassifier	0.3517776452	0.4279714493	0.5289469904	0h 25m 53s

Fuente: Elaboración propia (2025).

En esta configuración, de menos etiquetas, la combinación de TF-IDF con *LinearSVC* volvió a destacarse como una de las más equilibradas, alcanzando los mejores valores de F1 macro (0.61) y F1 micro (0.64), con una mejora leve respecto al dataset anterior. Esta mejora en el rendimiento podría atribuirse a la menor dispersión de etiquetas y a una representación más densa del espacio de clases, lo que favorece los clasificadores de margen como SVC. Los

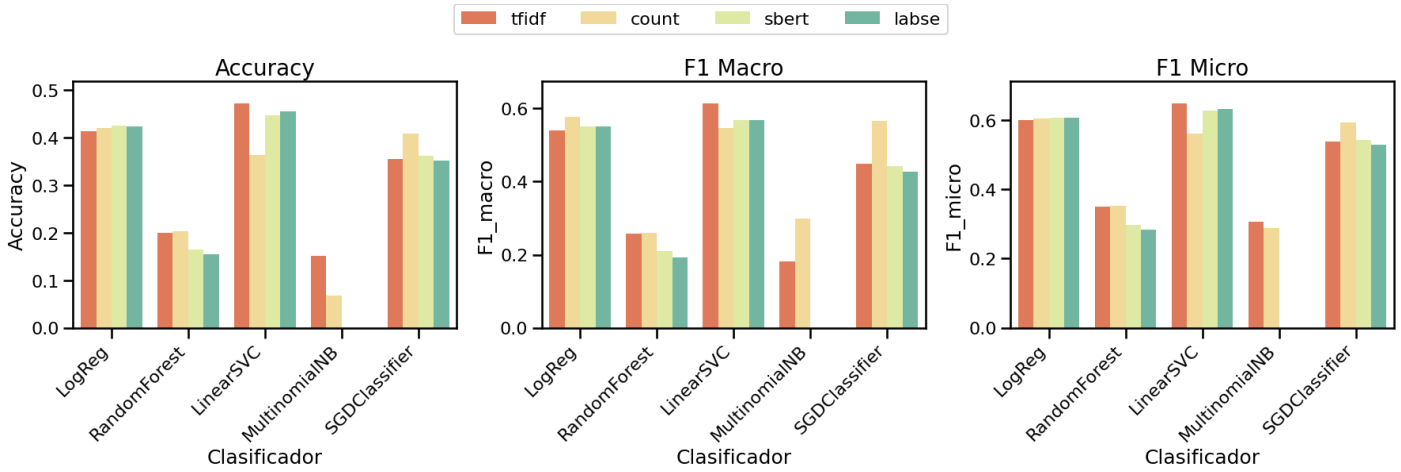
embeddings densos (LaBSE y SBERT) combinados con *LinearSVC* mostraron nuevamente un rendimiento competitivo. La combinación de LaBSE + *LinearSVC* logró el mejor F1 micro global (0.63), ligeramente superior al obtenido con 61 etiquetas. Sin embargo, como se observó anteriormente, estos modelos implican un costo computacional alto, superando las 2 a 4 horas de entrenamiento en GPU, dependiendo del clasificador.

El clasificador Logistic Regression mostró un ligero aumento de rendimiento al combinarse con SBERT y LaBSE en este nuevo conjunto, alcanzando F1 macro y micro superiores al 0.55 y 0.60 respectivamente, ubicándose como una opción confiable aunque sin superar a SVC. En contraste, Random Forest y Naive Bayes continuaron ofreciendo resultados limitados, especialmente en combinación con embeddings.

Un resultado interesante es el aumento del desempeño de *SGDClassifier* cuando se combina con frecuencias absolutas (*CountVectorizer*), obteniendo valores de F1 macro (0.56) y F1 micro (0.59) notoriamente superiores a los obtenidos con TF-IDF y comparables a Logistic Regression. Además, con tiempos de entrenamiento por debajo de los 3 minutos, se posiciona como una alternativa eficiente y escalable.

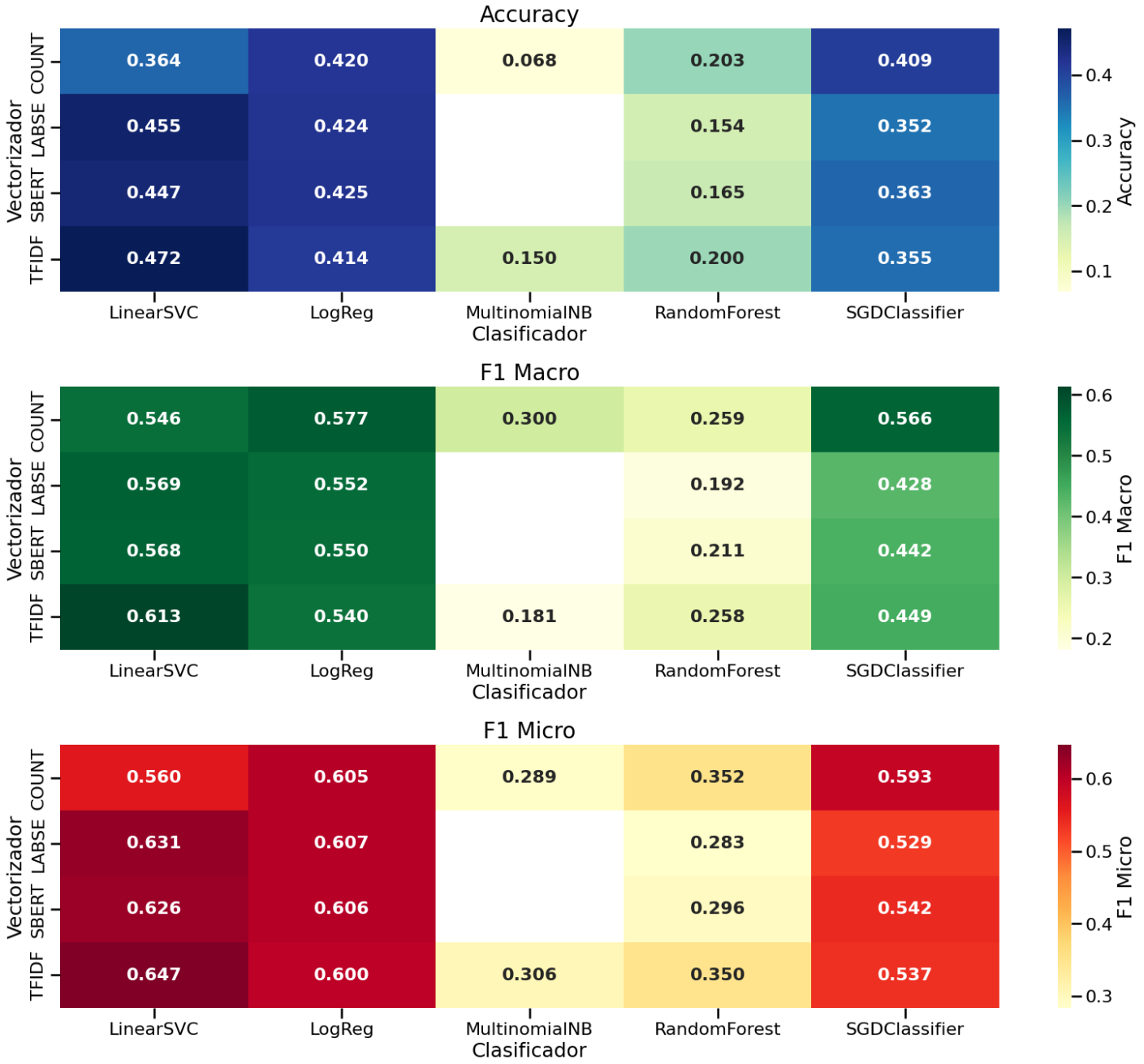
En términos generales, el pasaje de 61 a 37 etiquetas mejoró los resultados de casi todas las combinaciones, redujo la complejidad del espacio de predicción y permitió tiempos de entrenamiento ligeramente menores en varios clasificadores. Una estrategia de selección de etiquetas basada en frecuencia o representatividad puede ser beneficiosa en contextos multilabel con alta dispersión.

Figura 16 - Comparación del rendimiento de los clasificadores según el vectorizador utilizado con 37 etiquetas.



Fuente: Elaboración propia (2025).

Figura 17 - Mapas de calor del rendimiento por combinación de vectorizador y clasificador con 37 etiquetas. Las celdas de colores más oscuros indican un mejor desempeño del clasificador combinado con el tipo de representación correspondiente.



Fuente: Elaboración propia (2025).

DISCUSIÓN

Los resultados obtenidos sugieren que la combinación de representaciones clásicas (TF-IDF) con clasificadores lineales como *LinearSVC* sigue siendo una de las alternativas más robustas para la clasificación multilabel en casos como el que acabamos de mostrar. A pesar

de su simplicidad, esta configuración logró desempeños comparables —e incluso superiores en ciertos casos— a los obtenidos con representaciones más costosas computacionalmente, como SBERT o LaBSE.

El uso de embeddings contextuales permitió mejorar la detección de etiquetas minoritarias, especialmente al combinarse con SVM o regresión logística, pero los tiempos de entrenamiento elevados limitan su aplicabilidad directa en flujos de trabajo con recursos computacionales restringidos. Se hace necesario sopesar las diferentes ventajas y dificultades al optar entre priorizar el rendimiento y la eficiencia.

Una limitación detectada tiene que ver con la escasa representación de muchas etiquetas en el dataset original. Si bien el análisis de cobertura acumulada y la selección de subconjuntos más frecuentes ayudaron a mitigar esta situación, futuras líneas de trabajo podrían incluir estrategias de aumento de datos, remuestreo (*resampling*) o generación sintética de ejemplos minoritarios.

El trabajo hasta aquí realizado es inicial y exploratorio y no contempla aún una validación cruzada exhaustiva ni optimización hiper paramétrica sistemática, lo cual deja abierto un margen de mejora adicional en términos de ajuste fino de los modelos.

Finalmente, se podría destacar que el presente trabajo se apoya sobre resultados previos de detección automática de idioma en ítems del repositorio, y extiende esa línea de investigación hacia un nuevo campo aplicado, confirmando las ideas previas de que las herramientas de procesamiento automático pueden complementar las tareas curatoriales que se realizan en SEDICI.

CONCLUSIONES

Los resultados de este estudio vienen a reafirmar la utilidad del aprendizaje supervisado como estrategia viable para la automatización de tareas complejas de curaduría como la asignación de metadatos de materia a ítems de repositorios institucionales. A través de la comparación sistemática de múltiples combinaciones de representaciones vectoriales y clasificadores multilabel, fue posible identificar configuraciones prometedoras que logran un balance entre rendimiento y eficiencia computacional.

Por restricciones de tiempo y capacidad de procesamiento, las tareas anteriores se llevaron a cabo con configuraciones bastante básicas, tanto de las representaciones textuales como de los algoritmos clasificadores. Sin embargo, luego de la evaluación inicial de las distintas combinaciones de vectorizadores y clasificadores, se identificaron configuraciones prometedoras, especialmente aquellas que emplean TF-IDF o LaBSE combinadas con clasificadores como LinearSVC. No obstante, aún existen márgenes de mejora significativos tanto en la representación de los textos como en los modelos utilizados y en la calidad de los datos de entrada.

En futuros trabajos se implementarán mejoras para el desempeño del sistema de clasificación mediante una serie de intervenciones en tres niveles fundamentales: los datos, los clasificadores, y las representaciones vectoriales. En cuanto a la mejora en la calidad de los datos se podrían implementar nuevas técnicas de limpieza y preprocesamiento sobre los resúmenes, con el objetivo de reducir el ruido, eliminar información redundante o irrelevante, y mejorar la coherencia textual. Esto incluye normalización, eliminación de símbolos no textuales, y eventualmente segmentación por oraciones para facilitar modelos posteriores que operan sobre unidades lingüísticas finas. Para la optimización de clasificadores se pueden testear distintas variantes y configuraciones internas de cada clasificador ajustando manualmente algunos valores relevantes, como el número máximo de iteraciones en SVM y SGD, o la activación de mecanismos de parada anticipada o bien estrategias de balanceo de clases, como la asignación de pesos inversamente proporcionales a la frecuencia de cada etiqueta (*class_weight='balanced'*), con el objetivo de mitigar el impacto del fuerte desbalance en la distribución de etiquetas. Las representaciones vectoriales de los textos también son ampliamente mejorables: para frecuencias absolutas y TF-IDF, se podría explorar diferentes configuraciones de n-gramas de palabras y de caracteres para favorecer la captura de mejores patrones morfológicos y estructuras sintácticas características del lenguaje académico. En el caso de modelos basados en embeddings contextuales como SBERT y LaBSE, podrían evaluar diferentes estrategias de agregación de embeddings[7] por oración: máximo, mínimo, promedio que permitieron generar una representación semántica optimizada del resumen completo. Además, se hace necesario comparar el rendimiento de modelos entrenados con resúmenes y palabras clave por separado frente a aquellos que utilizan una combinación conjunta de ambos campos, considerando que cada tipo de texto puede aportar señales semánticas distintas para la predicción de materias.

Respecto a la interoperabilidad con vocabularios controlados de uso extendido, como los esquemas de clasificación UNESCO, OECD Fields of Science, o JEL (Journal of Economic Literature), cabe señalar que esta primera etapa se concentró exclusivamente en evaluar la viabilidad técnica de distintos métodos de clasificación automática utilizando el esquema temático propio del repositorio SEDICI. La asignación automática de materias externas implicaría un proceso de reetiquetado de todo el corpus, con criterios de correspondencia entre vocabularios que deberían ser definidos previamente, y que posiblemente requieran el diseño de modelos generativos o sistemas de recomendación semántica. Esta línea de trabajo, sin duda relevante para favorecer la interoperabilidad y estandarización entre repositorios, será objeto de futuras investigaciones.

[7] En el contexto de modelos como SBERT o LaBSE, cada oración o segmento del texto se convierte en un vector que resume su significado. Para construir una única representación del texto completo (que pueda usarse como entrada del clasificador), se aplican estrategias de agregación; es decir, se debe elegir una forma de combinar los embeddings de las distintas oraciones. Por ejemplo, se pueden promediar todos los vectores para obtener una representación general, o tomar el valor máximo o mínimo en cada dimensión. Estas estrategias pueden resaltar distintos aspectos del contenido y afectar el rendimiento del modelo.

Finalmente, en cuanto a las implicaciones prácticas para los gestores de repositorios, este trabajo contribuye a reflexionar sobre el potencial que tienen las técnicas de aprendizaje automático para mejorar la eficiencia, consistencia y escalabilidad de los procesos curatoriales, especialmente en contextos de crecimiento sostenido de las colecciones. Un sistema de clasificación automática puede integrarse a los flujos de ingestión de metadatos como una herramienta de apoyo a los catalogadores que pueda sugerir materias relevantes de manera inmediata y basada en el contenido. Esto no solo agiliza la carga de nuevos registros, sino que también permite detectar inconsistencias en asignaciones previas y facilita la actualización masiva de etiquetas temáticas cuando se adoptan nuevas taxonomías. Por último, se hace necesario destacar la importancia de la adaptación de estas herramientas a las necesidades específicas del entorno institucional, evaluando continuamente no sólo las métricas de desempeño, sino también la escalabilidad, la interpretabilidad y la facilidad de integración con los sistemas existentes de catalogación y recuperación. En última instancia, la propuesta de automatizar tareas y mejorar la calidad de los datos avanza en dirección a la construcción de un ecosistema de herramientas inteligentes que asistan la tarea de los administradores y catalogadores de los repositorios y que sean capaces de enriquecer el procesamiento documental en vista de las mejores prácticas para la preservación y difusión de la producción intelectual en el marco del acceso abierto.

DEDICATÓRIA ESPECIAL

A Nora Mora, mi profesora de matemáticas, que me tenía paciencia porque, según ella, yo era poeta.

BIBLIOGRAFÍA

Amat, J. R. (2020). *Máquinas de Vector Soporte (SVM) con Python*. Zenodo. https://zenodo.org/doi/10.5281/zenodo.10006330

Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. In Y. Lechevallier & G. Saporta (Eds.), *Proceedings of COMPSTAT'2010* (pp. 177-186). Physica-Verlag HD. https://doi.org/10.1007/978-3-7908-2604-3_16

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.*, *9*, 1871-1874. https://www.jmlr.org/papers/volume9/fan08a/fan08a.pdf

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2022). *Language-agnostic BERT Sentence*

Embedding (arXiv:2007.01852). arXiv. https://doi.org/10.48550/arXiv.2007.01852

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2ª ed.). Springer.

Nusch, C. J., Cagnina, L. C., Errecalde, M. L., Antonelli, L., & De Giusti, M. R. (2025). Detección de idiomas como tarea de curaduría de datos para repositorios institucionales: Desempeño de bibliotecas disponibles y modelos de lenguaje. In M. Garro (Ed.), *Actas de la Conferencia Internacional BIREDIAL-ISTEC 2024 (pp. 16-31).* Universidad de Costa Rica. https://sedici.unlp.edu.ar/bitstream/handle/10915/179030/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, *12*(85), 2825-2830. https://www.jmlr.org/papers/v12/pedregosa11a.html

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* (arXiv:1908.10084). arXiv. https://doi.org/10.48550/arXiv.1908.10084

Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In T. Fawcett, N. Mishra (Eds.), *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, (pp. 616-623). AAAI Press. https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf

ANEXO 1

RESUMEN BIOGRÁFICO DE LOS AUTORES

Carlos Javier Nusch

Profesor y Licenciado en Letras por la Universidad Nacional de La Plata y Máster en Humanidades Digitales por la Universidad de Educación a Distancia de España. Ha publicado varios artículos sobre trabajo académico colaborativo, repositorios digitales, digitalización de patrimonio cultural, análisis del discurso político y literatura clásica, medieval y moderna. Trabaja en el Servicio de Difusión de la Creación Intelectual (SEDICI) de la UNLP, en el Proyecto de Enlace de Bibliotecas (PREBI) y en el repositorio CIC-Digital (CICPBA). Es miembro del Comité Asesor del Centro de Servicios en Gestión de Información (CESGI) y personal del Observatorio Medioambiental La Plata (UNLP - CICPBA - CONICET). Coordina la Oficina de Relaciones Institucionales del Consorcio Iberoamericano para la Educación en Ciencia y Tecnología (ISTEC). Participa como docente colaborador ad honorem en el curso de posgrado “Bibliotecas y Repositorios Digitales. Tecnología y aplicaciones” de la Facultad de Informática de la UNLP. Ha participado en proyectos sobre Oralidad, Escritura, Humanidades Digitales Recursos Aca-

démicos, Harvesting, OAI-PMH, Visibilidad Web, Repositorios Abiertos, Producción Académica y Científica, Accesibilidad financiados por la UNLP, la CICPBA y el ISTEC. ORCID: https://orcid.org/0000-0003-1715-4228.

Leticia Cecilia Cagnina

Doctora en Ciencias de la Computación, Magíster en Ciencias de la Computación y Licenciada en Ciencias de la Computación. Se desempeña como docente investigadora en la Universidad Nacional de San Luis (UNSL). Es Profesora Adjunta en el Departamento de Informática de la Facultad de Ciencias Físico-Matemáticas y Naturales de la UNSL. Además, es Investigadora Categoría Adjunto en la Carrera de Investigador Científico y Tecnológico del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET). Su experiencia profesional se enfoca en el campo de la Informática e Inteligencia Artificial, con especialidad en Procesamiento del Lenguaje Natural (PLN). Ha realizado importantes avances en el desarrollo y aplicación de técnicas de PLN en la bioinformática y la detección automática de riesgo en la Web. Su trayectoria académica incluye la dirección y participación en proyectos de investigación en instituciones nacionales e internacionales. Es co-directora del proyecto "Aprendizaje automático y toma de decisiones en sistemas inteligentes para la Web" y ha sido parte del proyecto "Web Information Quality Evaluation Initiative" financiado por la Unión Europea. Además, ha contribuido a proyectos relacionados con la detección de depredadores sexuales en conversaciones de chat y la evaluación de la calidad de contenido web. ORCID: https://orcid.org/0000-0001-7825-2927.

Silvia Peloche

Bibliotecóloga egresada de la Facultad de Humanidades y Ciencias de la Educación de la UNLP en 2015. Actualmente se encuentra cursando la Licenciatura en Bibliotecología en esa misma facultad. Miembro del Comité de Expertos en Repositorios Digitales de la Biblioteca Electrónica de Ciencia y Tecnología para el Sistema Nacional de Repositorios Digitales. Desde 2009 se desempeña como catalogadora en SEDICI y desde 2014 en CIC-Digital. Ha brindado cursos y capacitaciones en diferentes ámbitos en su especialidad. ORCID: https://orcid.org/0009-0009-1930-6521.

Gonzalo Luján Villarreal

Doctor en Ciencias Informáticas, es director de PREBI-SEDICI de la Universidad Nacional de La Plata, director del Centro de Servicios en Gestión de Información (CESGI) de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, coordinador informático de revistas científicas de la Universidad Nacional de La Plata y profesor de la Facultad de Informática de la misma universidad. ORCID: https://orcid.org/0000-0002-3602-8211.

Ariel Lira

Licenciado en Informático por la Universidad Nacional de La Plata, desde 2006 forma parte del equipo de PREBI-SEDICI. Se especializa en repositorios digitales de publicaciones y datos, ciencia abierta y preservación digital. Participa en el desarrollo de herramientas y servicios para la gestión y difusión de la producción académica. Su trabajo contribuye a fortalecer el ecosistema de comunicación científica en acceso abierto. ORCID: https://orcid.org/0000-0003-3647-3101.

Marcelo Luis Errecalde

Profesor Exclusivo en la Universidad Nacional de San Luis, (Argentina) y dirige el Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la Facultad de Cs. Físico, Matemáticas y Naturales. Trabaja desde hace más de 20 años en temáticas vinculadas a la Inteligencia Artificial, el aprendizaje automático, la minería de textos y la Web y el Procesamiento del Lenguaje Natural. Colabora con diferentes grupos líderes de España, México, Alemania, Austria y Grecia en áreas como la calidad de la información en la web, detección de plagio, detección de depredadores sexuales en la web y determinación del perfil del autor (DPA). Actualmente, el foco de atención en la DPA se centra en la determinación del género, la edad, la orientación política y los rasgos de personalidad de los autores de documentos en la Web. Como resultado de estos trabajos de investigación se han desarrollado sistemas que son actualmente los más efectivos a nivel mundial para la detección de fallas de calidad en Wikipedia y la detección anticipada de casos de depresión y anorexia en la Web. En la actualidad, sus direcciones de tesis de postgrado se centran en la detección anticipada de riesgos en la Web (depresión, suicidio, anorexia, entre otros), integración de conocimiento externo en los modelos de aprendizaje automático y transparencia e interpretabilidad de los grandes modelos del lenguaje. ORCID: https://orcid.org/0000-0001-5605-8963.

Lucas E. Folegotto

Diseñador en Comunicación Visual por la Universidad Nacional de La Plata, Argentina. Realiza tareas de diseño en el Servicio de Difusión de la Creación Intelectual (SEDICI), el Proyecto de Enlace de Bibliotecas (PREBI) de la UNLP y en el repositorio CIC-Digital desde 2015. Desde 2016 es parte del Centro de Servicios en Gestión de Información (CESGI) y desde 2008 trabaja como docente en las carreras de Diseño en Comunicación Visual y Diseño Multimedial de la Facultad de Bellas Artes. ORCID: https://orcid.org/0009-0004-0912-7638.

Leandro Antonelli

Obtuvo el título de Licenciado en Informática en el año 1998 momento en el cual ingresó al Centro de Investigación LIFIA. En el año 2003 obtuvo el título de Magíster en Ingeniería de Software y en el 2012 el de Doctor en Ciencias Informáticas. Todos los títulos otorgados por la Universidad Nacional de La Plata. Leandro Antonelli se ha desempeñado tanto en la academia como en la industria. En la academia ha atravesado distintas instancias de la docencia, comenzando como ayudante allá por el año 1996. Actualmente se desempeña como profesor en materias de grado y posgrado, y también es Director de la carrera en Dirección de Proyectos de Tecnología Informatica en Universidad Abierta Interamericana. También realizó investigación principalmente en ingeniería de requerimientos, con publicaciones en conferencias nacionales e internacionales, como así también en revistas. En la industria ha trabajado en reparticiones públicas como así también en ámbitos privados (para clientes nacionales e internacionales). Se ha desempeñado en distintos roles, comenzando como desarrollador en el año 1993 y actualmente se desempeña como ingeniero de software, especializándose tanto en la gestión de requerimientos como en la gestión de proyectos en general (tanto ágiles – es Scrum Master certificado-, como tradicionales). ORCID: https://orcid.org/0000-0003-1388-0337.

Marisa Raquel De Giusti

Doctora en Ciencias Informáticas, Ingeniera en Telecomunicaciones y Profesora en Letras de la Universidad Nacional de La Plata (UNLP), Argentina. Es Profesora de Posgrado en la Facultad de Informática de la UNLP y Directora de la Iniciativa Liblink del Consorcio Iberoamericano para Educación en Ciencia y Tecnología (ISTEC). Reviste además como Investigadora Emérita de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC). Integra el Comité de Expertos del Sistema Nacional de Repositorios Digitales (SNRD) y el Comité Asesor en ciencia abierta y ciudadana del Ministerio de Ciencia Tecnología e Innovación de Argentina. Sus publicaciones y desarrollos localizables en el repositorio SEDICI de la UNLP recorren áreas entre las que se incluyen la gestión de la información, la preservación digital, el diseño de experimentos y otras temáticas muy diversas de investigación desarrolladas a lo largo de su extensa trayectoria científica. ORCID: https://orcid.org/0000-0003-2422-6322.

ANEXO 2

REQUERIMIENTOS DE EQUIPO TÉCNICO PARA LA PRESENTACIÓN DE LA PONENCIA

Ninguno.