

**Escuela Superior Politécnica del Litoral**

**Facultad de Ingeniería en Electricidad y Computación**

Clasificación de Productos Académicos de la ESPOL según su ODS mediante  
técnicas de Inteligencia Artificial

TECH-369

**Proyecto Integrador**

Previo la obtención del Título de:

**Ingeniero en Ciencias Computacionales**

Presentado por:

Isaac Miguel Ponce Valarezo

Anthony Bryan Barco Lascano

Guayaquil - Ecuador

Año: 2024

## Declaración Expresa

---

Nosotros, Isaac Miguel Ponce Valarezo y Anthony Bryan Barco Lascano, acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

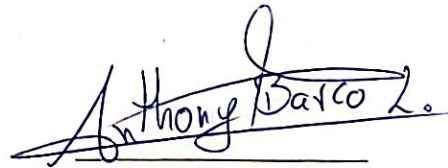
La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 18 de mayo del 2024.



Isaac Miguel Ponce  
Valarezo



Anthony Bryan Barco  
Lascano

## **Evaluadores**

---

**PhD. Luis Eduardo Mendoza Morales**

Profesor de la Materia

---

**MSC. Lenin Eduardo Freire Cobo**

Tutor de Proyecto

## Resumen

Ante la creciente necesidad de mejorar la clasificación de productos académicos vinculados a los Objetivos de Desarrollo Sostenible (ODS), este proyecto se centra en el desarrollo de una herramienta automatizada que optimiza dicho proceso mediante técnicas avanzadas de Inteligencia Artificial. El objetivo principal es ofrecer una solución que reduzca el margen de error asociado a la clasificación manual y el tiempo, incrementando así la eficiencia y precisión en el análisis de productos de uso académico. Para ello, se utilizó un modelo basado en el Procesamiento de Lenguaje Natural (PLN) y aprendizaje supervisado, implementado a través de la arquitectura BERT multilingüe. Tras entrenar y probar el modelo con productos académicos en español e inglés, se obtuvo una tasa de efectividad del 87% en la correcta clasificación de los productos según los ODS. Este avance no solo reduce la carga de trabajo manual, sino que también impulsa a ESPOL a fortalecer su visibilidad en los rankings universitarios y consolidar su compromiso con la sostenibilidad, mejorando la capacidad de la institución para adaptarse a las exigencias de los ODS hasta el 2030.

**Palabras claves:** ODS, Clasificación de textos, BERT, Procesamiento de Lenguaje Natural, Aprendizaje supervisado.

## ***Abstract***

*Given the growing need to improve the classification of academic products linked to the Sustainable Development Goals (SDGs), this project focuses on the development of an automated tool that optimizes this process through advanced Artificial Intelligence techniques. The main objective is to offer a solution that reduces the margin of error associated with manual classification and time, thus increasing efficiency and accuracy in the analysis of academic products. For this purpose, a model based on Natural Language Processing (NLP) and supervised learning, implemented through the multilingual BERT architecture, was used. After training and testing the model with academic products in Spanish and English, an 87% effectiveness rate was obtained in the correct classification of the products according to the SDGs. This progress not only reduces the manual workload, but also boosts ESPOL to strengthen its visibility in university rankings and consolidate its commitment to sustainability, improving the institution's capacity to adapt to the demands of the SDGs until 2030.*

*Key words: SDG, Text classification, BERT, Natural Language Processing, Supervised learning.*

## Índice general

Resumen.....	IV
Abstract .....	V
Abreviaturas .....	IX
Índice de tablas .....	X
Índice de Figuras.....	X
Capítulo 1 .....	1
1.    Introducción .....	2
1.1    Descripción del Problema.....	2
1.2    Justificación del Problema .....	2
1.3    Objetivos.....	3
1.3.1    Objetivo general.....	3
1.3.2    Objetivos específicos .....	3
1.4    Marco Teórico.....	4
1.4.1    Objetivos de Desarrollo Sostenible.....	4
1.4.2    Inteligencia Artificial aplicada en clasificación automatizada .....	4
1.4.3    Técnicas de IA: Procesamiento de Lenguaje Natural y aprendizaje supervisado .....	6
1.4.4    Inteligencia Artificial para la clasificación de textos.....	7
1.4.5    Herramienta basada en IA para la clasificación de documentos según los ODS.....	8
Capítulo 2 .....	2
2.    Metodología .....	11

2.1	Análisis .....	11
2.1.1	Requerimientos .....	11
2.1.2	Alcance y Limitaciones de la solución .....	13
2.1.3	Riesgos y Beneficios de la solución.....	14
2.1.4	Alternativas de solución del problema.....	14
2.2	Prototipado.....	15
2.3	Evaluación de la solución .....	23
2.3.1	Riesgos.....	23
2.3.2	Mejoras .....	24
2.4	Diseño de la solución .....	24
2.4.1	Diseño Conceptual .....	24
2.4.2	Diseño arquitectónico .....	25
2.5	Plan de implementación.....	33
Capítulo 3	.....	11
3.	Desarrollo.....	35
3.1	Fase 1: Procesamiento de los datos.....	35
3.1.1	Obtención de Datos.....	35
3.1.2	Preparación de datos .....	36
3.1.3	Visualización de datos .....	36
3.2	Fase 2: Entrenamiento del modelo.....	37
3.3	Fase 3: Levantamiento de la API.....	38
3.4	Etapa 4: Construcción del módulo web .....	39

3.4.1	Vista Principal.....	39
3.4.2	Vista de Resultados.....	41
3.4.3	Vistas adicionales.....	42
3.5	Resultados y Análisis .....	45
3.5.1	Procesamiento de los datos. ....	45
3.5.2	Entrenamiento del modelo .....	49
3.5.3	Construcción del módulo web .....	54
3.6	Análisis de costos.....	55
Capítulo 4	.....	35
4.	Conclusiones y recomendaciones .....	58
4.1	Conclusiones.....	58
4.2	Recomendaciones .....	58
Referencias	.....	60
Anexos	.....	62



## Abreviaturas

API	Interfaz de Programación de Aplicaciones (Application Programming Interface)
ARGECON	Congreso Bienal del IEEE de Argentina
BERT	Bidirectional Encoder Representations from Transformers
ESPOL	Escuela Superior Politécnica del Litoral
GTSI	Gerencia de Tecnología y Sistemas de Información
IEEE	Institute of Electrical and Electronics Engineers
ISBN	Identification Standard Book Number
LDC	Lógica Difusa Compensatoria
ML	Machine Learning (Lenguaje de Máquina)
ODS	Objetivos de Desarrollo Sostenible
ONU	Organización de Naciones Unidas
PDF	Formato de Documento Portátil (Portable Document Format)
PLN	Procesamiento de Lenguaje Natural
PNUMA	Programa de las Naciones Unidas para el Medio Ambiente
RGB	Red Green Blue
VPS	Virtual Private Server (Servidor Virtual Privado)

## Índice de tablas

<b>Tabla 1:</b> Requerimientos funcionales del sistema de clasificación de productos académicos.....	12
<b>Tabla 2:</b> Requerimientos no funcionales del sistema de clasificación de productos académicos	12
<b>Tabla 3:</b> Historias de Usuarios.....	26
<b>Tabla 4:</b> Plan de implementación de solución .....	33
<b>Tabla 5:</b> Costos de desarrollo e implementación .....	56

## Índice de Figuras

<b>Figura 1:</b> Vista general de la plataforma web.....	16
<b>Figura 2:</b> Vista de archivo cargado.....	17
<b>Figura 3:</b> Vista de resultados: Análisis de archivo .....	18
<b>Figura 4:</b> Vista escritura de texto plano.....	18
<b>Figura 5:</b> <i>Vista de resultados: Análisis de texto plano</i> .....	19
<b>Figura 6:</b> <i>Ventana emergente: Requerimientos obligatorios en archivos</i> .....	20
<b>Figura 7:</b> <i>Ventana emergente: ESPOL y su contribución con los ODS</i> .....	20
<b>Figura 8:</b> <i>Feedback del usuario</i> .....	21
<b>Figura 9:</b> <i>Pantalla de carga: Analizando archivo</i> .....	22
<b>Figura 10:</b> <i>Pantalla de carga: Analizando Texto</i> .....	22
<b>Figura 11:</b> <i>Modelo “4+1” Vistas de Krunchten</i> .....	25
<b>Figura 12:</b> Diagrama de estado .....	27
<b>Figura 13:</b> Diagrama de Componentes .....	29
<b>Figura 14:</b> <i>Diagrama de Secuencia</i> .....	30
<b>Figura 15:</b> Diagrama de Despliegue .....	32

<b>Figura 16:</b> Estructura funcional del modelo BERT .....	38
<b>Figura 17:</b> <i>Ventana Principal</i> .....	40
<b>Figura 18:</b> <i>Ventana resultados: Análisis de texto</i> .....	41
<b>Figura 19:</b> <i>Ventana resultados: Análisis de archivos</i> .....	42
<b>Figura 20:</b> <i>Ventana emergente: Card ODS</i> .....	43
<b>Figura 21:</b> <i>Ventana de carga</i> .....	43
<b>Figura 22:</b> <i>Ventana de error</i> .....	44
<b>Figura 23:</b> <i>Ventana emergente: Requerimientos obligatorios</i> .....	45
<b>Figura 24:</b> <i>Distribución de caracteres en título</i> .....	46
<b>Figura 25:</b> <i>Distribución de caracteres en problemática</i> .....	46
<b>Figura 26:</b> <i>Distribución de caracteres en resultados</i> .....	47
<b>Figura 27:</b> <i>Distribución de caracteres en impacto</i> .....	48
<b>Figura 28:</b> Matriz de Confusión Normalizada .....	50
<b>Figura 29:</b> Resultados de clasificación parcial .....	51
<b>Figura 30:</b> Resultados de clasificación parcial .....	52
<b>Figura 31:</b> Resultados de clasificación parcial .....	53
<b>Figura 32:</b> Resultados de clasificación parcial .....	54

# Capítulo 1

## **1. Introducción**

### **1.1 Descripción del Problema**

La Escuela Superior Politécnica del Litoral (ESPOL) enfrenta el desafío de medir y evidenciar su contribución a los Objetivos de Desarrollo Sostenible (ODS) en diversos productos académicos, como los contenidos de curso, tesis de grado, tesis de posgrados, programas de servicio social, proyectos de alumnos, convenios, así como proyectos de investigación, obras con Código Internacional Normalizado para Libros (ISBN, por sus siglas en inglés de International Standard Book Number), y patentes. La clasificación y asignación manual de ODS, actualmente a cargo del personal académico, podría resultar en inconsistencias y falta de precisión, afectando el reconocimiento de la institución en los rankings universitarios y su compromiso con la sostenibilidad.

Los sistemas manuales son propensos a errores humanos, lo que puede llevar a una clasificación inconsistente de los ODS. La variabilidad en la interpretación de los criterios entre diferentes evaluadores, sumada al alto volumen y diversidad de los productos académicos, resulta en un proceso ineficiente y lento. Este desafío representa una problemática sectorial, evidenciando la necesidad de un método estandarizado y automatizado que asegure la correcta identificación y reporte de contribuciones a los ODS en el ámbito académico y de investigación.

### **1.2 Justificación del Problema**

Este problema es de gran relevancia, ya que las instituciones educativas de todo el mundo están bajo creciente presión para demostrar su compromiso con los ODS de la Organización de Naciones Unidas (ONU). La capacidad de rastrear y reportar de manera precisa y eficiente, cómo

sus actividades académicas y de investigación contribuyen a estos objetivos, es crucial para asegurar su posición en los rankings globales y para atraer fondos y colaboraciones.

El Programa de Sostenibilidad ESPOL busca fortalecer la integración de los ODS en sus actividades académicas y de investigación, asegurando así una contribución significativa a los objetivos globales de desarrollo sostenible.

Finalmente, el problema subraya la necesidad urgente de una solución automatizada y precisa, para mejorar la eficacia y la exactitud en la clasificación de productos académicos según su contribución a los ODS.

### **1.3 Objetivos**

#### ***1.3.1 Objetivo general***

Desarrollar una Interfaz de Programación de Aplicaciones o API, por sus siglas en inglés (Application Programming Interface) impulsada por técnicas de Inteligencia Artificial, específicamente Procesamiento de Lenguaje Natural (PLN) y aprendizaje supervisado, para la automatización del proceso de clasificación de productos académicos de la ESPOL según su contribución a los Objetivos de Desarrollo Sostenible (ODS).

#### ***1.3.2 Objetivos específicos***

1. Implementar un modelo de PLN capaz de identificar conceptos relevantes en textos académicos tanto en inglés como en español y categorizarlos según su relación con los ODS.

2. Entrenar el modelo clasificador utilizando un conjunto de datos representativo de textos académicos de la ESPOL optimizando la precisión y relevancia en la identificación de ODS.
3. Desarrollar una API que gestione las solicitudes al modelo de PLN previamente entrenado y que sea altamente compatible con los sistemas académicos y de gestión de información de la ESPOL.
4. Diseñar una interfaz web accesible e intuitiva que permita a los usuarios hacer uso directamente de la API, y que proporcione resultados claros y comprensibles sobre la clasificación del texto.

## **1.4 Marco Teórico**

### ***1.4.1 Objetivos de Desarrollo Sostenible***

Los ODS constituyen una serie de 17 objetivos de alcance mundial, establecidas por las Naciones Unidas en 2015, con el propósito universal de erradicar la pobreza, preservar el medio ambiente y asegurar el bienestar y la tranquilidad para todos los individuos[1]. Estos objetivos se orientan hacia la disminución de la pobreza extrema y la disparidad económica, mediante la promoción del acceso equitativo a oportunidades económicas, recursos esenciales y servicios fundamentales, con el fin de asegurar que todos los individuos cuenten con los medios necesarios para cubrir sus necesidades básicas[2].

### ***1.4.2 Inteligencia Artificial aplicada en clasificación automatizada***

La IA se define como la integración de tecnologías que emplean datos, algoritmos y capacidad computacional para exhibir comportamientos inteligentes, permitiendo a las máquinas

llevar a cabo tareas que tradicionalmente requerían intervención humana[3]. En el ámbito de la clasificación automatizada, la IA se aplica para analizar y categorizar grandes volúmenes de datos de forma rápida y precisa, disminuyendo la necesidad de intervención humana y aumentando la eficiencia en la toma de decisiones[4].

Gracias al progreso de los algoritmos de aprendizaje automático, las técnicas de procesamiento de lenguaje natural y las redes neuronales artificiales, entre otras, se ha posibilitado el desarrollo de sistemas capaces de procesar grandes cantidades de datos y tomar decisiones precisas y consistentes, obteniendo una amplia variedad de aplicaciones, las más destacadas[5]:

- Clasificación de textos: Se emplea frecuentemente para categorizar textos en distintas clases, como noticias, artículos académicos o correos electrónicos. Un ejemplo es la detección y filtrado de correo spam, un problema relevante debido al gran volumen de este tipo de mensajes. Un artículo publicado en la Revista Iberoamericana de Inteligencia Artificial mostró como un modelo con alta precisión en el filtrado de correos spam y determinó la necesidad de construir modelos dinámicos que se adapten a cambios como el concepto de ruido, al que definen como presencia de datos irregularmente etiquetados o erróneos en los conjuntos de datos utilizados para entrenar y evaluar sus modelos[6].
- Clasificación de productos: En la conferencia bienal del IEEE de Argentina (ARGENCON) en 2022, se presentó una iniciativa que busca automatizar la clasificación de publicaciones de artículos destinados a la venta. El proceso inicial implica emplear técnicas de PLN para analizar y representar computacionalmente los textos[7].



- Clasificación de datos: Se usa para categorizar grandes volúmenes de datos en clases específicas, como información de salud, datos financieros o información recogida de un censo poblacional. Una investigación propuso determinar y clasificar el riesgo financiero y sus componentes, en las instituciones bancarias venezolana, desarrollando un modelo de clasificación basados en lógica difusa, utilizando los datos históricos financieros de los bancos. Esta investigación contribuye al desarrollo de herramientas de apoyo a la toma de decisiones en el área de gestión de riesgos bancarios[8].

#### ***1.4.3 Técnicas de IA: Procesamiento de Lenguaje Natural y aprendizaje supervisado***

El procesamiento de lenguaje natural (PLN), o NLP por sus siglas en inglés (Natural Language Processing) se centra en las interacciones entre el lenguaje humano y las computadoras. Es el punto donde la informática, la IA y la lingüística computacional convergen. El PLN es una técnica para que las computadoras analicen, comprendan y obtengan significado del lenguaje humano. A través de sus algoritmos se puede organizar y estructurar el conocimiento para realizar tareas como resúmenes automáticos, traducción, reconocimiento de entidades nombradas, extracción de relaciones, análisis de sentimientos, reconocimiento de voz y segmentación de temas [9], [10].

En el aprendizaje supervisado, los algoritmos utilizan datos previamente etiquetados o estructurados, lo que les permite predecir cómo debe clasificarse nueva información. Este enfoque requiere la participación humana para ofrecer retroalimentación[3]. El proceso de aprendizaje supervisado consta de dos etapas clave: la fase de entrenamiento y la fase de prueba. En la fase de entrenamiento, el modelo recibe un conjunto de datos que contiene tanto las

entradas como sus etiquetas asociadas, con el propósito de aprender la relación entre ambas. Luego, en la fase de prueba, se evalúa el desempeño del modelo con un conjunto de datos que no ha visto previamente, lo que permite medir su capacidad de generalización y precisión en la predicción de nuevos datos[11], [12].

#### ***1.4.4 Inteligencia Artificial para la clasificación de textos***

La implementación de Inteligencia Artificial en este contexto tiene el potencial de mejorar de manera sustancial la eficacia y precisión del proceso de clasificación, al mismo tiempo que disminuye la carga de trabajo manual y elimina las discrepancias originadas por la interpretación subjetiva de los criterios[13].

Un estudio realizado en la Universidad Politécnica de Valencia abordó problemas de clasificación de texto como análisis de sentimientos y detección de temas en tweets mediante el uso de modelos basados en aprendizaje profundo con el desarrollarlo de sistemas basados en redes neuronales recurrentes y convolucionales, usando diferentes representaciones del texto como vectores de palabras, representaciones secuenciales y de documento, y también detección automática de lenguaje ofensivo cuyo objetivo era predecir si un comentario era inapropiado o no [14].

Uno de los trabajos orientados a la clasificación lingüística se centra en clasificar automáticamente textos en español según el tipo de palabra usando Lógica Difusa Compensatoria (LDC) y el algoritmo de Distancia de Edición de Levenshtein. El artículo explica la utilización de un corpus de conocimiento previo en español, compuesto por patrones léxicos clasificados en categorías como artículo, sustantivo, pronombre, adjetivos, etc. El algoritmo puede reconocer nuevos patrones y analizar textos para encontrar patrones comunes, esto es, por

ejemplo, si el texto es "la casa azul", se categoriza "la" como artículo, "casa" como sustantivo y "azul" como adjetivo[15].

#### ***1.4.5 Herramienta basada en IA para la clasificación de documentos según los ODS***

Los sistemas inteligentes, como las tecnologías de aprendizaje automático, poseen un enorme potencial para impulsar el avance hacia las metas de desarrollo sostenible. En este contexto, la asignación de documentos textuales a los ODS se convierte en una tarea fundamental para organizaciones como el Programa de las Naciones Unidas para el Medio Ambiente (PNUMA). Tradicionalmente, este proceso ha sido llevado a cabo por expertos que dedican una cantidad significativa de tiempo a mapear textos con los ODS, basándose en su conocimiento personal de los vínculos entre los temas tratados y los ODS. Sin embargo, este método puede ser propenso a errores, especialmente cuando los documentos contienen temas fuera del área de especialización de los expertos[13], [16].

Para superar esta limitación, la ONU ha creado una herramienta denominada SDG Meter, esta plataforma emplea técnicas avanzadas de IA para analizar textos y determinar su relevancia con respecto a cada uno de los 17 ODS[17]. En particular, SDG Meter se basa en el modelo de Representación de Codificador Bidireccional de Transformadores (BERT por sus siglas en inglés de Bidirectional Encoder Representations from Transformers), una técnica de procesamiento del lenguaje natural que ha demostrado ser altamente eficaz en la clasificación de textos con múltiples etiquetas. El algoritmo de SDG Meter ha sido entrenado con aproximadamente 3000 textos y etiquetas obtenidos de las categorías "Noticias", "Artículos invitados" y "Policy Briefs" del sitio web IISD-ODS obteniendo precisión del 98%. Esto representa un avance significativo en la integración de sistemas autónomos de cómputo con los propósitos de sostenibilidad global

de las Naciones Unidas, proporcionando una metodología robusta y validada para el mapeo de documentos textuales a los ODS[18].

## **Capítulo 2**

## **2. Metodología**

Para la solución, se optó por un sistema de clasificación automatizada, se desarrolló una API impulsada por técnicas de IA, específicamente procesamiento de PLN y aprendizaje supervisado, para realizar la clasificación de manera completamente automatizada. Esta solución fue seleccionada debido a su capacidad para mejorar significativamente la precisión, consistencia y eficiencia en la clasificación de productos académicos.

### **2.1 Análisis**

Para el desarrollo de este proyecto, se llevaron a cabo varias reuniones con el cliente con la finalidad de recopilar toda la información necesaria que permita, mediante un análisis detallado, establecer los requerimientos funcionales y no funcionales de la aplicación web. Esto comprende desde el momento en que el usuario carga un documento o ingresa un texto para conocer los ODS relacionados, hasta mostrar en una interfaz los resultados finales; tomando en cuenta también, el rendimiento y disponibilidad del producto.

#### ***2.1.1 Requerimientos***

Durante el análisis del proyecto se recopilaron varias funcionalidades que fueron obtenidas como parte de las reuniones grupales con el GTSI, estas suponen las expectativas y necesidades que tiene el cliente en cuanto al proyecto. En la Tabla 1 y Tabla 2 se describen los requerimientos funcionales y no funcionales para el producto; como parte de su identificación se estable la codificación RF-# para requerimientos funcionales y NF-# para requerimientos no funcionales.

**Tabla 1***Requerimientos funcionales del sistema de clasificación de productos académicos*

<b>Categoría</b>	<b>ID</b>	<b>Descripción</b>
<b>Acceso y Subida de Archivos</b>	RF-001	La aplicación debe permitir el acceso a la API mediante una interfaz web.
	RF-002	La aplicación debe permitir cargar productos académicos en formato PDF.
	RF-003	La aplicación debe permitir cargar el contenido de productos académicos.
<b>Validación de Archivos</b>	RF-004	La aplicación debe mostrar una alerta si el formato del archivo seleccionado no es PDF.
	RF-005	La aplicación debe mostrar una alerta si el peso del archivo PDF supera el límite permitido.
	RF-006	La aplicación debe considerar que la longitud del texto plano no excede el máximo de caracteres permitido.
<b>Confirmación de Subida</b>	RF-007	La aplicación debe confirmar con una alerta que el archivo PDF ha sido subido con éxito si cumple con el peso permitido.
<b>Análisis y Clasificación</b>	RF-008	La aplicación debe analizar y clasificar productos académicos en español e inglés
<b>Resultados y Visualización</b>	RF-009	La aplicación debe mostrar el resultado de la clasificación por ODS para productos académicos en español e inglés
	RF-010	La aplicación debe mostrar el porcentaje de relación de cada ODS con el producto académico subido.

**Tabla 2***Requerimientos no funcionales del sistema de clasificación de productos académicos*

<b>Categoría</b>	<b>ID</b>	<b>Descripción</b>
<b>Rendimiento</b>	NF - 001	La API debe procesar y analizar los documentos o texto en un tiempo razonable, proporcionando los resultados en un plazo que no afecte negativamente la experiencia del usuario (idealmente en menos de 10 segundos para documentos de hasta 2MB).
<b>Seguridad</b>	NF - 002	La API debe implementar mecanismos de autenticación del GTSI de ESPOL para evitar el abuso del API por parte de usuarios externos.
<b>Compatibilidad</b>	NF - 003	La API debe ser compatible con los sistemas académicos existentes de ESPOL, esto implica que siempre que sea posible se seguirán los lineamientos técnicos del GTSI.
<b>Usabilidad</b>	NF - 004	La interfaz web debe ser intuitiva y fácil de usar, proporcionando una experiencia de usuario fluida y accesible, incluso para aquellos con conocimientos técnicos limitados.
<b>Tolerancia a fallos</b>	NF - 005	La API debe ser capaz de manejar errores y fallos de manera robusta, proporcionando mensajes de error claros y recuperándose automáticamente de problemas menores sin intervención del usuario.
<b>Mantenibilidad</b>	NF - 006	El código de la API debe estar bien estructurado y documentado para facilitar su mantenimiento futuro por parte del equipo de desarrollo.

## **2.1.2 Alcance y Limitaciones de la solución**

**2.1.2.1 Alcance.** La solución incluirá una interfaz web intuitiva y accesible que permita a los académicos cargar productos en formato PDF o texto plano para su análisis y clasificación según los ODS tanto en español como en inglés, proporcionando resultados detallados sobre la relación con dichos objetivos. El proyecto permite que el personal que analiza el material académico evalúa y asigna los ODS, reduzca altos tiempos de clasificación manual realizados mediante sus estándares puntuales ya establecidos, evite posibles errores humanos o inconsistencias a causa de criterios divididos.

**2.1.2.2 Limitaciones.** La solución estará limitada a la clasificación de productos académicos en PDF de hasta 10 megabytes (10MB) del idioma español e inglés, o en su defecto, desde 128 hasta 4096 caracteres de texto plano para el análisis textual de párrafos; los archivos ingresados deberían tener una estructura que permita identificar un “resumen” o “abstract” en una sola columna del documento.

Las capacidades de análisis y clasificación pueden verse limitadas por los recursos de hardware y software disponibles en el servidor donde se despliega la API y la interfaz web. Aunque se espera que el tiempo de procesamiento sea razonable, puede haber limitaciones en el rendimiento bajo cargas extremadamente altas o debido a la complejidad del análisis de documentos extensos. Es importante tener en cuenta que el modelo está diseñado para trabajar únicamente con los ODS existentes y se mantendrá vigente hasta el año 2030 según la agenda de las Naciones Unidas.



### ***2.1.3 Riesgos y Beneficios de la solución***

***2.1.3.1 Riesgos.*** Uno de los principales riesgos es la precisión del análisis multilingüe, especialmente en términos de comprensión contextual y semántica en diferentes idiomas. Errores en la clasificación podrían afectar la credibilidad de los resultados y la utilidad práctica de la aplicación. Además, la dependencia de recursos computacionales y algoritmos PLN pueden introducir riesgos de rendimiento y también de escalabilidad, especialmente durante picos de demanda o con grandes volúmenes de datos.

***2.1.3.2 Beneficios.*** La automatización del proceso de clasificación de productos académicos permite, a sus usuarios y a la ESPOL, evaluar rápidamente cómo sus investigaciones se alinean con los ODS. Esto facilita la contribución a metas globales de desarrollo sostenible de manera más ágil y efectiva. Además de mejorar la eficiencia en la evaluación del impacto académico, esta automatización fomenta la conciencia y la acción hacia la sostenibilidad dentro de la comunidad académica. Asimismo, la visualización clara y detallada de los resultados proporciona retroalimentación instantánea y valiosa para los investigadores, facilitando la toma de decisiones informadas y la planificación estratégica en términos de enfoque investigativo y colaboración internacional.

### ***2.1.4 Alternativas de solución del problema***

Una solución alternativa a la problemática podría ser BERT, una técnica avanzada de procesamiento del lenguaje natural desarrollada por Google, que ha demostrado una notable eficacia en tareas de clasificación de texto con múltiples etiquetas. Este modelo es especialmente adecuado para este problema porque puede capturar el contexto de las palabras en una frase

permitiendo comprender el significado profundo de los textos académicos, lo cual es crucial para determinar con precisión cómo se alinean con los diferentes ODS [18].

Al implementar BERT para la clasificación de productos académicos, se podría desarrollar una API que realice consultas a este modelo, permitiendo un análisis automatizado y eficiente de los textos académicos. Sin embargo, esta solución presenta algunas limitaciones. Modelos base de BERT pre-entrenados sólo aceptan textos en inglés, por lo que sería necesario entrenar el modelo para que también pueda procesar textos en español, lo cual requiere una considerable cantidad de datos y recursos de procesamiento. También, existiría el riesgo de que el modelo no capture todas las sutilezas contextuales de textos especializados en ambos idiomas, lo que podría afectar la precisión en algunos casos. Finalmente, la dependencia de actualizaciones y mantenimientos regulares es crucial para asegurar que el sistema permanezca efectivo frente a nuevas publicaciones y cambios en los criterios de clasificación.

## **2.2 Prototipado**

A continuación, se presentará las pantallas principales desarrolladas para la plataforma web en un modelo de alta fidelidad como primera perspectiva de solución del proyecto.

La Figura 1 muestra la vista general que tendrá el usuario, la cual cuenta con encabezado seguido de la sesión de análisis, aquí se puede procesar texto plano o en su defecto cargar un archivo con el botón “subir”. También se muestra al usuario los 17 ODS a los que serán asignados a la producción científica que se pueda analizar y en el pie de página información importante que expone este proyecto de investigación.

Figura 1

Vista general de la plataforma web

**OBJETIVOS DE DESARROLLO SOSTENIBLE** EVALUANDO EL IMPACTO SOCIAL MIDIENDO LA CONTRIBUCIÓN DE SU TEXTO A LOS ODS **SOSTENIBILIDAD espol**

**Cuantificando la Relación entre su Escrito y los ODS**

Ingresar tu texto aquí (Sólo inglés o español)

Subir Procesar 0/1000

1 NO POBREZAS 2 HERRIBRE CERO 3 SALUD Y BIENESTAR 4 EDUCACIÓN DE CALIDAD 5 IGUALDAD DE GÉNERO 6 AGUA LIMPIA Y SANEAMIENTO

7 ENERGÍA LIMPIA Y ACCESIBLE 8 TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO 9 INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA 10 REDUCCIÓN DE LAS DESIGUALDADES 11 CIUDADES Y COMUNIDADES SOSTENIBLES 12 PRODUCCIÓN Y CONSUMO RESPONSABLES

13 ACCIÓN POR EL CLIMA 14 VIDA SUBMARINA 15 VIDA DE ECOSISTEMAS TERRESTRES 16 PAZ, JUSTICIA E INSTITUCIONES SÓLIDAS 17 ALIANZAS PARA LOGRAR LOS OBJETIVOS

**OBJETIVOS DE DESARROLLO SOSTENIBLE**

**ACERCA DE**

La Escuela Superior Politécnica del Litoral (ESPOL) genera una gran cantidad de documentos académicos (propuestas de investigación, artículos, tesis, etc.) que abordan diversas temáticas. Relacionar manualmente estos documentos con los Objetivos de Desarrollo Sostenible (ODS) es un proceso tedioso para los expertos y depende en gran medida de su conocimiento personal sobre la vinculación entre las temáticas y los ODS. Si bien la ESPOL cuenta con especialistas en múltiples campos, podrían pasarse por alto contribuciones a los ODS que se encuentran fuera de su área de expertise. Nuestro sitio web pretende ayudar a los académicos de Espol, y cualquiera que lo requiera, a clasificar automáticamente los productos académicos según su contribución a cada uno de los 17 ODS.

Nuestro sitio está impulsado por Inteligencia Artificial (IA), específicamente por las técnicas de Procesamiento de Lenguaje Natural (PLN) y Aprendizaje Supervisado. Inspirados por el éxito de modelos como **BERT** (*Representaciones Bidireccionales de Codificadores a partir de Transformadores*), desarrollados por investigadores de Google, nuestra plataforma utilizará técnicas similares para el pre-entrenamiento del PLN. Para lograr una clasificación precisa, entrenaremos nuestro algoritmo con un conjunto de datos masivo de productos académicos previamente etiquetados de acuerdo a su contribución a los ODS. De esta manera, identificará patrones lingüísticos que le permitan clasificar automáticamente nuevos documentos con una precisión del 97%.

Escuela Superior Politécnica del Litoral  
Campus Gustavo Galindo - 2019  
Guayaquil - Ecuador  
Tel. (593-4) 3788 000

Sostenibilidad ESPOL  
Políticas

Integrantes  
Recursos

Ranking  
Contactos

ODS

Si el usuario opta por subir un archivo, el botón “subir” abrirá una ventana del explorador de archivos que le permitirá elegir un PDF para ser analizado, la vista del archivo en el sitio web se presenta en la Figura 2.

Figura 2

Vista de archivo cargado



Posteriormente, para el análisis, se procesa la información con el botón “procesar” que luego de consultar al API mostrará una nueva pestaña (ver Figura 3) en donde se encuentra los ODS y el porcentaje de significancia arrojado por el análisis, el usuario también podrá visualizar los datos del archivo que procesó.

**Figura 3**

*Vista de resultados: Análisis de archivo*



A continuación, se presentará otra perspectiva del procesamiento, en este caso se va a procesar el texto plano que se encuentra en la caja de texto (ver Figura 4).

**Figura 4**

*Vista escritura de texto plano*



Asimismo, la nueva pestaña mostrará los resultados obtenidos: los ODS en relación y el porcentaje de relevancia (ver Figura 5).

**Figura 5**

*Vista de resultados: Análisis de texto plano*

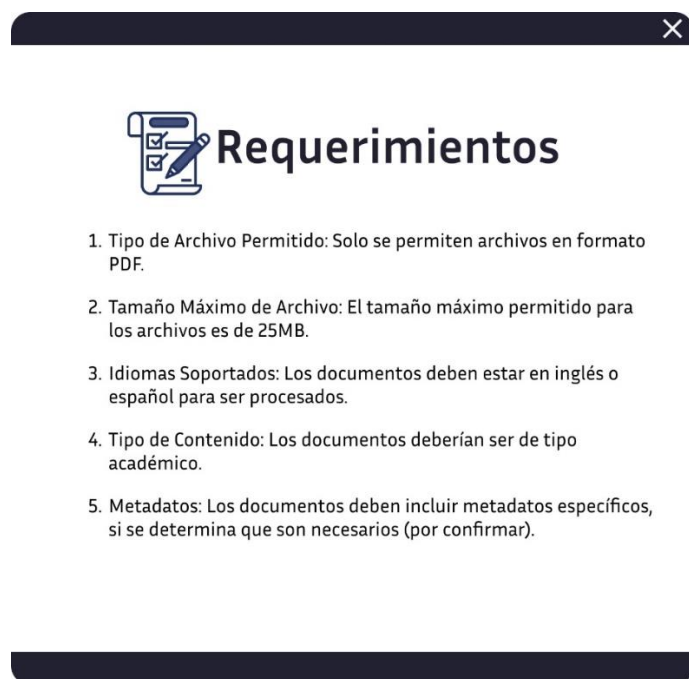


Por otro lado, se muestran a continuación otros recursos utilizados mejorar la experiencia de usuario, mostrar información relevante o proporcionar feedback.

La Figura 6 muestra una ventana emergente, misma que surgirá una vez dado clic en el botón “subir”, esta ventana proporciona información importante sobre los requerimientos inamovibles del documento a subir para garantizar el procesamiento de análisis sin ningún tipo de error.

**Figura 6**

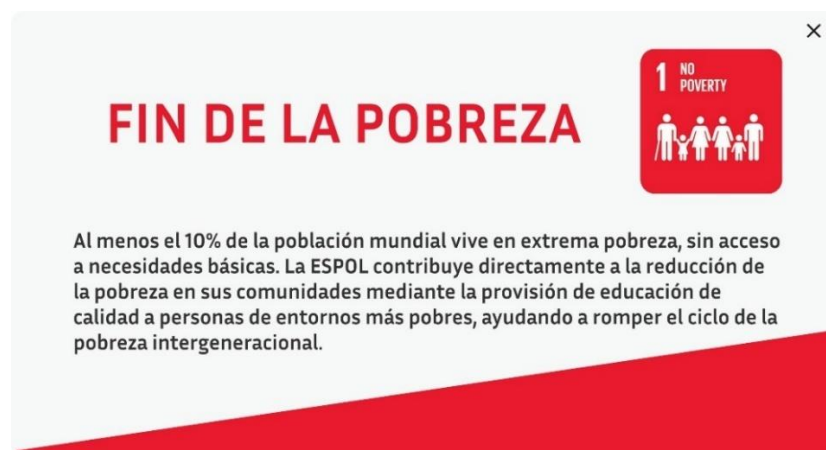
*Ventana emergente: Requerimientos obligatorios en archivos*



La Figura 7 es otro tipo de ventana emergente que surge como resultado de dar clic en los íconos de los ODS de la pantalla principal (ver Figura 1), aquí se muestra el compromiso que tiene ESPOL con las metas de las Naciones Unidas

**Figura 7**

*Ventana emergente: ESPOL y su contribución con los ODS*

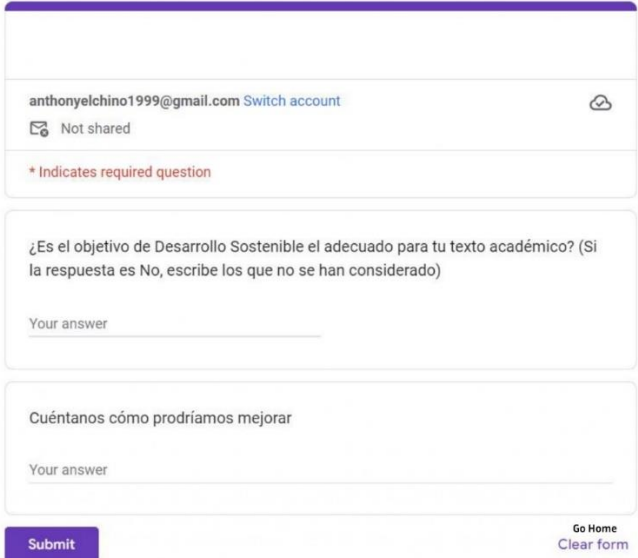


La Figura 8 es la implementación del feedback que hará el usuario de los resultados, esta se mostrará como un formulario de Google o Microsoft y recogerá información que irá directamente a un correo institucional.

### Figura 8

*Feedback del usuario*

¡DÉJANOS TUS COMENTARIOS! ×



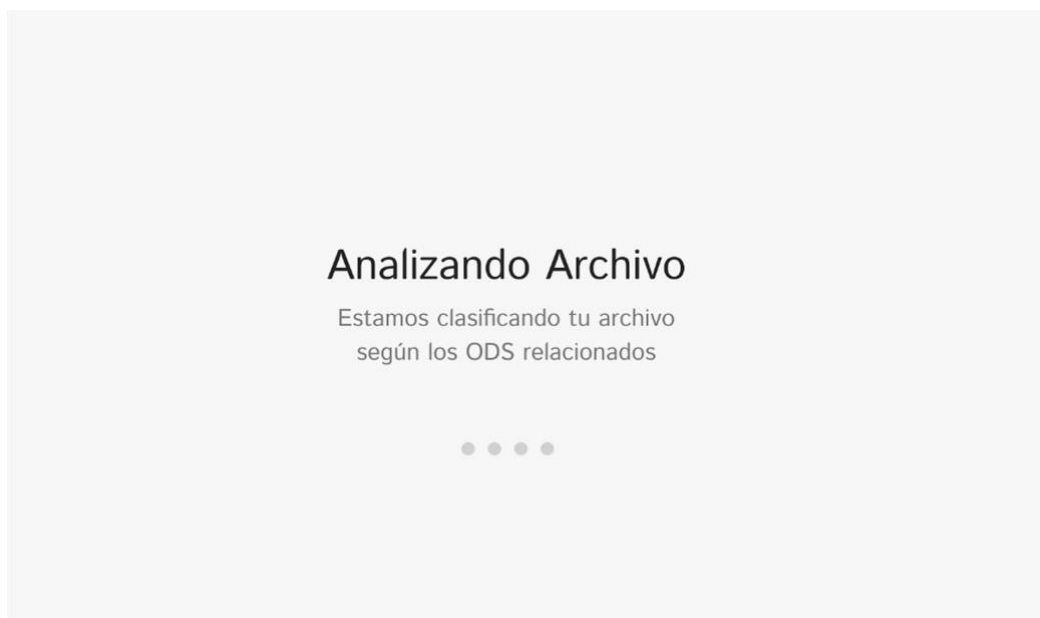
The screenshot shows a Google Forms interface. At the top, the title '¡DÉJANOS TUS COMENTARIOS!' is displayed in large, bold, dark blue letters. Below the title, the user's email 'anthonyelchino1999@gmail.com' is visible with a 'Switch account' link and a cloud icon. A 'Not shared' status is also shown. A red asterisk indicates a required question. The main question is '¿Es el objetivo de Desarrollo Sostenible el adecuado para tu texto académico? (Si la respuesta es No, escribe los que no se han considerado)'. Below the question is a text input field labeled 'Your answer'. A second question, 'Cuéntanos cómo podríamos mejorar', is followed by another 'Your answer' text input field. At the bottom left is a purple 'Submit' button, and at the bottom right are links for 'Go Home' and 'Clear form'.

También se desarrolló pantallas de carga que se muestran durante cualquiera de los procesamientos, (ver Figura 9) para el análisis de archivos y (ver Figura 10) para el análisis de texto plano.



**Figura 9**

*Pantalla de carga: Analizando archivo*

**Figura 10**

*Pantalla de carga: Analizando Texto*



## 2.3 Evaluación de la solución

La solución propuesta cumplió mayormente con los requerimientos funcionales (Tabla 1) y no funcionales (Tabla 2). El resultado fue una interfaz web intuitiva y eficiente que permitió a investigadores cargar productos académicos en formato PDF o escribir texto plano para su análisis y clasificación según su relación con los ODS. La solución incluyó características que garantizan informar al usuario sobre el idioma, formato y tamaño límite de los archivos (ver Figura 6). La interfaz soportó tanto el análisis de documentos en español como en inglés, mostrando resultados detallados y fácil de visualizar en relación con cada ODS (ver Figuras 4 y 5).

### 2.3.1 Riesgos

Los riesgos que pueden ocurrir en la solución son varios, entre ellos se tiene:

- Posible sobrecarga del servidor con archivos grandes; por ello se implementó un límite de tamaño de archivo de 10MB y optimizamos el manejo de archivos grandes para evitar sobrecargas.
- Problemas de precisión en la clasificación multilingüe; se prevé realizando pruebas exhaustivas y ajustes en los algoritmos de clasificación para mejorar la precisión tanto en español como en inglés.

Otros riesgos que se deben tomar en cuenta son:

- Problemas de rendimiento debido a la alta concurrencia de usuarios durante períodos de carga máxima.
- Problemas de interoperabilidad con los sistemas académicos de ESPOL.

- Retrasos en la actualización del modelo de clasificación debido a cambios en los estándares o criterios de los ODS.

### **2.3.2 Mejoras**

Incluir una función de previsualización del archivo antes de la carga para asegurarse de que el usuario sube el archivo correcto, esto permitiría a los usuarios verificar el contenido del archivo antes de subirlo, reduciendo errores y mejorando la experiencia del usuario.

Realizar pruebas de usabilidad con usuarios reales para obtener feedback y mejorar la interfaz de usuario, esto proporcionaría insights valiosos sobre la usabilidad y funcionalidad de la interfaz, permitiendo mejoras continuas y alineadas con las necesidades del usuario.

Incorporar soporte multilingüe completo al modelo para procesar archivos en otros idiomas adicionales y no solo limitarse al inglés y español, esto ampliaría el alcance y la utilidad de la aplicación para usuarios de diferentes regiones y lenguas, facilitando una adopción más global.

## **2.4 Diseño de la solución**

### **2.4.1 Diseño Conceptual**

**2.4.1.1 Modelo de PLN.** El modelo BertForSequenceClassification se basa en la arquitectura BERT (Bidirectional Encoder Representations from Transformers) [18] y está diseñado para tareas de clasificación de secuencias. Su núcleo es el BertModel, que incluye un BertEmbeddings con representaciones de palabras de 768 dimensiones y codificaciones posicionales y de tipo de token, junto con un BertEncoder compuesto por 12 capas (BertLayer), cada una con atención auto-regresiva y una red neuronal intermedia con activación GELU. Cada capa utiliza mecanismos de atención con matrices de consulta, clave y valor de 768 dimensiones,

y se complementa con normalización de capa y dropout. La salida del encoder se pasa a través de un BertPooler que aplica una capa densa con activación Tanh. Finalmente, el modelo incluye un Dropout adicional y una capa Linear para la clasificación final, que proyecta las representaciones a 17 clases posibles.

**2.4.1.2 API para la clasificación.** Habiendo desarrollado el modelo, se diseñó el API que gestiona las solicitudes al modelo de PLN previamente entrenado, permitiendo, luego, la integración con los sistemas académicos y de gestión de información de la ESPOL. La API fue creada para ser bilingüe, soportando textos tanto en inglés como español.

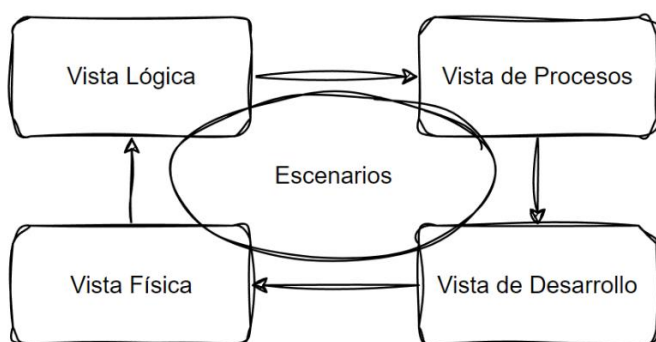
**2.4.1.3 Interfaz Web.** Se desarrolló una interfaz web accesible e intuitiva que permite a los usuarios cargar documentos y obtener resultados claros sobre la clasificación de los textos según los ODS. La interfaz proporciona visualmente el análisis detallado y porcentajes de coincidencia con los objetivos compatibles.

## 2.4.2 Diseño arquitectónico

Para especificar el diseño del API se utilizó el Modelo “4+1” Vistas de Kruchten (ver Figura 11) [19].

**Figura 11**

*Modelo “4+1” Vistas de Kruchten*



**2.4.2.1 Vista de escenarios.** Al haber completado la fase de recolección de información, se establecen los diferentes escenarios que abarca el producto, es importante remarcar que el único escenario en el que se desenvuelve la usabilidad del producto es el rol de académico. En la Tabla 3 se incluyen las historias de usuario que han sido definidas y aprobadas. Para visualizar los detalles completos, en donde se encuentran los criterios de aceptación, remitirse a consultar los anexos.

**Tabla 3**

*Historias de Usuarios*

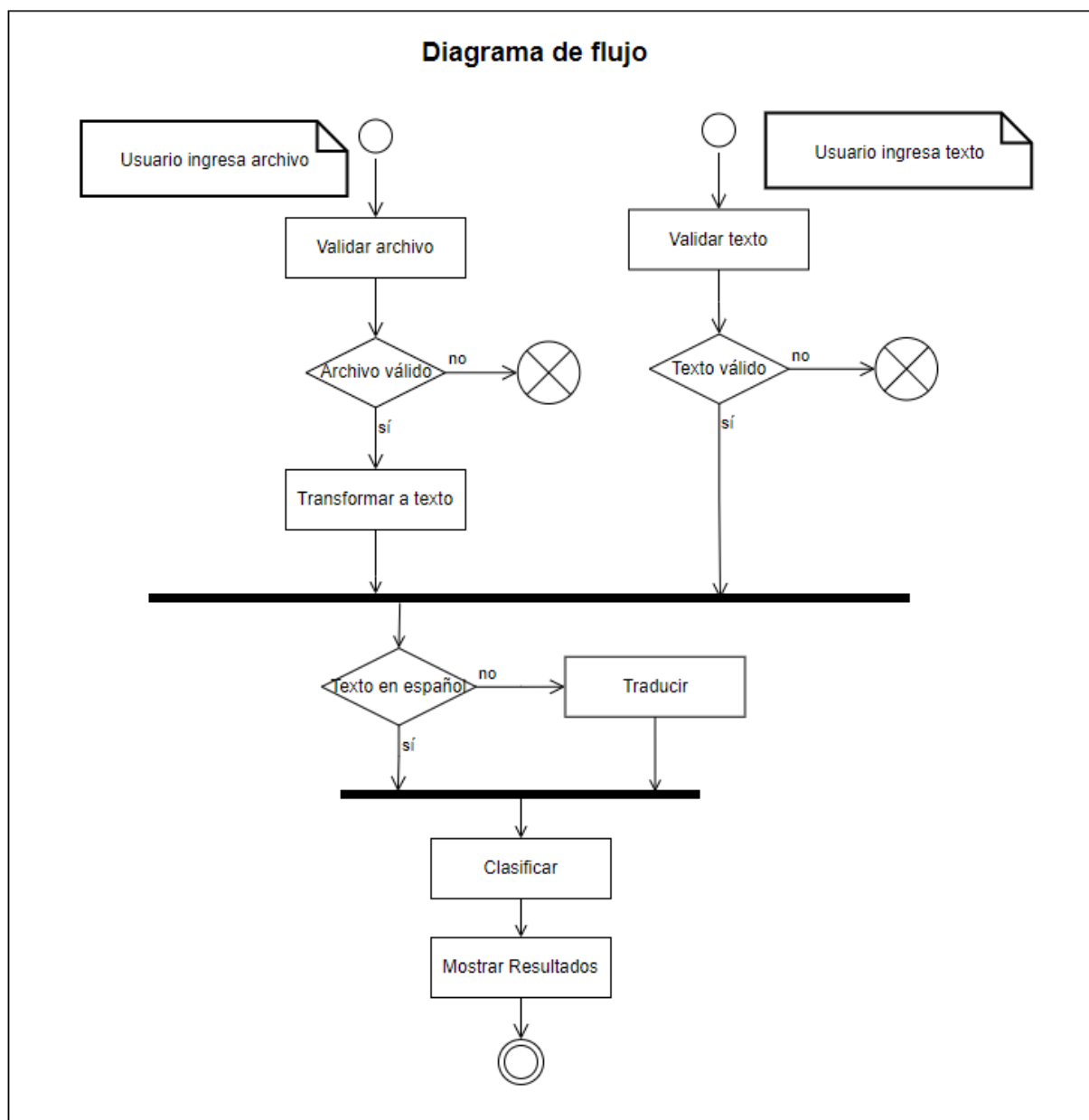
Código	Funcionalidad
WEB - 01	Necesito acceso a la API mediante una interfaz web.
PDF - 01	Necesito cargar el producto académico en formato PDF.
TXT - 01	Necesito cargar el contenido de un producto académico en texto plano.
LNG - 01	Necesito analizar el producto académico o su contenido tanto en español como en inglés.
OUT - 01	Necesito visualizar en qué porcentaje se relaciona cada ODS con el producto académico.

Ver Anexo A el despliegue total de las historias de Usuario.

**2.4.2.2 Vista lógica.** La vista lógica cubrirá toda funcionalidad que se esperan del sistema (ver Figura 12).

Figura 12

Diagrama de estado



En la Figura 12, el diagrama representa un flujo de trabajo donde un usuario ingresa un documento en PDF o texto, el contenido, a través de una interfaz web, el cual es enviado a una API para su preprocesamiento. Posteriormente, el contenido se procesa mediante un modelo

PLN, que identifica y clasifica las categorías de los ODS, retornando estos resultados a la interfaz web para que sean mostrados al usuario final.

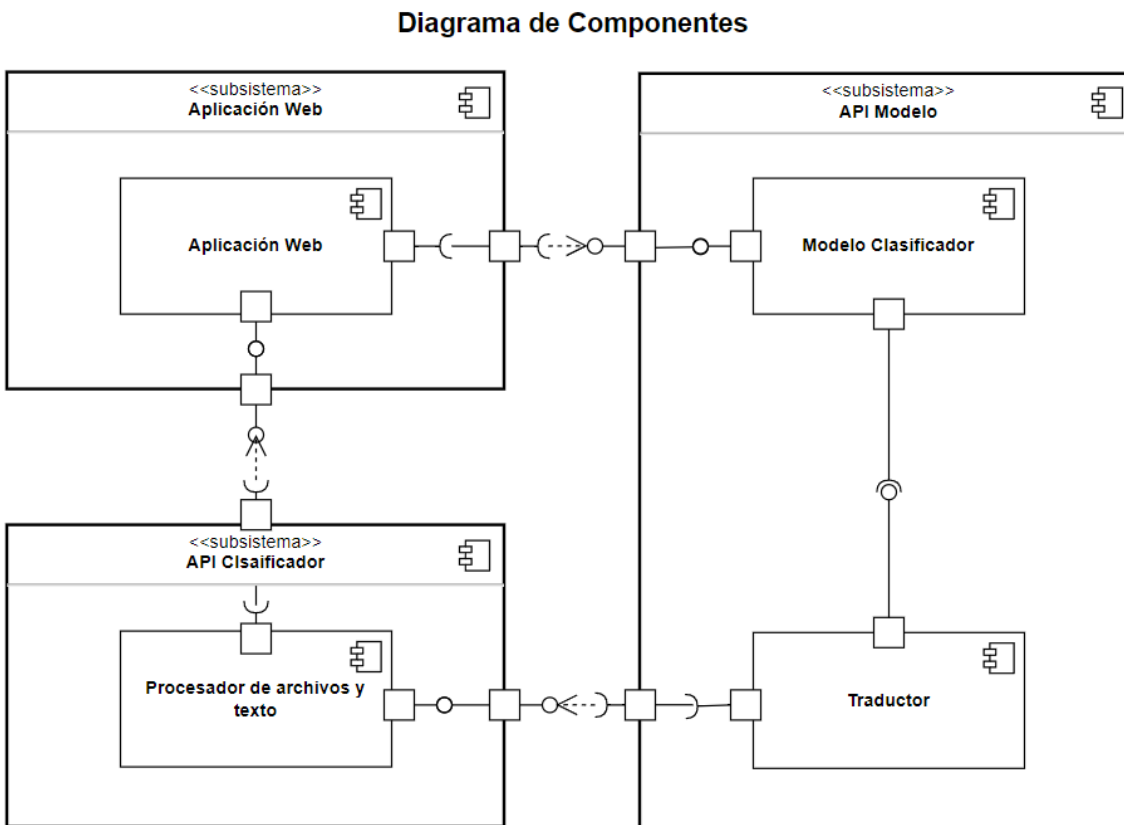
**2.4.2.3 Vista de desarrollo.** Se desarrollarán los componentes que se requieren para la implementación (ver Figura 13). Basando el sistema web en la en una arquitectura de modelo de capas donde:

- La Interfaz Web representa la capa de presentación que interactúa directamente con el usuario final.
- La API de Procesamiento de Texto actúa como una capa de lógica, manejando las operaciones principales como el procesamiento de archivos, la detección de idiomas, y la clasificación de los datos.

Para el desarrollo del proyecto se utilizará un marco de trabajo (framework) que emplea las librerías de React[20] basado en el lenguaje de JavaScript, esto servirá para llevar a cabo el Frontend. Por otro lado, en el Backend se manejará con el framework .NET Core[21].

Figura 13

Diagrama de Componentes

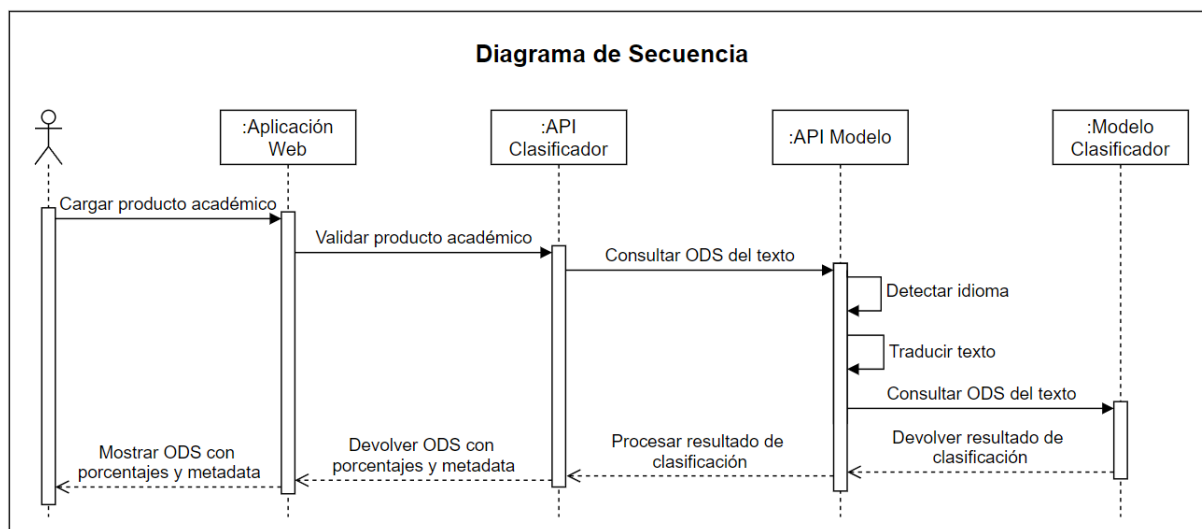


**2.4.2.4 Vista de procesos.** La vista de procesos se centra en el comportamiento dinámico del sistema, es decir, cómo los componentes del sistema interactúan entre sí durante la ejecución (ver Figura 14).



Figura 14

## Diagrama de Secuencia



## Actores Involucrados:

- Usuario: Inicia el proceso cargando un documento o texto para su análisis.
- Aplicación Web: Actúa como el intermediario entre el usuario y el backend, enviando el texto a la API y mostrando los resultados al usuario.
- API con PLN: Este componente realiza la mayor parte del procesamiento, procesando el texto y consultando el modelo de aprendizaje automático.
- Modelo ML Dockerizado: Un modelo de aprendizaje automático encapsulado en un contenedor Docker, que recibe las palabras clave para asignar los ODS y calcula el porcentaje de coincidencia.

## Flujo del Proceso:

- Carga de Documento o Texto: El proceso comienza cuando el usuario carga un documento o introduce un texto.

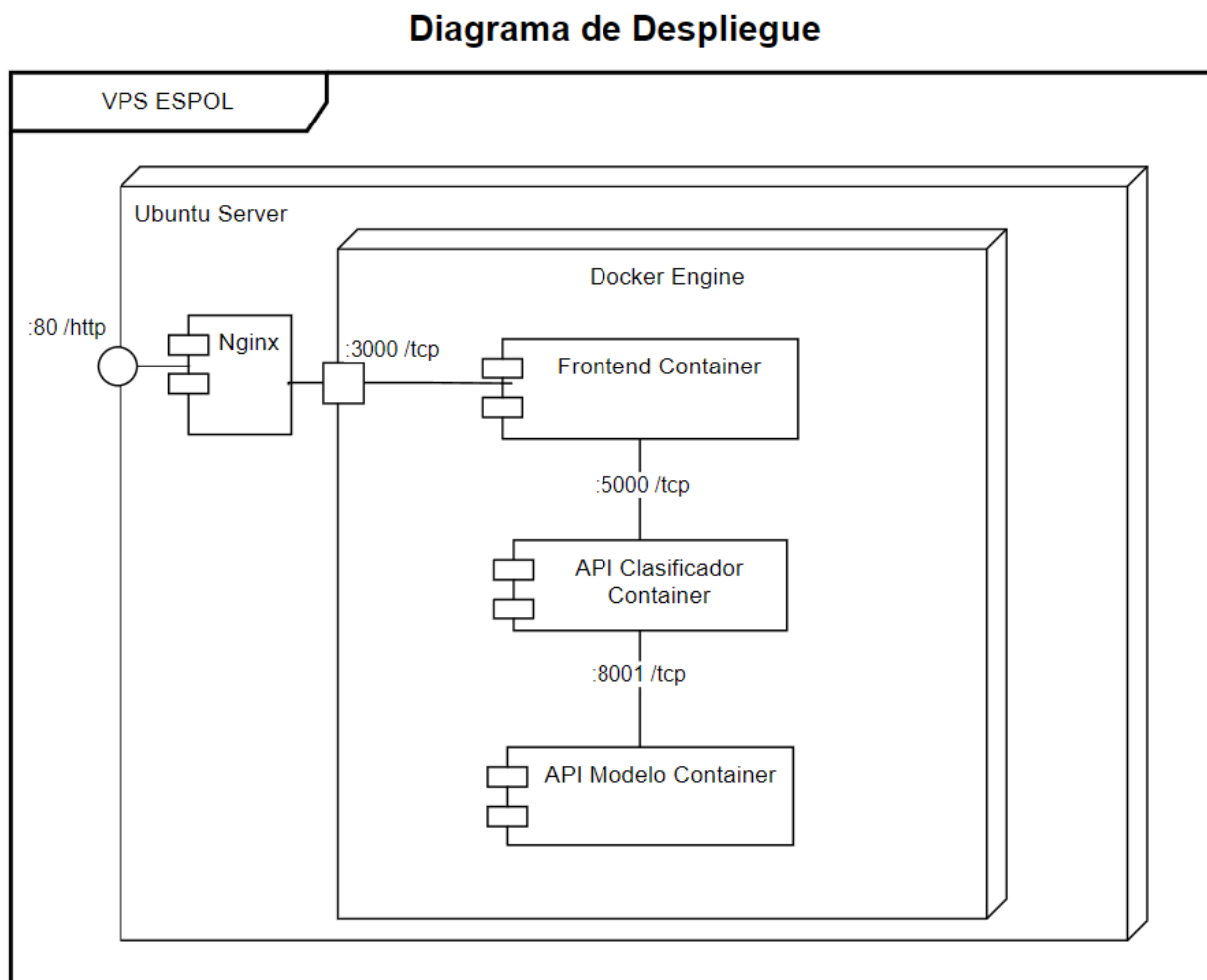
- Envío del Texto para Análisis: Aplicación Web envía el contenido a la API.
- Procesamiento del Texto: La API analiza el contenido, detectando el idioma (español o inglés) y extrae el texto final para el análisis.
- Consulta al Modelo ML: El texto final se envía al modelo de aprendizaje automático dockerizado.
- Devolución de Resultados: El modelo ML responde con los ODS asignados y el porcentaje de coincidencia, que luego se devuelve al usuario a través de la Interfaz Web.

En resumen, el diagrama de secuencia muestra cómo un usuario carga un documento o texto a través de la interfaz web. El contenido se envía a una API basada en PLN que identifica el idioma y consulta un modelo de ML contenerizado para asignar su respectivo ODS.

**2.4.2.5 Vista física.** En el diagrama de despliegue (ver Figura 15) se detallarán los componentes necesarios para su despliegue:

- Frontend, representa el desarrollo del prototipo, el sitio que podrá ser utilizado por el usuario final, donde se inicia el input hacia el API y se renderizan los resultados.
- Banckend, procesa el contenido textual de archivos o directamente texto plano, detecta el idioma y clasifica el contenido para asignar el ODS mediante el modelo PLN.

Figura 15

*Diagrama de Despliegue*

El diagrama de despliegue describe la arquitectura de un sistema alojado en un servidor virtual privado (VPS) de ESPOL. En el frontend, el "Procesador de input" gestiona la entrada de datos del usuario, mientras que el "Renderizador de resultados" muestra los resultados finales. En el backend, la "Validación de Input" verifica la validez de los datos antes de pasarlos al "Preprocesador de Texto", que prepara el texto para el análisis del modelo PLN. La comunicación entre estos componentes está gestionada por una API central.

## 2.5 Plan de implementación

La tabla 4 detalla un plan para iniciar la implementación de la solución. Para consultar el cronograma semanal de las actividades ver Apéndice A.

**Tabla 4**

*Plan de implementación de solución*

ID	TÍTULO	DÍAS
	Diseño de solución	
1.1	Diseño de arquitectura backend	3
1.2	Diseño prototipo de alta fidelidad (frontend)	3
1.3	Revisión, modificación y aceptación del prototipo	2-3
	Desarrollo	
2.1	Entrenamiento y afinamiento del modelo de NLP	5
2.2	Dockerización e integración del modelo NLP	4
2.3	Levantamiento de la base de datos NoSQL	3
2.4	Integración con los servicios del GTSI	5
2.5	Pruebas y corrección de la integración con el GTSI	2-3
	Revisión y retroalimentación	
3.1	Pruebas y corrección preliminar con el cliente	2-3
3.2	Prueba y corrección final con GTSI	3
3.3	Prueba y corrección final con el cliente	3
3.4	Desarrollo de manual técnico.	3
3.5	Entrega final	1

Ver la planificación semanal en el Anexo B.

## **Capítulo 3**

### **3. Desarrollo**

En este capítulo se detalla el plan de implementación mismo que se presentará en diferentes fases que destacan en desarrollo de la solución: Preprocesamiento de los datos, entrenamiento del modelo, levantamiento de la API, construcción del módulo web, Integración y pruebas.

#### **3.1 Fase 1: Procesamiento de los datos**

En esta fase se realiza el procesamiento de los datos utilizando conocimientos de ingeniería de datos.

##### ***3.1.1 Obtención de Datos***

La fase inicial de recopilación de datos se llevó a cabo en colaboración con la Gerencia de Tecnología y Sistemas de Información (GTSI) de ESPOL, quienes proporcionaron conjunto de datos primario en formato CSV. El dataset contiene información sobre proyectos alineados con los Objetivos de Desarrollo Sostenible (ODS), con un total de 7272 entradas. Cada entrada incluye un título del proyecto, el ODS correspondiente, una descripción del problema que se aborda, los resultados esperados, y el impacto proyectado, todo en formato HTML.

Los datos proporcionados resultaron ser insuficientes para el entrenamiento del modelo. Por lo tanto, se decidió buscar un conjunto de datos más amplio y detallado. Se optó por utilizar el OSDG Community Dataset (OSDG-CD), que es el dataset de la comunidad para los Objetivos de Desarrollo Sostenible. Este dataset ofrece un mayor volumen de información y una metodología detallada para su obtención. [22]

### ***3.1.2 Preparación de datos***

El dataset original, compuesto por 43,025 filas y 7 columnas, incluía identificadores de documentos (doi y text\_id), el contenido textual (text), la meta de desarrollo sostenible asignada (sdg), y métricas de etiquetado como labels\_negative, labels\_positive, y el porcentaje de acuerdo entre etiquetas (agreement). Para simplificar el análisis, primero se seleccionaron solo las columnas relevantes: text, sdg, y agreement, manteniendo así la información esencial en un formato más manejable.

A continuación, se aplicaron filtros para asegurar la calidad del dataset. Se convirtieron los valores de la columna agreement a numéricos y se filtraron los registros para incluir únicamente aquellos con un acuerdo superior o igual al 75%. Este filtrado permitió centrarse en ejemplos donde existía un alto consenso entre las etiquetas, lo que es crucial para la precisión de cualquier análisis posterior.

Finalmente, se aplicó un último filtro basado en la longitud del texto, eliminando aquellos registros cuyo contenido superaba los 1024 caracteres. Este paso fue importante para garantizar que los textos fueran de una longitud manejable, facilitando su procesamiento y análisis. Como resultado de todo este preprocesamiento, se obtuvo un dataset final compuesto por 20,861 filas y 3 columnas, optimizado para un análisis más preciso y eficiente.

### ***3.1.3 Visualización de datos***

En esta etapa se generaron tablas y visualizaciones con el fin de analizar los datos de forma ordenada con las características necesarias para el procesamiento y entrenamiento del modelo. Se crearon tablas descriptivas uniendo las variables claves (título, problemática,

resultados, impacto). Esto permitió conocer la distribución de la información en cada una de estas características permitiendo la creación de gráficos estadísticos como histogramas enfocados en visualizarla distribución de caracteres en cada variable clave con el objetivo de determinar la variable más relevante para el entrenamiento.

### **3.2 Fase 2: Entrenamiento del modelo**

El proceso de entrenamiento comenzó con la división del dataset en conjuntos de entrenamiento y validación, asegurando una distribución estratificada de las clases. Los índices de los datos fueron etiquetados y los textos fueron tokenizados con el BertTokenizer para convertirlos en una forma compatible con el modelo BERT. Se añadieron tokens especiales, se generaron máscaras de atención y se ajustó la longitud de las secuencias.

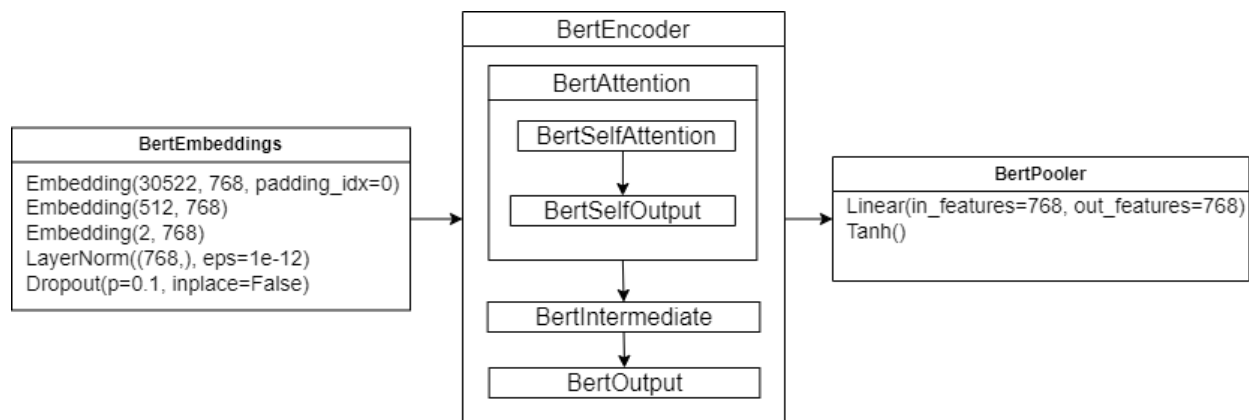
A continuación, se prepararon los datos tokenizados en tensores y se configuró el modelo BertForSequenceClassification para clasificación con 17 etiquetas. Los datos se cargaron en DataLoader utilizando RandomSampler para el entrenamiento y SequentialSampler para la validación, estableciendo un tamaño de lote de 8.

Durante el entrenamiento, se utilizó el optimizador AdamW con una agenda de tasa de aprendizaje ajustada. En cada época, el modelo fue entrenado y actualizado con la retropropagación, y se guardó el estado del modelo. El modelo fue evaluado en el conjunto de validación al final de cada época para calcular la pérdida y la puntuación F1 ponderada, proporcionando una visión del rendimiento y de los ajustes necesarios. En la figura 16 se muestra grosso modo la estructura del modelo de BERT implementado.



**Figura 16**

*Estructura funcional del modelo BERT*



### 3.3 Fase 3: Levantamiento de la API

Para poner en marcha la API, se comenzó instalando Docker en la máquina. El primer paso fue configurar el repositorio de Docker para que el sistema pudiera acceder a los paquetes necesarios. Esto implicó agregar una clave de seguridad y configurar el repositorio en el sistema. Una vez hecho esto, se procedió a instalar Docker y se verificó que la instalación se había realizado correctamente ejecutando un comando de prueba que confirmaba que Docker estaba funcionando bien.

A continuación, se instaló y configuró Nginx para gestionar el tráfico hacia la API. Primero, se actualizó el sistema e instaló Nginx. Luego, se ajustó la configuración de Nginx para redirigir las solicitudes hacia la API. Esto incluyó especificar el puerto en el que estaba funcionando la API y ajustar algunos detalles para asegurar que Nginx manejara las solicitudes correctamente. Finalmente, se habilitó la configuración de Nginx y se revisó su estado para confirmar que estaba funcionando adecuadamente.

Por último, se construyeron y ejecutaron las aplicaciones necesarias usando Docker. Esto incluyó crear y ejecutar contenedores tanto para la API como para el frontend de la aplicación. Como alternativa, se utilizó una herramienta que simplificó este proceso para que todos los componentes funcionaran de manera coordinada. Si también se necesitaba gestionar el código fuente del proyecto, se instaló y configuró Git LFS para descargar los archivos grandes requeridos. Con estos pasos, el entorno de la API quedó listo para funcionar.

### **3.4 Etapa 4: Construcción del módulo web**

En esta fase se desarrolla el prototipo en base a los requerimientos funcionales y no funcionales tratados por el cliente, se presentan las ventanas desarrolladas con su respectivo desempeño.

#### ***3.4.1 Vista Principal***

La vista principal de la aplicación (ver figura 17) está diseñada con un encabezado que precede a una caja de texto. Esta caja de texto está destinada a recibir los textos o párrafos en inglés o español para su análisis, con un límite de caracteres mínimo de 128 y máximo de 4096 por entrada. Se optó por 3 botones, un botón que permite la carga de archivos en formato PDF, y otros 2 botones independientes para procesar el texto y los archivos por separado.

La interfaz también presenta los 17 ODS representados como íconos, los cuales, al ser seleccionados, despliegan ventanas emergentes (ver Figura 20) que detallan el compromiso de la ESPOL con cada uno de los ODS. Finalmente, se ha implementado una sección denominada “Acerca de”, en el pie de página, donde se expone la problemática abordada y la solución

propuesta, con el objetivo de que el usuario comprenda más a fondo el funcionamiento y propósito del modelo desarrollado.

**Figura 17**

*Ventana Principal*

**OBJETIVOS DE DESARROLLO SOSTENIBLE**

**EVALUANDO LA CONTRIBUCIÓN DE SU PRODUCTO ACADÉMICO A LOS ODS**

**SOSTENIBILIDAD espol**

**Cuantificando la Relación entre su Producto Académico y los ODS**

Ingresar tu texto aquí (Sólomente inglés o español)

Seleccionar archivo Sin archivos seleccionados ⓘ 0/4096

Procesar archivo Procesar texto

**1 FIN DE LA POBREZA**  
**2 HAMBRE CERO**  
**3 SALUD Y BIENESTAR**  
**4 EDUCACIÓN DE CALIDAD**  
**5 IGUALDAD DE GÉNERO**  
**6 AGUA LIMPIA Y SANEAMIENTO**

**7 ENERGÍA ASEQUIBLE Y NO CONTAMINANTE**  
**8 TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO**  
**9 INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA**  
**10 REDUCCIÓN DE LAS DESIGUALDADES**  
**11 CIUDADES Y COMUNIDADES SOSTENIBLES**  
**12 PRODUCCIÓN Y CONSUMO RESPONSABLES**

**13 ACCIÓN POR EL CLIMA**  
**14 VIDA SUBMARINA**  
**15 VIDA DE ECOSISTEMAS TERRESTRES**  
**16 PAZ, JUSTICIA E INSTITUCIONES SÓLIDAS**  
**17 ALIANZAS PARA LOGRAR LOS OBJETIVOS**

**OBJETIVOS DE DESARROLLO SOSTENIBLE**

**ACERCA DE**

La Escuela Superior Politécnica del Litoral (ESPOL) genera una gran cantidad de documentos académicos (propuestas de investigación, artículos, tesis, etc.) que abordan diversas temáticas. Relacionar manualmente estos documentos con los Objetivos de Desarrollo Sostenible (ODS) es un proceso tedioso para los expertos y depende en gran medida de su conocimiento personal sobre la vinculación entre las temáticas y los ODS. Si bien la ESPOL cuenta con especialistas en múltiples campos, podrían pasarse por alto contribuciones a los ODS que se encuentran fuera de su área de expertise.

Nuestro sitio web pretende ayudar a los académicos de Espol, y cualquiera que lo requiera, a clasificar automáticamente los productos académicos según su contribución a cada uno de los 17 ODS.

Nuestro sitio está impulsado por Inteligencia Artificial (IA), específicamente por las técnicas de Procesamiento de Lenguaje Natural (PLN) y Aprendizaje Supervisado.

Inspirados por el éxito de modelos como BERT (Representaciones Bidireccionales de Codificadores a partir de Transformadores), desarrollados por investigadores de Google, nuestra plataforma utilizará técnicas similares para el pre-entrenamiento del PLN.

Para lograr una clasificación precisa, entrenaremos nuestro algoritmo con un conjunto de datos masivo de productos académicos previamente etiquetados de acuerdo a su contribución a los ODS. De esta manera, identificará patrones lingüísticos que le permitan clasificar automáticamente nuevos documentos con una precisión del 87%.

### 3.4.2 Vista de Resultados

La solución posee 2 vistas de resultados similares. La primera vista (ver figura 18) es para mostrar los resultados del texto, esta consta de su ODS respectivo y el porcentaje de coincidencia con el texto proporcionado

**Figura 18**

*Ventana resultados: Análisis de texto*



La segunda vista (ver figura 19) muestra de igual forma el ODS y un porcentaje de coincidencia como la anterior, pero además proporciona al usuario la información del documento cargado junto con la información del texto analizado.

Figura 19

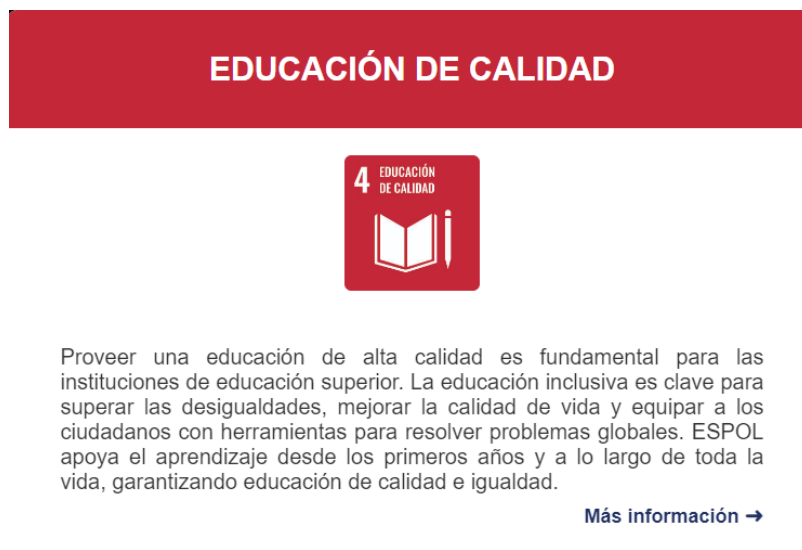
Ventana resultados: Análisis de archivos



### 3.4.3 Vistas adicionales

Estas vistas son recursos que se utilizan para fomentar una buena experiencia de usuarios y proporcionar información adicional entre ellas tenemos:

La ventana emergente (ver Figura 20) de los ODS que posee la información de ESPOL y su relación con los ODS, misma que se despliega después de dar clic en los ODS en forma de íconos que se encuentran en la ventana principal (ver Figura 17).

**Figura 20***Ventana emergente: Card ODS*

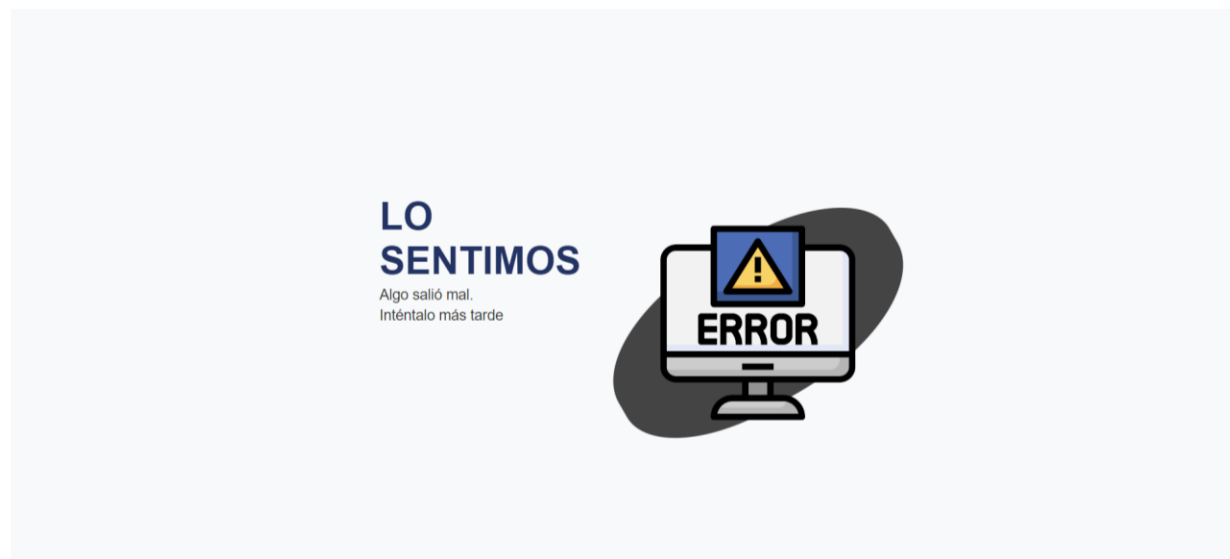
La ventana de carga (ver Figura 21) que aparece mientras el API está analizando el texto o el documento cargado. Dado que la ventana es un componente de React se debe especificar, de una manera u otra, qué es lo que se está analizando, para que el usuario reciba el mensaje acorde al contexto.

**Figura 21***Ventana de carga*

La ventana de error (ver Figura 22) que se presenta para indicar al usuario que existió algún error en el procesamiento del contenido. Esta ventana nace de la necesidad de mostrarle al usuario que algo salió mal evitando que el sistema web se caiga.

**Figura 22**

*Ventana de error*



La ventana emergente de requerimientos obligatorios (ver Figura 23) que surge cuando el usuario se encuentra en la pantalla principal o cuando se da clic al ícono de información para advertir de las limitaciones del programa, mejorar la experiencia de usuario e informar de manera concreta los requerimientos mínimos e importantes que debe poseer el archivo.

**Figura 23**

*Ventana emergente: Requerimientos obligatorios*



## Información

Nuestro sistema procesa archivos PDF académicos (hasta 10MB) en **español o inglés**.

Asegúrate de que el resumen (*abstract*) esté entre las primeras páginas y sea de una columna para un análisis óptimo.

OK

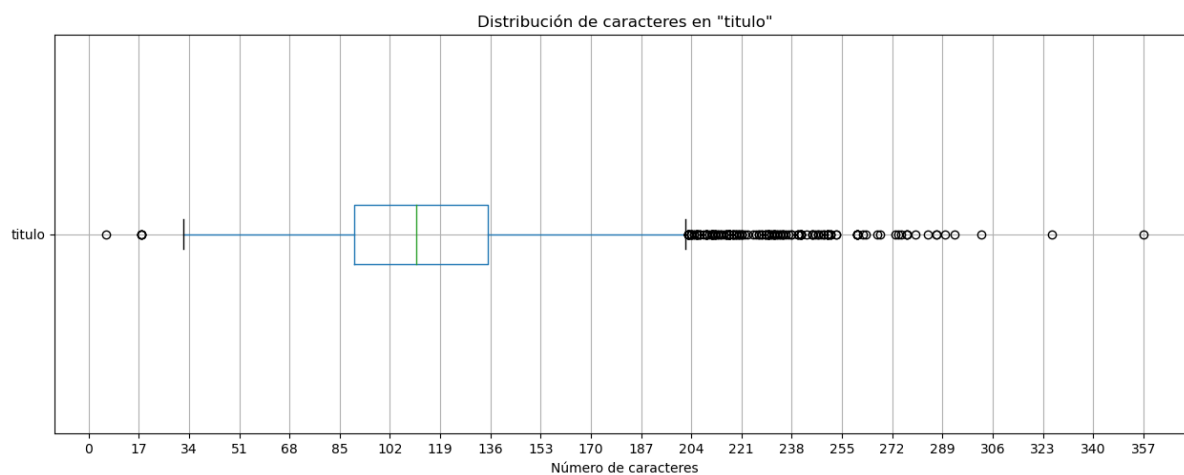
En caso de ver el formulario de feedback consultar Anexo C.

## 3.5 Resultados y Análisis

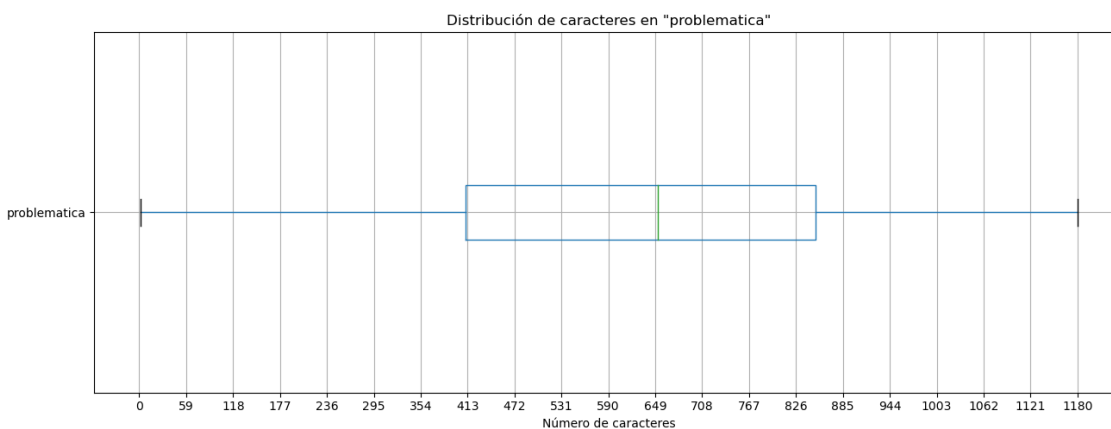
### 3.5.1 *Procesamiento de los datos.*

Del procesamiento se construyeron varios gráficos de cajas para analizar la distribución de caracteres en las variables significativas y elegir cuál o cuáles serían conveniente para el entrenamiento (ver Figura 24, 25, 26 y 27).



**Figura 24***Distribución de caracteres en título*

El gráfico de la Figura 24 muestra la distribución de caracteres en "título", con una mediana de 119 caracteres y un rango intercuartílico entre 102 y 136 caracteres. La mayoría de los títulos están en este rango, aunque hay valores atípicos, especialmente en el extremo derecho, con títulos que alcanzan hasta 357 caracteres. También hay algunos títulos cortos, con un mínimo de 34 caracteres. La concentración de títulos entre 204 y 238 caracteres sugiere que la mayoría son de longitud moderada, con algunos fuera de la norma.

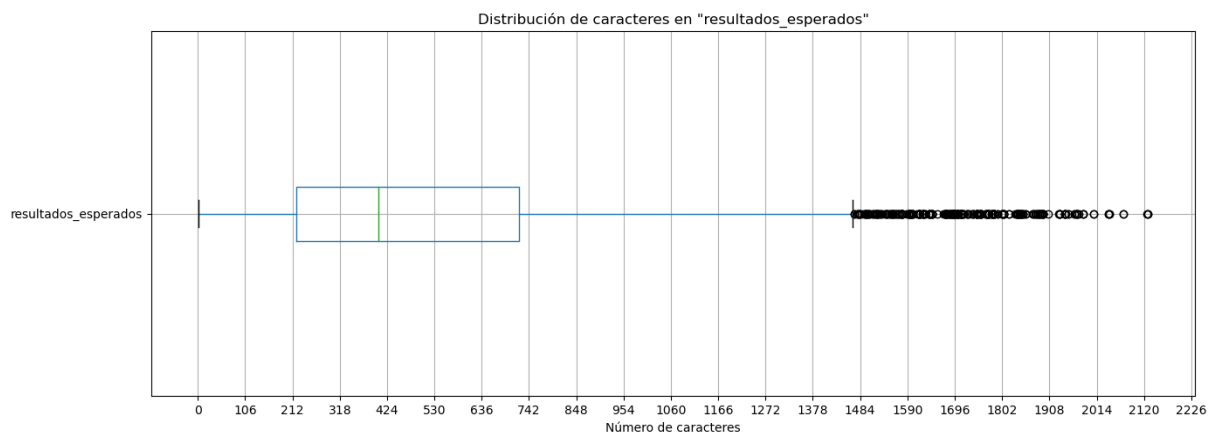
**Figura 25***Distribución de caracteres en problemática*

El gráfico de la Figura 25 muestra la distribución de caracteres en el campo "problemática". Se observa que la mayoría de los textos tienen entre aproximadamente 472 y 826 caracteres, con un valor mediano cercano a 649 caracteres. Hay algunos puntos que se ubican como valores atípicos en el extremo derecho, lo que indica que hay muy pocos textos con una cantidad de caracteres superior a los 1180. La variabilidad en la longitud de los textos es moderada, con una caja intercuartílica relativamente amplia, sugiriendo una diversidad en la cantidad de caracteres utilizados.

En la Figura 26, el campo "resultados\_esperados" presenta una distribución diferente. La mediana se sitúa alrededor de 424 caracteres, y el rango intercuartílico está comprendido entre 318 y 636 caracteres. Sin embargo, hay una gran cantidad de valores atípicos en el extremo derecho, lo que indica que hay textos que contienen hasta más de 2200 caracteres. Esto sugiere una mayor variabilidad en los textos de "resultados\_esperados", con algunos documentos siendo significativamente más extensos que otros.

**Figura 26**

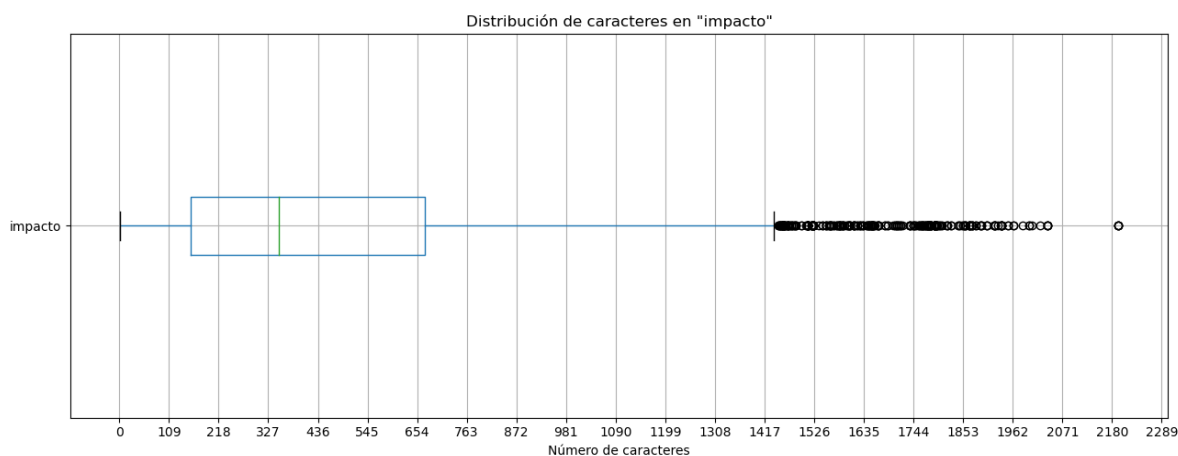
*Distribución de caracteres en resultados*



El gráfico de la Figura 27, "impacto", muestra una distribución similar a "resultados\_esperados", con una mediana de alrededor de 545 caracteres y un rango intercuartílico que va desde 436 a 763 caracteres. Al igual que en el gráfico anterior, se observa una notable cantidad de valores atípicos, con textos que llegan a tener más de 2200 caracteres. Esto indica que, aunque la mayoría de los textos siguen una longitud relativamente estándar, existe una minoría que incluye un número de caracteres considerablemente mayor, sugiriendo que algunos artículos profundizan mucho más en el tema del impacto.

**Figura 27**

*Distribución de caracteres en impacto*



Basado en los análisis de distribución de caracteres, es razonable suponer que el título podría descartarse en el entrenamiento del modelo de IA, ya que, aunque tiene una longitud moderada y estable, presenta varios valores atípicos y su capacidad informativa parece limitada en comparación con los campos de problemática, resultados esperados e impacto, los cuales muestran una mayor diversidad y extensión, lo que indica que contienen información más rica y relevante para el análisis. Esto sugiere que esos campos podrían aportar más valor al modelo en términos de contenido significativo.

### 3.5.2 Entrenamiento del modelo

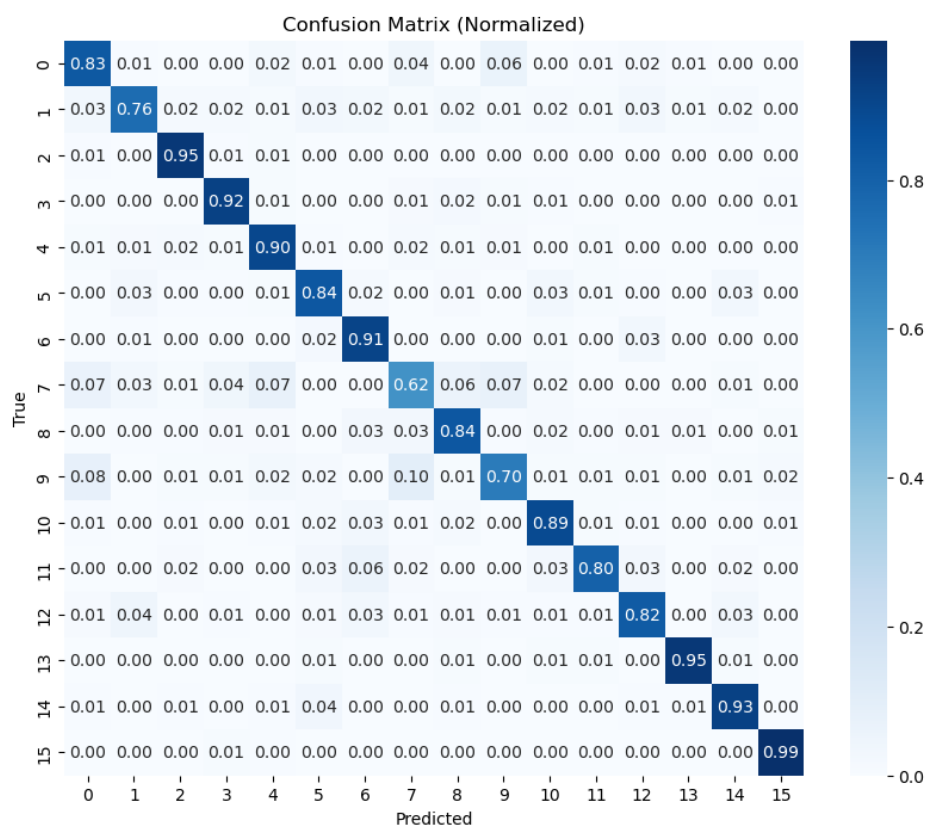
Luego de 10 épocas de entrenamiento y sobre una muestra de 3136 ejemplos que el modelo no vio durante el proceso de entrenamiento, se calcula la exactitud (accuracy), precisión, recuperación (recall), y el F1-score, todos con un enfoque ponderado. Los resultados obtenidos fueron:

- **Accuracy:** 0.8744
- **Precision:** 0.8737
- **Recall:** 0.8744
- **F1-score:** 0.8739

Estos valores indican un alto rendimiento del modelo en la clasificación de textos, con un equilibrio adecuado entre precisión y recuperación. En la Matriz de Confusión de la Figura 28, se observa el desempeño del modelo en la clasificación de cada uno de los ODS. El modelo mostró una precisión cercana al 100% en la clasificación de textos relacionados con el ODS 15: Vida de Ecosistemas Terrestres, mientras que para el ODS 7: Energía Asequible y no Contaminante, la precisión fue de apenas un 62%.

Figura 28

*Matriz de Confusión Normalizada*




















De una muestra de 36 trabajos académicos de materia integradora se obtuvieron los siguientes resultados al comparar la clasificación realizada por los estudiantes con la clasificación realizada por el modelo de PLN. En la Figura 29 se observa que el 42% de los productos académicos coinciden en la asignación del ODS más relevante. En la Figura 30 observamos el 58% restante de los proyectos que no coinciden en el ODS más relevante.

**Figura 29***Resultados de clasificación parcial*

Código Proyecto	ODS Estudiantes	ODS Modelo PLN	
0	INGE-1983	[2, 9, 12]	[2, 12, 14, 6, 3]
5	INGE-2100	[15, 8, 11]	[15, 12, 6, 11, 3]
6	ADMI-906	[11, 13, 17]	[11, 6, 5, 15, 2]
15	INGE-1988	[3]	[3, 14, 2, 6, 12]
16	INGE-2106	[9]	[9, 11, 2, 14, 16]
21	INGE-2155	[6, 11, 12, 13, 15]	[6, 15, 12, 2, 14]
22	INGE-2011	[9, 12]	[9, 8, 5, 4, 2]
24	ADMI-879	[8, 9, 17]	[8, 12, 9, 4, 5]
26	ARTE-409	[3, 4]	[3, 10, 14, 16, 13]
29	INGE-2096	[6, 11, 14]	[6, 14, 15, 2, 10]
31	INGE-2187	[6, 7, 11]	[6, 7, 11, 15, 2]
33	MATE-177	[7]	[7, 12, 6, 10, 3]
34	TECH-312	[4]	[4, 9, 16, 12, 13]
35	TECH-310	[9, 17]	[9, 4, 8, 11, 2]
36	INGE-2159	[6, 8, 15]	[6, 15, 7, 2, 11]

 <b>1 FIN DE LA POBREZA</b>
 <b>2 HAMBRE CERO</b>
 <b>3 SALUD Y BIENESTAR</b>
 <b>4 EDUCACIÓN DE CALIDAD</b>
 <b>5 IGUALDAD DE GÉNERO</b>
 <b>6 AGUA LIMPIA Y SANEAMIENTO</b>
 <b>7 ENERGÍA ASEQUIBLE Y NO CONTAMINANTE</b>
 <b>8 TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO</b>
 <b>9 INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA</b>
 <b>10 REDUCCIÓN DE LAS DESIGUALDADES</b>
 <b>11 CIUDADES Y COMUNIDADES SOSTENIBLES</b>
 <b>12 PRODUCCIÓN Y CONSUMO RESPONSABLES</b>
 <b>13 ACCIÓN POR EL CLIMA</b>
 <b>14 VIDA SUBMARINA</b>
 <b>15 VIDA DE ECOSISTEMAS TERRESTRES</b>
 <b>16 PAZ, JUSTICIA E INSTITUCIONES SÓLIDAS</b>
 <b>17 ALIANZAS PARA LOGRAR LOS OBJETIVOS</b>

**Figura 30***Resultados de clasificación parcial*

	Código Proyecto	ODS Estudiantes	ODS Modelo PLN
1	INGE-2000	[8, 12]	[9, 8, 14, 11, 4]
2	INGE-2001	[8]	[12, 7, 2, 15, 14]
3	INGE-2002	[16, 3, 9]	[7, 8, 12, 6, 9]
4	INGE-2003	[8, 9, 13]	[9, 8, 11, 5, 12]
7	ADMI-866	[4, 17]	[8, 4, 12, 10, 13]
8	MATE-168	[9]	[12, 2, 1, 14, 10]
9	INGE-2015	[9]	[11, 9, 1, 6, 15]
10	INGE-2022	[12]	[11, 1, 15, 9, 6]
11	INGE-2024	[11, 3, 6]	[6, 11, 15, 2, 7]
12	INGE-2025	[3, 6]	[6, 2, 7, 15, 11]
13	INGE-1990	[9, 12]	[14, 16, 0, 2, 9]
14	INGE-2058	[4, 8, 9]	[6, 14, 7, 2, 4]
17	INGE-2201	[3, 6, 9]	[6, 2, 15, 12, 7]
18	ADMI-937	[3, 9, 16]	[8, 10, 16, 3, 4]
19	ADMI-871	[1]	[11, 6, 3, 5, 15]
20	INGE-2094	[8, 12, 14]	[14, 2, 16, 0, 13]
23	ADMI-900	[8, 17]	[4, 16, 1, 3, 0]
25	INGE-2123	[4, 9]	[9, 16, 5, 4, 2]
27	ARTE-402	[13, 14, 15]	[15, 11, 6, 10, 4]
28	ARTE-396	[8, 4]	[16, 8, 5, 9, 4]
30	INGE-2017	[4, 9]	[9, 16, 4, 8, 0]
32	INGE-2065	[8, 9, 12, 17]	[7, 6, 12, 14, 8]

En la Figura 31 vemos las asignaciones de ODS que coinciden en la primera o segunda prioridad de forma bidireccional, esto representa un 48% de los proyectos que no fueron clasificados con coincidencia exacta.

**Figura 31***Resultados de clasificación parcial*

	<b>Código Proyecto</b>	<b>ODS Estudiantes</b>	<b>ODS Modelo PLN</b>
<b>1</b>	INGE-2000	[8, 12]	[9, 8, 14, 11, 4]
<b>4</b>	INGE-2003	[8, 9, 13]	[9, 8, 11, 5, 12]
<b>7</b>	ADMI-866	[4, 17]	[8, 4, 12, 10, 13]
<b>9</b>	INGE-2015	[9]	[11, 9, 1, 6, 15]
<b>11</b>	INGE-2024	[11, 3, 6]	[6, 11, 15, 2, 7]
<b>12</b>	INGE-2025	[3, 6]	[6, 2, 7, 15, 11]
<b>17</b>	INGE-2201	[3, 6, 9]	[6, 2, 15, 12, 7]
<b>25</b>	INGE-2123	[4, 9]	[9, 16, 5, 4, 2]
<b>28</b>	ARTE-396	[8, 4]	[16, 8, 5, 9, 4]
<b>30</b>	INGE-2017	[4, 9]	[9, 16, 4, 8, 0]

Finalmente, en la Figura 32 incluimos un análisis manual para resolver la disputa entre la clasificación hecha por los estudiantes y la realizada por el modelo de PLN. Como resultado de la disputa cada método de clasificación se queda con el 50% en comparación con la revisión manual. Esto significa que en 6 ocasiones el modelo no fue capaz de predecir un ODS significativo. Si analizamos el total de la muestra el modelo arrojó respuestas correctas el 83% de las veces.



**Figura 32***Resultados de clasificación parcial*

	<b>Código Proyecto</b>	<b>ODS Estudiantes</b>	<b>ODS Modelo PLN</b>	<b>Revisión Manual</b>
0	INGE-2001	[8]	[12, 7, 2, 15, 14]	[12]
1	INGE-2002	[16, 3, 9]	[7, 8, 12, 6, 9]	[9]
2	MATE-168	[9]	[12, 2, 1, 14, 10]	[9]
3	INGE-2022	[12]	[11, 1, 15, 9, 6]	[11]
4	INGE-1990	[9, 12]	[14, 16, 0, 2, 9]	[9]
5	INGE-2058	[4, 8, 9]	[6, 14, 7, 2, 4]	[9]
6	ADMI-937	[3, 9, 16]	[8, 10, 16, 3, 4]	[8]
7	ADMI-871	[1]	[11, 6, 3, 5, 15]	[11]
8	INGE-2094	[8, 12, 14]	[14, 2, 16, 0, 13]	[14]
9	ADMI-900	[8, 17]	[4, 16, 1, 3, 0]	[17]
10	ARTE-402	[13, 14, 15]	[15, 11, 6, 10, 4]	[15]
11	INGE-2065	[8, 9, 12, 17]	[7, 6, 12, 14, 8]	[9]

### **3.5.3 Construcción del módulo web**

Como resultado se obtuvo un producto de uso intuitivo para el usuario que cumple con las funcionalidades requeridas. Durante la fase de desarrollo, se aplicaron implementaciones ágiles, lo que permitió iteraciones rápidas y una continua adaptación a las necesidades del cliente. El diseño se centró en la experiencia del usuario, asegurando una interfaz clara y fácil de navegar, con resultados sencillos de interpretar. Además, se llevaron a cabo pruebas exhaustivas de usabilidad y rendimiento, garantizando la fiabilidad y eficiencia del módulo en diversas plataformas y dispositivos.

Los retos que implicó la solución giran alrededor del consumo de la API, pues para que esta funcione con regularidad se tiene que enviar la información a procesar en el formato

correcto y la respuesta entregada debe ser procesada correctamente para su renderización. Al final, el cliente aceptó el producto final, destacando su satisfacción por cumplir con los objetivos establecidos.

### **3.6 Análisis de costos**

Debido a que el sistema se albergará en los servidores de la ESPOL, los gastos relacionados con la infraestructura y el alojamiento de servidor no se considerarán en este análisis. En cambio, se concentra únicamente en los costos directos del desarrollo del sistema y del modelo PLN, proporcionando un desglose detallado de los recursos financieros necesarios para su creación e implementación.

Para este análisis financiero se llevará a cabo la remuneración por hora trabajada, para calcular el salario por hora de un programador junior que gana \$1200 mensuales, necesitamos hacer algunas suposiciones[23]:

Horas trabajadas al mes: Generalmente, se considera una jornada laboral de 8 horas diarias, 5 días a la semana. Esto equivale a 40 horas semanales.

Número de semanas trabajadas al mes: Asumiremos un promedio de 4.3 semanas trabajadas al mes (52 semanas al año / 12 meses).

Cálculo: Horas trabajadas al mes:  $40 \text{ horas/semana} * 4.3 \text{ semanas/mes} = 172 \text{ horas/mes}$ .

El salario por hora:  $\$1200/\text{mes} / 172 \text{ horas/mes} \approx \$6.98/\text{hora}$  que finalmente se redondeará a \$7.00/hora.

Entonces, según estos cálculos, un programador junior que gana \$1200 mensuales, sin considerar algún otro beneficio o bonificación, ganaría aproximadamente \$7.00 por hora. En la Tabla 5 se aprecia el desglose de las actividades realizadas para lograr el producto final.

**Tabla 5**

*Costos de desarrollo e implementación*

Componente	Horas de Trabajo	Costo por Hora (USD)	Costo Total (USD)
<b>Levantamiento de Requerimientos</b>			
Definición de Objetivos y Plazos Establecidos	5	7.00	35.00
Historias de Usuario	5	7.00	35.00
Requerimientos Funcionales	10	7.00	70.00
Requerimientos no Funcionales	10	7.00	70.00
<b>Diseño y Planificación</b>			
Desarrollo de Mockups	20	7.00	140.00
Diseño Arquitectónico	15	7.00	105.00
Implementación de Diagrama ER	30	7.00	210.00
Prototipado	40	7.00	280.00
<b>Desarrollo del Proyecto</b>			
<i>Desarrollo del Front-End</i>			
Modelado de las Vistas	50	7.00	350.00
Maquetación	50	7.00	350.00
Manejo de Interactividad	50	7.00	350.00
<i>Desarrollo del Back-End</i>			
Procesamiento de Datos	75	7.00	525.00
Construcción del Modelo	100	7.00	700.00
Entrenamiento del Modelo	150	7.00	1,050.00
Levantamiento del API	75	7.00	525.00
<b>Pruebas e Integración</b>			
Pruebas Unitarias	20	7.00	140.00
Integración del Front-End con el Back-End	50	7.00	350.00
Pruebas de Integración	30	7.00	210.00
Pruebas de Usabilidad	20	7.00	140.00
<b>Puesto en Producción</b>			
Despliegue en el Servidor	30	7.00	210.00
Documentación del Proyecto	20	7.00	140.00
<b>Gestión</b>			
Gestión del Proyecto	30	7.00	210.00
Reuniones de Seguimiento con el Cliente	20	7.00	140.00
	905		6,335.00

Finalmente, este proyecto deja un recaudo de \$6,335 con 905 horas trabajadas.

## Capítulo 4

## **4. Conclusiones y Recomendaciones**

### **4.1 Conclusiones**

- En este proyecto, se logró desarrollar un sistema automatizado capaz de clasificar productos académicos según su alineación con los Objetivos de Desarrollo Sostenible (ODS). Utilizando técnicas avanzadas de inteligencia artificial (IA), específicamente procesamiento de lenguaje natural (PLN), se creó una API que mejora significativamente la precisión y eficiencia en la clasificación de textos académicos.
- El modelo de clasificación automatizada, inspirado en BERT, demostró ser efectivo en la asignación de ODS con un 87% de efectividad, aunque se identificaron limitaciones, especialmente en cuanto al procesamiento de textos en español y la necesidad de datos y recursos adicionales para mejorar el rendimiento del sistema.
- Se destaca que el producto final cumple con los objetivos establecidos, ofreciendo una interfaz intuitiva y una experiencia de usuario mejorada. Las pruebas realizadas aseguraron la fiabilidad y eficiencia de la API, lo que fue valorado positivamente por el cliente. No obstante, la implementación del sistema también reveló desafíos relacionados con el consumo correcto de la API.

### **4.2 Recomendaciones**

- Ampliación del modelo para textos en español: Es recomendable entrenar el modelo con una mayor cantidad de datos en español, lo que permitiría mejorar su precisión y capturar mejor las sutilezas contextuales de textos académicos en este idioma o en su defecto crear un modelo que sólo procese contenido de este idioma.

- Actualización continua: Para el año 2030 es posible que los ODS se actualicen por lo que se recomienda realizar un nuevo entrenamiento y/o desarrollo de un nuevo modelo con los criterios que incluya la actualización del modelo con nuevos datos y ajustes según cambios en los criterios de clasificación de los ODS.
- Optimización del consumo de API: Se recomienda continuar mejorando el manejo del consumo de la API para minimizar errores en el procesamiento de datos, maximizar el rendimiento y garantizar una experiencia de usuario fluida, independientemente del formato y tamaño de los textos procesados.
- Mejora continua: Implementar técnicas ontológicas basadas en el reconocimiento de palabras claves en los textos académicos para aumentar la precisión del clasificador.

## Referencias

- [1] *Objetivos de Desarrollo Sostenible (ODS)*, 2015. [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [2] F. Gómez Isa, «Los Objetivos de Desarrollo Sostenible (ODS): Hacia un nuevo contrato social intra e inter-generacional», *Est Deu*, vol. 70, n.º 2, pp. 191-224, dic. 2022, doi: 10.18543/ed.2650.
- [3] L. Rouhiainen, *Inteligencia Artificial: 101 cosas que debes saber hoy sobre nuestro futuro*, Primera edición. Barcelona, España: Editorial Planeta, S.A., 2018.
- [4] C. C. Aggarwal, «Machine Learning for Text: An Introduction», en *Machine Learning for Text*, Cham: Springer International Publishing, 2018, pp. 1-16. doi: 10.1007/978-3-319-73531-3\_1.
- [5] S. J. Russell, P. Norvig, y E. Davis, *Artificial intelligence: a modern approach*, 3rd ed. en Prentice Hall series in artificial intelligence. Upper Saddle River: Prentice Hall, 2010.
- [6] J. R. Méndez, F. Fdez-Riverola, y F. Díaz, «Sistemas inteligentes para la detección y filtrado de correo spam: una revisión», vol. 11, 2007.
- [7] S. Vallejos *et al.*, «Clasificación automática de publicaciones de productos en sitio de comercio electrónico», en *2022 IEEE Biennial Congress of Argentina (ARGENCON)*, San Juan, Argentina: IEEE, sep. 2022, pp. 1-8. doi: 10.1109/ARGENCON55245.2022.9939985.
- [8] C. Martínez, «Uso de las Técnicas de Preprocesamiento de Datos e Inteligencia Artificial (Lógica Difusa) en la Clasificación/Predicción del Riesgo Bancario», Tesis de grado, Universidad de Los Andes, Venezuela, 2007. [En línea]. Disponible en: [http://iies.faces.ula.ve/CDCHT/CDCHT%20E2170309B/PREGRADO/TESIS\\_CARLOS\\_MARTINEZ.pdf](http://iies.faces.ula.ve/CDCHT/CDCHT%20E2170309B/PREGRADO/TESIS_CARLOS_MARTINEZ.pdf)
- [9] M. M. Lopez y J. Kalita, «Deep Learning applied to NLP», 8 de marzo de 2017, *arXiv*: arXiv:1703.03091. Accedido: 31 de mayo de 2024. [En línea]. Disponible en: <http://arxiv.org/abs/1703.03091>
- [10] A. Cortez Vásquez, H. Vega Huerta, y J. Pariona Quispe, «Procesamiento de lenguaje natural», *Revistas de investigación Universidad Nacional Mayor de San Marcos*, vol. 6, n.º 2, pp. 45-54, dic. 2009.
- [11] IBM, «¿Qué es el aprendizaje supervisado?» [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/supervised-learning>
- [12] G. V. González, «Aprendizaje Supervisado: Métodos, Propiedades y Aplicaciones», Trabajo Fin de Grado, Universidad de Málaga, Málaga, España, 2022.

- [13] D. G. Retuerta, R. A. Bondía, J. P. Tejedor, y J. M. M. Rodríguez, «Inteligencia artificial para la asignación automática de categorías constructivas», 2018. [En línea]. Disponible en: <https://api.semanticscholar.org/CorpusID:217023639>
- [14] J. Á. González Barba, «Aprendizaje profundo para el procesamiento del lenguaje natural», *Master Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital, IARFID, Universidad Politécnica de Valencia*, vol. 6, n.º 2, p. 105, jul. 2017.
- [15] H. M. Q. Cosgaya, J. M. D. Lugo, y G. M. E. Segovia, «Clasificación automática de textos mediante comparación difusa y sistema experto en un controlador domótico», *Multidisciplinas de la Ingeniería*, oct. 2016, [En línea]. Disponible en: <https://api.semanticscholar.org/CorpusID:267475197>
- [16] Naciones Unidas, «La Inteligencia Artificial como herramienta para acelerar el progreso de los ODS», OBJETIVOS DE DESARROLLO SOSTENIBLE. [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/2017/10/la-inteligencia-artificial-como-herramienta-para-acelerar-el-progreso-de-los-ods/>
- [17] UNEP, «Measuring how your text is related to the sustainable development goals», SGD meter - UN environment programme. [En línea]. Disponible en: <https://www.unep.org/>
- [18] J. Devlin, M.-W. Chang, K. Lee, y K. Toutanova, «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding», 24 de mayo de 2019, *arXiv: arXiv:1810.04805*. Accedido: 9 de junio de 2024. [En línea]. Disponible en: <http://arxiv.org/abs/1810.04805>
- [19] P. Kruchten, «Architectural blueprints—The “4+ 1” view model of software architecture», *IEEE Software*, vol. 12, n.º 6, pp. 42-50, 1995.
- [20] *React*. (2023). JavaScript. [En línea]. Disponible en: <https://es.react.dev/>
- [21]. *NET Core*. (2022). Microsoft Corporation. [En línea]. Disponible en: <https://dotnet.microsoft.com/>
- [22] *OSDG Community Dataset (OSDG-CD)*. (1 de abril de 2024). UNDP IICPSD SDG AI Lab. [Dataset]. Disponible en: <https://zenodo.org/records/11441197>
- [23] Ministerio del Trabajo, «Salarios mínimos sectoriales y tarifas 2024». 31 de mayo de 2024. [En línea]. Disponible en: <https://www.eluniverso.com/noticias/economia/sueldos-ecuador-2024-tablas-sectoriales-excel-pdf-nota/>



## Anexos



## Anexo A: Historias de usuario

Identificador (ID) de la Historia	Enunciado de la Historia				Criterios de Aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Criterio de Aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
WEB - 01	Como académico	Necesito acceso a la API mediante una interfaz web.	Con la finalidad de clasificar un producto académico según su relación con los ODS.	1	Interfaz web	En caso de que el usuario quiera hacer uso del API a través de una interfaz web.	Cuando el usuario visite la url designada para el efecto.	Se mostrará una interfaz web en la que se pueden subir los PDF o el texto plano de los productos académicos y visualizar el resultado de su clasificación según su relación con su ODS.
PDF - 01	Como académico	Necesito cargar el producto académico en formato PDF.	Con la finalidad de clasificar el contenido según su relación con los ODS.	1	Formato de archivo incorrecto.	En caso de que el formato del archivo seleccionado sea distinto de PDF.	Cuando se suba el archivo para su análisis.	Se mostrará una alerta indicando el tipo de archivo aceptado.
				2	Peso de archivo excede límite permitido.	En caso de que el peso del archivo seleccionado supere el límite permitido.	Cuando se suba el archivo para su análisis.	Se mostrará una alerta indicando el peso máximo permitido.
				3	Archivo PDF con peso menor al máximo permitido.	En caso de que el formato del archivo seleccionado sea PDF y el peso	Cuando se suba el archivo para su análisis.	Se mostrará una alerta confirmando que el archivo se ha subido con éxito.

						menor al máximo permitido.		
TXT - 01	Como académico	Necesito cargar el contenido de un producto académico en texto plano.	Con la finalidad de clasificar el texto según su relación con los ODS.	1	Longitud del texto excede el máximo de caracteres.	En caso de que el texto exceda el límite máximo de caracteres.	Cuando se suba el texto para su análisis.	Se mostrará una alerta indicando el máximo de caracteres.
				2	Longitud del texto es menor al máximo de caracteres.	En caso de que la longitud del texto sea menor al límite máximo de caracteres.	Cuando se suba el texto para su análisis.	Se mostrará una alerta indicando que el texto ha sido subido con éxito.
LNG - 01	Como académico	Necesito analizar el producto académico o su contenido tanto en español como en inglés.	Con la finalidad de clasificar el contenido según su relación con los ODS.	1	Contenido en español.	En caso de que el PDF o el texto estén en español.	Cuando se suba el archivo o el texto para su análisis.	Se mostrará el resultado de la clasificación por ODS.
				2	Contenido en inglés.	En caso de que el PDF o el texto estén en inglés.		
OUT - 01	Como académico	Necesito visualizar en qué porcentaje se relaciona cada ODS con el producto académico.	Con la finalidad de asignar al producto académico el/los ODS correspondiente/s.	1	Porcentaje de relación con cada ODS.	En caso de que se haya subido un PDF o texto, ya sea este en español o en inglés, con éxito.	Cuando se envíe el archivo para su análisis.	Se mostrará un listado de los ODS ordenados de mayor a menor según el porcentaje de relación con el producto y mostrándose el porcentaje junto a cada ODS.



## Anexo C: Encuesta de retroalimentación


 

### Retroalimentación - Clasificador ODS

Este es un pequeño formulario que pretende escuchar al usuario y guardar sus comentarios para mejorar el desarrollo del modelo de Inteligencia Artificial.

Ante cualquier duda o inquietud puede dejar un correo en nuestro canal:  
[sostenibilidad@espol.edu.ec](mailto:sostenibilidad@espol.edu.ec)


\* Obligatorio

1. ¿Consideras que los datos de la clasificación fueron precisos \* 

Sí


No

Tal vez

2. Califica la precisión del modelo de IA al identificar el ODS más relevante para tu consulta. \* 


3. **Déjanos tus comentarios.**

Puedes mencionar el motivo de tu elección anterior, también puedes incluir alguna recomendación o simplemente describir cómo te sientes utilizando nuestro modelo.



Escriba su respuesta

Enviar

 Microsoft 365

Este contenido lo creó el propietario del formulario. Los datos que envíes se enviarán al propietario del formulario. Microsoft no es responsable de las prácticas de privacidad o seguridad de sus clientes, incluidas las que adopte el propietario de este formulario. Nunca des tu contraseña.

Microsoft Forms | Encuestas, cuestionarios y sondeos con tecnología de inteligencia artificial [Crear mi propio formulario](#)

El propietario de este formulario no ha proporcionado una declaración de privacidad sobre cómo utilizarán los datos de tus respuestas. No proporciones información personal o confidencial. | [Términos de uso](#)