

**APLICACIÓN WEB CON IA BASADA EN NLP PARA DETECTAR NOTICIAS FALSAS, UTILIZANDO SCRUM Y PRINCIPIOS DE ARQUITECTURA POR CAPAS PARA UNA IMPLEMENTACIÓN ÁGIL Y EFECTIVA**

**NLP-POWERED AI WEB APPLICATION FOR DETECTING FAKE NEWS, USING SCRUM AND LAYERED ARCHITECTURE PRINCIPLES FOR AGILE AND EFFECTIVE IMPLEMENTATION**

**Samuel Andres Celis Lizcano<sup>1</sup>, PhD. Jose Gerardo Chacon<sup>2</sup>**

<sup>1</sup> Universidad de Pamplona Tel.: 57 3229169092 Email: samuelcelislizcano@gmail.com

<sup>2</sup> Grupo Inteligência de dados y Computacional (GIIDAC), Universidad de Pamplona, Tel +57 3153964235, Email: jose.chacon @unipamplona.edu.co, ORCID <https://orcid.org/0000-0002-6582-3142>

**Abstract:** This project aims to develop a web application with artificial intelligence (AI) based on natural language processing (NLP) to detect fake news. Its importance lies in combating disinformation circulating on social media, promoting access to truthful information, and strengthening trust in digital media. News will be obtained from public online sources, such as news portals, social media, and verified news databases, to feed and train the detection models. The agile Scrum methodology will be used for its development, which allows for flexible, collaborative, and efficient management. Key functionalities such as text analysis and automatic classification will be defined in the Sprint Planning phase. During the Development phase, these tasks will be implemented in short cycles. During the Daily Scrum, the team will review progress and resolve blockers. At the end of each cycle, a Sprint Review will be held to evaluate the results and a Retrospective will be held to propose continuous improvements.

The system will be designed using a layered architecture, organizing the solution into presentation, business logic, and data access layers, which favors scalability and maintenance. The Model-View-Controller (MVC) design pattern will also be applied, ensuring a clear separation of responsibilities between the AI logic (model), the user interface (view), and workflow coordination (controller). The final product will be a modular, functional web application capable of detecting fake content. Functional, integration, and model accuracy tests will be performed to validate its operation, ensuring the system's quality and effectiveness in different scenarios.

**Keywords:** Fake News, Artificial Intelligence, Natural Language Processing (NLP), Web Application

**Resumen:** El presente proyecto tiene como objetivo desarrollar una aplicación web con inteligencia artificial (IA) basada en procesamiento de lenguaje natural (NLP) para detectar noticias falsas. Su importancia radica en combatir la desinformación que circula en redes sociales, promoviendo el acceso a información veraz y fortaleciendo la confianza en los medios digitales. Las noticias serán obtenidas de fuentes públicas en línea, como portales informativos, redes sociales y bases de datos de noticias

verificadas, para alimentar y entrenar los modelos de detección. Para su desarrollo, se empleará la metodología ágil Scrum, que permite una gestión flexible, colaborativa y eficiente. En la Planificación del Sprint se definirán funcionalidades clave como el análisis de textos y la clasificación automática. Durante la fase de Desarrollo, estas tareas se implementarán en ciclos cortos. En el Daily Scrum, el equipo verificará avances y resolverá bloqueos. Al finalizar cada ciclo, se realizará una Revisión del Sprint para evaluar los resultados y una Retrospectiva para proponer mejoras continuas. El sistema se diseñará bajo una arquitectura por capas, organizando la solución en capas de presentación, lógica de negocio y acceso a datos, lo que favorece la escalabilidad y el mantenimiento. Asimismo, se aplicará el patrón de diseño Modelo-Vista-Controlador (MVC), que garantiza una separación clara de responsabilidades entre la lógica de IA (modelo), la interfaz de usuario (vista) y la coordinación del flujo (controlador). El producto final será una aplicación web modular, funcional, capaz de detectar contenido falso. Para validar su funcionamiento se realizarán pruebas funcionales, de integración y de precisión del modelo, asegurando la calidad y efectividad del sistema en distintos escenarios.

**Palabras clave:** Noticias Falsas, Inteligencia Artificial, Procesamiento de Lenguaje Natural (NLP), Aplicación Web

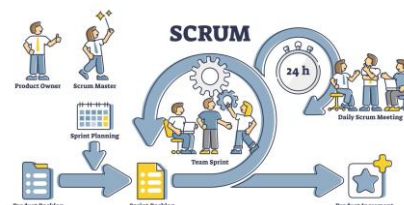
## 1. INTRODUCCION

Durante el desarrollo de la aplicación web con inteligencia artificial (IA) basada en procesamiento de lenguaje natural (NPL) para detectar noticias falsas se busca brindar una forma de combatir la desinformación en las redes sociales, permitiendo a los usuarios tener una herramienta que les permita verificar las noticias que leen día a día en sus Facebook o cuentas de X por nombrar algunas plataformas. Al ser una herramienta gratuita podrá ser usada por reporteros o investigadores para verificar fuentes e identificar la confiabilidad de las noticias. Por medio de investigaciones se identificó un proyecto publicado el 31 de mayo del 2023 el cual tiene como título “A Multi-tier Model and Filterin Approach to Detect Fake News Using Machine Learning Algorithms” el cual nos relata como por medio de un modelo multinivel logran etiquetar noticias de acuerdo a ciertos parámetros que ayudan a verificar la veracidad de la información, después de manejar la información por cada capa del modelo, finalmente enviando los resultados a la herramienta WEKA para aprobar los resultados a partir de ambos conjuntos de datos. (Chiung Chang Yu a, 2023) Teniendo presente la importancia de la información para realizarse su respectivo análisis, se optó por utilizar las redes sociales como Facebook y X (anteriormente llamado Twitter), siendo estos los sitios en donde la información de las noticias suele ser alterada para llamar a la gente y difundir de manera incorrecta los mensajes, la idea de utilizar estas dos redes sociales fue gracias al artículo publicado 4 de octubre del 2020 el cual lleva como título “Fake News and Aggregated Credibility: Conceptualizing a Co-Creative

Medium for Evaluation of Sources Online” en el cual se nos plantea como para debatir sobre la credibilidad de las fuentes en línea no es solo suficiente utilizar algoritmos informáticos por lo tanto los autores a lo largo del artículo plantean una propuesta conceptual que busca respaldar la evaluación de la credibilidad de las fuentes en línea utilizando el crowdsourcing (Agnieszka Jaff, 2020).

Para poder llevar a cabo el objetivo principal de este proyecto se eligió utilizar la metodología SCRUM por sus múltiples iteraciones de sus fases que encontramos en la Ilustración 1, las cuales son: Planificación del Sprint, Desarrollo, Daily Scrum, Revisión del Sprint, retrospectiva. Gracias a la naturaleza de SCRUM que consiste en iteraciones de cuantos Sprints se hubiesen definido en el principio del proyecto permite enfocarse en una parte del proyecto a la vez, dejando así paso para realizar todo por capas de manera efectiva, en caso de no lograr una parte del proyecto en el tiempo esperado, en el Daily Scrum se reajustarán los tiempos para no afectar el tiempo general del proyecto permitiendo así terminar lo que no se logró hacer en su tiempo determinado.

**Figura 1 Fases de la Metodología SCRUM**



**Nota:** Adaptado de Figura 1, por ILUNION, 2025, <https://www.ilunion.com/es/blog-puntoilunion/scrum>

De la mano de la metodología se implementará la arquitectura por capas ayudando a orientar los sprints y las tareas planteadas para cumplir los objetivos específicos, esta arquitectura permite la concentración en 3 componentes principales que se seleccionan gracias a un patrón de diseño seleccionado, siendo este el MVC (Modelo – Vista – Controlador) permitiendo el desarrollo desde las 3 perspectivas necesarias para realizar un proyecto escalable y fácil de mantener en el tiempo. Siguiendo una definición formal de la arquitectura por capas, tenemos la siguiente: “Se centra en una distribución jerárquica de las roles y responsabilidades proporcionando una separación efectiva de las preocupaciones (cada cual se encarga de lo que le corresponde)” (costanzo, 2025)

El proyecto se divide en 3 capítulos principales los cuales son la parte del Front-End, Back-End y el modelo encargado de la clasificación de las noticias, cada capítulo tiene sus respectivas actividades para lograr cumplir con el desarrollo de cada sección de la aplicación logrando así cumplir con éxito el objetivo principal del proyecto que es desarrollar una aplicación web con IA basada en NLP utilizando la metodología Scrum para detectar noticias falsas con alta precisión de manera ágil y efectiva.

## 2. OBJETIVOS

### Objetivo General

Desarrollar aplicación web con IA basada en NLP utilizando la metodología Scrum para detectar noticias falsas con alta precisión de manera ágil y efectiva.

### Objetivos Específicos

- Investigar herramientas y tecnologías necesarios para el desarrollo de la aplicación web con IA basada en NLP.
- Establecer las herramientas y tecnologías más precisas para en el desarrollo de la aplicación web con IA basada en NLP.
- Crear la aplicación web con IA basada en NLP.
- Validar la calidad para detectar Fake News de la aplicación web con IA basada en NLP.

## 3. RESULTADOS

### Objetivo específico I

Para dar solución a la investigación de herramientas y tecnologías necesarios para el desarrollo de la aplicación web con IA basada en NLP se optó por clasificar las tecnologías dependiendo de las capas del desarrollo, estas son en parte Front-End, Back-End y el modelo de clasificación de las noticias. Para comenzar con la investigación se empezó por las herramientas necesarias, como el IDE necesario para el proyecto.

**Herramientas:** Las herramientas consultadas en la investigación fueron las siguientes dependiendo de su orientación en el proyecto:

Controladores de versiones:

- GitHub
- GitLab
- Bitbucket

IDE's para el desarrollo:

- Visual Studio Code
- IntelliJ IDEA:
- PyCharm

Pruebas en la conexión con la API:

- Postman

Software para la gestión de las tareas del equipo por medio de la metodología Scrum:

- Trello
- Jira

Representación del modelo de usuario:

- Figma

**Tecnologías:** Las tecnologías consultadas en la investigación dependiendo de su capa de desarrollo fueron las siguientes:

Front-End:

- HTML5
- CSS3
- Tailwind CSS
- Bootstrap
- JavaScript
- React
- Angular
- Vue.js

Back-End:

- Node.js
- Spring Boot
- Django
- Flask
- Express.js
- MongoDB
- PostgreSQL
- MySQL

Modelo de Clasificación de las noticias:

- Python
- Scikit-learn
- spaCy
- NLTK:
- TensorFlow
- PyTorch
- Transformers (Hugging Face)
- Pandas
- NumPy

### Objetivo Especifico II

Teniendo en cuenta toda la información consultada con anticipación, se puede tomar una decisión respecto a las herramientas y

tecnologías más precisas para cada capa en el desarrollo de la aplicación web con IA basada en NLP.

### Selección de las Herramientas

**Figura 2 Comparación de Herramientas de versiones**

Herramienta	Ventajas	Desventajas
GitHub	<ul style="list-style-type: none"> <li>- Comunidad más grande y activa</li> <li>- Integración directa con Actions (CI/CD)</li> <li>- Compatible con Hugging Face, VS Code y otras herramientas</li> <li>- Amplia documentación y recursos</li> <li>- Interfaz intuitiva y moderna</li> </ul>	<ul style="list-style-type: none"> <li>- Algunas funciones avanzadas (como seguridad avanzada) requieren GitHub Pro</li> </ul>
GitLab	<ul style="list-style-type: none"> <li>- CI/CD integrado más completo en su versión gratuita</li> <li>- Permite repos privados ilimitados</li> <li>- Se puede instalar en servidores propios (self-hosted)</li> </ul>	<ul style="list-style-type: none"> <li>- Interfaz más compleja para nuevos usuarios</li> <li>- Menor comunidad para proyectos de IA/NLP</li> </ul>
Bitbucket	<ul style="list-style-type: none"> <li>- Integración nativa con Jira y Trello</li> <li>- Repositorios privados ilimitados en plan gratuito</li> <li>- Adecuado para proyectos pequeños y medianos</li> </ul>	<ul style="list-style-type: none"> <li>- Menor comunidad open-source</li> <li>- Pocas integraciones específicas para proyectos de IA</li> <li>- Interfaz menos intuitiva</li> </ul>

**Nota:** En la Figura 2 podemos encontrar las ventajas y desventajas de cada controlador de versiones investigado en el objetivo específico I. En base a la información de la Figura 2 podemos concluir que el controlador de versiones a utilizar en el proyecto será **GitHub**.

Para la selección del IDE se realizó una tabla de comparación del IDE con más compatibilidad con las tecnologías a utilizar en cada capa del desarrollo.

**Figura 3 Comparación de IDEs con tecnologías**

IDE	Tecnologías						
	HTML/CSS/JS	React / Angular	Django / Flask	Spring Boot	Python (IA/NLP)	TensorFlow / PyTorch	Integración con Git
Visual Studio Code	Excelente soporte	Muy buena con extensiones	Soporte completo	Requiere configuración adicional	Completo	Con plugin	Nativo y fluido
IntelliJ IDEA	Compatible con plugin	Compatible con plugin	Mejor con plugin adicionales	Soporte nativo (ideal para Java)	Básico (no es su fuerte)	Requiere configuración	Integración nativa
PyCharm	Limitado (más básico)	Menor compatibilidad	Excelente (ideal para Django/Flask)	No recomendado para Java	Ideal para Python	Soporte completo	Muy buen soporte

**Nota:** En la Figura 3 encontramos la comparación de los IDEs para trabajar con todas las tecnologías consultadas en el desarrollo del Objetivo Específico I.

En base a la información plasmada en la Figura 2 se va a utilizar **Visual Studio Code (VSCode)** para el desarrollo de cada una de las capas del proyecto, esto por su casi completa compatibilidad con las tecnologías investigadas. La herramienta de **Postman** será utilizada para realizar diversas pruebas de respuesta del servidor una vez conectado con la parte del

Front-End, esto para comprobar la efectividad de los métodos y se logren cumplir los resultados esperados.

Para seleccionar la herramienta para llevar el control de las actividades programadas por la metodología SCRUM se realizó una tabla comparativa entre los dos softwares investigados.

**Figura 4 Comparación entre los Software de manejo de SCRUM**

Característica	Trello	Jira
Facilidad de uso	Muy fácil e intuitivo	Más complejo, curva de aprendizaje más alta
Soporte para Scrum	Limitado (requiere plantillas o Power-Ups)	Soporte nativo (backlogs, sprints, epics, etc.)
Gestión de tareas	Listas y tarjetas personalizables	Completa y estructurada con flujos de trabajo
Seguimiento de progreso	Básico (tableros Kanban simples)	Avanzado (burndown charts, velocity, etc.)
Integraciones	Amplias (Slack, GitHub, Google Drive, etc.)	Excelente (con Bitbucket, GitHub, Confluence, etc.)
Escalabilidad para equipos grandes	Limitada	Alta, ideal para equipos de desarrollo
Coste	Gratuito con funciones básicas	Gratuito hasta 10 usuarios, luego es de pago
Automatización	Mediante Power-Ups (Butler)	Flujo de trabajo automatizado más robusto
Visualización del proyecto	Intuitiva (tableros tipo Kanban)	Detallada, aunque más técnica
Recomendado para	Equipos pequeños o proyectos simples	Equipos de desarrollo ágil, proyectos complejos

**Nota:** En la figura 4 se presentan las características principales de cada software de control de la metodología SCRUM.

En base a la información proporcionada en la Figura 4 para el manejo de los tiempos del proyecto se seleccionó **Trello** siendo este el software que más se ajusta a las características del proyecto.

Para la representación de las interfaces de usuario se eligió la herramienta de **Figma**, esto por su fácil manejo y posibilidad de mostrar el desplazamiento entre ventanas del desarrollo, logrando darles a los usuarios interesados una perspectiva de cómo será el resultado final.

### Selección de las Tecnologías

Como parte del desarrollo del segundo objetivo específico del proyecto se presenta la información para decidir cuál tecnología es más apropiada para cada capa del desarrollo.

### Tecnología para el Front-End

**Figura 5 Fortalezas de cada tecnología de desarrollo**

Tecnología	Fortalezas principales
HTML/CSS/JavaScript	<ul style="list-style-type: none"> <li>- Base fundamental del desarrollo web</li> <li>- Compatible con todos los navegadores</li> <li>- Fácil de aprender e implementar</li> <li>- Ideal para prototipos rápidos y proyectos simples</li> </ul>
React	<ul style="list-style-type: none"> <li>- Reutilización de componentes</li> <li>- Alto rendimiento con Virtual DOM</li> <li>- Gran ecosistema (Next.js, Redux, etc.)</li> <li>- Flexible e integrable con backends y APIs</li> <li>- Muy popular en aplicaciones modernas</li> </ul>
Angular	<ul style="list-style-type: none"> <li>- Arquitectura completa para aplicaciones grandes</li> <li>- Inyección de dependencias y TypeScript nativos</li> <li>- Vinculación bidireccional de datos</li> <li>- Ideal para aplicaciones empresariales estructuradas</li> </ul>
Vue.js	<ul style="list-style-type: none"> <li>- Curva de aprendizaje suave</li> <li>- Sintaxis clara y fácil de usar</li> <li>- Requiere menos configuración inicial que Angular</li> <li>- Componente reactivo similar a React, pero más sencillo</li> <li>- Ideal para proyectos escalables con bajo nivel de complejidad inicial</li> </ul>

**Nota:** En la figura 5 se nos presentan las fortalezas de cada tecnología para el desarrollo del Front-End del proyecto.

En base la información suministrada en la figura 5 podemos concluir lo siguiente:

- **HTML/CSS/JS:** Base necesaria y útil en cualquier stack, pero limitado sin un framework.
- **React:** Recomendado para proyectos modernos, modulares y dinámicos como el proyecto.
- **Angular:** Más adecuado para equipos grandes y aplicaciones empresariales estructuradas.
- **Vue.js:** Perfecto si buscas una combinación de simplicidad y escalabilidad.

De acuerdo al análisis realizado, se optó por utilizar **React** para el desarrollo de la parte Front-End del proyecto, esto por su compatibilidad con el objetivo del proyecto.

### Tecnología para el Back-End

**Figura 6 Comparación de las fortalezas de cada tecnología para el Back-End**

Tecnología	Fortalezas principales
Django	<ul style="list-style-type: none"> <li>- Framework robusto con ORM, autenticación y <del>admisión</del> integrado</li> <li>- Excelente seguridad por defecto</li> <li>- Ideal para proyectos con IA en Python</li> </ul>
Flask	<ul style="list-style-type: none"> <li>- <del>Microframework</del> ligero y flexible</li> <li>- Ideal para APIs rápidas</li> <li>- Muy usado en proyectos con IA/NLP</li> </ul>
Spring Boot	<ul style="list-style-type: none"> <li>- Escalable y estructurado</li> <li>- Ideal para entornos empresariales</li> <li>- Soporte nativo para servicios REST</li> </ul>
Node.js	<ul style="list-style-type: none"> <li>- Entorno de ejecución rápido y eficiente basado en JavaScript</li> <li>- Ideal para manejo de múltiples conexiones y eventos</li> <li>- Compatible con todo el ecosistema JS</li> </ul>
Express.js	<ul style="list-style-type: none"> <li>- Framework minimalista para Node.js</li> <li>- Permite crear APIs REST de forma rápida y flexible</li> <li>- Gran comunidad y soporte</li> </ul>

**Nota:** En la figura 6 se nos presenta las fortalezas de cada tecnología investigada para el Back-End.

En conformidad con la información presentada en la figura 6 podemos concluir lo siguiente:

- **Node.js + Express.js** ofrecen una integración fluida y nativa gracias al uso de JavaScript.
- **Django y Flask** son potentes para proyectos que integran IA con Python.
- **Spring Boot** es recomendable para aplicaciones robustas con Java.

De acuerdo a la información recolectada la tecnología seleccionada para el desarrollo Back-End de la aplicación va a ser **Flask** por su flexibilidad y rapidez a la hora de los desarrollos.

### Tecnología para el modelo de clasificación de Fake News

**Figura 7 Comparación de Modelos y librerías de Clasificación**



Herramienta / Librería	Propósito principal	Fortalezas clave
Python	Lenguaje de programación base	- Sintaxis simple y legible - Gran ecosistema de librerías de IA/NLP
Scikit-learn	Machine learning tradicional	- Ideal para modelos clásicos (Naive Bayes, SVM) - Fácil integración con Pandas
spaCy	Procesamiento de lenguaje natural	- Rápido y eficiente para tareas NLP básicas - Buen soporte para pipelines
NLTK	Procesamiento lingüístico académico	- Extensa base de datos lingüísticos - Ideal para prototipos y aprendizaje
TensorFlow	Framework de deep learning	- Alta eficiencia en redes neuronales - Gran comunidad y herramientas de producción
PyTorch	Framework de deep learning	- Más flexible y dinámico que TensorFlow - Preferido en investigación
Transformers (Hugging Face)	Modelos preentrenados para NLP (BERT, RoBERTa, etc.)	- Alta precisión en clasificación de texto - Compatible con PyTorch y TensorFlow
Pandas	Manipulación de datos tabulares	- Ideal para preprocesar datasets - Integra con Scikit-learn y NumPy
NumPy	Cálculo numérico y matrices	- Base matemática de otras librerías - Rápido y versátil

**Nota:** En la figura 7 podemos encontrar la comparación de todos los modelos y librerías de clasificación compatibles con el proyecto.

En base a la información recopilada, se consideró la creación de un modelo personal que evalúe ciertos parámetros para dar la clasificación de las noticias, por lo tanto, el modelo será realizado bajo el lenguaje en general de **Python** con las librerías que se vayan necesitando a lo largo de los cambios en la idea principal del modelo.

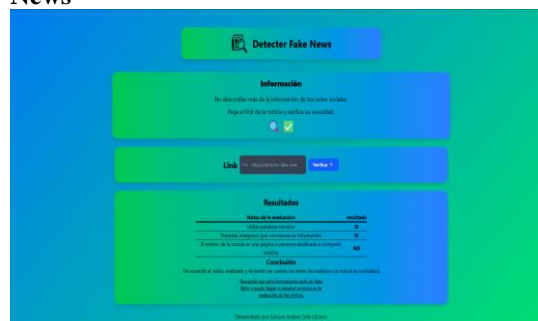
### Objetivo Especifico III

Para el cumplimiento del objetivo específico número III el cual consiste en crear la aplicación web con IA basada en NLP, se dividió este objetivo en 3 fases para mejor tiempo de desarrollo y ejecución en el proyecto. Las fases selectas fueron las ya mencionadas anteriormente en el documento (**Front-End, Back-End y Modelo de clasificación**). Para el inicio del proyecto se comenzó por la parte Front-End para dar a los usuarios un adelanto de la orientación de los elementos y dar a observar las funcionalidades que se podrán utilizar cuando el desarrollo este culminado.

### Front-End

Para el desarrollo Front-End se eligió la tecnología de React, bajo esta se realizó la interfaz gráfica que el usuario va a utilizar.

**Figura 8 Interfaz Gráfica de Detector Fake News**



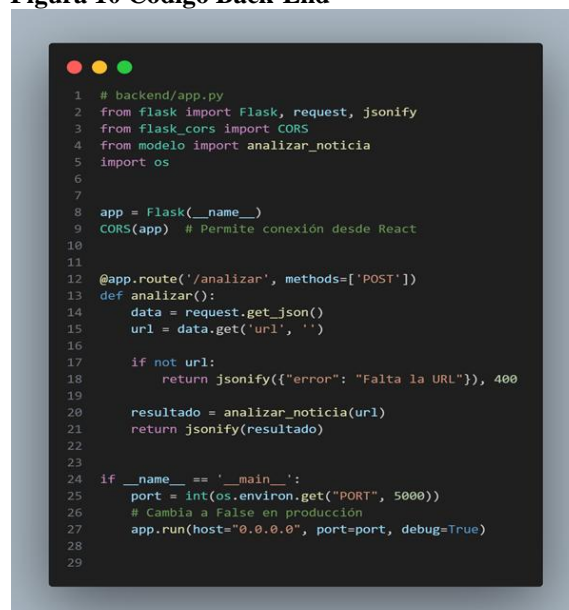
**Ilustración 9 Pagina inicial**



### Back-End

Para la parte del desarrollo Back-End, siendo esta parte la segunda fase del desarrollo de la aplicación web, tenemos una conexión muy simple, esto debido a que no se necesita un sistema de base de datos en la aplicación, solo se necesita establecer la conexión con el modelo y ya. Para dar cumplimiento a esto se utiliza una función llamada analizar con el método POST para realizar la debida conexión, en la figura 10 podemos observar cómo está compuesto el Back-End de la aplicación.

**Figura 10 Código Back-End**



En la figura 10 también podemos encontrar la configuración de los puertos y host para que la

aplicación Render que es donde se encuentra desplegado la parte Back-End y el modelo funcionen.

### Modelo de Clasificación

En el desarrollo del modelo de clasificación siendo esta la última fase del desarrollo de la aplicación web, tenemos un modelo creado en base a unos parámetros de evaluación basados únicamente en el texto referente a la noticia (este modelo no interpreta las imágenes o videos adjuntos a las noticias). Por medio de análisis de sentimientos y unas Apis de palabras claves se logra identificar la intención de los textos y dar un veredicto de la veracidad de las noticias. En las ilustraciones 11,12 y 13 podemos observar la composición del código y su forma de trabajar.

**Figura 11 Composición principal del Modelo de Clasificación**

```
1 # backend/model.py
2 import requests
3 from bs4 import BeautifulSoup
4 import re
5 import datetime
6 # from transformers import pipeline
7 import os
8
9 # classifier = pipeline("sentiment-analysis",
10 #                       model="distilbert-base-uncased-finetuned-sts-z-en")
11
12 # URLS de listas de palabras positivas y negativas en inglés y español
13 URL_POSITIVAS_EN = "https://ptrckpry.com/course/wordlist/positive.txt"
14 URL_NEGATIVAS_EN = "https://ptrckpry.com/course/wordlist/negative.txt"
15 URL_POSITIVAS_ES = "https://raw.githubusercontent.com/clea/lemarios/master/palabras-positivas.txt"
16 URL_NEGATIVAS_ES = "https://raw.githubusercontent.com/clea/lemarios/master/palabras-negativas.txt"
17
18
19 def cargar_lista_palabras(url, archivo_cache):
20     if os.path.exists(archivo_cache):
21         with open(archivo_cache, "r", encoding="utf-8") as f:
22             return [line.strip() for line in f if line.strip()]
23     else:
24         resp = requests.get(url)
25         palabras = [line.strip() for line in resp.text.splitlines() if line.strip()]
26         with open(archivo_cache, "w", encoding="utf-8") as f:
27             f.write("\n".join(palabras))
28         return palabras
29
30
31 palabras_positivas_en = cargar_lista_palabras(
32     URL_POSITIVAS_EN, "positivas_en.txt")
33 palabras_negativas_en = cargar_lista_palabras(
34     URL_NEGATIVAS_EN, "negativas_en.txt")
35 palabras_positivas_es = cargar_lista_palabras(
36     URL_POSITIVAS_ES, "positivas_es.txt")
37 palabras_negativas_es = cargar_lista_palabras(
38     URL_NEGATIVAS_ES, "negativas_es.txt")
```

**Figura 12 Composición secundaria del Modelo de Clasificación**

```
1 alarm_words = [
2     "increíble!", "no lo vas a creer", "impactante", "última hora", "alerta", "escándalo"
3 ]
4
5
6 def detectar_idioma(texto):
7     # Detección simple: si hay muchas palabras con tildes, asume español
8     if re.search("[áéíóú]", texto.lower()):
9         return "es"
10    # Si hay palabras comunes en inglés
11    if re.search(r"[b(t|h|e|l|a|n|d|s|i|s|a|r|e|f|o|t|o|i|n|t|h|a|t|i|s|f|o|r|j|o|n|w|t|h|)b]", texto.lower()):
12        return "en"
13    # Por defecto, español
14    return "es"
15
16
17 def analizar_sentimiento_simple(texto, idioma):
18     texto = texto.lower()
19     if idioma == "es":
20         negativos = sum(palabra in texto for palabra in palabras_negativas_en)
21         positivos = sum(palabra in texto for palabra in palabras_positivas_es)
22     else:
23         negativos = sum(palabra in texto for palabra in palabras_negativas_en)
24         positivos = sum(palabra in texto for palabra in palabras_positivas_en)
25     if negativos > positivos and negativos > 1:
26         return "NEGATIVE"
27     elif positivos > negativos and positivos > 1:
28         return "POSITIVE"
29     else:
30         return "NEUTRAL"
31
32
33 def analizar_noticia(url):
34     resultado = {
35         "fuente_confiable": False,
36         "buena_redaccion": False,
37         "fecha_actual": False,
38         "tiene_fuentes": False,
39         "sin_sensacionalismo": True,
40         "veredicto": "Análisis incompleto"
41     }
```

**Figura 13 Composición final del Modelo de Clasificación**

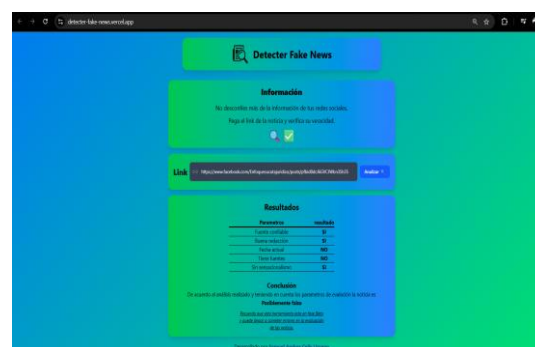
```
1 try:
2     response = requests.get(url, timeout=10)
3     soup = BeautifulSoup(response.text, 'html.parser')
4     texto = soup.get_text(separators=" ", strip=True)
5
6     idioma = detectar_idioma(texto)
7
8     dominios_confiables = ["bbc.com", "cnn.com",
9                             "reuters.com", "elpais.com", "chequeado.com"]
10    resultado["fuente_confiable"] = any(
11        dominio in url for dominio in dominios_confiables)
12
13    errores = re.findall(
14        r'\b[kzg](5,){1}["a-zA-Z0-9A-Z0-9@1000000-9,;:;?!(\|)\"]+', texto)
15    resultado["buena_redaccion"] = len(errores) < 5
16
17    fechas = re.findall(r'\d{1,2}/\d{1,2}/\d{2,4}', texto)
18    if fechas:
19        try:
20            fecha_detectada = datetime.datetime.strptime(
21                fechas[0], '%d/%m/%Y')
22            resultado["fecha_actual"] = abs(
23                (datetime.datetime.now() - fecha_detectada).days) < 90
24        except:
25            pass
26
27    resultado["tiene_fuentes"] = bool(
28        re.search(r"http[s]?://[a-z]+", texto)) or "según" in texto.lower()
29
30    resultado["sin_sensacionalismo"] = not any(
31        word in texto.lower() for word in alarm_words)
32
33    emocion = analizar_sentimiento_simple(texto[:500], idioma)
34    if emocion == "NEGATIVE":
35        resultado["veredicto"] = "Falsa noticia"
```

### Objetivo Especifico IV

Para dar cumplimiento al objetivo específico IV el cual consistía en validar la calidad para detectar Fake News de la aplicación web con IA basada en NLP, se utilizaron diferentes noticias de las páginas de noticias de Facebook. Páginas que habitualmente las personas consultan y consumen a diario.

Los resultados de estas pruebas los podemos observar en las ilustraciones 14,15,16 y 17.

### Ilustración 14 Prueba 1



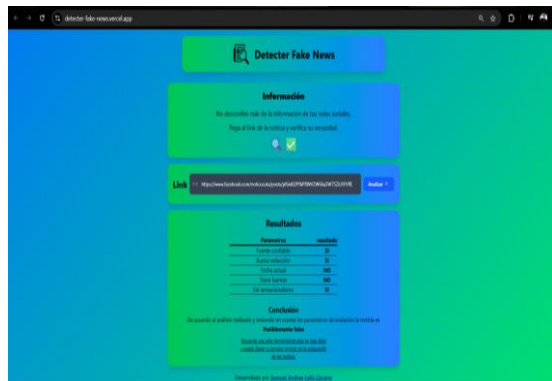
**Nota:** Podemos observar que la primera prueba se realizó con un link de la página de enfoque Cúcuta jurídico, página dedicada a subir noticias de la ciudad de Cúcuta.

### Figura 15 Prueba 2



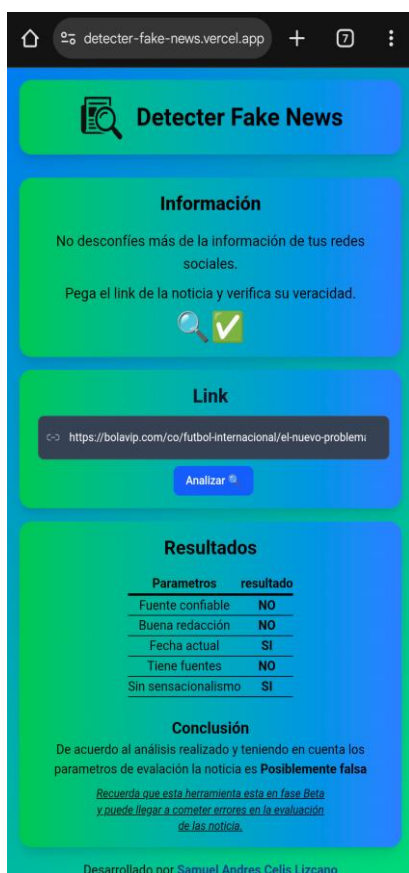
**Nota:** En esta prueba podemos observar que el link probado es de una página llamada bolavip dedicada a subir noticias de futbol colombiano y todo lo relacionado a este.

**Figura 16 Prueba 3**



**Nota:** En esta prueba podemos observar como el link utilizado es de una página llamada noticucuta, que se dedica a subir lo último en noticias de la ciudad de Cúcuta.

**Figura 17 Prueba 4**



**Nota:** En esta última prueba se volvió a utilizar el link de una noticia de bolavip, pero desde un dispositivo móvil (celular).

## 4 CONCLUSIONES

- Durante la investigación de las herramientas y tecnologías necesarias para el desarrollo pudimos observar la gran variedad de opciones que existen actualmente en el mundo para llevar a cabo el desarrollo de un proyecto.
- En el momento de la selección de las herramientas y tecnologías necesarias para cada capa del desarrollo se logró comprender para que tipos de desarrollo son mejor cada una de las tecnologías existen, permitiendo así elegir la más apropiada con el objetivo general del proyecto.
- Al momento de construir aplicaciones web, el utilizar la arquitectura por capaz el proyecto se simplifica de una forma gigantesca, permitiendo trabajar de manera simultánea en ambas partes del proyecto sin ver afectaciones en el desarrollo de alguna parte del proyecto.
- El desplegar los componentes de la aplicación en diferentes plataformas ayuda a mantener en funcionamiento los diferentes servicios de la aplicación.
- El realizar las pruebas de validación expuso los pocos dominios aceptados como validos en la lógica del modelo, permitiendo sugerencias como agregar las fuentes por medio de alguna Api que contenga los dominios de páginas oficiales.

## 5. RECOMENDACIONES

De acuerdo a las pruebas realizadas con el modelo creado, se deben agregar librerías para permitir el análisis de fotografías y/o videos adjuntos a la noticia, para permitir tener nuevos criterios de evaluación para ayudar a realizar un veredicto un poco mas exacto de las noticias.

## REFERENCIAS

- Chiang Chang Yu a, \*. I. (2023). A Multi-tier Model and Filtering Approach to Detect Fake News Using. *INTERNATIONAL JOURNAL*, 643-651.1
- Agnieszka Jaff, K. U.-O.-M. (2020). Fake News and Aggregated Credibility: Conceptualizing a Co-Creative Medium for Evaluation of Sources Online. *International Journal of Ambient Computing and Intelligence*, 93-117.
- costanzo, m. (11 de 5 de 2025). Platzi. Obtenido de Platzi: <https://platzi.com/tutoriales/1248-pro-arquitectura/5439-patron-arquitectonico-de-capas-layers/#:~:text=La%20arquitectura%20en%20capas%20es,de%20lo%20que%20le%20cooresponde>.



HugoQueiroz Abonizio, J. I. (2020). *Language-Independent Fake News Detection: English, Portuguese, and Spanish Mutual Features*. *Future internet*, 12-87.

Amirhosein Bodaghi, J. O. (2021). *Thefakenewsgraphanalyzer:Anopen-sourcesoftwareforcharacterizing spreaders in large diffusion graphs*. *Software Impacts*.

Issa Qiqieh, O. A.-Z. (2025). *Anintelligent cyber threat detection: A swarm-optimized machine learning approach*. *AlexandriaEngineeringJournal*, 553-563.

Ana Julia Dal Forno, G. P. (2025). *Fake news detection algorithms – A systematic literature review*. *Data & Knowledge Engineering*, 158.

Pérez, C. R. (2019). *No diga fake news, di desinformación: una revisión sobre el fenomeno de las noticias falsas y sus implicaciones*. *COMUNICACIÓN*, 65 - 74.

Ariel José Torres Pineda, A. V. (2021). *Veracidad y consumo de fake news entre estudiantes de una universidad pública de Cartagena de Indias, Colombia*. *DOXA DIGITAL*, 153-174.

Cuadros Bustamante, D. C. (2021). *Análisis de fake news del periódico El Espectador y la Revista Semana sobre el covid-19, y el impacto que generan en los estudiantes de últimos semestres de Comunicación Social de la Universidad de Pamplona, sede Pamplona y la población sanalbertense*. *Universidad de Pamplona - Facultad de Artes y Humanidades*.

López-Borrull, A. L.-B.-G. (2018). *Fake news, ¿amenaza u oportunidad para los profesionales de la información y la documentación?* *e-repository*.