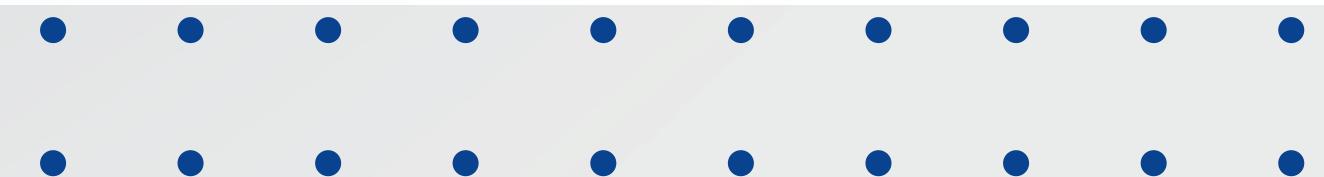


TF-IDFENSE

Alan Patricio González Bernal - A01067546
Alan Rodrigo Castillo Sánchez - A01708668

Overview

- 1.** Introducción
- 2.** Objetivo
- 3.** Selección de modelo
- 4.** Selección de variables
- 5.** Resultados
- 6.** Ventajas



Introducción

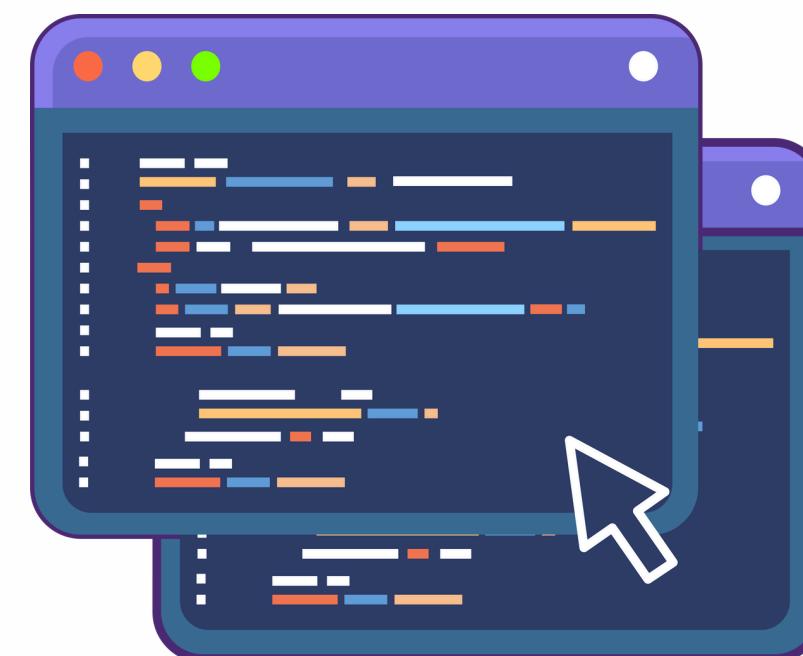
El plagio de código fuente es un problema creciente en entornos educativos, agravado por lo fácil que es modificar superficialmente un código para intentar ocultarlo lo más posible. Por desgracia, esto se ha visto recientemente impulsado aún más por el uso y crecimiento de la IA. Por ello es necesario desarrollar herramientas avanzadas que detecten no solo copias literales, sino también plagios disfrazados mediante cambios mínimos.





Objetivo

Nuestro objetivo fue desarrollar un programa capaz de identificar plagio entre códigos fuente escritos en Java mediante técnicas cuantitativas y análisis formal de lenguajes de programación, priorizando la simplicidad y bajo costo computacional.



Selección de modelo

TF-IDF & Tokenización



- Respaldado por documentos científicos
- Eficiente computacionalmente hablando
- Acompañado de las técnicas correctas, es poderoso
- En general, de las técnicas más utilizadas

ConPlag Dataset

1. TF-IDF (Term Frequency-Inverse Document Frequency)

- Técnica ampliamente utilizada para identificar la importancia de términos en un conjunto de documentos.
- Permite detectar qué tokens o estructuras son distintivas de un archivo
- Robusto ante cambios superficiales
- Eficiente computacionalmente

$$TF(t, d) = \frac{\text{number of times } t \text{ appears in } d}{\text{total number of terms in } d}$$

$$IDF(t) = \log \frac{N}{1 + df}$$

$$TF - IDF(t, d) = TF(t, d) * IDF(t)$$



2. Tokenización

ANTLR4

(ANother Tool for Language Recognition)

- Nos permite tokenizar códigos, para poder extraer la intención del mismo sin depender de las palabras específicas como variables o funciones.
- Definir qué tokens ignorar para reducir ruido, como';', '(', ')', '[', ']', '{', '}', etc.

Código fuente

```
if (x > 0)  
  
for x in y:  
    ...
```

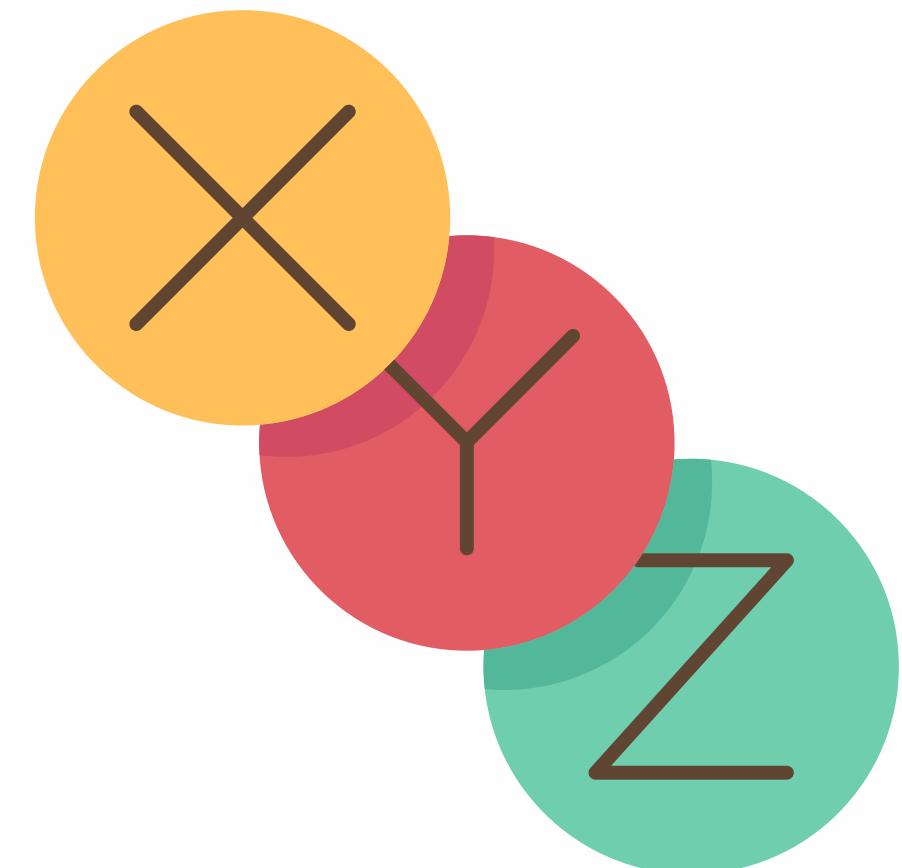
Tokenizado

```
IF_BLOCK  
  
FOR_BLOCK  
...
```

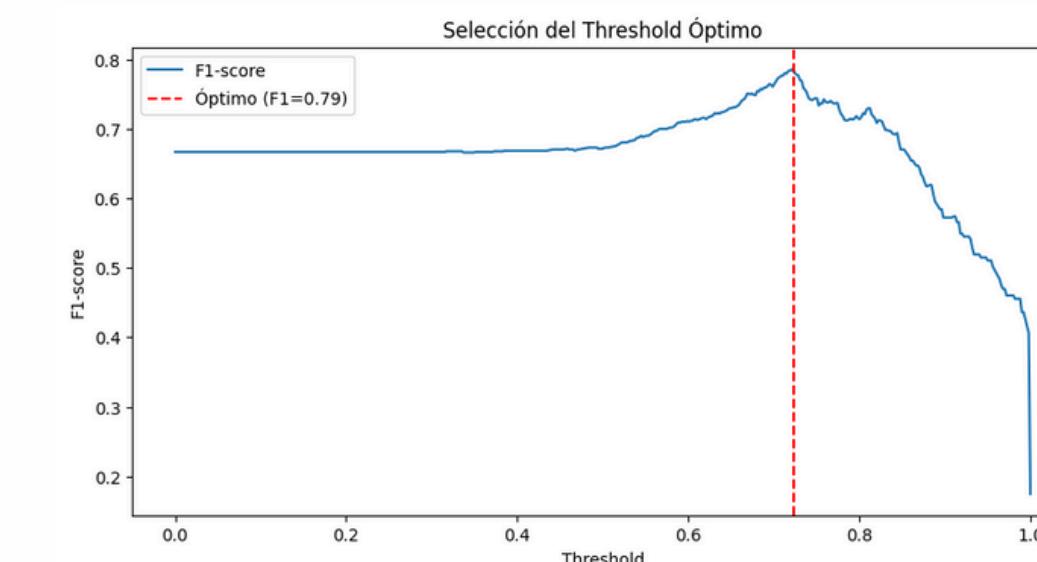
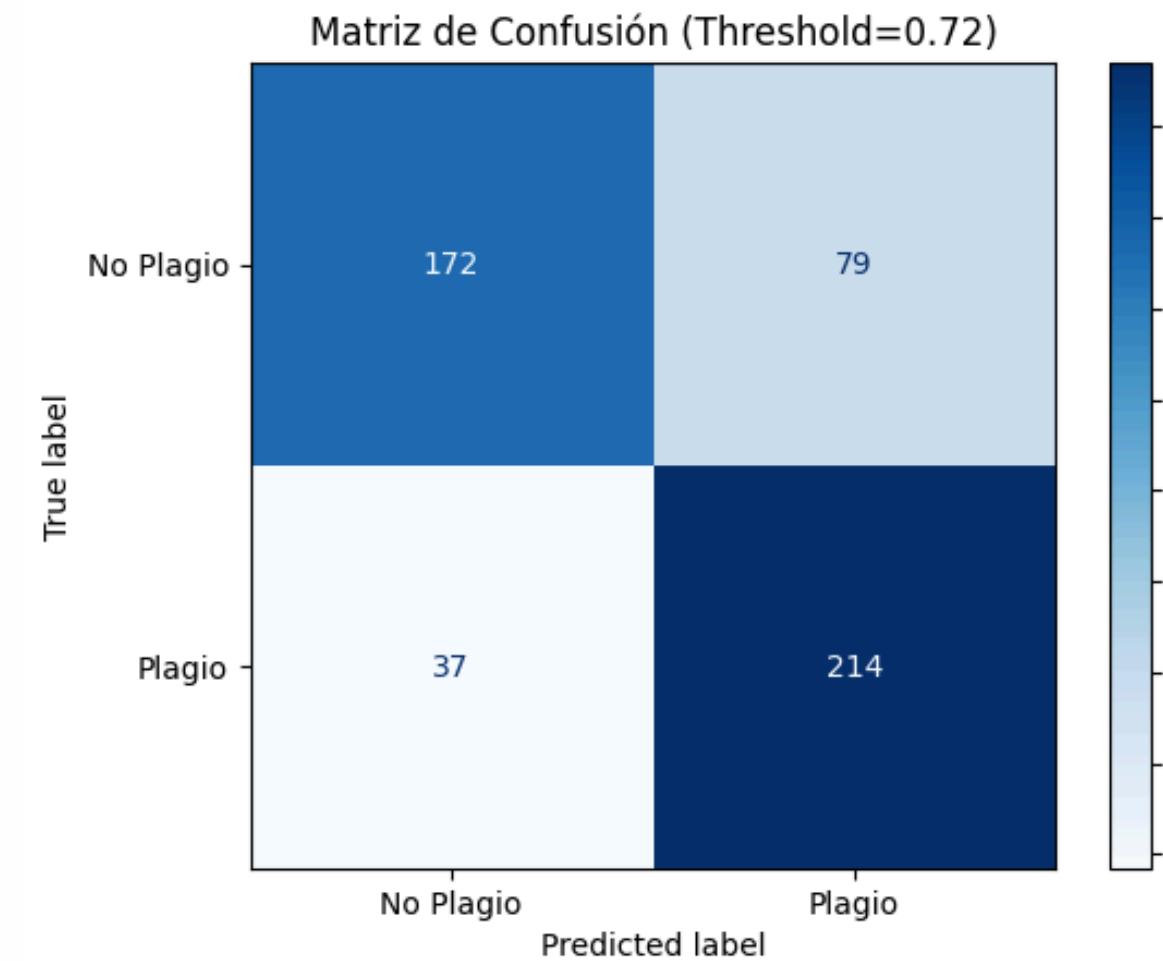
Selección de variables

- TF-IDF
 - Es buen método para detección de plagio
- Tokenización
 - Ampliamente utilizado en detección de plagio, ANTLR4 es el usado en nuestros papers
 - Ayuda a detectar plagios no literales
- Threshold dinámico
 - Se calcula en base al F1, asegura el mejor rendimiento para el modelo
- ConPlag Dataset
 - Ampliamente usado para esta problemática, fácil de importar y usar. Balanceado manualmente

- Factores de resultados
 - Accuracy
 - Precision
 - Recall
 - F1



Resultados



Accuracy	Precision	Recall	F1-Score
0.77	0.73	0.85	0.79

Ventajas

- Detección de plagio ante cambios superficiales y de estructura
- Simplicidad
- Precisión aceptable
- Reproducibilidad
- Bajo costo computacional

Criterio	WASTK, Fu et al. (2017)	TF-IDF-INSPIRED Cross Language Karnalim (2020)	TF-IDFense
Tokenización avanzada	✗	✗ Parcial	✓
Estructura sintáctica	✓ Alto: AST + kernel	✗	✓ Parcial
F1-Score	✓ Baja ~0.65	✗	✓ Alta ~0.79
Detección no literal	✗ Parcial	✓	✓
Simplicidad/portabilidad	✗	✓	✓

GRACIAS

Referencias

- O. Karnalim, "TF-IDF-inspired detection for cross-language source code plagiarism and collusion," *Computer Science*, vol. 21, no. 1, pp. 113–136, 2020. doi: 10.7494/csci.2020.21.1.3389
- E. Slobodkin y A. Sadovnikov, "ConPlag: a Dataset of Programming Contest Plagiarism in Java," Zenodo, Nov. 10, 2022. doi: 10.5281/zenodo.7332790
- D. Fu, Y. Xu, H. Yu, y B. Yang, "WASTK: A Weighted Abstract Syntax Tree Kernel Method for Source Code Plagiarism Detection," *Scientific Programming*, vol. 2017, Art. no. 7809047, 2017. doi: 10.1155/2017/7809047
- Oscar Karnalim, Sulistiani, H. Toba, y M. Joy, "Source Code Plagiarism Detection in Academia with Information Retrieval: Dataset and the Observation," *International Journal of Engineering Education*, vol. 35, no. 4, pp. 1062–1073, 2019
- Z. C. Lipton, C. Elkan y B. Narayanaswamy, "Optimal Thresholding of Classifiers to Maximize F1 Measure," en *Machine Learning and Knowledge Discovery in Databases*, T. Calders, F. Esposito, E. Hüllermeier y R. Meo, Eds., ECML PKDD 2014, *Lecture Notes in Computer Science*, vol. 8725, Springer, Berlín, Heidelberg, 2014, pp. 225–239
- R. S. Mehser and H. D. Joshi, "Detection of Source Code Plagiarism Utilizing an Approach Based on Machine Learning," *International Journal of Computing*, vol. 23, no. 1, pp. 78–84, 2024, doi: 10.41859/jc.211.34.98.

