



Fig. 1: Matriz de programación dinámica completada

METODOLOGÍA

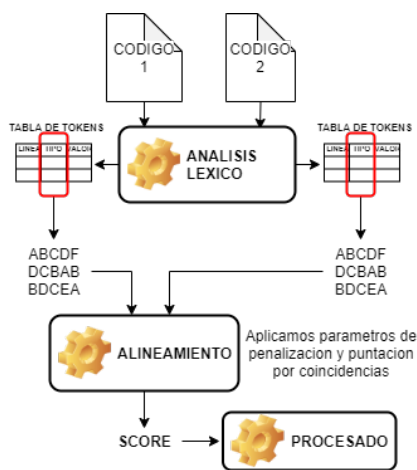


Fig. 2: Metodología

1. Lo primero que haremos sera generar la tabla de tokens haciendo el análisis léxico de cada uno de nuestros códigos que evaluaremos.

```
C:\Windows\system32\cmd.exe
```

'Linea 1	Tipo	MEJORQUE	Valor <	Posicion 9
'Linea 1	Tipo	IDENTIFICADOR	Valor bits	Posicion 10
'Linea 1	Tipo	DIV	Valor /	Posicion 14
'Linea 1	Tipo	STD	Valor std	Posicion 15
'Linea 1	Tipo	IDENTIFICADOR	Valor c	Posicion 18
'Linea 1	Tipo	PLUSPLUS	Valor ++	Posicion 19
'Linea 1	Tipo	IDENTIFICADOR	Valor h	Posicion 22
'Linea 1	Tipo	MAYORQUE	Valor >	Posicion 23
'Linea 2	Tipo	USING	Valor using	Posicion 25
'Linea 2	Tipo	NAMESPACE	Valor namespace	Posicion 31
'Linea 2	Tipo	STD	Valor std	Posicion 41
'Linea 2	Tipo	PUNTOCOMA	Valor ;	Posicion 44
'Linea 3	Tipo	VOID	Valor void	Posicion 46
'Linea 3	Tipo	IDENTIFICADOR	Valor swap	Posicion 51
'Linea 3	Tipo	PARIZQ	Valor (Posicion 55
'Linea 3	Tipo	IDENTIFICADOR	Valor int	Posicion 56
'Linea 3	Tipo	MULT	Valor *	Posicion 59
'Linea 3	Tipo	IDENTIFICADOR	Valor a	Posicion 61
'Linea 3	Tipo	COMA	Valor ,	Posicion 62
'Linea 3	Tipo	IDENTIFICADOR	Valor int	Posicion 64

Fig. 3: Tabla de tokens

- De nuestra tabla de tokens extraemos los tipos y generamos 2 vectores de tipos.



Fig. 4: Vector de tipos

3. Antes de realizar el alineamiento renombramos nuestros elementos de nuestro vector de tipos para todas las coincidencias.

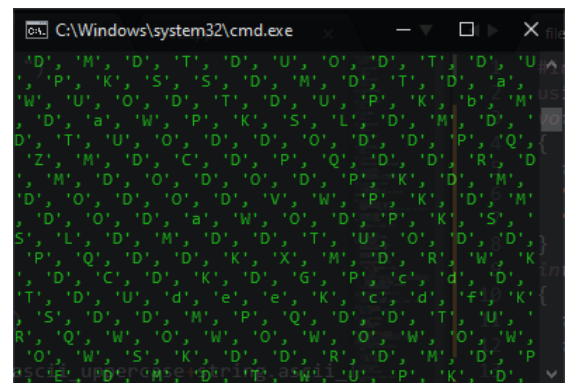


Fig. 5: Cambio de variable coincidencias

4. Luego realizamos el alineamiento mediante programación dinámica y puntuamos las coincidencias y asu vez penalizamos los gaps y aperturas
5. Terminado este proceso, tendremos un *score* el cual representa el numero de coincidencias entre uno y otro lo expresamos en términos de porcentaje para saber cuanta coincidencia hay entre los códigos.

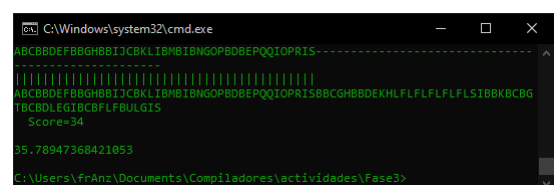


Fig. 6: Similitud

RESULTADOS

Para la parte de resultados se tomaron en cuenta varios escenarios de prueba y se muestran los resultado de similitud para cada situación ver la siguiente tabla.

ESCENARIO DE COPIA	SIMILITUD
Total	100 %
Total con modificación	100 %
Una parte de código	52 %
Diferentes códigos	30 %

CONCLUSIONES

- La detección de plagio a nivel de código no siempre se puede realizar a nivel textual ya que es posible que la persona que realiza el plagio cambie los nombres de las variables, funciones, etc.
- Como se ha visto en este trabajo se puede realizar mediante programación dinámica a partir de los vectores de tipos dándoles un tratamiento de secuencias.
- Sería posible mejorar este método realizando las comparaciones por ventanas es decir segmentos de códigos.
- Para trabajos futuros se considera trabajar métodos añadidos para mejorar la precisión.

VIDEO Y GITHUB

- Video Demostrativo Youtube
- Repositorio Github

REFERENCIAS

- [1] M. F. Abad-García, “El plagio y las revistas depredadoras como amenaza a la integridad científica,” *Anales de Pediatría*, vol. 90, no. 1, pp. 57.e1–57.e8, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1695403318305265>