

## Informe de Laboratorio 06

### Tema: Arbol AVL

#### INFORMACIÓN BÁSICA

<b>ASIGNATURA:</b>	Estructura de Datos y Algoritmos				
<b>TÍTULO DEL TRABAJO:</b>	Trabajo Final: Detector de plagio				
<b>NÚMERO DE TRABAJO:</b>	07	<b>AÑO LECTIVO:</b>	2023-A	<b>NRO. SEMESTRE:</b>	III
<b>FECHA DE PRESENTACIÓN:</b>	08/08/23	<b>HORA DE PRESENTACIÓN:</b>	23:59		
<b>INTEGRANTE (s)</b>				<b>NOTA (0-20)</b>	
Hidalgo Chinchay, Paulo Andre Betanzos Rosas, Taylor Anthony Villafuerte Ccapira Frank Alexis					
<b>DOCENTE(s):</b> Mg. Edith Giovanna Cano Mamani					

Tabla 1: Mi tabla extendida

#### INTRODUCCIÓN

En el campo de la investigación el robo de propiedad intelectual es un problema muy grande, y la preocupación ante este problema crece aún más debido a la facilidad que nos ofrece la tecnología para acceder a la información.

En el presente trabajo presentaremos la implementación de un detector de plagio textual, el cual buscará aspectos similares entre un párrafo base y varios documentos; y a partir de esto dará un veredicto.

#### MARCO CONCEPTUAL

##### TRIE

Un trie es una estructura de datos de tipo árbol que permite la recuperación de información. La información almacenada en un trie es un conjunto de claves, donde una clave es una secuencia de símbolos pertenecientes a un alfabeto. Las claves son almacenadas en las hojas del árbol y los nodos internos son pasarelas para guiar la búsqueda. El árbol se estructura de forma que cada letra de la clave se sitúa en un nodo de forma que los hijos de un nodo representan las distintas posibilidades de símbolos diferentes que pueden continuar al símbolo representado por el nodo padre. Por tanto la búsqueda en un trie se hace de forma similar a como se hacen las búsquedas en un diccionario[1]

##### PLAGIO

Según la RAE, el plagio es la acción de plagiar, el cual consiste copiar obras ajenas y darlas como propias[2,3]

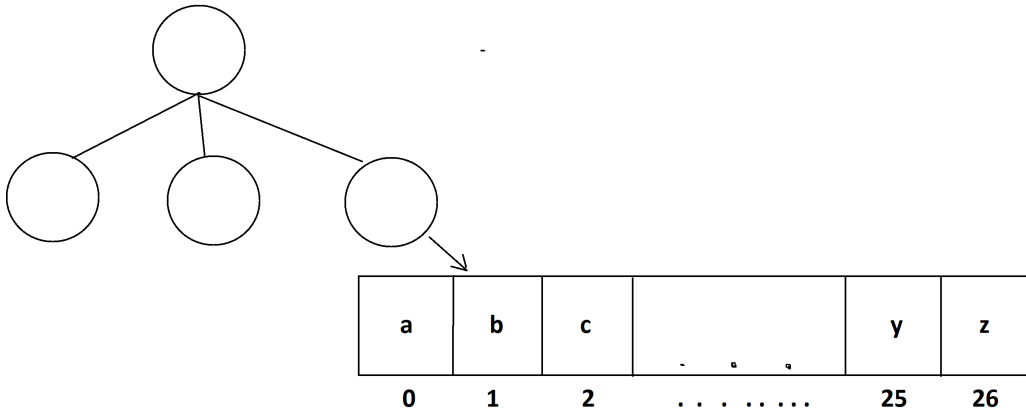
##### DETECTOR DE PLAGIO

Es el proceso de detectar las similitudes de contenido, el cual se realizaba mediante la detección humana. Gracias a la llegada de la tecnología, este proceso facilitó la tarea, que normalmente llevaba una gran cantidad de tiempo[4].

#### SOLUCIONES Y PRUEBAS

El detector de plagio requiere una comparación entre palabras del texto de referencia y el texto evaluado, por lo cual debemos almacenar las palabras del primer texto de tal manera que podamos encontrarlos con facilidad.

En este caso creamos un Trie, el cual es un árbol que almacenará las palabras. En nuestro código, cada Nodo del Trie tendrá un arreglo de 27 posiciones, representando a cada una de las letras del alfabeto. Según la letra que se vaya a almacenar se agrega un nodo hijo a la posición correspondiente.



Para corresponder la posición de las letras debemos asignar un valor numérico teniendo a la letra a como 0, b como 1 y así sucesivamente.

Para lograr esto usaremos el código ascii de cada letra, y le restaremos el valor de la 'a', así obtendremos un valor relativo a las posiciones del alfabeto.

$$a=97$$

$$a-a=0$$

$$b-a=1$$

$$c-a=2$$

$$d-a=3$$

.

.

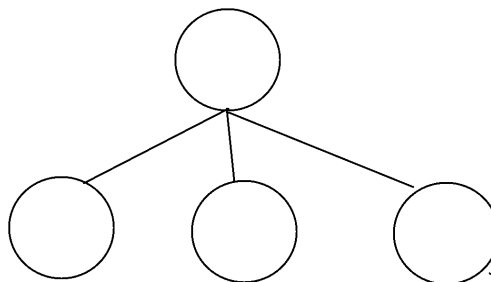
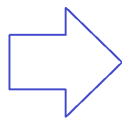
.

a	b	c	d	...
Nodo Hijo				

El procedimiento de detección de plagio requiere que tengamos un documento de referencia.

Usaremos la clase PlagiarismChecker para obtener el contenido de los documentos y agregar las palabras al Trie que será almacenado dentro de esta misma clase.

Mediante el botón de añadir archivo de la interfaz obtendremos el texto que contiene y agregaremos todas sus palabras al trie.



La métrica utilizada para el veredicto es el porcentaje de palabras similares del texto de referencia en relación a todas las palabras existentes en dicho texto. La clase ResulChacker buscará cada palabra en el trie e irá contando los resultados similares, luego lo comparará con la cantidad de palabras totales multiplicado por 0,2. Por tanto si el número de similitudes supera el 20 % el sistema alertará sobre plagio. Se compara cada uno de los documentos ingresados y la clase retornará un arreglo de booleanos que servirán a la interfaz para enviar un aviso de los resultados por cada documento.



**Plagio**



**No plagio**



**No plagio**



**Plagio**

...

En este caso usamos la clase ResulChacker dentro de PlagiarismChecker para poder enviarle el trie como parámetro.

### LECCIONES APRENDIDAS Y CONCLUSIONES

Durante el proceso de ejecución de este proyecto aprendimos a utilizar los árboles TRIE para facilitar la búsqueda de palabras específicas y verificar su existencia en un texto.

Ademas aprendimos, a grandes rasgos, como funciona un detector de plagio textual y cómo el usar TRIES facilita el proceso de comparación y determinar que tan similar es el documento de referencia con el párrafo evaluado.

### REFERENCIAS Y BIBLIOGRAFÍA

- [1]<https://es.wikipedia.org/wiki/Trie>
- [2]<https://dle.rae.es/plagio>
- [3]<https://dle.rae.es/plagiar#CU0knYP>
- [4]<https://www.ayudauniversitaria.com/detector-de-plagio/>