SISTEMA PARA VERIFICAR LA AUTENTICIDAD DE LOS TRABAJOS ENTREGADOS EN FORMATO DIGITAL PARA OBTENER EL GRADO DE LICENCIATURA EN EL INSTITUTO TECNOLÓGICO DE ACAPULCO BASADO EN EL ALGORITMO DE MIYER

Ing. Crisol Angelina Mendiola Piza [1], M.T.I. Eloy Cadena Mendoza[2],
M.T.I. Juan Miguel Hernández Bravo [3] y M.T.I. Rafael Hernández Reyna [4]

***Resumen*— En base al historial de los trabajos presentados por los alumnos del Instituto Tecnológico de Acapulco a partir del año 2007, podemos observar casos en los que se han detectado duplicidad de la información en las opciones de titulación de Tesis e Informe de Residencias. Por lo que se pretende diseñar un sistema que lleve a cabo una búsqueda, comparando el nuevo trabajo contra los existentes y alertar mediante un porcentaje de similitud este resultado. Para detectar estas similitudes entre los trabajos se ocupara la librería *Diff-Match-Patch* cuya función está basada en el algoritmo de *Myer*.**

.

***Palabras clave*— Librería, algoritmo de Myers y Diff-Match-Patch.**

Introducción

Los algoritmos de búsqueda están diseñados para localizar elementos con ciertas propiedades dentro de una estructura de datos, el concepto básico es buscar dentro del conjunto de palabras el patrón de coincidencia en el segundo texto. Existen diferentes tipos de búsqueda en que están basados los algoritmos de búsqueda por similitud, los más relevantes son búsqueda secuencial: se utiliza cuando el vector no está ordenado o no puede ser ordenado previamente, consiste en buscar el elemento comparándolo secuencialmente con cada elemento del vector hasta encontrarlo, o hasta que se llegue al final y búsqueda dicotómica: se utiliza cuando el vector en el que queremos determinar la existencia de un elemento está previamente ordenado. El resto del presente artículo está organizado de la siguiente manera: la sección 2 muestra la metodología que se usará para el desarrollo del sistema y enlista los requerimientos que ya han sido levantados. La sección 3 describe las herramientas utilizadas durante el desarrollo de esta prueba. La sección 4 se hablará de la librería y el algoritmo utilizado para la comparación. La sección 5 muestra los resultados obtenidos del funcionamiento del algoritmo de *Myer* y en la sección 6 se presentan las conclusiones y trabajos futuros.

Descripción del Método

Se decidió tomar como metodología el modelo de cascada, debido a que toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, para luego representarlas por separado del proceso en las especificaciones de los requerimientos del diseño del software, implementación y pruebas. En la figura 1 se puede apreciar el flujo de estas etapas de la metodología implementada.

[1] Ing. Crisol Angelina Mendiola Piza es estudiante de Maestría en Sistemas Computacionales en un programa PNPC en el Instituto Tecnológico de Acapulco, morrigan.yume@gmail.com (**autor corresponsal**).
[2] MTI. Eloy Cadena Mendoza es docente de Maestría del Instituto Tecnológico de Acapulco, eloy_cadena@yahoo.com
[3] MTI. Juan Miguel Hernández Bravo es docente y Jefe del departamento de Sistemas y Computación del Instituto Tecnológico de Acapulco, jhernandez@yahoo.com
[4] MTI. Rafael Hernández Reyna es docente de la Maestría en Sistemas del Instituto Tecnológico de Acapulco, jhernandez@yahoo.com

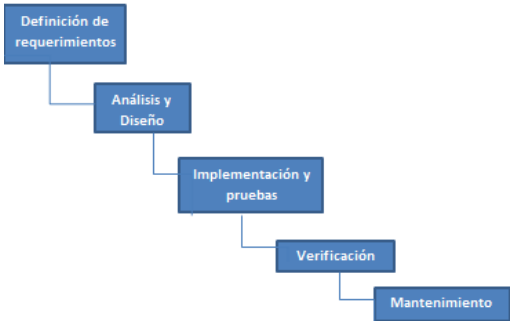

Figura 1. Metodología empleada para el sistema

Para el diseño del sistema es importante conocer los requerimientos funcionales y no funcionales, debido a que con ellos indicaremos que debe y no debe de hacer el sistema de acuerdo con las funciones que realizará el usuario en él. A continuación se listarán los requerimientos funcionales y no funcionales.

Requerimientos funcionales
- Autentificación de Usuario.
- Registrar Usuarios.
- Consultar Información de archivos cargados
- Cargar archivos.
- Modificar registro.

Requerimientos no funcionales
- Interfaz del sistema
- Ayuda acerca del sistema
- Mantenimiento
- Seguridad de la información

Herramientas

Se demostró la eficiencia de una biblioteca de búsqueda de similitudes en cadenas de texto, que implementa el algoritmo *diff de Myer* esta biblioteca se llama *Diff-Match-Patch*, utilizando como herramienta de programación *Eclipse*, con el lenguaje de programación Java. El equipo de cómputo en que se instaló dicha aplicación es una Lenovo con un procesador Intel i5 y 6 Gigas de Memoria Ram con sistema operativo Windows 10.

Algoritmo para la detección de similitudes

La utilidad diff, apareció por primera vez en el sistema operativo Unix, permitiendo generar las diferencias entre dos o más archivos o los cambios realizados entre una versión y otra del mismo archivo. El resultado de las diferencias encontradas se conoce como *diff*. Esta *librería* fue implementada con el algoritmo para detectar plagio en documentos invirtiendo los resultados que devuelve la herramienta, devuelve las igualdades encontradas entre dos archivos, utilizando el código fuente del proyecto *open source* google-*diff-match-patch* [1] como implementación del algoritmo.

Diseño del algoritmo de detección de similitudes

Para el diseño del algoritmo de detección de similitudes se utiliza como base el proyecto *open source* google- *diff-match-patch* [1] que se encarga de verificar desigualdades y similitudes entre dos textos ingresados. De los resultados entregados por el algoritmo se utilizan los resultados similares para contar cuántas palabras del texto fueron plagiadas textualmente y si el número de palabras iguales al texto original supera el cincuenta por ciento es

considerado como una copia y realiza el cálculo porcentual de copia utilizando una regla de tres simple; caso contrario automáticamente coloca en cero el porcentaje de plagio.

F*uncionamiento del algoritmo diff*

En informática, la utilidad *diff* es una herramienta de comparación de datos que calcula y muestra las diferencias entre dos archivos. A diferencia de las nociones de la distancia de edición utilizadas para otros propósitos, la diferenciación está orientada a líneas en lugar de caracteres. El comando *diff* muestra los cambios realizados en un formato estándar, de manera que tanto los humanos como las máquinas pueden entender los cambios y aplicarlos: dado un archivo y los cambios, el otro archivo puede crearse.

El *diff* comando se ejecuta desde la línea de comandos, pasándole los nombres de dos archivos: La salida del comando representa los cambios necesarios para transformar el archivo original en el nuevo archivo. *diff* original nuevo. Si los directorios son originales y nuevos, se ejecutará diff en cada archivo que exista en ambos directorios. Una opción, descenderá recursivamente cualquier subdirectorio correspondiente para comparar archivos entre directorios. *–r* . [2]

original :

```
1 Esta parte del
2 se documento  ha mantenido
3 igual que la versión  de la versión
4  . No debería
5 mostrarse  si no
6 cambia  . De lo contrario, ese
7  no estaría ayudando a
8 a  comprimir el tamaño de los
9  cambios.
10
11 Este párrafo contiene
12  textos que están desactualizados.
13 Se eliminará en el
14 futuro próximo  .
15
16 Es importante deletrear
17  revisar este documento. El
18 por  otro lado, un
19. palabra mal escrita no es
20  el fin del mundo.
21 Nada en el resto de
22  este párrafo necesita
23 ser cambiado  . Las cosas se
24 pueden añadir  después.
```

nuevo :

```
1 Este es un aviso importante
2  ! Debe
3  , por lo tanto se encuentra en el
4  al principio de este
5  documento!
6
7 Esta parte del
8 se documento  ha mantenido
9 igual a  de la versión a la
10 versión  . No debería
11 mostrarse  si no
12 cambia  . De lo contrario, ese
13  no estaría ayudando a
14 a  comprimir el tamaño de los
15  cambios.
16
17 Es importante deletrear
18  revisar este documento. El
19 Por otro lado, un
20  palabra mal escrita no es
21  el fin del mundo.
22 Nada en el resto de
23  que necesita este párrafo para
24  ser cambiado. Las cosas se
 pueden añadir después de
25 26
27 Este párrafo contiene
28  nuevas adiciones importantes
29  a este documento.
```

Figura 2. Ejemplo de dos textos original y nuevo

El comando *diff* comparando el texto original y el texto nuevo produce como resultado la siguiente salida:

nomenclatura	Descripción
a	Significa *added* es decir se agrega un nuevo texto en esta sección como resultado de la comparación de ambos textos.
d	Significa *deleted* es decir este texto será omitido del resultado ya que se eliminará debido a que se encontró igual el texto en ambas secciones.
c	Significa *changed* es decir que este resultado debe ser cambiado ya que presenta una gran

	similitud con el otro texto comparado.
0,1,2,3…29	Son los índices en donde se encuentra cada párrafo a ser evaluado, para el ejemplo se marca hasta el 29 por que el segundo contiene 29 párrafos.

Tabla 1. Describe cada uno de los subíndices mencionados en el ejemplo.

Texto de salida generado

0 al 6 //los índices 0-6 corresponden a los párrafos que se están evaluando en esta sección.
//el índice a1 indica que se está agregando un texto nuevo a partir de la unión de estas dos comparaciones, es decir la
//***a*** significa *added*, que en esta salida de la evaluación da como resultado un texto nuevo.
//A continuación se muestra la salida de texto donde se genera un texto nuevo como resultado de la combinación del
//texto original y el segundo texto.
> Este es un importante
> ¡aviso! Debería
> por lo tanto estar ubicado en
> el comienzo de esto
> Documento!
>
11 d 16
//Resultado de la salida del texto
<Este párrafo contiene
<texto que está desactualizado
<Se eliminará en el
<futuro cercano.
<
17c18
<Revisa este documento. En

> verifique este documento En
24al 26,29
> Este párrafo contiene
> nuevas incorporaciones importantes
> a este documento.

Licencia
Diff-Match-Patch está licenciado bajo la Licencia *Apache 2*.

Resultados

Para comprobar la utilidad de esta librería de diff se realizó la prueba con los datos propuestos en la figura 2, copiando el texto en la aplicación desarrollada en Java. El texto en este caso es corto para observar como es el funcionamiento del algoritmo en los resultados arrojados, pero también se hizo una ejecución de un texto de 6 hojas y un total del texto original de 2221 caracteres.

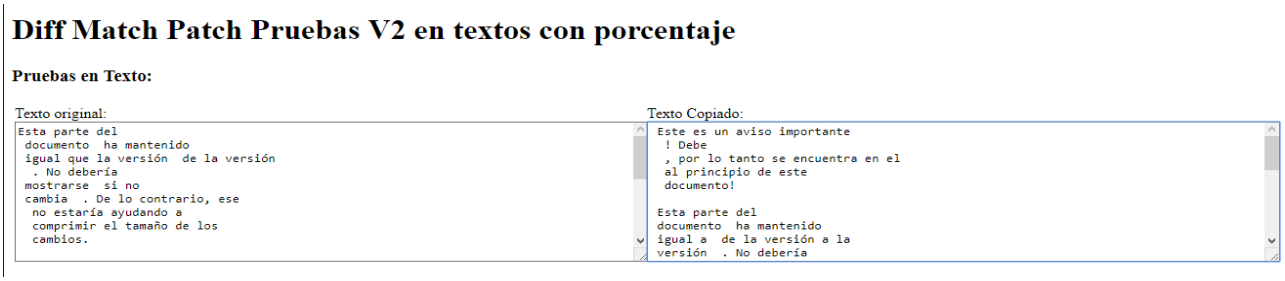

Figura 3. Datos ingresados del ejemplo de la figura 2 para ser evaluados por diff

Una vez aplicada la librería de diff se observa en su salida de resultado como se evalúa cada párrafo mostrándonos un porcentaje de similitud al ser evaluado cada párrafo pero también arroja como resultado el texto igual al ser comparados el texto original y el texto copiado. Con esta extracción del texto es posible comenzar a sacar un porcentaje de similitud ya que tomamos como base el total de caracteres por párrafo y se aplica una regla de tres para saber a cuanto equivale el texto igual del texto original. El tiempo de ejecución de esta comparación fue de 0.016 *s*. Una vez ejecutado el texto de prueba que se muestra en la figura 3 el resultado arrojado es el siguiente:

nomenclatura	Descripción
A	Significa *added* es decir se agrega un nuevo texto en esta sección como resultado de la comparación de ambos textos.
B	Significa *deleted* es decir este texto será omitido del resultado ya que se eliminara debido a que se encontró igual el texto en ambas secciones.
C	Significa *changed* es decir que este resultado debe ser cambiado ya que presenta una gran similitud con el otro texto comparado.
+0,1,2,3…536	Son los índices del número de caracteres a ser evaluado del texto original (primero)
-0,1,2,3…536	Son los índices del número de caracteres a ser evaluado del texto nuevo (segundo)
@@ -l, s + l, s @@	Forma original del encabezado principal de la sección (encabezado general). La información del rango contiene dos secciones, el rango para el fragmento del archivo original está precedido por un símbolo menos, y el rango para el nuevo archivo está precedido por un símbolo más, s donde l es el número de línea de inicio y s es el número de líneas a las que se aplica el cambio para cada archivo respectivo. En algunas versiones de diff, cada rango puede omitir la coma y los valores finales s, en cuyo caso s por defecto es 1.
@@	Son separadores para indicar que rango de caracteres se está avaluando.
%	Muestra el porcentaje de similitud encontrado en la evaluación de este rango de caracteres.

@@ -1,12 +1,124 @@ // En esta sección se muestra el rango por caracteres que fueron evaluados, el símbolo -
//nos indica que comprobemos este documento original, en 1 y con el símbolo de más que lo consultemos en el
//documento nuevo.
//Cada rango de fragmentos es del formato l, 12 donde l es el número de línea de inicio y 124 es el número de líneas a las que se aplica el cambio
+ Este es un aviso importante%0A ! Debe%0A , por lo tanto se encuentra en el%0A al principio de este%0A documento!%0A%0A
Esta parte // la letra a%0A nos indica el porcentaje de texto que se agrego
@@ -157,22 +157,9 @@ // En esta sección se muestra el rango por caracteres que fueron evaluados
ual
-que la versi%C3%B3n
+a
de
@@ -169,17 +169,30 @@ // En esta sección se muestra el rango por caracteres que fueron evaluados
versi%C3%B3n la letra a%C nos indica el porcentaje de texto que se cambio
-%0A
+ a la%0A versi%C3%B3n
. No d
@@ -317,107 +317,8 @@
s.%0A%0A
- Este p%C3%A1rrafo contiene%0A textos que est%C3%A1n desactualizados.%0A Se eliminar%C3%A1 en el%0A futuro pr%C3%B3ximo .%0A%0A
Es
@@ -368,16 +368,19 @@
to. El%0A
+Por
otro la
@@ -386,16 +386,17 @@
@@ -386,16 +386,17 @@
ado, un%0A

```
+
 palabra
@@ -457,16 +457,29 @@
 to de%0A
+que necesita
 este p%C3%A1r
@@ -487,17 +487,14 @@
 afo
-necesit
+par
 a%0A
+
 ser
@@ -502,18 +502,16 @@
 cambiado
-
 . Las co
@@ -536,13 +536,90 @@
 dir
-
 despu%C3%A9s
+ de%0A %0A Este p%C3%A1rrafo contiene%0A nuevas adiciones importantes%0A  a este documento
 .
```

Conclusiones

Como trabajo a futuro sobre la implementación de este algoritmo en la elaboración de un sistema para la detección de similitudes en trabajos en formato electrónico, se desea realizar una modificación al algoritmo base de Diff para obtener un resultado más limpio ya que en las ejecuciones mostradas el resultado que arroja es la evaluación pura y también imprime el párrafo similar y el porcentaje en evaluación en cada párrafo, y de igual manera llevar a cabo una comparación más amplia a mas textos existentes en nuestra fuente de búsqueda y no sea uno a uno como se han estado realizando para conocer cómo funciona el algoritmo.

Referencias

[1] Google-diff-match-patch, Proyecto open source de implementación del algoritmo diff, [En línea], 05-04-2019, http://code.google.com/p/google-diff-match-patch/

[2] Wikipedia, Utilidad diff, [En línea], 08-04-2019, http://en.wikipedia.org/wiki/Diff