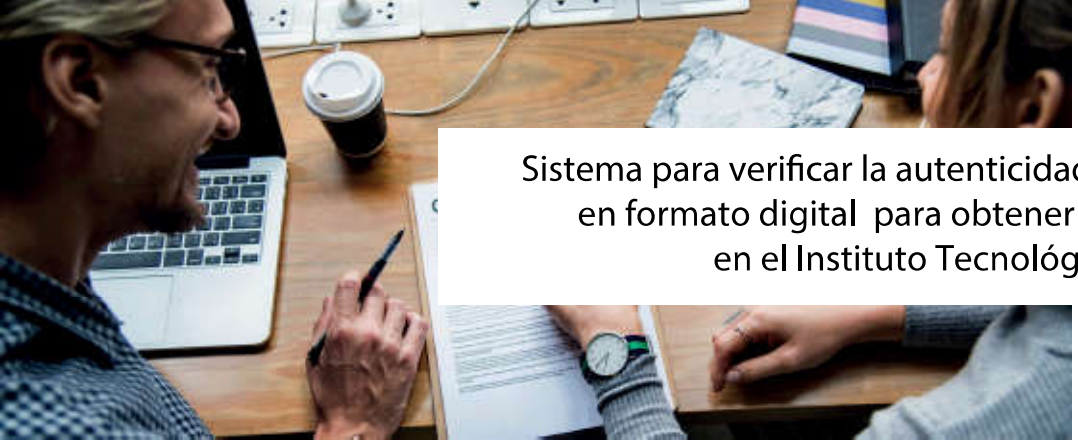

Sistema para verificar la autenticidad de los trabajos entregados en formato digital para obtener el grado de Licenciatura en el Instituto Tecnológico de Acapulco

Ing. Crisol Angelina Mendiola Piza[1], Instituto Tecnológico de Acapulco, Acapulco Gro., México. CP 39902
Ing.mendiola89@gmail.com, M.T.I. Eloy Cadena Mendoza [2], Instituto Tecnológico de Acapulco, Acapulco Gro., México. CP 39902,eloy_cadena@yahoo.com, M.T.I. Juan Miguel Hernández Bravo[3], Instituto Tecnológico de Acapulco, Acapulco Gro., México. CP 39902 jhernandez@yahoo.com, M.T.I. Rafael Hernández Reyna[4] Instituto Tecnológico de Acapulco, Acapulco Gro., México. CP 39902, rhernan7@yahoo.com.mx

Ing.mendiola89@gmail.com
(744)5870916

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.

Palabras clave:
Búsqueda por similitud, archivos electrónicos, base de datos, algoritmos metaheurísticos, estrategias de paralelización, paralelismo.

Abstract.
Based on the work history by the students of the Instituto Tecnológico de Acapulco since 2007, we can observe cases in which we have detected duplication of information in the Thesis and Residency Report options. What is intended is a system to carry out a search, comparing the new work against the current state and the alert by a percentage of similarity in this result.

Keywords:
Search by similarity, electronic files, database, metaheuristic algorithms, parallelization strategies, parallelism.

I.INTRODUCCIÓN

El proceso de titulación comienza una vez que el egresado cumpla con la acreditación del 100% de los créditos de su plan de estudios y acreditación de un programa de lengua extranjera. El egresado comienza su proceso de titulación una vez que entrega la documentación necesaria en el Departamento de Servicios Escolares y éste le genera un juego de recibido que deberá de entregar a la coordinación de titulación junto con una opción de titulación. Las opciones de titulación pueden ser: a) Tesis Profesional, b) Tesina, c) Proyecto de Investigación, d) Informe de Estancia e) Examen Global por Áreas de Conocimiento, f) Proyecto de Innovación Tecnológica, g) Informe Técnico de Residencia Profesional, según lo mencionado en el Manual de Lineamientos Académico-Administrativos del Tecnológico Nacional de México de Titulación Integral para la retícula 2010. Estas opciones de titulación varían respecto al plan de estudios que cursó el egresado. De estas opciones antes mencionadas solo se realizarán las comparaciones de las Tesis y Memoria de Residencias Profesionales, cuyo historial se encuentra disponible a partir del año 2005 en archivos electrónicos en formato PDF, almacenados en un equipo de cómputo en la Tesiteca del Instituto Tecnológico de Acapulco. Se opta trabajar con estas dos opciones de titulación ya que en un análisis realizado sobre la cantidad de alumnos titulados con respecto a las opciones que maneja la institución, se observa que estas dos tienen el mayor índice de titulación por las opciones mencionadas. La gráfica mostrada en la figura 1 representa el histórico de las titulaciones que se han realizado desde el año 2012 al 2017.

Figura 1. Total de alumnos titulados.

Con la intensión de disminuir la copia de reporte de los trabajos derivados de los proyectos afines se desarrollará un sistema que realice la comparación del nuevo trabajo entregado contra los ya existentes. Dicho sistema comenzará a analizar el texto desde el título hasta el marco teórico; limitando un porcentaje de similitud tolerable hasta un valor determinado, en cuanto se detecte que el porcentaje es mayor a esto se sigue realizando la comparación del demás contenido del archivo contra el existente para poder obtener un porcentaje final de similitud de las partes que conforman dicho documento.

Objetivo general

Detectar similitudes en los escritos de los trabajos de titulación a nivel licenciatura.

Objetivos específicos

a) Desarrollar un sistema que realice la comparación de trabajos electrónicos en formato PDF.
b) Alertar mediante un porcentaje de similitud a los interesados.
c) Conocer las características de las versiones Acrobat 7.0, Acrobat 8.0, Acrobat 9.0, Acrobat 9.1 y Acrobat X (10) de los archivos electrónicos en formato PDF.

Hipótesis

Alertar a los interesados mediante un porcentaje de similitud arrojado de la comparación del trabajo de titulación entregado con los archivos en formato electrónico PDF de los egresados titulados desde el 2005, debido a que en este año es cuando se obtiene el primer archivo en formato PDF de un trabajo de titulación de la carrera de Arquitectura. Esta comparación se llevará a cabo cuando el estudiante presente la propuesta de titulación en la División de Estudios Profesionales. Si el porcentaje de similitud es mayor al propuesto por las academias, se les alerta a los interesados de la posible copia con un trabajo de titulación ya existente.

II. MATERIAL Y MÉTODOS

Se desarrollaron dos aplicaciones para realizar las comparaciones de eficiencia con dos diferentes algoritmos. Para el manejo del algoritmo Knuth-Morris-Pratt (KMP) [1] se utilizó Netbeans IDE 8.0.1 con el lenguaje de programación C++ [2] y para el algoritmo de Diff de Myer [3] se desarrolló la aplicación con Visual Studio 2018 con el lenguaje de programación C# [4]. El equipo de cómputo en que crearon dichas aplicaciones es una Lenovo con un procesador Intel i5 y 6 Gigas de Memoria Ram con sistema operativo Windows 10.

Metodología

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 [5]. En la figura 2 se puede apreciar el flujo de estas etapas de la metodología implementada.

Figura 2. Metodología empleada para el sistema

En la etapa inicial se definen los requerimientos funcionales y no funcionales. Estos son los servicios, las restricciones y las metas del sistema y se obtienen mediante la consulta a los usuarios del sistema. Una vez obtenidos se definen a detalle y cada uno servirá como una especificación del sistema. 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

Análisis y diseño

En este apartado se modelan los diagramas diseñados para procesos principales que se llevarán a cabo en el sistema. Para el modelado de caso de uso se elaboró un diagrama para el proceso de recepción de expediente y la búsqueda del trabajo en el sistema, el diagrama se muestra en la figura 3.

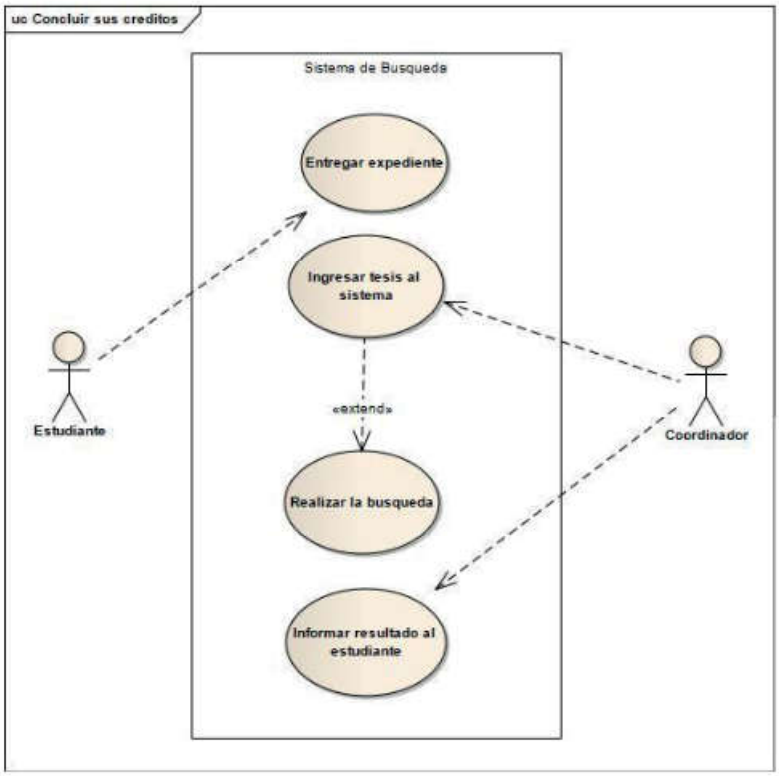

Figura 3. Diagrama de casos de uso

La descripción de los actores es la siguiente:

Estudiante: Persona egresada de la institución que cumple con todos los requisitos para integrar su expediente de titulación. Coordinador: Persona encargada de recibir el expediente e interactuar con el sistema.

Descripción de los casos de uso mencionados en el diagrama:

Entregar expediente:
1. El egresado pasa a Servicios Escolares para entregar los documentos necesarios e iniciar el proceso de integración de expediente de titulación.
2. Servicios Escolares entrega una copia de este expediente generado.
3. Entrega esta copia del expediente junto con un archivo electrónico de la Tesis o Memoria de Residencia a la coordinación de Titulación.

Ingresar tesis al sistema:
1. El coordinador valida sus datos de inicio de sesión en el sistema.
2. Ingresa al menú principal y selecciona el filtro en donde desea iniciar a buscar contra los archivos.

Realizar la búsqueda:
1. El sistema tiene el filtro inicial y comienza la búsqueda.
2. Se termina la búsqueda y se obtiene el porcentaje de resultado.
3. Si el porcentaje es alarmante procede a realizar una segunda búsqueda, de lo contrario si fue menor el porcentaje arrojado se acepta el expediente y se continúa con el trámite de titulación.

Informar resultados al estudiante:
1. El porcentaje arrojado de la búsqueda es mayor (alarmante) en todo el documento, se le informa al estudiante de una posible duplicidad y debido a esto no se le aceptará por el momento ese documento hasta que tenga las correcciones pertinentes.
2. El porcentaje arrojado es menor, el coordinador acepta el expediente de titulación y su archivo en electrónico.

A continuación, se describe el diagrama de secuencia como se muestra en la figura 4 del proceso de búsqueda del sistema:

Usuario: El usuario ha iniciado sesión, una vez hecho esto se dirige a solicitar al sistema acceder a la página de búsquedas.

Base de datos: En esta base de datos se estarán almacenando todos los archivos en electrónico para realizar las comparaciones y los procesos almacenados para realizar estas comparaciones.

Conexión de la base de datos: El gestor confirma la conexión con la Base de Datos, quien a su vez confirmará al procesador de los resultados, esté se encargará de generar el script de consulta y lo enviará a la Base de Datos directamente. Cuando el procesador de resultados recibe los resultados del gestor de la base de datos, los procesa y genera la página de resultados que se devolverá al usuario.

Filtro de Búsqueda: La petición anterior mediante el método post envía la solicitud al procesador de resultados para este punto, ésta solicita conectarse a la base de datos mediante la clase ConexionBD, que se conecta directamente con el gestor de bases de datos.

Sesión principal: el menú principal atiende esta petición y responde enviando la página. Allí el usuario especifica las palabras de búsqueda (en este caso primero se atiende a seleccionar el filtro del tema), y envía la petición a Index.

Petición de la búsqueda: El usuario inicia la sesión y se dirige a la ventana principal donde seleccionará el filtro de búsqueda.

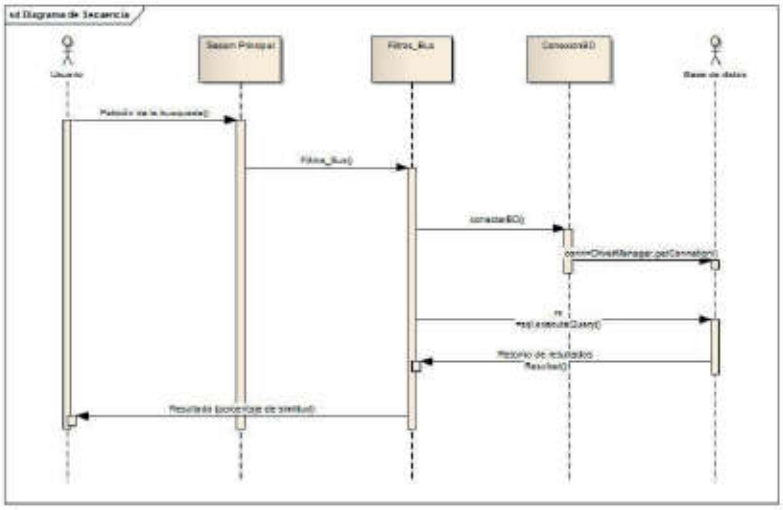

Figura 4. Diagrama de secuencia

III. RESULTADOS

Se hicieron pruebas con dos algoritmos diferentes para tener un comparativo de la eficiencia de cada uno. Estos dos algoritmos fueron el algoritmo Knuth-Morris-Pratt (KMP) implementado en una aplicación de consola con el lenguaje de programación C++ y el segundo es Diff-Match-Patch que es una librería de uso, en esta librería se implementa el algoritmo Diff de Myer. Para esta segunda prueba se utilizó el lenguaje de programación C# y con el Visual Studio se creó una interfaz más de acuerdo a la que vamos a desarrollar en nuestro proyecto.

Este primer algoritmo KMP es algoritmo de búsqueda secuencial de texto llamado algoritmo de fuerza bruta, el cual recorrerá el texto carácter a carácter, buscando coincidencias con los caracteres de las palabras buscada en cada uno de los párrafos. Se sitúa el patrón en la primera posición y se compara carácter por carácter hasta encontrar un valor o llegar al final del patrón, se pasa a continuación a la siguiente posición y se repite este proceso. Termina hasta alcanzar el final del texto, no existe un pre procesamiento del patrón. [6]

Figura 5. Aplicación de consola en C++

En la figura 5 se ingresa un texto regular de tres renglones y posteriormente se procede a ingresar un párrafo para compararlo contra el original, se observa en la tabla de fallas que donde se marca un índice de 1 es que se encontró una palabra similar en esa posición, pero a partir de la posición 59 es donde se localizó la cadena completa de similitud; como resultado final nos arroja que si se encontró una similitud en este texto, nos pregunta si deseamos seguir buscando más similitudes, de lo contrario salimos del programa.

Para esta segunda prueba se utilizó una biblioteca de alto rendimiento que se puede utilizar en varios lenguajes de programación para manipular texto sin formato. Las bibliotecas Diff Match y Patch ofrecen algoritmos robustos para realizar las operaciones necesarias para sincronizar texto sin formato. Este algoritmo funciona mediante la búsqueda de forma recursiva en el centro del partido de dos secuencias, con la secuencia de comandos de edición más pequeña. Una vez hecho esto sólo el partido inicial se memoriza, y las dos sub-secuencias anteriores y posteriores que se comparan de nuevo de forma recursiva hasta que no hay nada más para comparar. Para encontrar la pareja central se realiza haciendo coincidir los extremos de sub-secuencias en la menor medida posible, ya que en cualquier momento no será posible aumentar el guion de la edición 1, explorando cada posición más alejada alcanzada hasta allí para cada diagonal y ver hasta los nuevos partidos que pueden ir, si un partido encuentra una palabra del otro extremo el algoritmo acaba, al encontrar el partido central.

El resultado de la implementación de este algoritmo se muestra en la Figura 6, con la ayuda de esta librería ingresamos un párrafo de mayor tamaño que en la aplicación anterior. Una vez ingresado el párrafo inicial ingresamos el siguiente texto a comparar, se tiene como resultado de la búsqueda dos secciones una donde se muestra el texto que se encontró igual y el que no es igual, mostrando este resultado en cada uno de los apartados de la aplicación. Si el texto es completamente igual la parte que muestra el texto diferente quedará en blanco y todo el texto que es igual se mostrará completamente el párrafo.

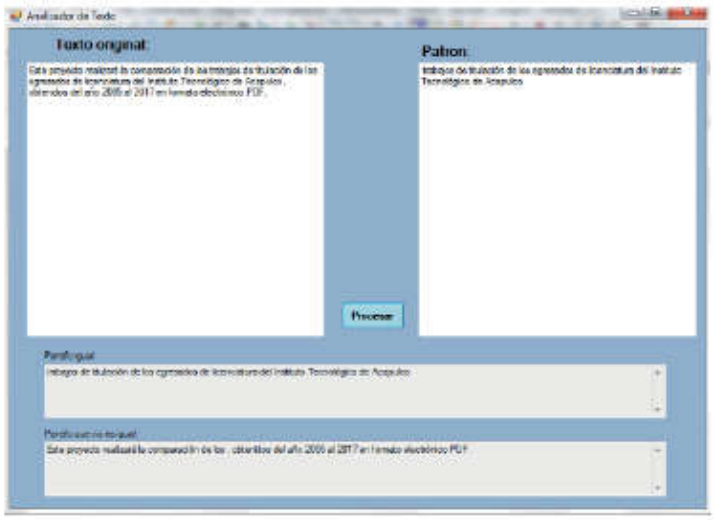

Figura 6. Implementación de la librería Diff-Match-Patch

IV.CONCLUSIONES

Con los resultados obtenidos al comparar textos con estos dos diferentes algoritmos se determinó que es más eficaz trabajar con la librería de Diff-Match-Patch basada en el algoritmo Diff de Myer, debido a que procesa una mayor cantidad de caracteres y sus búsquedas son más exactas, incluso si tomamos diferentes fragmentos de un mismo párrafo intercalando el orden y colocarlo como un texto nuevo. Se detectó que con el algoritmo de KMP es solo eficiente hasta enunciados de no más de 10 palabras ya que solo en nuestra tabla de índices de coincidencias se localizaban las búsquedas por palabras, pero no por la oración completa y al final aunque se refleje en la tabla dichas coincidencias, como resultado final arroja que no se encontró alguna similitud en los textos.

V. BIBLIOGRAFÍA

Pablo, C.P., (2012). Algoritmia. Obtenida el 10 de Octubre del 2018, de la página electrónica: http://bibing.us.es-/proyectos/abreproy/12077/fichero/memoria%252Fpor_capitulos%252F02.algoritmia.pdf

Yan, S.V., (2011). Código informático. Obtenida el 20 de Noviembre del 2018, de la página electrónica: https://xcodigoinformatico.blogspot.com/2011/07/algoritmo-knuth-morris-pratt-string.html

Eugene W. Myers (1986). Un algoritmo de diferencia de O (ND) y sus variaciones, vol. 1 No. 2. [Versión electrónica] Algorithmica, 251-266.

Google Open Source (2018Diff Match Patch. Obtenida el 5 de Diciembre del 2018, de la página electrónica: https://opensource.google.com/projects/diff-match-patch

I. Somerville, (2011). Ingeniería de Software. México, México. Pearson Educación de México.

Google, S. (s.f.). Búsqueda secuencial de texto. Obtenida el 27 de Septiembre de 2018, de la página electrónica: https://sites.google.com/site/busquedasecuencialdetexto/