Estudio comparativo de los API's de búsqueda de Google, Yahoo y Bing para el desarrollo de aplicaciones anti plagio de textos en documentos.

Jorge Cárdenas Monar[1], César Villacís Silva[2], Marco Vergara[3]
[1] Tata Consultancy Services, Quito-Ecuador, jorgeluis.c@tcs.com
[2] Departamento de Ciencias de la Computación Escuela Politécnica del Ejército, Sangolquí-Ecuador, cjvillacis@espe.edu.ec
[3] Departamento de Ciencias de la Computación Escuela Politécnica del Ejército, Sangolquí-Ecuador
mavergara@espe.edu.ec

RESUMEN

El desarrollo de aplicaciones de detección de plagio han aumento considerablemente en los últimos años debido al uso de API's de búsqueda como componentes básicos del aplicativo. El presente artículo propone el realizar un estudio comparativo teórico y practico de los API's de búsqueda de los motores más conocidos, Google, Yahoo y Bing, para ser tomados en cuenta como componentes fundamentales en el desarrollo de aplicaciones anti plagio de textos en documentos. Para llevar a cabo el estudio, se realizó una comparativa teórica donde se definieron criterios de comparación en base a la documentación de los API's y la ponderación de cada uno de los criterios está basado en la relevancia que cada uno tiene en el desarrollo de este tipo de aplicaciones. En la comparativa práctica los criterios fueron definidos en base al diseño de un prototipo básico de software de una aplicación anti plagio de textos en documentos, que consume los tres API's de búsqueda y utiliza la tecnología Windows Forms para su interfaz gráfica, y las pruebas de funcionalidad realizadas con el prototipo. Finalmente se definió cual es el mejor API de búsqueda para el desarrollo de estas aplicaciones tomando en cuenta las comparativas teórica y práctica y con este API se desarrolló un prototipo final utilizando la tecnología ASP.NET MVC 3 para su interfaz gráfica web. El estudio comparativo utilizó partes de las metodologías ágiles SCRUM (Planificación) y XP (Codificación y Pruebas) y UML para el modelado del prototipo.

Palabras Clave: API, Google, Yahoo, Bing, Plagio, SCRUM, XP, UML

ABSTRACT

The development of plagiarism detection applications had increased considerably in recent years due to the use of search API's as basic components of the application. This article proposes a theoretical and practical comparative study of the search API's engines best known, Google, Yahoo and Bing, to be taken into account as critical components in the development of anti-plagiarism applications of text in documents. To carry out the study, was made a theoretical comparative were defined comparison criteria based on the documentation of the API's and the weighting of each criterion is based on the relevance that each one have in the development of such applications. In the practical comparative were defined criteria based on the design of a basic prototype software application of an anti-plagiarism application of text in documents, which consume the three search API's and use the Windows Forms technology for its graphical interface, and functionality tests performed on the prototype. Finally, was defined which is the best search API for the development of these applications taking into account the theoretical and practical comparative and with this API was developed the final prototype using ASP.NET MVC 3 as technology for its Web GUI. The comparative study uses parts of the agile methodologies SCRUM (Planning) and XP (Coding and Tests) and UML for modeling the prototype.

Keywords: API, Google, Yahoo, Bing, Plagiarism, SCRUM, XP, UML

1. INTRODUCCIÓN

En los últimos años el plagio de textos en documentos se ha extendido de manera considerable a escala global debido a la gran cantidad de información que se dispone en internet y a la facilidad de acceso a la misma [1].

Tomando en cuenta que la copia directa de información desde el internet sin citar la fuente licenciada con derechos de copia o copyright en inglés representa un delito en contra de los derechos de autor, el utilizar aplicaciones que detectan plagio en documentos se ha convertido en una herramienta primordial para universidades grandes de todo el mundo que no toleran la copia, inclusive imponiendo severas sanciones a los implicados, como la expulsión de la institución, como es el caso de la universidad de Virginia en los Estados Unidos [2].

En un inicio, la creación y mantenimiento de este tipo de aplicaciones tenía un costo muy elevado debido, principalmente, a la infraestructura que debían manejar para poder verificar plagio en la enorme cantidad de sitios que tiene la internet y en sus propios documentos, y la complejidad del algoritmo para realizar el análisis semántico y sintáctico del lenguaje utilizado en el documento. Con estas características nacieron aplicaciones muy conocidas como TurnItIn, la aplicación más utilizada en Estados Unidos y Europa para esta tarea, la cual maneja su propia infraestructura y base documental que cada cierto tiempo ejecuta un escaneo de millones de sitios en internet para tenerla actualizada, además de algoritmos propietarios para el análisis de plagio [3]. Con la aparición de la web 2.0 y específicamente los API's públicos para todo tipo de necesidades, el entorno cambió radicalmente. Los API's públicos de búsqueda de Google, Yahoo y Bing abrieron el camino para las primeras aplicaciones de detección de plagio distintas que centraron su esfuerzo en el algoritmo de detección de plagio y no en las fuentes de comparación ya que los API's se encargarían de esa ardua tarea. Así aparecieron aplicaciones como Plagium.com [4], una aplicación web que permite detectar plagio en un texto ingresado por el usuario; este aplicativo utiliza los API's de Google y Bing para obtener las fuentes de plagio y mediante su propio algoritmo de detección de plagio muestra los resultados gratuitamente al usuario de forma gráfica y textual.

El apogeo de las aplicaciones de detección de plagio similares a Plagium.com hizo que surja una pregunta muy importante a tomar en cuenta en el diseño de este tipo de aplicaciones. ¿Cuál es el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos? La respuesta a esta pregunta es el propósito de esta investigación en las que se consideran los motores de búsqueda más conocidos y representativos de internet, Google con su API Google Custom Search Engine, Yahoo con su API Yahoo BOSS API y Bing de Microsoft con su API Bing Search API. A cada uno se lo puso a prueba contra criterios de comparación teórica (características especiales de cada API) y de comparación práctica (rendimiento y calidad de resultados devueltos por el API).

Se ha realizado un análisis para determinar cuáles son los criterios de comparación más adecuados para identificar el mejor API en el ámbito teórico y práctico. En el ámbito teórico se utiliza como fuente de información para la generación de criterios la documentación de cada API y en el ámbito práctico se utilizó el diseño y las pruebas de funcionalidad realizadas sobre un prototipo de software desarrollado bajo la plataforma .NET con el lenguaje C# encargado de consumir cada uno de los API's y devolver los resultados necesitados utilizando como algoritmo de detección de plagio la utilidad diff [5], implementada por primera vez en el sistema operativo Unix. Para validar el correcto funcionamiento del prototipo se desarrolló la interfaz gráfica o GUI en Windows Forms. El mejor API que resultó del estudio comparativo fue utilizado en un prototipo final diseñado como aplicación web con la tecnología ASP.NET MVC 3. Para el desarrollo de la investigación se utilizaron las metodologías ágiles Scrum (Planificación) y XP (Codificación y Pruebas) y además UML para el diseño del prototipo de software.

En consecuencia, las contribuciones de este estudio son: Obtener los criterios de comparación teóricas y prácticas para establecer el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos; diseñar y construir una aplicación de escritorio con una GUI en Windows Forms que utilice los tres API's mencionados para realizar las pruebas funcionales basadas en los criterios comparativos prácticos; identificar el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos basados en la comparación teórica y practica y diseñar y construir una aplicación web con una GUI en ASP.NET MVC 3 que utilice el mejor API de los tres investigados.

2. MATERIALES Y MÉTODOS

2.1 Fundamentos teóricos

Los API's utilizados para la comparativa teórica y su documentación son los siguientes:

a) Google Custom Search API JSON/Atom [6].
b) Yahoo BOSS API [7].
c) Bing Search API [8].

2.2 Algoritmo de detección de plagio

La utilidad diff, que apareció por primera vez en el sistema operativo Unix, permite 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 utilidad fue implementada como el algoritmo para detectar plagio en documentos invirtiendo los resultados que devuelve la herramienta, es decir, devuelve las igualdades encontradas entre dos archivos, y se utilizó el código fuente del proyecto open source google-diff-match-patch [9] como implementación del algoritmo.

2.3 Metodologías Utilizadas

Para el desarrollo del estudio comparativo se utilizó: SCRUM como metodología de manejo de proyectos tomando como partes importantes de la metodología la planificación, artefactos (Product Backlog, Sprint Backlog) y el proceso iterativo; XP como metodología de desarrollo de software tomando como partes importantes de la metodología los estándares de codificación, pruebas unitarias y de funcionalidad e historias de usuarios para definir requisitos específicos del prototipo; UML como estándar de lenguaje de modelado.

3. DISEÑO E IMPLEMENTACIÓN

3.1 Criterios de comparación teórica

En base a la documentación de cada uno de los API's de búsqueda utilizados en el estudio y la experiencia de terceros (blogs, foros, sitios web), se determinaron los criterios de comparación teórica que se muestran en la Tabla I con sus respectivas ponderaciones las cuales están basadas en la importancia del criterio en el ámbito del desarrollo de aplicaciones anti plagio de textos en documentos

Tabla I: Criterios de comparación teórica con su ponderación respectiva

Criterio	Descripción	Ponderación
Tipos de documentos de búsqueda	Si el API permite especificar tipos de documentos, cuántos y cuáles de estos tienen más importancia en la búsqueda de plagio.	30
Resumen extenso	Si el resumen devuelto por el API tiene una extensión considerable para tomarlo en cuenta y buscar coincidencia; la extensión se basa en número de caracteres (300 o más).	30
Fecha de última actualización del sitio	Si el API devuelve junto con cada uno de los resultados la última fecha de actualización del sitio o la última vez que paso el crawler por el mismo.	20
Madurez del API	Grado de madurez del API utilizado, tomando como referencia el número de versiones y si es una versión estable o en fase de desarrollo.	20
Tipos de formatos de respuesta	Cantidad de formatos de respuesta que ofrece el API y cuantos son relevantes, ejemplo: JSON, XML, ATOM.	20
Tipos de fuentes de búsqueda	Cantidad de tipos de fuentes de búsqueda del API de donde se puede obtener resultados relevantes de plagio, por ejemplo: web, imágenes, noticias,	30

	blogs.	
Número máximo de resultados por página	Cantidad de documentos que devuelve el API por página.	20
Filtrado de pornografía	Si el API tiene la opción de filtrar las búsquedas por contenido pornográfico.	20
Lenguajes que soporta	Cantidad de lenguajes que soporta el API para las búsquedas de plagio.	30
Implementación de OpenSearch	Si el API implementa una o varias de las tecnologías que plantea OpenSearch.	20
Autorización OAuth	Si el API tiene la opción de autenticarse en el API a través de OAuth.	20
Sitios preferenciales de búsqueda	Si el API permite especificar por cuales sitios buscar primero.	30
Restringir sitios de búsqueda	Si el API permite especificar cuáles sitios puede incluir en la búsqueda y cuáles no.	20
Resumen libre de HTML	Si el API permite devolver el resumen de cada uno de los resultados en texto plano sin HTML.	30
Filtrado de duplicados	Si el API tiene la opción de filtrar los resultados duplicados en la búsqueda actual.	30
Costo de las peticiones	Costo por utilización del API, basada en un promedio entre 1.000, 10.000 y 100.000 peticiones diarias (Costo 1.000 peticiones + Costo 10.000 peticiones + Costo 100.000 peticiones) / 3.	30
Configuración del API	Facilidades para la configuración del API, si posee interfaz web.	10
Monetización	Si el API permite generar ingresos mediante publicidad.	10
Tipos de fuentes de búsqueda simultáneos	Si el API tiene la opción de buscar en algunos tipos de fuentes de búsqueda al mismo tiempo.	20
Número aproximado de aplicaciones utilizando el API	Cantidad aproximada de aplicaciones de terceros que consumen el API y este forma parte fundamental del diseño del aplicativo.	10
Numero de aplicaciones anti plagio utilizando el API o directamente el motor de búsqueda	Cantidad aproximada de aplicaciones anti plagio de textos o documentos que consumen el API o utilizan el motor de búsqueda directamente y que este forme parte fundamental del diseño del aplicativo.	20
Total		470
Total API / 100		(# * 100) / 470

3.2 Diseño del modelo del prototipo de software de una aplicación anti plagio de textos en documentos

Para el diseño del modelo de la aplicación se ha considerado UML (lenguaje de modelado unificado). La Figura 1 muestra el diagrama de clases de la aplicación abstraído en paquetes.

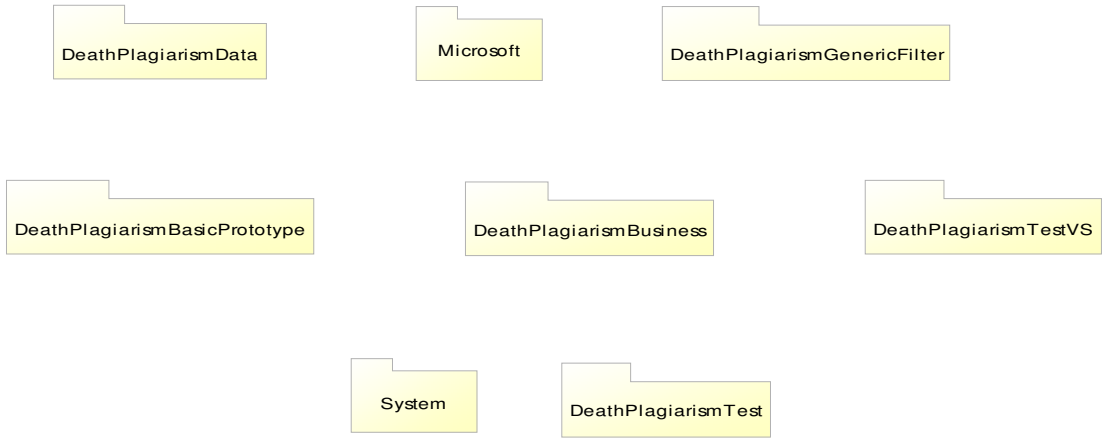

Fig. 1: Diagrama de clases del prototipo de software en paquetes

La funcionalidad de cada uno de estos paquetes se muestra en la Tabla II.

Tabla II: Descripción de los paquetes de clases del prototipo

Paquete	Descripción
DeathPlagiarismData	Encargado de consumir los diferentes API's de búsqueda, la implementación del algoritmo diff, el algoritmo para manejar la autenticación OAuth y un Crawler.
Microsoft	Referencias de la dll Microsoft.Http utilizadas para la conexión a los servicios web.
DeathPlagiarismGenericFilter	Encargado de crear objetos genéricos que deben ser entregados a la capa superior (DeathPlagiarismBusiness) independientemente del API utilizado, ejecutar el análisis de plagio entre el documento base y el sitio de internet utilizando el algoritmo diff de la capa inferior (DeathPlagiarismData) y almacenar y leer del disco duro las páginas descargadas como texto de los análisis de plagio previos.
DeathPlagiarismBasicPrototype	Interfaz gráfica en Windows Forms del prototipo.
DeathPlagiarismBusiness	Encargado de leer documentos y crear objetos debidamente estructurados para su presentación en la capa superior. Para obtener estos objetos sigue un flujo que comprende las siguientes fases: • Lectura del documento. • Enviar cada párrafo del documento contra los API's seleccionados para obtener los sitios de donde posiblemente se plagió. • Verificar el porcentaje de plagio del párrafo contra los sitios encontrados. • Almacenar cada resultado en objetos jerarquizados. • Devolver los resultados a la capa de presentación.
DeathPlagiarismTestVS	Encargado de ejecutar las pruebas de unidad del prototipo utilizando el marco de trabajo MSUnit.
System	Referencias del nombre de espacio System utilizadas en lectura y escritura de archivos y conexión a web services mediante sockets.
DeathPlagiarismTest	Encargado de ejecutar las pruebas de unidad del prototipo utilizando el marco de trabajo NUnit.

3.3 Diseño del algoritmo de detección de plagio

Para el diseño del algoritmo de detección de plagio se utilizó como base el proyecto open source google-diff-match-patch [9] el cual 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 cuantas palabras del texto fueron plagiadas textualmente y si el número de palabras iguales al texto original supera el quince por

ciento es considerado como plagio y realiza el cálculo porcentual de copia utilizando una regla de tres simple; caso contrario automáticamente coloca en cero el porcentaje de plagio. El funcionamiento del algoritmo diff se encuentra detallado en internet en el enlace [5].

3.4 Criterios de comparación práctica

En base al diseño del prototipo de software y pruebas de funcionalidad sobre el mismo, se determinaron los criterios de comparación práctica con sus respectivas ponderaciones las cuales están basadas en la importancia del criterio en el ámbito del desarrollo de aplicaciones anti plagio de textos en documentos. Los criterios se muestran en la Tabla III.

Tabla III: Criterios de comparación práctica con sus respectivas ponderaciones

Criterio	Descripción	Ponderación
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos en la primera ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas en la primera ejecución (2 párrafos). Tiempo promedio de los 2 resultados obtenidos: Suma de tiempos de espera / 2.	30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas desde la segunda ejecución iterando 3 veces seguidas (2 párrafos). Tiempo promedio de los 6 resultados obtenidos: Suma de tiempos de espera / 6.	30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos en la primera ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas en la primera ejecución (5 párrafos). Tiempo promedio de los 5 resultados obtenidos: Suma de tiempos de espera / 5.	30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	El tiempo promedio en milisegundos de espera en el API para obtener los 5 resultados de cada una de las consultas especificadas desde la segunda ejecución iterando 3 veces seguidas (5 párrafos). Tiempo promedio de los 15 resultados obtenidos: Suma de tiempos de espera / 15.	30
Tiempo necesario para el desarrollo del consumo del API de búsqueda	Tiempo en horas necesitadas para desarrollar el consumo del API de búsqueda.	25
Similitud de los sitios consultados mediante el API vs los de la web del proveedor del API	Cantidad de sitios iguales que devuelve el API con respecto a los sitios que devuelve la web del proveedor del API en la primera página.	50
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 2 párrafos	Tiempo en milisegundos de espera en obtener los 5 resultados de cada uno de los dos párrafos del documento.	30
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 5 párrafos	Tiempo en milisegundos de espera en obtener los 5 resultados de cada uno de los cinco párrafos del documento.	50
Total		275
Total API / 100		(# * 100) / 275

4. RESULTADOS

Los resultados obtenidos en la comparación teórica se muestran en la Tabla IV.

Tabla IV: Resultados de los criterios de comparación teórica

Criterio	API		
	Google	Yahoo	Bing
Tipos de documentos de búsqueda	25/30	28/30	30/30
Resumen extenso	10/30	30/30	10/30
Fecha de última actualización del sitio	0/20	20/20	20/20
Madurez del API	10/20	20/20	20/20
Tipos de formatos de respuesta	20/20	20/20	20/20
Tipos de fuentes de búsqueda	5/30	30/30	25/30
Número máximo de resultados por página	10/20	20/20	20/20
Filtrado de pornografía	20/20	20/20	20/20
Lenguajes que soporta	30/30	20/30	30/30
Implementación de OpenSearch	20/20	20/20	20/20
Autorización OAuth	20/20	20/20	0/20
Sitios preferenciales de búsqueda	30/30	0/30	0/30
Restringir sitios de búsqueda	20/20	20/20	0/20
Resumen libre de HTML	30/30	30/30	30/30
Filtrado de duplicados	30/30	0/30	0/30
Costo de las peticiones	20/30	0/30	30/30
Configuración del API	10/10	10/10	10/10
Monetización	10/10	10/10	10/10
Tipos de fuentes de búsqueda simultáneos	0/20	20/20	20/20
Número aproximado de aplicaciones utilizando el API	10/10	10/10	5/10
Numero de aplicaciones anti plagio utilizando el API o directamente el motor de búsqueda	20/20	20/20	10/20
Total	**350/470**	**368/470**	**330/470**
Total API / 100	**74,47**	**78,30**	**70,21**

Los resultados obtenidos en la comparación práctica se muestran en la Tabla V.

Tabla V: Resultados de los criterios de comparación práctica

Criterio	API		
	Google	Yahoo	Bing
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos en la primera ejecución	10/30	20/30	20/30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 2 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	20/30	10/30	20/30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos en la primera ejecución	20/30	10/30	20/30
Tiempo promedio de espera en el API de búsqueda utilizando un documento de 5 párrafos ejecutando 3 veces seguidas el análisis comenzando desde la segunda ejecución	30/30	20/30	30/30
Tiempo necesario para el desarrollo del consumo del API de búsqueda	15/25	5/25	15/25
Similitud de los sitios consultados mediante el API vs los de la web del proveedor del API	50/50	25/50	0/50

Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 2 párrafos	10/30	30/30	20/30
Tiempo de espera para obtener los resultados del análisis de plagio en un documento con 5 párrafos	50/50	30/50	10/50
Total	**205/275**	**150/275**	**135/275**
Total API / 100	**74,54**	**54,54**	**49,09**

Para definir el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos se decidió ponderar cada comparación como muestra la Tabla VI.

Tabla VI: Ponderaciones para cada uno de las comparativas realizadas

Estudio	Peso en porcentaje
Comparativo teórico	40%
Comparativo práctico	60%
Total	**100%**

Basados en la ponderación establecida utilizando la fórmula: $Y = \frac{X*40}{100}$ y la fórmula $Y = \frac{X*60}{100}$, los resultados obtenidos por los diferentes API's se muestran en la Tabla VII.

Tabla VII: Resultados de las comparativas realizadas basadas en la ponderación definida.

Estudio/API	Google	Yahoo	Bing
Comparativo teórico	29,79/40	31,32/40	28,08/40
Comparativo práctico	44,72/60	32,72/60	29,45/60
Total	**74,51/100**	**64,05/100**	**57,54/100**

Como muestran las tablas anteriores el mejor API en el ámbito de la comparativa teórica para el desarrollo de aplicaciones anti plagio de textos en documentos es el API de Yahoo con 78,30%, en segundo lugar está el API de Google con 74,47% y por último se encuentra el API de Bing con 70,21%. Mientras que en el ámbito de la comparativa práctica el mejor API de búsqueda es el API de Google con 74,54, luego le sigue el API de Yahoo con 54,54 y en último lugar el API de Bing con 49,09. Si bien estos resultados identifican cuantitativamente cual puede ser el mejor API para el desarrollo de este tipo de aplicaciones hay que tomar en cuenta ciertas características de los API's de búsqueda que permiten identificarlos como posibles candidatos para su puesta en producción en una aplicación de detección de plagio. En el caso de Google su API es rápido, y devuelve resultados muy relacionados a lo buscado pero lastimosamente está en fase Labs conocida formalmente como versión beta la cual es susceptible a cambios como también es muy probable que el proyecto del API de búsqueda sea abandonado por Google como lo han hecho con los dos API antecesores a este: Google SOAP Search API y Google Ajax Search API. También conviene tomar en cuenta que el API no posee tantas opciones de tipos de fuentes donde buscar plagio como Yahoo o Bing, por ejemplo Google no puede buscar como tipos de fuentes imágenes o dentro de blogs como Yahoo lo hace. En el caso de Yahoo de igual manera el API devuelve resultados rápidamente y con resultados debidamente relacionados a lo buscado pero lastimosamente su costo está basado en unidades que contiene cierto número de peticiones, en el caso de peticiones de búsqueda en la web el costo es de 0,80 centavos por 1.000 consultas pero al aumentar la cantidad de consultas el costo se eleva por ejemplo si se realizaran hasta 10.000 consultas en un día el costo sería de 8 dólares mientras que en Google las mismas 10.000 costarían 5 dólares. Por último el API de Bing que es el que peores resultados obtuvo tanto en la comparativa teórica y práctica es el más rápido de los API en devolver resultados pero estos son poco o nada relacionados a lo buscado provocando que el algoritmo de detección de plagio busque falsos positivos y se pierda tiempo en la detección de plagio en general.

5. TRABAJOS RELACIONADOS

No existe suficiente documentación sobre trabajos directamente relacionados, dado que un estudio comparativo de los API's de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos puede ser perjudicial a la imagen de las empresas cuyos API's fueron analizados. Sin embargo hay análisis

comparativos de herramientas de detección de plagio como el reporte escrito por Jurriaan Hage, Peter Rademaker y Nik_e van Vugt [10].

6. CONCLUSIONES Y TRABAJO FUTURO

Basados en los resultados obtenidos por la comparación teórica, el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos es el API de Yahoo mientras que en los resultados obtenidos por la comparación práctica el mejor API de búsqueda para el desarrollo de aplicaciones anti plagio de textos en documentos es el API de Google y finalmente basados en los resultados y la ponderación definida para cada comparativa, el mejor API para el desarrollo de aplicaciones anti plagio de textos en documentos es el API de Google a pesar de estar en una versión Labs y que no tiene tantas opciones de tipos de fuentes donde buscar, le sigue Yahoo a pesar de ser el API con mas opciones de fuentes donde buscar incluso en Blogs pero pierde gran parte de la comparativa por su costo elevado en las peticiones realizadas al API y por último está el API de Bing porque es el que brinda los peores resultados relacionados con la búsqueda realizada lo que conlleva a perdida de tiempo en la detección de plagio general del documento.

Se puede mejorar la implementación del algoritmo diff para aumentar el rendimiento de la aplicación, como también se podría agregar mas API's públicos de búsqueda para probar el rendimiento y funcionalidad de estos y verificar si brindan mejores resultados o un mejor rendimiento que los API's analizados en el estudio, además aumentar características al aplicativo y desarrollar el código necesario para la detección de plagio de imágenes usando los mismos API's de búsqueda y un algoritmo diff para archivos binarios.

7. REFERENCIAS

[1] The Daily, Internet Leads to Increased Plagiarism, [En línea], 11-01-2012, http://dailyuw.com/news/2007/may/21/internet-leads-to-increased-plagiarism/

[2] PilotOnline.com, Students expelled from U.Va.shipboard program for plagiarism, [En línea], 11-01-2012, http://hamptonroads.com/2008/08/university-virginia-students-accused-plagiarism-expelled-program&usg=ALkJrhiuKQ8pm48hIKOvod0UeAdZCA16Jw

[3] GssLatino, Copiar del internet o de wikipedia, ¿es un delito?, [En línea], 11-01-2012, http://dailyuw.com/news/2007/may/21/internet-leads-to-increased-plagiarism/

[4] Plagium, Sitio de detección de plagio utilizando API's de búsqueda, [En línea], 11-01-2012, http://www.plagium.com/

[5] Wikipedia, Utilidad diff, [En línea], 11-01-2012, http://en.wikipedia.org/wiki/Diff

[6] Google Custom Search API Json/Atom, Página principal de la documentación del API de búsqueda de Google, [En línea], 11-01-2012, http://code.google.com/intl/es-ES/apis/customsearch/v1/overview.html

[7] Yahoo BOSS API, Página principal de la documentación del API de búsqueda de Yahoo, [En línea], 11-01-2012, http://developer.yahoo.com/search/boss/boss_api_guide/

[8] Bing API, Página principal de la documentación del API de búsqueda de Bing, [En línea], 11-01-2012, http://msdn.microsoft.com/en-us/library/dd251056.aspx

[9] Google-diff-match-patch, Proyecto open source de implementación del algoritmo diff, [En línea], 11-01-2012, http://code.google.com/p/google-diff-match-patch/

[10] Jurriaan Hage, Peter Rademaker and Nike van Vugt , A comparison of plagiarism detection tools, [En línea], 11-01-2012, http://www.cs.uu.nl/research/techreps/repo/CS-2010/2010-015.pdf