UNIVERSIDAD POLITÉCNICA SALESIANA

CARRERA DE INGENIERÍA DE SISTEMAS

Tesis previa a la obtención del Título de: Ingeniero de Sistemas

TÍTULO:

# DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA GENÉRICA PARA DESARROLLAR Y PROBAR NUEVAS TÉCNICAS DE DETECCIÓN DE PLAGIO EN TEXTOS.

AUTORES: Manuel Fernando Barrera Maura
Nestor Hernán Fajardo Heras

DIRECTOR: Ing. Vladimir Robles Bykbaev

Cuenca, Septiembre del 2014

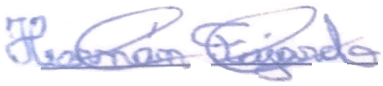
DECLARATORIA DE RESPONSABILIDAD:

Los conceptos desarrollados, los análisis realizados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

A traves de la presente declaracion cedemos los derechos de propiedad intelectual correspondiente a este trabajo a la Universidad Politecnica Salesiana, segun lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

*Cuenca, septiembre del 2014*

Manuel Barrera



Hernán Fajardo

CERTIFICACIÓN:

Ing. Vladimir Robles Bykbaev

**Certifica:**

Haber dirigido y revisado prolijamente cada uno de los capítulos del informe de monografía realizada por los señores Manuel Fernando Barrera Maura y Nestor Hernán Fajardo Heras

*Cuenca, septiembre del 2014*

Ing. Vladimir Robles Bykbaev

# DEDICATORIA

Esta tesis se la dedico a mis padres, Alejandrina Heras y Ernesto Fajardo, quienes fueron quienes me apoyaron incondicionalmente para llegar a esta instancia de mis estudios, adicionalmente dedico este trabajo a mis compañeros y amigos que brindaron su ayuda en el desarrollo de este trabajo: Manuel Barrera, Belizario Ochoa y Elizabeth Andrade, gracias por su apoyo.

Hernán

# DEDICATORIA

Gracias a esas personas importantes en mi vida, que siempre estuvieron listas para brindarme toda su ayuda, especialmente a mis padre y hermano que siempre me brindaron su apoyo, a mi amigo Hernan el cual me brindo su ayuda en todo momento, y finalmente a mis compañeros de la universidad a los cuales nunca olvidare.

Manuel

## RESUMEN

En este trabajo se presenta un modelo de plataforma de software para desarrollar y evaluar los algoritmos de detección de plagio. La plataforma se basa en un diseño modular escalable, que implementa un conjunto de servicios que posibiliten realizar automáticamente tareas como: análisis sintáctico y semántico a través de WordNet y Freeling, extracción automática de texto de múltiples formatos de archivos (PDF, Word y texto), extracción de contenido de páginas web (el uso de algunos motores de búsqueda como Google, Yandex, Yahoo, Bing), el almacenamiento, la carga y el uso de algoritmos de detección de plagio. Estos servicios permiten a un programador desarrollar el código centrando el esfuerzo en el diseño del algoritmo y la base matemática/estadística. Actualmente, la plataforma se probó usando varias consultas de texto (n-gramas), y los resultados de rendimiento son prometedores.

# AGRADECIMIENTOS

Agradecemos de manera muy especial al Ing. Vladimir Robles y al Ing. Cristian Timbi por todo el apoyo que nos ha brindado para que este trabajo se lleve a cabo.

Manuel y Hernán

# ÍNDICE GENERAL

# ÍNDICE DE FIGURAS



# ÍNDICE DE CUADROS

# CAPÍTULO I

# INTRODUCCIÓN AL PLAGIO Y REVISIÓN DE LAS TÉCNICAS DE DETECCIÓN

# 1

# INTRODUCCIÓN AL PLAGIO Y REVISIÓN DE LAS TÉCNICAS DE DETECCIÓN

## 1.1 INTRODUCCIÓN

### 1.1.1 *Definición de Plagio*

En la actualidad existen muchas definiciones sobre lo que es plagio, la mayoría de autores lo definen básicamente como una copia de ideas, pensamientos u obras y presentarlas e incluso publicarlas como propias, pero este trabajo se apegará más a la definición de la IEEE sobre plagio, que es el siguiente: "plagiar es reusar las ideas, procesos, resultados o palabras de alguien más sin mencionar explícitamente la fuente y su autor." [21] esta definición es la más apropiada para este trabajo debido a que para las técnicas de detección que se implementarán un factor importante será el hecho de referenciar las fuentes bibliográficas en los trabajos académicos. Es importante aclarar que no en todos los casos el hecho de tomar textos y parafrasearlos o reusarlos sin referencia se los puede considerar plagio, siempre y cuando sean de dominio general[16], como en el uso de fechas o acontecimientos públicos, por ejemplo: En un texto se puede tener lo siguiente "La batalla de Pichincha ocurrió el 24 de mayo de 1822" de lo cual muchos otros textos pueden contener una frase similar, entonces en este tipo de casos no se puede referenciar a un autor que haya tenido la idea original, pero en el caso de que se tratase de una opinión o interpretación de estos hechos como por ejemplo recitaciones, editoriales, entre otros similares, en los que intervienen ideas propias de los autores, estos deben ser referenciados

### 1.1.2 *Casos de Plagio*

A continuación mencionaremos dos casos de plagio, el primero se trata del periodista Fareed Zakaria el cual trabajaba en el Time y CNN el cual admitió haber plagiado algunos párrafos de un ensayo de la profesora Lepore, para ponerlos en su artículo. Como consecuencia del plagio cometido por parte del periodista a este le suspendieron su programa de televisión en CCN y su columna en el periódico de Time fue suspendida por un mes [13].

El segundo caso se trata acerca del plagio de un artículo de la Dra. Gabriela Piriz Álvarez, el cual fue publicado en la revista médica Uruguay el 20 de marzo del 2004, dos años después partes de este artículo, como la introducción, 16 párrafos y 2 bibliografías, aparecieron publicadas en un artículo de internet, este artículo pertenecía al departamento de salud del gobierno de navarra. La Dra. Gabriela Piriz les hizo saber al departamento de salud del gobierno de navarra que uno de sus artículos contenía plagio, pero lo único que hizo la institución fue cambiar algunas frases, pero el plagio aún continuaba, cabe destacar que el artículo jamás fue retirado [28].

Según una encuesta realizada por la ATL (Association of Teachers and Lecturers) en el 2008 a profesores de escuelas de Gran Bretaña, indicó que el 58 % de los profesores consideran el plagio como un

problema serio, y el 28 % de estos docentes indicaron que al menos el 50 % de los trabajos entregados contenían plagio de Internet, incluso afirmaron que algunos trabajos llegaban con anuncios de las páginas web [29], de lo que se puede deducir que en este tipo de casos los estudiantes no se tomaron el tiempo para leer el contenido del trabajo que presentaron, además podemos decir que estos casos son ejemplos claros del síndrome de copy-paste planteado por Hermann Maurer y Narayanan Kulathuramaiyer que indican que el acceso al amplio contenido de información en la Web son un factor que degrada la calidad de los trabajos científicos [17]. De acuerdo a esta misma encuesta la ATL indicó que más del 50 % de los profesores afirmaron que los estudiantes no tienen comprensión de lo que es el plagio [29].

## 1.2 PRINCIPALES TIPOS DE PLAGIO

El plagio tiene una gran diversidad de clasificaciones, que pueden incluir diferentes áreas o tipos de obras, por ejemplo plagio en obras musicales, obras literarias, imágenes, etc. pero en este trabajo se procurará centrarse en los principales tipos de plagio en textos y se detallan a continuación.

### 1.2.1 *Plagio de Autoría:*

Este implica que una persona presenta un trabajo que fue realizado enteramente por otro como suyo, este tipo de casos ocurre en el mundo académico cuando un estudiante presenta un trabajo en el que él no participó en el desarrollo [1], se puede decir que este no es un tipo de plagio en el que es factible realizar una detección automática.

### 1.2.2 *Plagio de Ideas*

El plagio de ideas se da cuando se adopta las ideas, pensamientos o teorías de otras personas o autores sin darles el crédito por ser quienes las desarrollaron [1], en estos casos desde la perspectiva de la ética es apropiarse de manera inadecuada de lo que otro ha desarrollado, ha investigado o comprendido [5]. Este es uno de los tipos de plagio más complicados de detectar ya que una misma idea se puede expresar de diferentes maneras, lo que no es fácil de detectar con el procesamiento automático del lenguaje [1]. Es importante decir que en este tipo de casos se debe considerar si las ideas son o no de dominio general para poder determinar si realmente existe plagio [16].

### 1.2.3 *Plagio Accidental*

Suelen darse casos en los que se reutiliza información adquirida de diferentes fuentes, y no se da el crédito a las personas correctas, pero de una manera no intencionada, suelen ser principalmente por desconocimiento de la correcta forma de realizar referencias o por desconocer de manera global lo que implica el plagio; pero a pesar de ser no intencionado, se está cayendo en un problema de ética [16].

### 1.2.4 *Plagio Literal (palabra por palabra)*

Este tipo de plagio es relativamente el más fácil de detectar, consiste en tomar un texto y presentarlo sin realizar ningún cambio, es el caso de copiar y pegar planteado por Maurer y Kulathuramaiyer [17], para este tipo de plagio muchas veces solo basta con ingresar parte del texto en un buscador web, entonces podremos obtener directamente las fuentes de donde posiblemente fue plagiado.

### 1.2.5 *Plagio Traducido*

Este tipo de plagio consiste en traducir información de fuentes en otros idiomas y darlos como propios, este tipo de casos se da en su mayoría cuando no existe suficiente información de un tema en el idioma en el que se presenta el documento plagiado, por lo cual se recurre a información en otro idioma, normalmente suele ser del idioma ingles debido a que la mayor cantidad de información contenida en la web esta en este lenguaje, suele usarse herramientas de traducción automáticas para este tipo de plagio [1].

### 1.2.6 *Plagio por Paráfrasis*

Este tipo de plagio consiste en tomar textos, ya sea frases o párrafos y cambiarle la estructura sintáctica manteniendo el mismo significado sin usar referencias para darlas como propias [1], dentro de este tipo se pueden encontrar otros subtipos o técnicas de plagio que son:

*Paráfrasis por sinonimia*

Este tipo de plagio consiste en reemplazar palabras del texto original por otras equivalentes, es decir por sinónimos, esto altera la sintaxis del texto, pero mantiene la misma semántica.

*Ejemplo:*

Texto original tomado de Wikipedia:
*Uno de los servicios que más éxito ha tenido en Internet es la World Wide Web*
Texto plagiado:
*Una de las herramientas que más triunfo ha tenido en Internet es la World Wide Web*

*Paráfrasis por antonimia*

Este tipo de plagio consiste en reemplazar palabras representativas del texto original por sus antónimos, pero implica un cambio de orden en la estructura para mantener el mismo significado, usualmente suele ser usado en comparaciones para modificar la sintaxis de las oraciones [1]

*Ejemplo.*

Texto original:
*El idioma ingles es más utilizado en internet que el español.*
Texto plagiado:
*El idioma español es menos utilizado en internet que el inglés.*

*Paráfrasis por hiperónimos*

Este tipo de plagio consiste en reemplazar palabras representativas del texto original por otras palabras más genéricas con las que se puede referir a la palabra o palabras originales, pasa de lo específico a lo general [1]

*Ejemplo:*

Texto original:
*Los automóviles y motocicletas en la avenida están atrapadas por el tráfico.*
Texto plagiado:
*Los vehículos en la avenida están atrapados por el tráfico.*

*Paráfrasis por cambio de orden*

En este tipo de plagio suele cambiarse el orden de la oración es decir se la divide en dos o más partes y se intercambia las posiciones, puede ser el intercambio en el orden del sujeto y el predicado [1]

*Ejemplo:*

Texto original:
*Los automóviles y motocicletas en la avenida están atrapados por el tráfico.*
Texto plagiado:
*En la avenida están atrapados los automóviles y motocicletas por el tráfico.*

*Paráfrasis con uso de definiciones*

Consiste en reemplazar palabras representativas del texto original por su correspondiente definición, de esta manera se mantiene el mismo sentido pero se altera la estructura del texto [1]

*Ejemplo:*

Texto original:
*La teoría del Big Bang es un modelo científico que trata de explicar el origen del Universo*
Texto plagiado:
*La teoría de una fuerte explosión a inicios del tiempo es un modelo científico que trata de explicar el origen del Universo.*

*Paráfrasis por eliminación.*

Consiste en alterar la estructura del texto eliminando palabras menos representativas del texto, pero que no alteren el sentido que tenía el texto original [1].

*Ejemplo:*

Texto original:
*La teoría del Big Bang es un modelo científico que trata de explicar el origen del Universo*
Texto plagiado:
*La teoría del Big Bang es un modelo que trata de explicar el origen del Universo*

*Paráfrasis por agregación.*

Consiste en agregar palabras de menor importancia al texto original, con la finalidad de alterar la estructura pero sin alterar el significado [1].

*Ejemplo:*

Texto original:
*La teoría del Big Bang es un modelo científico que trata de explicar el origen del Universo.*
Texto plagiado:
*La conocida teoría del Big Bang es un modelo científico que trata de explicar el origen del Universo.*

## 1.3 ANÁLISIS DE LAS TÉCNICAS DE DETECCIÓN DE PLAGIO TEXTUAL.

En la actualidad se han desarrollado diferentes tipos de técnicas para la detección de plagio en textos, unas con mejores resultados que otras, se puede decir que estas técnicas realizan su análisis desde diferentes enfoques, entre los que están [7]:

- Detección de plagio Intrínseco.
- Detección de plagio con comparación a fuentes externas.
- Detección de plagio translingue

A continuación se detallan algunas de las técnicas más importantes de acuerdo a estos enfoques.

### 1.3.1 *Detección de plagio Intrínseco*

Este tipo de detección se lo realiza utilizando características del estilo de escritura, obtenidas a partir del mismo documento, para este tipo de análisis no es necesario disponer de fuentes externas con las cuales comparar el texto, pero conlleva una mayor complejidad en los algoritmos para detectar plagio como por ejemplo el uso de inteligencia artificial, en este tipo de técnicas de detección de plagio, no se identifica las posibles fuentes, sino más bien se determina la probabilidad de que el documento contenga información plagiada [7],[15]

*Técnica de detección por estilo del autor*

Este tipo de técnica intenta detectar el plagio en un documento, en función del estilo de escritura del autor, es decir, busca partes de texto dentro del documento que se sospecha que tiene plagio, que no tengan el mismo estilo de escritura del autor. El gran problema de esta técnica es que es necesario contar con el estilo de escritura del autor del documento. Esta técnica es muy utilizada cuando realizamos una detección intrínseca, es decir no contamos con fuente externas, para comparar con el documento sospechoso de plagio, la búsqueda de partes sospechosas se realiza a partir del mismo documento [15]. El uso de redes neuronales artificiales puede ser una herramienta adecuada para poder reconocer el estilo de escritura de un determinado autor, esto se logra cuantificando determinados aspectos del estilo del autor,

como por ejemplo el número de signos de puntuación, palabras más usadas, errores gramaticales que suele cometer, etc.

*Técnica de detección de cambios de complejidad.*

Este tipo de técnica consiste en detectar cambios en la complejidad del texto dentro de un fragmento del documento, esto se logra a través de comparar el estilo de escritura de dicho fragmento con el estilo de escritura del resto del documento, cuando existe un cambio brusco que sobrepasa un umbral tolerado, se considera que ese fragmento es una posible inserción de información de fuentes externas, de no estar citado, se considera un posible plagio [7].

### 1.3.2 *Detección de plagio con comparación a fuentes externas.*

Este tipo de técnicas conllevan el tener disponibilidad de acceso a las posibles fuentes de donde pudo existir plagio, este enfoque permite identificar de manera clara las fuentes y fragmentos plagiados, si es que existiese, pero realizando un análisis comparativo entre el documento sospechoso y dichas fuentes [7]. Es importante aclarar que las fuentes externas pueden ser de diferentes tipos, esto incluye documentos publicados en la web e incluso puede ser un corpus de trabajos de diferentes estudiantes para determinar si existe copia entre estos.

*Técnica vectorial*

Este tipo de técnica utiliza medidas de similitud, así como también el peso de frecuencia del término, esto quiere decir que se le da un mayor valor a las palabras que más se repiten en el documento, también toma en cuenta las palabras menos comunes de cada documento, mientras más se repitan ese tipos de palabras, las posibilidades de plagio aumentan. Las técnicas de similitud más usadas es la de coseno, la cual consiste en medir qué tan parecido es un documento al otro. Existen otras técnicas de similitud como la del producto interno, la cual suma las palabras parecidas en los dos documentos, mientras más palabras similares existen en ambos documentos, la probabilidad de plagio es mayor [39].

*Técnica de n-gramas*

Esta técnica divide al documento en trozos con una cierta granularidad o también llamada longitud, dependiendo del tamaño que tengan estos, la probabilidad de detectar el plagio puede variar, es decir si los trozos de texto son muy pequeños las probabilidades de encontrar plagio aumenta, debido principalmente a que al ser pedazos de texto muy pequeños existe más posibilidad que estos se repitan en otro documento, ocurre lo contrario si los trozos de texto son muy grandes, aquí la probabilidad de encontrar plagio disminuye, debido a que con solo cambiar un par de palabras en el texto este puede parecer diferente al original [39].

*Técnica de la huella digital*

Esta técnica al igual que la de n-gramas divide al documento en porciones de texto, pero con la gran diferencia que esta utiliza una función hash para transformar esa porción de texto en números, estos números

llegan a ser la huella digital de la porción de texto que representan. Mediante la comparación de las diferentes huellas digitales de varios documentos se puede verificar si existe plagio entre los documentos [25].

### 1.3.3 *Técnicas de detección de plagio translingue*

*CL-ASA*

Es el método que utiliza un diccionario para detectar plagio en otros idiomas, esta técnica consiste primero en traducir los documentos en un idioma en común, para luego comparar los dos documentos [36], este método tiene un grave problema y es que consume muchos recursos debido a que se realiza dos procesos, primero traducir luego comparar. Para poder aplicar este método se necesita normalizar ambos textos, a un idioma en común [36]. Existen varias formas de normalizar entre las cuales están.

*Normalización por peso de traducción*

Esta técnica aplica la siguiente fórmula:
$p(x,y) = \frac{w(x,y)}{w(x)}$ [17] en donde:
x = es una palabra en un idioma
y= es una palabra en un idioma diferente de la palabra x
w(x,y)= peso de la traducción
w(x)= peso de traducción posibles.

*Normalización de la similitud*

Esta técnica aplica la siguiente fórmula:
Snorm(d | d')=$\frac{S(d,d')}{|d|}$[17] en donde:
d= es un documento
| d | = número total de palabras del documento d
d'= un documento en un idioma diferente al documento d

*CL-CNG*

Detección de plagio utilizando n-gramas, este método es muy recomendable cuando se intentan comparar dos lenguajes que tienen similitudes sintácticas [26], utiliza trigramas de caracteres (CL-C3G).
La formula aplicada en este método es la siguiente:
S(d,d')= $\frac{(DxD')}{(|d|x|d'|')}$ [21] en donde:
D y D' son proyecciones vectoriales y
d y d' son n-gramas de carácter

*CL-ESA*

Este método utiliza un corpus multilingüe comparable, el cual tiene que estar alineado por tema e idioma, el corpus más utilizado es la enciclopedia Wikipedia [26]
Para detectar la similitud entre los documentos se aplica la siguiente formula:
D={sim(d,c) V c∈CI} [21] en donde:
sim(d,c) = calcula las similitud entre los documentos d y c.
CI = es una colección de documentos.

## 1.4 ESTUDIO DE LAS HERRAMIENTAS DE DETECCIÓN DE PLAGIO EXISTENTES

Hasta el día de hoy se ha desarrollado una gran gama de herramientas de software útiles para detectar plagio en textos y en otros tipos de trabajos, pero debemos aclarar que no existe ninguna que sea 100 % efectiva ya que en muchos casos la complejidad de los textos o las mismas técnicas de plagio existentes y usadas causan una confusión en el análisis que estas efectúan [37], es por ello se debe tener presente que los sistemas o herramientas de software, orientadas a esta área, nunca tendrán la palabra final al momento de tomar acciones contra quien ha cometido plagio [1], siempre será responsabilidad de quienes usan las herramientas, en el caso del área académica de los docentes, el asegurarse por estos y otros medios si existe o no plagio. Se puede decir que el objetivo de todas estas herramientas es facilitar la tarea de analizar si existen casos de plagio y ayudar a detectar las posibles fuentes de donde pudo ocurrir dicho plagio.

### 1.4.1 *Tipos de Herramientas*

En la actualidad existen diferentes tipos de herramientas desarrolladas para detectar plagio, las cuales podemos clasificar principalmente en dos tipos:

- Herramientas Online.
- Herramientas Locales (de escritorio).

Cabe recalcar que dentro de este tipo de herramientas también podemos clasificarlas en herramientas de libre distribución, de código abierto, y privativas.

### 1.4.2 *Beneficios de las herramientas existentes*

En general las herramientas de software para detección de plagio desarrolladas hasta el día de hoy ofrecen varios beneficios entre ellos están:

- Permiten un análisis de grandes cantidades de documentos de manera rápida.
- Algunas herramientas permiten un análisis automático del estilo de escritura en base a técnicas de inteligencia artificial u otras similares, lo que potencia su efectividad [7].
- Permiten identificar los fragmentos plagiados tanto en el documento original como en el sospechoso [7].
- Existe un amplio soporte a diferentes formatos de documentos de texto, esto incluye documentos en Word, PDF, etc.
- Estas herramientas permiten a los educadores ahorrar tiempo en la revisión de casos de plagio, y concentrarse en evaluar la calidad del contenido de los trabajos [38].
- Desde la perspectiva de los estudiantes estas herramientas pueden ser usadas para mejorar la escritura en lo referente al uso correcto de referencias, ya que detectará si existen fuentes no citadas.

### 1.4.3 *Debilidades de las herramientas existentes*

Entre las debilidades de las herramientas de software para detección de plagio se tienen las siguientes:

- Ninguna de las herramientas existentes son 100 % precisas en la detección de plagio por el hecho de la complejidad de las técnicas de plagio [37].
- Pocas herramientas disponen de una detección eficiente de plagio traducido.
- Es difícil escoger una herramienta, ya que de acuerdo a estudios realizados a varias de estas herramientas, con un único documento, los resultados varían entre el 20 % al 40 % [37].

### 1.4.4 *Descripción de algunas herramientas existentes*

*DOCODE (Document Copy Detection)*

Esta herramienta fue desarrollada por académicos de la Universidad de Chile, el líder de este proyecto fue Juan Velásquez, investigador perteneciente al área de Ingeniería Industrial, quien trabajó con un equipo multidisciplinario especializado en psicología, lingüística, entre otros [4]. Docode ha participado en diferentes concursos de sistemas detectores de plagio como el PAN y ha obtenido importantes premios, como el del 2001, donde ganó el mundial en sistemas de detección de plagio y obtuvo una buena posición en el concurso PAN [8]. Actualmente dispone de tres versiones para su uso desde la web, dependiendo de la cantidad de documentos que se desee analizar, puede ser gratuito o pagado [8].

*Turnitin*

Es un sistema de detección de plagio en línea cuyo servicio es pagado, fue desarrollado en Estados Unidos por iParadigms. Cuenta con soporte para documentos en español en formato PDF, provee servicios para detectar plagio tanto en trabajos académicos a universidades y escuelas como también a editoriales, periódicos, etc. De acuerdo a su página oficial, alrededor de 800 mil educadores usan esta herramienta [38, 37].

Este sistema dispone de una gran base de datos de documentos archivados, publicaciones y libros, además tiene acceso a millones de páginas web que pueden ser rastreadas como posibles fuentes [38]. Asimismo, cabe destacar que está orientado a facilitar a los educadores la revisión de trabajos académicos, brinda funcionalidades para comentar los trabajos, revisar las referencias y detectar el porcentaje de probabilidad de plagio [38].

*Viper*

Esta herramienta fue creada por la compañía de servicios educativos All Answers Limited y permite ingresar diferentes tipos de documentos, entre los cuales están Word, PDF, Power Point, etc. Asimismo, la herramienta realizará un escaneo en la web para encontrar documentos que tengan similitudes con el ingresado. Una de las características que posee, es que se puede escanear el mismo documento cuantas veces se

desee y cuando encuentre que el documento que se ingresó concuerda con un ensayo de otro estudiante, indicará qué porcentaje de similitud tienen. Otro aspecto a destacar, es que nunca se muestra el ensayo con el cual nuestro documento concuerda. Viper cuenta con más de 14 billones de páginas web, 2 millones de artículos de estudiantes y con miles de publicaciones de libros, editoriales y revistas para realizar las comparaciones, según la página oficial posee más de 2 millones de usuarios y es totalmente gratuita [40].

*WCopyFind*

WCopyFind fue desarrollado en el año 2004 en la Universidad de Virginia, la técnica que utiliza para descubrir plagio es la basada en n-gramas, por lo cual el documento que se va a analizar puede estar el cualquier idioma. Una de las grandes desventajas de esta herramienta es que una vez se realiza una alteración al documento, como por ejemplo, cambios de palabras por sus sinónimos, antónimos, hiperónimos e hipónimos, su rendimiento en la detección de plagio es relativamente bajo. Otro problema es que a pesar que puede analizar documentos en diferentes idiomas no detecta plagio multilingüe.WCopyFind cuenta con una licencia Open Source [12].

*Plagtracker*

Este es un detector de plagio web de pago, por lo que se requiere una cuenta premiun para acceder a todos sus servicios y recibir reportes detallados. Cuenta con acceso a más de 14 billones páginas web, además de 5 millones de artículos académicos de bases de datos de universidades. En su versión gratuita permite ingresar texto en un editor para verificar su originalidad, su página principal brinda accesibilidad con diferentes idiomas incluyendo inglés y español [33].

*Plagiarism Checker*

Esta es una herramienta web hasta cierto nivel libre, permite ingresar el texto en un editor para buscar plagio, su versión premiun permite realizar un análisis más profundo, subir archivos directamente y algunas otras funcionalidades más, esta herramienta es parte de un conjunto pertenecientes a dustball que ofrece diferentes tipos de servicios tanto para estudiantes, docentes y personas en general [3].

*PlagScan*

Esta es una herramienta web pagada, no cuenta con una versión trial libre, ofrece servicios tanto a usuarios individuales como a organizaciones, este sistema cuenta con premios tanto a la facilidad de uso como al rendimiento en lo referente a detección de plagio, ofrece soporte para diferentes formatos de documentos, entre ellos .doc, txt, html, etc. [32].

*Grammarly Plagiarism Checker*

Es un detector de plagio web que se encuentra en inglés su funcionamiento es relativamente sencillo, lo único lo que se hace es pegar el texto que se sospecha tiene plagio en el recuadro de la página principal, este detector de plagio no acepta documentos, es decir solo se puede pegar pedazos de texto, aparte de detectar plagio este detector corrige las

faltas de ortografía, puntuación y algunos parámetros más. El informe que presenta es simple, solo informa si tiene plagio o no, si tiene faltas de ortografía, signos de puntuación etc. Para poder recibir un informe completo es necesario registrarse debido a que esta herramienta es pagada. Plagiarism Check no puede detectar plagio que este demasiado parafraseado, ya que cuando se le realizo pruebas con un texto que tenía plagio de este tipo, esta herramienta no lo detecto, tampoco tiene soporte para detectar plagio translingue. Se puede decir que es una herramienta fácil de manejar [31].

*Plagiarism detector*

Este es un detector de plagio web, esta herramienta tiene soporte para diferentes tipos de texto como doc, docx, odt y txt, también se puede pegar trozos de texto para verificar si tiene plagio, el informe que presenta es un poco más detallado que el que presentaba la herramienta anteriormente mencionada. Cuando detecta plagio esta herramienta muestra la parte del texto que esta plagiado así como la fuente de donde se presume se plagio, también presenta el porcentaje del texto que tiene plagio. Esta herramienta es sencilla de manejar y totalmente gratuita [6].

*Dupli Checker*

Es un detector de plagio web, tiene soporte solo para dos tipos de documentos .doc y .txt, también se puede pegar trozos de texto para verificar si tiene plagio, cuando esta herramienta detecta plagio en un documento, nos muestra la fuente y las pates del texto que probablemente se copiaron, su modo de uso es fácil, se sube o se pega el texto que se sospecha tiene plagio, esta herramienta es totalmente gratuita [10].

### 1.4.5 *Comparación entre herramientas*

Se realizó una comparación de precisión de resultados de algunas herramientas anteriormente descritas, cabe recalcar que solo se comparó las herramientas que eran de uso gratuito o tenían versiones triales, por lo que algunas no estaban a su máximo rendimiento.

*Análisis comparativo del corpus*

Se utilizó cuatro documentos, dos con plagio y dos sin plagio para medir el nivel de precisión de las herramientas cuyos resultados fueron los siguientes:

| Herramienta | Documento | Fuentes | % plagio real | Fuentes Detectadas | % plagio detectado | Precisión |
|---|---|---|---|---|---|---|
| http://plagiarism-detect.com | Ps4 vs X box one | 1 | 40 | 0 | 0 | 0 |
| http://www.grammarly.com | Ps4 vs X box one | 1 | 40 | 0 | 0 | 0 |
| http://www.duplichecker.com | Ps4 vs X box one | 1 | 40 | 0 | 0 | 0 |
| http://www.plagtracker.com | Ps4 vs X box one | 1 | 40 | 0 | 0 | 0 |
| http://plagiarism-detect.com | evolucion de las consolas | 0 | 0 | 5 | 2 | 98 |
| http://www.grammarly.com | evolucion de las consolas | 0 | 0 | 0 | 0 | 100 |
| http://www.duplichecker.com | evolucion de las consolas | 0 | 0 | 0 | 0 | 100 |
| http://www.plagtracker.com | evolucion de las consolas | 0 | 0 | 3 | 6 | 94 |
| http://plagiarism-detect.com | Que es Facebook | 1 | 90 | 10 | 33 | 43 |
| http://www.grammarly.com | Que es Facebook | 1 | 90 | 0 | 100 | 10 |
| http://www.duplichecker.com | Que es Facebook | 1 | 90 | 88 | 0 | 10 |
| http://www.plagtracker.com | Que es Facebook | 1 | 90 | 27 | 80 | 90 |
| http://plagiarism-detect.com | Reporte sobre Facebook | 3 | 0 | 0 | 0 | 100 |
| http://www.grammarly.com | Reporte sobre Facebook | 3 | 0 | 0 | 0 | 100 |
| http://www.duplichecker.com | Reporte sobre Facebook | 3 | 0 | 0 | 0 | 100 |
| http://www.plagtracker.com | Reporte sobre Facebook | 3 | 0 | 0 | 0 | 100 |

Cuadro 1: Análisis comparativo del corpus

# CAPÍTULO II

# ANÁLISIS Y DISEÑO DE LA PLATAFORMA DE DETECCIÓN DE PLAGIO

# 2

# ANÁLISIS Y DISEÑO DE LA PLATAFORMA DE DETECCIÓN DE PLAGIO

En este capítulo se definirán los requerimientos tanto de funcionamiento del módulo de comunicación como los de la plataforma genérica como tal. Asimismo, se presenta el diseño del sistema y se estudiarán alternativas para las herramientas de soporte de la plataforma

## 2.1 ANÁLISIS DE REQUERIMIENTOS PARA LA PLATAFORMA.

En esta sección se presentan los principales requerimientos que deberá cubrir plataforma, se han clasificado en 5 subcategorías que se indican a continuación:

- Acceso a la Web
- Funcionalidad de análisis léxico semántico
- Carga y lectura de documentos
- Gestión de algoritmos y servicios
- Acceso a la plataforma (Interfaz Gráfica de Usuario)

### 2.1.1 *Requerimientos de acceso a la Web*

- Capacidad de conexión con diferentes buscadores web:
  - Google
  - Yahoo
  - Bing
- Extraer metadatos de documentos buscados en la web: Esto se realiza en virtud de que se requiere tener un mejor filtrado de la información que se recolecta de internet, al obtener los metadatos, tales como keywords, temáticas tratadas en el documento, etc., se puede validar de mejor manera las posibles fuentes, relacionando los metadatos con el tema que se está buscando en la web.
- Almacenar las búsquedas o los documentos encontrados en el disco duro: Este aspecto tiene la finalidad de mejorar la velocidad de búsqueda en internet, ya que se pretende manejar una caché con las búsquedas realizadas para posteriormente evitar el alto consumo de los debido al acceso a la web. Las opciones que se plantean para el manejo de dicha caché es almacenar los enlaces resultantes de determinadas búsquedas o almacenar directamente los documentos web de dichos enlaces para no descargarlos nuevamente.
- Capacidad de evitar bloqueos por múltiples consultas en las búsquedas: En trabajos anteriores se ha podido constatar que uno de los principales problemas que existe es el bloqueo de los motores de búsqueda cuando se efectúan varias consultas

continuas [2]. Dicho problema se produce principalmente en el acceso a Google y para solucionar esta dificultad se plantea el análisis de la factibilidad de implementación de las siguientes alternativas:

- Intentar evitar el balanceador de carga de Google, accediendo directamente a las IPs de los servidores de búsqueda, se debe analizar si esta alternativa es viable y evita la detección de las consultas continuas. Asimismo, se debe comprobar si no se infringe ninguna normativa.
- Reducir el número de consultas que se realizará en un determinado periodo de tiempo, utilizando delays (tiempos de retardo o pausas) entre cada solicitud al motor de búsqueda. Con ello, se debe analizar el impacto que tendrán dichos retardos en la velocidad de respuesta de la plataforma.
- Acceder al motor de búsqueda utilizando saltos a proxys, es decir, alternar el acceso al motor de búsqueda a través de diferentes servidores proxy que brindan su servicio en la web, de esta manera se consigue un enmascaramiento de la IP de origen, evitando el bloqueo, se debe analizar la velocidad de respuesta en las búsquedas y la factibilidad de consumir dicho servicio de proxy. Asimismo, se debe comprobar si no se infringe ninguna normativa.

### 2.1.2 *Requerimientos de funcionalidad de analisis lexico semantico*

Estos requerimientos se refieren a las funcionalidades de procesamiento de texto que deberá tener la plataforma, entre los importantes están:

- Soporte de funciones de análisis léxico.
  - Eliminación de stop words
  - Obtener las palabras más relevantes (palabras con mayor carga semántica)
- Acceso a un diccionario de:
  - Sinónimos
  - Antónimos
  - Significados
  - Traducciones
- Brindar funciones de separación de textos por palabras, frases y párrafos.

### 2.1.3 *Requerimientos de carga de documentos*

Este tipo de requerimientos se refiere a la lectura de documentos que serán analizados por la plataforma, para ello deben cubrir las siguientes funcionalidades.

- Soporte de documentos en los formatos comúnmente usados, tales como: PDF, DOC, DOCX y texto plano.

- Debe brindar una interfaz entre la plataforma y los repositorios de documentos, es decir, provee el servicio de extracción de texto de dichos documentos, para así poder manipularlo en los módulos de análisis, una alternativa para esto es el uso de APIs ya implementadas.
- De ser factible, el sistema debe reconocer de manera automática el formato de archivo que se va a leer, la manera más fácil para esto es el reconocimiento de patrones de las extensiones del archivo.

### 2.1.4 *Requerimientos de gestión de algoritmos y servicios*

- Los algoritmos deben funcionar como una librería, es decir, que se los pueda invocar desde una capa superior.
- Permite cargar nuevos algoritmos en tiempo de ejecución, es decir, la plataforma no deberá detenerse para subir nuevos algoritmos, lo que se quiere evitar es que la plataforma se detenga cada vez que un usuario quiera subir un nuevo algoritmo, ya que esto puede ocasionar molestias a los otros usuarios del sistema, La solución que se plantea inicialmente a este requerimiento es la carga dinámica de librerías, lo que permitirá incorporar los algoritmos como un archivo ejecutable posterior al arranque del sistema.
- Determinar un porcentaje de probabilidad de plagio en los documentos analizados. La plataforma debería ser capaz de devolver un nivel de confidencia del porcentaje de plagio del documento que se está analizando, así como las fuentes de donde probablemente se sacó el texto tomado sin referencias. Esto se lo hará mediante algoritmos de detección de plagio, en este caso se programarán dos algoritmos.
- Se requiere generar reportes de los resultados de los análisis, los cuales podrán ser presentados en un formato de documento o en la misma GUI (Interfaz Gráfica de Usuario, por sus siglas en inglés) en la que esté implementado.

### 2.1.5 *Requerimientos para el acceso a la plataforma y GUI*

- La plataforma debe permitir trabajar con múltiples usuarios concurrentes, es decir, se pretende que la plataforma pueda ofrecer sus servicios a varios usuarios a la vez, para lo cual se busca que la plataforma funcione en un entorno web.
- Adicionalmente, la plataforma debe poder ejecutarse en un entorno de escritorio, con la finalidad de su uso en pruebas, de ahí que es importante que sea independiente de la GUI por la que se accede y administra.

## 2.2 DEFINICIÓN DE MÓDULOS Y COMPONENTES PRINCIPALES

En base a los requerimientos planteados, se ha optado por dividir la plataforma de detección en los 5 módulos, los cuales serán desarrollados de manera independiente, con el objetivo de que ningún módulo dependa de otro para su funcionamiento, excepto uno central, que se

encargara de integrar todos estos módulos y brindar el servicio a los algoritmos.

*Módulo de conexión a Internet y web crawling:*

Este módulo cubrirá todos los requerimientos de acceso a la web anteriormente planteados, como se mencionó anteriormente se tiene planeado que este módulo pueda buscar y descargar contenido de los tres principales buscadores bing, yahoo y google, sólo se pretende descargar el contenido de los primeros resultados de la búsqueda, ya que se los consideran los más relevantes.

Para realizar las búsquedas se descargara las páginas de resultados de una búsqueda para con un posterior tratamiento, ya sea con el uso de librerías, expresiones regulares y/u otras técnicas extraer los enlaces de resultados.

Se pretende que el contenido que se encuentran en páginas html sea descargado de manera textual, es decir el contenido y no el código, mientras que los contenidos que se encuentran en formatos como PDF o Word, sean guardados como archivos para luego extraer su contenido.

Adicionalmente el módulo tendrá un archivo de configuración en el cual se podrá cambiar los parámetros de funcionamiento, entre las cuales están, el uso de conexion por proxy, definir con qué buscadores se trabajará etc.

*Módulo de Análisis Léxico y Semántico:*

Este módulo cubrirá las funcionalidades de análisis léxico, para lo cual se pretende usar otros sistemas y librerías adicionales enfocados a esa área, como por ejemplo freeling y bases de datos léxicas, la cual al combinarse y además implementando funciones propias brindará un completo analizador léxico.

Para las funcionalidades de acceso a diccionarios, se crearán estructuras que representen los elementos necesarios tales como: palabras, párrafos, sinónimos, traducciones, etc.

Este módulo se encarga de integrar el analizador léxico y la base de datos léxica y demás funcionalidades, para que brinde una interfaz a las demás capas o módulos que acceden a los servicios de análisis.

*Módulo de extracción de información textual:*

Este módulo debe encargarse de realizar la carga de documentos en diferentes formatos, aquí se pretende utilizar diferentes librerías para la obtención de texto en diferentes formatos, los formatos a los cuales se dará soporte son, pdf, doc, docx y txt, de preferencia las librerías deberán ser gratuitas, este módulo seleccionará de manera automática la librería a usar reconociendo el formato de archivo por su extensión.

El módulo devolverá el contenido del documento y sus metadatos, el único parámetro que recibirá es la ubicación del documento con el cual se pretende trabajar.

*Módulo de administración y gestión automatizada de servicios y algoritmos(Integrador):*

De manera general este módulo se encargará de integrar todos los módulos y los servicios que requieren los algoritmos, además permitirá administrarlos y ejecutarlos, se encargara de realizar la búsqueda de un determinado algoritmo que se desee utilizar, los cuales están registrados en una base de datos, a la cual se conectará de manera previa a la carga de dicho algoritmo, para obtener los metadatos tales como ruta donde esta almacenado, entre otros.

La interfaz gráfica que se implemente deberá conectarse a este módulo central, los parámetros que recibe este módulo desde la GUI son:

- La ruta en donde se encuentre el texto sospechoso.
- El tipo de análisis que realizará, (contra archivos locales o con una búsqueda en internet)
- Si desea realizar una comparación local, deberá enviar la ruta de los archivos que son posibles fuentes.
- El nombre del algoritmo con el cual se va a trabajar.

Por factores de rendimiento este módulo deberá cargar todos los servicios y algoritmos disponibles al iniciar el sistema, permitiendo tenerlos cargados en memoria al momento de una petición de análisis.

Para la gestión de algoritmos se requiere que estos mantengan una estructura definida en una interfaz por la cual el módulo realiza la carga y ejecución, dicha estructura será disponer de los siguientes métodos:

- Establecer texto de posibles fuentes: se le pasa un arreglo con todos lo textos que son considerados posibles fuentes.
- Establecer texto sospechoso: se le pasa el texto a analizar.
- Ejecutar: Se realiza el proceso de comparación, aquí se deberá implementar toda la lógica propia del algoritmo.
- Devolver resultados: Deberá retornar una estructura con los resultados parciales y totales de análisis sobre los textos detección.

*Módulo de generación de reportes*

Este módulo será el encargado de generar y enviar los informes, los cuales serán el resultado del análisis realizado por un algoritmo detección, se generará un documento de resumen del análisis, a partir de los resultados devueltos por la ejecución de un algoritmo, se pretende que dicho informe contenga lo siguiente

- Porcentaje de plagio del documento sospechoso.
- Posibles fuentes del plagio.

Estos informes serán enviados a al correo del usuario que solicitó el análisis, el cual será ingresado en la interfaz de usuario.

Este módulo está orientado principalmente al servicio de la plataforma en línea, es decir cuando está desplegado en la web y se tenga acceso concurrente por múltiples usuarios, para ello se plantea que cada solicitud se agregara en una cola y al realizarse el análisis y se obtenga los resultados, estos llegaran de manera automática al usuario.

*Módulo de acceso concurrente*

Este módulo será el encargado de exponer el servicio de detección de plagio a través de un acceso web, es decir se dará acceso a múltiples usuarios, se encargara de manejar las solicitudes de análisis y de devolver los resultados a una interfaz gráfica web y a través de correo electrónico.

## 2.3 DISEÑO DE LA PLATAFORMA

A continuación se presenta el diseño modular inicial de la plataforma de detección de plagio.

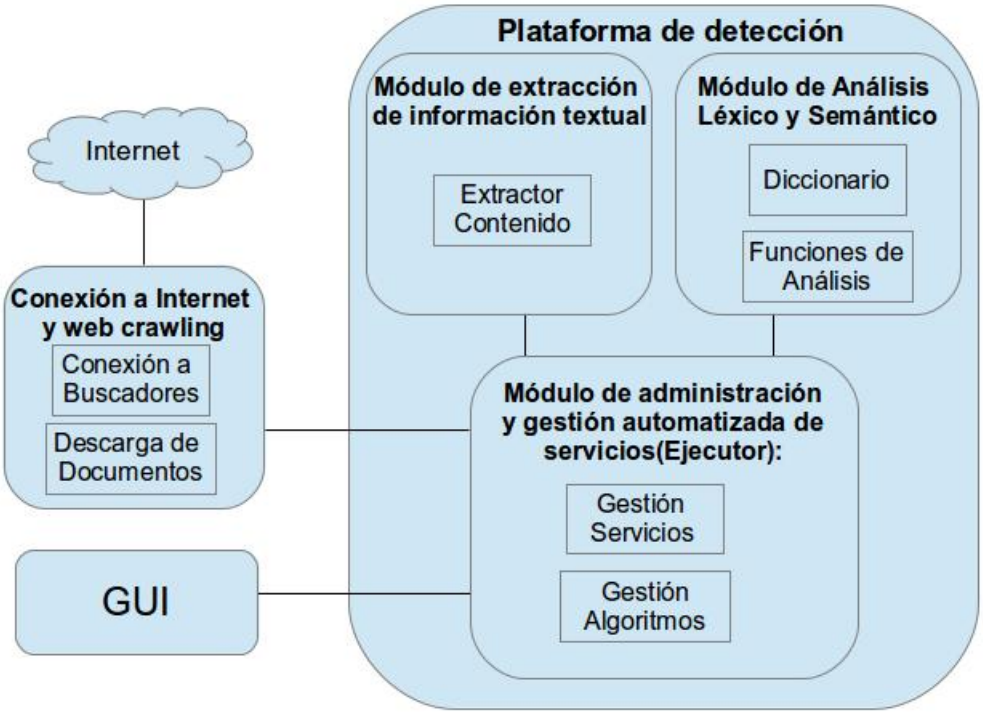



Figura 1: Diseño modular inicial de la plataforma

El diagrama anterior permite apreciar la funcionalidad básica que deberá tener el sistema, es decir permite ver la separación entre la plataforma, la conexión a la web y la interfaz de usuario. Este diseño hará que cada módulo pueda trabajar independientemente del otro, lo que conlleva a que la plataforma pueda funcionar con internet o localmente.

Este tipo de diseño también permite que el sistema sea escalable, es decir si en un futuro se necesite agregar otro módulo se lo pueda hacer fácilmente sin alterar el funcionamiento del resto de la plataforma.

Solo en caso de que la plataforma requiera analizar un documento contra fuentes de internet, accede de manera automática al modulo de "Conexion a internet y web crawling".

Los algoritmos se almacenarán en un repositorio y serán registrados en una base de datos que contendrá todos metadatos de estos, así podrá realizarse una carga dinámica incluso cuando está en ejecución.

Un factor a tomar en cuenta para el rendimiento es la posibilidad de mantener una cache de búsquedas o documentos descargados, para

evitar el continuo acceso a la web mejorando el tiempo de respuesta, este planteamiento se lo hace tomando en consideración que quienes lo usarán en primera instancia serán docentes, que por lo general analizan varios documentos del mismo tema, lo que aumenta la probabilidad de tener búsquedas repetitivas.

## 2.4 SELECCIÓN DE HERRAMIENTAS Y API'S DE SOPORTE.

En esta sección se definen las principales herramientas de software que serán usadas para el desarrollo del proyecto entre ellas están las siguientes:

### 2.4.1 *Selección del Lenguaje de Programación:*

Para este proyecto se utilizara un lenguaje de programación orientado a objetos para facilitar la modularidad del sistema, dicho lenguaje es JAVA:

JAVA: Es un lenguaje de programación con tecnología capaz de dar soporte a programación orientada a objetos, su principal ventaja es el hecho de ser multiplataforma, para el desarrollo en esta plataforma se requeriré el JDK (Java Development Kit) el cual cuenta con herramientas especiales como por ejemplo el javac que es el que compila los programas, mientras que para los usuarios finales se requiere el JRE (Java Runtime Enviroment) que es el que permite ejecutar las aplicaciones [23].

### 2.4.2 *IDE (Entorno de desarrollo integrado)*

Ya que se ha optado por el lenguaje de programación JAVA se ha elegido como IDE principal de desarrollo a NetBeans, por las facilidades que brinda para programar en dicho lenguaje.

Adicionalmente se usará Eclipse como un IDE alternativo para el desarrollo de componentes como la interfaz Web, entre otros.

### 2.4.3 *Analizador Lingüístico*

El analizador lingüístico nos permitirá realizar un análisis de tipo sintáctico de textos, lo que será de utilidad para los diferentes algoritmos de detección de plagio, el que se ha seleccionado es freeling por ser libre y además es considerado como uno de los mejores existentes. Freeeling: Este es un conjunto de herramienta de análisis de lenguaje, es software libre que se distribuye bajo licencia GNU, este software fue creado y hasta el día de hoy es liderado por Lluís Padró en la Universidad Politécnica de Catalunya [14].

Este software permite el análisis y procesamiento de lenguaje, tiene soporte para diferentes idiomas y ofrece diferentes funcionalidades, este ofrece una interfaz vía línea de comandos y un API lo que facilita la integración con el software en desarrollo [14].

### 2.4.4 *Diccionario de sinónimos y antónimos*

Este diccionario es básicamente una base de datos de palabras con sus respectivos sinónimos y antónimos, en base a la tesis previa a este proyecto se utilizará WordNet.

WordNet Esta es una base de datos que originalmente contiene información en inglés, la cual para futuros trabajos podría ser implementada para un análisis multilingüe, pero existe una variante en español conocida como EuroWordNet que es la que será utilizada en este proyecto [43]

El esquema de la base de datos léxica es el siguiente:

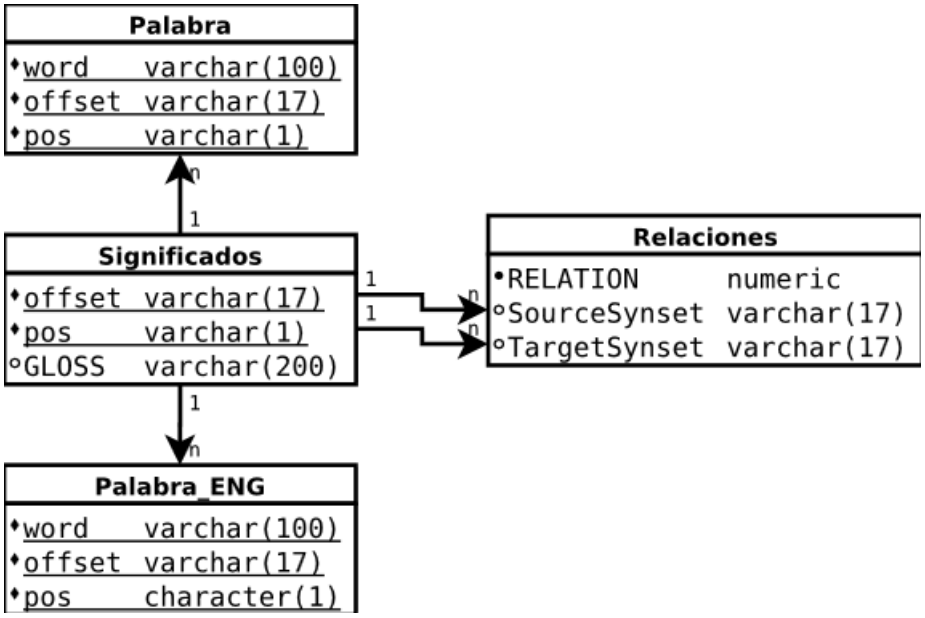



Figura 2: Esquema de la base de datos léxica

### 2.4.5 *APIs de Conexión a Buscadores*

*HTML Parser:*

Esta librería se utilizará en el módulo de conexión y web crawling, para extraer el texto de las páginas html. Html parser es una biblioteca para java que permite manipular documentos html, principalmente para la extracción y transformación de estos documentos [20].

### 2.4.6 *APIs de lectura de documentos*

Se utilizarán diferentes APIs para la lectura de documentos ya que cada uno da soporte a diferentes formatos entre ellos están:

POI: Es un api desarrollado por Apache, el cual permite la manipulación de diferentes formatos de documentos entre los cuales están Excel, Word, PowerPoint, OpenXML entre otros. Este api permite obtener los metadatos de un documento, como título y autor, además de leer y escribirlos en los diferentes documentos soportados [34].

Docx4j: Es un api para java que permite la manipulación de documentos Microsoft open XML, entre los formatos soportados están docx, pptx, xlsx. Este api fue desarrollado por Plutext Pty Ltd en 2008 [9].

PDFBox: Es una librería que permite la manipulación de documentos tipo pdf, este api permite la creación, lectura y obtención del contenido de este tipo de documentos, esta herramienta es desarrollada por Apache [30].

iText: Es una biblioteca para java que permite la manipulación de documentos pdf, tienes algunos funciones entre las cuales están, generar documentos dinámicos a partir de archivos XML o base de datos, agregar firmas digitales a documentos pdf, cortar y manipular documentos pdf, entre otras funciones [22].

### 2.4.7 *Gestores de Base de datos*

Los gestores de bases de datos seleccionadas para trabajar son, para la etapa de desarrollo e utilizara una base de datos embebida, por su facilidad de implementación y portabilidad, la cual es HSQL, mientras que para la etapa de despliegue se utilizará un gestor de base de datos postgres, por motivos de disponibilidad, rendimiento y licenciamiento.

HSQLDB es una base de datos hecha enteramente el lenguaje java, esta puede ser utilizada de dos formas, la primera es arrancar en un servidor de base de datos y la otra manera es que esté integrada directamente a una aplicación, en donde no es necesario conectarse a un servidor de base de datos [19]. El lenguaje que maneja esta base de datos es el SQL estándar y es totalmente gratuita.

PostgreSQL "es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales" [35].

### 2.4.8 *Servidor de aplicaciones*

*JBoss*

La elección como el servidor de aplicaciones fue JBoos, debido a su compatibilidad con java, disponibilidad para puesta en producción de la plataforma, posibilidad de ejecutarse en diferentes entornos de sistemas operativos [24]

## 2.5 PREPARACIÓN DEL CORPUS DE PRUEBAS Y DISEÑO DEL PLAN DE EXPERIMENTACIÓN.

Las pruebas que se pretenden realizar al sistema son las siguientes

### 2.5.1 *Pruebas del módulo de conexión a Internet*

En esta sección se analizará el número de consultas que se puede realizar de manera continua a los diferentes buscadores en un periodo de tiempo. Además se analizará el tiempo de respuesta en relación al ancho de banda disponible y la longitud de los textos buscados, para

ello se pretende utilizar un texto de ejemplo y se extraerán N gramas para buscarlos en la web utilizando el módulo en cuestión.

### 2.5.2 *Pruebas del módulo de analisis lexico y semantico*

En estas pruebas se genera un reporte con los resultados que retornen las funcionalidades principales del modulo de analisis lexico y semantico, dicho reporte posteriormente será interpretado para validar los resultados.

### 2.5.3 *Pruebas de algoritmos de detección de plagio*

Se pretende desarrollar de uno a dos algoritmos para probar la funcionalidad de la plataforma, adicionalmente se prevé comparar los resultados que retorne dicho algoritmo con los del estado del arte.

CAPÍTULO III

# IMPLEMENTACIÓN DE LA PLATAFORMA DE DETECCIÓN DE PLAGIO Y SELECCIÓN DE ALGORITMOS BASE

# 3

# IMPLEMENTACIÓN DE LA PLATAFORMA DE DETECCIÓN DE PLAGIO

En este capítulo se detalla el desarrollo y diseño final del sistema con sus respectivos módulos, se ha implementado utilizando tecnologías y paradigmas de programación orientada a objetos para obtener un alto nivel de modularidad. Entre los elementos más importantes utilizados de la programación orientada a objetos tenemos.

- Interfaces

Son clases que contienen métodos abstractos, aquí se especifica lo que se debe implementar pero no como hacerlo, la clase que implemente las interfaces obligatoriamente debe también implementar todo los métodos de la interfaz[27]

- Clases Abstractas:

Su funcionamiento es similar a las interfaces y la gran diferencia entre las clases abstractas y las interfaces en que en las interfaces todos sus métodos deben ser abstractos, mientras que en abstractas sólo es necesario que solo uno lo sea[27]

- Herencia:

La herencia consiste en que un clase pueda acceder a métodos y atributos, de otra clase la cual se la suele llamar clase padre y tratarlos como si fueran suyos, evitando que se programe código ya implementado[27]

- Clasificadores de Alcance (estático):

Los métodos y datos estáticos , son aquellos que no están asociados a una instancia sino a una clase, lo que permite, la característica de estos es que no existe una copia de los mismo para cada objeto, si no es el mismo para toda la clase, lo que permite que siempre ocupe un solo lugar en la memoria[27]. Adicionalmente para alcanzar un alto nivel de modularidad se ha utilizado un servicio propio de JAVA que permite cargar código en tiempo de ejecución

- ClassLoaders:

El classloader forma parte del java Runtime Environment y cuyo objetivo es el de localizar bibliotecas, leer sus contenidos y cargar clases que se encuentren dentro de las bibliotecas, normalmente la bibliotecas se encuentran en archivos tipo JAR. Cabe destacar que la carga de clases desde una biblioteca es un proceso que conlleva mayor complejidad de implementación, en este caso se han definido interfaces con los servicios que deben cumplir las librerías a cargar. [41]

El desarrollo de la plataforma se lo realizó en 2 estándares de Java, el núcleo en Java SE y el acceso concurrente en Java EE, es por ello que el núcleo inicialmente no está orientado a ser de alta concurrencia, sino a implementar la lógica necesaria para la ejecución de algoritmos de detección de plagio de manera independiente de la interfaz por la que se decida acceder, sea Web o Escritorio, para el accesos concurrente a través de la web se usa un servidor de aplicaciones, en este caso JBOSS, que maneja mediante colas y servicios web el acceso a la plataforma, es decir permite el acceso concurrente a una sola instancia de la plataforma.

## 3.1 IMPLEMENTACIÓN DEL MÓDULO DE COMUNICACIÓN.

El el módulo de comunicación o conexión a Internet se divide en dos partes:

- Descarga de contenido de páginas y documentos de la web
  - Descarga de archivos.
  - Extracción del contenido textual de documentos web.
- Conexión a motores de Búsqueda
  - Descarga de la páginas de resultados
  - Obtención de links de resultados

Trabaja de manera independiente y transparente, los parámetros que recibe y retorna son:

- Para las búsquedas recibe el texto que se desea enviar a los motores de búsqueda y retorna un arreglo con los enlaces de resultados.
- Para obtener el contenido textual sólo recibe una cadena de texto con el enlace de donde se desea obtener, retorna una cadena con el contenido textual de la página o documento del enlace.

### 3.1.1 *Descarga de contenido de páginas y documentos de la web*

Su funcionalidad radica en 2 partes:

- Descarga de archivos.
- Extracción del contenido textual de documentos web.

*Descarga de archivos*

Esto permite obtener el código html de una página para realizar un post procesamiento u obtener los archivos que no sean html como documentos PDF,word, etc.

Para la descarga de páginas se usa la clase URL de java, la cual nos servirá para abrir la conexión a la página web y así acceder al contenido de esta, se obtiene un stream del contenido y se lo descarga.

*Conexiones HTTP y HTTPS:*

En el módulo tiene dos métodos para descargar las páginas dependiendo del protocolo que utilicen las misma, debido a que existen páginas que utilizan protocolos http o https, dependiendo del enlace que reciba, identifica el método que debe utilizar para realizar la descarga.

*Extracción del contenido textual de documentos web*

Esta parte se encarga de extraer el contenido de las posibles fuentes a partir de los links obtenidos con los motores de búsqueda, su funcionalidad permite retornar el contenido en una cadena de texto, sin importar el tipo de documento web al que esté accediendo, ya sea html, pdf, word, etc. interactúa con los enlaces de los resultados que se obtienen desde los motores de búsqueda, para esto se usa una librería conocida

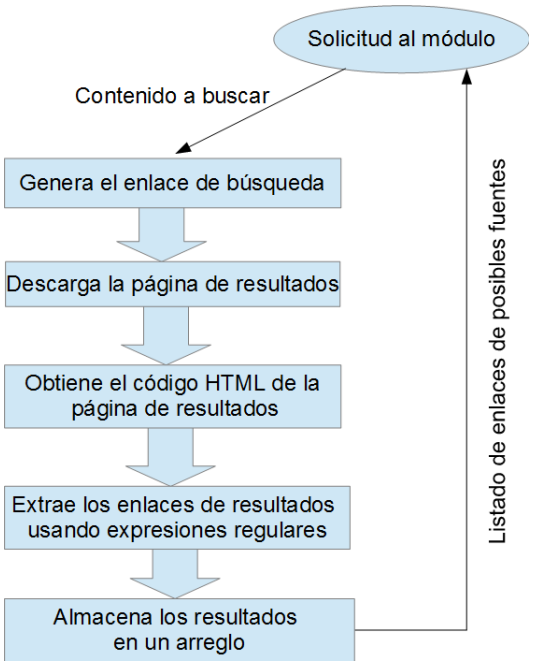



Figura 3: Secuencia del proceso que se realiza con cada motor de búsqueda

como HTML Parser en caso de ser una pagina web, y en caso de ser un documento como PDF, Word o TXT, se descarga el documento como un archivo temporal y se reutiliza las funciones implementadas en el módulo de carga para leer el contenido.

### 3.1.2 *Conexión a motores de Búsqueda*

Esta parte se encarga de realizar las búsquedas en la web, implementa métodos para poder realizar un mayor número de consultas y extraer los links de las posibles fuentes. El método utilizado para obtener un mayor número de consultas fue mediante el uso de web proxys. De manera general el proceso de búsqueda se lo realiza de la siguiente forma: El módulo recibe un parámetro de texto con el contenido que se desea buscar en la web, este debe verificar los motores de búsqueda que se encuentran configurados como activos y si se procede a descargar el contenido html de la página de resultados de dicho buscador para extraer los links de resultados. La 3.1 muestra la secuencia del proceso que se realiza con cada motor de búsqueda:

.

El funcionamiento de este módulo se divide en 2 partes principales:

- Descarga de la páginas de resultados
- Obtención de links de resultado

*Descarga de la paginas de resultados*

Para poder realizar las búsquedas en los diferentes motores, se arma un link, que va estar compuesto de la dirección url del motor de búsqueda y un parámetro adicional, el cual va ha ser lo que se va a mandar a descargar como una página html. Una vez descargada la página con los resultados de la búsqueda, es decir las posibles fuentes, esta pasará al extractor de links segun el motor utilizado para obtener los resultados de la búsqueda obtenidos por cada buscador.

*Optimización del número de consultas*

La forma en la que se evitó el bloqueo por saturacion de consultas fue usando la alternativa de uso de proxys web, debido a que fue la mejor alternativa de las inicialmente propuestas, actualmente se encuentra implementado el salto con un solo servidor web proxy http://webproxy.net obteniendo buenos resultados.

*Web Proxy:*

Son sitios, que permiten cargar paginas pero ocultando los datos de direccionamiento en la red, tales como IP y MAC del usuario final, lo que hace es convertirse en un intermediario entre el cliente y el servidor.

El procedimiento que se usa es alternar las conexiones accediendo directamente desde la misma plataforma al buscador, y accediendo a través de este intermediario, esto evita que el buscador conozca o detecte el número real de consultas que se le está realizando desde la plataforma, al no enviar consultas masivas en cortos periodos de tiempo el sistema no es detectado como un software robot.

*Desventajas del uso de servidores web proxy*

- Dependencia: El hecho de que para acceder a los buscadores se requiera pasar por un intermediario, implica que en caso de que el servidor web proxy falle o se quede fuera de servicio, la plataforma de detección de plagio puede presentar problemas de conectividad.
- Programación adicional:Los enlaces que son de resultados de las búsquedas obtenidos usando el servidor web proxy, tienen un formato diferente al de los obtenidos al acceder directamente al buscador, estos aumentan cierta estructura para que al acceder a ellos se continúen redireccionando a través de dicho servidor proxy, se requiere programar su propia forma de extracción de links que realice un procesamiento adicional para obtener los enlaces reales de los resultados de la búsqueda.
- Altos tiempos de respuesta:Al estar accediendo a través de un intermediario, el tiempo de latencia en la conexión aumenta, esto sacrifica la velocidad de respuesta de toda la plataforma, pero se compensa ya que no se requiere el uso de delays (tiempos de espera) entre una busqueda y otra, además con la implementación de conectividad paralela con mas servidores web proxy, este efecto seria imperceptible, e incluso mejoraria aun mas la velocidad de respuesta.

*Ventajas del uso de servidores web proxy*

- Oculta la dirección real del cliente final: Debido a que al acceder a los motores de búsqueda a través de un web proxy se esta realizando un ocultamiento de identidad del cliente final, se puede realizar un mayor número de consultas continuas sin ser detectado como un software robot.
- Permite un alto número de consultas paralelas: Si se aumenta el número de intermediarios, se puede conseguir el realizar consultas de manera paralela, lo que reduce el tiempo de respuesta de la

plataforma, cabe recalcar que este aspecto es relativo al ancho de banda que se disponga para el acceso a Internet, si no se dispone de una velocidad de acceso considerablemente alta, el uso de paralelismo podría reducir el rendimiento antes que mejorarlo.

*Obtencion de links de resultado*

Para poder extraer los links, cada conexión a un buscador contará con su propia clase para poder hacerlo, incluyendo a las conexiones a través de los web proxy, todas estas clases van a implementar una interfaz la cual es IExtractorLinks que lo único que tendrá serán dos métodos obligatorios

- setContenidoHtml: Recibe un parámetro de tipo string con el contenido fuente de la página de resultados de la búsqueda
- getLinks: Retorna un arreglo de cadenas de texto el cual contendrá todos los enlaces de las posibles fuentes obtenidas con determinado buscador.

La identificación y extracción de los links es a través de expresiones regulares, usando las librerías Regex propio del lenguaje Java. Cabe destacar que para cada buscador se realizará una expresión regular diferente, debido a que cada uno tiene una página de resultados con una estructura diferente, una vez obtenidos los links, estos serán guardados en un arreglo que será retornado como resultado final de una búsqueda.

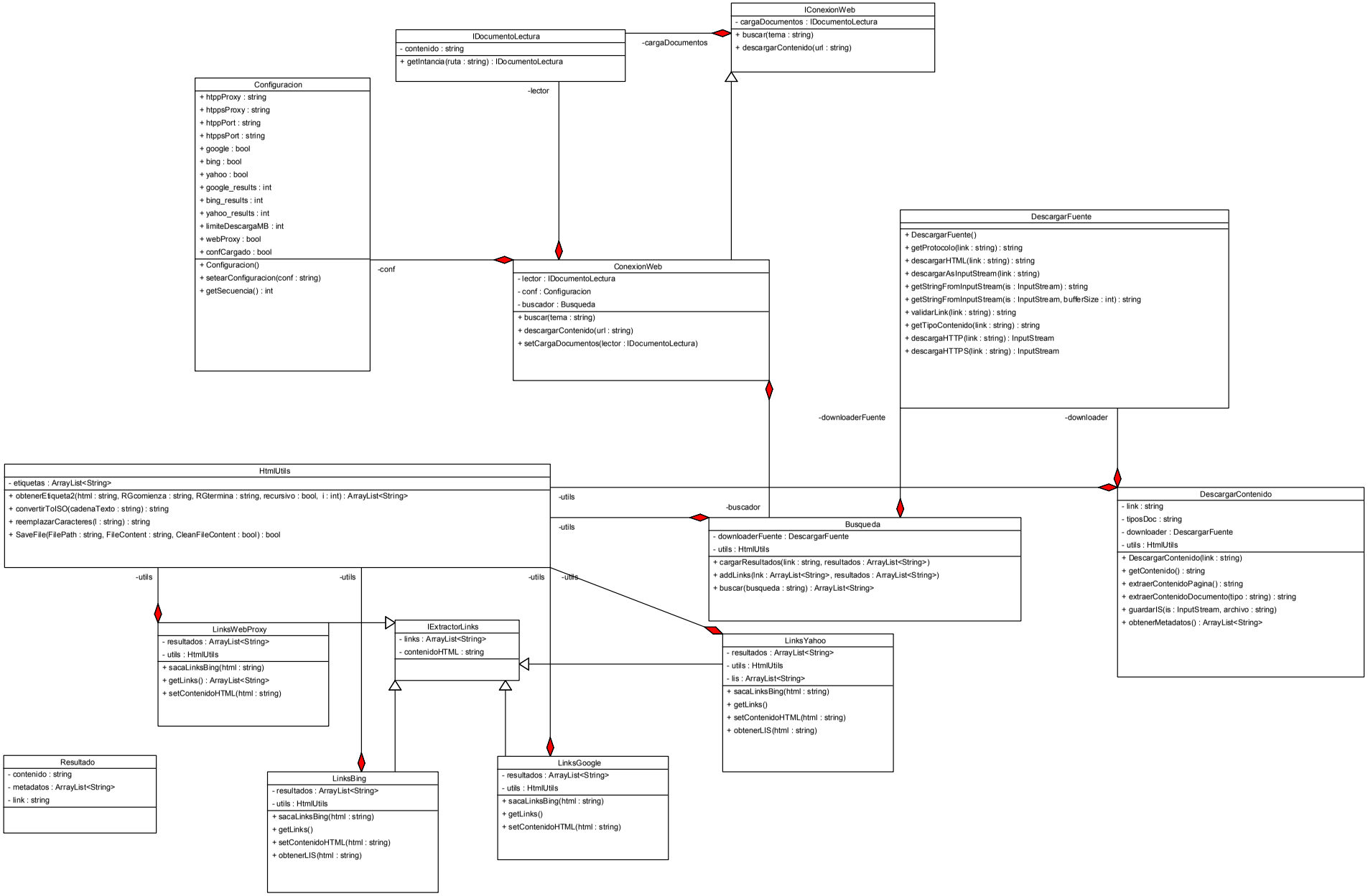



Figura 4: Diagrama de clases del modulo de comunicacion

## 3.2 IMPLEMENTACIÓN DEL MÓDULO DE ADMINISTRACIÓN CENTRAL

Dentro de la plataforma se tiene un módulo integrador, una de sus funcionalidades será ser el encargardo de administrar todos los servicios y parámetros que necesitan los algoritmos, para ello primero se realizará la integración de todos los servicios mediante la utilización de interfaces, las cuales están implementadas en todos los módulos de servicios,

### 3.2.1 *Gestión Automatizada de Servicios*

El módulo maneja los servicios como objetos que pueden ser tratados de 2 formas:

- Como objetos compartidos: Son servicios que se instancian una sola vez a nivel de toda la plataforma, y se comparten declarándose con alcance clasificador (static), dichos servicios son:
  - Servicio de acceso a motores de búsqueda de internet [18]
  - Servicio de análisis léxico.
- Como múltiples instancias: En algunos casos se requiere realizar procesos paralelos como la descarga de documentos de internet, en este caso para cada ejecución de un algoritmo se realiza una nueva instancia [18]. El mismo caso sucede con la lectura de archivos desde disco, básicamente se tienen múltiples instancias de los siguientes servicios:
  - Servicio de descarga de documentos de internet.
  - Servicio de lectura de documentos en disco.

Es importante mencionar que los procesos paralelos a los que se hace referencia se presentan cuando existen múltiples ejecuciones del algoritmo de detección o diferentes algoritmos se da en paralelo.

### 3.2.2 *Gestión automatizada de algoritmos*

Para la administración de algoritmos se refiere a que el módulo será el encargado de buscar, instanciar y ejecutar los algoritmos que el usuario solicita, y se lo hace mediante la utilización de una base de datos en donde estarán registrados todos los algoritmos disponibles en la plataforma, la base de datos utilizada será postgres, y mediante consultas SQL se recupera los metadatos necesarios del algoritmo a utilizar en ese momento, este módulo dispone de dos interfaces para el manejo de los algoritmos.

#### *Tipos de algoritmos*

Para la plataforma se han definido dos tipos de algoritmos, para cada uno de estos existen interfaces que se han definido para que la plataforma pueda interactuar con ellos:

#### *Algoritmos de Comparación*

Son los algoritmos encargados de analizar un documento y comparar la similitud que estos tienen frente a otros documentos considerados

como posibles fuentes, su objetivo es devolver el porcentaje de plagio que tendría dicho documento. Dependerá del algoritmo el porcentaje de plagio que se detecte, ya que no necesariamente los algoritmos devolverán resultados similares.

*Algoritmos de Extracción*

Estos son los encargados de extraer del documento palabras o frases claves las cuales servirán para buscar las posibles fuentes de plagio de este documento en la web, recibe como parámetro el documento sospechoso y se aplicara la lógica para extraer las características más relevantes del texto.

## 3.3 ANÁLISIS Y SELECCIÓN DE ALGORITMOS BASE DE DETECCIÓN DE PLAGIO

Existen diferentes tipos de algoritmos de detección de plagio pero se ha optado por implementar los siguientes:

### 3.3.1 *Vector Space Model:*

Vector Space model es un técnica utilizada comúnmente para la indexación, filtrado y recuperación de información, esta técnica utiliza vectores para representar un documento en lenguaje natural [42]. En la presente tesis utilizaremos la técnica del Vector Space Model junto con la similitud del coseno para comparar dos documentos y ver cuál es el porcentaje de plagio que existe entre estos, el porcentaje de plagio se determinará de la siguiente manera, si al comparar dos documentos el resultado es cercano a cero la probabilidad de plagio es menor, pero si se acerca a uno la probabilidad aumenta.

La forma de utilizar Vector Space model en nuestro caso es de la siguiente manera guardamos todas las palabras del documento en un hashtable, así como también el número de veces que se repite, esto se hará con los dos documentos que queremos comparar, después en el hash se quedaran solo las palabras comunes entre los dos documentos para finalmente aplicar similitud de coseno entre los dos hash y verificar el porcentaje de plagio.

### 3.3.2 *N Gramas*

la técnica de n-grams consiste en representar todo en sentido del documento,pero en un número de caracteres menor a lo que se encuentra en el, esto se logra mediante la agrupación de palabras contiguas del texto, previamente se debe realizar una eliminación de stopwords y signos de puntuación para que la técnica tenga una mayor eficacia[39]. dependiendo del número de palabras que contenga un n-grama probabilidad de plagio puede aumentar o disminuir, en nuestro caso iremos probando n-gramas de diferente tamaño, hasta un máximo de 5 palabras. Para la comparación tanto el documento sospechoso como el original serán divididos en el mismo número de n-gramas para luego ser comparados,cabe destacar que esta técnica no será muy eficiente cuando se aplique un plagio por sinonimia.

## 3.4 IMPLEMENTACIÓN DEL MÓDULO DE DETECCIÓN DE PLAGIO.

Este módulo es la combinación de varios submódulos de servicios, un módulo integrador y un módulo de acceso concurrente (acceso web) será la parte encargada de ejecutar los algoritmos de detección de plagio como tal y brindar la interfaz lógica de acceso al servicio de análisis.

### 3.4.1 *Servicios de la plataforma*

La plataforma brinda los siguientes servicios para los algoritmos de detección:

- Analizador lexico y semantico
- Lectura de documentos
- Adicionalmente se considera al módulo de comunicación detallado en el punto 3.1 como un servicio que brinda la plataforma.

*Analizador Lexico y Semantico:*

Este módulo tiene la funcionalidad de brindar los servicios de análisis de texto a los algoritmos, dispone de varias utilidades con el uso de herramientas adicionales y funciones programadas internamente.

*Estructuras de datos*

Para realizar los procesos de análisis maneja diferentes estructuras de datos, tales como:

- Palabra: Esta estructura permite manejar las palabras como un objeto propio de java, en las instancias de estos objetos se guardan los datos obtenidos con el analizador léxico y de la base de datos léxica, es decir: sinónimos, antónimos, lema, etiqueta eagle, etc.
- Párrafo: Esta estructura permite manejar los párrafos como un objeto propio de java, en las instancias de estos objetos se guardan colecciones de palabras, accede de manera rápida a funciones como eliminar stop words y obtener las palabras más frecuentes dentro de la colección de datos que almacena.
- NGrama: Esta estructura permite manejar a los ngramas como un objeto propio de java, en las instancias de estos objetos se guardan colecciones de parrafos, esta estructura no cuenta con funciones relevantes

*Funciones*

Las funcionalidades de este módulo se pueden clasificar en 3 categorías:

- Acceso transparente al analizador léxico, Freeling.
- Acceso transparente a la base de datos léxica MultiWordNet.
- Funciones de análisis adicionales.

*Acceso transparente al analizador léxico, Freeling.*

Este módulo usa funcionalidades del analizador léxico para realizar procesos tales como:

*Clasificación de tipo de palabras:*

Freeling permite obtener un identificador conocido como etiquetas eagle, las cuales permiten reconocer el tipo de palabra de acuerdo al contexto. Esta funcionalidad permite que los desarrolladores de algoritmos puedan filtrar o realizar operaciones basándose en la prioridad del tipo de palabras que deseen analizar.

*Obtener lema de una palabra:*

Adicionalmente, Freeling permite obtener la palabra raíz de otra, o también conocida como lema, esto es útil ya que existen diferentes variantes de una palabra pero se da la facilidad de que el algoritmo de detección de plagio, pueda realizar una comparación de las palabras raíces.

Los resultados obtenidos por estas funciones, son almacenados en las instancias de la clase Palabra, es decir por cada palabra existente en el texto a analizar, se realiza la carga de la etiqueta eagle y el lema de dicha palabra.

El uso de las funciones de Freeling, es a través de una librería propia de esta herramienta que permite el acceso de manera más transparente desde un programa Java.

*Acceso transparente a la base de datos léxica.*

El análisis con el uso de la base de datos léxica se lo hace por medio de consultas SQL a la misma, por medio de JDBC, esta base de datos es usada como un diccionario de la cual se obtienen para cada palabra del texto información tales como:

- Listado de sinónimos
- Listado de antónimos
- Listado de posibles significados
- Listado de traducciones a inglés

Al igual que los resultados obtenidos por el analizador léxico, los resultados obtenidos desde la base de datos léxica son almacenados en las instancias de la clase Palabra, es decir por cada palabra existente en el texto a analizar, se realiza la carga de los valores de los parámetros mencionados en la lista anterior.

*Funciones de análisis adicionales.*

El módulo dispone de varias funcionalidades de análisis adicionales que se encuentran programadas en una clase llamada AnalisisUtils, dichas funcionalidades se detallan a continuación:

- Eliminación de stop words

| Forma | Lema | Etiqueta |
|---|---|---|
| el | el | TDMSo |
| los | el | TDMPo |
| lo | el | TDCSo |
| la | el | TDFSo |
| las | el | TDFPo |
| al, del | al,del | SPCMS |
| a, ante,bajo,con | a,ante,bajo,con | SPSoo |
| sea,si,ya,que,como | sea,si,ya,que,como | CSoo |
| o,u,y,sino,pero,ni,e | o,u,y,sino,pero,ni,e | CCoo |

Cuadro 2: Etiquetas Eagle de freeling para stop words

Este método recibe como parámetros un array de palabras (párrafo), de las cuales serán filtradas las stop words las cuales son consideradas palabras de menor relevancia, estas deben ser eliminadas para que el procesamiento del algoritmo de detección sea más eficiente.

Las palabras de menor relevancia para los textos son detectadas a través de las etiquetas Eagle, tomando en cuenta que se consideran stop words las palabras con las etiquetas detalladas en el cuadro 2

- Obtener la palabras más frecuentes

El funcionamiento de este método consiste en dado un array de palabras (párrafo o documento), usado un mapeo entre la frecuencia de repetición de una determinada palabra se filtra solo las de mayor repetición, y se devolverá otro array de las palabras con las más frecuentes dentro del array original.

- Separación y fragmentación del texto

Su funcionamiento consiste en que dado un texto, estos métodos devolverán un arreglo donde se encontrarán en cada posición un fragmento del texto original, separado por algún criterio como por ejemplo:

- Párrafos: separa el texto por saltos de carro
- Palabras: Se eliminan todos los espacios dobles y saltos de carro, y se separan por espacios en blanco.
- N gramas: Se obtienen fragmentos del texto de N palabras.

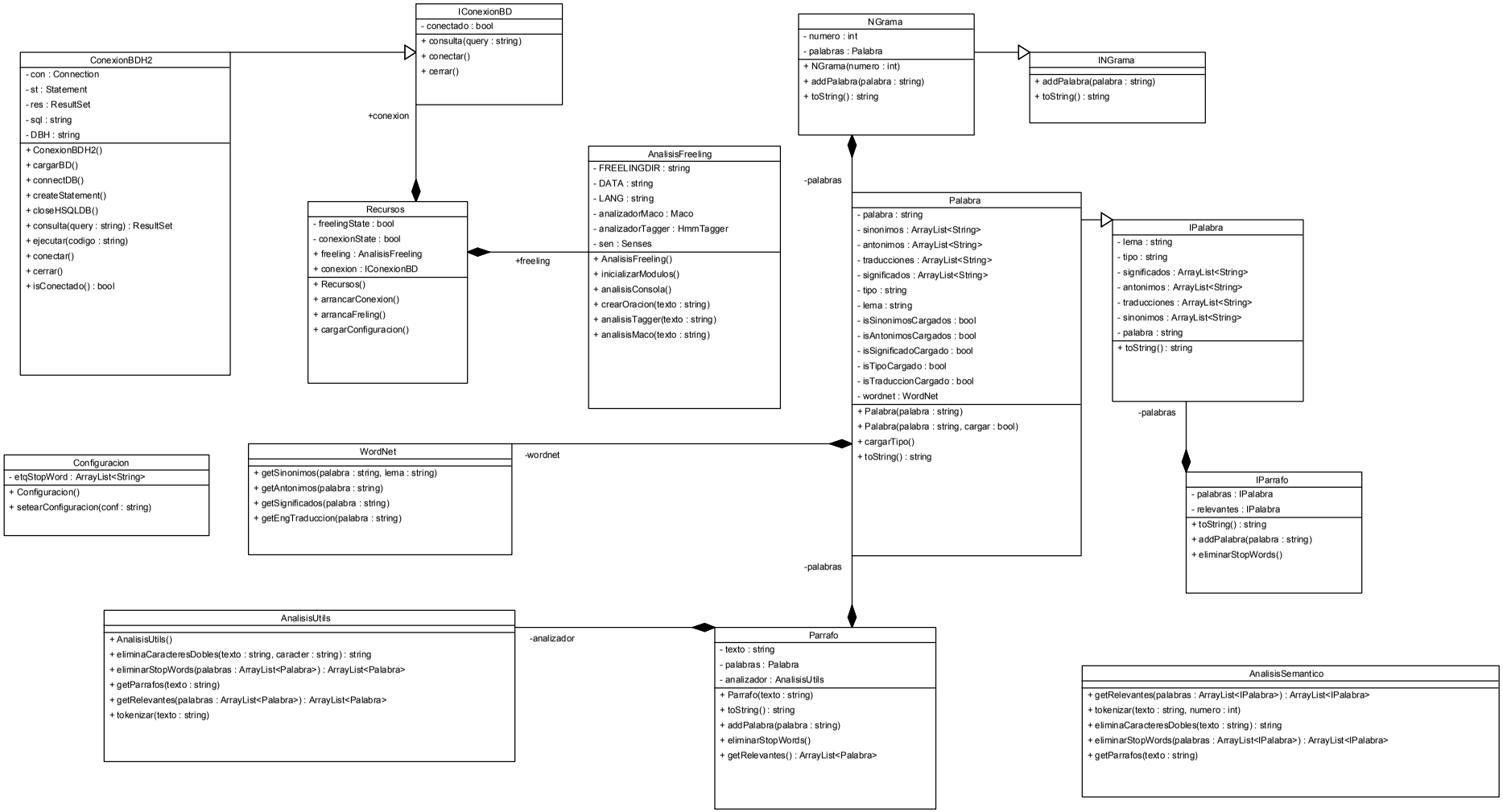



Figura 5: Diagrama de clases de Analizador Lexico y Semantico

*Lectura de documentos*

Este módulo es el encargado de leer los documentos, ya sean los sospechosos o los considerados como posibles fuentes.

*Funcionamiento:*

El proceso de lectura se realiza de manera transparente para las demás capas o módulos, este módulo recibe de parámetro solo la ruta de donde esta almacenado el documento y retorna todo el contenido de texto de dicho documento en un objeto.

Se ha desarrollado una clase independiente para cada tipo de archivo compatible, las cuales implementan una interfaz llamada ILector, cada clase hace uso de las librerías adecuadas para realizar el proceso de lectura, cabe recalcar que el módulo maneja estas clases de lectura usando la interfaz anteriormente mencionada, esto además permite que se puedan integrar otros formatos de archivos compatibles de manera más simple en futuros desarrollos.

El módulo inicialmente reconoce el tipo de archivo que se está intentando leer de acuerdo a la extensión de este para que de acuerdo a esto instanciar el lector adecuado, Adicionalmente cuando el archivo no tiene una extensión o no tiene una compatible, se procede a leer el archivo como si fuera texto plano.

␣

*Lectores por formato*

Actualmente la plataforma es compatible con 3 formatos de archivos, considerados como los más comúnmente usados, los cuales son: Word, PDF y Texto plano

- Lector PDF: Para archivos PDF se ha utilizado la librería pdfbox, debido a que en la pruebas realizadas era la que mejor permitía leer este tipo de documentos, sin dar conflictos con los

documentos que contenían imágenes.

- Lector Word: Este lector maneja 2 formatos, internamente realiza un reconocimiento entre estos:
  - Archivos .doc para el cual se ha utilizado la librería Apache POI, se ha inclinado por esta librería por su popularidad y por ser gratuita.
  - Archivos .docx para el cual se ha utilizado la librería Docx4j principalmente por ser una librería de código libre.
- Lector TXT: Este lector hace uso solo funciones propias de Java, específicamente de las clases en el paquete java.io, lee el contenido y lo retorna

### 3.4.2 *Módulo integrador*

Este módulo es el puente entre el módulo de acceso concurrente, los módulos de servicios y los algoritmos como tal. Sus funcionalidades radican en 2 secciones:

- Gestión automatizada de servicios y algoritmos: esta funcionalidad se detalla en el punto 3.2.1 y 3.2.2 respectivamente

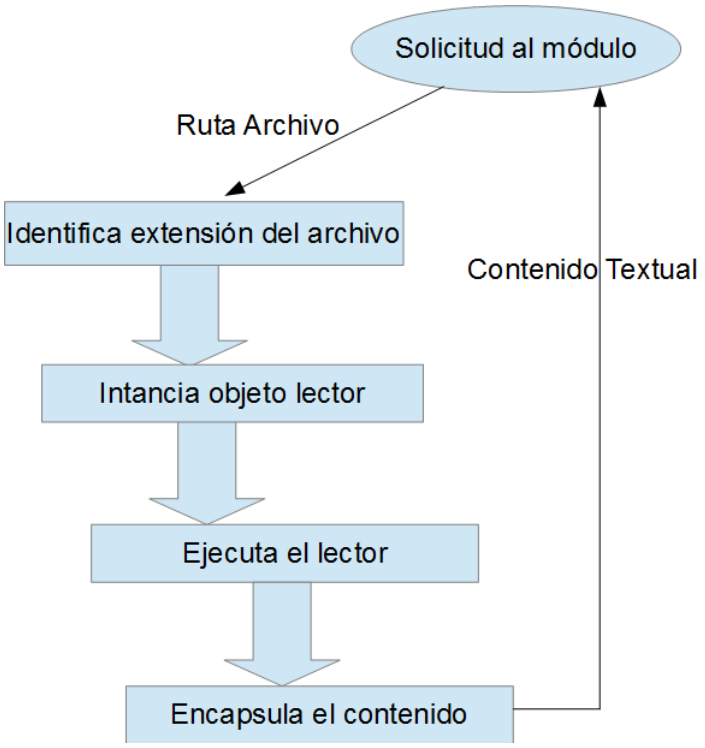



Figura 6: Secuencia del proceso que sigue el lector de documentos

*Diagramas de clases de Carga de documentos*

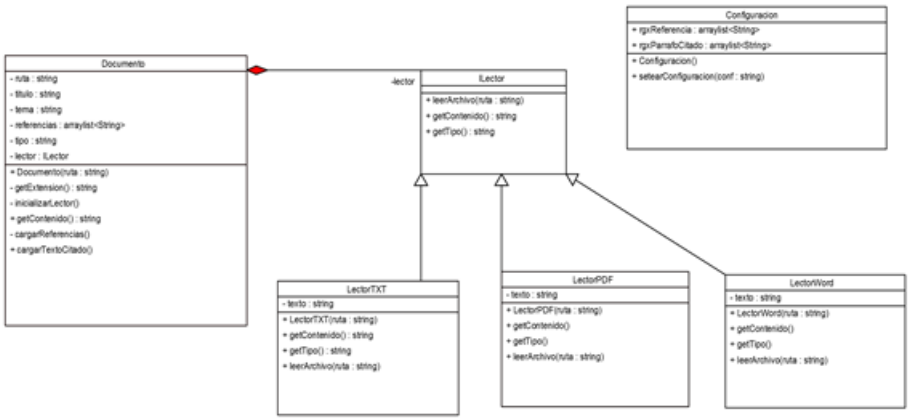



Figura 7: Diagrama de clases de Carga de documentos

- Ejecución de algoritmos

*Ejecución de algoritmos*

Para la ejecución de algoritmos el proceso que se sigue es el siguiente:

- Carga dinámica de un algoritmo como librería
- Inyección de servicios y parámetros al algoritmo
- Ejecución del algoritmo
- Obtención de resultados.

*Carga dinámica de algoritmos como librerías*

Para la carga dinámica de los algoritmos se usa un componente propio del lenguaje JAVA, conocido como cargadores de clase o ClassLoader, que permite realizar una importación de una librería .JAR en tiempo de ejecución, para ello se usan interfaces previamente definidas que los algoritmos deben implementar para que sean compatibles con la plataforma, caso contrario el algoritmo subido a la plataforma no podrá funcionar y sobre todo no será cargado a la misma para su utilización, las interfaces que el algoritmo tiene que implementar, cuentan con métodos específicos entre los principales están:

- Ejecutar
- Obtener resultado final

Adicionalmente el algoritmos tiene que estar dentro de un paquete el cual deberá tener una estructura de nombres específico, que deben ser registrados en la base de datos de los metadatos de los algoritmos, no se puede tener algoritmos con el mismo nombre dentro de una estructura de paquetes similar ya que al momento de cargarlos sólo se considerará el primero que fue cargado.

*Inyección de servicios y parámetros al algoritmo*

Una vez cargados los algoritmos pasan a un contenedor, a partir de ahí cada vez que un cliente solicita un análisis, se genera una nueva instancia de estos a los cuales se les inyecta los servicios y parámetros para ejecutarlos.

Los servicios y parámetros que se le inyectan al algoritmo son:

- La instancia del Analizador Léxico
- El texto sospechoso
- En el caso de ser un algoritmo de comparación y no de extracción de características se le pasa los textos que son posibles fuentes.

Dependiendo si en la solicitud de análisis se solicita una búsqueda en la web, se ejecuta el algoritmo de extracción de características para obtener los parámetros que se buscarán en los motores de búsqueda, una vez hecho esto se obtiene la información desde internet y se lo inyecta al algoritmo de comparación como posibles fuentes.

*Ejecución del algoritmo*

Como se había mencionado con anterioridad el módulo integrador maneja al algoritmo a través de la interfaz correspondiente (Algoritmo de comparación o de extracción) se invoca el método de ejecutar del algoritmo, en esta sección se procederá a ejecutar los proceso propios del algoritmo, en el cual se plasma la lógica de los desarrolladores para realizar estos procesos de comparación o de extracción.

*Obtención de resultados.*

Cuando el algoritmo es ejecutado, dentro de este se debe almacenar los resultados de comparación o de extracción en estructuras definidas, a las cuales la plataforma accedera posteriormente al terminar la ejecución a través de la interfaz, invocando el método obtener resultados del algoritmo; a partir de estos resultados, se generan informes para presentarlos al usuario.

*Diagrama de clases del Módulo Integrador*

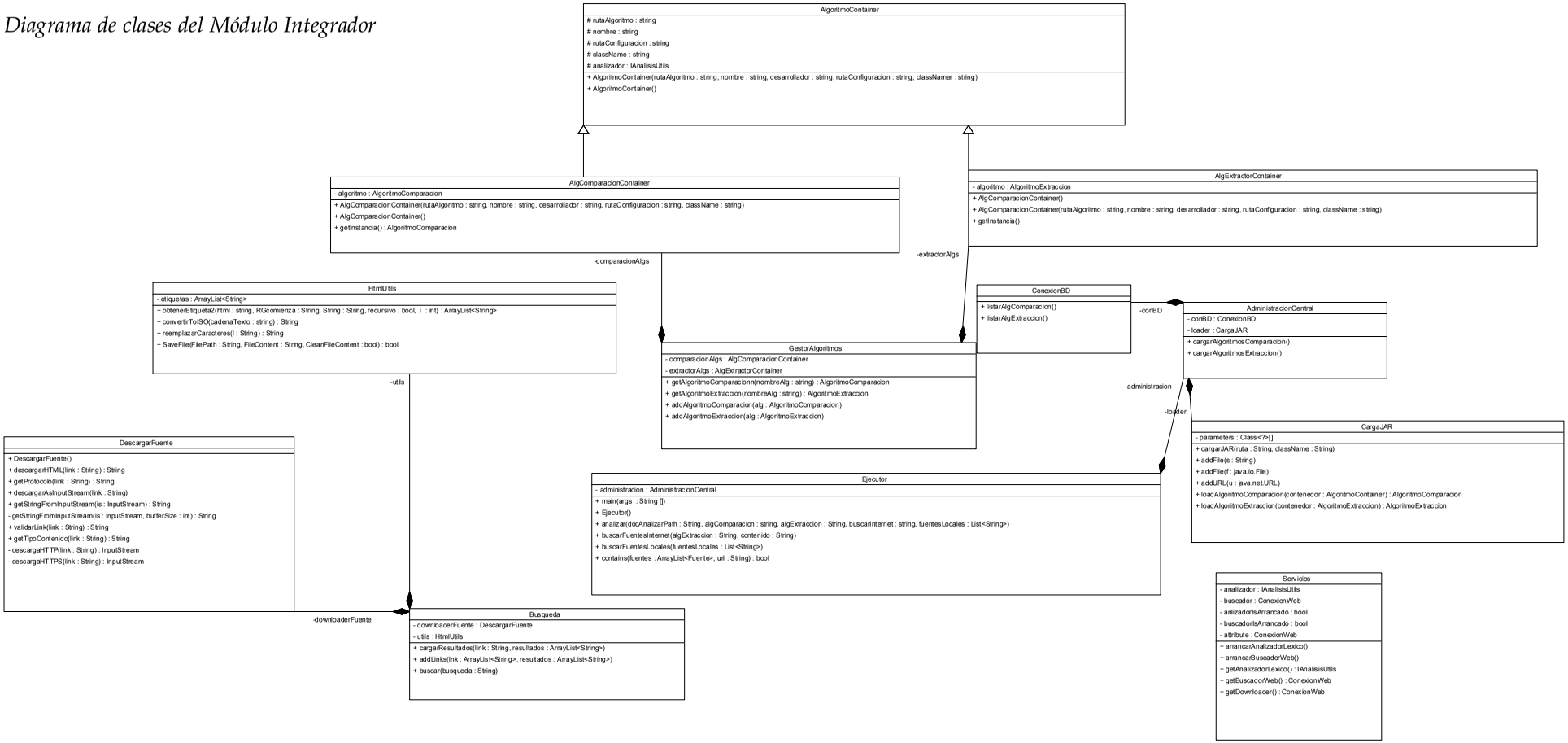



Figura 8: Diagrama de clases del Módulo Integrador

### 3.4.3 *Módulo de acceso concurrente*

Para el acceso concurrente se utilizó un servidor de aplicaciones, que recibe las solicitudes de análisis y registro de algoritmos a través de Web Services de tipo Soap, el frontend del sistema accedera a dichos Web Services y les enviará los parámetros necesarios para realizar un análisis o administrar los algoritmos. Este módulo además contará con un manejo de solicitudes a través de JMS, para atender múltiples usuarios a manera de una cola, se manejara identificadores de peticiones para que los usuarios envíen sus solicitudes y posteriormente puedan consultar el estado del análisis, es decir un usuario envía una solicitud y recibe un código con el cual podrá acceder al servicio y verificar el estado de su petición el cual podrá ser:

- En espera.
- Procesando
- Error de procesamiento
- Terminado

# CAPÍTULO IV

# EJECUCIÓN DE PRUEBAS Y ANÁLISIS DE RESULTADOS

# 4

# EJECUCIÓN DE PRUEBAS Y ANÁLISIS DE RESULTADOS

## 4.1 EJECUCIÓN DEL PLAN DE PRUEBAS.

Las pruebas que se realizaron para poder observar el correcto funcionamiento y rendimiento de la plataforma fueron las siguientes:

- Pruebas de la plataforma en general:

A fin de verificar el correcto funcionamiento de la plataforma, se simuló un proceso de análisis de plagio real y para ello se enviaron múltiples consultas usando n-gramas de 1 a 10 palabras. En la Figura 6 se aprecian los tiempos de respuesta obtenidos por los tres motores de búsqueda empleados. Se puede observar que Bing posee tiempos de respuesta del orden de los 800ms, Google de 2000ms y Yahoo de 2500 [18].

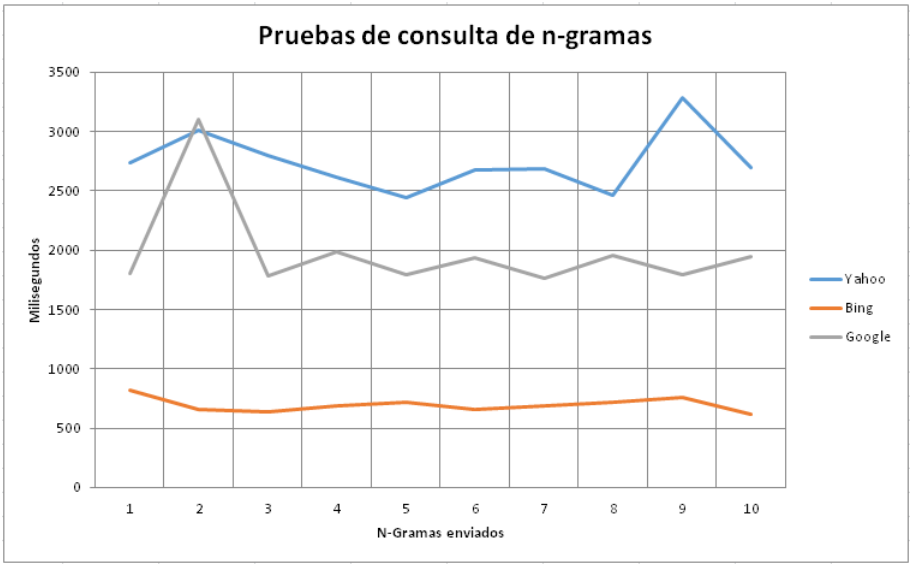



Figura 9: Tiempo de respuesta de los buscadores empleados en la plataforma usando n-gramas (de 1 a 10)[18]

- Prueba del módulo de carga

para medir la velocidad de carga y extracción de información textual de los archivos que se desea analizar. En la tabla 3 se presenta una cuadro comparativo de la velocidad de lectura de acuerdo al formato con un texto de 5272 palabras y 35254 caracteres[18].

- prueba de modulo de analisis lexico

Al igual que en los casos anteriores, también se evaluaron algunas de las funcionalidades del módulo de análisis léxico utilizando textos y palabras de prueba. Dichos resultados se detallan a continuación[18].

| Formato | Velocidad de carga a la plataforma |
|---|---|
| PDF | 948 ms |
| DOC | 514 ms |
| DOCX | 4371ms |
| Texto plano | 81ms |

Cuadro 3: Velocidad de lectura de acuerdo al formato

*Texto de pruebas*

El texto usado para pruebas es el que se transcribe en el siguiente párrafo:

El plagio tiene una gran diversidad de clasificaciones, que pueden incluir diferentes áreas o tipos de obras, por ejemplo plagio en obras musicales, obras literarias, imágenes, etc. pero en este trabajo se procurará centrarse en los principales tipos de plagio en textos y se detallan a continuación

*Análisis de un documento*

El analizador es capaz de reconocer y separar el documento en párrafos y llevar a cabo diversas operaciones.A continuación presentamos algunas de las más utilizadas y el resultado que producen cada una de ellas:

Eliminación de Stop Words

plagio tiene gran diversidad clasificaciones,pueden incluir diferentes áreas tipos obras por ejemplo plagio obras musicales obras literarias imágenes pero este trabajo se procurará centrarse principales tipos de plagio en textos y se detallan a continuación

*Tokenizar en NGramas*

por ejemplo (bi-gramas)

El plagio plagio tiene

tiene una

una gran...

Cálculo de la palabra más relevante de acuerdo a la mayor frecuencia de aparición:

argmax(fa) = plagio,

Donde fa = frecuencia de aparición

Adicionalmente se tienen funciones de preparación para el análisis que son llamadas de manera automática, por ejemplo, el eliminar caracteres duplicados como espacios, saltos de carro, etc. Otra funcionalidad de este tipo es eliminar los caracteres que no son alfanuméricos (por ejemplo: la coma, el punto y coma, etc.)[18].

*Análisis de un párrafo (palabra por palabra)*

El análisis efectuado se basa inicialmente en reconocer los párrafos del documento, para luego empezar a analizar palabra por palabra. Por cada palabra que se encuentre en el texto, el analizador léxico es capaz de devolver sus características léxicas, como son la etiqueta Eagle (la cual es un código de representación morfológica de una palabra de acuerdo a FreeLing), el tiempo en el que se encuentra, si está en plural

o singular, etc., además, se obtiene el lema, un listado de los sinónimos, antónimos, traducciones y posibles significados de esa palabra. Por ejemplo con la palabra "mayoría"[18].

*Características Generales*

Palabra: mayoría
Tipo: NCFSoo Lema: mayoría
Traducciones Bulk
Majority Absolute_majority
Legal_age
Majority
Sinónimos
Mayoría
Mayoría absoluta
Antónimos
Minoría

*Significados*

Más de la mitad de los votos

*Prueba del algoritmo de detección de plagio*

Finalmente, se probó la plataforma con un algoritmo de detección de plagio basado en la técnica Vector Space Model (Modelo de espacios de vector) y similitud de cosenos, el cual consiste en representar cada documento como un vector en el espacio y buscar la diferencia entre estos usando la similitud de cosenos, cuyo resultado está en el rango de 0 a 1[18]

En este caso, el proceso consiste en guardar todas las palabras del documento en un hashtable, así como también el número de veces que se repite, esto se hará con los dos documentos que queremos comparar. Luego de ello, en el hash se quedarán solo las palabras comunes entre los dos documentos, para finalmente aplicar similitud de coseno entre los dos vectores como se muestra en la Figura y verificar el porcentaje de plagio [18].

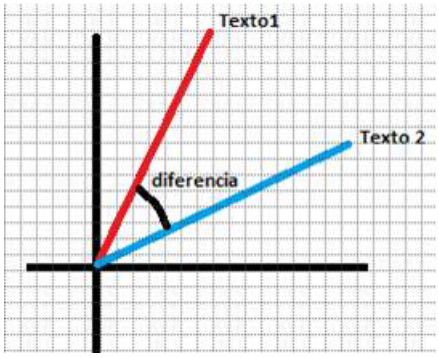



Figura 10: Vectores con palabras comunes a los dos textos

A continuación se muestra una pequeña parte de los textos comparados.

*texto1*

El creador de Facebook es Mark Zuckerberg, estudiante de la Universidad de Harvard. La compañía tiene sus oficinas centrales en Palo Alto, California. La idea de crear una comunidad basada en la Web en que la gente compartiera sus gustos y sentimientos no es nueva, pues David Bohnett, creador de Geocities, la había incubado a fines de los años 1980. Una de las estrategias de Zuckerberg ha sido abrir la plataforma Facebook a otros desarrolladores.

Entre los años 2007 y 2008 se puso en marcha Facebook en español traducido por voluntarios,7 extendiéndose a los países de Latinoamérica. Casi cualquier persona con conocimientos informáticos básicos puede tener acceso a todo este mundo de comunidades virtuales.

*texto2*

Fue creado por estudiantes universitarios, se dice que originalmente Facebook no fue planeado sino que resulto más de lo que realmente se esperaba, los principales fundadores de esta red fueron: Mark Zuckerberg junto a Eduardo Saverin, Chris Hughes y Dustin Moskovitz siendo el primero el CEO actual de la empresa que es hoy Facebook.

Modelo de Negocio Su negocio está basado en la información de los usuarios, es decir cuando los usuarios comparten sus gustos y preferencias, la red analiza esta información para ofrecer publicidad personalizada, además Facebook vende las estadísticas de preferencias a empresas grandes a quienes les interesa y sirve esta información.

*texto3*

El plagio tiene una gran diversidad de clasificaciones, que pueden incluir diferentes áreas o tipos de obras, por ejemplo plagio en obras musicales, obras literarias, imágenes, etc. pero en este trabajo se procurará centrarse en los principales tipos de plagio en textos y se detallan a continuación. Al comparar los textos se obtuvieron los siguientes resultados:

| Documento 1 | Documento 2 | % Similitud |
|---|---|---|
| Texto 1 | Texto 1 | 100 % |
| Texto 2 | Texto 2 | 100 % |
| Texto 3 | Texto 3 | 100 % |
| Texto 1 | Texto 2 | 25.51 % |
| Texto 1 | Texto 3 | 20.053 % |
| Texto 2 | Texto 3 | 0.0844 % |

Cuadro 4: Resultado de la prueba del algortimos con diferentes textos

Para contrastar se han comparado los documentos contra sí mismos para verificar que se detecte un 100 % de similitud.

## 4.2 ANÁLISIS DE PRECISIÓN, COBERTURA Y F-MEASURE

Este análisis cubre la funcionalidad de los algoritmos con los que ha sido probada la plataforma, cabe destacar que estos resultados variaran

conforme el algoritmo que se ejecute, se ha usado un texto de prueba y se muestran los resultados relevantes en la tabla de analisis F-Measure.

*Tema: Algoritmos genéticos*

*Texto:*

Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico. En los años 1970, de la mano de John Henry Holland, surgió una de las líneas más prometedoras de la inteligencia artificial, la de los algoritmos genéticos.1 Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica...

Para saber si el algoritmos implementado en la plataforma acertó o no con respecto al tema entre los dos documentos, es decir los dos tratan des mismo tema. Se va tomar como referencia un porcentaje mayor a 30 es decir si es porcentaje entre documentos con el mismo tema es mayor a 30 será un acierto, y será un desacierto si el porcentaje entre documentos que no traten del mismo tema, sea mayor a 30.

| Texto | Tema | Porcentaje detectado | Acierto/Desacierto |
|---|---|---|---|
| El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado tremenda popularidad en todo el mundo durante los últimos años. Se presentarán aquí los conceptos básicos que se requieren para abordarla... | Algoritmos Genéticos | 33 % | Acierto |
| John Holland desde pequeño, se preguntaba cómo logra la naturaleza, crear seres cada vez más perfectos (aunque, como se ha visto, esto no es totalmente cierto, o en todo caso depende de qué entienda uno por perfecto). Lo curioso era que todo se lleva a cabo a base de interacciones locales entre individuos... | Algoritmos Genéticos | 40 % | Acierto |
| El estudio de la genética permite comprender qué es lo que exactamente ocurre en el ciclo celular, (replicar nuestras células) y reproducción, (meiosis) de los seres vivos y cómo puede ser que, por ejemplo, entre seres humanos se transmiten características biológicas genotipo (contenido del genoma específico de un individuo en forma de ADN), características físicas fenotipo, de apariencia y hasta de personalidad... | genética | 21 % | Acierto |
| La genética es una ciencia, y por lo tanto como tal, implica "un conocimiento cierto de las cosas por sus principios y sus causas". Entonces... ¿cuáles son estas cosas que como ciencia la genética estudia?, pues, la "Herencía Biológica", y la "Variación". Y, sus principios y causas, son las "leyes y principios" que gobiernan las "semejanzas" y "diferencias" entre los individuos de una misma... | genética | 25 % | Desacierto |
| Basados en modelos computacionales de la evolución biológica natural, los algoritmos genéticos pertenecen a la clase de los algoritmos evolutivos, junto con la programación evolutiva, la evolución de estrategias y la programación genética... | Algoritmos Genéticos | 42 % | Acierto |

Cuadro 5: Comparacion del algoritmo con diferentes documentos

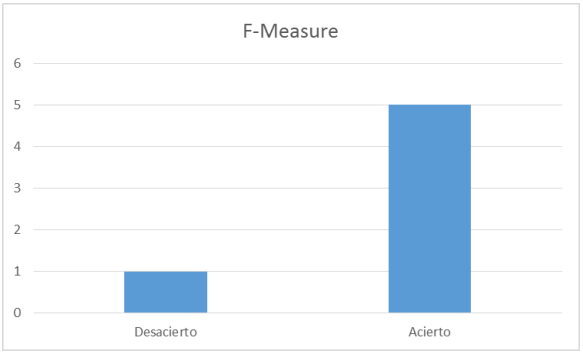



Figura 11: F-Measure de los documentos

## 4.3 ANÁLISIS DE PROBLEMAS PRESENTADOS

*Problemas:*

- Dificultad para el reconocimiento de párrafos en la carga de documentos en formato pdf debido a la forma de lectura de las librerías.
- Problema de acceso al buscador Yandex: Bloquea de manera instantánea, no se logró evitar el ser tomados como un software robot. Esto provocó que en la plataforma no se tome en cuenta para la búsqueda a dicho motor, debido a que al hacerle un par de consultas continuas a este buscador, inmediatamente nos enviaba un capcha los cuales son muy complicados de llenarlos automáticamente mediante programación, de ahí el descarte de este buscador.
- Un problema que se presentó es el hecho de que una vez subido el algoritmo, en caso de ser modificado, los cambios se reflejarán al reiniciar la plataforma. Esto se debe a que los classloader que estamos utilizando no reconocen otras instancia del algoritmos sólo la primera que se hizo, por lo tanto si se quiere modificar un algoritmo existente, cuando la plataforma esta en marcha no se lo podrá hacer, se requerirá reiniciar el sistema para volverlos a cargar a memoria con las nuevas modificaciones.
- Se tendrán problemas de conectividad, si un web proxy desaparece o cambia de dominio, ya que no se lograra acceder al buscador a través del servidor proxy. debido a que la plataforma va alternando entre proxys y google para evitar el bloqueo, por lo tanto se tiene un dependencia del proxy para que la plataforma siga trabajando sin que google la detecte como un software robot.

## 4.4 COMPARACIÓN DE RESULTADOS RESPECTO AL ESTADO DEL ARTE.

En este aspecto el análisis del sistema no se va centrar en que tan preciso son los resultados de los algoritmos usados, ya que su funcionalidad primordial es ser una plataforma para algoritmos de detección de plagio, es decir brinda servicios para el desarrollo y prueba de algoritmos nuevos, por lo tanto el análisis se enfocará en plataformas que brindan servicios similares a esta

Se ha tomado como referencia articulos de sistemas similares, para realizar esta comparación, ya que no se dispone de un acceso lo suficientemente amplio para realizar la comparación de manera funcional.

*Estudio de las técnicas de detección de plagio textual y análisis de sinonimia en ensayos y desarrollo de un sistema prototipo[2]*

Al comparar nuestra plataforma con la desarrollada por Bernardo Ullauri y Andrea Flores , la cual está desarrollada en un lenguaje de programación java, y también cuenta con una interfaz web y soporte para documentos como pdf, Word y txt, se puede apreciar que Plamdep, es una plataforma es mucho más modular, debido a que en esta se tiene la facilidad de brindar servicios que permiten la creación y prueba de nuevos algoritmos, a diferencia del sistema propuesto por ellos, la cual no brinda las mismas funcionalidades para la creación y prueba de nuevos algoritmos sino más bien ejecuta un algoritmo como tal.

Otra diferencia entre las plataformas es que la esta cuenta con una versión para desarrollador la cual va permitir probar algoritmos nuevos antes de subir estos hacia la plataforma a modo de producción, a diferencia de la otra la cual no cuenta con esa posibilidad. Como se puede observar las dos plataformas son completamente diferentes, Plamdep está orientada a brindar servicios que permitan la creación de nuevos algoritmos, mientras que la otra está más dedicada la detección de plagio basado en sinonimia y ngramas.

*Utilización de la plataforma Hadoop para la detección de potencial plagio con indicadores de probabilidad de certeza de las tareas enviadas a un Sistema de Administración de Cursos (aplicable para SIDWeb o Metis[11]*

Esta plataforma utiliza Hadoop para poder procesar una gran cantidad de documentos, a diferencia de la nuestra la cual no utiliza ninguna herramienta adicional para este procesamiento, el trabajo realiazado por Eduardo Segundo Cruz Ramírez y Diego Armando Lavayen Alarcón, tiene como finalidad la detección de plagio entre documentos, soporta diferentes tipos de formatos entre los cuales están, pdf, Word,txt,html, zip, rar etc[11], en esta característica podemos ver una diferencia respecto a Plamdep, la cual es que esta ultima no soporta formatos zip ni rar, otra es que Plamdep está más orientada a la creación y prueba de algoritmos nuevos. Por lo tanto ofrece servicios tales como, un analizador léxico, acceso a una base de datos de sinónimos y antónimos entre otras características, cosas que la plataforma de Eduardo y Diego no ofrecen.

*Sistema de información de detección de plagio en documentos digitales usando el módulo document fingertprinting [25]*

La tesis realizada por Fernando Emilio está dividida en módulos cada módulo ofrece diferentes servicios, entre los cuales están la de permitir que los documentos que sean ingresados para el análisis, sean guardados en un repositorio permitiendo así un mayor rendimiento[25], a diferencia de la nuestra la cual no tiene dicha opción, pero cabe destacar que los módulos que ellos implementen en su trabajo, no están disponibles para los desarrolladores que pretendan utilizarlos, estos tampoco funcionan independientemente, una de las gran ventaja que

tienes la plataforma Plamdep es que puede ser utilizada mediante la web, ocurre lo contrario con la de Emilio la cual es solo de escritorio.

# CAPÍTULO V

# CONCLUSIONES Y RECOMENDACIONES

# 5 CONCLUSIONES

## 5.1 CONCLUSIONES

- Para el desarrollo de la plataforma se requirió el uso de diferentes tecnologías brindadas por los lenguajes de programación orientado a objetos, esto facilitó la integración de los diferentes módulos.
- Asimismo, a fin de que la aplicación sea modular, se requirió una programación usando paradigmas de desarrollo orientado a objetos para obtener buenos resultados tanto en la ejecución de algoritmos como en el manejo de la concurrencia. .
- A pesar de haber diferentes librerías y software que facilitan el desarrollo de plataformas similares, se ha optado por construir toda la plataforma usando software libre, lo cual demuestra que se pueden obtener buenos resultados usando este tipo de software.
- Las facilidades de portabilidad que brindó el lenguaje de programación usado fueron las que facilitaron que plamdep pueda ser ejecutado en diferentes entornos tales como desktop, web, móvil o diferentes sistemas operativos, adicionalmente sobre este aspecto es el hecho de que si los desarrolladores usan librerías adicionales son fácilmente integrables a la plataforma.
- Es importante mencionar PlaMDeP es una plataforma de soporte al desarrollo de algoritmos de detección de plagio académico innovadora, ya que la mayoría de contribuciones científicas que se realizan en la actualidad se centran en el estudio de nuevas formas de detectar plagio o analizar el contenido de documentos.

## 5.2 RECOMEDACIONES

- Para el problema de la actualización de cambios de los algoritmos que ya han sido cargados en la plataforma, se recomienda el uso de ClassLoaders de contexto, los cuales permitirán cargar diferentes versiones de los algoritmos en tiempo de ejecución, es decir sin requerir el reinicio de la plataforma.
- Mejorar el rendimiento del acceso a la web, ampliando el número de web proxys que brindan acceso a los buscadores. Esto no solo mejorara el rendimiento sino también permitirá estar más preparados en caso de que un proxy falle o deje de brindar el servicio, debido a que la plataforma no se verá demasiado afectada por ese hecho, al tener disponibles proxys alternativos.

## 5.3 TRABAJO FUTURO

- Desarrollar nuevos algoritmos de detección y extracción de características para ser implementados y desplegados en la plataforma, para que de esta forma el usuario tenga una mayor variedad de

algoritmos al momento de utilizar la plataforma, actualmente solo consta de un algoritmo de cada tipo.

- Crear bindings para desarrollar algoritmos de detección en otros lenguajes de programación consumiendo los servicios brindados por la plataforma desarrollada, ya sea usando interfaces tales como Web Services, comunicación por sockets o implementaciones de Corba. Con esto la plataforma en el futuro daría soporte a algoritmos desarrollados en c, c++, o cualquier lenguaje que pueda consumir las interfaces anteriormente mencionadas, y no solo se limite a aceptar algoritmos desarrollados en Java como lo hace actualmente.

- Crear otras interfaces de usuario para el consumo del servicio, tales como interfaces en dispositivos móviles, etc. Esto haría que la plataforma puede ser consumida en sistemas operativos de teléfonos móviles, tales como android, ios etc, haciendo que el usuario pueda utilizar la plataforma de plagio desde diferentes dispositivos.

- Permitir que los cambios que se realicen a los algoritmos, sean visibles en la plataforma sin tener que reiniciarla, permitiendo así que un desarrollador de un algoritmo puede subir modificaciones del mismo y que esas modificaciones se reflejen al momento de subir el algoritmo modificado

# BIBLIOGRAFÍA


[1] Marta Vila y Paolo Rosso Alberto Barrón Cedeño. Detección automática de plagio, de la copia exacta a la paráfrasis, Agosto 2013. URL http://users.dsic.upv.es/~prosso/resources/BarronEtAl_JLF10.pdf. (Cited on pages 4, 5, 6, 7, and 10.)

[2] Benito Bernardo León Ullauri Andrea Elizabeth Flores Vega. Estudio de la técnicas de detección de plagio textual y análisis de sinonimia en ensayos y desarrollo de un sistema prototipo. Master's thesis, Universidad Politécnica Salesiana, 2013. (Cited on pages 20 and 60.)

[3] Plagiarism Checker. Plagiarism checker, febrero 2014. URL http://www.dustball.com/cs/plagiarism.checker/. (Cited on page 12.)

[4] Publimetro Chile. Presentan eficaz detector de plagio, enero 2012. URL http://www.publimetro.cl/nota/teknik/docode-presentan-eficaz-detector-de-plagio/xIQldz!j4NqmBnDTZUI/. (Cited on page 11.)

[5] Revista Virtual Universidad Católica del Norte. Para evitar el plagio: reflexiones y recomendaciones. las ideas en préstamo, Agosto 2013. URL http://www.redalyc.org/articulo.oa?id=194220464001. (Cited on page 4.)

[6] Plagiarism Detector. Plagiarism detector, diciembre 2013. URL http://plagiarism-detect.com/. (Cited on page 13.)

[7] José Manuel Martín-Ramos Diego A. Rodríguez-Torrejón. Leap: Una referencia para la evaluación de sistemas de detección de plagio con enfoque intrínseco, Julio 2012. URL http://users.dsic.upv.es/grupos/nle/ceri/papers/ceri2012_torrejon_leap.pdf. (Cited on pages 7, 8, and 10.)

[8] Docode. Docode, agosto 2013. URL http://www.docode.cl/. (Cited on page 11.)

[9] docx4j. docx4j, marzo 2013. URL http://www.docx4java.org/trac/docx4j. (Cited on page 27.)

[10] duplicheck. duplicheck, febero 2014. URL http://www.duplichecker.com/. (Cited on page 13.)

[11] Diego Armando Lavayen Alarcón Eduardo Segundo Cruz Ramírez. Utilización de la plataforma hadoop para la detección de potencial plagio con indicadores de probabilidad de certeza de las tareas enviadas a un sistema de administración de cursos (aplicable para sidweb o metis. Master's thesis, Escuela Superior Politécnica del litoral, 2010. (Cited on page 60.)

[12] Paolo Rosso Enrique Valles Balaguer. Detección de plagio y análisis de opciones, marzo 2013. URL http://repository.dlsi.ua.es/513/1/VallesRosso_PLNE10.pdf. (Cited on page 12.)

[13] El Espectador. Suspenden a periodista de time y cnn por un caso de plagio. URL http://www.elespectador.com/impreso/cultura/medios/articulo-suspenden-periodista-de-time-y-cnn-un-caso-de-plagio. (Cited on page 3.)

[14] Freeling. Freeling, junio 2013. URL http://nlp.lsi.upc.edu/freeling/. (Cited on page 25.)

[15] Dario G. Funez and Marcelo L. Errecalde. Detección de plagio intrínseco usando la segmentación de texto, 2011. URL http://sedici.unlp.edu.ar/handle/10915/18580. (Cited on page 7.)

[16] Bilal Zaka Hermann Maurer, Frank Kappe. Plagiarism - a survey, Diciembre 2008. URL http://jucs.org/jucs_12_8/plagiarism_a_survey/jucs_12_08_1050_1084_maurer.pdf. (Cited on pages 3 and 4.)

[17] Narayanan Kulathuramaiyer Hermann Maurer. Coping with the copy-paste-syndrome, febrero 2013. URL http://www.editlib.org/noaccess/26479. (Cited on pages 4, 5, and 9.)

[18] Vladimir Robles-Bykbaev1 Cristian Timbi-Sisalima1 Eduardo Calle-Ortiz Hernán Fajardo-Heras, Manuel Barrera-Maura. Plamdep: Una plataforma modular para el desarrollo y evaluación de algoritmos de detección de plagio académico. *INGENIUS*, 11:32–41, junio 2014. (Cited on pages xiv, 37, 53, 54, and 55.)

[19] HSQLDB. Hsqldb, julio 2013. URL www.hsqldb.org. (Cited on page 27.)

[20] htmlparser. htmlparser, febrero 2001. URL http://htmlparser.sourceforge.net/. (Cited on page 26.)

[21] ieee. Plagiarism, febrero 2006. URL http://www.ieee.org/publications_standards/publications/rights/plagiarism_FAQ.html. (Cited on pages 3 and 9.)

[22] iText. itext, julio 2014. URL http://itextpdf.com/. (Cited on page 27.)

[23] Java. Java, febrero 2014. URL http://java.com/es. (Cited on page 25.)

[24] JBoss. Jboss, agost 2014. URL http://www.jboss.org/. (Cited on page 27.)

[25] Fernando Emilio Alva Manchego. Sistema de información de detección de plagio en documentos digitales usando el módulo document fingertprinting. Master's thesis, Pontificia Universidad Católica de Perú, 2010. (Cited on pages 9 and 60.)

[26] Paolo Rosso Marco Franco Salvador, Parth Gupta. Detección de plagio translingue utilizando el diccionario estadístico de babelnet, diciembre 2013. URL http://www.repositoriodigital.ipn.mx/handle/123456789/14682. (Cited on page 9.)

[27] Andrea Plaza Mauricio Ortiz. *Programación Orientada a Objetos con Java y UML*, volume 1. 2014. (Cited on page 31.)

[28] Revista médica de Uruguay. Plagios y fraudes en la era de la globalización, Enero 2014. URL http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0303-32952006000200001. (Cited on page 3.)

[29] ATL (Association of Teachers and Lecturers). School work plagued by plagiarism, January 2008. URL http://www.atl.org.uk/media-office/media-archive/School-work-plagued-by-plagiarism-ATL-survey.asp. (Cited on page 4.)

[30] PDFBox. Pdfbox, junio 2013. URL http://pdfbox.apache.org/. (Cited on page 27.)

[31] Grammarly Plagiarism. Grammarly plagiarism, diciembre 2013. URL http://www.grammarly.com/?q=plagiarism&gclid=CIiU6c2dmrkCFc-Y4AodZAUAqw. (Cited on page 13.)

[32] PlagScan. Plagscan, agosto. URL http://www.plagscan.com/es/. (Cited on page 12.)

[33] Plagtracker. Plagtracker, febrero 2012. URL http://www.plagtracker.com/. (Cited on page 12.)

[34] POI. Poi, agosto 2013. URL http://poi.apache.org/. (Cited on page 26.)

[35] Postgresql. Postgresql, julio 2014. URL http://www.postgresql.org.es/sobre_postgresql. (Cited on page 27.)

[36] Diego Rodrigez ; Alberto Barron; Grigori Sidorov; Jose Martin; Paolo Rosso. Influencia del diccionario en la traducción para la detección de plagio translingue, febrero 2014. URL http://users.dsic.upv.es/grupos/nle/ceri/papers/ceri2012_torrejon_barron.pdf. (Cited on page 9.)

[37] Santos Urbina; Rosa Ozollo; Jose Gallardo; Cristina Marti; Aina Torres; Maria Torrens. Analisis de herramientas para la detección del ciberplagio, febrero 2012. URL http://gte.uib.es/pape/gte/sites/gte.uib.es.pape.gte/files/CIBERPLAGIO.pdf. (Cited on pages 10 and 11.)

[38] Turnitin. Turnitin, 8 2013. URL http://turnitin.com/es. (Cited on pages 10 and 11.)

[39] José Fernando Sánchez Vega. Detección automática de plagio basada en la distinción y fragmentación del texto reutilizado, Enero 2013. URL ccc.inaoep.mx/~villasen/tesis/TesisMaestria-FernandoSanchez.pdf. (Cited on page 8.)

[40] Viper. Viper, febrero 2012. URL http://es.scanmyessay.com/. (Cited on page 12.)

[41] Wikipedia. classloader, febrero 2013. URL http://es.wikipedia.org/wiki/Java_classloader. (Cited on page 31.)

[42] Wikipedia. Vectorspace, junio 2014. URL http://es.wikipedia.org/wiki/Modelo_de_espacio_vectorial. (Cited on page 38.)

[43] Global Wordnet. Global wordnet, julio 2014. URL http://globalwordnet.org/?page_id=38. (Cited on page 26.)

Part VI

# ANEXOS

# ANEXOS

## A.1 IMPLEMENTACIÓN DE LOS SERVICIOS DE LA PLATAFORMA EN UN ALGORITMO

Para poder implementar los servicios de la plataforma en un algoritmo, lo primero que se tiene que hacer es crear un proyecto java con algún IDE al cual se le deberá agregar la librería jar plamdep-developer.jar, el cual contendrá todo lo necesario para poder utilizar los servicios tales como analizador léxico, párrafos, fuentes etc; dependiendo del tipo de algoritmo que se desea realizar se tendrá que realizar una herencia (extends) de las clases AlgoritmoComparacion o de AlgoritmoExtracion.

Lo que va hacer este extends, es obligar a que el algoritmo implementa dos métodos, los cuales serán necesarios para que el algoritmo pueda ser compatible a la plataforma, los métodos obligatorios son, ejecutar en donde irá la lógica del algoritmo, y GetResultadoFinal en donde se implementará los resultados que el algoritmo devuelve a través de un objeto.

Cabe destacar que si el algoritmo necesita utilizar una librería adicional para su funcionamiento, no se tendrá ningún problema debido a que los classLoader permiten cargar esas librerías adicionales pero se debe tomar en cuenta que esas librerías deben estar especificadas en el classpath del algoritmo, una vez hecho todo lo mencionado anteriormente se podrá subir el algoritmo a la plataforma, pero cabe mencionar que el algoritmo que se pretende subir debe estar bien probando ya que si se quiere modificar el algoritmo subido, se tendrá que reiniciar la plataforma, por ello se ha creado una versión orientada a los desarrolladores en donde se podrá realizar pruebas a los algoritmos antes de subirlo a la plataforma para su despliegue en ambiente de producción.

## A.2 DESPLIEGUE DE LA PLATAFORMA EN LA WEB

Para el despliegue de la plataforma se requiere un servidor de aplicaciones Jboss, se a testeado usando la versión 7.1.1 en el cual se deberán configurar algunos parámetros, dichos parámetros se configuran en el archivo standalone.xml del servidor ubicado en la ruta: "jboss-as-7.1.1.Final\standalone\configuration", es importante asegurarse que se tiene los permisos de lectura y escritura del archivo, principalmente en entornos Linux, adicionalmente recuerde que al realizar los cambios en la configuración se requiere reiniciar el servidor de aplicaciones para que los cambios tengan efecto.

### A.2.1 *Conexión a la base de datos (DATASOURCE)*

La aplicación requiere una conexión a una base de datos Postgresql, para lo cual el servidor debe disponer de un datasource a dicha base de datos, para esto se deberá modificar el archivo standalone.xml del servidor para agregar la nueva conexión, existen diferentes maneras de crear un datasource pero en este caso se usara la siguiente:

- Copiar el JDBC de Postgresql "postgresql-9.3-1101.jdbc41.jar" en la carpeta de despliegue de aplicaciones del servidor "jboss-as-7.1.1.Final\standalone\deployments"
- En medio de las etiquetas <datasources> </datasources> agregar el siguiente contenido: *<datasource jndi-name="java:jboss/datasources/plamdepDS" pool-name="corresponsaliaLocalDS" enabled="true" use-java-context="true"> <connection-url>jdbc:postgresql://localhost:5432/plamdep</connection-url> <driver>postgresql-9.3-1101.jdbc41.jar</driver> <security> <user-name>postgres</user-name> <password>admin</password> </security> </datasource>*
- En estas etiquetas se deberán modificar los parámetros de acuerdo a los de la base de datos que se usara.
- Es importante que el nombre del datasource sea "java:jboss/datasources/plamdepDS" ya que desde la aplicación se hace referencia a ese nombre.

### A.2.2 *Configuración de colas JMS*

El servidor deberá brindar la funcionalidad de colas de mensajes JMS a la aplicación para lo cual se requiere que exista la siguiente cola de mensajes de igual manera en el archivo standalone.xml:

Se deberá agregar las etiquetas requeridas para dar soporte a colas JMS para lo cual se puede tomar como referencia el archivo standalone-full-ha.xml ubicado junto al archivo standalone.xml, lo importante de este aspecto es agregar una nueva cola de mensajes que deberá tener los siguientes parámetros:

*<jms-destinations> <jms-queue name="plamdepQueue"> <entry name="queue/plamdep"/> <entry name="java:jboss/exported/jms/queue/plamdep"/> </jms-queue> <jms-topic name="testTopic"> <entry name="topic/plamdep"/> <entry name="java:jboss/exported/jms/topic/plamdep"/> </jms-topic> </jms-destinations>*

### A.2.3 *Configuración de cuentas de correo*

Para el envío de notificaciones por correo electrónico el servidor sera el encargado de establecer las conexiones a los servidores smtp, para lo cual se requiere realizar la siguiente configuración en el archivo standalone.xml:

*<subsystem xmlns="urn:jboss:domain:mail:1.1"> <mail-session jndi-name="java:/OtherMailSession" from="javaarm@gmail.com"> <smtp-server ssl="true" outbound-socket-binding-ref="mail-smtp-gmail"> <login name="javaarm@gmail.com" password="your_password"/> </smtp-server> </mail-session> </subsystem> <outbound-socket-binding name="mail-smtp-gmail"> <remote-destination host="smtp.gmail.com" port="465"/> </outbound-socket-binding>*

Este ejemplo sirve para usar una cuenta de correo de Gmail, pero en caso de usar una cuenta diferente se deberán cambiar los parámetros de conexión.