UNIVERSIDAD POLITECNICA SALESIANA

SEDE CUENCA

FACULTAD DE INGENIERIAS

CARRERA: INGENIERIA DE SISTEMAS

Tema:

“ESTUDIO DE LAS TÉCNICAS DE DETECCIÓN DE PLAGIO TEXTUAL Y ANÁLISIS DE SINONIMIA EN ENSAYOS Y DESARROLLO DE UN SISTEMA PROTOTIPO”

Tesis previa a la obtención del

Título de Ingeniero de Sistemas

AUTORES:

Andrea Elizabeth Flores Vega

Benito Bernardo León Ullauri

DIRECTOR:

Ing. Vladimir Robles Bykbaev.

CUENCA – ECUADOR

2012

Breve reseña del autor e información de contacto

Andrea Elizabeth Flores Vega
Estudiante de la Carrera de Ingeniería de Sistemas
Facultad de Ingenierías
Universidad Politécnica Salesiana
angieflores88@gmail.com

Benito Bernardo León Ullauri.
Estudiante de la Carrera de Ingeniería de Sistemas
Facultad de Ingenierías
Universidad Politécnica Salesiana
bbernardoleon@gmail.com

Ing. Vladimir Robles B.

CERTIFICA

Haber dirigido y revisado prolijamente cada uno de los capítulos del informe de tesis, realizada por la Srta. Andrea Flores Vega y el Sr. Bernardo León Ullauri, y por cumplir los requisitos autorizo su presentación.

Cuenca, Abril del 2012

Ing. Vladimir Robles B.
Director de Tesis

DECLARACIÓN DE RESPONSABILIDAD

Nosotros, Andrea Elizabeth Flores Vega portadora de la cédula de ciudadanía 0703665794 y Benito Bernardo León Ullauri portador de la cédula de ciudadanía 0105569883, estudiantes de la Facultad de Ingenierías en la especialidad de Ingeniería de Sistemas, certificamos que los conceptos desarrollados, análisis realizados, así como los criterios vertidos en la totalidad del presente trabajo, son de exclusiva responsabilidad de los autores.

Cuenca, Abril del 2012

Andrea Flores Vega. **Bernardo León Ullauri.**

DEDICATORIA

Una meta más en mi vida que he podido cumplir, pero no lo hubiese logrado sin Dios, que es el ser que cada día me da una nueva oportunidad de vivir, de luchar, me ha facilitado los medios para poder cumplir este sueño hecho realidad, es por eso que hoy le dedico este logro a él, por ser mi fortaleza porque a pesar de tantas adversidades pude sentir su presencia, la siento cada día en mi familia.

Lo siento en mi papá cuando cada día sé que se esfuerza por ser mejor, por darnos un techo donde cobijarnos, cuando comparte sus experiencias de vida esperando que nos sirva para un futuro, por darnos cariño, le quiero mucho papi a pesar de que no siempre se lo diga.

Lo siento en mi mamá que se sacrifica mucho cada día, es una mujer luchadora, increíblemente fuerte y suave a la vez, porque sin su apoyo este sueño no hubiese sido posible, este sueño es tanto mío como suyo, y lo hicimos realidad juntas, le quiero mucho mami.

Lo siento en mis hermanos y mi hermana que cada día aportaron en mi lucha dándome ánimos para continuar, que me escuchan y hacen mis días diferentes y lindos a la vez, los quiero mucho.

Lo siento cada día en la persona que Dios puso en mi camino de una manera tan especial, con la persona que sueño en tener un futuro y que cada día quiero más, por ser una persona maravillosa y por ser un hombre increíble, te quiero Alfonso.

Todos forman una parte muy especial en mi corazón, hoy nos toco celebrar un logro en mi vida, que lo comparto con ustedes y se los dedico a ustedes por ser el pilar y la fortaleza que he necesitado y que necesitare durante mi vida.

Angie

El presente trabajo esta dedicado a mi familia. A mis padres por darme la vida, amor, respeto y educación, pero sobre todo, va dedicado a mis hermanos Diego y Esteban ya que es gracias a los dos que soy quien soy. Todos ellos me enseñaron la importancia de no dejarse vencer por los problemas sino más bien saber afrontarlos y vencerlos.

También dedico este trabajo a todos los amigos que conocí a lo largo de la carrera: Fatima, Andrea, Pablo, Victor, William, Eva y otros que siempre estuvieron ahí para brindarme apoyo y creer en mi.

Benito Bernardo Leon Ullauri.

AGRADECIMIENTOS

Agradezco en primer lugar a Dios, por ser el dueño de mi vida y de mis horas, estoy aquí y ahora gracias a él, gracias por cada día de vida, por cada nueva oportunidad para ser mejor, por todo lo que me has dado y lo que me has quitado también, porque todo me ha servido para ser más fuerte y seguir luchando por cumplir con mis metas, mis sueños, mis anhelos.

Gracias papi por su paciencia, por sus esfuerzos, por su empeño y dedicación para con nosotros, porque después de Dios usted es uno de los seres que me dio la vida, gracias por dejarme existir y por enseñarme que la vida no siempre es color rosa, pero hay que saberla llevar y luchar para que cada día sea mejor.

Gracias mami por haberme dado la vida, por cuidarme, por quererme, por luchar siempre para que podamos cumplir nuestros sueños, gracias por haberme dado la oportunidad de lograr este sueño en mi vida, gracias por su apoyo constante, gracias por su cariño y su comprensión.

Tati, gracias por tu apoyo, por entenderme, por tus consejos, Dios fue muy bueno al darme una hermana tan linda como tú, gracias por formar parte de mi vida.

Beto y Fabri, gracias por ser unos lindos hermanos, por apoyarme y darme ánimo cada vez que lo necesito, por hacerme sonreír y por cada experiencia que compartimos.

Gracias Vladi, porque a más de ser un buen profesor ha sido un gran amigo, ha sido un apoyo y todo lo que hemos logrado con este proyecto ha sido porque usted nos ha sabido guiar, un día usted me ofreció su amistad, sepa que usted también puede contar con la mía.

Gracias a todos mis compañeros y compañeras, por cada día compartido con ustedes, por cada experiencia por la que hemos pasado.

Gracias a mis amigos y amigas que a más de ser mis compañeros de aula, supieron ser verdaderos amigos, que me ayudaron y me apoyaron en momentos difíciles, que muchas veces nos toco levantarnos juntos para poder seguir. En especial quiero agradecer a mis mejores amigas Angie Plaza y Fatima Baculima, por ser un gran apoyo en mi vida, por escucharme y por aconsejarme y a mis amigos; Bernardo por compartir este sueño hecho realidad, por tu paciencia en todo este tiempo y porque hemos luchado juntos y hemos realizado el ultimo de nuestros proyectos en la Universidad, fue un gusto haberlo realizado contigo y sabes que puedes contar

conmigo siempre que lo necesites, a Pablo quien me supo ayudar a levantar cada vez que me sentía derrotada, siempre en el momento justo con una palabra de aliento, a Luis gracias por el apoyo que me brindaste y por motivarme a ser mejor cada día, espero que en tu vida también tengas muchos éxitos.

Gracias Alfonso Carmona, por ser un pilar en mi vida, por darme ánimos y fuerza para seguir luchando, por ser la persona que cambió mi mundo, por llenar mis días de luz y de felicidad, gracias por cada detalle de amor que tienes conmigo y por hacerme sentir que soy una persona especial y bendecida por haberte conocido, eres mi gran amor y lucharemos juntos por nuestros sueños, te quiero.

Angie

Realmente debo agradecer a mucha gente que ha caminado a mi lado a lo largo de la carrera universitaria. No se puede (ni se debe) avanzar solo.

En primer lugar agradezco a mi mejor amiga a lo largo de toda la carrera Eva Andrade que a pesar de discutir algunas veces siempre me ha tratado con respeto y consideración, siempre ha estado para escucharme y si le ha sido posible ayudarme. Gracias "pera".

Debo agradecer también a Valeria Farez que a pesar de estar en otra carrera siempre se daba tiempo para compartir su tiempo y alegria conmigo, al igual que Eva, una gran amiga dentro y fuera de la Universidad.

No me perdonaría olvidarme de mencionar a quienes fueron mis mejores consejeros, amigos y complices de la Universidad; obviamente me refiero a todos los "viejos" del Grupo de música moderna de la universidad, gente gracias a la cual podía eliminar todo el estres de la ingeniería a través de la música. German, Isa, Lenin, Juan, Raul, Xavier, Esteban, Juan Miguel y Marcelo. Gracias por su amistad.

Agradezco a angie por haber sido una excelente compañera de tesis, por saber entenderme y valorar mi trabajo, te agradezco mucho por tu comprensión, seriedad y responsabilidad, ha sido un placer trabajar contigo porque si bien nos enfocabamos en avanzar en nuestro trabajo también te permitias contarme cosas y hacerme reir. Hemos desmentido aquella creencia de que hacer la tesis entre 2 personas termina en pelea; de hecho, esta tesis nos ha servido para ser mas amigos aún. Gracias "angie girl".

También agradezco a aquellos compañeros gracias a los cuales podía aprender cosas nuevas o discutir sobre las cosas que ya sabía como son William Solís, Remigio Hurtado y Claudio Calle gracias a ustedes por permitirme aprender también de ustedes.

Agradezco a mi familia por el ejemplo que me supieron dar cuando crecí. Evitar la mediocridad y luchar por los sueños que se tienen y siempre hacerlo con una sonrisa en el rostro, gracias, porque lo que aprendí de mi familia es lo que mas útil me ha resultado a lo largo de la vida y lo que he puesto en practica a diario es obra de sus enseñanzas.

No puedo evitar agradecer a nuestro director de tesis Vladimir Robles quien es un modelo a seguir. Gracias Ingeniero por todo el conocimiento y por la amistad que nos ha sabido brindar siempre. Se necesitan mas docentes como usted, que eviten la mediocridad pero que así también sepan mezclarse con los estudiantes. Gracias por todo, en verdad.

Finalmente agradezco a Dios por darme paz, felicidad, familia, amigos y excelentes profesores.

Sé que seguramente estoy olvidando a alguien pero sé que sabrá entender lo olvidadizo que soy. Gracias a los demás profesores por sus enseñanzas y gracias a mis demás compañeros por su amistad y apoyo.

Benito Bernardo Leon Ullauri.

Contenido

Índice de Imágenes

Índice de Tablas

Índice de Ecuaciones

1. INTRODUCCION AL PLAGIO EN LAS INSTITUCIONES EDUCATIVAS.

Plagiar es: “Copiar en lo sustancial obras ajenas, dándolas como propias” [8].

El plagio es una actividad que está presente en diversos ámbitos de la vida, puede ir desde el ámbito laboral hasta el ámbito educativo, sobre este último trataremos a mayor profundidad a lo largo de este capítulo.

La educación es uno de los principales pilares del desarrollo personal de una persona, sin embargo, este se ve afectado debido a la ausencia de valores y respeto por las demás personas en diversos aspectos entre estos el intelectual, ya que en la actualidad el plagio se ha visto arraigado en el ámbito académico, siendo muchas veces poco detectado por los educadores debido a la habilidad de los estudiantes para llevar a cabo este actividad. Actualmente es una actividad muy común que empieza desde las instituciones de educación básica hasta las grandes universidades, donde se ha ido perfeccionando la habilidad para llevar a cabo el plagio, surgiendo así un sinnúmero de tipos de plagio como son el parafraseo, la sinonimia, la copia textual, etc.

Entre los causantes de este tipo de actitud están el acceso ilimitado a grandes cantidades de información, además que los estudiantes en la actualidad llevan una formación poco investigativa, ya no se molestan en leer, en entender, sino se limitan a copiar y pegar información de internet o de libros sin sus debidas referencias. Otro causante puede ser la falta de tiempo para poder llevar a cabo todas las actividades encomendadas a los estudiantes, además de la ética de la persona y el respeto por lo ajeno son valores que se construyen a lo largo de una vida, es por ello que las instituciones educativas deben formar a las estudiantes de una manera integral y a su vez presentar ciertas normativas que dentro de la institución educativa penalicen el plagio.

Sin embargo, existen casos de plagio que han llegado a tener serios conflictos, un ejemplo de esto es el caso de Karl Theodor Zu Guttenberg quien asistió a la Universidad de Bayreuth para hacer su doctorado en derecho, además de ocupar un excelente cargo como Ministro de Defensa, Zu Guttenberg obtuvo su título de doctorado con honores, pero se realizaron investigaciones que demuestran que su tesis tiene una gran cantidad de plagio, ya que existen ideas que no fueron planteadas por él y que no llevan las debidas referencias como lo indicaba el reglamento de la universidad, ante esta acusación lo que se alegó fue la falta de tiempo y la presión que se tuvo para mantener de manera satisfactoria tanto su carrera como el ejercicio de su profesión. Este hecho tuvo repercusiones a nivel laboral y a nivel académico ya que le

fue quitado su título de doctorado, y en sí su imagen se vio devastada por este suceso [31].

Otro caso de análisis se presenta como una investigación realizada en 23 facultades universitarias de Estados Unidos que muestran los siguientes resultados con respecto al plagio académico [28].

- Al menos cuatro de diez estudiantes plagiaron trabajos de la Web, en el último año.
- Esos mismos estudiantes declararon haber plagiado al menos una vez en el último año información de internet.
- El 38% de los estudiantes dijeron haber plagiado información de la red, alguna vez, ya sea copiando, pegando, parafraseando o citando ideas de otros como propias.
- El hecho más sorprendente es que más de la mitad de los estudiantes no consideró dicha conducta como deshonestidad intelectual.

El plagio se ha convertido en un daño que no tiene únicamente repercusiones a nivel académico sino que ha generado un conflicto social ¿Los profesionales que se forman en la actualidad son lo suficientemente aptos para satisfacer las expectativas del sector empleador? La respuesta salta a la vista, para una empresa no basta con saber que una persona tiene un título sobre sus conocimientos adquiridos, sino es necesario corroborar que aquello que se dice sea cierto, es por esto que en la actualidad la mayoría de empresas admiten a sus empleados siempre y cuando aprueben ciertos prerrequisitos tales como pruebas de aptitud, de conocimientos, y es que no es solo una secuela a nivel personal sino a nivel educativo ya que se crea una falta de confianza en las instituciones educativas y la forma en que estas emiten sus conocimientos hacia su población estudiantil.

1.1 Planteamiento de la problemática.

El plagio se considera como una adquisición de una obra sin dar la correspondiente acreditación o sin la respectiva autorización del creador.

En nuestro medio el plagio se ha vuelto una práctica muy común, no solo en el campo académico sino en otros campos, por lo general la mayoría de estudiantes lo realizan de forma consciente o inconsciente, uno de los motivantes es la falta de lectura y de interés por el tema, las personas ya no se molestan en leer, debido a la falta de cultura o de tiempo, llevando esta situación a una desencadenante cultura de plagio.

Sin embargo, el plagio es un problema que ha estado presente desde hace mucho tiempo atrás, pero en la actualidad esta tendencia se ha ido incrementado debido a la presencia de diferentes medios de comunicación, pero sobre todo por la gran acogida del internet en nuestro medio, este ha sido beneficioso en muchos aspectos de nuestras vidas diarias, sin embargo cuando ha sido mal utilizado ha perjudicado a la propiedad intelectual.

“El plagio académico ocurre cuando quien escribe usa repetidamente más de cuatro palabras de una fuente impresa sin el uso de comillas y sin una referencia precisa a la fuente original en un trabajo que el autor presenta como su propia investigación y estudio” [1].

En la actualidad existen un sinnúmero de programas web que colaboran con el análisis de documentos para la detección de plagio, estos programas trabajan con la detección de plagio a nivel textual, pero no colaboran con las diversas variedades de plagio que podemos encontrar, ya que este se ha vuelto una costumbre perfeccionista a lo largo del tiempo y por ende presenta varios inconvenientes al momento de identificarlo. En virtud de ello hemos visto oportuno la implementación de un prototipo que a más de colaborar con la detección de plagio textual también colabore con la detección a nivel de sinonimia.

Cuando nos referimos al plagio en base a la sinonimia podemos decir que es la copia de documento sin realizar las referencias respectivas, sin embargo este puede tener ciertas variantes ya que algunas palabras son remplazadas por sus sinónimos, este es otro tipo de plagio que intentamos detectar con el prototipo propuesto.

Es importante indicar que existen otros tipos de plagio más complejos de detectar como es el caso del parafraseo, el plagio de ideas, entre otros, pero para esta investigación nos enfocaremos en el plagio textual y por sinonimia tratando de plasmarlo en un prototipo que colabore en su detección en el ámbito educativo. Para

ello está claro que siempre debe existir un umbral para establecer que un documento ha sido plagiado, el mismo que será definido por el docente.

1.2 Análisis de la realidad estudiantil con respecto al plagio.

El plagio no es algo nuevo de nuestra época, es algo por lo que hemos atravesado desde hace mucho tiempo atrás, en diversos lugares del mundo y en diferentes épocas.

Podemos tomar como ejemplo la biblia que es un libro escrito y transcrito desde hace tiempo atrás en el que se pueden encontrar los mismos pasajes con diversos autores.

En la actualidad en el Ecuador y en el mundo esta es una actividad presente y arraigada en la mayoría de personas, ya que por lo menos una vez en su vida cedieron ante esta práctica, sin embargo cabe recalcar que esta no se limita al aspecto académico sino al aspecto profesional también, existen un sinnúmero de casos que se pueden encontrar que relatan situaciones respecto al tema.

El plagio académico está presente en todas las instituciones educativas, en mayor o menor grado, a continuación se muestra las posibles causas del mismo:

Esta situación se atribuye a varios factores entre estos podemos mencionar:

- Tratar de optimizar tiempo y esfuerzo.
- Presentar trabajos de calidad, con el fin de obtener buenas calificaciones.
- Desconocimiento sobre la forma adecuada de referenciar citas o textos.
- Falta de instrucciones por parte de los docentes al momento de enviar trabajos.

Sin embargo cabe indicar que las ideas que expresamos en la actualidad no son en su totalidad originales ya que siempre es necesario basarse en ideas planteadas y plasmadas por otros autores. Esto nos permite fundamentar nuestras opiniones y trabajos investigativos.

El aprendizaje por imitación es una de las formas en la que los seres humanos asumimos el conocimiento, Albert Bandura psicólogo investigador define las siguientes etapas en las que se divide el aprendizaje por imitación como se menciona [6]:

- **Atención**. Observar y a la vez identificar la información relevante.
- **Memoria**. Almacenar la información que se logró identificar como principal y recordarla en el momento oportuno.
- **Motivación**. Es la capacidad que impulsa a desarrollar alguna actividad.
- **Comunicación**. Recordar o acordarse de la información que se retuvo con anticipación, y reproducirla cada vez que sea necesario.

Es por ello que hacemos hincapié en que un documento no puede ser completamente idea de una persona cuando de trabajos investigativos se trata ya que es necesario que el alumno cumpla con la pirámide de aprendizaje que se expuso anteriormente.

Por ello se debe considerar que se debe basar en conocimiento fundamentado por diversos autores, en virtud de esto el prototipo propuesto hará un análisis y presentará una tentativa sobre el documento a ser analizado, pero será decisión del profesor asumir que un documento es o no plagiado.

Una de las formas en las que se puede mitigar de cierta forma esta situación en una institución es precisamente que ésta cuente con las normativas necesarias en caso de presentarse estos casos.

Cabe acotar además que la Constitución del Ecuador defiende los derechos de autor, este artículo expresa lo siguiente:

El artículo número 22 de la Constitución de la República publicada en el R.O. No. 449, de 20 octubre de 2008, determina que: “Las personas tienen derecho a desarrollar su capacidad creativa, al ejercicio digno y sostenido de las actividades culturales y artísticas, y a beneficiarse de la protección de los derechos morales y patrimoniales que les correspondan por las producciones científicas, literarias o artísticas de su autoría” [3].

No está por demás mencionar que si la Constitución defiende los derechos de autor, las instituciones educativas deberían crear códigos de ética y moral que apoyen a mantener los derechos mencionados, y las sanciones respectivas por la falta presente.

1.3 Estado del arte.

En la actualidad existen diversas formas de plagio y una forma de contrarrestar este tipo de actividades es el análisis de plagio desde diferentes perspectivas tomando en consideración lo siguiente [6]:

- **Plagio por traducción**: se puede considerar plagio el traducir un texto y tomarlo como propio, por lo general el idioma utilizado es el inglés.
- **Detección por estilometría**: trata de encontrar inconsistencia en el estilo de escritura.
- **Copia textual**: como su nombre indica, consiste en copiar textualmente pensamientos, textos, etc., de otros autores y no realizar las debidas referencias.
- **Copia modificada**: consiste en trabajar sobre una copia textual y sobre esta realizar ciertas modificaciones del texto, estas modificaciones pueden ser reemplazo de palabras por:
- **Sinónimos**: es el reemplazo de una palabra por un sinónimo.
- **Antonimia**: reemplazo de una palabra por su antónimo.
- **Generalización**: se considera el reemplazo de una palabra por una de uso más común.
- **Sustitución palabra por definición**: consiste en reemplazar una palabra por su respectivo significado.
- **Plagio por eliminación:** Consiste en trabajar sobre una copia textual e ir eliminando ciertas palabras.
- **Plagio por segmentación**: está constituido por una copia de plagio textual, sin embargo el cambio radica cuando se utilizan signos de puntuación para separar ciertas ideas.
- **Plagio por paráfrasis**: consiste en cambiar el orden original de la frase, haciendo que esta se re-estructure sin perder el significado.

Como se ha podido observar existen varias formas de plagiar y se ha visto la necesidad de crear herramientas que tengan en consideración los puntos que se mencionaron con anterioridad, sin embargo en la actualidad existen páginas web y programas que son de libre uso que no satisfacen todas las expectativas planteadas, por ello hemos realizado pruebas sobre las siguientes herramientas web:

PROGRAMAS	Texto Idéntico	Sinonimia	Paráfrasis	Acotaciones
Plagium	Si	No	No	Es un programa web Existe una búsqueda rápida y avanzada, para la segunda se necesita registrarse.
Plagiarism Detect	No	No	No	Tiene un costo Para probar la aplicación permite ingresar un máximo de 100 palabras
Dupli Checker	Si	No	No	Es un programa web Permite subir archivos
See Sources	Si	No	No	Es un programa web Permite subir archivos
Doc Cop	Si	No	No	Es un programa web Se debe registrar para acceder a los servicios.
Safe Assign/ MyDropBox	Si	No	No	Es un programa web No reconoce las tildes

Tabla 1. Nivel de detección de plagio de programas web.

Estos programas obtienen buenos resultados con lo que es detección de plagio textual, pero no pudimos alcanzar buenos resultados con respecto al plagio por sinonimia y parafraseo. Es importante mencionar que el corpus que hemos utilizado ha sido creado por nosotros mismos, el mismo que contiene una extensión de 2 páginas que están constituidas por plagio textual y plagio por sinonimia.

Sin embargo existen otras herramientas de uso privativo que podrían detectar este tipo de plagio, pero debido a los costos que esto conlleva se han realizado pruebas con los programas que se mencionaron anteriormente.

Como se ha podido observar para realizar un análisis que satisfaga varios tipos de plagio se deben considerar una serie de observaciones tales como:

Evitar sensibilidad ante la presencia de signos de puntuación y espacios.

Se debe definir un tamaño por frase con el que se vaya a trabajar para la consideración de plagio.

La detección debe ser independiente de la forma en la que las palabras se encuentren formando una frase.

1.3.1 Problemas al momento de detectar plagio.

1.3.1.1 Diversos contextos

Como ya hemos mencionado, el plagio existe en los diferentes aspectos en los que nos desenvolvemos y puede ir desde el plagio académico hasta el plagio en el ámbito profesional, además el plagio puede estar presente en la música, en las imágenes, en los textos, pero para el presente análisis no enfocaremos en el plagio por texto.

Dentro del plagio por texto se estudiará el plagio textual, que consiste en buscar de manera idéntica el texto a través de algunos buscadores web que devolverán resultados con las coincidencias exactas sobre las frases consultadas.

Otro aspecto por cubrir es la detección de plagio por sinonimia, para apoyarnos se buscarán e implementarán algunas herramientas que son de utilidad para el análisis de palabras con sus respectivos sinónimos, esta tarea además podrá ser llevada a cabo con la ayuda de tesauros y paquetes propios para el análisis.

1.3.1.2 Detección de plagio con referencia.

Cuando hablamos de detección con referencia, se puede decir que emplea dos partes; la primera parte es el documento original que vamos a comparar y la segunda es el o los documentos contra los que se va a comparar el primer documento, esta es una de las técnicas que más se utilizan para la detección de plagio [5].

1.3.1.3 Detección de plagio sin referencia.

A diferencia de la detección con referencia podemos decir que cuando vamos a trabajar con este tipo de detección nos basamos solo en un documento, el análisis se realiza de manera intrínseca, es decir, sobre el mismo documento, los puntos a considerar son la sintaxis del documento, la estructura y la extensión tanto de las frases como de los párrafos y la frecuencia de los signos de puntuación [5].

1.3.2 Metodologías utilizadas.

Parece una tarea fácil la detección de plagio pero la realidad es que detrás de todo esto existen muchas normativas y procesos que se deben seguir si es que se tratan de conseguir resultados adecuados, aunque no está por demás de considerar que puede existir un margen de error debido a los diversos tipos de plagio existentes en la actualidad.

1.3.2.1 N-gramas

Al referirnos a n-gramas se menciona que es una sub-secuencia de n elementos de una secuencia dada [7].

En nuestro caso se utiliza para darle un tratamiento estadístico y encontrar las coincidencias entre documentos, así como la frecuencia de aparición en el documento. A los N-gramas también se les puede otorgar una distribución de probabilidad la cual permite encontrar mediante una ecuación cual es la probabilidad de aparición de la siguiente palabra.

1.3.2.2 Stop Words.

Cuando hablamos de stop words o palabras vacías como se conoce en español, nos referimos a todas aquellas palabras que por sí solas carecen de significado, tales como: artículos, pronombres, preposiciones, etc. No existe una base de datos de stop words fija, ya que se presenta algunas variaciones de estas, es por ello que nuestro necesidad hemos tratado de imponernos nuestro propio almacén de stop words que nos permite filtrar palabras que no son de importancia para el análisis que se requiere realizar.

1.3.2.3 Funciones de distancia.

Estas funciones nos sirven para comparar la similitud entre documentos, estas pueden ser calculadas mediante datos cualitativos[1] o cuantitativos[2], existen algunas técnicas que se pueden utilizar las cuales se mencionan a continuación:

- Coeficiente de Jaccard
- Coeficiente de Dice

[1] Cualitativo. Cuando hablamos de que nos enfocaremos en la calidad.
[2] Cuantitativo. Cuando hablamos de que nos enfocaremos en la cantidad.

- Coeficiente de Overlap
- Coeficiente de Roger y Tanimoto
- Coeficiente de Sokal y Michener
- Coeficiente de Czekanowski

Todas estas funciones de distancia nos ayudan a obtener un porcentaje de similitud que existe. Dicho porcentaje variará de acuerdo a los puntos de análisis que se realice en cada uno de los coeficientes.

1.4 Motivos para la elaboración de la tesis.

En esta época de auge tecnológico es muy importante tratar de fomentar una cultura de ética y respeto hacia el trabajo intelectual de las demás personas, sin embargo este es un hecho que muchos estudiantes no ponen en práctica.

Estos aspectos repercuten directamente sobre los estudiantes que se están formando, ya que la creatividad para elaborar sus ensayos o sus trabajos disminuye notablemente a pesar de que la tecnología les ayuda mucho, las ideas que se generan no son pensadas por ellos mismos y no se preocupan por leer y redactar lo que han entendido sino simplemente realizar una copia textual o en otros casos utilizar la sinonimia con el afán de disimular el plagio cometido.

Los docentes suelen estar resignados a este tipo de eventos que lo único que hacen es formar profesionales poco capacitados al momento de enfrentarse al mundo, personas poco creativas que no saben expresar sus ideas, ya que para hacerlo se necesita inculcar una cultura de lectura y escritura. Sin embargo cabe recalcar que se entiende por plagio la copia sin las debidas referencias o autorizaciones de las personas de quienes se tomaron las ideas, ya que desde el punto de vista educativo se apoya el hecho de tomar una idea debidamente referenciada y adaptarla para obtener mejores resultados.

Uno de los motivos que inspiraron a la elaboración de esta tesis, es la colaboración con los docentes en el ámbito educativo para ayudarles a formar profesionales muy capaces en todos los aspectos, sin embargo el docente se ve limitado por la falta de tiempo y el número de alumnos a los que imparte ciertas materias ocasionando que este tipo de faltas no sean detectadas a tiempo y sigan sucediendo. Es por este motivo que hemos creído conveniente la implementación de un prototipo[3] de detección de plagio el cual no remplazará al maestro pero si ayudará a discernir y tomar una decisión sobre cuáles son los trabajos plagiados y cuales fueron inspirados por ciertos autores, dejando así al criterio del docente la disposición final sobre el trabajo presentado.

A continuación revisamos los objetivos que se busca alcanzar con la elaboración del presente trabajo.

[3] En el documento se utilizará la palabra prototipo o sistema para hacer referencia al sistema prototipo que desarrollaremos.

1.4.1 Objetivo General

- Elaborar un estudio de las técnicas de detección de plagio textual y análisis de sinonimia en ensayos y desarrollo de un sistema prototipo

1.4.2 Objetivos Específicos

- Conocer a profundidad la utilización de N-gramas.
- Establecer los posibles formatos que será capaz de identificar el programa.
- Definir cuál será el sistema a utilizar como motor de búsqueda.
- Indagar y revisar sobre las posibles herramientas que colaborarán en el proceso del desarrollo del prototipo
- Desarrollar un sistema prototipo de detección de plagio literal y análisis de sinonimia.
- Diseñar un plan de experimentación para medir el rendimiento del sistema.
- Preparar un corpus que permita realizar la evaluación del sistema.
- Analizar y determinar el mejor esquema de implementación del sistema prototipo.

2. TÉCNICAS Y HERRAMIENTAS ÚTILES EN EL PROCESO DE DETECCIÓN.

2.1. Introducción sobre Procesamiento del Lenguaje Natural.

Sabemos que el lenguaje natural es una de las formas de expresión más antiguas del planeta, y el conocimiento que se ha obtenido a través del tiempo ha sido comunicado en primera instancia a través del lenguaje hablado, luego a través del lenguaje escrito que se utiliza hasta la actualidad, pero también a través de medios digitales que permiten inmortalizar el conocimiento en general.

El Procesamiento del Lenguaje Natural: “El PLN se concibe como el reconocimiento y utilización de la información expresada en lenguaje humano a través del uso de sistemas informáticos” [9].

Es evidente que para nosotros el conocimiento adquirido es entendido y asimilado, no siendo de la misma forma para un computador, dando lugar al nacimiento de esta ciencia, para ello existe varios puntos de análisis sobre un texto, los que facilitan el análisis a nivel computacional, entre estos tenemos [9]:

2.1.1 Morfológico.

Es la categorización que se realiza dentro de una oración, es decir, clasificar cada una de las palabras de acuerdo al tipo, número, grado y género. Este análisis debe estar acorde a las categorías, es decir, si es adjetivo, adverbio, artículo, etc. Cabe recalcar que este análisis está relacionado con el análisis léxico.

2.1.2 Sintáctico.

“La sintaxis estudia la forma en que se combinan las palabras para formar sintagmas y oraciones correctas, determinando el papel estructural de cada palabra y sintagma” [10]. En otras palabras se refiere a la estructura correcta que debe tener una frase.

2.1.3 Semántico.

Hace énfasis en el sentido y al significado de las palabras que se conjugan para formar frases, este es un nivel de análisis complejo debido a los diversos significados que puede tener una palabra, de acuerdo al contexto presentado.

2.1.4 Pragmático.

El análisis pragmático es el encargado de dar el contexto y el significado de la frase en general, se considera uno de los análisis más complejos ya que la forma en que este sea planteado puede originar diversas interpretaciones.

2.2 Planteamiento de los Modelos del Lenguaje

Los modelos del lenguaje se encuentran asociados con la Psicología Cognitiva y la Inteligencia Artificial, que son necesarias para emular la capacidad de aprendizaje del sistema nervioso, enfocándose en el lenguaje

Entre las principales áreas sobre las que se trabajan son: el reconocimiento del habla, reconocimiento óptico y la traducción automática de idiomas.

Para entender mejor los modelos, se analizan a continuación un conjunto de los mismos [16].

2.2.1 Modelo Secuencial.

Este modelo se trabaja por etapas o procesos, los cuales son: oír, comparar, aceptar, interpretar y comprender, estas etapas se relacionan entre sí, ya que la salida de una puede ser la entrada de otra. Este modelo trabaja con una secuencia de caracteres denominada discurso, que debe estar definida lo menos ambigua posible para poder darle una interpretación acorde al contexto. A pesar de los esfuerzos este método ha presentado inconvenientes al momento de la interpretación de los resultados, ya que por más simple que una frase parezca siempre habrá un análisis complejo detrás de esta.

Para realizar un análisis secuencial es necesario partir de un estudio por los niveles inferiores, como el análisis morfológico, sintáctico o el semántico, para luego profundizar con una exploración contextual.

2.2.2 Modelos de Integradores.

A diferencia de los modelos secuenciales este modelo propone un análisis en paralelo y es de carácter interactivo, ya que no sigue un orden jerárquico para obtener los resultados, sino que cada resultado puede interactuar con ese nivel o con otros niveles. La diferencia con el modelo anterior radica principalmente en que el modelo secuencial presta mayor importancia a los significados literales mientras que en el modelo integrador presta igual importancia tanto a los significados literales como a los significados no literales o los que presentan varios significados siendo su uso un tanto ambiguo dentro del análisis.

2.2.3 Modelos Constructivos Lingüísticos.

Se fundamenta en el "modelo de competencia de Chomsky" y operan a nivel superficial o sintáctico aplicando "reglas inversamente transformacionales" con el objetivo de encontrar una interpretación semántica adecuada [22].

Otro modelo basado en la lingüística es el de Chairs y Chairs [18] que hace un análisis sobre los niveles morfológico, sintáctico y semántico. Independientemente de la forma en la que estos son analizados, se busca obtener un significado un tanto superficial, utilizando una diversidad de procesos que el autor no menciona.

En la actualidad los modelos del lenguaje más representativos son los de n-gramas y gramáticas estocásticas.

2.2.4 Los MLE o modelos de lenguaje estocásticos.

Se utilizan para definir una función de probabilidad la misma que permite saber la probabilidad de que una frase aparezca dentro de un texto dado.

2.2.4.1 Modelos Condicionales

Estos modelos se caracterizan por utilizar el teorema de probabilidad para calcular la presencia de un término en determinada frase.

$$p(s) = p(w_1 w_2 \dots w_n) = \prod_{i=1}^{n} p(w_i|h_i),$$

Ecuación 1.Probabilidad para calcular la presencia de un término [18].

Donde la probabilidad de ocurrencia de ***s*** *donde* ***hi*** *se denomina la historia de la palabra* ***wi*** *y está definida como* ***hi = w1 . . . wi−1*** *[18].*

Sin embargo, existen un sinnúmero de frases que pueden aparecer, además el costo de implementación es alto debido al extenso vocabulario que existe, es por esto que a la fórmula de probabilidad se le aplica una variable "**φ**" que ayuda a disminuir los parámetros dentro del modelo, es decir, "la probabilidad de una palabra wi no depende de la historia completa, y ésta es limitada por una relación de equivalencia **φ**" [18] quedando el teorema de la siguiente manera:

$$p(s) = p(w_1 w_2 \dots w_n) \approx \prod_{i=1}^{n} p(w_i | \Phi(h_i)),$$

Ecuación 2. Probabilidad para calcular la presencia de un término aplicando la variable φ [18].

*Donde **φ** es la clase de equivalencia correspondiente a la historia **hi**. De esa manera se puede reducir el número de parámetros a estimar en el modelo [18].*

2.2.4.2 Modelos de n-gramas.

Los N-gramas son modelos de probabilidad que permiten predecir la siguiente palabra a utilizar dentro de una frase, son muy utilizados en reconocimiento del habla y reconocimiento de escritura, en la siguiente sección se explica con mayor detalle el uso de n-gramas.

2.2.5 Gramáticas Estocásticas.

La gramática estocástica sirve como complemento a los lenguajes formales, ya que está fundamentada en teorías matemáticas como las cadenas de Markov.[4] Como definición podríamos decir que una gramática estocástica es un lenguaje en el que las cadenas que los conforman tienen un peso asociado.

[4] www.edicionsupc.es/ftppublic/pdfmostra/OE03502M.pdf

2.3 Estudio y revisión de los N-gramas.

Los N-gramas se encargan de modelizar secuencias de palabras, estos son utilizados en procesos estadísticos para modelos de lenguaje como el reconocimiento de voz, reconocimiento de caracteres, traducciones, etc.
Inicialmente para realizar el análisis correspondiente a un texto se deben realizar ciertas acciones, es decir, realizar un pre-procesamiento de la información que nos permitirá suavizar el nivel de detección de plagio, para ello se considerarán los siguientes aspectos.

2.3.1 Filtración de stop words.

Se conoce como stop words a todas las palabras que no aportan con significado al momento de analizar el lenguaje natural, con este preámbulo podemos explicar que hemos definido nuestra propia base de datos que para nuestro criterio se ha creído conveniente filtrar, sin embargo cabe recalcar que existen distintos listados de stop words que pueden ser encontrados en la web.
Se pueden considerar como stop words:

- **Verbos copulativos** o verbos de unión. “El papel principal de estos verbos es hacer una conexión o vínculo del sujeto gramatical y lo que del sujeto se dice, se habla o se predica […] entre estos están; yacer, semejar, parecer, ser y estar” [11].
- **Preposiciones.** Se utilizan en las frases para relacionar las ideas estas pueden indicar el origen, el destino, etc. Por ejemplo: a, ante, con, de, para, por, salvo, que, sin, etc.

Además de la filtración de palabras comunes en nuestro léxico se ha creído conveniente filtrar caracteres especiales como: acentos, signos de puntuación, signos de exclamación, signos de preguntas y signos que no concuerdan con el conjunto de caracteres alfanuméricos, este análisis se ha realizado con el objetivo de trabajar con palabras que no sean comunes y que nos puedan decir más sobre la temática del documento.

2.3.2 Construcción de n-gramas

Existe una variedad de formas para la utilización de n-gramas, cuando hablamos de "n" nos referimos al valor que puede tomar por ejemplo; bigrama (2-gramas), trigrama (3-gramas), etc.

A su vez los N-gramas pueden ser utilizados de varias formas de acuerdo a la necesidad que se presente:

- **Por sílabas.**

  Texto: "Esta mañana"

  Bigrama:

  Es ta -- ta ma -- ma ña -- ña na

- **Por palabras.**

  Texto: "Esta mañana me levante tarde."

  Bigrama:

  Esta mañana -- mañana me -- me levante -- levante tarde

- **Por números.** Representan la longitud de la palabra.

  Texto: "Esta mañana me levante tarde."

  Bigrama:

  4 6 -- 6 2 -- 2 7 -- 7 5

Cabe recalcar que los N-gramas son utilizados con mucha frecuencia para predicciones, por lo que se les aplica funciones estadísticas, pero debido al planteamiento de nuestro problema hemos tomado el principio de la utilización de los N-gramas más no el uso de probabilidades. Además para nuestra aplicación se ha creído conveniente filtrar los espacios en blanco, para la elaboración de los N-gramas lo cual suavizará el proceso de detección de plagio.

La unidad de comparación que se utiliza es más bien asimétrica, ya que se compara cada uno de los n-gramas del primer documento contra todos los N-gramas del segundo documento, es necesario hacer hincapié en que este tipo de comparaciones se realiza en pares de documentos.

2.3.3 Detección rígida

Al referirnos a detección rígida estamos hablando de una dureza al momento de detectar plagio, para ello lo que hacemos es obtener un texto, este puede ser en formato PDF[5], DOC[6] o DOCX[7] y leerlo para convertirlo en texto plano, una vez realizado este proceso es necesario ir dividiendo el texto para que este sea enviado a buscar en Internet.

Para dividir el texto es necesario plantearse varias opciones; estos pueden ser por frases, párrafos, por páginas, tomando en consideración que se toma como plagio cuando existen más de cuatro palabras juntas iguales a un texto diferente sin las debidas referencias [1].

A pesar de esto también existen diversos inconvenientes al momento de analizar frases ya que se pueden presentar como posible plagio: ecuaciones, referencias mal escritas, etc. Es por esto que es muy importante recordar que el programa final realizará un análisis de todo el documento pero será criterio del profesor decidir si el texto es o no plagiado.

Para el proceso de búsqueda, se ha visto la necesidad de comunicar nuestra aplicación con diversos motores de búsqueda.

Antes de enviar una cadena de texto a ser analizada por los buscadores se debe filtrar las frases referenciadas, para ello nos basaremos en el instructivo de graduación disponible en la página de la Universidad Politécnica Salesiana[8].

Una vez filtrados estos datos, se envía la cadena para que realice búsquedas, los resultados obtenidos son almacenados en tablas hash para luego aplicar análisis como medidas de distancia, estas medidas que se detallan más adelante.

5 PDF. Formato de Documento Portable.
6 DOC. Es un formato de archivo para procesar textos que es utilizado por Microsoft Office.
7 DOCX. Office Open XML es un formato de archivo para procesar textos.
8 Instructivo de Graduación aprobado por el Consejo Superior de la Universidad Politécnica Salesiana, http://www.ups.edu.ec/c/document_library/get_file?uuid=3ae9fed7-fb52-4f24-ba39-7b5f15c10d77&groupId=10156

2.4. Estudio y revisión de los Tesauros.

Todo comenzó con la idea de llevar un orden en los documentos, luego por exponerse que este orden se debería llevar a cabo por temas, continuando por incorporar el término de lenguaje documental, el mismo que dio origen al término tesauro documental.

Tesauro es una palabra de origen griego *thesauros* que significa almacén o tesorería. Hablando conceptualmente un tesauro es un conjunto de palabras que se utilizan para representar significados.

Según la norma ISO 2788 / TC 46 estos pueden ser definidos según su estructura y su función:

- **Función**. Como un instrumento de control terminológico utilizado para trasponer a un lenguaje más estricto el idioma natural empleado en los documentos y por los indizadores [15].
- **Estructura**. Es un vocabulario controlado y dinámico de términos que tienen entre ellos relaciones semánticas y genéricas y que se aplica a un dominio particular del conocimiento [15].

2.4.1 Elementos o estructura.

Para poder entender de una manera más adecuada se debe conocer los términos que se utilizan al momento de manejar un tesauro así como su función dentro del mismo.

Entre estos términos mencionamos: las unidades léxicas, descriptores, no descriptores y relaciones que existen entre términos de un tesauro, tomado de [15].

- **Unidades léxicas.** Estas unidades están conformadas por categorías, pueden ser: descriptores o términos.
- **Descriptores.** También se lo conoce como término de indización, ya que éste designará los conceptos que representan mayor importancia dentro del documento y sobre el cual se realizarán las búsquedas.

  Sin embargo estos descriptores pueden tener ciertas características, como los descriptores pre-coordinados que se utilizan cuando existen términos compuestos, o los descriptores post-coordinados que son utilizados cuando los términos compuestos poseen varios significados, los mismos que son combinados en el momento de indizarlos.

A pesar de esto pueden surgir algunos inconvenientes como la ambigüedad, a continuación presentamos dos casos concretos y la forma de tratarlos:

-**Sinonimia**: esta puede ser mitigada a través de relaciones de equivalencia (descriptor pre-coordinado).

-**Polisemia**: se utiliza relaciones semánticas que permiten analizar el contexto en el que se las está utilizando (descriptor post-coordinado).

Otras formas de evitar la ambigüedad es seleccionar una expresión antes que únicamente un término con el objeto de disminuir los errores. Además se ha incluido un tipo de calificador que se encuentra entre paréntesis para definir mejor una palabra por ejemplo: “Cabo (geografía)” este calificador permite filtrar la información que no coincida con el contexto adecuado.

Además se ha incorporado el uso de notas de alcance cuyo objetivo es describir los posibles sentidos que pueden tener un término, estas notas pueden ser por definición o explicativas.

- **No descriptores.** Se caracterizan por ser sinónimos de los descriptores, estos por lo general no son utilizados para la indización, pueden aparecer dentro de un documento, sin embargo al ocurrir esta situación realizan un llamado a un descriptor con el que se encuentren relacionado semánticamente para obtener los resultados adecuados.
- **Relaciones de equivalencia**. Al hablar de este tema podemos describir como la relación que existe entre los términos descriptores y los no descriptores, sin embargo al entrar a este tema es necesario especificar la diferencia entre sinónimos y cuasi-sinónimos[9].
- Además de controlar los sinónimos también se controla la homonimia[10], antonimia[11].
- **Relaciones jerárquicas.** Se utiliza para referirse a descriptores que se encuentran en una misma clase, pero que tienen diferentes grados estos pueden ser de superioridad o de subordinación por ejemplo: Transporte/aéreo.

[9] Cuasi-sinónimos: son términos que comúnmente se los utiliza con diferente significado, pero para la indización dentro del tesauro se los utiliza como sinónimos, por ejemplo: "Ascensión vertical"→ Ascensor.

[10] Homonimia: son palabras que se pronuncian igual o se escriben de igual forma pero que tienen significados diferentes.

[11] Antonimia: son palabras que tienen un significado opuesto.

- **Relaciones asociativas.** Cuando nos referimos al término asociativas es porque existe una mutua relación entre los términos de tal manera que el orden no afectará los resultados obtenidos, es decir, “si A se asocia a B, B se relaciona con A”.

2.4.2 Elaboración de un Tesauro.

Para la elaboración de un tesauro se requiere una serie de pasos, además el tiempo que se demora en la construcción del mismo será relativo al número de palabras que abarque, en promedio un tesauro que contenga 2000 a 3000 términos tarda en construirse entre 6 y 8 meses. Para su construcción se debe considerar ciertas características entre estas tenemos [15]:

- Las palabras deben estar clasificadas por temática.
- El tesauro debe ser dinámico para poder agregar o retirar datos.
- Debe estar normalizado.
- Se utiliza un lenguaje especializado de acorde al tema.
- Debe servir como puente entre un lenguaje normal y un lenguaje normalizado, de tal manera que la información pueda ser documentada correctamente.

2.4.3 Presentación del tesauro [15].

Entre las formas de presentarse encontramos las siguientes:

- **Presentación alfabética:** se distribuyen alfabéticamente tanto los descriptores como los no descriptores además de las respectivas relaciones entre estos.
- **Presentación sistemática:** está conformada por la categoría (principal) y por el índice alfabético (auxiliar), de tal manera que permita a los usuarios dirigirse a la sección adecuada.
- **Presentación gráfica:** utiliza figuras como árboles y flechas para representar los términos y las relaciones que existen entre estos, utilizando para ello índices alfabéticos.

A continuación se muestra la notación para representar las relaciones de un tesauro.

- Nota de Alcance (NA)
- Use (USE)
- Usado por (UP)
- Término Genérico (TG)

- Termino Específico (TE)
- Término Relacionado (TR)

2.5 Estudio y revisión de las técnicas basadas en medición de similitud y sinonimia.

2.5.1 Medidas de distancia o medidas de similitud.

Las medidas de distancia tienen por objetivo encontrar la similitud o la diferencia en base a datos cuantitativos o cualitativos, para nuestro caso utilizaremos datos cuantitativos sobre los cuales se pueden observar los datos de los gramas utilizados en nuestro caso, el objetivo es analizar los conjuntos con sus cadenas de texto respectivas, las mismas que son comparadas de forma asimétrica como ya se ha mencionado, una vez realizado este proceso se le pueden aplicar diversas medidas sobre los datos obtenidos, para ello se ha creído conveniente exponer el siguiente listado de símbolos con los respectivos significados [14]:

Dónde:

a: Número de caracteres presentes en los dos individuos,

b: Número de caracteres presentes en i y ausentes en k,

c: Número de caracteres presentes en k y ausentes en i,

d: Número de caracteres ausentes en los dos.

2.5.1.1 Coeficiente de Jaccard.

También denominado índice de Jaccard está caracterizado por tratar de encontrar similitud entre dos conjuntos de datos, la cual nos dice que el coeficiente es obtenido mediante la intersección de los dos conjuntos “a” divido para la unión de los conjuntos “a + b + c”

$$J = \frac{a}{a + b + c}$$

Ecuación 3. Coeficiente de Jaccard [14].

2.5.1.2 Coeficiente Overlap

Esta es una medida que se encuentra muy relacionada con el índice de Jaccard, lo que hace es calcular la similitud en base a los conjuntos A y B, para ello realiza una operación de intersección de los conjuntos y los divide para el valor mínimo de los conjuntos, es decir, escoge el conjunto menor para realizar la operación. En otras palabras lo que trata de hacer el coeficiente de Overlap es demostrar que tan contenido esta un conjunto dentro del otro.

$$O = \frac{a}{\min(|b|, |c|)}$$

Ecuación 4. Coeficiente de Overlap [5].

2.5.1.3 Coeficiente de Dice y Sorense

Es una medida de distancias que a diferencia de las otras se caracteriza por darle un valor doble a los datos que tienen presencia dentro de los conjuntos, Sorensen sostiene que “la presencia de una especie provee mayor información que su ausencia” [14]. Se calcula dando valor doble a la presencia o a la intersección de los conjuntos “2a” sobre el valor doble de la intersección “2a” más las ausencias en cada uno de los conjuntos “b + c”

$$D = \frac{2a}{2a + b + c}$$

Ecuación 5. Coeficiente de Dice [14].

2.5.1.4 Coeficiente de Roggers y Tanimoto

En esta medida de distancia se le aporta como valor más significativo a las diferencias antes que a las semejanzas. Se calcula sumando el número de caracteres presentes y ausentes en los dos individuos “a + d” divido para el número de caracteres presentes y ausentes en los dos individuos más del doble valor de los valores presentes en un uno de los conjuntos y ausentes en el otro “a+2b+2c+d”

$$RT = \frac{a + d}{a + 2b + 2c + d}$$

Ecuación 6. Coeficiente de Roggers y Tanimoto [14].

2.5.1.5 Coeficiente de Sokal y Michener (Coeficiente de concordancia simple).

Este coeficiente da el mismo valor tanto a la ausencia como a la presencia de datos dentro de los conjuntos. Se calcula sumando el número de caracteres presentes y ausentes en los dos individuos, divido para la unión de todos los conjuntos “a + b + c + d”.

$$SM = \frac{a + d}{a + b + c + d}$$

Ecuación 7. Coeficiente de Sokal y Michener [14].

2.5.1.6 Coeficiente de Czekanowski

Este coeficiente tiene una gran parecido con el coeficiente de Jaccard, la diferencia radica en que les da un valor doble a las presencias de los datos en ambos conjuntos, cuando hablamos de presencias se hace referencia a las intersecciones que existen entre los dos conjuntos a comparar. Se calcula otorgando un doble valor en los valores presentes en los dos individuos “2a”, divido para el doble valor de los valores presentes en los dos individuos más el valor presente en cada uno de los conjuntos y ausentes en el otro “2a+b+c”

$$C = \frac{2a}{2a + b + c}$$

Ecuación 8. Coeficiente de Czekanowski [13].

Existe un sinnúmero de coeficientes que nos pueden ayudar al cálculo de similitud, sin embargo estos se caracterizan porque sus resultados generan valores entre 0 y 1, que indican el grado de similitud de los documentos.

Otra de las principales características de estos coeficientes mencionados, es que estos se basan o contabilizan la presencia o la ausencia de datos encontrados a lo largo de los documentos que se están comparando, además de tener la capacidad de acoplarse de acorde a la longitud del documento.

Todos estos coeficientes tienen sus aspectos positivos y negativos, sin embargo, se deberá realizar un sinnúmero de pruebas, que permitirán dar un criterio más amplio y escoger el coeficiente que se acople a nuestras necesidades.

2.5.2 Sinonimia.

Con respecto a la sinonimia podemos mencionar que es una de las técnicas que se utilizarán para determinar la carga semántica que poseen cada una de las palabras dentro un determinado texto. Una vez identificadas las palabras de mayor carga semántica se les aplica la técnica Zipf[12] que permite determinar las palabras de mayor longitud, siendo estas las más adecuadas a la temática presente en el documento. Al tener identificados estos factores de análisis se procede a aplicar sinonimia a cada una de estas palabras de mayor carga semántica, para a su vez utilizar N-gramas que permitirán realizar un análisis y determinar con la ayuda de los métodos de detección de similitud cuales son los resultados obtenidos con respecto al análisis realizado.

Otra punto a considerar en el análisis de sinonimia es el contexto de una frase, es decir, el significado que engloba toda la frase, es por ello que si alteramos una palabra debemos cuidar de la semántica y a su vez del contexto, de tal manera que el sentido de la oración no se vea alterado para sea entendido por el lector.

[12] Técnica Zipf: esta técnica nos dice que las palabras más relevantes de un documento son las de mayor longitud.

2.6 Análisis de otras técnicas de soporte.

2.6.1 Cadenas de Markov.

Estas se denominan así por su creador Andrei Markov, se caracterizan porque trabajan en base a probabilidades y procesos estocásticos, es semejante al funcionamiento de los N-gramas ya que "la probabilidad de que ocurra un evento depende del evento inmediato anterior" [20].

Para construir una cadena de Markov adecuada se necesita de los siguientes elementos [19]:

- Un conjunto de estados del sistema.
- La definición de transición.
- Una ley de probabilidad condicional, que defina la probabilidad del nuevo estado en función de los anteriores.

Se entiende como estados al conjunto de variables aleatorias. Se denomina transición al proceso de cambiar el valor del estado.

Las cadenas de Markov pueden clasificarse según sus tipos, entre estos tenemos [20]:

- Cadenas irreducibles
- Cadenas positivo-recurrentes
- Cadenas regulares
- Cadenas absorbentes
- Cadenas de Markov en tiempo continuo.

Las cadenas de Markov pueden ser utilizadas en diversas situaciones, por ejemplo en meteorología se utiliza para crear modelos climatológicos.

Inclusive Google utiliza cadenas Markov para controlar el PageRank de acorde a los valores obtenidos por la cadena.

2.6.2 Analizadores Sintácticos.

Un analizador sintáctico se utiliza en una fase de análisis de un compilador, se considera un estado inicial de estudio del lenguaje natural, la función de este analizador es fragmentar la entrada en diversas estructuras, por lo general la estructura de datos que se utiliza es la de árbol, que permite mantener una jerarquía de los diversos componentes de la frase. Previo a este análisis se debe realizar un análisis léxico, el mismo que crea *tokens* que luego serán analizados sintácticamente generando así la estructura de datos predefinida.

2.6.3 Roles Semánticos.

Estos se encargan de describir la relación semántica que tiene una frase con respecto al predicado, es decir, siempre existirá una frase que tenga el mismo sentido a pesar de que sus componentes sintácticos se encuentren modificados con respecto al orden dentro de la oración o frase [21].

A su vez lo que se trata de analizar con los roles semánticos es la representación genérica con respecto al contexto de una oración, de tal manera que esta no se vea influenciada por el idioma que se esté tratando. Con esta finalidad se creó la teoría gramatical Linking, está teoría define que "la representación sintáctica de los argumentos de un predicado es predecible a partir de la semántica" [21].

Con respecto al etiquetado de roles semánticos es necesario contar con un corpus de entrenamiento que permita posteriormente el reconocimiento de datos. Una de las herramientas que se dispone en la actualidad para el etiquetado semántico es Framenet, desarrollada por Baker, Fillmore y Lowe en el año de 1998.

Framenet se enfoca en la creación de *frames semánticos* en donde cada uno de ellos hace referencia a un escenario, con su respectivo nombre que indica la relación semántica, todo esto se encuentra en un corpus de idioma inglés que contiene alrededor de 3040 verbos [30].

Sin embargo existe una versión de Framenet Español desarrollada por Rüggeberg en el año 2004, cuyo objetivo es identificar la diversidad de relaciones semánticas que pueden existir en base al léxico de predicados del idioma español, el creador de esta versión en nuestro idioma pone a disposición un diccionario online [30].

Algo que se debe recalcar dentro de este tema es que el etiquetado semántico depende de gran manera de los analizadores sintácticos tanto para el aprendizaje como para su correcto funcionamiento.

2.6.4 Ley de Zipf.

Esta ley fue planteada por el lingüista Harvard George Kingsley Zipf en el año de 1935, quien plantea que en el lenguaje común las palabras más cortas se usan con mayor frecuencia a diferencia de las palabras más largas. Para analizar las palabras en un determinado texto se menciona que: "las palabras más frecuentes tienden a ser cortas, ya que hacen la comunicación más eficiente que usando palabras largas" [22].

Además científicos cognitivos apoyan la ley de Zipf alegando que “la previsibilidad de lo que una persona dice se ve más influenciado por la longitud de la palabra que la frecuencia con la que esa persona la usa” [22].

El uso de esta ley es muy útil dentro de nuestro sistema ya que permite clasificar las palabras de mayor relevancia en el análisis por sinonimia.

3. PREPARACIÓN DEL CORPUS Y EVALUACIÓN DE HERRAMIENTAS

3.1 Diseño y preparación del corpus para la experimentación:

Es de vital importancia realizar pruebas del sistema con corpus que se ajusten de la mejor forma a un entorno real de operaciones, por tanto, la mayoría de estos corpus deberán estar basados en documentos reales. Nos referimos a mayoría y no totalidad de documentos reales puesto a que deberemos probar los peores casos, los cuales no existen en abundancia en documentos reales, es por este motivo que se generará una pequeña parte del corpus considerando cualquier eventualidad que se pudiera suscitar alguna vez.

Por este motivo, podemos decir que el corpus resultante estará conformado por un porcentaje de archivos reales (PDF, DOC, DOCX, TXT) y un porcentaje menor de archivos creados por nosotros, los cuales contendrán:

- Gran cantidad de texto
- Formatos para los cuales es más complicado extraer texto (PDF, DOCX)
- Uso de notación numérica y simbólica

3.1.1 Creación de los archivos que conforman el corpus.

Para el caso de pruebas de análisis textual se crearán archivos con frases obtenidas textualmente de sitios web pudiendo obtenerse un archivo que no tenga ningún significado como unidad, el cual nos será útil solamente para realizar pruebas.

Para el caso de pruebas de análisis con sinonimia el texto del documento será escrito por nosotros. El procedimiento es el siguiente:

- Se toma un texto que exista en la web.
- Se crean varias versiones de ese texto con palabras cambiadas por sus respectivos sinónimos.
- Cada versión del texto se diferencia la una de la otra en las palabras modificadas y en las variantes de sinónimos para cada una de estas palabras.

En las pruebas se espera recuperar el texto original al aplicar sinónimos sobre los sinónimos de la prueba. Algo muy parecido a una doble negación en lógica matemática. Por ejemplo, si el estudiante ha cambiado la palabra “dormir” por la

palabra “descansar” el sistema aplicará un sinónimo a la palabra “descansar” con la esperanza de recuperar la palabra “dormir” original.

Una vez obtenido un texto resultante, enviaremos a los buscadores las palabras clave del texto procesado.

Resulta evidente notar que a medida que los archivos aumenten en extensión al sistema le tomará más tiempo y recursos completar su tarea, por tanto, tendremos que categorizar los elementos de nuestro corpus en archivos cortos, medios y extensos. Esto se hace con la finalidad de conseguir que sean cercanos a la realidad tomando en consideración cada tipo de archivo y su categoría.

3.1.1.1 Archivos reales:

La siguiente es una lista con los archivos reales que se pueden encontrar en la web con su respectivo enlace y categoría dentro del corpus:

archivo	URL	categoría
elizalde(tesis).pdf	http://bit.ly/y9eZwe	extenso
estudio.doc	http://bit.ly/zhIVMm	corto
informe.docx	http://bit.ly/zDoqGD	extenso
MYRNA_estudiosdecaso.pdf	http://bit.ly/n9lmNC	medio
otras_palabras.pdf	http://bit.ly/Az1StL	extenso
3historias.pdf	http://bit.ly/ygn9pK	corto
aldea.pdf	http://bit.ly/z8gEl8	corto
fantasmas.pdf	http://bit.ly/vZ2lzJ	corto
expulsionmoriscos.txt	http://bit.ly/zj7Qxo	extenso
parsewiki-corto.txt	http://bit.ly/ykUX7H	corto

Tabla 2. Corpus disponibles en la web

De esta lista crearemos versiones para los archivos cortos, con la finalidad de probar la funcionalidad de sinonimia.

3.2 Instalación y pruebas de las herramientas.

3.2.1 Análisis de herramientas.

Se han investigado varias herramientas que trabajan con el análisis de textos con diferentes alternativas, entre todas las utilidades que éstas presentan nos hemos enfocado en el análisis por sinonimia, y a pesar de que existe una variedad de alternativas no todas satisfacen nuestras expectativas puesto que la mayoría de herramientas se encuentran en diversos idiomas siendo limitadas las que se encuentran disponibles en español. Se ha consultado la página Global WordNet Association, en donde nos muestra un tabla en la que se puede observar las diferentes herramientas con su respectivo idioma (Ver tabla 3), a más de ello hemos encontrado otra herramienta denominada FrameNet que también está disponible en idioma español.

Lenguaje	Recurso	Desarrolladores	Contactos	Licencia
Inglés	WordNet 2.0	Princeton University	Christiane Fellbaum e-mail: Fellbaum@princeton.edu	Princeton U (free download)
Inglés	EuroWordnet	University of Sheffield	Yorick Wilks e-mail:yorick@dcs.sheffield.ac.uk	ELDA/ERA (restricted, license fee required)
Español	EuroWordnet	UNED/UPC/UB	Felisa Verdejo e-mail: felisa@lsi.uned.es	ELDA/ERA (restricted, license fee required)

Tabla 3. Fragmento de la tabla Global Wordnet Organization [29].

A continuación se detallará con mayor énfasis las herramientas que cumplen con los requerimientos y sobre las cuales se realizarán las pruebas necesarias.

3.2.1.1 WordNet

WordNet fue desarrollado en la Universidad de Princeton en el año de 1985, bajo la dirección del psicolingüístico George A. Miller, este proyecto recibió apoyo gubernamental ya que se encontraban interesados en la traducción automática [26].

Esta herramienta fue liberada bajo licencia BSD y consiste en una gran base de datos léxica, que trabaja con diversas relaciones morfológicas entre estas los sinónimos denominados *"synonym sets"* de forma abreviada *"synset"* pero además de esta utilidad también trabaja con antonimia, hiperonimia[13] y meronimia[14]. Dentro de la base de datos cada *synset* posee un significado, a su vez todos estos se encuentran interconectados por relaciones semánticas, sintácticas o léxicas [26].

Las relaciones semánticas son lo que ha permitido que WordNet sea más que un tesauro que se ve limitado a solo encontrar las palabras con su respectivo significado. Wordnet desde su creación fue pensado y validado para experimentar con teorías psicolingüísticas, es por ello que se trabaja con cuatro categorías sintácticas [26]:

- Nombres
- Verbos
- Adjetivos
- Adverbios

Estas categorías a su vez dan paso a que ciertas palabras pertenezcan a más de una categoría, produciéndose lo que en un tesauro común no se daría, esta redundancia se debe a la base de análisis que tiene Wordnet, esta se fundamenta en una matriz de vocabulario, propuesta por Miller, cuyo funcionamiento se trata de ejemplificar en la tabla 4 [26].

[13] Hiperonimia: se utiliza para reemplazar una palabra por otra de uso más general.

[14] Meronimia: es una palabra que forma parte del significado de otra, por ejemplo; "dedo es merónimo de mano"

Significados Léxicos	Formas Léxicas				
	F1	F2	F3	…..	Fn
M1	E1.1	E1.2			
M2		E2.2			
M3			E3.3	…..	
Mn					Em.n

Tabla 4. Matriz de vocabulario de Wordnets [26].

Se puede interpretar la Tabla 4 de la siguiente forma: “La entrada E1.1 implica que la forma léxica F1 puede usarse para expresar el significado M1. En el caso de que haya dos entradas en la misma columna, la forma léxica es polisémica; si hay dos entradas en la misma fila, las dos formas léxicas son sinónimas” [26].

Como se puede ver en la Tabla 4, esta matriz puede ser analizada de dos formas: la primera consiste en una lectura por columna lo que se interpreta como los diversos sentidos que una palabra puede tener de acuerdo al contexto, la segunda forma de leer la matriz consiste en recorrerla por filas lo que permite interpretar un concepto de diversas formas [26].

Con el planteamiento de esta matriz se satisfacen los problemas de sinonimia y de polisemia, entendiéndose por este último cuando una palabra tiene varios significados.

El análisis que utiliza Wordnet es muy útil y es por esto que existen otras variantes de la herramienta como el Eurowordnet, que fue desarrollado en el año de 1994 como un proyecto para satisfacer las necesidades de la población europea. Es por ello que en esta versión se incorpora diversos idiomas como el holandés, italiano, alemán, francés, checo, estonio y español. Cabe recalcar que no todas las versiones de WordNet se encuentran de forma gratuita, debido a esto nosotros hemos utilizado la herramienta Freeling de libre distribución que incorpora el Eurowordnet para el idioma español [26].

Wordnet está conformado por un Índice Inter-Lingual (ILI), es un esquema en el que se encuentran las posibles conexiones que puede tener una determinada palabra con sus respectivos sinónimos independientemente del idioma que se esté utilizando.

3.2.1.2 FrameNet

Surgió como un proyecto en el *International Computer Science Institute* en California-Estados Unidos, se fundamenta en marcos semánticos que consisten en la relación que existe entre una palabra, su significado y todo el conocimiento relacionado a esa palabra. Lo que implica que si se conoce el significado de los temas relacionados con la palabra “venta”, se infiere también sobre el conocimiento acerca de la mercancía, dinero, comprador, vendedor, etc., de tal manera que los marcos semánticos tienen una perspectiva más amplia sobre el lenguaje, dando como resultado construcciones gramaticales [39].

FrameNet está conformado por 10.000 unidades léxicas que implican las palabras con sus respectivos significados, 800 marcos semánticos [39].

Para acceder a la base de datos de consultas de FrameNet se lo realiza de manera web.

Ilustración 1. Base de datos de FrameNet [30].

Sin embargo, no basta con analizar herramientas que colaboren únicamente con la obtención de sinónimos sino también buscar herramientas que permitan analizar la carga semántica de una frase. Una de las herramientas encontradas fue FreeLing que se detalla a continuación:

3.2.1.3 FreeLing

Fue desarrollado en la Universidad Politécnica de Cataluña conjuntamente con el Centro de Investigación TALP y en la actualidad se encuentra a cargo de este proyecto Lluis Padró. Esta es una herramienta de código abierto que permite trabajar con analizadores sintácticos, los que permiten clasificar las palabras de acorde al léxico de los idiomas con los que se trabaja, en la actualidad FreeLing soporta varios idiomas: asturiano, catalán, castellano, galés, gallego, inglés, italiano, portugués y ruso, evidentemente difirieren cada uno de los servicios que brinda freeling de acuerdo al idioma [24].

Se mencionan a continuación los servicios que se utilizarán y que están soportados para el idioma español [24]:

- **Separador de palabras (*Tokenizer*):** proceso que permite analizar una entrada de texto, devolviendo un conjunto de palabras denominadas *word* las mismas que se encuentran separadas de acuerdo a un carácter o cadena de separación.
- **Fragmentar la oración (*Sentence Splitter*):** detectan las oraciones que existen en un texto y crea anotaciones para cada oración que se haya encontrado.
- **Etiquetador (*POS tagger*):** es el encargado de etiquetar el texto y clasificarlo de acuerdo a las etiquetas EAGLES[15] que mencionaremos posteriormente. El etiquetador recibe como parámetros la salida del separador de palabras.
- **Anotaciones sobre FreeLing (*WN sense annotation*):** clasifica las oraciones y devuelve información sobre la categoría (*parole*), lema (*forma*), etc.
- **Sinónimos (*synset*):** recibe una lista de frases “sentence” y obtiene una lista con los posibles sinónimos que se pueden presentar, esta función la realiza con la ayuda de WordNet, sin embargo cabe destacar que esta opción se encuentra en una versión de prueba para Java. Es por ello que puede presentar diversos inconvenientes al momento de la implementación.

[15] EAGLES es un sistema de etiquetado para esquemas morfosintácticos en idioma español.

3.2.1.3.1 Instalación de la librería FreeLing.

Ahora que hemos indicado en qué consiste la herramienta FreeLing procedemos a detallar los procesos de instalación usados durante el desarrollo de esta tesis.

Nos hemos basado en el manual de usuario [31] para realizar la instalación del código fuente. Se van a detallar las dependencias existentes en este manual de usuario, las cuales cambian de número de versión pero no más que eso. Nuestra versión instalada es la 2.2.2.

Primero debemos preparar al sistema con las herramientas necesarias para que se pueda compilar el código fuente, para ellos utilizamos <<build-essential>> que contiene diversos paquetes que nos servirán para generar archivos binarios, el comando es el siguiente:

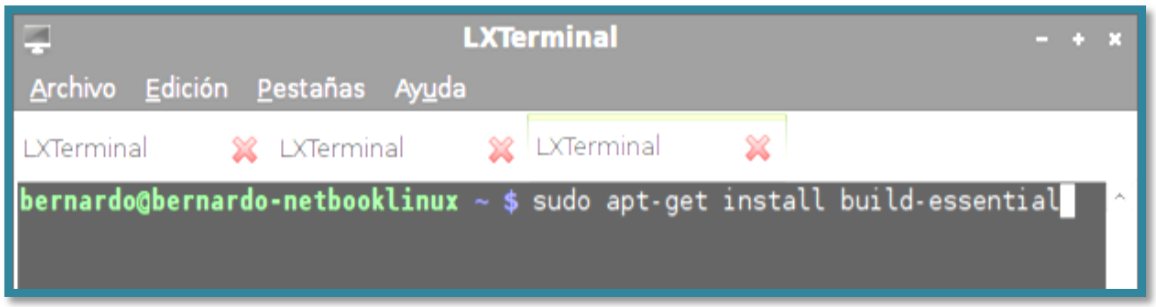


Ilustración 2. Comando que permite compilar código fuente.

Una vez que tenemos las herramientas necesarias procedemos a instalar las dependencias:

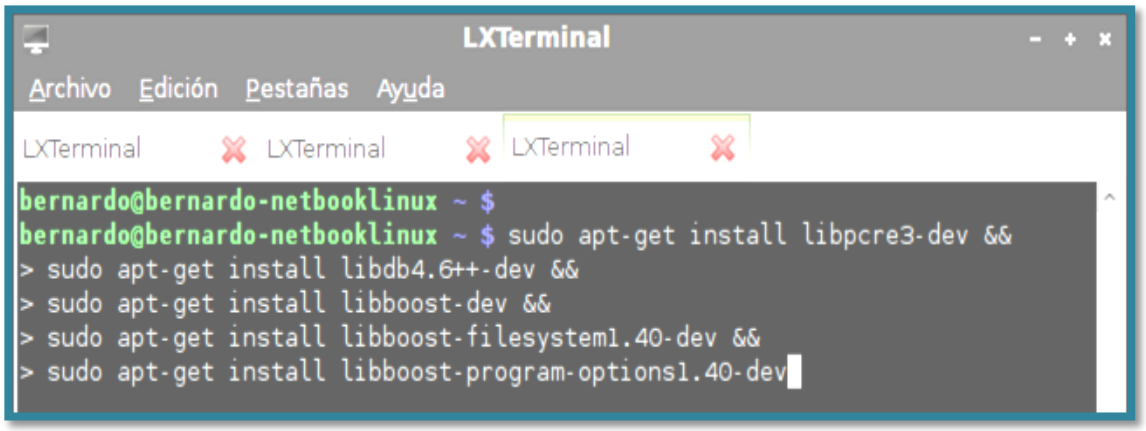


Ilustración 3. Secuencia de instalación de las dependencias necesarias.

Lo más probable es que el nombre de estas dependencias haya cambiado y por tanto *apt* nos indique que no existe alguno de los paquetes arriba expuestos. En dicho caso lo ideal es buscar el nombre actualizado del paquete. Por ejemplo, para el caso de <<libdb4.6++-dev>> (nombre desactualizado) nosotros obtuvimos el nombre actualizado con el comando *"apt-cache search libdb"* de la siguiente forma:

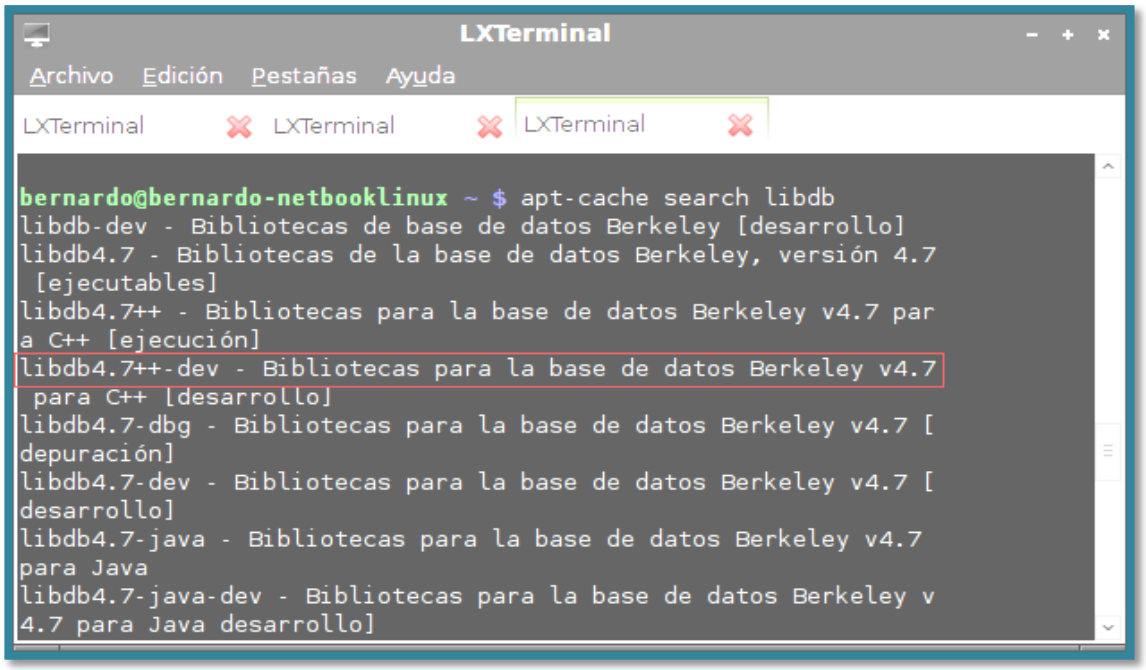


Ilustración 4. Buscando en los repositorios la versión más actual del paquete de desarrollo para libdb

Y entre la lista de resultados que nos devuelve encontramos "libdb4.7++-dev" por lo tanto, al ejecutar el comando "apt-get install libdb4.7++-dev" si nos instalará la dependencia.[16]

Las siguientes 2 dependencias (libfries y libomlet) han de instalarse desde el código fuente, cuya descarga está disponible desde el sitio web de FreeLing.

Tanto para libfries-1.2.1 [33] como para libomlet-1.0.1 [34] el procedimiento de instalación sigue la secuencia estándar de comandos, los mismos que serán explicados más adelante.

[16] En medida de lo posible hay que utilizar específicamente las versiones de dependencias que se indican en el manual de usuario. Solo debe instalarse una versión de dependencia más actual en caso de no existir la versión que el manual de usuario indique.

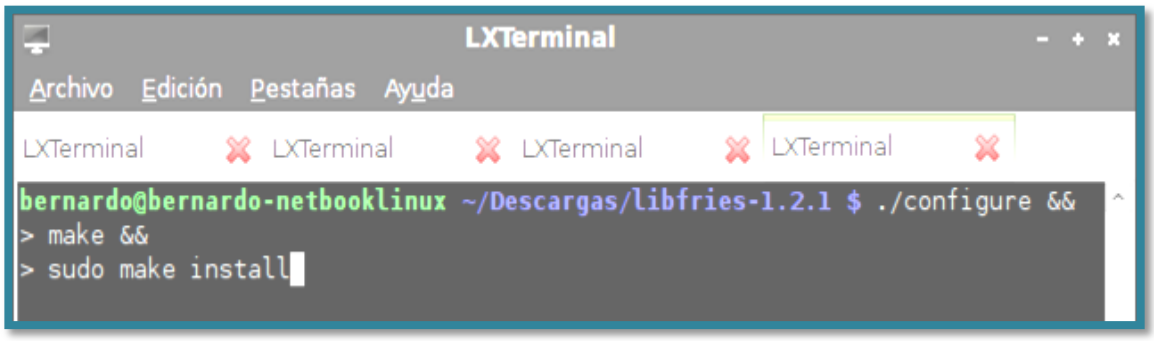


Ilustración 5. Secuencia estándar de comandos para compilar e instalar aplicaciones desde su código fuente.

Procedemos a dar una breve introducción sobre lo que cada uno de estos comandos realiza:

- **Configure:**

  Este procedimiento se encarga de verificar que se cumplen con los requisitos para poder compilar el código fuente. Si nos falta algún requerimiento, por ejemplo una versión especifica de una librería, debemos satisfacerlo antes de continuar, caso contrario no podremos avanzar.

- **Make:**

  El procedimiento más importante de la instalación de cualquier código fuente, es la compilación en sí, y puede durar mucho y presentar ciertos errores. En caso de error el proceso terminará indicándonos el archivo en donde se generó el error y deberemos modificarlo.
  Si todo se ejecutó correctamente no saldrá ningún error en la salida y sólo nos quedará instalar lo que hemos compilado.

- **Make install:**

  Lo que aquí hacemos es simplemente copiar los binarios que hemos compilado a los directorios de nuestro sistema como por ejemplo: /usr/bin.

Una vez que hayamos instalado estas dos librerías copiando sus archivos a la carpeta **/usr/local/lib** (en nuestro caso). Finalmente instalaremos la librería FreeLing, dentro de la misma carpeta **/usr/local/lib**.

Por tanto, descomprimimos nuestro archivo fuente [32] y una vez ahí dentro ejecutamos desde consola la secuencia de comandos estándar que ya especificamos para la instalación de *libfries* y *libomlet.*

Todo esto habrá instalado la librería para ser usada desde línea de comandos, pero, lo que nosotros necesitamos es poder acceder a su API de Java (actualmente en desarrollo) la cual ya mencionamos anteriormente. Es por eso que aún tenemos que realizar una tarea final de instalación.

Dentro de la carpeta **FreeLing-2.2.2/APIs/java** deberemos ejecutar desde consola solamente una instrucción, la cual es **make.** Si todo ha ido bien nos generará un fichero Jar, el cual finalmente será el que agreguemos a nuestro proyecto para poder acceder a las funcionalidades de FreeLing, desde nuestro código Java.[17]

Finalmente para poder usar FreeLing hay que indicar donde se encuentran sus librerías con el siguiente comando: **export LD_LIBRARY_PATH=usr/local/lib/.**

3.2.1.3.2 Sistema de Etiquetado de FreeLing

FreeLing utiliza un sistema de etiquetado estandarizado EAGLES, para realizar su análisis morfológico, a continuación se especifican las categorías [25]:

- Adjetivos
- Adverbios
- Artículos
- Determinantes
- Nombres
- Verbos

[17] Ya sea que tratemos de instalar FreeLing o cualquier otro tipo de código fuente, es muy importante leer el archivo de instrucciones, llamado "readme" o "léeme" que normalmente acompaña al código fuente puesto que en él se detallan aspectos de la instalación que no constan en el sitio web. la herramienta Wordnet ya está incluida en el FreeLing a través del archivo senses30.src.

- Pronombres
- Conjunciones
- Numerales
- Interjecciones
- Abreviaturas
- Preposiciones
- Signos de puntuación.

Después de haber realizado un análisis sobre las palabras de mayor relevancia para nuestro estudio hemos considerado las siguientes categorías: verbos, adverbios, adjetivos, nombres, debido a que aportan mayor carga semántica.

Estas categorías se rigen bajo un estándar “estructura EAGLES” el cual se detalla en los siguientes gráficos:

Pos.	Atributo	Valor	Código
1	Categoría	Adjetivo	A
2	Tipo	Calificativo	Q
3	Grado	Apreciativo	A
4	Género	Masculino Femenino Común	M F C
5	Número	Singular Plural Invariable	S P N
6	Caso	-	0
7	Función	Participio	P

Tabla 5. Estructura EAGLES para adjetivos [25].

Pos.	Atributo	Valor	Código

1	Categoría	Adverbio	A
2	Tipo	General	Q
3	-	-	0
4	-	-	0
5	-	-	0

Tabla 6. Estructura EAGLES para adverbios [25].

Pos.	Atributo	Valor	Código
1	Categoría	Nombre	N
2	Tipo	Común Propio	C P
3	Género	Masculino Femenino Común	M F C
4	Número	Singular Plural Invariable	S P N
5	Caso	-	0

6	Género Semántico	-	0
7	Grado	Apreciativo	0

Tabla 7. Estructuras EAGLES para nombres [25].

Pos.	Atributo	**Valor**	**Código**
1	Categoría	Verbo	V
2	Tipo	Principal Auxiliar	M A
3	Modo	Indicativo Subjuntivo Imperativo Condicional Infinitivo Gerundio Participio	I S M C N G P
4	Tiempo	Presente Imperfecto Futuro Pasado	P I F S
5	Persona	Primera Segunda Tercera	1 2 3

6	Número	Singular Plural	S P
7	Género	Masculino Femenino	M F

Tabla 8. Estructuras EAGLES para verbos [25].

Al momento de realizar consulta se introduce la Forma la misma que será procesada devolviendo una etiqueta, como se puede observar en el siguiente ejemplo:

Forma	Lema	Etiqueta
Cantada	Cantar	VMP00SF
Cantadas	Cantar	VMP00PF
Cantado	Cantar	VMP00SM
cantados	Cantar	VMP00PM

Tabla 9. Etiquetas EAGLES para verbos [25].

FreeLing trabaja con este tipo de estandarización para el manejo de etiquetas, se considera importante trabajar solo con las cuatro categorías nombradas con anterioridad debido a que solo se les aplicará sinónimos a las palabras con mayor carga semántica, lo que incurrirá en menores costos de procesamiento para el computador.

Además del manejo de categorías, hemos creído pertinente emplear el género de una palabra, ya que al ser cambiada por su sinónimo pretendemos que el contexto de la oración no se vea afectado de mayor manera.

3.3 Análisis de resultados

Procedemos a realizar un análisis sobre las herramientas a utilizar y verificar, de acuerdo a los resultados que se obtengan para cada una de ellas, la factibilidad para su uso en la creación de nuestro sistema.

3.3.1 Resultados de FreeLing

Se han logrado los siguientes resultados relacionados con la herramienta:

- Se ha logrado una conexión desde Java con FreeLing
- Se ha probado la detección de carga semántica y la respuesta es favorable.
- Se realizaron pruebas relacionadas con sinonimia e inicialmente los resultados no fueron los esperados, sin embargo se consultó con los desarrolladores del proyecto FreeLing en donde se nos sugirió utilizar estructuras de datos para recuperar directamente la información almacenada en los diccionarios que incorpora la herramienta. Implementando dichas sugerencias, las cuales se detallan en la sección 3.4.1, pudimos finalmente obtener los sinónimos adecuados para cada palabra.

3.3.2 Resultados de FrameNet

Esta herramienta queda descartada para su utilización, debido a que solo posee una interfaz web, siendo más compleja la conexión con nuestra aplicación que será desarrollada en lenguaje Java, que con la herramienta anterior se acopla satisfactoriamente ya que fue desarrollada para ser utilizada con Java, a pesar de que existieron ciertos inconvenientes que lograron ser superados.

3.3.3 Conexiones a Internet

Como desarrolladores del sistema, conocemos de antemano que uno de los posibles cuellos de botella se puede encontrar en la conexión hacia internet, no solo para la realización de las búsquedas, sino también para la extracción del texto de los sitios web que lo permitan, además de la velocidad de descarga de documentos de los sitios web que no nos permiten acceder al texto directamente.

3.4 Selección de herramientas de soporte

3.4.1 Wordnet en FreeLing.

Debido a que el API de Java se encuentra en desarrollo, este tenía errores al momento de obtener sinónimos. Presentamos este problema (y otros problemas relacionados) a los desarrolladores de FreeLing a través del foro de su sitio web obteniendo como respuesta que toda la base de datos de sinónimos estaba contenida dentro del archivo «senses30.src», el cual no era más que un archivo de texto plano.

Este archivo esta formateado de modo que para cada palabra existen varios números que identifican synsets, los cuáles junto a «subparoles» -que a su vez espefican si se trata de verbos, nombres, adjetivos, etc- permite reconocer los sinónimos correspondientes para una palabra dada.

Esto puede entenderse de forma más clara mediante el siguiente diagrama de entidad-relación:

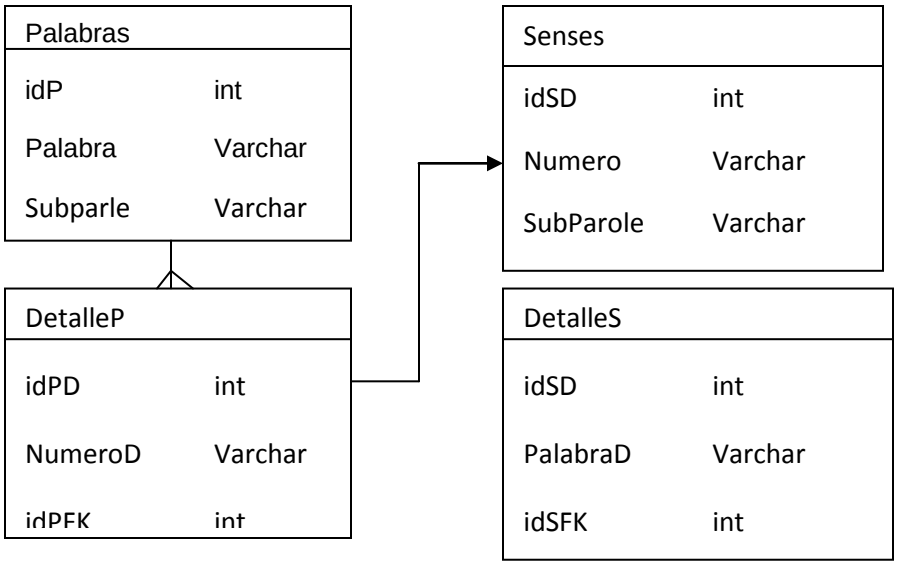


Ilustración 6. Diagrama entidad-relación que explica como recuperamos desde la base de datos la lista de sinónimos correspondientes a una palabra

De tal modo, que para cada palabra podíamos obtener un conjunto de varias *PalabraD*, es decir, sus sinónimos.

La idea ya estaba planteada, lo que quedaba por hacer era implementarla y debido a que el archivo era muy extenso, debíamos utilizar una estructura de datos eficiente. Es así que decidimos recurrir a las bases de datos. Además estas nos proveen varias ventajas extras:

- Generar índices para agilizar las consultas. En nuestro caso no presenta desventajas puesto que los datos no se actualizan, solo se consultan.
- Permiten recuperar los datos de forma sencilla, simplemente mediante una consulta SQL.

3.4.2 Sqlite

La base de datos que hemos decidido utilizar es Sqlite porque además de presentar las ventajas que ya mencionamos es una base de datos que ocupa muy poco espacio y sobre todo no requiere instalación lo cual favorece la portabilidad.
Soporta perfectamente el estándar SQL y tiene un tiempo de respuesta muy bueno.
Para utilizarlo en Java hemos usado el proyecto SQliteJDBC [35]. Solo hay que descargar el archivo Jar y agregarlo a nuestro proyecto.

3.4.3 Python

Utilizamos Python como lenguaje de scripting para realizar dos tareas relacionadas con la parte de sinonimia: migrar la información a la base de datos y cambiar el formato del archivo de EuroWordNet senses30.src. para poder obtener los sinónimos de acuerdo a un esquema adecuado, es decir, que en lugar de presentar la palabra «andar» como sinónimo de "caminando" (lo cual no estaba bien) nos presentara la palabra "andando" que poseía la forma requerida.
La gran ventaja de este lenguaje es que ofrece una rápida implementación y eso era precisamente lo que necesitábamos para realizar este tipo de tarea que se iba a ejecutar una sola vez.

3.4.4 Linux Bash

Muchas veces el primer resultado que devuelven los buscadores no son páginas HTML o similares que contengan texto listo para ser extraído. En varias ocasiones estos primeros resultados son archivos en formatos PDF, DOC y DOCX. los cuales necesitan ser descargados antes de poder acceder a su contenido. Es por eso que utilizamos un

script bash de GNU/Linux que ejecuta el comando wget para poder realizar las descargas. Este script, llamado descargador.sh recibe como parámetros la URL completa del documento y el nombre con el que se llamará al archivo descargado.

3.4.5 wget

Este es un comando GNU/Linux muy potente que nos ha permitido realizar las descargas de los documentos desde internet e incluso poder continuar las descargas interrumpidas, establecer un tiempo de espera en caso de que la descarga no pueda continuar y poder evitar que el programa se quede colgado por una descarga. Este comando acepta como parámetro el número de intentos de descarga; lo que permite evitar bloqueos en la ejecución del programa.

4. DISEÑO DEL SISTEMA

4.1 Análisis de la Implementación.

Para la implementación del prototipo de detección de plagio se ha investigado sobre diversas herramientas y formas en las que se puede llevar a cabo este proceso.

Como primer punto se pretende limitar cuáles serán los formatos de archivos que el programa ser capaz de reconocer, para ello nos hemos planteado tres tipos de archivos por reconocer: PDF, DOC y DOCX, que son los más utilizados en nuestro medio para la presentación de trabajos, deberes, tesis, etc.

Una vez definidos los formatos a trabajar estos deberán ser transformados a texto plano que es la forma más sencilla de trabajar con archivos extensos, este proceso nos servirá para los análisis de plagio por sinonimia y plagio textual.

En lo que respecta a detección de plagio textual se pretende definir un delimitador que nos indique los rangos del texto a analizar, para ello se plantean las siguientes opciones: análisis por frase u oración, análisis por párrafos y análisis por página.

Independientemente de cuál sea el delimitador que se establezca, se procurará que el texto por analizar no contenga referencias, en caso de tenerlas no existirá plagio, ignorando de esta manera el texto referenciado.

Además se deben realizar conexiones con buscadores que permitan obtener datos de las copias textuales que existan y que no contengan las debidas referencias, entre los buscadores escogidos están: Google, Yahoo!, Bing, Ask, y el buscador ruso Yandex; los cuales nos devolverán información en caso de que exista plagio textual, caso contrario no se deberán obtener resultados.

Para la detección de plagio por sinonimia se planifica realizar en primer lugar un tratamiento al texto que se va analizar, para ello se filtraran *stop words,* las cuales no son más que palabras que no aportan verdadero contenido al texto, es decir, no nos indican el tema a tratar en el documento, como ejemplos tenemos*: debe, haber, como, estar, el, uno, etc.*, estas serán filtradas para realizar una búsqueda de plagio no tan rigurosa como se lo realiza en el plagio textual.

Una vez realizado el filtrado de *stop words*, también se debe considerar la eliminación de caracteres especiales (tildes, acentos y signos de puntuación). Para continuar con este proceso es necesario elaborar un esquema en el que se permita dividir todo el texto por N-gramas, el número adecuado que debe llevar N debe ser definido en base a la experiencia que se obtenga en el momento de realizar las pruebas respectivas.

Una vez efectuado este proceso se hace una análisis del texto filtrado para saber cuáles son las palabras con mayor carga semántica, para esto se utiliza la herramienta FreeLing. Cuando ya fueron filtradas todas estas palabras se procederá a escoger solo 32 palabras de todas las obtenidas[18], las mismas que serán escogidas por diferentes criterios: de forma estocástica o a través de la ley de Zipf.

Cuando se haya concluido con el proceso anterior se debe aplicar sinónimos a un subconjunto de los 32 elementos escogidos anteriormente, y se envía al buscador estas palabras para que sean analizadas y encontrar resultados para poder realizar un análisis comparativo entre el documento que se encontró en la web y el documento que se está analizando.

Para la comparación entre documentos se utilizan las medidas de similitud que fueron revisadas en la sección 2.5: coeficientes de Jaccard, Dice, Overlap, Roger y Tanimoto, Sokal y Michener y el coeficiente de Czekanowski.

El objetivo de utilizar estos coeficientes es obtener un resultado sobre el nivel de plagio que existe en un documento, el valor que se puede obtener se encuentra en un rango de 0 a 1. Se considera el valor de 0 como la ausencia de N-gramas iguales, y a 1 como la totalidad de N-gramas iguales.

Para sintetizar todas las ideas que se han expuesto en este apartado presentamos a continuación un diagrama de bloques (Ilustración 7) mediante el cual se observa de forma secuencial como funcionara el sistema.

[18] En base a pruebas efectuadas sobre el buscador Google descubrimos que el número máximo de palabras que este admite es de 32.

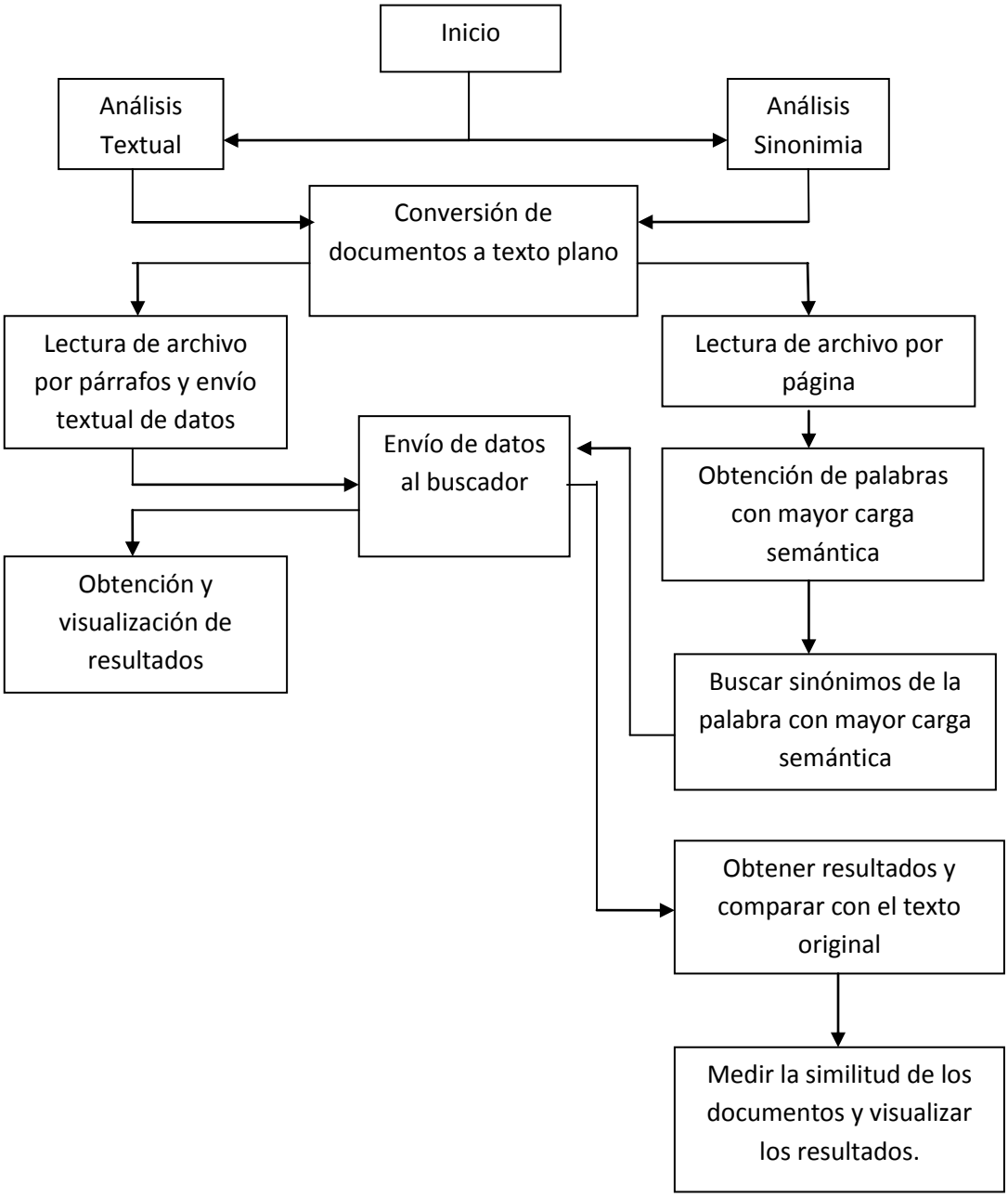


Ilustración 7. Diagrama de bloques que indica el funcionamiento general del sistema.

4.2. Diseño general de la aplicación en UML.

La aplicación ha sido organizada en paquetes para separar cada una de las funcionalidades que provee. A continuación se explica la funcionalidad de cada paquete junto a sus clases más importantes.

4.2.1 Paquete: Calculos

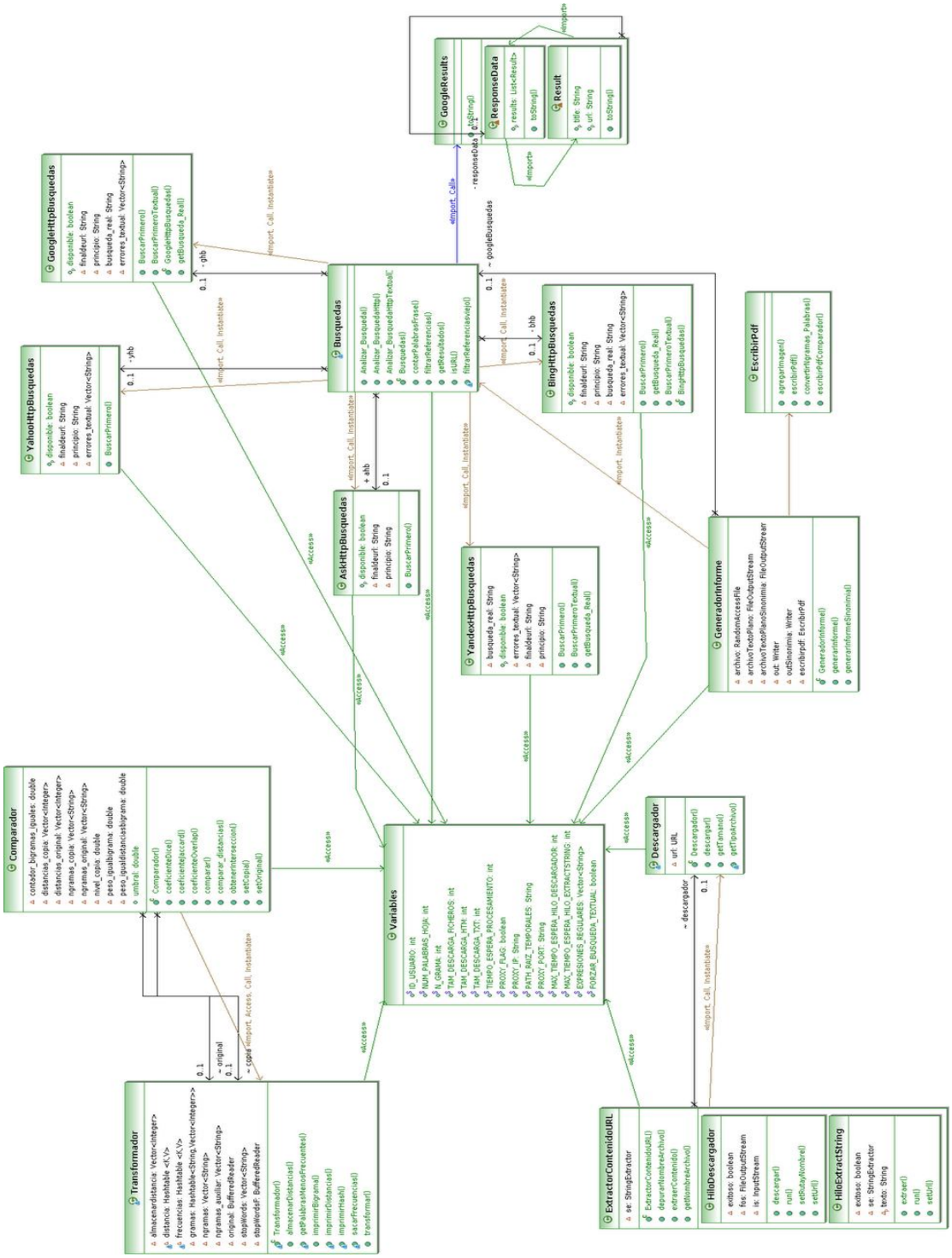


Ilustración 8. Diagrama de clases para el paquete Cálculos.

Este es el paquete más grande de todos y además es donde la verdadera lógica de la aplicación se encuentra. Entre las clases más importantes del paquete tenemos:

- **Transformador**: Esta clase se encarga de limpiar el texto. Es decir, borra *stopwords*. Elimina caracteres mal formados, caracteres especiales y acentos. Finalmente con el resultado forma los gramas (bigramas, trigramas y n-gramas) y los almacena en un vector.
- **Comparador**: Esta clase recibe 2 archivos a ser comparados, luego hace uso de la clase Transformador para obtener los gramas de cada uno de los archivos. A continuación busca los gramas coincidentes entre los 2 archivos probados y almacena estos gramas coincidentes. Finalmente, se obtiene coeficientes (Overlap en nuestro caso) que indican el grado de copia existente entre los archivos.
- **Búsquedas**: Clase utilizada para análisis textual y análisis por sinonimia. Esta es la clase que administra las búsquedas. Para evitar que Google considere que estamos abusando de sus servicios y bloquee nuestras peticiones, nos hemos apoyado en 4 buscadores más:
    - **Bing**: El buscador de Microsoft provee con buenos resultados y es comparable con el servicio de búsqueda de Google. URL: http://www.bing.com.
    - **Yahoo**. No tan a la altura como Bing, sin embargo cumple con su trabajo. No siempre encuentra resultados, pero los encontrados resultan afines con la búsqueda. URL: http://www.yahoo.com
    - **Ask**: Útil solo para búsquedas textuales para las cuales si encuentra sitios coincidentes. No es muy bueno al enviarle un conjunto de palabras clave para búsqueda, debido a que en caso de encontrar resultados no siempre son muy relevantes. URL: http://www.ask.com
    - **Yandex**: Un excelente buscador ruso que puede realizar búsquedas en español. Hemos decidido utilizarlo debido a que ofrece buenos resultados tanto para búsqueda con sinonimia como para búsquedas textuales. Sobre este buscador destacamos lo siguiente: En caso de no encontrar resultados textuales no devuelve ningún resultado y esa característica lo hace sobresalir sobre Google en lo referente a

búsquedas textuales debido a que este último elimina las comillas y realiza una búsqueda no textual. URL: http://www.yandex.ru

Debido a que estos 4 buscadores funcionan bien en búsquedas textuales decidimos no usar Google para este tipo de búsquedas reservándolo para explotarlo en análisis por sinonimia. Asimismo, para análisis por sinonimia evitaremos usar el servicio de Ask debido a su pobre desempeño para esta labor. Finalmente debemos destacar que Google es el único buscador que devuelve resultados para 32 palabras por búsqueda. Los demás buscadores llegan hasta máximo 24 palabras por búsqueda.

De los resultados de búsqueda hemos de tomar solo la primera incidencia puesto que es la más afín a la búsqueda, las demás coincidencias se ignoran y se procede a la siguiente búsqueda.

- **ExtractorURL**: Esta clase es usada para el plagio por sinonimia, en el cual se necesita recuperar contenido desde Internet, ya sea contenido de un sitio web o ficheros TXT, DOCX, PDF o DOC, para lo cual recibe la URL objetivo y trata de extraer su contenido. Debido a que la velocidad de conexión puede ser muy baja al programa le puede tomar mucho tiempo intentar extraer contenido, es por eso que para que el sistema no se quede bloqueado por una descarga lenta hemos implementado un par de hilos que serán los encargados de extraer el contenido desde Internet. En caso de que haya pasado un cierto tiempo delimitado por el usuario sin tener éxito en la descarga de contenido entonces un hilo controlador dará por terminadas dichas descargas y procederá con la ejecución del programa; esta última medida evidentemente podría producir ficheros mal descargados y esto a su vez en errores. En caso de presentarse uno de estos errores el programa dará por ignorada la URL inicial y procederá con su normal funcionamiento. Para asegurarse que este tipo de errores no ocurran el servidor deberá contar con una buena velocidad de conexión.
- **GeneradorInforme**: Una vez que el/los análisis haya(n) terminado de realizarse, se recolectara la información obtenida y se almacenara en un archivo TXT y PDF una lista de URLs encontradas, junto con el texto sospechoso de copia y un valor de coeficiente de Overlap asociado (para el caso de análisis por sinonimia).

4.2.2 Paquete:IngresoDatos

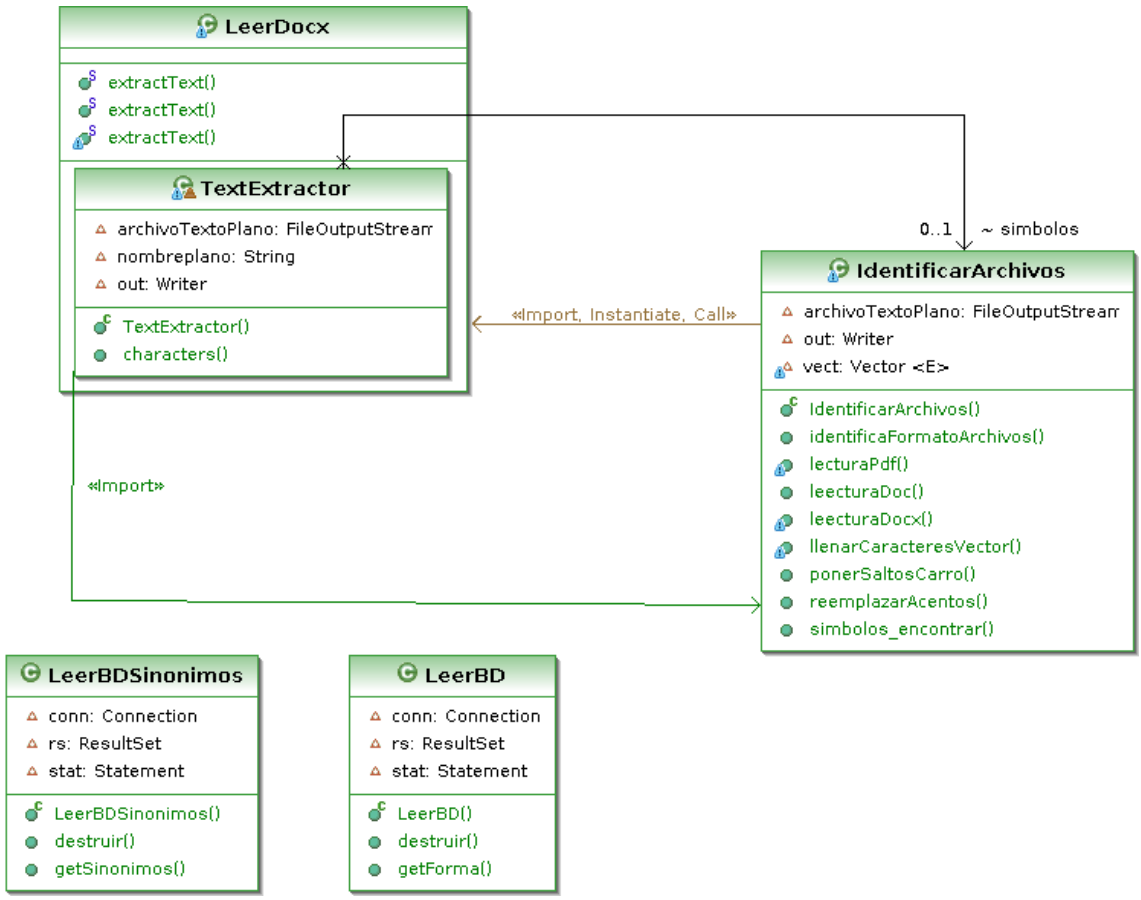


Ilustración 9. Diagrama de clases para el paquete IngresoDatos.

Este paquete contiene las clases necesarias para convertir los archivos origen en archivos de texto plano, de tal manera que sean más fáciles de procesar por el sistema. Además implementa las clases que gobiernan el acceso a las bases de datos que son utilizadas para el análisis por sinonimia. Se detallan sus clases más importantes a continuación:

- **IdentificarArchivos**: Esta clase tiene la función de identificar si el archivo que estamos por analizar esta en formato DOC, DOCX o PDF y según el caso enviara al archivo al método indicado para su conversión a TXT. Para cada formato de archivos se utiliza una librería especifica que nos ayuda en la

conversión; de hecho, para el formato DOCX se utiliza una clase aparte. Al final se genera un archivo TXT con el mismo nombre del archivo original.

- **LeerBDSinonimos**: Esta es la clase que se conecta con la base de datos de sinónimos. Recibe una palabra y devuelve un *synset* para dicha palabra.
- **LeerBD**: Esta clase se conecta con otra base de datos que nos devuelve la forma correcta que debe tener un sinónimo devuelto por la clase LeerBDSinonimos. Por ejemplo, si solicitamos un sinónimo para caminando, el *synset* nos devolverá andar, es decir, la forma base del sinónimo, sin embargo, pero nosotros necesitamos que nos devuelva la forma correcta: *andando*. Esta clase se encarga de devolvernos la forma que buscamos para el sinónimo dado.

4.2.3 Paquete:Textual

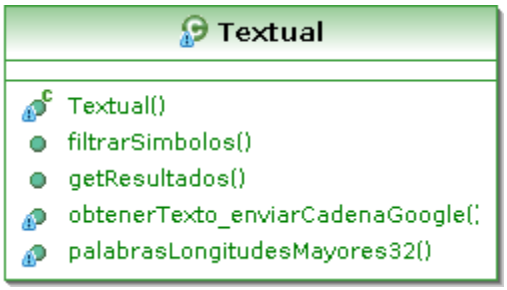


Ilustración 10. Clase Textual.

Como se observa, este paquete posee solo una clase, la cual va tomando del texto bloques de máximo 32 palabras seguidas y las envía a la clase Búsquedas para que se encargue de encontrar una URL que coincida textualmente con dicho bloque. Para que la búsqueda sea textual se delimita el bloque entre comillas. Realiza iterativamente este proceso hasta que ha terminado de recorrer el archivo. Implementa un método que obtiene los resultados de las búsquedas para generar un informe final mediante la clase GeneradorInforme del paquete Cálculos.

4.2.4 Paquete: Sinonimia

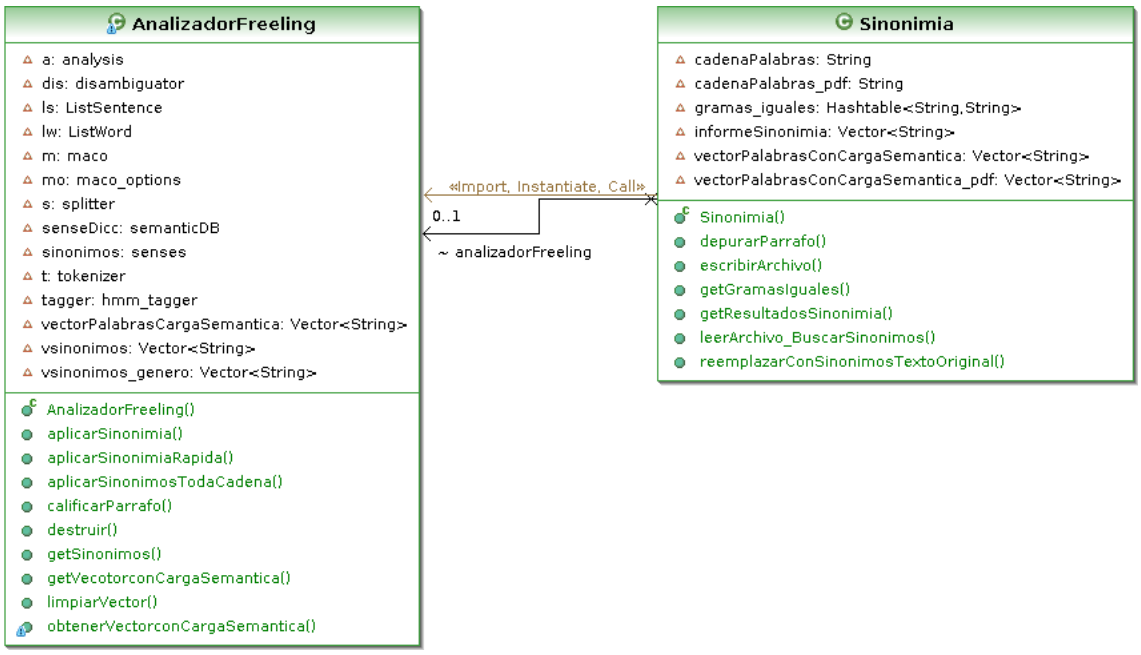


Ilustración 11. Diagrama de clases para el paquete Sinonimia.

Este paquete es el encargado de utilizar las funcionalidades que nos provee la librería FreeLing para realizar el análisis de nuestro texto y posteriormente obtener los sinónimos. Para ello hemos implementado la clase AnalizadorFreeling. Se explican a continuación las clases que forman parte del paquete:

- **AnalizadorFreeling:** Esta clase hace uso del API para Java de FreeLing, en especial su función para categorizar palabras y entregarnos sus etiquetas EAGLE correspondientes. Una vez que nos indica si se trata de un verbo, un adverbio, nombre o adjetivo procedemos a buscar con esos datos -más la palabra- el respectivo *synset* en la base de datos que hemos adaptado para dicho fin.
- **Sinonimia:** La principal función de esta clase es recorrer el texto a analizar, dividirlo en partes más pequeñas como **páginas** y para cada página buscar una lista de palabras más relevantes. Para ello utiliza el método de Zipf el cual nos indica que las palabras más relevantes son las más largas. Una vez que obtenemos nuestra lista de palabras (máximo 32 palabras) procedemos a enviarlas a los buscadores en forma de una frase con *tags* o palabras clave. Es

muy probable que dicha frase no tenga ningún sentido como unidad, lo que nos interesa es encontrar los sitios que coincidan con la mayor cantidad de dichas palabras. Apenas se haya encontrado un sitio coincidente se procede a extraer su contenido desde internet para posteriormente convertirlo a n-gramas mediante la clase *Transformador* del paquete *Calculos* y luego encontrar los gramas coincidentes entre nuestra página actual y el contenido extraído desde Internet mediante la clase *Comparador* del mismo paquete *Calculos*. Al final se obtiene un coeficiente de Overlap para dicha página y se almacena para el informe final; luego, proseguimos con las siguientes hojas hasta terminar con el documento.

4.2.5 Paquete: Principal

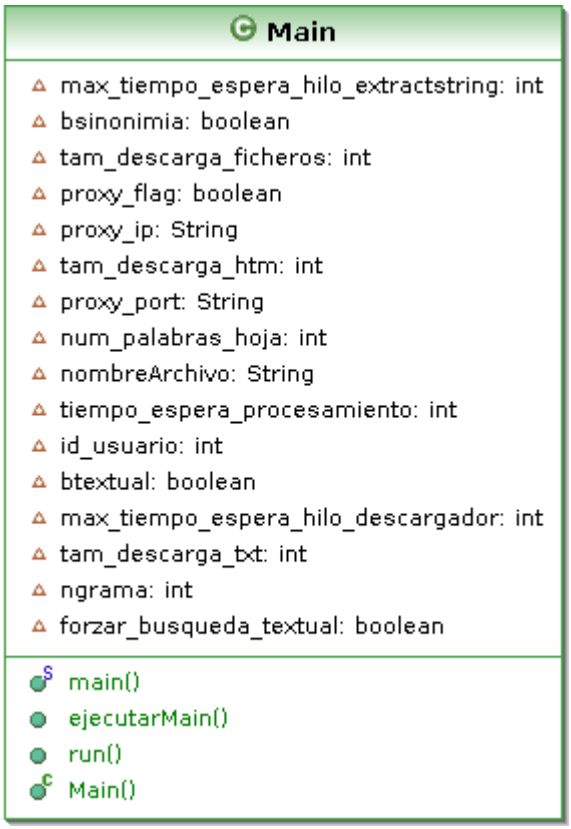


Ilustración 12. Clase principal.

Esta clase es la encargada de recibir y almacenar los parámetros que envía el usuario para poder correr el programa. Mediante los parámetros que el sistema recibe

podremos modificar los tiempos de espera en caso de que poseamos una conexión muy lenta, la cantidad de palabras por página, tamaños de descarga permitidos, el valor de los gramas (bigramas, trigramas y n-gramas en general) entre otras configuraciones.

Todos los parámetros son muy descriptivos por sí mismos, sin embargo adjuntamos un breve detalle para cada uno de estos:

- **TAM_DESCARGA_FICHEROS**. Establece el tamaño máximo en bytes de los archivos a descargar.
- **TAM_DESCARGA_TXT**. Establece el tamaño máximo de los archivos de texto plano que se descargaran.
- **TAM_DESCARGA_HTM**. Establece el tamaño máximo de los sitios web que se descargaran.
- **NUM_PALABRAS_HOJA**. A menor tamaño el análisis tiene más oportunidades de obtener coincidencias (es más agresivo) pero más lento, a mayor tamaño el análisis es más rápido pero menos agresivo.
- **TIEMPO_ESPERA_PROCESAMIENTO**. Es el tiempo que se le permite a la clase Transformador procesar un archivo. Este tiempo de espera debe estar dado en milisegundos.
- **N_GRAMA**. Numero de N-gramas a analizar.
- **MAX_TIEMPO_ESPERA_HILO_EXTRACTSTRING**. Tiempo de espera en milisegundos del hilo que extrae texto de los sitios web, por defecto el hilo espera 1 minuto.
- **MAX_TIEMPO_ESPERA_HILO_DESCARGADOR**. Tiempo de espera en milisegundos del hilo descargador, usado para descargar PDF, DOC y DOCX. Por defecto el hilo espera 2 minutos.
- **ID_USUARIO**. Nombre de la carpeta personal del usuario, para evitar conflictos entre usuarios concurrentes. Esto es debido a que se crean archivos temporales y se desea evitar que la instancia de un usuario elimine los archivos temporales de otro usuario. Esto puede ocurrir si todos los usuarios comparten el mismo

directorio de trabajo. Es por eso que se decide asignar a cada usuario un id -que a su vez representa el nombre de su carpeta- para separar los archivos temporales de los usuarios.

- **EXPRESIONES_REGULARES**. Establece un conjunto de expresiones regulares gracias a las cuales el sistema identificará referencias en el texto. Los elementos de este conjunto de expresiones se separan mediante el uso de llaves, por ejemplo: *{.\\d+}{\\'\\d+}*.

Los siguientes parámetros están disponibles en caso de que se quiera instalar el sistema en un servidor distinto:

- PROXY_FLAG. Valor lógico que indica si se usa proxy o no.
- PROXY_IP. En caso de usar proxy, indica su dirección IP.
- PROXY_PORT. En caso de usar proxy, indica el puerto.
- PATH_RAIZ_TEMPORALES. En este *path* debe estar el jar. Los archivos de base de datos y los scripts para descargas.

Todos estos parámetros son almacenados como variables en la clase *Variables* del paquete *Calculos* y, como se aprecia en el diagrama de dicho paquete, son usadas por todo el sistema.

Finalmente, cabe acotar que esta clase implementa la interfaz Runnable de tal modo que pueda correr en un hilo independiente, esto nos es útil en la parte web del sistema ya que mientras se ejecuta el programa no queremos que el sitio parezca colgado, por eso lo ejecutamos en un hilo separado de la presentación web.

4.2.6 Dependencias

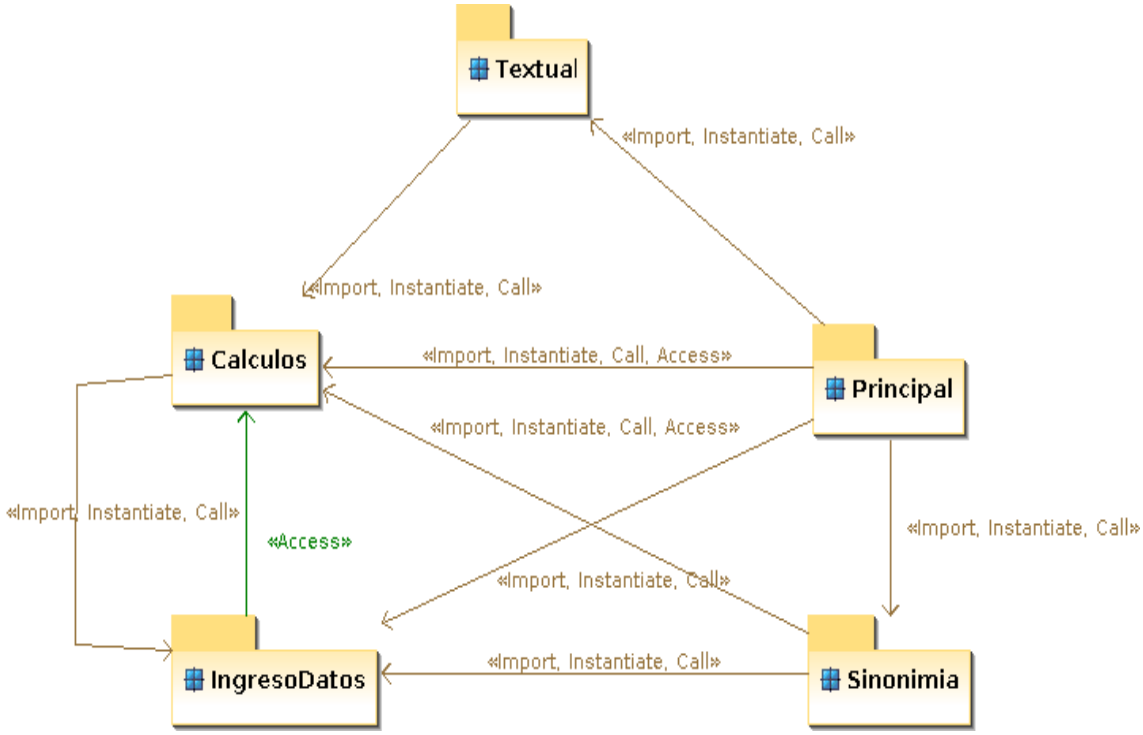


Ilustración 13. Diagrama de dependencias entre los paquetes que conforman el sistema.

Finalmente en este diagrama (Ilustración 13) se detallan las dependencias existentes entre los paquetes que conforman el sistema.

4.3. Diseño del esquema de conexión con el motor de búsqueda.

Para todos los buscadores se utiliza el mismo esquema logrando que todos funcionen de igual forma. A continuación se detalla mediante un diagrama de flujo el funcionamiento del motor de búsqueda (Ilustración 14):

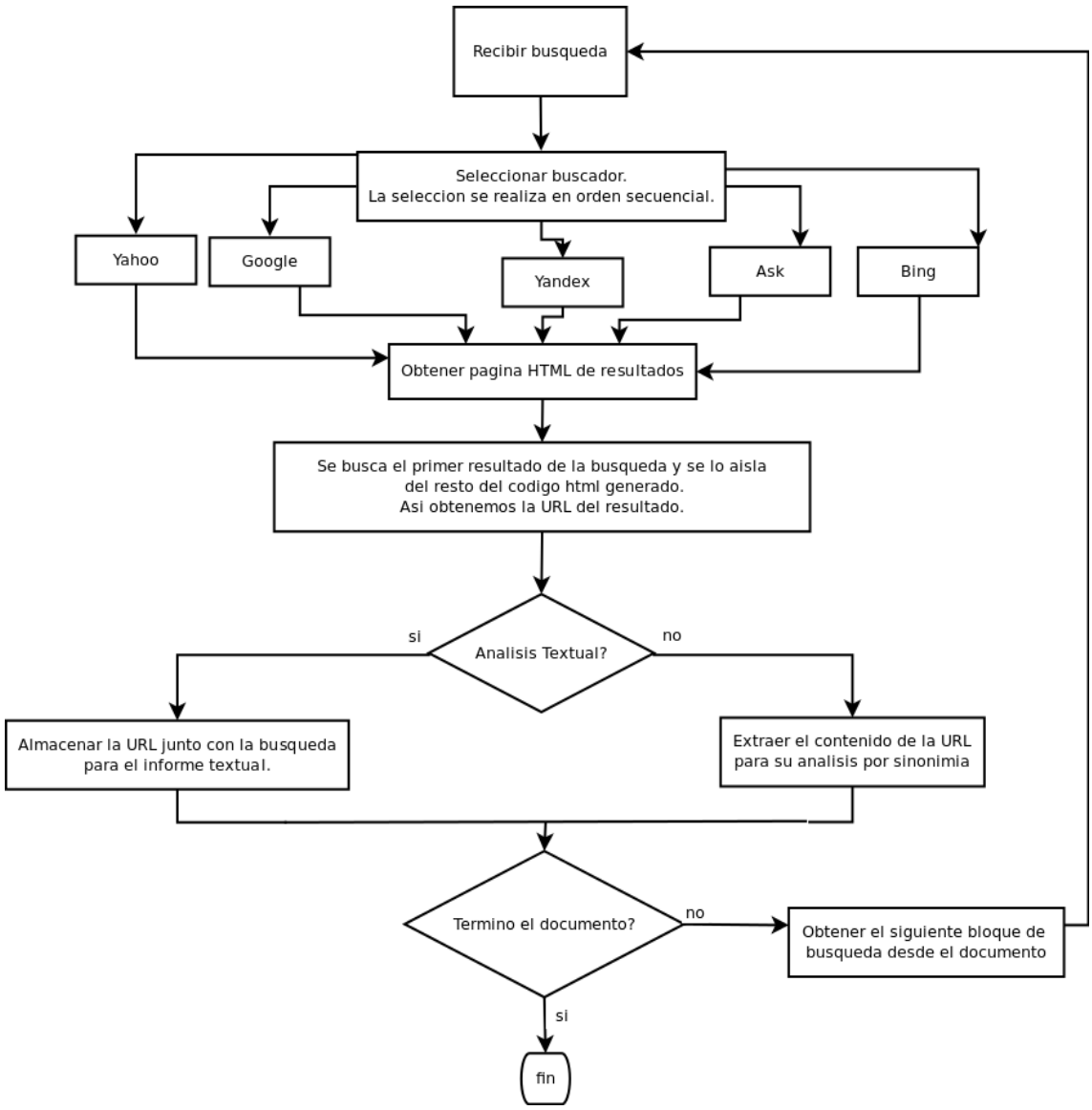


Ilustración 14. Diagrama de flujo que detalla el funcionamiento del motor de búsqueda.

Como se observa en la ilustración 14, para cada búsqueda que debamos realizar se selecciona un buscador. Esta selección se realiza siempre de la misma forma: primero buscamos por Yahoo, luego por Google, en tercer lugar por Ask, luego por Bing y

finalmente por Yandex. Luego se repite el ciclo de selección. Hemos de hacer notar una excepción en este punto: si el análisis que estamos realizando es textual entonces evitaremos usar Google para que este no considere a nuestro sistema como abusivo, sin mencionar el hecho de que Google elimina las comillas de las búsquedas textuales cuando no hay resultados y vuelve a realizar una búsqueda que no es textual; en cambio, si el análisis es por sinonimia evitaremos usar Ask debido a su pobre desempeño en búsquedas no textuales.

Ahora, de presentarse el caso en el que alguno de los buscadores considere a nuestro sistema como abusivo y, en consecuencia, decida restringirnos el acceso a sus servicios, el sistema desactivará a dicho buscador y procederá a realizar las búsquedas solamente con los buscadores sobrantes. Se mantendrá dicha política hasta que el sistema haya terminado su ejecución. La próxima vez que se inicie volverá a intentar con todos los buscadores en espera de que ya no consideren abusivo al sistema.

Cuando la búsqueda haya sido realizada con éxito podremos obtener el código HTML de la página con los resultados. De todo este código HTML tendremos que identificar los resultados de búsqueda y aislarlos. Para cada buscador existe un código inicial y final que delimita cada resultado de la búsqueda. Por ejemplo, para Google estos códigos son los siguientes:

```
String principio="a href="+'\u0022'+"/url?q=";
String finaldeurl="&amp;";
```

Ilustración 15. Extracto del código fuente para realizar búsquedas con Google

Ahora bien, estos delimitadores son válidos durante el tiempo que esta tesis ha sido elaborada, sin embargo, no se garantiza que no vayan a cambiar. En caso de que esto suceda se deberá compilar nuevamente el proyecto cambiando las variables *principio* y *finaldeurl* de la clase asociada al buscador que ha presentado el cambio. Para saber cuáles son los nuevos delimitadores se deberá analizar el código HTML de una página de resultados del buscador y a continuación buscar dichos delimitadores comunes a todos los resultados de búsqueda.

Depende del tipo de análisis que se ha llevado a cabo para realizar el siguiente paso. Si el análisis es textual sencillamente se almacenara la URL devuelta por el buscador junto con el texto original correspondiente para su presentación en el informe final.

En caso de que el análisis se realice por sinonimia la URL devuelta es procesada. Se extrae el contenido de la URL y se procede a comparar dicho contenido con una página del texto original (cada página del texto original necesita una búsqueda nueva) con el fin de encontrar n-gramas coincidentes.

Finalmente, si el texto original aún no ha sido completamente procesado se obtiene un nuevo bloque o página para ser buscado y todo el proceso se repite. Caso contrario se dan por terminadas las búsquedas.

4.4. Especificación de los módulos de trabajo con N – Gramas.

Usamos n-gramas específicamente en 2 clases de nuestro proyecto, las cuales son: *Transformador* y *Comparador*.

Se puede notar que dichas clases forman parte del paquete *Sinonimia* por lo cual el uso de n-gramas, evidentemente, está limitado solo a este tipo de análisis.

La primera clase que utiliza n-gramas es la clase **Transformador** y esta es la encargada de crear y almacenar en un vector todos los N-gramas existentes en el documento. En esta clase se filtran las *stop words* de las cuales previamente ya se habló, por lo tanto los n-gramas almacenados no poseen ninguna de estas *stop words.*

Una vez que se ha transformado el texto en n-gramas tanto para la página actual que se está analizando como para el contenido extraído desde Internet correspondiente a dicha página y se han almacenado estos dos conjuntos de n-gramas en sus respectivos vectores se procede a llamar a la clase **Comparador**, la cual recibe estos 2 vectores para comparar sus n-gramas. En caso de encontrar N-gramas coincidentes entre los vectores estas coincidencias son a su vez almacenadas en un nuevo vector que almacenara los N-gramas comunes entre los 2 textos a comparar.

Las coincidencias que se hubiesen encontrado nos servirán para el cálculo de similitud para lo cual se utilizara uno de los coeficientes revisados en la sección 2.5, el cual dará una respuesta con un rango entre 0 y 1, este valor nos indicara el nivel de similitud posee un documento con respecto a otro.

Finalmente es necesario mencionar que los documentos a comparar deben transformarse utilizando un mismo valor de “N” para su análisis con n-gramas.

4.5. Especificación de los módulos de conexión con herramientas de soporte.

Debido a que nuestro sistema recurre a su vez a otros sistemas para realizar ciertas tareas procedemos a especificar a continuación como nos conectaremos con dichos sistemas para acceder a sus servicios.

4.5.1 FreeLing.

En principio pretendíamos acceder a las bondades de esta herramienta a través de la consola del sistema operativo mediante la instrucción:

Runtime.getRuntime.exec(analyze 'palabra'.).

Sin embargo, mediante este enfoque perdíamos control sobre la aplicación además de ser poco flexible para las tareas que necesitábamos. Es por esto que vamos a optar por utilizar el API de Java que ya hemos mencionado.

4.5.2 Sqlite:

En nuestro proyecto existen 2 conexiones a esta base de datos embebida las cuales han sido encapsuladas en 2 clases distintas que comparten el código de la conexión, salvo el nombre de la base de datos.

Primero hemos tenido que descargarnos el JAR que nos permite acceder a esta base de datos desde el sitio oficial del proyecto "SqliteJDBC"[19] y agregarlo al proyecto. Luego, el procedimiento es similar al de cualquier base de datos.

- Indicamos el driver a usar con la línea: *Class.forName("org.sqlite.JDBC");*
- Indicamos el JDBC que usaremos más el nombre de la base de datos a la cual vamos a acceder con la línea: *jdbc:sqlite:nombreBD*
- Utilizamos las clases *Statement* y *ResultSet* de la misma forma estándar que se aplicaría a cualquier otra base de datos.

[19] http://www.zentus.com/sqlitejdbc/

4.5.3 Wget.

En fases tempranas del sistema se pretendía obtener documentos desde internet a través del comando wget disponible en GNU/Linux y desde ahí controlar los "timeouts" de las descargas, sin embargo los resultados no siempre eran positivos por lo que optamos realizar las descargas desde el mismo código Java.

Previamente habíamos logrado realizar descargas con esta herramienta a través de un archivo de comandos llamado *descargador.sh* en el cual se incluía información sobre el proxy que se debía usar, a continuación se realizaba un llamado a wget con los parámetros "URL origen" y "Nombre de salida del archivo". Estos parámetros se enviaban desde Java.

Mediante los parámetros *timeout* y *tries* de *wget* reducíamos el tiempo de espera a 30 segundos y los intentos de descarga a 1.

A continuación adjuntamos el código que usábamos en nuestro archivo: *descargador.sh*

```
export http_proxy="172.16.0.129:3128"
wget -c $1 -O $2 --timeout=30 --tries=1
echo "finalizado"
```

Ilustración 16. Secuencia de comandos GNU/Linux que se utilizaban originalmente para realizar la descarga de ficheros.

Tal como ya ocurrió en el punto 4.5.1 decidimos obtener el control de las descargas mediante nuestro código Java, por lo que implementamos un conjunto de hilos dentro de la clase *ExtractorContenidoURL* cuyo objetivo es realizar las respectivas descargas de ficheros. La forma en la que realizamos las descargas desde Java es conectarnos con la URL del fichero y mediante un bucle leemos y escribimos en un fichero local todos los bytes del fichero existente en Internet.

4.6. Diseño del plan de experimentación

Con el objetivo de medir la eficacia y la eficiencia del sistema se lo pondrá a prueba. Esta prueba consiste en presentar varios documentos al sistema, los cuales a su vez contendrán texto plagiado con y sin referencias. Llamaremos corpus a este conjunto de documentos de prueba. Este corpus estará compuesto por documentos que pueden tratar diferentes temáticas, poseer diferentes extensiones de contenido, etc.

Vamos a utilizar cierta metodología para realizar nuestras pruebas, para lo cual especificaremos ciertos aspectos como: los resultados que deseamos obtener de nuestras pruebas, especificar variables que nos resulten de interés, es decir, que nos interesa medir, posibles problemas que podamos encontrar al realizar nuestra experimentación, etc. Finalmente analizaremos estos resultados para medir la eficiencia y eficacia del sistema.

A continuación se presenta el diseño del plan de experimentación que se debe realizar, los corpus con los que se experimentarán se encuentran disponibles en la sección 3.1.

Resultados:

Obtener porcentajes de plagio lo más afines a la realidad.

Plan de Experimentación. Variar los siguientes parámetros.

- Extensión del documento.
- Nivel de plagio en un documento.
- Temática de los documentos.
- Número de palabras por página que serán analizadas.
- Tiempo que toma en descargar un archivo.
- Tiempo que toma comparar un archivo.
- Valor de N de los N-gramas.
- Tamaño de descarga de un archivo.

Tiempo disponible:

- Tiempo que retorne resultados.

Variables de Interés.

- Tiempo de retorno de resultados.
- Exactitud del análisis.

Perturbación.

- Tiempos de procesamiento y descargas demasiado largos.
- Conversión no satisfactoria de los documentos a texto plano.
- Referencias y bibliografía no especificadas.
- Velocidad de Conexión.

Tratamiento estadístico de los resultados.

- Análisis de precisión.
- Análisis de cobertura.
- Análisis de F-measure.
- Calculo del AVP (Average Precision)

Complejidad de la Interfaz

La interfaz a realizar debe ser lo más sencilla posible de tal manera que sea fácil de utilizar para el usuario.

5. DESARROLLO DEL PROTOTIPO

5.1. Implementación del prototipo de detección de plagio

A pesar de diseñar la aplicación, probar las herramientas e investigar sobre las técnicas de detección de plagio al implementar el sistema pueden presentarse escenarios que no hayan sido considerados con anterioridad. A continuación presentamos detalles sobre la implementación que no fueron contemplados al momento de diseñar la aplicación.

5.1.1 Incorporar referencias y bibliografía para el análisis mediante.

Expresiones regulares.

Existen diversas formas de realizar referencias de un texto, por ejemplo: la Universidad Politécnica Salesiana posee un documento denominado “instructivo de graduación” donde se especifica la manera de realizar citas bibliográficas, otro ejemplo es el “estilo Vancouver” en el que se define como realizar las referencias bibliográficas en el ámbito de Medicina o Ciencias de la Salud.

Debido a la variedad de normativas para realizar referencias hemos creído conveniente que éstas sean incorporadas por el usuario, sin embargo, para que estas sean agregadas el usuario deberá conocer de expresiones regulares, las mismas que deben ser puestas entre “llaves” <<{..Expresión regular..}>>. El programa incluye cuatro expresiones regulares por defecto en caso de que el usuario no desee añadir más, estas son:

- **\\[\\d+\\]\\.** Valor numérico contenido entre corchetes y finalizado en punto. Por ejemplo: *[2].*
- **\\.\\d+** Punto seguido de un valor numérico. Por ejemplo: *.4*
- **\\’\\d+** Comillas seguidas por un valor numérico. Por ejemplo: *“3*
- **\\.\\s*\\[\\d+\\]** Punto seguido de algún o ningún espacio, finalmente seguido por un valor numérico contenido entre corchetes. Por ejemplo: *. [1]*

Todas estas formas de referencias pueden estar presentes en cualquier lugar del párrafo que se está analizando.

5.1.2 Librerías

Para la obtención de datos de archivos PDF hemos utilizado las librerías: PDFBox e iText, las que permiten leer y escribir archivos en este formato.

Para la lectura de archivos DOC utilizamos la librería APACHE POI que se encargará de seguir el estándar Office Open XML (OOXML) que permite manipular diversos formatos desde Java.

Para la manipulación de archivos DOCX se ha utilizado una variedad de librerías que pertenecen al paquete docx4j y todas las librerías adjuntas a éste.

5.1.3 Conexión con los buscadores

Inicialmente se pensó en utilizar Google como buscador oficial, pero debido a la velocidad con la que se realizan las consultas Google detecta nuestro sistema como un robot y a su vez nos bloquea el acceso a su servicio, es por esto, que se utilizaron varios buscadores, entre estos; Yandex, Bing, Ask, Yahoo y Google. De esta manera se alterna con cada uno de estos buscadores disminuyendo de cierta forma la velocidad de las consultas, cabe recalcar que Bing es el buscador que nos permite seguir realizando múltiples búsquedas sin bloquearnos.

5.1.4 Hilos

El tiempo de respuesta de nuestro sistema se ve limitado por diversos factores entre estos se encuentra la conexión a Internet, el tiempo de descarga de archivos, el tamaño del archivo, entre otros, lo que se pretende con nuestro sistema es agilizar el tiempo de respuesta, se elaboró la primera versión de nuestro sistema en el que se obtuvieron resultados bastante acertados pero en tiempos largos, estos tiempos serán mencionados con más detalle en la sección 5.3.

Debido a esta situación, se vio la necesidad de mejorar los tiempos de respuesta en las búsquedas que se realizan utilizando hilos, sin embargo, no se pensó en primera instancia en esta alternativa ya que los buscadores al detectar altas velocidades para las consultas procede a lanzar excepciones como ya se mencionó, sin embargo, luego de realizar pruebas con los buscadores, nos percatamos que el buscador Bing no nos bloqueaba el acceso a sus servicios a pesar de realizar muchas consultas en paralelo a sus servidores.

Esto fue lo que originó la idea de plantear una versión 2 del sistema que permite agilizar los tiempos de respuesta utilizando hilos para las conexiones, limitándonos a utilizar un *FixedThreadPool*[20] de 5 conexiones simultaneas, cabe mencionar que es necesario limitar el pool ya que de no ser así, realizaremos un gran número de conexiones a internet lo que ocasionaría que la conexión se vea afectada.

El sistema en la actualidad presenta un mejor rendimiento con respecto a velocidad y a precisión, toda esta información se ve respalda en la sección 5.3.

5.1.4 Detección Local

Esta es una de las opciones que se incorpora a nuestro sistema, la detección local consisten en subir al sistema documentos los cuales serán comparados, y devolverá dos informes indicando las partes que son exactas entre estos documentos además se calculará el porcentaje de similitud que existen entre estos documentos.

[20] Objeto que representa un pool de hilos a ejecutar, con la particularidad de que provee con la capacidad de limitar la cantidad máxima de creación de hilos. Para más información léase: http://techgarbage.wordpress.com/2010/04/01/thread-pools-en-java-2/

5.2. Pruebas de funcionamiento.

Se han realizado pruebas a lo largo del presente trabajo, se han variado tiempos de comparación y descarga, tamaños de archivos, tamaño para n-gramas, se ha visto la necesidad de eliminar las tildes y caracteres especiales que dificultan la obtención de información, actuando como perturbaciones al momento de realizar las pruebas necesarias, además el tamaño de las cadenas que se envían al buscador han tenido que modificarse para poder tener una mayor oportunidad de que el texto plagiado sea encontrado por nuestro sistema. Con la finalidad de realizar nuestras pruebas hemos diseñado y descargado varios corpus para probar el adecuado funcionamiento del sistema.

Por ejemplo uno de los documentos disponibles que tenemos en nuestro corpus es "sist_heredados.doc" diseñado por nosotros y cuyo contenido abarca plagio textual, obteniendo los siguientes resultados:

- **Tema:** Sistemas heredados
- **Tiempo:** 3 minutos
- **Extensión:** corto
- **Número de hojas:** 2 hojas

Realizando un análisis previo del documento se ha podido cuantificar la cantidad de contenido plagiado que existe en el mismo, este experimento fue realizado de forma manual para ser comparado con el obtenido por el sistema.

El documento posee 11 bloques de texto que contienen plagio de los cuales el sistema detecto 5 bloques, con estos datos podremos además realizar los respectivos cálculos de *Precision, Cobertura y F-Measure*.

5.3 Ejecución del plan de experimentación

Gracias al plan de experimentación realizado en la sección 4.6 hemos podido medir el funcionamiento de nuestro prototipo. Los datos obtenidos son útiles para determinar las áreas en las que se puede mejorar el prototipo así como también determinar la precisión del sistema. A continuación presentamos los resultados del experimento.

Documento	Categoría	Extensión (No. hojas)	N-gramas	No. Palabras /hoja	Análisis			Tiempos de Respuesta (minutos)
					Sinonimia	Textual	con reducción	
estudio.doc	corto	8	4	50	si	si	si	23
MYRNA_estudiosdecaso.pdf	medio	37	4	50	si	si	si	20
3historias.pdf	corto	8	4	50	si	si	si	4
aldea.pdf	corto	1	4	50	si	si	si	1
fantasmas.pdf	corto	8	4	50	si	si	si	45
otras_palabras.pdf	extenso	102	4	50	si	si	si	110
sist_heredados_sinonimia.doc	corto	2	4	50	si	si	si	16
sist_heredados_textual.doc	corto	2	4	50	si	si	si	12
Agentes Inteligentes_sinonimia.docx	corto	4	4	50	si	si	si	17
Agentes Inteligentes_textual.docx	corto	4	4	50	si	si	si	6
DSS_sinonimia.docx	corto	4	4	50	si	si	si	12
DSS_textual.docx	corto	4	4	50	si	si	si	11
Motivacion_sinonimia.docx	corto	4	4	50	si	si	si	9
Motivacion_textual.docx	corto	4	4	50	si	si	si	10
Endodoncia	corto	8	4	50	si	si	si	10

Tabla 10. Experimentación por análisis Textual y por Sinonimia

Documento	Categoría	Extensión (No. hojas)	N-gramas	No. Palabras/ hoja	Análisis			Tiempos de Respuesta (minutos)
					Sinonimia	Textual	Textual con reducción	
sist_heredados_sinonimia.doc	corto	2	2	50	no	no	si	6
sist_heredados_textual.doc	corto	2	2	50	no	no	si	6
Agentes Inteligentes_sinonimia.docx	corto	4	2	50	no	no	si	10
Agentes Inteligentes_textual.docx	corto	4	2	50	no	no	si	8
DSS_sinonimia.docx	corto	4	2	50	no	no	si	10
DSS_textual.docx	corto	4	2	50	no	no	si	6
Motivacion_sinonimia.docx	corto	4	2	50	no	no	si	13
Motivacion_textual.docx	corto	4	2	50	no	no	si	7
Endodoncia	corto	8	2	50	no	no	si	8

Tabla 11. Experimentación por análisis por Sinonimia

De acuerdo a la experimentación analizada hemos llegado a determinar cuáles son los valores óptimos que nos permiten mejorar los resultados del sistema:

- Alteración del valor de n-gramas: A medida que el valor de “N” decrece el sistema encontrará un mayor número de coincidencias, sin embargo, esto puede llevar a un cálculo mayor de palabras coincidentes. Hemos creído conveniente tener un valor N que sea lo más neutral posible, en base a las pruebas y a la documentación existente sobre plagio que indica que **cuatro** palabras continuas que no posean referencia son consideradas como plagio[21] consideramos que el valor óptimo de N debe ser 4.

[21] www.flacso.org.ec/docs/plagioacademico.ppt

- Palabras por página: Considerando como página a un conjunto de palabras y tomando en cuenta que para cada página se realiza una búsqueda en Internet hemos probado con los siguientes números de palabras por página.
    - 150
    - 100
    - 50

  Siendo el valor óptimo de 50 palabras, ya que se generan una cantidad suficiente de páginas que a su vez representan más búsquedas en Internet que sean precisas y relevantes con el contenido de la página. De esta forma se aumentan las posibilidades de encontrar plagio. Todo esto implica que a menor cantidad de palabras por página mayor posibilidad de aciertos.

- Tamaño de descarga del archivo: Se ha considerado prudente descargar 5 megabyte tanto para archivos PDF, DOC y DOCX. Para el caso de ficheros de texto plano, los cuales contienen información que es mucho más liviana se limitó su tamaño máximo de descarga a 1 megabyte. Con estos valores hemos equilibrado la eficacia y la eficiencia del prototipo.
- Tiempo de descarga de un archivo: limitar el tamaño de descarga no es suficiente ya que pueden surgir problemas con el servidor en donde se encuentra alojado el archivo, es por ello que también nos vemos limitados a controlar el tiempo que le podría tomar al archivo en descargarse.

  El tiempo que equilibra velocidad y eficacia es 2 minutos según las pruebas realizadas con los siguientes tiempos:

    - 5 minutos
    - 2 minutos
    - 1 minuto
- Tiempo para la comparación de archivos: para el plagio por sinonimia se debe realizar una comparación de archivos, para ello inicialmente se lo realizaba hasta que el proceso termine, sin embargo debido al tiempo que este podía tomar y de acorde a la extensión que tenga cada uno de ellos se ha definido un tiempo máximo de 2 minutos para la comparación entre archivos, evitando de esta manera posibles cuellos de botella que limiten la velocidad de respuesta del sistema.

- Temática de los documentos: con el desarrollo de la experimentación hemos podido notar que la temática del documento también influencia en la capacidad de respuesta que el sistema tenga, ya que entre más científico o elaborado sea un documento, mayor será el tiempo que tome en detectar la existencia de plagio.

Como se puede observar en la Tabla 10 y Tabla 11 los tiempos se respuesta son altos a pesar de que existen buenos resultados, pensando en esto se tomó en consideración mejorar el sistema de detección plagio con la implementación de hilos como ya se mencionó en el capítulo 5.1. A continuación se muestra la Tabla 12 y la Tabla 13 con las respectivas configuraciones y los tiempos de respuesta que se obtuvieron.

Documento	Categoría	Extensión (No. hojas)	No. Palabras/hoja	Análisis			Tiempos de Respuesta (minutos)
				Sinonimia	Textual	Textual con reducción	
estudio.doc	corto	8	50	si	si	si	3
informe.docx	extenso	106	50	si	si	si	8
MYRNA_estudiosdecaso.pdf	medio	37	50	si	si	si	14
3historias.pdf	corto	8	50	si	si	si	3
aldea.pdf	corto	1	50	si	si	si	1
fantasmas.pdf	corto	8	50	si	si	si	3
sist_heredados_sinonimia.doc	corto	2	50	si	si	si	3
sist_heredados_textual.doc	corto	2	50	si	si	si	2
Agentes Inteligentes_sinonimia.docx	corto	4	50	si	si	si	4
Agentes Inteligentes_textual.docx	corto	4	50	si	si	si	3
DSS_sinonimia.docx	corto	4	50	si	si	si	1
DSS_textual.docx	corto	4	50	si	si	si	2
Motivacion_sinonimia.docx	corto	4	50	si	si	si	3
Motivacion_textual.docx	corto	4	50	si	si	si	3
Endodoncia	corto	8	50	si	si	si	4

Tabla 12. Análisis por sinonimia y textual con tiempos mejorados.

Documento	Categoría	Extensión (No. hojas)	No. Palabras/hoja	Análisis			Tiempos de Respuesta (minutos)
				Sinonimia	Textual	Textual con reducción	
estudio.doc	corto	8	50	no	si	si	1
informe.docx	extenso	106	50	no	si	si	2
MYRNA_estudiosdecaso.pdf	medio	37	50	no	si	si	1
3historias.pdf	corto	8	50	no	si	si	1
aldea.pdf	corto	1	50	no	si	si	1
fantasmas.pdf	corto	8	50	no	si	si	1
sist_heredados_sinonimia.doc	corto	2	50	no	si	si	1
sist_heredados_textual.doc	corto	2	50	no	si	si	1
Agentes Inteligentes_sinonimia.docx	corto	4	50	no	si	si	1
Agentes Inteligentes_textual.docx	corto	4	50	no	si	si	1
DSS_sinonimia.docx	corto	4	50	no	si	si	1
DSS_textual.docx	corto	4	50	no	si	si	1
Motivacion_sinonimia.docx	corto	4	50	no	si	si	1
Motivacion_textual.docx	corto	4	50	no	si	si	1
Endodoncia	corto	8	50	no	si	si	1

Tabla 13. Análisis Textual con tiempos mejorados.

Como se ha podido observar los tiempos de respuesta se han reducido notablemente tanto para la detección en sinonimia como en textual.

5.3.1 Perturbaciones encontradas

- Un factor importante para que la detección eficiente es la velocidad de conexión a Internet ya que de ser lenta al sistema le tomará más tiempo completar el

análisis. Asimismo es probable que los ficheros se descarguen mal desde Internet evitando que se puedan analizar de forma adecuada.

- Con respecto a la utilización de bibliografía y referencias, podemos decir que estas no son en su totalidad acertadas, ya que existen diversas formas de anotar referencias, es por ello que se lo ha implementado como un parámetro configurable dentro del sistema.
- Los tiempos de descarga y el tamaño de los archivos demasiado extensos se han contrarrestado a través de las restricciones que se han impuesto para los tamaños de archivos buscando en lo posible un funcionamiento eficiente y con resultados ajustados a la temática del documento.

A pesar de haber llegado a estas conclusiones, luego de haber realizado la experimentación debemos realizar los cálculos respetivos a la *Precision, Cobertura, F-Measure* y *Average Precision*, los cuales serán analizados en el capítulo 6.

6. ANÁLISIS DE RESULTADOS

El presente capítulo se enfoca en el análisis de los resultados que se obtuvieron una vez que se ejecutaron todas las pruebas respectivas, se toma una muestra de todos los elementos que se analizaron para poder obtener las medidas precisión, cobertura y F-Measure.

6.1. Análisis de precisión, cobertura y F – Measure.

6.1.1 Precisión

Se define como “la fracción de casos recuperados que son relevantes” [23], está definida por la siguiente fórmula:

$$precision = \frac{relevant\ documents\ \ \cap \{retrieved\ document\}}{|\{retrieved\ documents\}|}$$

Ecuación 9. Calculo de precisión [23].

- **Precisión para análisis por sinonimia**

Documento	Documentos Recuperados	Documentos Relevantes	Precisión
Endodoncia	33	33	1
Sistemas heredados	5	2	0.4
Agentes Inteligentes	6	3	0.5
		Total	0.63

Tabla 14. Precisión búsquedas con Sinonimia

- **Precisión para análisis textual**

Documento	Documentos Recuperados	Documentos Relevantes	Precisión
Endodoncia	40	40	1
Sistemas heredados	5	5	1
Agentes Inteligentes	6	6	1
		Total	1

Tabla 15. Precisión búsquedas Textual

6.1.2 Cobertura

Se entiende por cobertura a "la fracción de los documentos que son relevantes para la consulta que se ha recuperado correctamente" [34]. En otras palabras, es la relación entre los documentos relevantes y la totalidad de documentos que se esperaba obtener y que posiblemente no se obtuvieron.

$$cobertura = \frac{relevant\ documents\ \cap \{retrieved\ documents\}}{|\ relevant\ documents\ |}$$

Ecuación 10. Cálculo de cobertura [23].

- **Cobertura para sinonimia**

Documento	Documentos Recuperados	Documentos Relevantes	Cobertura
Endodoncia	77	33	0.43
Sistemas heredados	11	2	0.18
Agentes Inteligentes	6	3	0.5
		Total	0.37

Tabla 16. Cobertura búsquedas Sinonimia

- **Cobertura para textual**

Documento	Documentos Recuperados	Documentos Relevantes	Cobertura
Endodoncia.doc	77	40	0.52
Sistemas heredados	11	5	0.45
Agentes Inteligentes	10	6	0.6
		Total	0.52

Tabla 17. Cobertura búsquedas Sinonimia

6.1.3 F-measure

Se considera la media armónica entre la precisión y la cobertura y sirve para medir la exactitud de una prueba [34]. Por lo tanto, mediante esta medida podemos conocer el desempeño del sistema para cada uno de los 2 tipos de análisis.

$$F - Measure = 2\frac{precision * recall}{precision + recall}$$

Ecuación 11. Cálculo de F-measure [23].

6.1.3.1 F-Measure por Sinonimia

Documento	Precision * Recall	Precision + Recall	F-Measure
Endodoncia	0.43	1.43	0.6
Sistemas heredados	0.072	0.58	0.25
Agentes Inteligentes	0.25	1	0.5
		Total	0.45

Tabla 18. F-Measure búsqueda Sinonimia.

6.1.3.2 F-Measure por Textual

Documento	Precisión * Recall	Precisión + Recall	F-Measure
Endodoncia	0.52	1.52	0.68
Sistemas heredados	0.45	1.45	0.62
Agentes Inteligentes	0.6	1.6	0.75
		Total	0.70

Tabla 19. F-Measure búsqueda Textual

Una vez obtenidos estos valores podemos percatarnos de la gran diferencia que se obtiene entre al aplicar uno u otro análisis a un documento en concreto. En la siguiente gráfica se puede apreciar de mejor forma la mencionada diferencia:

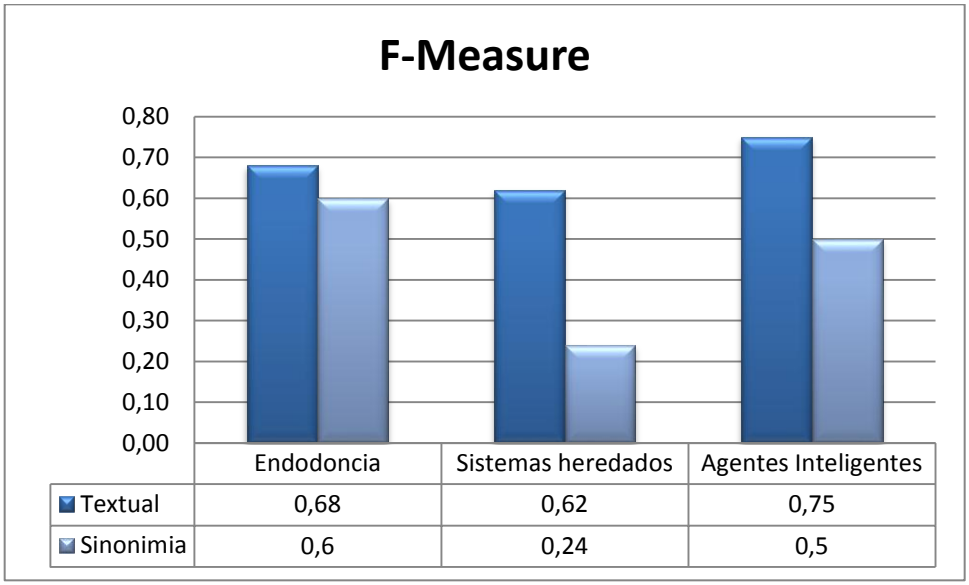


Ilustración 17. Diferencia de resultados entre análisis textual y sinonimia.

Los resultados de la medida F-Measure dejan mucho que desear para el conjunto de documentos sobre los cuales se hizo la prueba, esto es debido a que en muchos casos los buscadores no devuelven resultados relacionados con la temática del documento a analizar.

En el caso de análisis textual la medida F-Measure nos indica un nivel aceptable en los resultados que el sistema entrega. Es posible mejorar estos resultados si se fuerza la búsqueda textual, tal como indicamos en la sección 1.2. del Manual del Usuario

6.2. Cálculo de AVP (Average Precision)

Debido a nuestras necesitas el cálculo del AVP será realizado mediante el promedio de la precisión de los valores obtenidos con anterioridad.

- **Precisión en sinonimia**

Precision en sinonimia		Precisión en textual	
Documento	Precisión	Documento	Precisión
Endodoncia.doc	1	Endodoncia.doc	1
Sist_heredados.doc	0.4	Sist_heredados.doc	1
Agentes Inteligentes.docx	0.5	Agentes Inteligentes.docx	1

Tabla 20. Valores de Precisión en el análisis textual y sinonimia.

$$AVP = \frac{1 + 0.4 + 0.5 + 1 + 1 + 1}{6} = 0.82$$

Ecuación 12. Cálculo del AVP

El *Average Precision* en nuestro caso es de 0.685 lo que nos indica la precisión de nuestro sistema detector de plagio.

6.3. Comparación con el estado del arte

Unas de las principales medidas utilizadas para la detección de plagio son la precisión y la cobertura, sin embargo estas medidas se ven afectadas y tienden a ser inversamente proporcionales, es decir, si bien podemos obtener buenas respuestas obteniendo precisión seguramente la cobertura no obtendrá los mejores resultados, es por ello que se utiliza F-Measure como una media entre estos dos análisis.

Para la comparación respectiva de nuestra tesis se ha considerado pertinente realizarlo utilizando F-Measure como medida de comparación.

A continuación vamos a comparar nuestros resultados de F-Measure con el estado del arte actual sobre detección de plagio. Como veremos cada autor trata este problema desde diferentes perspectivas y por lo tanto obtienen diferentes resultados.

Estudio y desarrollo de nuevos algoritmos de detección de plagio. [5]

Al comparar nuestro sistema con los resultados obtenidos por Victoria Elizalde, quien en su tesis también utiliza coeficientes de similitud y N-gramas, notamos que para la obtención de similitud Elizalde utiliza el Coseno con una longitud de cinco N-gramas por palabras, además también utiliza el coeficiente de Dice con una longitud de siete palabras por cada grama, obteniendo como resultado en el caso del Coseno un *F-Measure* de 0.727 y en el caso de Dice un *F-Measure* de 0.96, cabe recalcar que este análisis es en base a la detección de plagio textual no riguroso y con una base de documentos fijos [5].

Nuestro análisis, a diferencia de Elizalde, radica en la comparación de un documento contra una serie de documentos obtenidos en la web es por ello que nuestro coeficiente para medir similitud es el de Overlap debido a que la comparación se hace por fracciones del documento original, en donde cada una de estas fracciones son enviadas a comparar contra paginas completas de internet, es por ello que Overlap se acopla a nuestras necesidades ya que para determinar el coeficiente de similitud lo realiza en base a la longitud del documento más pequeño, el número de N-gramas que hemos utilizado es de 4 en base a las pruebas realizadas, los resultados obtenidos con respecto al cálculo de *F-Measure* es de 0.45 para análisis por sinonimia y 0.70 para análisis por textual.

Como se ha podido observar los resultados obtenidos varían de acorde al coeficiente utilizado, otro punto en el que radica la diferencia es en la longitud de los N-gramas por palabras que se utiliza.

Sobre la importancia de la reducción del espacio de búsqueda en la detección automática de plagio [40]

Este trabajo de Barrón-Cedeño obtiene un resultado general de F-Measure igual a 0.68 el cual es mejorado a un valor igual a 0.75 al reducir el espacio de búsqueda, es decir reducir la cantidad de documentos origen que se utilizaran para encontrar plagio. Como es evidente nuestro sistema no reduce el espacio de búsqueda ya que este encierra todos los documentos de internet indexados por los 5 buscadores que estamos utilizando. Finalmente Barrón-Cedeño no especifica en su trabajo si el plagio que pretende detectar es textual o por sinonimia, simplemente realiza sus pruebas sobre el corpus METER [41] el cual posee versiones textuales, modificadas (posible sinonimia) y nuevas (no tienen relación con el documento original). Notamos que su valor de *F-Measure* es un poco más alto que el nuestro, sin embargo, nosotros no reducimos el espacio de búsqueda.

Detección de plagio en documentos. Sistema externo monolingüe de altas prestaciones basado en n-gramas contextuales [42]

En este trabajo los autores atacan la problemática con técnicas diferentes a las nuestras mencionando el uso de “n-gramas contextuales” y de “monotonía referencial”, utilizan el corpus PAN’09 [43] y presentan resultados para bigramas y trigramas para su sistema externo. Resulta curioso notar que a pesar de utilizar diferentes técnicas a las nuestras obtienen los mismos resultados para *F-Measure* que nosotros, es decir un valor igual a 0.70. Hacemos notar que los autores obtienen este resultado al utilizar trigramas ya que cuando utilizan bigramas su valor de *F-Measure* cae hasta un valor de 0.65.

Plagiarism Detection using ROUGE and WordNet [44]

En este trabajo los autores pretenden realizar análisis por sinonimia así como también análisis textual. En el trabajo se indica que utilizaran el coeficiente de Jaccard para

detectar el plagio por sinonimia así como también la base de datos de WordNet para encontrar sinónimos. Como se aprecia este trabajo de Chien-Ying, Jen-Yuan y Hao-Ren tiene un gran parecido con nuestro sistema; sin embargo pretenden encontrar plagio en idioma Ingles, esto se evidencia al utilizar la base de datos de WordNet.

En este trabajo pretenden encontrar plagio sin remover *stop words* y sin realizar un pre-procesado como eliminación de puntos, tildes, y demás.

Los resultados que obtienen al eliminar *stop words* superan ligeramente nuestro valor de *F-Measure* en plagio textual, sin embargo no utilizan 4-gramas sino uni-gramas. Para el caso de sinonimia obtienen un valor cercano a 0.70, lo cual también indica un buen resultado a favor de los uni-gramas.

Cuando se eliminan las *stop-words* su valor de F-Measure para análisis textual no varía, sin embargo logran aumentar su valor de F-Measure para análisis por sinonimia, el cual es muy cercano a 0.70 con lo cual obtienen un mejor resultado que el nuestro de 0.45.

A pesar de que este trabajo tiene muchas similitudes con nuestro sistema, hemos de mencionar que también exploraban otras técnicas como skip-bigramas, uso de WordNet y el uso de técnicas como *longest common subsequence* (LCS) [22] sin mencionar que los idiomas sobre los cuales se realizan los análisis son diferentes.

Diseño e Implementación de una técnica para la detección de plagio en documentos digitales [45]

Finalmente realizaremos la comparación de nuestro sistema contra un trabajo que pretende realizar análisis textual.

El autor analiza segmentos de un documento contra segmentos de otro documento y procede a ordenar alfabéticamente y/o por frecuencia de aparición a los N-gramas de cada uno de los documentos y realiza la comparación con los últimos m n-gramas que hayan resultado del proceso de ordenamiento.

Como podemos ver su técnica para analizar plagio textual es muy diferente a la nuestra así como los resultados de *F-Measure* que obtiene: 0.8 y 0.9 (en análisis exhaustivo) frente al valor de 0.7 que obtuvimos en nuestro sistema.

Si bien obtiene unos resultados más que aceptables hay que notar que el autor posee una base de documentos originales, por lo que analiza cada segmento del documento

[22] Técnica de comparación de archivos.

sospechoso contra cada segmento del documento existente en la base. En cambio nosotros utilizamos un enfoque de hoja versus página web.

Autores	Alcance	Precisión	Cobertura	F-Measure
Elizalde	No especifica sinonimia	0.965	0.790	0.869
Cedeño-Rosso	No especifica sinonimia	0.77	0.74	0.75
Torrejón-Ramos	No especifica sinonimia	0.7989	0.6349	0.7075
Chien-Ying, Jen-Yuan y Hao-Ren	Textual - Sinonimia	-	-	0.70
León Oberreuter Gallardo	No especifica sinonimia	0.895	0.914	0.904
Flores-León	Textual	1	0.52	0.70
	Sinonimia	0.63	0.37	0.45

Tabla 21. Comparación de Precisión, Cobertura y F-Measure de diversos sistemas

6.4. Propuesta de mejoras y trabajo futuro

Se ha trabajado duro a lo largo de estos meses con la finalidad de cumplir satisfactoriamente los objetivos planteados para la elaboración de nuestra tesis, pese a ello, se pueden realizar sugerencias que podrían mejorar la calidad del trabajo realizado hasta la actualidad. Procedemos a sugerir una serie de mejoras a futuro para que el sistema pueda realizar un mejor trabajo en menos tiempo.

- **Mejorar la conversión de documentos a texto plano.**

Para la transformación de documentos nos hemos visto en la necesidad de utilizar librerías externas para Java que permitan transformar estos documentos, las librerías que encontramos son “PDFBox”, “Apache POI” y “docx4j” que facilitaron la conversión de los archivos PDF, DOC y DOCX respectivamente, sin embargo estos no pudieron recuperar de una manera satisfactoria el texto, ya que incorporaban espacios y saltos de carro donde no existían, tornando dificultoso el proceso de obtención de párrafos al momento de leer un archivo. Se sugiere que en futuras versiones de este prototipo se considere actualizar a librerías mucho más compatibles con los formatos a convertir.

- **Reconocimiento de imágenes con sus debidas referencias.**

Se recomienda la posibilidad de analizar las imágenes que el documento pueda contener con la finalidad de poder identificar uno de los plagios más comunes a un nivel más avanzado.

- **Tratamiento de idiomas**

Se recomienda que el sistema de detección de plagio vaya más allá del idioma español, de tal forma que tenga la capacidad de abarcar diversos idiomas entre estos el inglés que es uno de los más utilizados y que facilitaría la detección cuando por lo general existe plagio de un documento en inglés traducido al idioma español, escenario en el cual nuestro prototipo no podría detectar plagio.

- **Mejorar la eficiencia**

El prototipo es bastante eficaz, pero su capacidad de respuesta se ve limitada por diversos factores, como son la velocidad de conexión de Internet, la temática del documento y la extensión del documento, estos factores influyen directamente sobre la velocidad de respuesta del sistema prototipo, en una próxima versión se recomienda

trabajar sobre estos puntos para lograr obtener resultados más eficientes en este aspecto. Somos conscientes de que los principales cuellos de botella que afectan directamente al desempeño de nuestro sistema prototipo son:

- Conectividad con Internet (factor externo).
- Velocidad de conversión de documentos a texto plano (factor interno).

CONCLUSIONES Y RECOMENDACIONES

Al finalizar el presente trabajo de investigación, hemos podido concluir que el prototipo presentado colaborará de manera eficaz a la detección de plagio. Para mayor comodidad del usuario se lo ha realizado en un entorno Web, de tal manera que el proceso que se lleva a cabo sea transparente al usuario, permitiendo obtener únicamente el informe necesario para la comprobación del mismo.

A lo largo del desarrollo del prototipo se han presentando una serie de inconvenientes, entre estos, la herramienta FreeLing que tiene escasa información con respecto al API de Java, es por ello que nos vimos en la necesidad de consultar con el desarrollador de la herramienta Lluis Padró a través del foro de la página oficial del proyecto, quien nos supo responder y guiar en este proceso.

Una de las grandes limitaciones de nuestro sistema es la dependencia que tenemos de los buscadores, ya que hemos escogido 5 buscadores que nos ayudarán a la obtención de datos, estos son: Bing, Yahoo, Yandex, Ask y Google. Es importante mencionar que fue necesario la utilización de todos estos buscadores ya que al utilizar solo un buscador este presenta complicaciones al detectar el programa y el constante envío de solicitudes, limitando la búsqueda al producir una excepción, bloqueando futuras búsquedas.

En un análisis por sinonimia resulta interesante concluir que se obtienen mejores resultados al no cambiar palabras (al azar) por sinónimos para realizar la búsqueda, sino más bien al obtener una lista de palabras importantes mediante la ley de Zipf y realizar la búsqueda con estas palabras originales sin sinónimos. Por lo tanto podemos concluir que el mejor enfoque para detectar sinonimia no necesariamente debe recurrir a la utilización de sinónimos.

También podemos concluir que por más que se elijan palabras importantes dentro de una hoja gracias a la ley de Zipf los resultados que los buscadores lanzan no siempre son acordes a la temática del documento original ocasionando varios falsos positivos. También debemos resaltar el hecho de que para un análisis de sinonimia generalmente

se obtienen mejores resultados al mantener las palabras importantes que Zipf devuelve sin aplicar sinónimos debido a que es poco probable que se cambien por sinónimos a palabras importantes del texto y debido a que los sinónimos pudieron haber sido cambiados en palabras menos importantes que las devueltas por Zipf, de tal forma que al enviar palabras importantes sin ningún cambio podemos dar con el documento original e indirectamente detectar plagio por sinonimia en palabras menos importantes.
Con respecto al estado del arte encontramos razones por las cuales muchos sistemas actuales presentaban resultados nulos al solicitar inclusive análisis textuales y es que se enfocaban en la rapidez para entregar resultados, sin embargo, al desarrollar nuestro sistema nos hemos percatado que realizar análisis de plagio, en especial por sinonimia, resulta extremadamente pesado por lo tanto dichos sistemas no podrían encontrar un número significativo de resultados sin perder velocidad.

Resulta interesante también recalcar todas las tecnologías usadas para desarrollar este sistema, desde conexiones a motores de bases de datos hasta el manejo de hojas de estilos CSS pasando por gestionar Hilos y programar sitios con JSP, configurar servicios en Linux, instalar desde código fuente, etc. así como también se utilizo mucha información aprendida durante la carrera como la creación de planes de experimentación y la importancia de un diseño de software previo. Se mejoraron nuestras habilidades para investigar especialmente a no tener miedo a la documentación oficial tanto de Java como de cualquiera de las librerías que hemos utilizado.

Consideramos nuestros resultados para plagio textual como aceptables en comparación con el estado del arte actual. Sin embargo, no podemos decir lo mismo para nuestros resultados sobre detección de plagio por sinonimia. Es importante mejorar dicho apartado en futuras versiones del sistema. A pesar de esto, el valor agregado que generamos al añadir la posibilidad de programar una revisión múltiple (útil para revisar trabajos de todos los estudiantes de un aula) nos permite concluir que el proyecto es útil y puede ser usado como herramienta frecuente por parte de cualquier docente.

Se recomienda que para futuras versiones se pueda extraer texto de páginas como “scribd.com” cuyo contenido esta embebido mediante tecnologías como flash dentro del sitio web, esto indudablemente mejorará los resultados que se obtengan.

Después de haber desarrollado esta tesis nos hemos dado cuenta que resulta muy difícil avanzar sin una base teórica. Mucho se fue aprendiendo a medida que se iba avanzando en el desarrollo del prototipo. Si hubiésemos tenido desde un principio una idea mas clara de lo que íbamos a realizar se hubiera diseñado mejor el sistema, por lo tanto, para trabajos futuros recomendamos documentarse primero sobre detección de plagio. Quizá el trabajo mas didáctico para este fin sea: “Diseño e implementación de una técnica para la detección de plagio en documentos digitales” [45].

Muchas veces el documento que extraía desde Internet, a pesar de coincidir con las palabras del documento a analizar, no estaba acorde a la temática que pretendíamos analizar. Por ejemplo al buscar plagio en un documento sobre odontología el sistema llega a extraer desde Internet un documento relacionado a medicina debido a que coinciden en palabras como “infección” o “hueso”.
Por lo tanto recomendamos antes de extraer un documento desde Internet extraer su meta-información de tal forma que podamos conocer la temática que este trata y verificar si dicha información esta dentro del contexto de lo que deseamos analizar. En caso de no coincidir se debería evaluar el siguiente documento devuelto como resultado de la búsqueda.

Como última recomendación para futuras versiones del sistema, se sugiere mejorar la técnica para obtener los sinónimos que serán remplazados, el objetivo de esta propuesta es hacer que aquellos sinónimos que tengan mayor relación con el contexto original sean los elegidos.

REFERENCIAS:

[1] TELLO Estefanía y ZEPEDA Beatríz, *El plagio académico,* fecha de recuperación: 16-oct-2011, www.flacso.org.ec/docs/plagioacademico.ppt

[2] CAMPOS GARCÍA Martha Patricia, Apuntes sobre redacción y plagio académico, fecha de recuperación: 19-oct-2011, http://201.234.71.135/portal/uzine/Volumen19/desc/11_redaccion_plagio.pdf

[3] PAZMIÑO YCAZA Antonio, Universidad Católica de Santiago de Guayaquil, Revista Jurídica de Propiedad Intelectual, Tomo 4, http://www.revistajuridicaonline.com/images/stories/revistas-juridicas/propiedad-intelectual-tomo-4/propiedad-intelectual-tomo4.pdf

[5] ELIZALDE Victoria, Estudio y desarrollo de nuevos algoritmos de detección de plagio, fecha de recuperación: 28-nov-2011, http://www.dc.uba.ar/inv/tesis/licenciatura/2011/elizalde

[6] GARCÍA G. R and RODRÍGUEZ E.G, Fraude y plagio académico en los ambientes virtuales de aprendizaje, fecha de recuperación: 28-nov-2011, http://www.distancia.unam.mx/contenido/historico/foroeducativodos/Guillermo%20Roquet%20trabajo%20escrito.pdf

[7] Wikipedia, N-grama, fecha de recuperación: 26-nov-2011, http://es.wikipedia.org/wiki/N-grama

[8] Diccionario de la Lengua Española, fecha de recuperación: 18-nov-2011, http://buscon.rae.es/draeI/

[9] SOSA Eduardo, Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones (Parte I), fecha de recuperación: 01-nov-2011, http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actual_bases_tericas_y_aplicaciones_parte_i.html

[10] ARAUJO Lourdes, Procesamiento de Lenguaje Natural, fecha: 01-nov-2011, http://tabasco.torreingenieria.unam.mx/gch/PLN/cap1.pdf

[11] VARIOS AUTORES, Verbos Copulativos, fecha de recuperación: 04-nov-2011, http://www.123teachme.com/learn_spanish/sp_verbos_copulativos

[12] VARIOS AUTORES, How can you search Google Programmatically Java API, fecha de recuperación: 04-nov-2011 http://stackoverflow.com/questions/3727662/how-can-you-search-google-programmatically-java-api

[13] VILLARDÓN José Luis Vicente, Análisis de coordenadas principales, fecha de recuperación; 04-nov-2011, http://biplot.usal.es/DOCTORADO/3CICLO/BIENIO-04-06/ACP/COORPRIN(apuntes).pdf

[14] RODRÍGUEZ SALAZAR María Elena/ ÁLVAREZ HERNÁNDEZ Sergio/ BRAVO NÚÑEZ Ernesto, Coeficientes de asociación, primera edición, Plaza y Valdez S.A de C.V, México.

[15] Wikipedia, Tesauros: Concepto elaboración y mantenimiento, fecha de recuperación: 12-dic-2011, http://web.usal.es/~alar/Bibweb/Temario/Tesauro.PDF

[16] DIEZ ITZA Eliseo, El lenguaje: estructuras, modelos, proceso y esquemas: un enfoque pragmático, Servicio de publicaciones Universidad de Oviedo, pp. 75

[17] FERNANDEZ GALLARDO Pablo, El secreto de google y el álgebra lineal, fecha de recuperación; 19-noviembre-2011, http://neodev.es/download/google_sema.pdf

[18] AMAYA ROBAYO Fredy, Acerca de los modelos del lenguaje basados en gramáticas estocásticas, fecha de recuperación: 10-diciembre-2011, http://www.unicauca.edu.co/matematicas/publicaciones/articulo_fredy_amaya1.pdf

[19] Wikipedia, Modelos de Markov, fecha de recuperación: 07-enero-2012, www.edicionsupc.es/ftppublic/pdfmostra/OE03502M.pdf

[20] Wikipedia, Cadena de Markov, fecha de recuperación: 07-enero-2012, http://es.wikipedia.org/wiki/Cadena_de_Markov

[21] SEGURA BEDMAR Isabel, MARTÍNEZ FERNÁNDEZ José L, MARTÍNEZ Paloma Una Propuesta para el Etiquetado Automático de Roles Semánticos, fecha de recuperación: 07-enero-2012, dirección web: http://www.sepln.org/revistaSEPLN/revista/37/38.pdf

[22] CSCAZORLA, La ley de Zipf: El porqué de las palabras cortas y largas, fecha de recuperación: 07-enero-2012, dirección web: http://www.xatakaciencia.com/sabias-que/el-por-que-de-las-palabras-cortas-y-largas

[23] GARCIA MATEOS, Gines, Medida del tiempo y la memoria de un programa, fecha de recuperación: 07-enero-2012, Dirección web: http://dis.um.es/~ginesgm/medidas.html

[24] PADRÓ Lluís, Analizadores Multilingües en FreeLing, fecha de recuperación: 08-enero-2012, http://linguamatica.com/index.php/linguamatica/article/view/115/133

[25] Universidad Politécnica de Cataluña, EAGLES, fecha de recuperación: 08-enero-2012, http://www.lsi.upc.edu/~nlp/tools/parole-sp.html

[26] Antonio Moreno Ortiz, Wordnet, fecha de recuperación: 12-enero-2012, http://elies.rediris.es/elies9/2-4-2.htm

[27] LEINER DE LA CABADA Marie, La amenaza del plagio en el ámbito educativo, fecha de recuperación: 18-ene-2012, http://dialnet.unirioja.es/servlet/dcfichero_articulo?codigo=2734233&orden=0

[28] NUÑEZ Miguel Ángel, El plagio como amenaza, fecha de recuperación: 20-ene-2011, http://miguelangelnunez.suite101.net/el-plagio-como-amenaza-a8443

[29] Global WordNet Organization, Wordnets in the world, fecha de recuperación: 25-ene-2012, http://www.globalwordnet.org/gwa/wordnet_table.html

[30] SATO Hiroaki, FrameNet, fecha de recuperación: 25-ene-2012, http://sato.fm.senshu-u.ac.jp/frameSQL/sfn20/notes/index2.html

[31] Europapress, “Por plagio la Universidad de Bayreuth retira el doctorado de Derecho al ministro de Defensa” en Europapress, Miércoles, 20 de noviembre 2011, http://www.europapress.es/internacional/noticia-universidad-bayreuth-retira-doctorado-derecho-ministro-defensa-20110223232341.html

[32] CARAZO GIL, F. Javier, Instalar Apache Tomcat 7, fecha de recuperación: 24-febrero-2012, Dirección web: http://www.linuxhispano.net/2011/05/20/instalar-apache-tomcat-7/

[33] ESPEN Dan, When should I set LD_LIBRARY_PATH?, fecha de recuperación: 24-febrero-2012, Dirección web: http://linuxmafia.com/faq/Admin/ld-lib-path.html

[34] Wikipedia, Precision and Recall, fecha de recuperación: 22-feb-2012, http://en.wikipedia.org/wiki/Precision_and_recall

[35] Blog at WordPress, It's a bird it's a plane it depends on your classifier's threshold, fecha de recuperación: 22-feb-2012,http://sanchom.wordpress.com/tag/average-precision/

[36] BARRÓN CEDEÑO Alberto; VILA Marta; ROSO Paolo, Detección automática de plagio: de la copia exacta a la paráfrasis, fecha de recuperación: 01-mar-2012, http://clic.ub.edu/files/ling-forense-plagio.pdf

[37] Wikipedia, DOC, fecha de recuperación: 03-mar-2012 http://es.wikipedia.org/wiki/DOC

[38] SATO Hiroaki, About FrameNet, fecha de recuperación: 03-mar-2012 https://framenet.icsi.berkeley.edu/fndrupal/about

[39] Wikipedia, FrameNet, fecha de recuperación: 03-mar-2012, http://es.wikipedia.org/wiki/FrameNet

[40] BARRÓN CEDEÑO Alberto; ROSO Paolo, On the Relevance of Search Space Reduction in Automatic Plagiarism Detection, fecha de recuperación: 05-abril-2012, http://www.sepln.org/revistaSEPLN/revista/43/articulos/art16.pdf

[41] CLOUGH Paul; GAIZAUSKAS Rob, The Meter Corpus, fecha de recuperación: 05-abril-2012, http://nlp.shef.ac.uk/meter/

[42] RODRÍGUEZ TORREJÓN Diego Antonio; RAMOS MARTÍN José Manuel, Detección de plagio en documentos. Sistema externo monolingüe de altas prestaciones basado en n-gramas contextuales, fecha de recuperación: 05-abril-2012, http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/797/651

[43] STEIN Benno, Computer Science and Media, fecha de recuperación: 05-abril-2012, http://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-pc-09.html

[44] CHIEN-YING Chen; JEN-YUAN Yeh;HAO REN Ke, Plagiarism Detection using ROUGE and WordNet, fecha de recuperación: 05-abril-2012, http://arxiv.org/pdf/1003.4065v1.pdf

[45] LEÓN Gabriel Ignacio, Diseño e Implementación de una técnica para la detección de plagio en documentos digitales, fecha de recuperación: 05-abril-2012, http://www.cybertesis.uchile.cl/tesis/uchile/2010/cf-oberreuter_gg/pdfAmont/cf-oberreuter_gg.pdf

ANEXOS

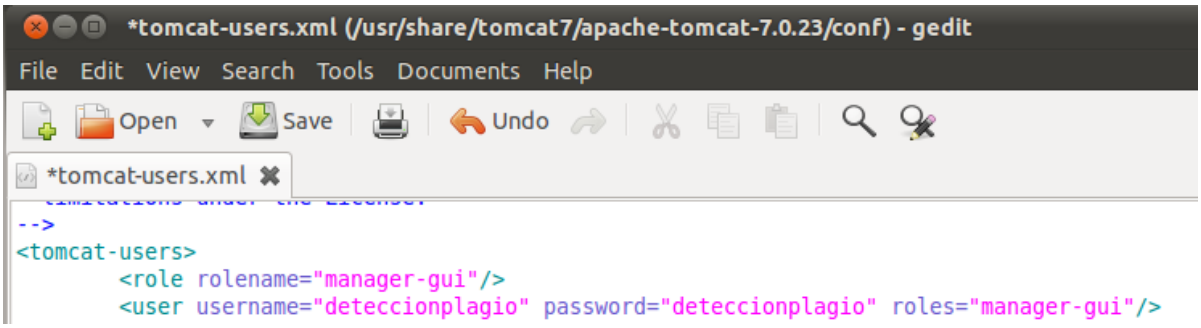
ANEXO 1: INSTALACION DEL SERVIDOR APACHE TOMCAT

Para servir el sitio web a los posibles clientes en el lado del servidor se utilizó la última versión estable, es decir Apache Tomcat 7.

Primero se descargó el paquete comprimido desde el sitio oficial del proyecto[23], luego se creó un directorio llamado *tomcat7* encargado de albergar los contenidos del paquete descargado de tal forma que la ruta de instalación fue: */usr/share/tomcat7/apache-tomcat-7.0.23/.*

Luego, según [32] se deben definir las variables de entorno **JAVA_HOME** y **JRE_HOME**, sin embargo, no fue necesario en nuestro caso puesto que estas se encontraban definidas correctamente.

Inicialmente Tomcat, por razones de seguridad, no trae ningún usuario activado por defecto, por lo tanto tuvimos que crear un usuario y asignarle un rol de administrador. Para hacer esto se tuvo que editar el archivo */usr/share/tomcat7/apache-tomcat-7.0.23/conf/tomcat-users.xml* dentro del cual se definen las credenciales del usuario. Las credenciales que nosotros hemos definido han sido las mismas con las que se accede al servidor; obviamente no podemos mencionarlas, pero incluimos un ejemplo con otras credenciales para clarificar el contenido de este archivo.




```
<tomcat-users>
        <role rolename="manager-gui"/>
        <user username="deteccionplagio" password="deteccionplagio" roles="manager-gui"/>
```

Ilustración 18. Configuración del usuario administrador de tomcat7.

[23] http://newverhost.com/pub/tomcat/tomcat-7/v7.0.26/bin/apache-tomcat-7.0.26.tar.gz

Una vez que hemos configurado el usuario administrador solamente resta iniciar el servicio ejecutando el archivo **startup.sh** ubicado en la carpeta */usr/share/tomcat7/apache-tomcat-7.0.23/bin/.*

Debido a que el inicio es manual decidimos incluir el inicio del servicio junto con el arranque del sistema. Para conseguirlo editamos el fichero */etc/rc.local* y agregamos la instrucción que arranca Tomcat.

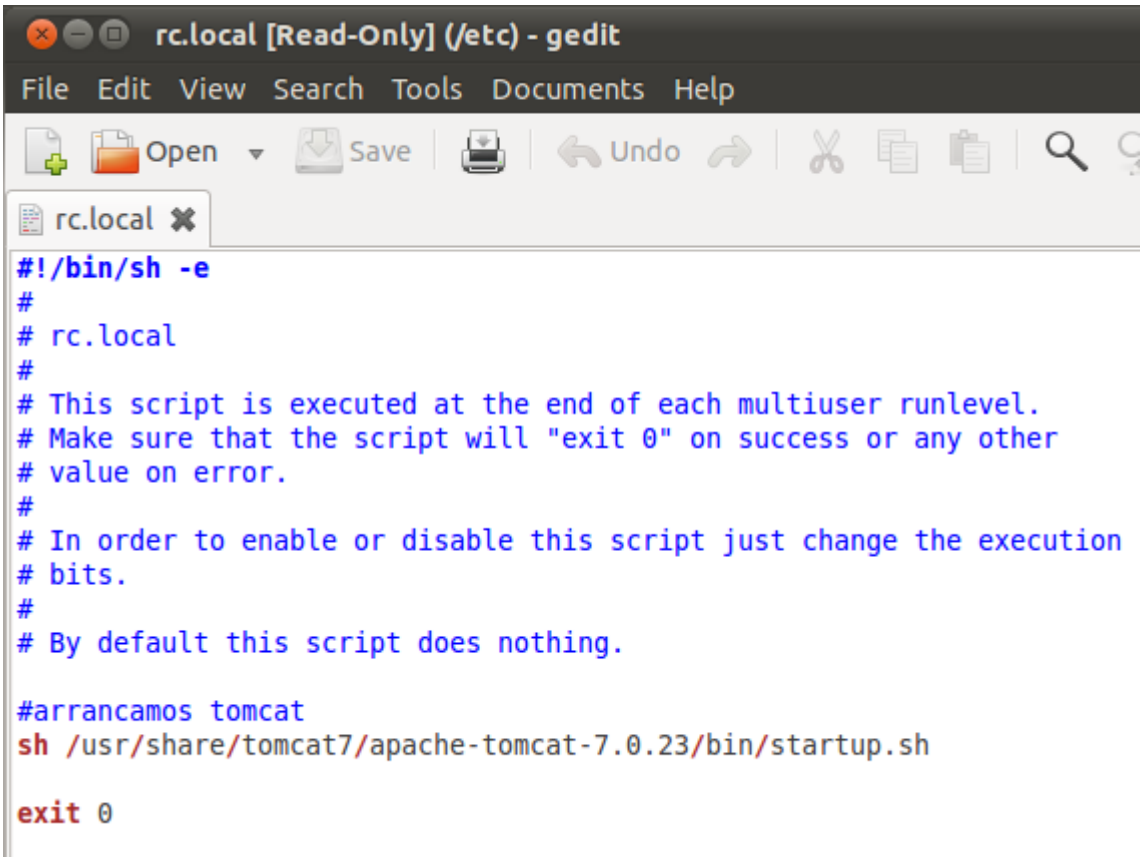


Ilustración 19. Contenido del fichero /etc/rc.local. Indicamos nuestro arranque de Tomcat.

Ahora, debido a que nuestro sistema necesita establecer la variable de entorno **LD_LIBRARY_PATH** para funcionar y que esta variable de entorno, por motivos de seguridad [33], solo puede ser usada durante la sesión actual del usuario obligando así

que para cada uso esta deba ser definida nuevamente. Debido a esto tuvimos que modificar nuestro fichero de arranque *startup.sh* y agregar en este fichero la instrucción que establece el valor para LD_LIBRARY_PATH de tal modo que para cada arranque de Tomcat éste pueda acceder a esta variable de entorno y consecuentemente cada aplicación web -como es el caso de nuestro sistema- pueda acceder a dicha variable. A continuación se muestra la modificación realizada al archivo *startup.sh*

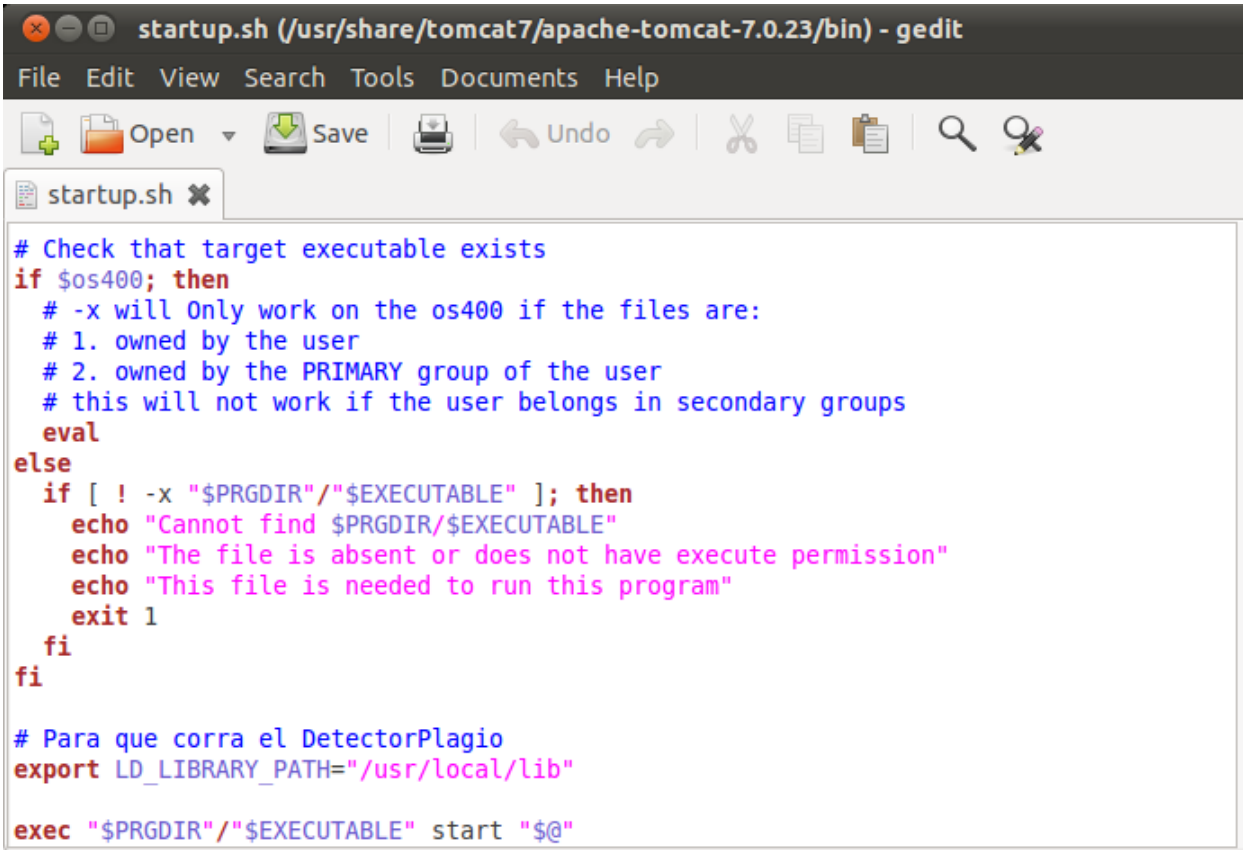


Ilustración 20.Extracto final del archivo startup.sh

ANEXO 2: MANUAL DEL USUARIO

ÍNDICE

Introducción

Este manual de usuario está elaborado con la finalidad de explicar cada uno de los componentes, funcionalidades y configuraciones del sistema. A medida que avance en la lectura del presente manual notará que la utilización del sistema no es difícil; sin embargo deberá poseer un conocimiento profundo sobre cómo funciona el sistema para poder modificar las configuraciones del mismo. No deberá preocuparse por las configuraciones debido a que los ajustes iníciales de las mismas le permitirán usar el sistema sin problema y sin la necesidad de que usted deba modificarlas; no obstante, se explicará en su momento cada una de las configuraciones existentes para que usted las pueda ajustar a sus necesidades si así lo desea.

El sistema de detección de plagio está encapsulado en un archivo JAR (Java Archive) el cual puede ser ejecutado desde línea de comandos. Sin embargo, se ha creado una página web cuya finalidad es permitir un fácil manejo del sistema para el usuario final.

Este manual está orientado a los usuarios, sin embargo, si usted es un desarrollador y desea realizar modificaciones sobre el código y/o instalar el sistema en sus servidores

deberá ponerse en contacto con los desarrolladores del sistema: Andrea Flores[24] y Bernardo León[25].

Note que antes de utilizar el sistema debe estar consciente de que el mismo solo analiza plagio en texto, por tanto, si existen diagramas u otro tipo de imágenes o contenido no textual el sistema los ignorará en su análisis.

1.1 Elementos del sistema

El sistema posee 2 secciones o funcionalidades las cuales son:

- Detección Web: Es decir, **busca en Internet** mediante el uso de los siguientes buscadores (Google, Bing, Yahoo, Ask y Yandex) fuentes originales como Sitios Web y documentos desde los cuales se pudo haber plagiado. Se considera plagio si encuentra contenido coincidente que no contiene las debidas referencias. Este análisis puede tomar mucho tiempo dependiendo de la extensión y temática del documento a analizar.
- Detección Local: Por otro lado en este tipo de análisis el sistema **compara dos documentos locales en busca de copia entre ellos** por lo que no buscará en Internet fuentes originales, lo que a su vez tiene como efecto un análisis rápido en comparación con la funcionalidad de Detección Web.

Los siguientes elementos pertenecen a la funcionalidad de Detección Web.

Ilustración 21.Principales elementos de la funcionalidad Detección Web del sistema.

24 Andrea Flores. angieflores88@gmail.com

25 Bernardo León. bbernardoleon@gmail.com

1. Botón para ir hacia la página que realiza la detección de plagio buscando fuentes en Internet. Dicha página es la que se acaba de mostrar en la imagen.
2. Botón para ir hacia la página que realiza la detección de plagio localmente en el servidor sin necesidad de recurrir a fuentes externas en Internet como por ejemplo Google.
3. Botón para seleccionar el archivo a analizar. Dicho archivo puede ser un ensayo, un trabajo, una tesis o algún otro tipo de documento del cual se sospecha posee contenido plagiado. Este archivo debe poseer alguna de las siguientes extensiones: DOC, DOCX, PDF o TXT, luego se debe procurar que no posea errores; es decir, que no hayan sido creados, modificados y guardados con editores ineficaces. Para conseguir esto procure que los archivos hayan sido creados con herramientas como Microsoft Office o Libre Office y así mismo evite que los archivos hayan sido creados o modificados con Herramientas como Abiword cuyo desempeño es limitado.
4. Botón para subir el archivo al servidor y dar inicio a la ejecución del sistema.
5. Enlace a las configuraciones avanzadas del sistema. Posteriormente se detallarán las configuraciones que se pueden modificar y que significan.
6. Opción para indicar al sistema que se desea analizar plagio textual en el archivo que se va a procesar. Toma menor tiempo que analizar plagio por sinonimia.
7. Opción para indicar al sistema que se desea analizar plagio por sinonimia en el archivo que se va a procesar. Toma más tiempo que analizar plagio textual.

Notar que las opciones 6 y 7, es decir, los tipos de análisis que se van a realizar no son excluyentes, por lo tanto se puede indicar que para un archivo se realice análisis textual y también análisis por sinonimia.

Los siguientes elementos pertenecen a la funcionalidad Detección Local:

Ilustración 22. Elementos de la funcionalidad Detección Local del sistema**.**

1. Es el primer archivo a ser comparado.
2. Es el segundo archivo a ser comparado.
3. El botón que sube los archivos al servidor y ejecuta el sistema para realizar la comparación entre estos 2 archivos.

Como se ha visto el sistema posee pocos elementos los cuales además son simples de entender y utilizar. Aun así, debemos explicar los dos tipos de análisis con los que se cuenta para realizar la Detección Web.

5.4.1.1 Análisis Textual

Se considera que un documento posee plagio textual cuando su contenido posee al menos 4 palabras continuas coincidentes con el contenido de otro documento. Considerando este precedente el sistema selecciona bloques (que no posean ningún tipo de referencia) de al menos 10 palabras continuas -en el peor de los casos selecciona 4 palabras continuas- y utiliza los buscadores ya mencionados para encontrar resultados en Internet que coincidan textualmente con el bloque. De encontrar un resultado coincidente el sistema lo almacena para generar el informe de análisis textual final; luego, procede con el siguiente bloque hasta finalizar con el archivo que ha subido al servidor.

5.4.1.2 Análisis por Sinonimia

Existen casos en los que se plagia texto -al no indicar referencia alguna hacia una fuente- y con esperanzas de encubrir el plagio se cambian algunas palabras de dicho

texto por sus sinónimos. De ésta forma una búsqueda textual no encontrará resultados evitando así que se pueda detectar el plagio.

Cuando se selecciona este tipo de análisis el sistema realiza una serie de procesos:

- Divide al documento que se desea analizar en páginas de las cuales se obtienen las palabras más importantes de cada una de ellas. Estas páginas no deben poseer necesariamente la misma cantidad de palabras que una página real, solamente significan que si una página es pequeña (posee pocas palabras) la búsqueda será más precisa.
- Una vez que se obtiene el conjunto de palabras más importantes de una página se eligen algunas de éstas palabras y se las reemplaza por sus respectivos sinónimos.
- Se busca éste conjunto de palabras mediante los buscadores con la esperanza de encontrar el sitio original desde el cual se sospecha que el contenido de la página ha sido plagiado.
- A continuación se procede a extraer el contenido desde Internet y compararlo con el contenido de la página que está siendo analizada.
- Se obtiene un índice de plagio entre la página y el contenido extraído desde Internet y se almacena el contenido de la página, el sitio desde el cual se sospecha se ha plagiado y el índice de plagio obtenido.
- Se avanza hasta la siguiente página y se repite el proceso.

Si una página posee referencias entonces dicha página no es analizada.

Este análisis hace un uso intensivo de Internet por lo que puede tomar mucho tiempo en realizarse si el servidor tiene problemas en la conexión, si el texto es extenso y si las páginas son muy pequeñas.

Ahora que conoce lo que cada uno de estos análisis significa y hace podrá decidir de acuerdo a sus necesidades cual utilizar cada vez que recurra a este servicio. No debe olvidar que puede seleccionar los 2 tipos de análisis si lo desea.

1.2 Configuraciones

Si desea ajustar el funcionamiento de la sección “Detección Web” del sistema deberá hacer clic en el vínculo de *configuraciones* presente en dicha sección. Recuerde que no es necesario modificarlas. A continuación se explican cada una de éstas configuraciones:

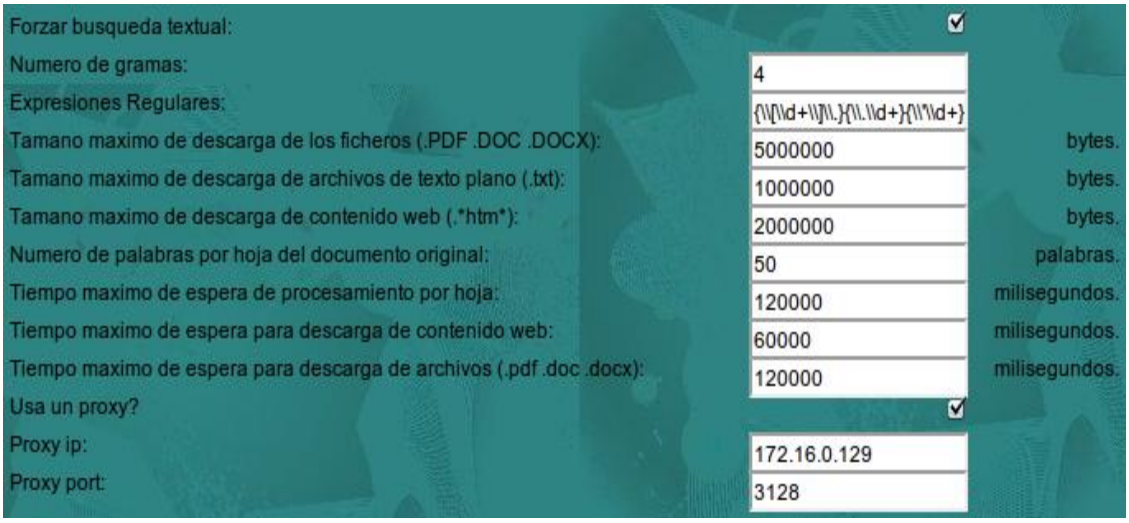


Ilustración 23. Configuraciones avanzadas disponibles en la sección Detección Web del sistema.

- **Forzar búsqueda textual:** Cuando se realiza análisis de plagio textual muchas veces los buscadores no encuentran coincidencias debido a que el texto a buscar es demasiado largo. Es por esto que si activamos esta opción el sistema irá eliminando cada vez una palabra al final de la búsqueda y, mientras la búsqueda tenga al menos 10 palabras, seguirá buscando con la esperanza de encontrar alguna coincidencia. Evidentemente realizar ésta tarea puede hacer que el análisis tome más tiempo, sin embargo, permite identificar plagio con mayor grado de aciertos. Se recomienda activar esta opción para textos que no posean muchas referencias, por otro lado se recomienda desactivar esta opción cuando el texto a analizar posee una gran cantidad de referencias. Esta opción viene activada por defecto. (Se utiliza en análisis textual).

- **Número de gramas:** Son utilizados para saber el número de palabras que conformarán cada grama, esto servirá como punto de comparación entre los

documentos para el proceso de sinonimia, el sistema utiliza este valor para obtener el coeficiente de similitud.

Un ejemplo del uso de N-gramas es el siguiente:

Palabras → “Esto ejemplifica N-gramas” N=2

El N-grama quedaría así: Esto ejemplifica | ejemplifica N-gramas.

En el sistema, el valor que se ha definido por defecto para el N-grama es de 4, pudiendo variar este valor; si es menor encontrará más coincidencias y si aumenta el valor es posible que no se encuentren gramas iguales entre documentos (se utiliza en análisis por sinonimia).

- **Expresiones regulares:** El sistema detecta si existen referencias basándose en expresiones regulares que las definan. Por defecto vienen incluidas 4 expresiones regulares, es decir, el sistema verificará los 4 tipos de referencias siguientes:
  - **\\[\\d+\\]\\.** Valor numérico contenido entre corchetes y finalizado en punto. Por ejemplo: [2].
  - **\\.\\d+** Punto seguido de un valor numérico. Por ejemplo: *.4*
  - **\\’\\d+** Comillas seguidas por un valor numérico. Por ejemplo: *“3*
  - **\\.\\s*\\[\\d+\\]** Punto seguido de varios o ningún espacio y todo esto seguido de un valor numérico contenido entre corchetes. Por ejemplo: .[1]

  Si sabe cómo escribir una expresión regular que defina el formato de una referencia que usted necesite puede añadir su expresión regular junto a las demás. Para hacerlo deberá encerrar su expresión entre llaves así: **{\\d+\\.}**.

  No olvide que el sistema verifica toda la lista de expresiones regulares. Si usted desea que se busque un solo tipo de referencia deberá borrar todas las demás expresiones regulares y quedarse solo con la que le interesa. No se olvide de encerrar entre llaves aun cuando exista sólo una expresión regular.

  Si desea que se analice todo el texto sin importar si existen o no referencias puede borrar todo el contenido de éste parámetro (se utiliza en análisis textual y análisis por sinonimia).

- **Tamaño máximo de la descarga de ficheros (PDF, DOC Y DOCX):** Cuando se realizan búsquedas en la web es muy probable que los resultados que devuelva la búsqueda no sean solo páginas web, pueden existir diversidad de formatos. El sistema está enfocado en los ficheros con extensión .PDF, .DOC y .DOCX. Se ha limitado el tamaño máximo que debe poseer el archivo, esto con el objetivo de agilizar el proceso de descarga, por defecto se ha definido que el tamaño máximo del archivo sea de 5 megabytes. Si un fichero sobrepasa los 5 megabytes será ignorado y no se descargará continuando así con el proceso de Detección de Plagio. Si considera que este tipo de archivos son muy importantes puede aumentar este valor. Por otro lado, si considera que la descarga de este tipo de archivos no es relevante y desea ganar velocidad en el análisis puede reducir el valor. El valor debe ser escrito en bytes (se utiliza en análisis por sinonimia).
- **Tamaño máximo de archivos de texto plano:** Al igual que en el punto anterior este parámetro indica el tamaño máximo que deberá poseer un archivo de texto plano como pueden ser aquellos con formato TXT. Si el fichero posee un tamaño mayor al establecido no será descargado y se ignorará. Puede aumentar el valor de este parámetro si desea que se descarguen archivos más grandes. Este valor también se encuentra dado en bytes (se utiliza en análisis por sinonimia).
- **Tamaño máximo de descarga de contenido web:** Igual a los 2 puntos anteriores. Se puede especificar en bytes el tamaño máximo que debe poseer un sitio web para empezar a extraer texto desde el mismo (se utiliza en análisis por sinonimia).
- **Número de palabras por página del documento original:** Al realizar análisis por sinonimia se divide el documento que se está analizando en **páginas** y de éstas se extraen las palabras más importantes para ser reemplazadas con sinónimos y realizar las búsquedas. Por tanto, si se eligen pocas palabras el sistema será más preciso en su análisis y así también demorará más en terminar de procesar el documento. Por otro lado, si se eligen muchas palabras por página el análisis será menos preciso a favor de una mayor velocidad en la ejecución del sistema. El valor mínimo de palabras por página deberá ser de *16* y el valor máximo deberá ser como mucho igual al número de palabras que posea el documento (se utiliza en análisis por sinonimia).

- **Tiempo máximo de espera de procesamiento por página:** Igualmente, en el análisis por sinonimia, una vez que se ha encontrado un resultado y se lo ha descargado se deberá comparar dicho documento con la *página* que generó la búsqueda inicial. Si el documento descargado es muy extenso, por ejemplo de más de 500 hojas, puede que procesarlo tome mucho tiempo. Si desea puede reducir éste tiempo máximo para agilizar la ejecución del sistema, mantener igual o aumentar el tiempo en caso de que considere que es muy importante analizar archivos extensos que el sistema descarga. Este tiempo debe darse en milisegundos (se utiliza en análisis por sinonimia).
- **Tiempo máximo de espera para descarga de contenido web:** Si el servidor tiene problemas de conexión o presenta una velocidad lenta de conexión a Internet es posible que al extraer texto desde sitios web, en especial de aquellos con mucho texto en su interior, el sistema se bloquee a la espera de que termine la extracción. En caso de presentarse tal escenario, este parámetro limita el tiempo que el sistema permanecerá bloqueado. En caso de sobrepasar este límite de tiempo se ignora la extracción y el análisis continúa sobre las siguientes *páginas*. Este valor también está dado en milisegundos (se utiliza en análisis por sinonimia).
- **Tiempo máximo de espera para descarga de archivos (PDF, DOC y DOCX):** Al igual que en el parámetro anterior si el sistema debe descargar algún fichero y dicha descarga toma mucho tiempo bloqueando al sistema entonces éste parámetro establece un límite de tiempo, el cual si es sobrepasado anula la descarga y procede con las siguientes *paginas* a analizar. Recuerde que es posible que al analizar muchas *páginas* el sistema proceda a realizar muchas descargas de archivos, por lo tanto, si este valor es muy alto y la conexión muy lenta el sistema puede tomar mucho tiempo en completar el análisis. Este tiempo también se encuentra dado en milisegundos (se utiliza en análisis por sinonimia).

Los siguientes 3 parámetros están relacionados con el servidor. Por tanto, si está instalando el sistema en otro servidor o si conoce que el sistema ha cambiado de servidor éstos parámetros pueden ser útiles (se utilizan en análisis textual y análisis por sinonimia).

- Usa un proxy?: Si el nuevo servidor en donde se aloja el sistema atraviesa un proxy para salir a internet debe activar esta casilla.
- Proxy IP: En caso de que el servidor atraviese un proxy, se deberá indicar la dirección IP de dicho proxy.
- Proxy port: Al igual que en el parámetro anterior si el servidor atraviesa un proxy, mediante este parámetro se especifica el puerto del proxy.

1.3 Solución de Problemas

Problema: Al intentar ejecutar el sistema obtengo un error relacionado con Apache.

Solución: Recargue el sitio web preferentemente en una nueva pestaña y vuelva a intentar.

Problema: He recargado el sitio en una nueva pestaña pero el error sigue apareciendo.

Solución: Si el problema persiste póngase en contacto con el administrador del servidor e infórmele sobre el error que está teniendo, de ser posible coméntele el mensaje y número de error que está recibiendo.

Problema: Soy el administrador del servidor y no sé cómo corregir el problema de un cliente.

Solución: Si el problema está relacionado al servicio Apache deberá reiniciarlo. Es recomendable que lea los archivos log de registros pertenecientes a Apache en búsqueda de algún otro problema.

1.4 Preguntas frecuentes

Pregunta: He indicado al sistema que analice un archivo pero está tomando demasiado tiempo, ¿es esto normal?

Respuesta: Si, el sistema puede tomar mucho tiempo en completar su ejecución especialmente si está realizando análisis por sinonimia y si el archivo que está analizando es extenso.

Pregunta: ¿Soy desarrollador, puedo colaborar con el proyecto?

Respuesta: Por supuesto, el sistema de Detección de Plagio es de código abierto y puede ser modificado y mejorado. Si lo desea puede comunicarse con los desarrolladores del sistema. Sus correos se encuentran en la sección de Introducción del presente manual.