Detecció de text ofuscat per evitar els detectors de plagi

Victor Avila Ferré

Resum—En l'àmbit acadèmic i professional hi ha una corrent molt amplia de plagis en els documents, atemptant contra la propietat intel·lectual de l'autor original. Per combatre'ls existeixen eines anomenades detectors de plagi, que permeten analitzar documents amb possibles irregularitats. Aquests detectors tenen una alta probabilitat de detecció, però poden ser enganyats utilitzant els anomenats mètodes d'ofuscació. Els mètodes d'ofuscació permeten al plagiador utilitzar tècniques no visibles per al sistema visual humà que confonen al detector, fent que els documents fruits del plagi no siguin detectats correctament. Aquest projecte té com a finalitat desenvolupar una llibreria capaç d'analitzar documents per detectar indicis d'ofuscació en el seu contingut. Els detectors de plagi podran fer ús d'aquesta eina per augmentar la seva eficàcia. El desenvolupament del projecte consta d'un treball de camp previ per la cerca de la informació necessària per la seva posterior realització, així com un treball de desenvolupament on es dura a terme la programació i les proves de funcionament de la llibreria.

Paraules clau—Ofuscació, Java, alfabet ciríl·lic, alfabet llatí, Tika, iText, detector, plagi, shuffling obfuscation, singleton, ODT, PDF, TXT, DOC, DOCX.

Abstract—In every academic and profesional field there is a wide variety of plagiarisms in documents, attempting to the intellectual property of the original author. In order to face such attempts, there are tools named "plagiarism detectors" that are able to analyse documents with possible irregularities. These detectors have a high chance of detection, but they can also be deceived by using the so-called obfuscation techniques. Obfusation techniques allow the plagiarist to use non visible techniques for the human visual system which confuse the detector, making plagiarized documents not being detected correctly. The main purpose of this project is to develop a library capable of analysing documents to detect evidences of obfuscation in its content. Plaguiarism detectors will be able to use this tool to increase its efficiency. The development of this project consists in a previous field work in order to search the necessary information for its later realization, and a development work where the programming and the funcionality tests of the library will be held.

Index Terms—Obfuscation, Java, cyrilic alphabet, latin alphabet, Tika, iText, detector, plagiarism, shuffling obfuscation, singleton, ODT, PDF, TXT, DOC, DOCX.

1 Introducció

ACTUALMENT, amb l'evolució de les tecnologies de la informació, la quantitat de fonts d'informació a les quals es pot accedir per ampliar el coneixements sobre qualsevol tema, augmenta exponencialment. A priori aquest augment sembla enormement útil, ja sigui en l'àmbit acadèmic com en el laboral, però pot convertir-se en una arma de doble tall si es utilitzada de manera inadequada. ¿Per que és un gran inconvenient utilitzat de forma inadequada? Perquè aquesta informació pot no ser utilitzada per ampliar els coneixements, sinó per estalviar temps i esforç, presentant-la com una creació pròpia. Aquesta presentació d'informació d'un altre autor com a pròpia s'anomena plagi. El plagi atempta contra la propietat intel·lectual i s'ha de combatre per aconseguir que els autors d'aquesta copia d'informació no obtinguin un reconeixement per allò què no es fruit del seu treball.

L'evolució de les tecnologies de la informació va obrir una porta d'accés a informació a traves d'una gran varietat de fonts d'informació. Aquest accés permet obtenir una extensa documentació vàlida per desenvolupar treballs de manera molt simple. Amb l'augment de les fonts d'informació va sorgir la necessitat d'utilitzar eines amb la capacitat de detectar plagis en documents de forma automàtica. Per cobrir aquesta demanda es varen crear els detectors de plagi.

Els detectors de plagi són de molta utilitat, ja que la seva detecció és molt efectiva, encara que no infal·lible. A mesura que ha passat el temps s'han anat descobrint tècniques per confondre aquests detectors sense tenir que modificar l'aspecte visual del contingut. Aquestes tècniques per confondre els detectors s'anomenen mètodes d'ofuscació de text.

El projecte es composa d'una part relacionada amb la cerca d'informació sobre els mètodes d'ofuscació i una altra part de desenvolupament d'una llibreria capaç de realitzar deteccions dels mètodes d'ofuscació més habituals. L'ús més comú d'aquesta llibreria serà per complementar un detector de plagi i així aconseguir una gamma més amplia de deteccions.

Aquest article estarà estructurat de la següent manera. Primer explicarà els objectius, les motivacions i l'estat de

- *E-mail de contacte: victor.avila@e-campus.uab.cat*

- *Mención realitzada: Tecnologies de la Informació.*
- *Treball tutoritzat per: Jordi Duran Cals (Departament d'Enginyeria de la Informació i de les Comunicacions)*
- *Curs 2015/16*

l'art en aquest projecte. Seguidament, s'analitzarà la metodologia i la planificació utilitzada. El continuarà amb l'expliacació dels resultats obtinguts de la cerca de documentació prèvia al desenvolupament de la llibreria. Els següents apartats seran el desenvolupament de la llibreria i les proves realitzades. Finalment, s'exposaran les conclusions obtingudes amb la realització del projecte.

2 Objectius

Els objectius principals i per tant els objectius que obligadament s'han de complir en aquest projecte són :

- **Recopilar i estudiar tota la informació possible sobre els mètodes d'ofuscació.** Aquesta informació obtinguda de la cerca, serà posteriorment estudiada per entendre els mètodes d'ofuscació i poder establir estratègies per combatre'ls.
- **Desenvolupar una llibreria capaç de detectar text ofuscat per evitar detectors de plagi.** Aquesta llibreria podrà servir per millorar els detectors de plagi, però no farà la seva detecció pròpia de text plagiat. La llibreria treballarà amb barems que s'hauran de superar per considerar que hi ha indicis d'ofuscació. Al no ser un detector amb resultats absoluts, haurà de decidir l'usuari, analitzant els resultats, si el text està realment ofuscat.

Com objectius secundaris que es desenvoluparan per ampliar la qualitat d'aquest projecte són:

- **Crear un eliminador d'ofuscació.** Afegir una funcionalitat en la llibreria que s'encarregui de donar la possibilitat d'eliminar l'ofuscació detectada. Pot ser molt útil per a posteriors deteccions de plagi.
- **Identificar les posicions dels caràcters ofuscats.** Possibilitat d'obtenir les posicions dels caràcters ofuscats per possibles utilitats posteriors, com podria ser el resaltament d'aquests caràcters.

3 Estat de l'art

El plagi no és una pràctica creada en l'actualitat, però ha augmentat el seu ús amb la informació digitalitzada. Realitzar el plagi en un document digitalitzat és molt més senzill i ràpid, tan sols s'ha de copiar i enganxar la informació del document original, sense perdre temps en redactar-ho.

Hi ha molta varietat de detectors de plagi al mercat i en condicions ideals són una bona eina per frenar el plagi, però no sempre les condicions són ideals. En els documents amb les condicions no ideals, ens trobem amb els mètodes d'ofuscació, que confonen els detectors de plagi per evitar la seva correcta detecció.

La majoria de detectors de plagi no contemplen aquestes ofuscacions. Altres les contemplen però no les eviten, sinó que deixen de funcionar quan es troben amb alguna anomalia. Els detectors més avançats i més cars, detecten algun dels mètodes més bàsics. Molts usuaris que no es poden permetre detectors de plagi de pagament, amb l'eina desenvolupada en aquest projecte podran obtenir resultats molt competents i en alguns casos millors que els obtinguts amb els detectors de pagament.

L'ofuscació de text sorgeix de la necessitat d'ocultar el plagi als detectors, i per tant, és una necessitat que va sorgir després de la creació d'aquests detectors. Això fa que sigui relativament nova i no hi hagi exemples de detectors d'ofuscació, així com tampoc hi ha una gran quantitat d'informació disponible sobre aquest àmbit. És per aquest motiu que s'ha de destinar un significant període de temps a la documentació prèvia al desenvolupament de la llibreria de detecció d'ofuscació.

Per el desenvolupament del projecte s'ha cercat i estudiat informació sobre els mètodes d'ofuscació de text existents. Els mètodes identificats en aquest treball de cerca són:

- Manipulació de les capes dels documents PDF[1].
- Substitució de lletres llatines per lletres ciríl·liques[1].
- Substitució d'espais per caràcters no habituals del color de fons[1].
- Canviar el format i modificar la puntuació[6].
- Evitar les bases de dades[4].
- Substituir paraules per sinònims d'aquestes[1][4].
- Canviar l'idioma del text[1].
- Resumir el contingut del text [3].

Els mètodes d'ofuscació identificats la llista anterior i detallats en els següents apartats poden aparèixer en diferents formats segons les seves característiques. Als documents PDF ens podem trobar: 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 i 3.8. Als editors de text més complexes (Microsoft Word, OpenOffice, etc.) ens podem trobar: 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 i 3.8. No es pot utilitzar 3.1 ja que aquesta es realitza modificant les capes dels documents PDF, per tant és exclusiu per aquest format. Als editors de text més senzills ens podem trobar: 3.2, 3.4, 3.5, 3.6, 3.7 i 3.8. No es pot utilitzar 3.1 per el motiu anterior i no es pot utilitzar 3.3 perquè en aquests editors bàsics no existeix la possibilitat de modificar el color de la font de text i per tant els símbols no es poden ocultar.

3.1 Manipulació de les capes dels documents PDF

Els fitxers PDF estan compostos per capes que es poden manipular. Aquest mètode consisteix en alterar la capa de text ocult, sense alterar la capa visual del document. El lector no notarà diferencia en el document, ja que la capa visual està intacta, però no podrà copiar el contingut en un altre lloc, per tant, el detector de plagi no podrà obtenir el text correcte per fer la comprovació.

3.2 Substitució de lletres llatines per lletres ciríl·liques

Aquest mètode s'aprofita de la similitud d'algunes lletres de l'alfabet ciríl·lic respecte de l'alfabet llatí. En la següent taula es pot observar la gran similitud entre aquestes:

Alfabet llatí	Alfabet ciríl·lic
'a'	'а'
'e'	'е'
'i'	'і'
'o'	'о'

Taula 2: Alfabets llatí i ciríl·lic

Aquesta substitució sovint es sol realitzar mitjançant

l'eina de *buscar i reemplaçar* dels editors de text, facilitant la seva aplicació. Existeixen detectors de plagi que detecten aquest tipus d'ofuscació, però hi ha d'altres més senzills, que no estan dissenyats per detectar-la i sobre els quals s'obté un grau d'engany molt elevat. Aquest mètode pertany a la categoria de *shuffling obfuscation*[6].

3.3 Substitució dels espais per caràcters del color de fons

En l'escriptura s'utilitza el caràcter espai per saber quan finalitza una paraula, és a dir, com a separador de paraules. La idea en la qual es basa aquest mètode, és molt senzilla: fer que totes les paraules estiguin unides, sense que la vista humana ho detecti i així dificultar al detector la comparació entre aquestes. La forma amb la qual es porta a terme aquesta idea, és substituint els espais per caràcters que no s'utilitzen en el text (normalment estrangers), canviant el color d'aquests per el del fons sobre el qual s'estan escrivint. En alguns casos poden sorgir problemes amb l'espaiat que es tindran que modificar. Un cop realitzada la tècnica, el lector no s'adonarà del canvi, ja que aquests caràcters ocults deixaran un espai en blanc, similar al espaiat habitual. Aquest mètode pertany a la categoria anomenada *shuffling obfuscation*[6].

3.4 Canviar el format i modificar la puntuació

Aquest mètode consisteix en afegir o eliminar puntuacions i símbols que no destaquin molt per al lector. També inclou canvis de format com podrien ser el canvi de dígits a nombres romans. És molt senzill d'aplicar, però no és tant efectiu com altres mètodes d'ofuscació. Aquest mètode pertany a la categoria anomenada *cosmetic obfuscation*[6].

3.5 Evitar les bases de dades

Els detectors treballen amb bases de dades, ja sigui directament (base de dades pròpia) o bé indirectament (alguna eina que fa servir utilitza una base de dades). Aquest mètode consisteix en evitar la informació d'aquestes bases de dades. Si la informació plagiada no es contemplada per la base de dades, el detector la considerarà original. Per poder fer servir aquest mètode, s'ha de fer un estudi de la base de dades i de la informació que volem plagiar. El problema està en que aquest estudi és molt complex i la majoria d'empreses de detecció no tan sols no aporten tota la informació sobre la base de dades, sinó que afegeixen elements d'imprevisibilitat. És pràcticament impossible saber si una informació està a la base de dades dels detectors.

3.6 Canvi de paraules per sinònims d'aquestes

Per realitzar aquest canvi sols s'han de trobar paraules amb el mateix significat i fer la substitució. Aquesta cerca es pot realitzar de forma manual o automàtica. Com la majoria de detectors comparen cadenes de text, aquests sinònims trencaran aquestes cadenes i confondran els detectors. El gran problema és que realitzar la reescriptura manualment pot ser molt costós, ja que per obtenir bons resultats, s'han de substituir al voltant d'un terç de les paraules. L'altra alternativa, realitzar el procés de forma automatitzada, implica una pèrdua de qualitat en el text, així com una revisió i reescriptura per part d'un ésser humà. Aquest mètode pertany a la categoria de *paraphrasing*[6].

3.7 Canviar l'idioma del text

Els detectors de text no solen tenir el text en diferents idiomes, per tant la traducció d'aquest text es pot utilitzar com a mètode d'ofuscació. Aquesta traducció es pot fer de forma manual o automàtica. De forma manual s'ha de conèixer el llenguatge d'origen i destí. En canvi, en la traducció automàtica, no és necessari el coneixement dels idiomes, però sí és recomanable aquest coneixement per revisar i reescriure el text, ja que mitjançant aquest procés automàtic la informació obtinguda és de baixa qualitat. Aquest mètode pertany a la categoria de *paraphrasing*[6].

3.8 Resumir el contingut d'un text

Una tècnica per evitar els detectors de plagi és resumint i compactant el text que es vol plagiar. En alguns casos la compactació d'una informació sense aportar coneixements propis i mantenint el significat del text original, pot ser considerat com a plagi. Aquest mètode pertany a la categoria de *paraphrasing*[6].

4 METODOLOGIA

La metodologia utilitzada durant el desenvolupament del projecte és l'anomenada **Scrum**[7][8]. Scrum és un tipus de metodologia àgil que divideix el desenvolupament del projecte en Sprints, que estableixen unes tasques a realitzar. Al final de cadascun dels Sprints, el Scrum Team presenta el treball completat al Product Owner i s'estableixen les tasques corresponents al següent Sprint. Els Sprints en aquest projecte tindran una duració de 2-3 setmanes.

Aquesta metodologia Scrum consta de tres rols diferenciats:

- **Product Owner**. Representa la veu del client. S'assegura de que el equip Scrum no es desviï de la perspectiva adequada.
- **Scrum Master**. S'assegura de que el procés Scrum s'utilitza correctament, eliminant els obstacles que impedeixen l'equip dur a terme el Sprint.
- **Scrum Team**. S'encarrega de desenvolupar el producte (anàlisi, desenvolupament, proves, etc.) i té la responsabilitat de l'entrega del producte.

En aquest projecte el rol de Product Owner està assignat al tutor del projecte, Jordi Duran Cals, mentre que els rols de Scrum Master i Scrum Team estan assignat l'alumne que realitza el desenvolupament, Victor Avila Ferré.

Les reunions de Scrum que s'utilitzen per dur a terme el projecte són:

- **Reunió de Planificació del Sprint**. Es realitza al començament de cada cicle de Sprint. La seva missió és seleccionar la feina que es realitzarà durant el Sprint i preparar el Sprint Backlog.
- **Reunió de Revisió del Sprint**. Consisteix en revisar el treball que s'ha completat al llarg del Sprint i presentar el treball completat al Product Owner a través d'una demostració.

A priori aquestes reunions són presencials i acordades prèviament entre el tutor i l'alumne. Si existeixen problemes per establir una reunió presencial, aquesta es pot realitzar mitjançant videoconferència, amb eines com Hangouts o Skype.

Els documents que s'utilitzen al llarg del desenvolupament del projecte són:

- **Product Backlog**. Document per a tot el projecte, on s'especifiquen tots els requisits del projecte amb les seves prioritats. Pot ser modificat per el Product Owner.
- **Sprint Backlog**. Document que especifica el subconjunt de requisits que es desenvoluparan al llarg del següent Sprint. Descriu com es realitzaran els requisits i quina duració tindrà cada tasca que composa un requisit.

En aquest projecte el Product Backlog es modifica mitjançant un consens entre el Product Owner i el Scrum Master.

En el Product Backlog hi ha un llistat d'èpics, històries d'usuari i tasques obtingudes de la descomposició del projecte total, amb els percentatges de treball estimats per cadascun. També hi ha un Scrum Board on cada tasca està en *stop*, *run* o *completed*, segons el estat en cada moment del desenvolupament. Aquest Scrum board també especifica el percentatge de projecte finalitzat. El llistat d'èpics es pot observar en l'*Apèndix 10*.

S'utilitza un document compartit entre l'alumne i el tutor per a cadascun dels documents de la metodologia Scrum, és a dir, un per a cadascun dels Sprint Backlogs i un per el Product Backlog. L'eina amb la qual es realitza aquesta compartició és Google Drive.

4.1 Sprints del Scrum

Els Sprints realitzats al llarg del projecte són de 2 o 3 setmanes cadascun. En tots hi ha feina de documentació no especificada en la taula de Sprints.

Objectius	Sprint
Treball de camp	0
Mòdul d'identificació del tipus de document	1
Interfície bàsica d'usuari	2
Mòdul de detecció del primer mètode d'ofuscació	3
Mòdul de detecció del segon mètode d'ofuscació	4
Mòdul d'eliminació d'ofuscació del text Mòdul de detecció de les posicions ofuscades	5

Taula 1: Sprints realitzats durant el desenvolupament

El **Sprint 0** tenia com objectiu obtenir tota la informació possible sobre l'àmbit del projecte per encaminar el desenvolupament d'aquest. Aquest Sprint té com a nombre el 0 perquè es va decidir la metodologia en aquest, i per tant, és el treball previ a l'aplicació de la metodologia. S'ha considerat Sprint perquè compleix les característiques d'aquests, ja que va ser de la duració establerta, i es va realitzar una feina concreta amb un pes important dins del projecte.

El **Sprint 1** va ser l'inici del desenvolupament, amb l'estudi i proves de les llibreries amb possibles utilitats. També es va dissenyar el primer diagrama de classes per estructurar la posterior programació. Aquest Sprint va finalitzar amb el desenvolupament complet del mòdul d'identificació i extracció.

El **Sprint 2** va continuar el desenvolupament amb la interfície bàsica d'usuari, amb la que poder interactuar amb la llibreria. Aquesta interfície ha anat evolucionant a mesura que ha avançat el desenvolupament i s'han afegit nous mòduls a la llibreria.

El **Sprint 3** es va centrar en el desenvolupament del detector d'ofuscació per substitució de lletres llatines per lletres ciríl·liques. Aquest Sprint va finalitzar amb el desenvolupament del mòdul d'aquesta detecció completament funcional. En aquest Sprint es van fer també millores en el mòdul de detecció.

En el **Sprint 4** es va desenvolupar el detector d'ofuscació per substitució d'espais per caràcters no habituals del color del fons. Aquest Sprint va finalitzar amb el desenvolupament del mòdul d'aquesta detecció completament funcional. En aquest Sprint es van fer millores en la manipulació dels paràmetres globals de la llibreria.

En el **Sprint 5** es van realitzar dos opcionals, el mòdul d'eliminació de l'ofuscació del text i el mòdul de detecció de les posicions ofuscades. En aquest Sprint també es varen fer millores en la mostra de resultats, afegint la classe global que els conté.

6 DESENVOLUPAMENT DE LA LLIBRERIA

En aquesta llibreria es desenvolupa la detecció per als mètodes d'ofuscació que pertanyen al grup de *shuffling obfuscation*. Per detectar aquest tipus d'ofuscació no és necessari tenir el document original que ha estat modificat, ja que la forma de fer aquesta detecció és mitjançant aproximacions estadístiques.

6.1 Frameworks útils

Un dels motius principals de la designació de Java com a llenguatge de programació per el desenvolupament del projecte, és la varietat de frameworks útils que poden ser utilitzades per facilitar algunes tasques de la programació de l'eina.

Després de fer una cerca de les possibles llibreries cal destacar:

- **Apache Tika**[10] és un framework que serveix per detectar el tipus de document i extreure el contingut de diversos formats d'arxiu.
- **iText**[9] és una framework que serveix per crear, adaptar, revisar i mantenir documents en el PDF.

Les dues llibreries poden ser d'utilitat per el desenvolupament de la llibreria, però iText es centra únicament en PDF. En canvi, Apache Tika pot detectar i extreure la majoria de formats dels documents, i per tant, aquesta és el framework escollida per ajudar al desenvolupament.

6.2 Parametrització de la llibreria

Les **dades estadístiques** són fonamentals per la detecció dels mètodes *shuffling obfuscation*, ja que l'estratègia per realitzar aquesta detecció es basa en aproximacions estadístiques. S'ha realitzat una cerca de les dades més importants i els valors idonis per utilitzar-los per defecte en l'aplicació.

Les dades més importants per a dur a terme el mòdul de detecció d'ofuscació per substitució de lletres llatines per lletres ciríl·liques, són les referents a les freqüències

d'aparició de les lletres que poden ser substituïdes per lletres ciríl·liques amb el mateix aspecte. Les lletres que poden ser substituïdes sense repercussions visuals, que tenen una elevada freqüència d'aparició en el llenguatge i que per tant són les analitzades per el detector són: a, e, i , o.

Les freqüències d'aparició d'aquestes lletres en el llenguatge espanyol són[11]:

- 'e' 13.68%
- 'a' 12.53%
- 'o' 8.68%
- 'i' 6.25%

Les freqüències d'aparició d'aquestes lletres en el llenguatge anglès són[12]:

- 'e' 12.7%
- 'a' 8.17%
- 'o' 7.5%
- 'i' 6.97%

En aquest desenvolupament s'utilitza per defecte unes freqüències estàndards obtingudes de la realització de la mitjana aritmètica entre les freqüències de l'anglès i l'espanyol anteriors. La freqüència estàndard utilitzada és:

- 'e' 13.19%
- 'a' 10.35%
- 'o' 8.09%
- 'i' 6.61%

Les dades més importants per a portar a terme el mòdul de detecció d'ofuscació per substitució d'espais per caràcters no habituals del color de fons són:

- La freqüència d'aparició de l'espai. Com a referència de freqüència d'aparició de l'espai s'utilitzarà un anàlisis de la novel·la *La Regenta*, de Leopoldo Alas (Clarín) en idioma espanyol, on la freqüència d'aparició de l'espai és de 17.0599%[11].
- La longitud mitjana de caràcters per paraula. Com a referència de la mitjana de caràcters per paraula s'utilitzarà la mitjana existent en l'obra *Hamlet*, de Shakespeare en idioma anglès, que és de 3.99 caràcters per paraula[14].

La llibreria de detecció d'ofuscació consta d'una classe *Singleton*, que conté els paràmetres globals inicialitzats per defecte amb les dades anteriors. Els valors per defecte inicialitzats no són absoluts, ja que segons l'idioma utilitzat o d'altres variables (globalitat en l'anàlisi, nivell de permissivitat, etc.), potser altres paràmetres siguin més eficients. És per aquests motius que se li ofereix a l'usuari de la llibreria, la possibilitat de modificar aquests paràmetres globals segons els seus propis criteris. Per cobrir aquesta possibilitat s'han afegit *setters*. En la interfície d'usuari que s'ha creat, es pregunta si l'usuari vol fer els anàlisis amb els paràmetres per defecte o prefereix modificar-los. En cas de que no esculli els paràmetres per defecte, haurà d'introduir un per un els seus propis. Tots els valors que es troben en aquest article són els que s'utilitzen per defecte, però poden ser modificats segons les necessitats de l'usuari.

6.3 Funcionalitats implementades

Abans de començar amb el desenvolupament de la llibreria, s'ha dissenyat un diagrama de classes per realitzar una programació estructurada. El diagrama inicial ha anat variant al llarg del desenvolupament, ja que s'ha anat afegint i modificant funcionalitats.

En la llibreria hi ha una classe per cada tipus de document, aquestes classes hereten d'una classe abstracta que conté els mètodes i atributs generals dels documents.

Existeixen dues classes *Singleton*, és a dir, classes que no permeten tindre diferent instancies d'elles mateixes. Aquestes classes corresponen al paràmetres globals de la llibreria i als resultats de les deteccions.

Existeixen també dues classes auxiliars que fan funcions molt concretes. La primera classe s'encarrega de detectar el tipus de document que es vol analitzar, mentre que la segona s'encarrega d'extreure el text que conté el document.

Cadascun dels detectors té una classe que l'implementa.

Finalment, ens trobem amb la classe *Main* que correspon a la interfície d'usuari. Aquesta interfície sols existeix per la realització de proves, és a dir, la llibreria final no contindrà aquesta classe.

Aquest disseny li aporta a la llibreria la propietat d'escalabilitat, ja que es poden afegir més formats, així com més detectors sense haver de fer grans modificacions en el codi. Cada cop que s'afegeixi detectors o formats nous s'afegirà una nova classe al disseny del diagrama, oferint una fàcil ampliació de la llibreria.

El diagrama de classes final es mostra en l'*Apèndix 1*.

A continuació es descriuran les funcionalitats més complexes del desenvolupament.

6.3.1 Mòdul de d'identificació de format i extracció

Aquest apartat es divideix en dos funcions de l'aplicació, la d'identificació i la d'extracció.

La funció d'identificació de format utilitza una crida a un mètode de la classe TypeDetector (encarregada de fer les deteccions de tipus de document mitjançant Tika) i com a resultat d'aquesta s'obté el tipus real del document. Utilitzant aquest mètode es permet eliminar l'ús d'extensions en la llibreria.

```
public class TypeDetector {
     String TypeExtract(File file) throws IOException{
        Tika tika = new Tika();
        String fileType = tika.detect(file);
        return fileType;
     }
}
```

Imatge 1: Codi d'identificació del tipus de document

La funció d'extracció la realitza una classe anomenada DocumentParser. L'extracció depèn del tipus de document, és per això que aquesta classe consta de tants constructors com tipus de documents permesos. Cadascun d'aquests constructors rep com a paràmetres el document de tipus específic i segons el tipus que rep, el tracta de la forma corresponent. Finalment, la classe consta d'un mètode que retorna un String amb el contingut del document.

```
private BodyContentHandler h = new BodyContentHandler();
private Metadata metadata = new Metadata();
private ParseContext pcontext = new ParseContext();
FileInputStream in = new FileInputStream(pdfDocument.getFile());
PDFParser pdfparser = new PDFParser();
pdfparser.parse(in, h, metadata,pcontext);
String text = h.toString();
```

Imatge 2: Codi d'extracció de text del document

6.3.2 Mòdul de detecció d'ofuscació per substitució de lletres llatines per lletres ciríl·liques

En aquesta tècnica de detecció d'ofuscació, el primer pas és dividir el text pla del document a analitzar en parts de 400 caràcters. Cadascuna d'aquestes parts és analitzada per el detector amb cadascuna de les possibles lletres substituïdes.

Per cadascuna de les possibles lletres es realitza un anàlisi que consta de les següents parts:

- Obtenir la freqüència d'aparició de la lletra analitzada en la part de text que s'està treballant.
- Si no existeix una diferència important respecte a la freqüència estadística escollida, es pressuposa que no hi ha ofuscació d'aquest tipus o que aquesta ofuscació és poc significant.
- Si existeix una diferència important respecte a la freqüència estadística escollida, s'obté la freqüència d'aparició de la lletra ciríl·lica amb la que es pot fer la substitució.
- Si no apareix la lletra substituïda amb una freqüència suficientment significativa, es pressuposa que no hi ha ofuscació d'aquest tipus o que aquesta ofuscació és poc significant.
- Si existeix una freqüència suficientment significativa, es considera que hi ha indicis de que el text ha estat ofuscat.

Les parts amb ofuscació es guarden per tal de poder mostrar el resultat un cop finalitzada tota la detecció.
Un cop conclosos els passos anteriors amb totes les lletres, es realitza el mateix procés amb la següent part de 400 caràcters.

```
Parameters parametrosGlobales = Parameters.getInstance();
Results resultados = Results.getInstance();
TypeDetector t = new TypeDetector();
fileType = t.TypeExtract(file);
switch(fileType){
case "application/pdf":
    PdfDocument document = new PdfDocument(path);
    document.cirilicObfuscationDetect();
    document.spaceObfuscationDetect();
    resultados.showSimpleResults();
    break;
case "application/msword":
    DocDocument document3 = new DocDocument(path);
    document3.cirilicObfuscationDetect();
    document3.spaceObfuscationDetect();
    resultados.showSimpleResults();
```

Imatge 3: Exemple d'utilització dels mòduls de detecció

6.3.3 Mòdul de detecció d'ofuscació per substitució d'espais per caràcters no habituals del color de fons

Per a poder realitzar aquest desenvolupament és necessari definir una col·lecció de delimitadors de paraules, formada per: {' ', '.', ',', '-', '_', ':', ';', '#', '&', '%', '*', '+', '/', '\', '=', '@', '|', '~'}.

Aquests separadors serveixen per segmentar el text en paraules, i poder calcular posteriorment la mitjana aritmètica de caràcters per paraula.

També és necessari definir quins caràcters no són habituals en l'escriptura de texts. Per detectar els caràcters no habituals en l'escriptura és més senzill establir quins són els habituals, ja que impliquen una petita porció del total. Com a caràcters habituals en l'escriptura, s'utilitza un rang dins dels caràcters anomenats Basic Latin de UTF-8[13]. Aquest rang està compost per els caràcters UTF-8 existents entre 32 i 126 (ambdós inclosos). La resta de caràcters, són considerats no habituals.

Com en l'anterior detector, el primer pas de l'algorisme és dividir el text en parts de 400 caràcters. Un cop obtingudes les parts es passa a l'anàlisi de cadascuna d'aquestes. Per analitzar cadascuna de les parts individualment, es segueixen els següents passos:

- Segmentar el text en cadenes de caràcters (paraules), utilitzant el diccionari de delimitadors.
- Un cop obtingudes les paraules per separat, es calculen les longituds mitjanes de les paraules que componen el text.
- Si aquesta longitud és semblat a l'esperada, es pressuposa que no hi ha ofuscació o no és suficientment significativa.
- Si aquesta longitud és molt dispar a l'esperada, és a dir molt més gran, s'activa l'alarma de possible ofuscació.
- En aquest moment s'analitzen els caràcters que no s'utilitzen habitualment i que es considerin possibles separadors per l'ofuscació, fent una col·lecció de caràcters no habituals i obtenint la freqüència d'aparició de cadascun.
- Si un caràcter no habitual es troba dins d'un rang de freqüència semblant al que hauria de estar l'espai (separador), es considera que aquest text analitzat està sofrint ofuscació i es guarda el resultat per la seva posterior mostra.

6.3.4 Mòdul de d'eliminació de l'ofuscació del text

Aquest mòdul permet a l'usuari la possibilitat d'eliminar l'ofuscació. Aquesta funció pot ser molt útil per realitzar una posterior detecció de plagi sobre el text net obtingut.

L'algorisme que utilitza aquest apartat és el següent:

- Escollir la primera part a analitzar.
- Comprovar si en aquesta part escollida s'ha detectat ofuscació.
- Si no s'ha detectat ofuscació no es fa ninguna substitució.
- Si s'ha detectat ofuscació, es substitueixen els caràcters substituts per els que haurien de estar en un text sense ofuscació.

Aquest procés es realitza en cadascuna de les parts amb les quals s'ha descompost el text. Un cop realitzat amb totes les parts, es guarda en la classe resultats un array amb totes les parts sense l'ofuscació. Aquesta classe resultats conté un mètode que permet obtenir en un únic String tot el text amb l'ofuscació eliminada.

```
public String toStringRevertedObfuscation(){
    String revertedText = "";
    for (int i=0;i<this.revertedObfuscation.length;i++){
        revertedText = revertedText + this.revertedObfuscation[i];
    }
    return revertedText;
}
```

Imatge 4: Codi del mètode per retornar el text sense ofuscació

6.3.5 Mòdul de de detecció de les posicions dels caràcters ofuscats

Aquest mòdul té com a objectiu donar a l'usuari de la llibreria la possibilitat de ressaltar aquests caràcters ofuscats.

L'algorisme que es segueix per obtenir aquestes posicions és el següent:

- Escollir la primera part a analitzar.
- Comprovar si en aquesta part escollida s'ha detectat ofuscació.
- Si no s'ha detectat ofuscació no es guarda ninguna posició.
- Si s'ha detectat ofuscació, s'analitza cadascun dels caràcters de la part i es guarda la posició dels que coincideixen amb el caràcter substitut.

Aquest procés es realitza en cadascuna de les parts amb les quals s'ha descompost el text.

Un cop realitzat amb totes les parts, es guarda en la classe resultats un array que indica si els caràcters han estat ofuscats o no.

La classe resultats també conté un mètode que permet obtenir un array d'enters, únicament amb els números de caràcter de les posicions on s'ha detectat l'ofuscació.

6.3.6 Mostra de resultats

La mostra de resultats consisteix en una classe *Singeton* amb tots els resultats obtinguts. En aquesta classe es guarden els resultats de les deteccions, les posicions on es troba l'ofuscació i el text amb l'ofuscació eliminada. Cadascun d'aquests resultats està guardat en la seva variable pertinent, que consta d'un *getter* i un *setter* per la seva manipulació. També hi ha un mètode per mostrar la informació detallada, i un altre per mostrar la informació simple. Podem observar aquests dos tipus de mostres de resultats en l'*Apèndix 6*, *Apèndix 7*, *Apèndix 8*, *Apèndix 9*.

Es pot utilitzar aquests mètodes per mostrar la informació o crear una forma pròpia de mostrar els resultats utilitzant els *getters* de les variables de la classe resultats.

6.4 Interfície d'usuari

Aquesta eina no necessita una interfície d'usuari molt complexa, ja que l'objectiu resultat és una llibreria. La finalitat de la interfície d'usuari desenvolupada és realitzar les proves i les demostracions d'ús de la llibreria de detecció d'ofuscació.

Aquesta interfície d'usuari desenvolupada s'executa mitjançant comandes i es compon de:

- Missatge de benvinguda.
- Advertència dels formats permesos.
- Introducció de la ruta del document a analitzar.
- Si el format és incorrecte, mostrarà un missatge comunicant que és erroni.
- Pregunta si es vol utilitzar els paràmetres per defecte. Si no es volen utilitzar, s'han d'introduir els escollits un per un.
- Pregunta sobre si la mostra de resultats ha de ser simple o detallada.
- Mostra dels resultats escollits.
- Pregunta si es vol obtenir el text eliminant l'ofuscació. Si es vol, es mostrarà el text sense ofuscació.

7 Resultats

Aquest apartat tracta les proves realitzades amb la interfície d'usuari sobre la llibreria desenvolupada. Els resultats a aquestes proves seran analitzades a continuació.

7.1 Proves inicials

En la fase de treball de documentació, abans de començar amb el desenvolupament de la llibreria, es van fer proves de funcionament del detector de plagi Viper davant texts amb ofuscació. Aquestes proves tenien com a finalitat comprovar l'efectivitat dels mètodes d'ofuscació per evitar la detecció de plagi en aquests detectors.

Per la realització d'aquestes proves es va instal·lar el software del detector de plagi Viper. Seguidament, es va plagiar un text d'una pàgina d'Internet i es va analitzar amb el detector de plagi. El resultat de la detecció va determinar que en el document existia plagi. Podem veure els resultats a l'*Apèndix 2* i *Apèndix 3*.

Seguidament es va realitzar ofuscació en el contingut del document amb els mètodes:

- Substitució de lletres llatines per lletres ciríl·liques.
- Substitució de espais per caràcters no habituals del color de fons.
- Canviar paraules per els seus sinònims.
- Resumir el contingut text.
- Canviar l'idioma del text.

En tots els documents amb ofuscació, Viper no era capaç d'identificar el plagi en els documents.

7.2 Creació de documents per la realització de les proves.

S'han creat una col·lecció de documents per provar el màxim de funcionalitats de la llibreria. Aquesta col·lecció permet obtenir diversitat de resultats i comprovar la correctesa de l'eina desenvolupada.

Per a les proves d'identificació i l'extracció del contingut del document, s'ha creat un document de cada tipus, amb el contingut *Soy un X*, on X és l'extensió del document.

Per a les proves dels detectors d'ofuscació s'ha creat un document amb una part de la novel·la *El Quijote*. Aquest text s'ha ofuscat per comprovar el funcionament correcte dels detectors.

Per la prova del detector d'ofuscació per substitució de lletres llatines per lletres ciríl·liques, s'ha ofuscat el text anterior amb diferents lletres. D'aquesta manera no obtenim un resultat constant en totes les parts, sinó que aconseguim diversitat de resultats. Podem veure'l en l'*Apèndix 4*.

Per la prova del detector d'ofuscació per substitució d'espais, s'ha ofuscat substituint els espais del texts per un caràcter no habitual. Un cop reemplaçats els espais s'han seleccionats i se'ls ha aplicat el color del fons, així no es distingeix de l'aspecte inicial i el lector no serà capaç de percebre el canvi. Podem veure'l en l'*Apèndix 5*.

Un cop realitzades les proves de detecció amb els documents anteriors, s'ha creat documents combinant ofuscació dels dos tipus. D'aquesta manera podem comprovar si els resultats combinats son correctes.

Finalment, s'ha creat un document amb el resultat de l'eliminació de l'ofuscació per tal de comprovar aquesta

funció.

7.3 Proves del mòdul d'identificació i extracció

Un cop completat el mòdul d'identificació i extracció, s'han realitzat les proves per comprovar el seu correcte funcionament. Aquestes proves consisteixen en:

- Introduir un document.
- Identificar el tipus de document que és.
- Extreure el contingut del document, obtenint un text pla.

En aquetes proves també comprovem la utilitat i el correcte funcionament de la llibreria Apache Tika, ja que totes les funcions que realitzarem amb aquesta, estan programades en aquest mòdul.

Per aquestes proves també s'utilitzen documents amb formats no suportats per la llibreria, per comprovar que són detectats.

Tots els resultats d'aquestes proves són correctes, ja que demostren que el mòdul funciona correctament, inclús en els casos en que l'usuari introdueix documents amb format no suportats.

```
Bienvenido al programa de detección de ofuscacion!
Los formatos compatibles con la aplicacion son: PDF,
TXT, DOC, DOCX, ODT.
Introduzca la ruta del documento a analizar:
C:/Users/Victor/Desktop/proves/pdf.xml
Format incorrecte
```

Imatge 5: Resultat prova de document amb format incorrecte

7.4 Proves de parametrització

En aquestes proves es comprova la correctesa en la introducció de paràmetres manualment. Amb la realització d'aquestes proves s'ha comprovat que no existeix ningun problema en la introducció de paràmetres, i que amb la introducció d'aquests els paràmetres globals es modifiquen correctament.

S'han comparat els resultats de les deteccions utilitzant els paràmetres per defecte amb els resultats de les deteccions utilitzant els paràmetres modificats. Aquests resultats han variat en funció dels paràmetres modificats.

Les proves més visuals per detectar que els resultats es modifiquen, és canviant la longitud de les parts i variant el barem que s'utilitza.

Obfuscated Parts	Part Size = 400	Part Size = 100
Threshold = 1.0	6 of 14	29 of 56
Threshold = 4.0	11 of 14	45 of 56
Threshold = 8.0	11 of 14	48 of 56

Taula 3: Resultats amb diferents paràmetres globals

En la taula 3 podem observar com fent les proves amb la longitud de les parts més petites i el barem més alt, s'obtenen resultats més estrictes. Si es disminueix massa la longitud de les parts, el detector pot deixar de funcionar de forma correcta.També es poden modificar les freqüències per a ajustar-les a diferents idiomes. Amb aquestes modificacions es pot fer més efectiva la detecció.

7.5 Proves de mostra de resultats

En aquestes proves es comproven els tipus de mostra de resultats existents. Hi ha tres tipus de mostra de resultats:

- Mostra de resultats de forma simple.
- Mostra de resultats de forma detallada.
- Mostra de resultats de forma personalitzada.

La mostra de resultats de forma simple permet a l'aplicació mostrar a l'usuari el número de parts on s'ha detectat indicis d'ofuscació i el percentatge que aquestes representen sobre el total de parts. Aquesta mostra permet veure de forma molt directa si s'han detectat indicis d'ofuscació, per tant, si sols es vol saber un resum global de la detecció aquesta pot ser la millor opció.

La mostra de resultats de forma detallada permet a l'usuari mostrar una ampliació de la mostra simple. En aquest cas, a més d'obtenir les parts ofuscades i el percentatge d'ofuscació, obtenim una descripció de cadascuna de les parts ofuscades. En aquesta descripció es mostra el fragment de text on s'ha detectat ofuscació, així com quin tipus d'ofuscació s'ha detectat. En el cas de detecció d'ofuscació per substitució de lletres llatines per lletres ciríl·liques, s'indicarà quines de les possibles lletres han sigut ofuscades en el fragment. Si no existeixen indicis d'ofuscació en ningun dels fragments analitzats, s'indicarà que no existeix ofuscació en el document. Aquest tipus de mostra de resultats es convenient quan es vol analitzar detingudament el resultat de la detecció.

La mostra de resultats de forma personalitzada és l'opció que dona llibertat a l'usuari per mostrar les dades com més li convingui. L'usuari podrà crear els seus propis missatges i utilitzar totes les dades resultants necessàries per crear la mostra. Per accedir a aquestes dades, es realitzaran crides als mètodes *getters* corresponents a les variables desitjades en la classe de resultats. És l'opció ideal per mostrar els resultats, ja que qui fa servir la llibreria pot mostrar la informació resultant al seu gust.

En la interfície desenvolupada es pregunta quin dels tres tipus es prefereix. S'ha creat una mostra de resultats personalitzada per comprovar la correctesa de l'accés a les variables que hi ha en la classe resultats.

Els resultats de les proves realitzades per comprovar la correctesa de la mostra de resultats han estat satisfactoris, ja que s'han mostrat correctament tots els tipus de mostra amb els diferents documents.

7.6 Proves del mòdul de detecció d'ofuscació per substitució de lletres llatines per lletres ciríl·liques

En aquestes proves s'avalua el correcte funcionament del primer mòdul de detecció desenvolupat. Per avaluar el correcte funcionament s'utilitza el document ofuscat amb totes les lletres ciríl·liques possibles. S'utilitzen les dues mostres de resultats, és a dir, la senzilla i la detallada, fent més èmfasi en la detallada. Podem veure els resultats detallats d'aquesta detecció en l'*Apèndix 6*.

Amb els resultats d'aquest test podem comprovar que tenim 11 parts ofuscades de 14 existents. Amb la mostra de resultats detallada es mostra el text ofuscat i la lletra amb la qual s'està ofuscant. Hi ha parts en que alguna de les lletres ofuscades no es detecta, això es degut a que aquestes

lletres apareixen amb una freqüència menor a l'esperada, i per tant no tenen una importància significativa dins de la part analitzada. Es poden ajustar els paràmetres per tal de ser més estrictes amb la detecció. Les parts en les que no es detecta ofuscació, són les quals s'han deixat amb el text original per tal de comprovar que el detector funcioni quan no hi ha ofuscació en alguna de les parts.

```
Introduzca la ruta del documento a analizar:
C:/Users/Victor/Desktop/proves/quijoteCirilic.pdf
Quiere utilizar los parametros por defecto?(S/N)
s
Indique 1=Resultados simples, 2=Resultados detallados
1
 ************************************************************
En el analisis de ofuscacion el resultado es el siguiente:
 ************************************************************
Se ha detectado ofuscacion en: 11 de 14 partes analizadas.
Esto significa un: 78% de partes respecto al total analizado.
```

Imatge 6: Resultat simple de la detecció per ofuscació ciríl·lica

Per la detecció d'ofuscació ciríl·lica no es pot distingir l'ofuscació en el text que es mostra en els resultats, ja que les lletres substituïdes segueixen tenint el mateix aspecte dins del String.

Per finalitzar aquest test, s'han realitzat proves amb el document original sense ofuscació per tal de comprovar el funcionament de la llibreria amb casos de documents no ofuscats. El resultat ha sigut correcte, obtenint 0 parts ofuscades de 14 existents.

7.7 Proves del mòdul de detecció d'ofuscació per substitució d'espais per caràcters no habituals del color de fons

En aquestes proves s'avalua el correcte funcionament del segon mòdul de detecció desenvolupat. Per avaluar el correcte funcionament s'utilitza el document ofuscat amb substitució d'espais per el caràcter "Ȼ" i convertit al color de fons. En aquest anàlisi també es fa més èmfasi en els resultats obtinguts de la mostra de resultats detallada. Podem veure els resultats detallats d'aquesta detecció en l'*Apèndix 8*.

Amb els resultats d'aquest test podem comprovar que tenim 4 parts ofuscades de 14 existents. En aquest document hi ha poques parts ofuscades perquè s'ha ofuscat menys quantitat de text en el document analitzat. Amb la mostra de resultats detallada es mostra el text ofuscat i la lletra amb la qual s'ha substituït l'espai.

```
Introduzca la ruta del documento a analizar:
C:/Users/Victor/Desktop/proves/quijoteSpace2.pdf
Quiere utilizar los parametros por defecto?(S/N)
s
Indique 1=Resultados simples, 2=Resultados detallados
1
************************************************************
En el analisis de ofuscacion el resultado es el siguiente:
************************************************************
Se ha detectado ofuscacion en: 4 de 14 partes analizadas.
Esto significa un: 28% de partes respecto al total analizado.
```

Imatge 7: Resultat simple de la detecció per ofuscació d'espais

Com ja s'ha explicat anteriorment els resultats depenen dels paràmetres i en aquest cas s'han utilitzat els que hi ha per defecte.

A diferència de la detecció anterior, en aquesta si que es pot percebre visualment l'ofuscació en el text que es mostra com ofuscat en els resultats detallats. El motiu és perquè en el String s'elimina el color de fons, eliminant al mateix moment el camuflatge d'aquest mètode.

Finalment, en aquest test també s'ha realitzat una detecció amb el document original, obtenint com a resultat 0 parts ofuscades de 14 existents.

7.8 Proves combinades dels dos mòduls de detecció

Les últimes proves de detecció són les realitzades amb el document ofuscat amb les dues tècniques d'ofuscació. En aquestes proves s'analitza si les dues deteccions són compatibles entre si, i per tant l'una no executa interferències sobre l'altra.

Amb els resultats d'aquest test podem comprovar que tenim 11 parts ofuscades de 14 existents, amb diferents tècniques d'ofuscació en les parts i amb un segment del document sense ofuscació.

Els resultats detecten que en 4 parts es combina l'ofuscació per substitució d'espais amb la de substitució de la lletra "e" llatina per la lletra "e" ciríl·lica. A partir de la cinquena part, ja sols existeix la de substitució de lletres llatines per ciríl·liques.

Amb aquest exemple es corrobora que no hi ha problema en detectar diferents ofuscacions en una mateixa part.

Podem veure els resultats detallats d'aquesta detecció en l'*Apèndix 7*.

7.9 Proves del mòdul d'eliminació de l'ofuscació del text

En aquestes proves s'avalua el correcte funcionament del mòdul d'eliminació d'ofuscació desenvolupat. Per el desenvolupament d'aquesta prova s'ha utilitzat el document que combina les dues deteccions. En la classe *Main* s'ha afegit una pregunta que en cas afirmatiu retorna el text del document amb l'ofuscació eliminada.

Per certificar que l'ofuscació ha estat eliminada correctament, el text resultant s'ha introduït en un document que posteriorment ha estat analitzat de nou per el detector. El resultat ha estat que no existia ofuscació, per tant l'eliminador d'ofuscació realitza correctament la seva tasca. Podem observar el procediment d'aquesta prova en l'*Apèndix 8* i *Apèndix 9*.

8 Conclusions

Els detectors de plagi aporten solucions molt vàlides per combatre els intents de suplantar a l'autor del text, però cada cop és més comú trobar-se amb estratègies d'ofuscació de text que permeten confondre aquest detectors. Per aquest motiu el projecte estava encaminat a l'estudi dels mètodes d'ofuscació de text i el desenvolupament posterior d'una llibreria capaç de detectar els mètodes pertinents a la classe de *shuffling obfuscation*. Aquests objectius han estat assolits completament, deixant així temps suficient per poder desenvolupar els objectius opcionals dels mòduls d'eliminació de text ofuscat i detecció de les posicions dels caràcters ofuscats. Aquests objectius opcionals

també han estat desenvolupats completament.

La utilització d'aquestes tècniques va sorgir per la necessitat d'evitar els detectors de plagi, això fa que tingui poca antiguitat, i per tant l'àmbit de detectors d'aquestes estratègies està poc evolucionat. La poca evolució en el desenvolupament d'eines que detectin aquest tipus d'estratègies fa que aquest sigui un projecte innovador que intenta cobrir les carències en aquest àmbit.

En aquest projecte els objectius principals eren l'estudi dels mètodes d'ofuscació existents i el desenvolupament d'una eina capaç de detectar els mètodes de *shuffling obfuscation*. Aquest mètodes són els que requereixen menys esforç per la seva aplicació, per tant els més comuns i els primers que tindrien que ser detectats.

La llibreria obtinguda en el desenvolupament del codi té com a finalitat principal ser utilitzada com a complement d'un detector, per augmentar així la seva eficàcia. Amb un bon ús d'aquesta eina es podria augmentar molt notòriament la robustesa del detector que l'utilitzi.

En quant a l'àmbit de programació de la llibreria, aquest projecte m'ha proporcionat una visió més completa sobre el desenvolupament orientat a la resolució d'un problema real. D'una eina que no esta plantejada en un enunciat, sinó d'una eina sobre la qual s'ha hagut de buscar informació i desenvolupar una estratègia per cobrir les mancances en el seu àmbit. A nivell personal amb l'estudi de la informació obtinguda en el treball de recerca, he adquirit coneixements en l'àmbit de l'ofuscació abans d'aquest projecte inexistents per a mi. També m'ha proporcionat un major coneixement sobre el desenvolupament de projectes reals amb la metodologia àgil Scrum, que a pesar de sols participar el tutor del projecte i jo, s'ha portat amb la major professionalitat possible.

Les majors dificultats amb les que m'he trobat durant el desenvolupament del projecte han estat la complexitat en la cerca d'informació sobre el tema del projecte, així com la dificultat de definir estratègies per poder desenvolupar els detectors. Aquestes dificultats són degudes a la poca evolució dins de la detecció d'ofuscació en el text per evitar detectors de plagi. No s'han trobat exemples de detectors amb els quals poder obtenir una idea bàsica de com afrontar aquest problema. Els detectors de plagi també han estat un problema degut al cost d'aquests, ja que la majoria són de pagament. El detector utilitzat en les proves (Viper) ha estat bloquejat en la nostra regió durant el desenvolupament del projecte. Aquest contratemps no ha afectat notòriament perquè totes les proves necessàries havien estat realitzades i documentades.

Per a desenvolupaments futurs, es podria intentar combatre el *paraphrasing*, ja és un dels tipus d'ofuscació més importants. Per realitzar aquest detector de *paraphrasing* s'hauria d'utilitzar un detector de plagi com ajuda, i amb la combinació dels dos desenvolupar la nova detecció.

Per finalitzar aquestes conclusions destacar que tant la detecció d'ofuscació com la de plagi, no poden ser automatitzades al 100%, ja que són problemes amb els quals ha de ser un ésser humà el que decideixi en última instància si segons el seus criteris propis, el resultats obtinguts són vàlids. Per tant, aquest detectors són una ajuda que alerta d'uns indicis que han de ser contrastats per qui els utilitzi un cop obtinguts els resultats.

BIBLIOGRAFIA

[1] Gillam, L., Marinuzzi, J., Ioannou, P. "TurnItOff: defeating plagiarism detection systems.", In: 11th Higher Education Academy-ICS Annual Conference, University of Durham, 24–26 Aug 2010, UK.

[2] H. Holi Ali, "Minimizing Cyber-Plagiarism through Turnitin: Faculty's & Students' Perspectives", IJALEL, vol. 2, no. 2, pp. 33-42, 2013.

[3] Roig, M., "Avoiding plagiarism, self-plagiarism, and other questionable writing practices: A guide to ethical writing", August, 2006.

[4] J. Bailey, "5 Sneaky Plagiarist Tricks That Don't Work - Plagiarism Today", Plagiarism Today, 2012. [Online]. https://www.plagiarismtoday.com/2012/08/07/5-sneaky-plagiarist-tricks-that-dont-work/. [Accedit: 01- Mar- 2016].

[5] Neuroskeptic, "How To Fool A Plagiarism Detector - Neuroskeptic", 2014. [Online]. http://blogs.discovermagazine.com/neuroskeptic/2014/04/17/how-fool-plagiarism-detector/#.VtXhnvnhCUl. [Accedit: 01- Mar- 2016].

[6] Kucecka, T., "Plagiarism Detection in Obfuscated Documents Using an N-gram Technique", In: Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 2, 2011.

[7] Es.wikipedia.org, "Scrum", 2016. [Online]. https://es.wikipedia.org/wiki/Scrum. [Accedit: 01- Mar- 2016].

[8] Proyectos Ágiles, "Qué es SCRUM", 2008. [Online]. http://proyectosagiles.org/que-es-scrum/. [Accedit: 01- Mar- 2016].

[9] "iText, a JAVA PDF library", SourceForge, 2016. [Online]. Disponible: https://sourceforge.net/projects/itext/. [Accedit: 13- Apr- 2016].

[10] "TIKA - Overview", www.tutorialspoint.com, 2016. [Online]. Disponible: http://www.tutorialspoint.com/tika/tika_overview.htm. [Accedit: 13- Apr- 2016].

[11] "Frecuencia de aparición de las letras", Wikipedia 2016. [Online]. Disponible: http://ttps://es.wikipedia.org/wiki/Frecuencia_de_aparici%C3%B3n_de_letras. [Accedit: 17- May- 2016].

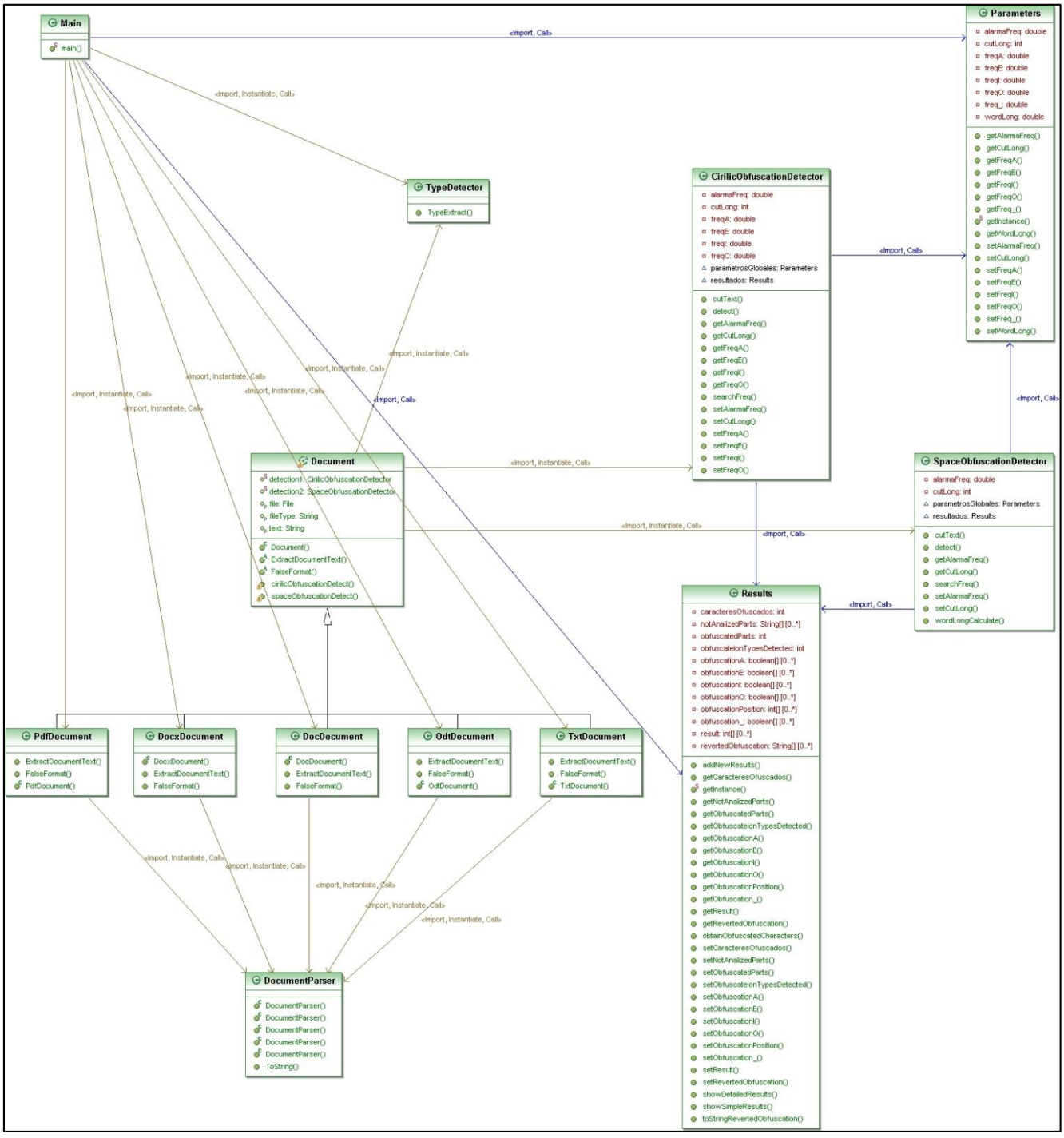
[12] "Letter frequency", Wikipedia, 2016. [Online]. Disponible: https://en.wikipedia.org/wiki/Letter_frequency. [Accedit: 17- May- 2016].

[13] "Unicode/UTF-8-character table", Utf8-chartable.de, 2016. [Online]. Disponible: http://www.utf8-chartable.de/unicode-utf8-table.pl. [Accedit: 17- May- 2016].

[14] "Linguistics Professor Finds Average Word Length In A Tweet Is Longer Than In Shakespeare", Adweek.com, 2011. [Online]. Disponible: http://www.adweek.com/socialtimes/linguistics-professor-finds-average-word-length-in-a-tweet-is-longer-than-in-shakespeare/456798. [Accedit: 17- May- 2016].

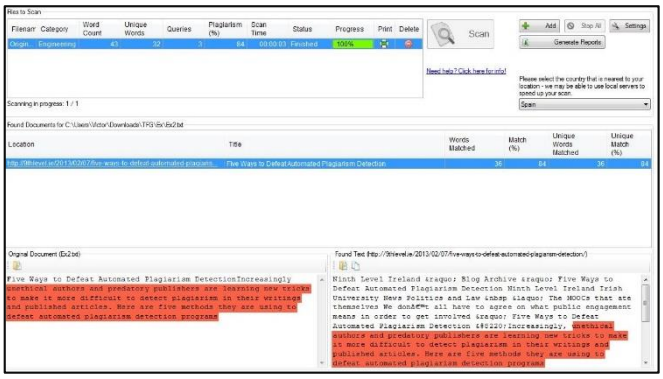
APÈNDIX

A1. DIAGRAMA DE CLASSES

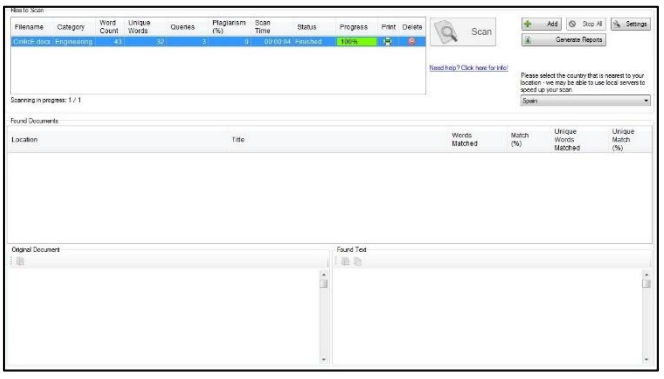


Apèndix 1: Diagrama de classes final de la llibreria amb el main afegit

A2. PROVES DE DETECTOR DE PLAGI VIPER

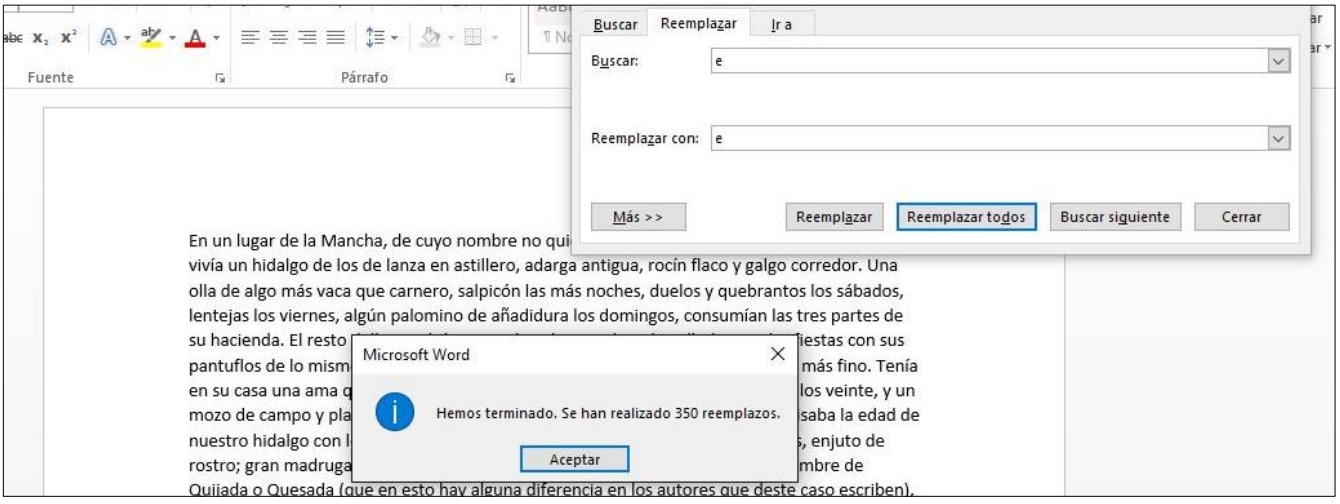


Apèndix 2: Resultats del detector de plagi Viper amb document plagiat original

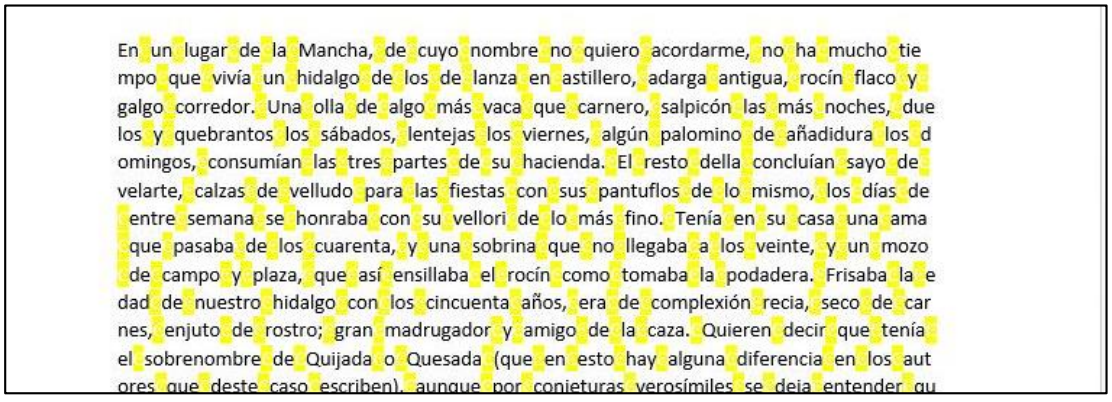


Apèndix 3: Resultats del detector de plagi Viper amb document plagiat ofuscat

A3. DOCUMENTS PER LES PROVES



Apèndix 4: Exemple d'ofuscació per subtitució de "e" llatina per "e" ciríl·lica



En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tie
mpo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y
galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, due
los y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los d
omingos, consumían las tres partes de su hacienda. El resto della concluían sayo de
velarte, calzas de velludo para las fiestas con sus pantuflos de lo mismo, los días de
entre semana se honraba con su vellori de lo más fino. Tenía en su casa una ama
que pasaba de los cuarenta, y una sobrina que no llegaba a los veinte, y un mozo
de campo y plaza, que así ensillaba el rocín como tomaba la podadera. Frisaba la e
dad de nuestro hidalgo con los cincuenta años, era de complexión recia, seco de car
nes, enjuto de rostro; gran madrugador y amigo de la caza. Quieren decir que tenía
el sobrenombre de Quijada o Quesada (que en esto hay alguna diferencia en los aut
ores que deste caso escriben), aunque por conjeturas verosímiles se deja entender qu

Apèndix 5: Exemple d'ofuscació per substitució d'espais resaltada

A3. Resultats de les proves de funcionament

```
--------------------------------------------------------------------------------
A continuacion los resultados detallados del analisis:
--------------------------------------------------------------------------------
Parte 1:
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
Indicios de ofuscacion por substitucion de A por A cirílicas
Indicios de ofuscacion por substitucion de I por I cirílicas
Indicios de ofuscacion por substitucion de O por O cirílicas
--------------------------------------------------------------------------------
Parte 2:
luían sayo de velarte, calzas de velludo para las fiestas con sus pantuflos de lo
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
Indicios de ofuscacion por substitucion de A por A cirílicas
Indicios de ofuscacion por substitucion de I por I cirílicas
Indicios de ofuscacion por substitucion de O por O cirílicas
--------------------------------------------------------------------------------
Parte 3:
mplexión recia, seco de carnes, enjuto de rostro; gran madrugador y amigo de la c
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
Indicios de ofuscacion por substitucion de A por A cirílicas
Indicios de ofuscacion por substitucion de I por I cirílicas
```

Apèndix 6: Exemple d'interfície d'usuari amb els resultats detallats de la detecció d'ofuscació ciríl·lica

```
A continuacion los resultados detallados del analisis:
--------------------------------------------------------------------------------
Parte 1:
En©un©lugar©de©la©Mancha,©de©cuyo©nombre©no©quiero©acordarme,©no©ha©mucho©tiempo©que©viv
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
--------------------------------------------------------------------------------
Parte 2:
luían©sayo©de©velarte,©calzas©de©velludo©para©las©fiestas©con©sus©pantuflos©de©lo©mismo,
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
--------------------------------------------------------------------------------
Parte 3:
mplexión©recia,©seco©de©carnes,©enjuto©de©rostro;©gran©madrugador©y©amigo©de©la©caza.©Qu
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
--------------------------------------------------------------------------------
Parte 4:
©Es,©pues,©de©saber,©que©este©sobredicho©hidalgo,©los©ratos©que©estaba©ocioso©(que©eran©
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
--------------------------------------------------------------------------------
Parte 5:
en©que©leer; y así llevó a su casa todos cuantos pudo haber dellos; y de todos ningunos
Resultados:
Indicios de ofuscacion por substitucion de E por E cirílicas
```

Apèndix 7: Exemple d'interfície d'usuari amb els resultats detallats de la detecció d'ofuscació amb els dos tipus combinats

```
--------------------------------------------------------------------------------
A continuacion los resultados detallados del analisis:
--------------------------------------------------------------------------------
Parte 1:
En©un©lugar©de©la©Mancha,©de©cuyo©nombre©no©quiero©acordarme,©no©ha©mucho©tiempo©que©viví
Resultados:
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
--------------------------------------------------------------------------------
Parte 2:
luían©sayo©de©velarte,©calzas©de©velludo©para©las©fiestas©con©sus©pantuflos©de©lo©mismo,©
Resultados:
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
--------------------------------------------------------------------------------
Parte 3:
mplexión©recia,©seco©de©carnes,©enjuto©de©rostro;©gran©madrugador©y©amigo©de©la©caza.©Qui
Resultados:
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
--------------------------------------------------------------------------------
Parte 4:
©Es,©pues,©de©saber,©que©este©sobredicho©hidalgo,©los©ratos©que©estaba©ocioso©(que©eran©l
Resultados:
Indicios de ofuscacion por substitucion de espacios por caracteres del color de fondo
Quiere obtener el texto eliminando la ofuscación?(s/n)
s
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que viví
```

Apèndix 8: Exemple d'interfície d'usuari amb els resultats detallats de la detecció d'ofuscació d'espais amb eliminació

```
Introduzca la ruta del documento a analizar:
C:/Users/Victor/Desktop/proves/quijoteSpaceCirilicRevertedObfuscation.txt
Quiere utilizar los parametros por defecto?(S/N)
s
Indique 1=Resultados simples, 2=Resultados detallados
1
********************************************************************************
En el analisis de ofuscacion el resultado es el siguiente:
********************************************************************************
Se ha detectado ofuscacion en: 0 de 14 partes analizadas.
Esto significa un: 0% de partes respecto al total analizado.
```

Apèndix 9: Exemple d'interfície d'usuari amb els resultats simples de la detecció d'ofuscació amb el document resultant de l'eliminació

TEMA - DESENVOLUPAMENT DEL PROJECTE	
	EPIC0 - Informe inicial
	EPIC1 - Estudi de llibreries útils Java
	EPIC2 - Identificar el document que es vol procesar
	EPIC3 - Decidir quins mètodes d'ofuscació es tractaran amb més prioritat
	EPIC4 - Implementació de una interfície d'usuari
	EPIC5 - Realització del Dossier 1
	EPIC6 - Implementació tècnica de detecció 1
	EPIC7 - Implementació tècnica de detecció 2
	EPIC8 - Realització Dossier 2
	EPIC9 - Implementació d'opcionals
	EPIC10 - Realització Dossier Final
	EPIC11 - Realització Article
	EPIC12 - Realització i preparació Presentació
	EPIC13 - Realització Póster

Apèndix 10: Llista d'èpics del projecte