

Universidad de las Ciencias Informáticas

Facultad 4



Título: Componente de detección de plagio en foros de la plataforma educativa Zera 2.0.

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Yunierkis Caraballo Mora

Tutor: Msc. Yorgelys González López

Ing. Agustín Castillo Cordero

La Habana.

Julio, 2016

Declaración de autoría

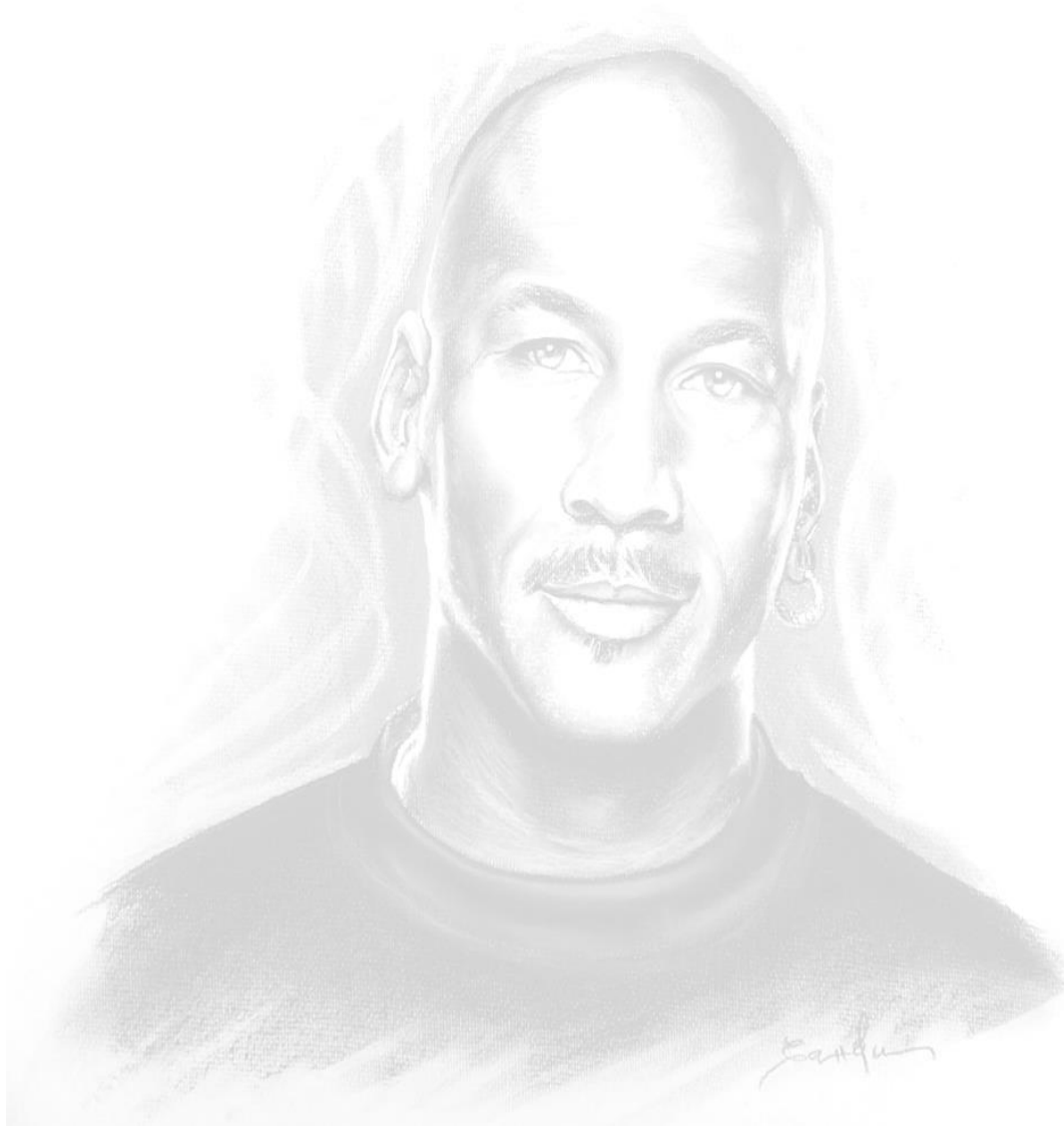
Declaro ser el autor del presente trabajo de diploma y otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yunierkis Caraballo Mora

Msc. Yorgelys González López

Ing. Agustín Castillo Cordero

Pensamiento



“Algunas personas quieren que algo ocurra, otras sueñan con que pasara, otras hacen que suceda”.

Michael Jordan

Dedicatoria

A mi mamá que ha sido mi máxima inspiración durante este trabajo, a ti te debo mi vida, gracias por todo.

Agradecimientos

A mis padres por su dedicación hacia mí y todo el cariño que me han dado siempre, gracias por formarme en la vida como la persona que soy y estoy muy orgulloso de ser tanto como ustedes, por marcar el camino correcto siempre a seguir, por confiar en mí, los quiero con la vida.

A mi hermano Yunior que me han apoyado tanto durante todos estos años y se ha portado como un padre más. Y a mi hermano Yunier que siempre me ha visto como su ejemplo y ha confiado en mí.

A mi novia Cleydi por su incondicional compañía, su entrega y por brindarme una segunda familia la cual se ha preocupado mucho por mí, te quiero.

A mi amigo Juan Carlos por portarse súper bien conmigo, no tengo manera para agradecerle toda su ayuda y horas de dedicación en el desarrollo del trabajo.

A mi tutor Agustín, ojalá todos puedan tener un tutor como él, nunca me dijo que no cuando se trataba de ayudarme y si tenía mucho trabajo abría un hueco y me atendía.

A mis tres amigos de siempre Randy, Adrián el ruso y Yaciel, que aunque estuvieron lejos de mí, no dejaron de llamarme y de animarme a continuar mis estudios.

A mis amigos de la universidad Víctor, Romilio, Neobel, Leonardo, El Pina y mis compañeros de apartamento, gracias a todos.

A todas aquellas personas que de una forma u otra aportaron su granito de arena y mostraron su interés por el desarrollo del trabajo: Dalianne, Andy, Yaritza, Katira, Roberto, Felo, Monedero, Marcos y Rogelio, gracias a todos y por todo.

Resumen

El plagio se considera como una alteración de la originalidad al constituir una actividad mecánica y poco creativa por parte de los estudiantes. ¿Cómo contribuir a la detección de plagio en los foros de la plataforma educativa Zera 2.0? es el problema científico que fundamenta la necesidad de desarrollar un componente que permita la detección de plagio para los foros de la plataforma Zera 2.0; siendo este el objetivo general de la presente investigación. Una vez descrito el plagio como fenómeno y su detección en las diferentes formas existentes, se caracterizaron y valoraron sistemas similares de gran uso mundialmente. Se documentaron los elementos tenidos en cuenta para la selección de las herramientas, lenguajes de programación y otras tecnologías. Conforme a la elección de AUP-UCI como metodología a utilizar en el desarrollo, se documentó la captura de requisitos, descripción de historias de usuario, modelo de análisis y diseño entre otros elementos que hicieron posible una implementación fluida y libre de excesivos cambios o errores. Se logró la implementación de un componente que permite la detección automática de plagio en los foros de la plataforma educativa Zera 2.0, para facilitar de esta forma la evaluación de los estudiantes por parte de sus profesores.

Índice

INTRODUCCIÓN.....	11
Capítulo 1: Fundamentación teórica.	16
1.1 Introducción.....	16
1.2 ¿Qué es plagio?	16
1.2.1 Detección de plagio	17
1.2.2 Campos de aplicación de la detección de plagio.....	17
1.2.3 Algoritmos para la detección de plagio en textos	18
1.2.4 Herramientas para la detección de plagio en textos.....	21
1.3 Plataforma educativa (Mooc).....	23
1.3.1 Plagio en la plataforma educativa (Mooc).	23
1.4 Metodología de desarrollo.	24
1.5 Herramientas y tecnologías a utilizar.....	31
1.5.1 Lenguaje y tecnologías del lado del cliente	31
1.5.2 Lenguaje y tecnologías del lado del servidor.....	33
1.5.3 Entorno de Desarrollo Integrado NetBeans.....	34
1.5.4 Lenguajes y tecnologías de modelado.....	34
1.5.5 Sistema gestor de base de datos PostgreSQL.....	35
1.5.6 Administrador de bases de datos PgAdmin.....	36
1.5.7 Servidor Web Apache	36
1.6 Conclusiones parciales.....	37
Capítulo2: Desarrollo de la solución propuesta.....	38
2.2 Propuesta del sistema	38

2.2.1 Técnica de aprendizaje automático para la clasificación	39
2.3 Modelo de dominio	41
2.3.1 Clases conceptuales del modelo de dominio	41
2.4 Requisitos funcionales.....	42
2.5 Requisitos no funcionales.....	42
2.6 Historias de Usuario	43
2.7 Actores del Sistema	49
2.8 Modelo de Diseño	49
2.8.1 Diagramas de clases del diseño	49
2.8.2 Diagramas de secuencia del diseño.....	51
2.8.3 Modelo de datos	53
2.9 Patrones de diseño	54
2.10 Patrón arquitectónico.....	55
2.11 Conclusiones parciales.....	56
Capítulo 3: Implementación y prueba.....	57
3.1 Introducción.....	57
3.2 Modelo de Implementación	57
3.2.1 Diagrama de despliegue	58
3.2.2 Diagrama de componentes	59
3.3 Pruebas de software	59
3.4 Nivel de prueba	60
3.4.1 Tipo de prueba: Prueba funcional	60
3.4.2 Método de Caja Negra.....	61
3.5 Diseño de casos de prueba.....	61

3.6 Resultados de las pruebas	64
3.7 Conclusiones parciales.....	67
Conclusiones Generales.....	68
Recomendaciones	69
REFERENCIAS BIBLIOGRÁFICAS.....	70

Índice de imágenes

<i>Ilustración 1: Técnica de aprendizaje automático Naïve Bayes.</i>	39
<i>Ilustración 2: Modelo del dominio.</i>	41
<i>Ilustración 3: Diagrama de clase del diseño</i>	50
<i>Ilustración 4: DS- Analizar similitud del comentario enviado con otros comentarios existentes.</i>	51
<i>Ilustración 5: DS-Eliminar el resultado de comparación de plagio y volver a analizar el comentario.</i>	51
<i>Ilustración 6: DS-Mostrar los comentarios similares a un comentario específico.</i>	52
<i>Ilustración 7: DS-Mostrar el resultado del análisis de cada algoritmo en un comentario.</i> ..	52
<i>Ilustración 8: DS-Exportar resultado de comparación de un comentario en formato PDF.</i>	53
<i>Ilustración 9: Modelo entidad-relación.</i>	54
<i>Ilustración 10: Modelo Vista Controlador</i>	56
<i>Ilustración 11: Diagrama de despliegue.</i>	58
<i>Ilustración 12: Diagrama de componentes.</i>	59

Índice de tablas

<i>Tabla 1: Fases AUP-UCI.....</i>	26
<i>Tabla 2: Disciplinas AUP-UCI.....</i>	28
<i>Tabla 3: HU-Analizar similitud del comentario enviado con otros comentarios existentes.</i>	43
<i>Tabla 4: HU- Eliminar el resultado de comparación de plagio y volver a analizar el comentario.....</i>	44
<i>Tabla 5: HU- Mostrar los comentarios similares a un comentario específico.</i>	45
<i>Tabla 6: HU-Mostrar el resultado del análisis de cada algoritmo en un comentario.</i>	46
<i>Tabla 7: HU-Exportar resultado de comparación de un comentario en formato PDF.....</i>	47
<i>Tabla 8: Descripción de los actores del sistema.</i>	49
<i>Tabla 9: CP-Eliminar el resultado de análisis de plagio y volver a analizar el comentario.</i>	62
<i>Tabla 10: CP-Analizar similitud del comentario enviado con otros comentarios existentes.</i>	62
<i>Tabla 11: CP-Mostrar los comentarios similares a un comentario específico.</i>	63
<i>Tabla 12: CP-Mostrar el resultado del análisis de cada algoritmo en un comentario.</i>	63
<i>Tabla 13: CP-Exportar resultado de comparación de un comentario en formato PDF.....</i>	63
<i>Tabla 14: No conformidades encontradas en la Iteración 1.....</i>	64
<i>Tabla 15: No conformidades encontradas en la Iteración 2.....</i>	65
<i>Tabla 16: Cantidad total de no conformidades en cada iteración.</i>	66

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) son un medio para transmitir y gestionar información y conocimiento. Las profundas transformaciones sociales que han provocado, muestran lo importante y necesario que resulta en la actualidad que los procesos formativos universitarios las empleen en su perfeccionamiento y de este modo contribuir a la formación de los profesionales que la sociedad necesita. La introducción de las TIC en el proceso de enseñanza-aprendizaje tiene gran repercusión en la expansión de procesos formativos que utilizan la modalidad a distancia y semipresencial (1).

Con la introducción de las TIC en el proceso de enseñanza-aprendizaje se contribuyó al surgimiento del aprendizaje electrónico (abreviatura en inglés e-learning), el mismo consiste en la educación y capacitación a través de Internet. Este tipo de enseñanza en línea permite la interacción del usuario con el material mediante la utilización de diversas herramientas informáticas. Además, es una revolucionaria modalidad de capacitación que posibilitó Internet, y que hoy se posiciona como la forma de capacitación predominante en el futuro. Este sistema ha transformado la educación, abriendo puertas al aprendizaje individual y organizacional. Es por ello que hoy día está ocupando un lugar cada vez más destacado y reconocido dentro de las organizaciones empresariales y educativas (2).

Una plataforma de e-learning es un espacio virtual de aprendizaje orientado a facilitar la experiencia de capacitación a distancia, tanto para empresas como para instituciones educativas. Este sistema permite la creación de "aulas virtuales"; en ellas se produce la interacción entre tutores y alumnos, y entre los mismos alumnos; como también la realización de evaluaciones, el intercambio de archivos, la participación en foros, chats, y una amplia gama de herramientas adicionales (3).

Beneficios de una plataforma de e-learning (4):

- Brinda capacitación flexible y económica.
- Combina el poder de Internet con el de las herramientas tecnológicas.

- Anula las distancias geográficas y temporales.
- Ofrece libertad en cuanto al tiempo y ritmo de aprendizaje.
- Posibilita un aprendizaje constante y nutrido a través de la interacción entre tutores y alumnos.

Se entiende por plataforma educativa como un sitio en la Web, que permite a un profesor contar con un espacio virtual en Internet donde sea capaz de colocar todos los materiales de su curso, incluir foros, wikis, recibir tareas de sus alumnos, desarrollar test, promover debates, chats y obtener estadísticas de evaluación, todo esto con fines pedagógicos (5).

Los Cursos en Línea Abiertos y Masivos (MOOC) como tipos de plataformas educativas han irrumpido de forma acelerada en el ámbito de la educación. Las principales universidades estadounidenses primero y, posteriormente, muchas otras entre ellas, algunas españolas están desarrollando a través de diversas plataformas en Internet, cursos bajo este formato. MOOC es un curso en línea destinado a la participación ilimitada y acceso abierto a través de la web. Además de los materiales de un curso tradicional, como son los vídeos, lecturas y cuestionarios, estos proporcionan foros de usuarios interactivos que ayudan a construir una comunidad para los estudiantes y profesores (6).

En las plataformas educativas los foros se pueden utilizar para cualquier actividad que requiera debate, discusión, etc. Lo más importante en su uso es especificar con mucha claridad para qué se utiliza cada espacio y moderar correctamente las dinámicas de comunicación. Algunas de las aplicaciones más interesantes de los foros son:

- Uso como espacio para exposición de preguntas frecuentes.
- Propuesta de temas de discusión para llegar a una conclusión final.
- Fomento de la socialización mediante espacios de comunicación no formal.
- Uso como herramienta básica de tutoría docente-estudiantes (es preferible en general a la tutoría individual porque permite resolver dudas recurrentes en público).
- Uso como herramienta de propuesta de temas y toma de decisiones.

- Uso como espacio de coordinación de actividades, resolución de dudas de cada actividad o tema, etc.
- El foro es el espacio donde mejor se atestigua la presencia del factor humano en el e-Learning (7).

La Universidad de las Ciencias Informáticas (UCI), apostó por el uso del e-learning con la creación del Centro de Tecnologías para la Formación (FORTES). En este centro surge la necesidad de una plataforma de cursos en línea masivos y abiertos, Zera 2.0. Esta plataforma educativa es un entorno informático en el cual se pueden encontrar muchas herramientas agrupadas y optimizadas para fines docentes. La misma cuenta con un conjunto de foros como espacios para introducir uno o varios temas, con el objetivo de lograr la participación de los estudiantes y evaluar a los mismos.

Para una correcta evaluación de los estudiantes, es imprescindible para un profesor tener en cuenta la verdadera autoría de sus comentarios; es decir, la seguridad de que no realizó un plagio. En los diferentes foros de la plataforma educativa Zera 2.0 se dificulta la detección rápida y eficiente de similitudes en los comentarios, ya que al ser esta plataforma un MOOC con una matrícula abierta, cuenta con una base de datos que posee un amplio volumen de información.

Por todo lo antes planteado, surge como **problema de investigación**: *¿Cómo contribuir a la detección de plagio en los foros de la plataforma educativa Zera 2.0?*

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar un componente que permita la detección de plagio para los foros de la plataforma Zera 2.0.

Por tanto, el presente trabajo centra su **objeto de estudio** en: proceso de detección de plagio en plataformas educativas y como **campo de acción** se tiene: sistemas para la detección de plagio en los foros de la plataforma educativa Zera 2.0.

Teniendo como referencia el problema a resolver y el objetivo general se plantea la siguiente **hipótesis**: Si se desarrolla un componente capaz de detectar el plagio en los

foros de la plataforma Zera 2.0, entonces los profesores podrán evaluar de una forma más segura y eficaz los comentarios enviados por los estudiantes.

Dicho objetivo general se desglosa en los siguientes **objetivos específicos**:

- Realizar un estudio del estado del arte.
- Identificar las funcionalidades y clases que deben ser implementadas.
- Realizar el diseño e implementación de las funcionalidades definidas.
- Aplicar pruebas que certifiquen la correcta ejecución de las funcionalidades desarrolladas.

Métodos Científicos

Métodos teóricos:

Analítico-sintético: Para el estudio de la teoría y la documentación relacionado con el objeto de estudio, permitiendo concluir las características del componente a desarrollar así como las principales tecnologías y herramientas a utilizar.

Histórico-lógico: Para estudiar la evolución histórica de los sistemas para la detección de plagio que son utilizados por plataformas educativas.

El presente documento se estructura en 3 capítulos que abarcan toda la investigación realizada:

Capítulo1: En este capítulo se definen los conceptos asociados a la investigación y se realiza un análisis de estudios anteriores relacionados con el tema para la elaboración del marco teórico de la investigación. Además, se describen las herramientas, metodologías y tecnologías a utilizar en la solución.

Capítulo2: Este capítulo describe los detalles principales relacionados con el análisis y diseño de la solución propuesta. Como punto de partida se define un modelo de dominio para dar paso a la identificación de los requisitos funcionales. Además, muestra la utilización de la metodología de desarrollo AUP-UCI enfocada en la investigación para el desarrollo de los diagramas y modelos definidos.

Capítulo3: En el capítulo se reseña el proceso de elaboración de los diagramas de componentes y de despliegue así como otros aspectos relacionados con la implementación. Se exponen los resultados de las pruebas aplicadas al sistema para comprobar la calidad y fiabilidad de la solución obtenida, asegurando su correcto funcionamiento.

Capítulo 1: Fundamentación teórica.

1.1 Introducción

En el presente capítulo se realiza un estudio del estado del arte de la detección de plagio en los foros de las plataformas educativas. Los objetivos principales son caracterizar la problemática, analizar el componente ético que convierte a la práctica del plagio en una actitud negativa, identificar los principales algoritmos, metodologías y herramientas utilizadas para la detección, caracterizar las plataformas educativas e investigar en ellas la presencia y detección de plagio a partir de herramientas existentes. Además, identificar las tecnologías necesarias a utilizar, para disminuir el tiempo de ejecución en la detección de plagio en los foros.

1.2 ¿Qué es plagio?

Entre las muchas posibilidades que ofrecen las nuevas tecnologías, se encuentra la opción de "*copiar y pegar*", tan al día en el entorno académico. Muchos creen que lo que está en Internet es un bien que no tiene autor, y que por lo tanto, está a disposición de quien quiera apropiarse de él para usarlo como quiera, sin tener que indicar autoridad alguna o fuente. Pero lo que circula en Internet, es el resultado del trabajo de personas cuya autoría debe ser reconocida (8).

La Real Academia Española define que el plagio es usar el trabajo, las ideas, o las palabras de otra persona como si fueran propias, sin acreditar de manera explícita de donde proviene la información. Es un tema debatido en las disímiles universidades pues se considera como una alteración de la originalidad al constituir una actividad mecánica y poco creativa por parte de los estudiantes. La apropiación de ideas, afirmaciones y textos de forma digital, es cada vez más habitual y creciente en el ámbito universitario, lo cual constituye un hecho delictivo por copiarlos y hacerlos pasar por su propia autoría (9).

1.2.1 Detección de plagio

La detección automática de plagio se basa en diversas técnicas de recuperación y extracción de información, así como de reconocimiento de formas, teoría de la información y procesamiento de lenguaje natural (10).

Esta tarea no implica únicamente el análisis de textos completos para determinar si han sido escritos por un autor determinado o no, sino que busca analizar un documento (o uno de sus fragmentos) para intentar determinar si realmente fue escrito por el autor que reclama haberlo hecho. Existen dos vertientes principales que buscan dar solución a este problema que, debido a su naturaleza, no son capaces de ofrecer el mismo tipo de información tras el análisis realizado. El primero de ellos es el conocido como análisis intrínseco de plagio, en el que el único recurso utilizado es el texto lo sospechoso por sí mismo. El segundo de ellos es la detección de plagio con referencia en donde se requiere contar con un conjunto de documentos originales con el afán de buscar el origen de los fragmentos potencialmente plagiados dentro de un texto sospechoso (10).

1.2.2 Campos de aplicación de la detección de plagio

El plagio puede ocurrir en distintos escenarios: ámbitos educativos, académicos o profesionales. Los documentos plagiados pueden ser informes, tareas de la escuela o la universidad, artículos periodísticos o códigos fuentes (11).

Existen distintas clasificaciones en cuanto a que partes de un texto (o lengua hablada) se pueden plagiar, ejemplo de esto son los siguientes (12):

De ideas: Este tipo de plagio es independiente de las palabras. A adopta las ideas, pensamientos o teorías de otra persona sin darles el crédito adecuado. Este tipo de plagio ocurre cuando la idea fuente no forma parte del conocimiento común.

Palabra por palabra: Se trata de la copia de una (parte importante de una) frase. *A* puede realizar una copia exacta e incluso efectuar algunas modificaciones. Si no hay referencia a la fuente, se estaría cometiendo plagio.

De fuentes: *A* incluye las referencias bibliográficas que otro autor ha incluido en su propio documento *D*. Sin embargo, *A* no señala que dichas referencias han sido extraídas de *D*. En ocasiones, *A* incluye las referencias sin siquiera haberlas consultado.

De autoría: *A* presume ser el autor de un documento entero que en realidad ha sido escrito por otra persona. Esto ocurre a menudo con estudiantes que entregan reportes de otras personas como suyos.

Por todo lo antes investigado se hace uso de la clasificación *palabra por palabra* al ser esta la más ajustable al problema de investigación planteado.

1.2.3 Algoritmos para la detección de plagio en textos

A nivel global existen varios algoritmos que, bajo diferentes conceptos, permiten detectar las coincidencias existentes entre dos documentos. A continuación se detallan algunas de las características particulares de cada uno de los algoritmos que más resaltan en el estudio realizado:

N-gramas: Los n-gramas son trozos de n palabras del texto. El empleo de estos proviene de los modelos de lenguaje y su utilización en el reconocimiento del habla. Los métodos basados en n-gramas tienen la misma estructura: se toman n-gramas del documento en general de forma superpuesta, lo cual hace que la cantidad de n-gramas de un texto de r palabras sea igual a $r - n + 1$ (13).

La correcta selección de los n-gramas de un documento es muy importante. Por ejemplo, si se eligen trozos demasiado pequeños, la probabilidad de que se repitan en otros textos será muy grande, sin importar que sean textos independientes (sin plagio). Por otro lado, elegir trozos muy grandes disminuye la posibilidad de que se encuentren en otro

documento y las pequeñas modificaciones o reescrituras como la omisión o cambio de alguna palabra, evitaría que las porciones plagiadas fueran detectadas. Este método suele ser combinado con otros métodos de análisis más detallados. El siguiente ejemplo muestra cómo se aplica la técnica basada en n-gramas de palabras, con $n = 3$. Los n-gramas que coinciden en ambos textos se muestran resaltados (13).

- Texto 1: “Plagiar es copiar en lo sustancial obras ajenas, dándolas como propias”.
- Texto 2 (sospechoso): “Plagiar es reusar en lo sustancial palabras ajenas, dándolas como propias”.

N-gramas del Texto 1: [plagiar es copiar], [es copiar en], [copiar en lo], [**en lo sustancial**], [lo sustancial obras], [sustancial obras ajenas], [obras ajenas dándolas], [**ajenas dándolas como**], [**dándolas como propias**].

N-gramas del Texto 2: [plagiar es reusar], [es reusar en], [reusar en lo], [**en lo sustancial**], [lo sustancial palabras], [sustancial palabras ajenas], [palabras ajenas dándolas], [**ajenas dándolas como**], [**dándolas como propias**].

Vector Space Model: El modelo de espacio vectorial es otro de los métodos utilizados para la detección de plagio. Basa su funcionamiento en la representación del contenido de los documentos en términos de vectores. Posteriormente, mediante fórmulas matemáticas, arroja los resultados de las similitudes. Según este modelo, cada expresión del lenguaje natural puede representarse como un vector de pesos de términos, o la unidad mínima de información, como una palabra o la raíz sintáctica de una palabra. Para determinar la similitud que existe entre un documento y una consulta se calcula la distancia que existe entre los vectores que los representan (13).

Este algoritmo calcula la similitud entre dos documentos utilizando la fórmula del coseno. Como resultado, se obtiene el valor del ángulo entre los vectores que representan los mismos. Mientras más pequeño sea el ángulo, más similares serán estos documentos. La desventaja de esta técnica radica en que el cambio del orden de las palabras puede cambiar el sentido de una oración, y precisamente este es un hecho que no se toma en

cuenta. Por lo que para hacerlo efectivo es necesario combinarlo con alguna otra técnica, por ejemplo con el análisis de n-gramas de palabras. A continuación se muestra un ejemplo de cómo se aplica el método (13).

Suponiendo que se tienen los siguientes textos de referencia (13):

- Texto 1: "Plagiar es reusar las ideas, procesos, resultados o palabras de alguien más sin mencionar explícitamente a la fuente y su autor".
- Texto 2: "Plagiar es copiar en lo sustancial obras ajenas, dándolas como propias".

Términos: [plagiar, reusar, copiar, inteligencia, ideas, artificial, procesos, alguien, palabras, fuente, autor, obras, propias, aprendizaje].

A partir de los términos anteriores se representan los vectores asociados a cada texto. Suponiendo que los pesos se asignen de la siguiente manera: 1 si aparece el término en el texto y 0 si no aparece, los vectores quedarían como sigue (13):

Texto1= [1 1 0 0 1 0 1 1 1 1 1 0 0 0]

Texto2= [1 0 1 0 0 0 0 0 0 0 1 1 0]

La fórmula del coseno para calcular la distancia entre los vectores es la siguiente:

$$\cos (Vx, Vy) = (Vx * Vy) / (|Vx| * |Vy|)$$

Dónde: Vx y Vy son los vectores de los textos 1 y 2 respectivamente.

Distancia de compresión normalizada (NCD por sus siglas en inglés): Es en realidad una familia de funciones que tienen como argumentos dos objetos (textos) y evalúa una fórmula fija expresada en términos de las versiones comprimidas de estos objetos, por separado y combinados. Por tanto, esta familia de funciones es parametrizada por el compresor utilizado. Si (x) y (y) son los dos objetos en cuestión, y C (x) es la longitud de la versión comprimida de (x) de usar el compresor C, entonces la:

$$NCD(x,y) = (C(x,y) - \min\{C(x), C(y)\}) / \max\{C(x), C(y)\}$$

1.2.4 Herramientas para la detección de plagio en textos.

Unplag: Motor de detección de similitud que comprueba trabajos académicos literalmente y al instante. Se lleva a cabo el registro de entrada en tiempo real contra índice Web en tiempo real asegurando que se reciban los resultados de similitud más precisos. Nos permite escanear documentos contra índices web de Bing y Yahoo en tiempo real. Para comparar su papel con cualquier otro documento dentro de su cuenta se ha creado una comparación de Docs vs Docs. Se tarda sólo 4 segundos por página para completar su comprobación de plagio. Se guarda el formato de archivo inicial mientras se aplica "máscara de duplicado" encima del texto de documento. El informe de plagio lo podemos revisar en línea o descargarlo en formato .pdf. Gestión de archivos fáciles y en formatos múltiples, comprueba .doc, .docx, .rtf, .txt, .odt, .html o .pdf. Después de que la comprobación está concluida, se puede ver la tasa de similitud y originalidad y una lista completa de fuentes coincidentes. Las similitudes encontradas dentro de un texto están resaltadas. Para ver las fuentes que necesitan citadas, se echa un vistazo a la tasa de plagio que cada fuente tiene. Unplag escanea toda la Web: repositorios, fuentes de contenido abierto y las bases de datos de la institución. Proporcionando resultados en tiempo real y no hay datos almacenados en caché (14).

Sus educadores pueden hacer sus propias bases de datos de los trabajos de los estudiantes directamente en sus cuentas Unplag personales. Con posibilidades de crear múltiples cuentas de los estudiantes, organizar todos sus documentos presentados en carpetas y permitir acceso a los archivos compartidos, ver los datos de contacto de los estudiantes, así como el historial de todas las comprobaciones, un instructor puede hacer frente con el flujo de trabajo de rutina mucho más rápido y contribuir mejor para prevenir el plagio en la universidad (15).

De Unplag se puede destacar la velocidad en realizar las comprobaciones y la gestión de archivos en diferentes formatos. En sentido negativo, imposibilidad de utilizarlo porque debe llevarse a cabo un pago por su uso, además, utiliza la Web para realizar la búsqueda de plagio.

Moodle: Es un sistema para el Manejo del Aprendizaje en línea gratuito, que permite a los educadores la creación de sus propios sitios web privados, llenos de cursos dinámicos que extienden el aprendizaje, en cualquier momento, en cualquier sitio. Ya sea usted un profesor, estudiante o administrador, Moodle puede cumplir sus necesidades. Su núcleo es extremadamente personalizable, viene con muchas características estándares (16).

Posee una característica que debe ser habilitada por un administrador para poder utilizar los siguientes plugins para la prevención de plagio:

Crot: Herramienta gratuita de código abierto. (Solamente trabaja con MySQL). Está dirigido a la detección de plagios digitales: asignaciones copiadas de sus colegas de la misma institución, así como las asignaciones copiadas de Internet. Este bloque se basa en tecnologías de documento de huellas digitales para la comparación de pares de documentos y el motor de búsqueda de MSN (abreviación de The MicroSoft Network), que es una colección de servicios de internet ofrecidos por Microsoft para llevar a cabo la búsqueda global. Debe llevarse a cabo un pago por suscripción (17).

Turnitin: Es un sistema de detección de plagio comercial que requiere una suscripción de pago para usar en la actualidad hay 3 opciones disponibles para utilizar Turnitin en Moodle 2.0 (18).

La integración Moodle direct es un módulo independiente de Moodle que es similar al tipo de asignación avanzada, sin embargo, añade una gama de funcionalidad específica Turnitin. La versión actual está disponible en módulos y plugins. El objetivo fundamental del desarrollo es para representar todas las funciones de Turnitin dentro del flujo de trabajo Moodle. El plugin direct funciona según el principio de asignar una asignación de Turnitin en Moodle, una vez asignada todas las presentaciones realizadas en cualquiera de los lados, están sincronizados para que refleje los cambios en ambas plataformas. Por lo tanto, es posible acceder a las asignaciones creadas usando el plugin accediendo a uno u otro sistema. Existe la opción de desactivar los correos electrónicos enviados a los

estudiantes lo que significa datos de acceso y mensajes de bienvenida para que no se envíen, lo que anima a los estudiantes a presentarse sólo a través de Moodle (18).

1.3 Plataforma educativa (Mooc).

Plataformas MOOC encontradas en el mercado:

Aprendo: Solución de software libre para la creación y gestión de cursos masivos en línea (MOOC), cuyo software y comunidad de desarrollo son accesibles en el sitio OpenMOOC, impulsado por la Universidad Nacional de Educación a Distancia de España (UNED), Telefónica, Universia y el CSEV (Centro Superior para la Enseñanza Virtual). Entre las funcionalidades de la plataforma se destacan las actividades de autoevaluación y herramientas que permiten valorar los progresos en el aprendizaje, y el ritmo al que se consigue (19).

Coursera: Cursos diseñados por destacados catedráticos de varias universidades de Pennsylvania, Michigan, que ya cuenta con 33 universidades asociadas, 121 cursos y más de un millón de estudiantes (19).

Lore: Se define como una comunidad de curiosos, que abarca todas las disciplinas, países y edades. Como una plataforma para el aprendizaje en el marco de un nuevo espacio para la educación, que permite explorar (19).

Canvas Network: “Aprendizaje en línea abierto, definido por ti”, es el lema de esta red abierta de cursos en línea, que proporciona a profesores, estudiantes e instituciones, el lugar y la plataforma para definir el mundo del aprendizaje en línea de una forma que tenga sentido para todos (19).

1.3.1 Plagio en la plataforma educativa (Mooc).

En los estudios realizados en el ámbito de los valores éticos y su aplicación en la educación virtual, destacan sin duda alguna los relacionados con la deshonestidad académica, en los que se han analizado situaciones relacionadas con aspectos como: el

fraude y el plagio de información, el uso inadecuado de los recursos académicos, la violación de derechos informáticos, la falta de cumplimiento de los derechos de autor, la realización de declaraciones falsas en la recopilación y presentación de información, así como la falta de cumplimiento en la normatividad académica vigente (20).

Tan solo en el aspecto del fraude y el plagio de información en Internet, se han realizado una gran cantidad de investigaciones para intentar dimensionar este tipo de conductas. Por ejemplo, en un estudio realizado en España se reporta que casi el 94% de los estudiantes entrevistados han reconocido haber ‘copiado y pegado’ algún texto localizado en Internet. En otro estudio, aplicado en Estados Unidos la Encuesta Nacional de Compromiso Estudiantil reporta que un 59% de los estudiantes que cursan programas de educación a distancia, admiten haber incurrido en algún tipo de fraude académico (con mucha frecuencia un 27% y con frecuencia un 32%). Este fenómeno no es exclusivo de los estudiantes, pues también han salido a la luz ciertos casos de profesores o investigadores que han presentado trabajos ajenos como propios, o incluso de directivos que han falsificado documentos para ostentarse con un título o grado académico que en realidad no poseen. Es decir, el asunto de la ética es algo que atañe a todos los diferentes actores inmiscuidos en el proceso de enseñanza-aprendizaje y además, es mucho más amplio que los aspectos inherentes a la deshonestidad académica (20).

Las instituciones educativas virtuales tienen frente a sí el gran reto de establecer una serie de principios y valores que normen el proceso educativo y a la vez lo encaminen hacia la aplicación de la ética en todas las actividades que se realicen, tanto por parte de los docentes como de los estudiantes (20).

1.4 Metodología de desarrollo.

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (software). Esta disciplina trasciende la actividad de programación, que es el pilar fundamental a la hora de crear una aplicación. El ingeniero de software se encarga de

toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto. Sin embargo todo este proceso no se puede llevar a cabo sin el apoyo de las metodologías de desarrollo de software, las cuales son usadas a la hora de planificar y controlar a lo largo del proceso de desarrollo de dicho software (21).

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Durante el desarrollo esta va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (22).

Variación de AUP para la UCI (AUP-UCI)

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (23).

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Todo esto tomando como apoyo el modelo CMMI-DEV v1.3, el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad. (23).

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero

modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre (23).

Tabla 1: Fases AUP-UCI

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades
Construcción		
Transición		

		requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los

proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto) (23).

Tabla 2: Disciplinas AUP-UCI

Disciplinas AUP-UCI	Objetivos de las Disciplinas AUP-UCI
Modelado de negocio	Comprender los procesos de negocio de una organización, para tener garantías de que el software desarrollado va a cumplir su propósito.
Requisitos	Desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.
Análisis y diseño	Si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos y una descripción que sea fácil de mantener, para ayudar a la estructuración del sistema (incluyendo su arquitectura).
Implementación	A partir de los resultados del Análisis y Diseño se construye el sistema.

Pruebas internas	Verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales al ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible, componentes de prueba ejecutables para automatizar las pruebas.
Pruebas de liberación	Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
Pruebas de aceptación	Es la prueba final antes del despliegue del sistema. Verifica que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.
Gestión de la configuración	Se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2.
Planeación de proyecto	
Monitoreo y control de proyecto	

Para modelar el negocio en la disciplina Modelado de negocio AUP-UCI propone las siguientes variantes (23):

- Casos de Uso del Negocio (CUN).
- Descripción de Proceso de Negocio (DPN).

- Modelo Conceptual (MC).

A partir de las variantes anteriores y teniendo en cuenta que existen 3 formas de encapsular los requisitos, se condicionan cuatro escenarios en AUP-UCI para modelar el sistema en la disciplina Requisitos, los escenarios son (23):

Escenario No 1: proyectos que modelen el negocio con CUN solo pueden modelar el sistema con Casos de Uso del Sistema (CUS).

Escenario No 2: proyectos que modelen el negocio con MC solo pueden modelar el sistema con Casos de Uso del Sistema (CUS).

Escenario No 3: proyectos que modelen el negocio con DPN solo pueden modelar el sistema con Descripción de Requisitos por Proceso (DRP).

Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

De los 4 escenarios propuestos por esta metodología, se escoge para modelar el sistema el escenario número 4, el mismo encapsula los requisitos en HU.

Analizando las características principales de esta metodología y teniendo en cuenta las ventajas y desventajas que propone, se opta por el uso de esta basándose en los siguientes puntos:

- La plataforma Zera 2.0 está diseñada según la metodología AUP-UCI por lo que se hace menos complejo a la hora de integrar el componente que se quiere desarrollar con dicha plataforma, al contar esta con una documentación desarrollada sobre la base de esta metodología.
- Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3 (23).

- Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a 1 la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11 (23).

Basándose en estos puntos se considera que AUP-UCI facilitará el desarrollo exitoso del proyecto y la correcta documentación del mismo.

1.5 Herramientas y tecnologías a utilizar.

1.5.1 Lenguaje y tecnologías del lado del cliente

HTML

HTML es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc (24). Versión a emplear HTML 5.0.

CSS

CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, etc. El lenguaje CSS se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que forman la sintaxis de las hojas de estilo. Cada regla consiste en un selector y una declaración, esta última va entre corchetes y consiste en una propiedad o atributo, y un valor separados por dos puntos (25). Versión a emplear 3.0.

JavaScript

JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos. JavaScript

contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias (26). Versión a emplear 1.8.

Bootstrap

Bootstrap es un marco frontal de extremo libre para el desarrollo web más rápido y más fácil. Incluye plantillas de diseño basadas en HTML y CSS para tipografía, formas, botones, tablas de navegación, carruseles de imágenes y muchas otras, así como complementos opcionales de JavaScript (27). Versión a emplear 3.0.

Ventajas de Bootstrap (27):

Es fácil de usar, cualquier persona con conocimientos simplemente básicos de HTML y CSS puede utilizarlo. CSS sensible de Bootstrap ajusta a los teléfonos, tabletas y ordenadores de sobremesa. Compatibilidad del navegador: Es compatible con todos los navegadores modernos (Chrome, Firefox, Internet Explorer, Safari y Opera).

JQuery

JQuery es uno de los complementos más esenciales para el desarrollo web, usado en millones de sitios en toda la web, ya que nos facilita mucho el desarrollo de aplicaciones enriquecidas del lado del cliente, en Javascript, compatibles con todos los navegadores. JQuery es un framework Javascript, por lo que es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, los frameworks son unas librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar. Como es normal, cada uno de los frameworks tiene sus ventajas e inconvenientes, pero JQuery es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración

en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework. Otra cosa muy interesante es la dilatada comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas en JQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, etc (28). Versión a emplear 3.0.

1.5.2 Lenguaje y tecnologías del lado del servidor

PHP

PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales (29). Versión a emplear 5.4.

Symfony2

Es un framework que agiliza el desarrollo de aplicaciones mediante la automatización de muchos de los patrones empleados para un fin determinado. Symfony es un marco completo diseñado para optimizar el desarrollo de aplicaciones web. Separa las reglas de negocio, la lógica del servidor, y las vistas de presentación. Contiene numerosas herramientas y clases dirigidas a acortar el tiempo de desarrollo de una aplicación web compleja. Además, automatizar tareas comunes para que el desarrollador pueda centrarse exclusivamente en los aspectos específicos de una aplicación. El resultado final

de estas ventajas significa que no hay necesidad de reinventar la rueda cada vez que una nueva aplicación web se construye. Symfony está completamente escrito en PHP. Se ha probado a fondo en varios proyectos del mundo real, y es realmente en uso para la alta demanda de sitios web de comercio electrónico. Es compatible con la mayoría de los motores de bases de datos disponibles, incluyendo MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se ejecuta en plataformas Windows y Linux (30). Versión a emplear 2.7.9.

1.5.3 Entorno de Desarrollo Integrado NetBeans

NetBeans IDE con sus editores, analizadores de código, y convertidores, puede rápidamente y sin problemas actualizar sus aplicaciones para utilizar las nuevas construcciones del lenguaje Java, como lo son operaciones funcionales y referencias de métodos. Establece el estándar para el desarrollo de tecnologías de vanguardia de la caja. Un IDE es mucho más que un editor de texto. Destaca código fuente sintácticamente y semánticamente. Le permite refactorizar fácilmente el código, con una serie de herramientas útiles y potentes, mientras que también proporciona plantillas de código, consejos de codificación y generadores de código. El editor soporta varios lenguajes de Java, C / C ++, XML y HTML, PHP, Javadoc, JavaScript y JSP. Debido a que el editor es extensible, se puede enchufar soporte para muchos otros idiomas (31). Versión a emplear 8.0.

1.5.4 Lenguajes y tecnologías de modelado.

Existen lenguajes y herramientas que se utilizan con el propósito de generar los artefactos que se llevan a cabo en el proceso de desarrollo del software para la construcción de la solución.

UML

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de

los 80's y principios de los 90s. UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos de un lenguaje de modelado y de un proceso. UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios (32).

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño. La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo (33).

Visual Paradigm

Visual Paradigm es una herramienta de software diseñado para los equipos de desarrollo de software para modelar el sistema de información de negocio y gestionar los procesos de desarrollo. Visual Paradigm soporta lenguajes de modelado de la industria y estándares clave tales como Unified Modeling Language (UML), SysML, BPMN, XMI, etc. Ofrece a las empresas de software completos set de herramientas que necesitan para la captura de los requisitos, análisis de procesos, diseño de sistemas, diseño de base de datos, y etc (34). Versión a emplear 8.0.

1.5.5 Sistema gestor de base de datos PostgreSQL

PostgreSQL ofrece muchas ventajas para su empresa o negocio con respecto a otros sistemas de bases de datos. Con PostgreSQL, nadie puede demandar por romper los acuerdos de licencia, ya que no hay coste de licencia relativa al software (35). Versión a emplear 9.4.

Esto tiene varias ventajas adicionales (35):

- Modelos de negocio más rentables con el despliegue a gran escala.
- No existe la posibilidad de ser auditado para el cumplimiento de licencias en cualquier etapa.
- Flexibilidad para realizar investigaciones concepto y pruebas de implementaciones sin necesidad de incluir los costos adicionales de licencias.
- Mejor soporte de los vendedores de software privativo.

1.5.6 Administrador de bases de datos PgAdmin

Es una herramienta multiplataforma (Sistemas Operativos Windows y Linux) para la administración de base de datos en diferentes versiones de PostgreSQL. Posee extensa documentación la cual responde a las necesidades de los usuarios. Nos permite desde hacer búsquedas SQL hasta desarrollar toda nuestra base de datos de forma muy fácil e intuitiva; directamente desde la interfaz gráfica (36). Versión a emplear 1.18.

1.5.7 Servidor Web Apache

El Proyecto Apache HTTP Server es un esfuerzo de desarrollo de software de colaboración cuyo objetivo es crear un sistema robusto, de grado comercial y de libre disposición en implementación del código fuente de un servidor HTTP (web). El proyecto es administrado conjuntamente por un grupo de voluntarios ubicados en todo el mundo, el uso de Internet y la Web para comunicarse, planificar y desarrollar el servidor y su documentación relacionada. Este proyecto forma parte de la Apache Software Foundation. Además, cientos de usuarios han contribuido con ideas, código y documentación para el proyecto. Este archivo está destinado a describir brevemente la historia del Servidor Apache HTTP y reconocer los muchos colaboradores (37).

Apache Software existe para proporcionar implementaciones robustas y de referencia de calidad comercial de muchos tipos de software. Debe seguir siendo una plataforma sobre la cual los individuos y las instituciones pueden construir sistemas fiables, tanto con fines experimentales y para fines de misión crítica (37). Versión a emplear 2.4.7.

1.6 Conclusiones parciales

Como resultado del estudio realizado en este capítulo sobre los principales conceptos relacionados con el dominio de sistemas para la detección de plagio y el plagio como problema principal. Se ha decidido implementar los algoritmos Vector Space Model, N-gramas y NCD, ya que estos algoritmos satisfacen los objetivos propuestos en el trabajo.

El desarrollo del componente se realizará siguiendo los pasos propuestos por la metodología de desarrollo AUP-UCI, debido a que es una metodología ágil que está adaptada para los proyectos de producción en la UCI.

Capítulo2: Desarrollo de la solución propuesta.

2.1 Introducción

En el presente capítulo se aborda el proceso de desarrollo del componente para la detección de plagio en los foros de la plataforma educativa Zera 2.0, principalmente en las etapas de análisis, diseño e implementación utilizando la metodología de desarrollo AUP-UCI. Se explica la creación de un modelo de dominio, que da lugar a la obtención de los requisitos funcionales y no funcionales con los cuales debe cumplir la aplicación. Se define la arquitectura y se hace referencia a los principales patrones de diseño utilizados. En fin, se documentan los elementos significativos para que se obtenga como resultado un producto con la calidad requerida por el cliente.

2.2 Propuesta del sistema

El proceso de evaluación de un estudiante en su participación en los foros de la plataforma Zera 2.0 constituye un hecho de gran importancia al ser esta la principal características con que cuenta dicho foro. Esta evaluación por lo general es un proceso que se realiza de una forma rigurosa por la tendencia a plagio existente en la docencia universitaria de nuestros días.

Con el objetivo de minimizar esta problemática se propone integrar a la plataforma Zera 2.0 un componente que proporcione la detección de plagio de tipo " *palabra por palabra* " en los foros de esta. Dicho componente consiste en un sistema que se va a ejecutar de forma automática, en el momento que un estudiante envíe su comentario, los algoritmos implementados se van a ejecutar haciendo una comparación de este comentario con los demás existentes en la Base de Datos, y luego de aplicar una técnica de aprendizaje automático como lo es el Naïve Bayes el mismo devuelve si es plagiado o no, además, de la funcionalidad que debe poseer el sistema de generar un reporte con las estadísticas de plagio encontradas.

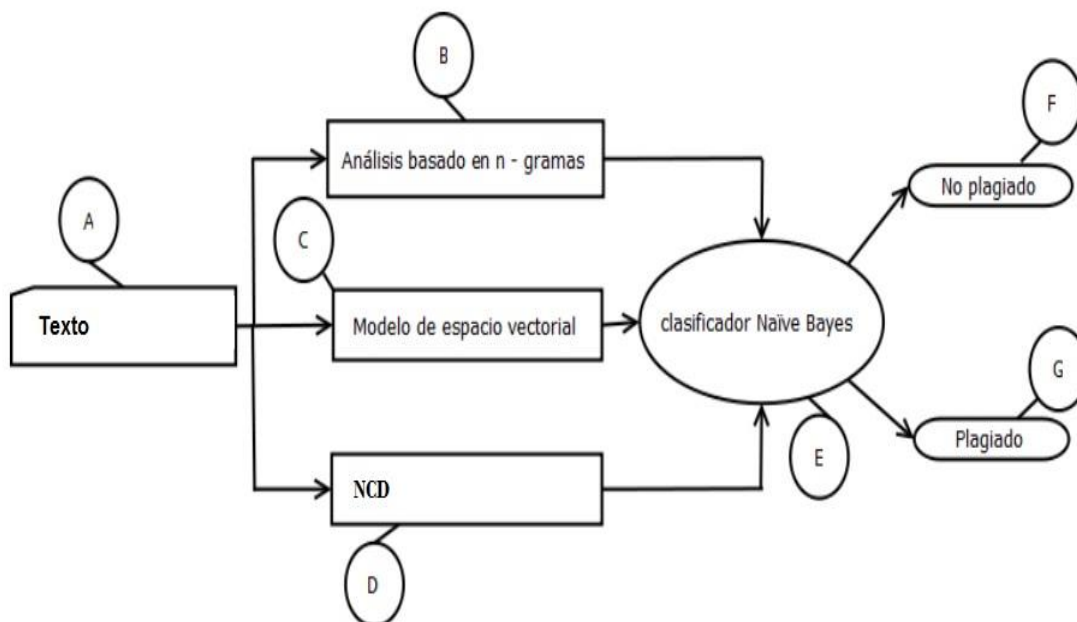


Ilustración 1: Técnica de aprendizaje automático Naïve Bayes.

A: Texto para analizar y definir el nivel de coincidencias con los ya publicados en el foro.

B, C, D: Algoritmos que procesan dicho texto en busca de similitudes con otros textos y como resultado informan un porcentaje.

E: Él tiene como entrada los valores resultantes de los algoritmos aplicados en la fase anterior. Con estos valores se ejecuta el algoritmo y su resultado definirá si el documento finalmente es plagiado o no.

F: No plagiado es uno de los tipos de clase para la clasificación. Este tipo afirma que no existe plagio.

G: Plagiado es el otro tipo de clase para la clasificación. Este tipo afirma la existencia de plagio.

2.2.1 Técnica de aprendizaje automático para la clasificación

Una vez empleados los algoritmos descritos para la detección de las coincidencias en el texto, se propone la utilización de una técnica de aprendizaje automático (rama de la

inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender). Esta técnica permitirá determinar si un texto es plagiado o no a partir de los resultados que arroje la aplicación de cada uno de los métodos anteriores. Dentro de la rama de la inteligencia artificial existen varios métodos de aprendizaje automático para la clasificación que permiten la realización de esta tarea. Luego de un estudio realizado se decidió explotar el algoritmo de clasificación Naïve Bayes o clasificador bayesiano ingenuo (38).

Naïve Bayes

Este algoritmo es ampliamente usado en procesos de clasificación. Se le considera como una forma especial, o como el modelo más simple de clasificación basado en una Red Bayesiana. Es utilizado para predecir la clase a la que pertenece una instancia determinada, suponiendo que las características de dicha instancia son independientes. Para el caso que se aplica, se toman como las “características” el resultado de cada uno de los algoritmos de detección de plagio, con el fin de clasificar un texto en las siguientes clases “Plagiado” o “No plagiado” (38).

Se tienen dos categorías para representar el resultado de la aplicación de cada uno de los algoritmos de detección de plagio (38):

- Bajo: Si como resultado del algoritmo se obtiene que el texto coincida con otros en menos de un 30%.
- Alto: Si como resultado del algoritmo se obtiene que el texto coincida con otros en más de un 30%.

Luego del resultado de cada algoritmo se comprueba si el porcentaje de similitud es alto o bajo. Después se obtienen las probabilidades por cada algoritmo separando las plagiadas y las no plagiadas, estas se multiplican por separado y según quien sea mayor se clasifica el texto en plagiado o no plagiado (38).

2.3 Modelo de dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema (39).

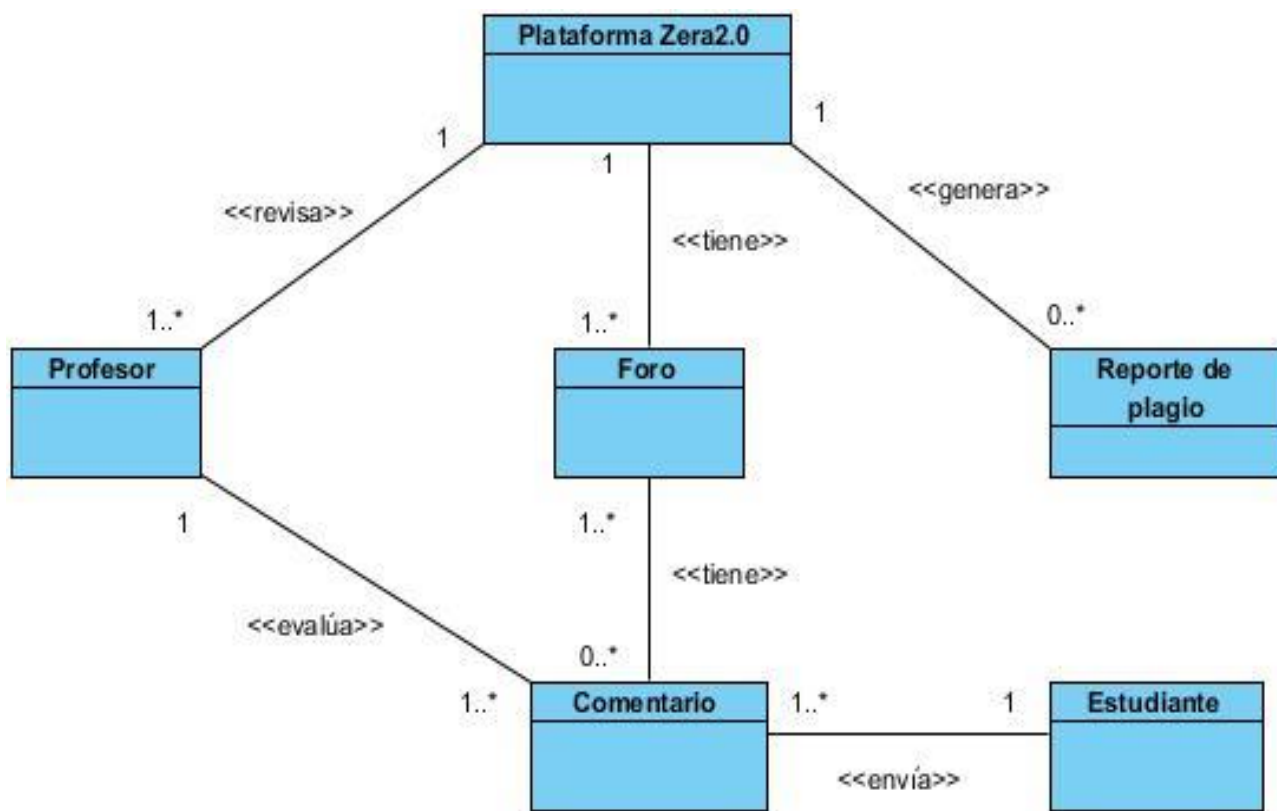


Ilustración 2: Modelo del dominio.

2.3.1 Clases conceptuales del modelo de dominio

Estudiante: Persona que envía los comentarios al foro.

Profesor: Persona que revisa el sistema después de enviado un comentario y lo evalúa para detectar plagio.

Problema: Situación a resolver mediante los algoritmos.

Comentario: Respuesta enviada por el estudiante.

Foro: Espacio virtual donde se suben los comentarios.

Plataforma Zera 2.0: Plataforma educativa donde se realiza todo este proceso

Reporte de plagio: Información que se genera tras la detección automática de plagio.

2.4 Requisitos funcionales

El componente debe permitir:

RF1: Analizar similitud del comentario enviado con otros comentarios existentes.

RF2: Eliminar el resultado de comparación de plagio y volver a analizar el comentario.

RF3: Mostrar los comentarios similares a un comentario específico.

RF4: Mostrar el resultado del análisis de cada algoritmo en un comentario.

RF5: Exportar resultado de comparación de un comentario en formato PDF.

2.5 Requisitos no funcionales

Los requerimientos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (40).

Requerimientos de Hardware.

PC Servidor

RNF1: Mínimo de 4GB de memoria RAM.

Requerimientos de usabilidad.

RNF2: El componente chequeará todos los comentarios entrantes de forma automática, con el objetivo de detectar el plagio en el momento que haya sido enviado.

RNF3: El componente debe poseer una interfaz fácil de utilizar para cualquier tipo de usuarios con conocimientos básicos de informática y manejo de ordenadores.

Apariencia o Interfaz Externa:

RNF4: El diseño de las interfaces debe ser sugerente al usuario, permitiendo mayor entendimiento y facilidad del uso del componente.

RNF5: El componente proporcionará claridad y correcta organización de la información, permitiendo la interpretación inequívoca de ésta.

2.6 Historias de Usuario

Las historias de usuario son un instrumento para el levantamiento de requerimientos para el desarrollo de un software. Su utilización es común cuando se aplican marcos de trabajo ágiles (41). A continuación se muestra la descripción de una de las HU identificadas, las demás se encuentran en los anexos.

Tabla 3: HU-Analizar similitud del comentario enviado con otros comentarios existentes.

Número: 1	Nombre del requisito: Analizar similitud del comentario enviado con otros comentarios existentes.	
Programador: Yunierkis Caraballo Mora	Iteración Asignada: 1	
Prioridad: Alta	Tiempo Estimado: 7	
Riesgo en Desarrollo: N/A	Tiempo Real: 6	
<p>Descripción:</p> <p>1- Objetivo: Permitir analizar la similitud del comentario enviado con otros comentarios existentes.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para analizar un comentario con los demás:</p> <ul style="list-style-type: none"> - Deben ejecutarse los algoritmos NCD, N-gramas y Espacio Vectorial. - Debe existir en el sistema al menos dos comentarios para realizar la comparación . <p>3- Flujo de la acción a realizar: Cuando el estudiante ejecuta la acción enviar comentario son ejecutados automáticamente los algoritmos implementados de forma tal que se guarde el análisis en la base de datos.</p>		

Observaciones:

N/A

Prototipo de interfaz:

N/A

Tabla 4: HU- Eliminar el resultado de comparación de plagio y volver a analizar el comentario.

Número: 2	Nombre del requisito: Eliminar el resultado de comparación de plagio y volver a analizar el comentario.	
Programador: Yunierkis Caraballo Mora	Iteración Asignada: 2	
Prioridad: Alta	Tiempo Estimado: 3	
Riesgo en Desarrollo: N/A	Tiempo Real: 2	
Descripción: 1-Objetivo: Permitir eliminar el resultado de comparación de plagio y volver a analizar el comentario. 2- Acciones para lograr el objetivo (precondiciones y datos): Para eliminar el resultado de comparación de plagio de los comentarios hay que: - Se debe haber realizado el análisis de cada comentario. - Debe existir en el sistema al menos 2 comentarios. - Se desee editar algún comentario. 3- Flujo de la acción a realizar: El sistema permite eliminar las estadísticas de plagio anteriormente guardadas en la base de datos, automáticamente cuando se de en la opción enviar comentario en el proceso de edición se vuelve a realizar el análisis de plagio y debe guardarse nuevamente en la base de datos.		

Observaciones: N/A
Prototipo de interfaz: N/A

Tabla 5: HU- Mostrar los comentarios similares a un comentario específico.

Número: 4	Nombre del requisito: Mostrar los comentarios similares a un comentario específico.	
Programador: Yunierkis Caraballo Mora	Iteración Asignada: 4	
Prioridad: Alta	Tiempo Estimado: 3	
Riesgo en Desarrollo: N/A	Tiempo Real: 2	
Descripción: 1- Objetivo: Permitir mostrar los comentarios similares a un comentario específico. 2- Acciones para lograr el objetivo (precondiciones y datos): Para mostrar un comentario similar hay que: <ul style="list-style-type: none"> - Se deben ejecutar los algoritmos implementados. - Debe existir al menos un comentario similar. - El comentario analizado debe haber dado plagiado. 3- Flujo de la acción a realizar: Se seleccione la opción Análisis de plagio y debe mostrarse una ventana donde sean expuestos los comentarios similares.		
Observaciones: N/A		
Prototipo de interfaz:		



Tabla 6: HU-Mostrar el resultado del análisis de cada algoritmo en un comentario.

Número: 3	Nombre del requisito: Mostrar el resultado del análisis de cada algoritmo en un comentario.	
Programador: Yulierkis Caraballo Mora	Iteración Asignada: 3	
Prioridad: Alta	Tiempo Estimado: 7 días	
Riesgo en Desarrollo: N/A	Tiempo Real: 5 días	
<p>Descripción:</p> <p>1- Objetivo: Permitir mostrar el resultado del análisis de cada algoritmo en un comentario.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para mostrar el resultado del análisis de cada algoritmo en un comentario: - Debe haberse guardado en la base de datos el análisis realizado.</p> <p>3- Flujo de la acción a realizar: Inicialmente se muestra al usuario autenticado un listado con los comentarios analizados, una vez seleccionado la opción Análisis de plagio, se podrá ver el análisis por algoritmo, además, debe decir si es plagiado o no.</p>		
Observaciones: N/A		

Prototipo de interfaz:

Análisis de plagio	
Probabilidad de plagio de cada algoritmo	
Algoritmo	Probabilidad
N-gramas	$0 \leq N \leq 1$
Espacio Vectorial	$0 \leq N \leq 1$
NCD	$0 \leq N \leq 1$

Bayes: Plagiado

Tabla 7: HU-Exportar resultado de comparación de un comentario en formato PDF.

Número: 5	Nombre del requisito: Exportar resultado de comparación de un comentario en formato PDF.	
Programador: Yunierkis Caraballo Mora	Iteración Asignada: 5	
Prioridad: Alta	Tiempo Estimado: 2 días	
Riesgo en Desarrollo: N/A	Tiempo Real: 2 días	
Descripción:		
1-Objetivo: Permitir exportar resultado de comparación de un comentario en formato PDF.		
2- Acciones para lograr el objetivo (precondiciones y datos):		

Para exportar resultado de comparación de un comentario:

- Debe haberse guardado en la base de datos el análisis realizado.

3- Flujo de la acción a realizar:

Se selecciona la opción Análisis de plagio del listado de comentarios y luego de salir la ventana con el análisis se seleccione la opción Exportar PDF, se muestra en el mismo el análisis de similitud por algoritmos ya realizado así como los comentarios similares en caso de que este haya sido plagiado.

Observaciones: N/A

Prototipo de interfaz:

Análisis de plagio

Probabilidad de plagio de cada algoritmo

Algoritmo	Probabilidad
N-gramas	$0 \leq N \leq 1$
Espacio Vectorial	$0 \leq N \leq 1$
NCD	$0 \leq N \leq 1$

Exportar a PDF

Bayes: Plagiado

Comentario:

Comentario similar	Fecha de inserción	Autor
--------------------	--------------------	-------

Aceptar

2.7 Actores del Sistema

Los actores son las distintas personas (o dispositivos) que usan el sistema o producto en el contexto de la función y comportamiento que va a describirse. Los actores representan los papeles que desempeñan las personas cuando opera el sistema. Con una definición más formal, un actor es cualquier cosa que se comunique con el sistema o producto y que sea externo a éste. Todo actor tiene uno o más objetivos cuando utiliza el sistema (42).

Tabla 8: Descripción de los actores del sistema.

Actor	Descripción
Profesor	Actor externo al sistema que interactúa directamente con este, en busca de posibles plagios en las respuestas enviadas por los estudiantes y revisión de los reportes generados automáticamente.
Estudiante	Encargado de subir los comentarios al foro.

2.8 Modelo de Diseño

El diseño crea una representación o modelo del software, proporciona detalles sobre arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema. El diseño es el lugar en el que se establece la calidad del software (42).

2.8.1 Diagramas de clases del diseño

Un Diagrama de Clases de Diseño muestra la especificación para las clases de una aplicación. Incluye la siguiente información (42):

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Navegabilidad.
- Dependencias.

A diferencia del Modelo Conceptual, un Diagrama de Clases de Diseño muestra definiciones de entidades software más que conceptos del mundo real.

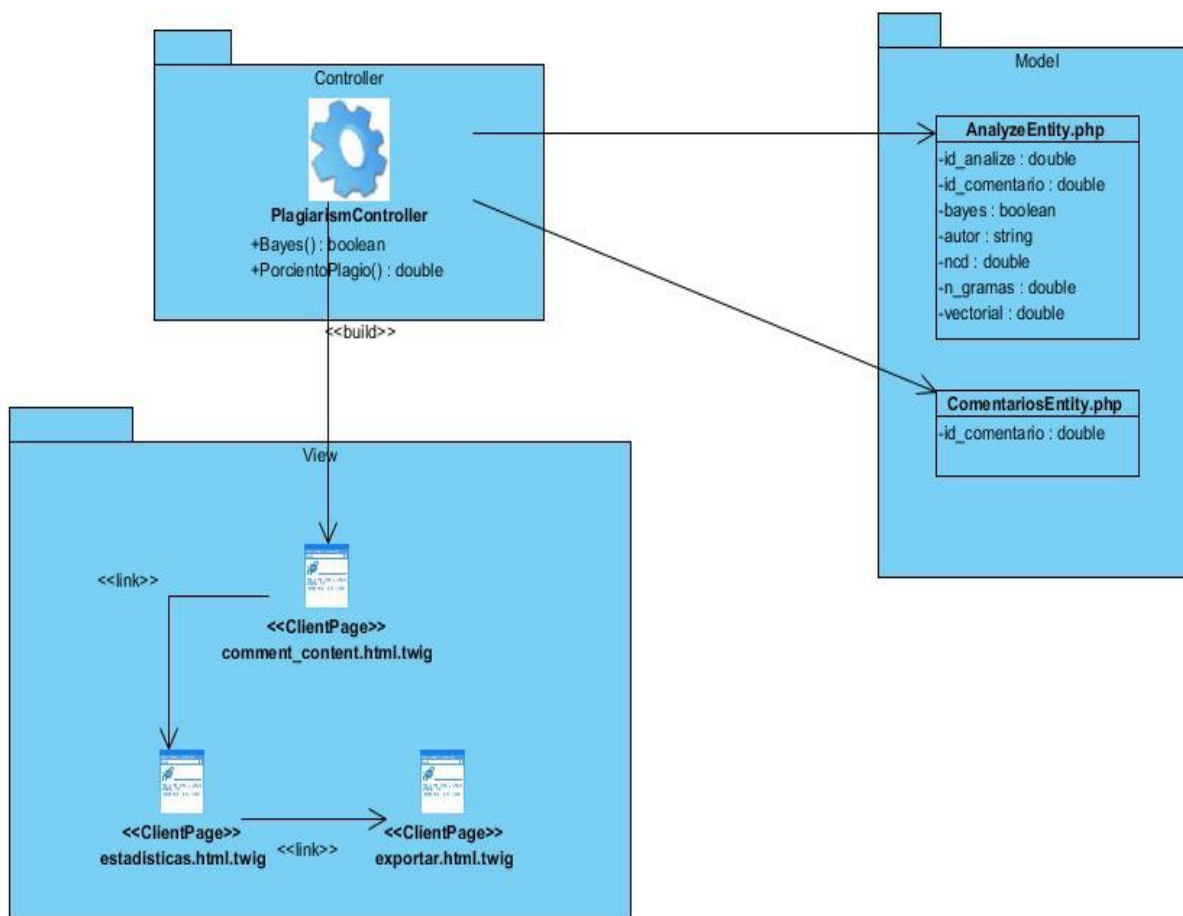


Ilustración 3: Diagrama de clase del diseño

2.8.2 Diagramas de secuencia del diseño

Un Diagrama de Secuencia describe comportamiento de sistemas, subsistemas y operaciones, representando los objetos que intervienen, así como los mensajes que intercambian, ordenados en el tiempo a través de la línea de vida de cada uno de los objetos y sus cajas de activación (42).

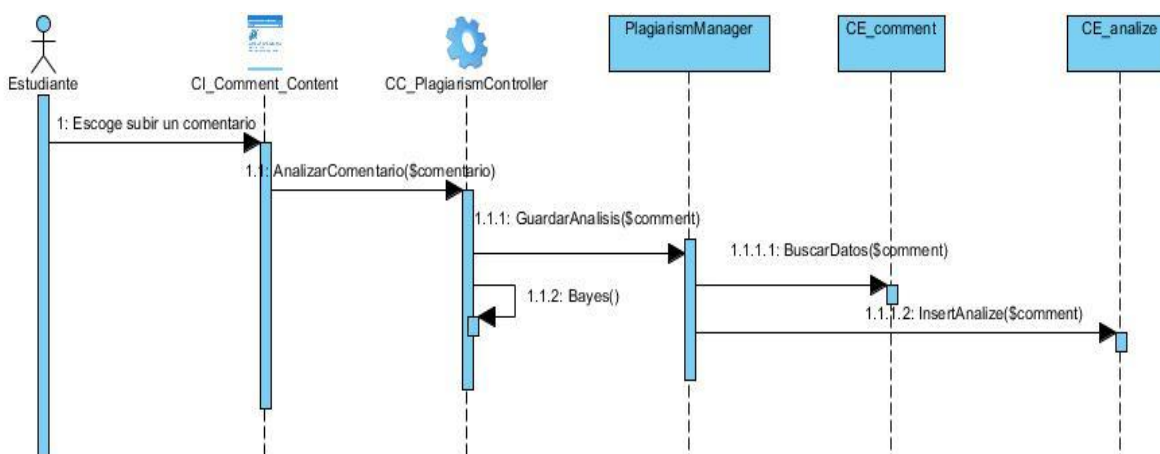


Ilustración 4: DS- Analizar similitud del comentario enviado con otros comentarios existentes.

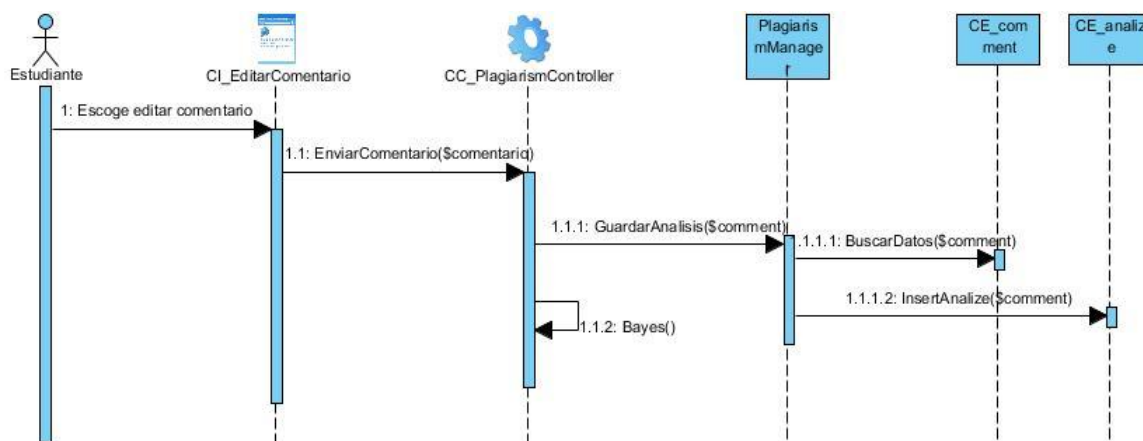


Ilustración 5: DS-Eliminar el resultado de comparación de plagio y volver a analizar el comentario.

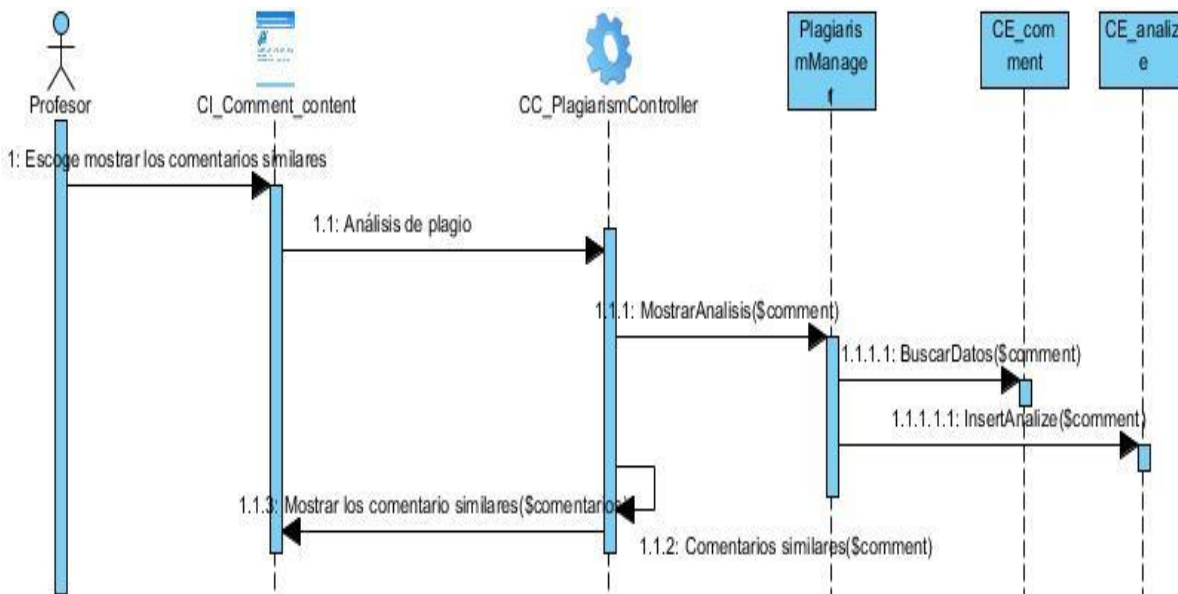


Ilustración 6: DS-Mostrar los comentarios similares a un comentario específico.

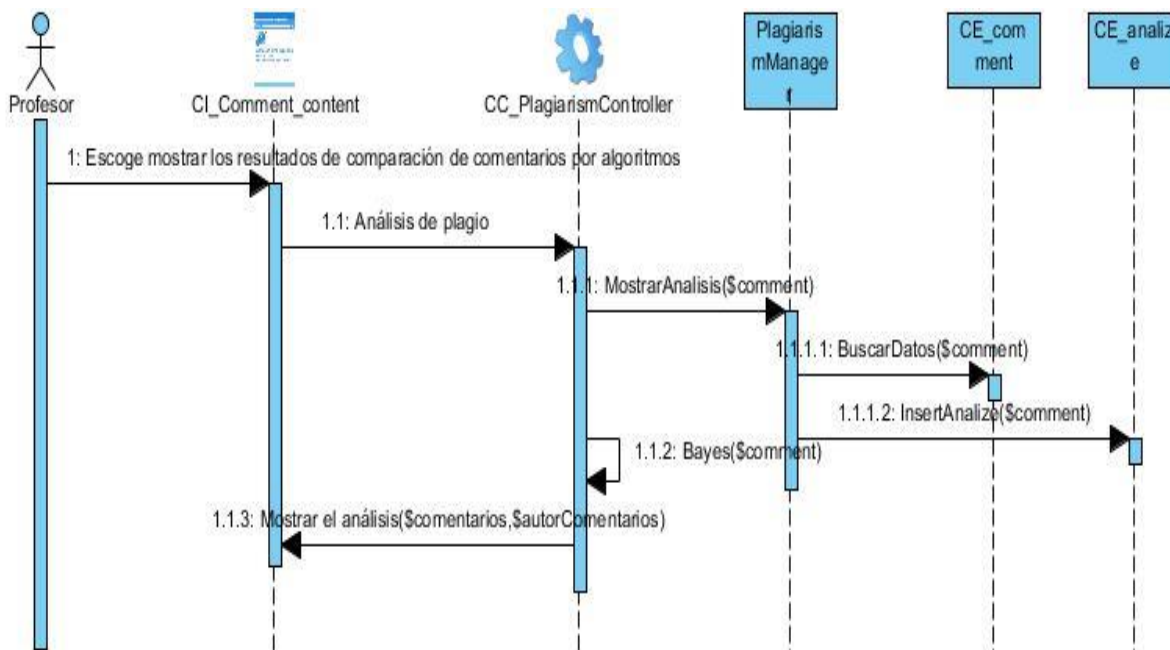


Ilustración 7: DS-Mostrar el resultado del análisis de cada algoritmo en un comentario.

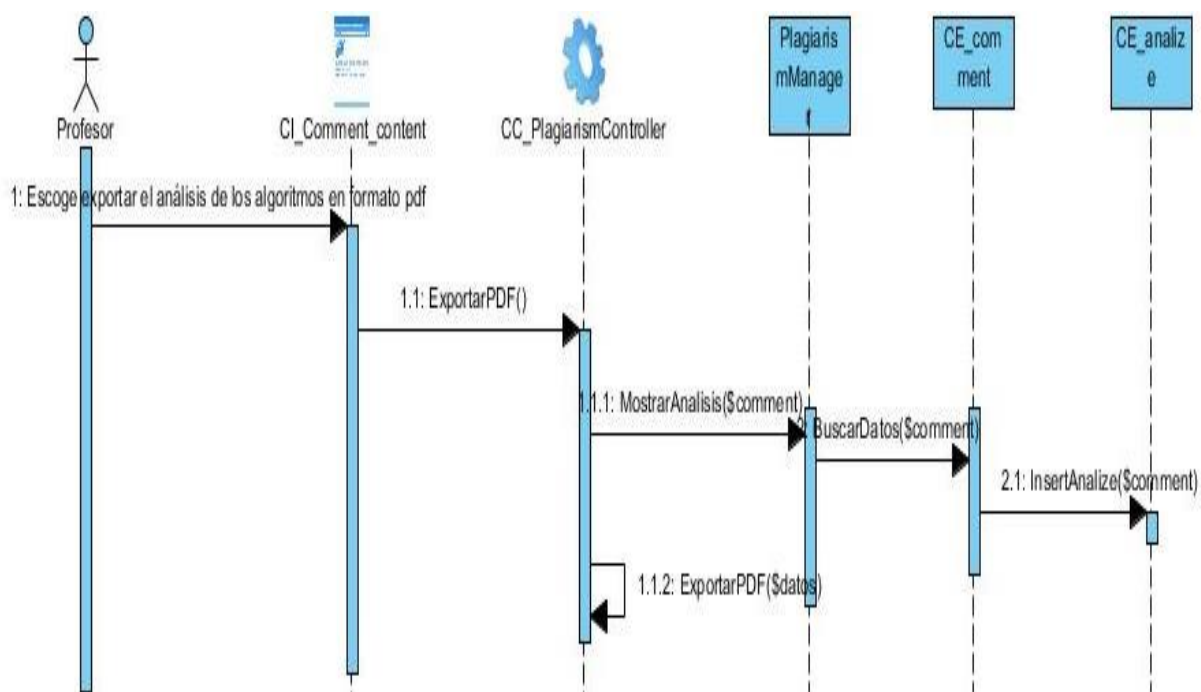


Ilustración 8: DS-Exportar resultado de comparación de un comentario en formato PDF.

2.8.3 Modelo de datos

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos o podríamos decir que es un conjunto de conceptos que permiten describir, a distintos niveles de abstracción, la estructura de una base de datos (43).

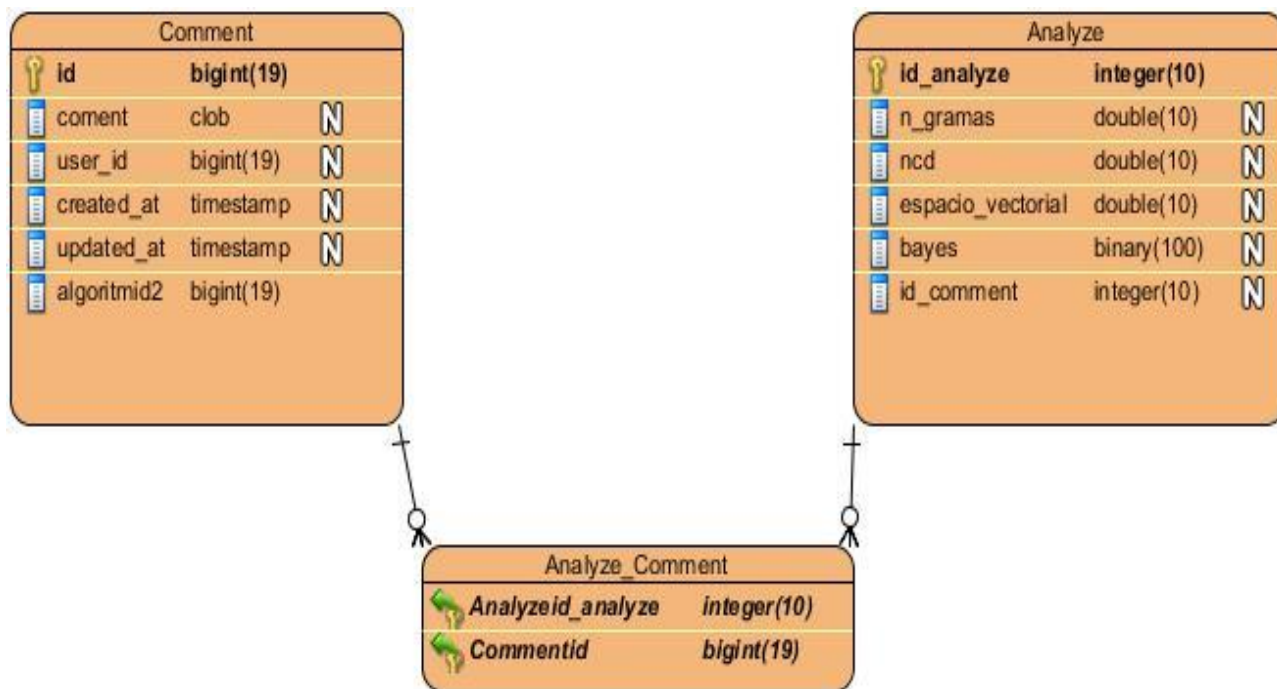


Ilustración 9: Modelo entidad-relación.

2.9 Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación (44).

Aunque nuestra aplicación sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas etc. En lugar de reinventar la rueda, podemos solucionar problemas utilizando algún patrón, ya que son soluciones probadas y documentadas por multitud de programadores (44).

En el diseño de la aplicación se hace uso de los patrones **GRASP**, los cuales son guías o principios que sirven para asignar responsabilidades a las clases (45).

Experto: Este patrón se pone en práctica para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica. El uso de este patrón permitirá a los objetos valerse de su

propia información para hacer lo que se les pide, favorece la existencia de mínimas relaciones entre las clases, lo que permite contar con un sistema robusto y fácil de mantener.

Controlador: Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación para aumentar la reutilización de código y a la vez tener un mayor control. Este patrón se evidencia en todas las clases controladoras.

Creador: El patrón se evidencia en las clases controladoras que son las encargadas de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento.

Alta Cohesión: El patrón se evidencia debido a que a cada una de las clases se le asignaron responsabilidades de tal forma, que estén estrechamente relacionadas entre sí y que puedan trabajar en una misma parte de la aplicación, sin que tenga mucha complejidad y no realicen un trabajo excesivo.

Bajo Acoplamiento: Se proporciona un bajo acoplamiento en el diseño debido a que las clases existentes tienen asignadas responsabilidades de tal forma que estas no dependan en gran medida de otras, permitiendo así tener sistemas más robustos y de fácil mantenimiento.

2.10 Patrón arquitectónico

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles: El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. La vista transforma el modelo en una página web que permite al usuario interactuar con ella. El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista (46).

Symfony utiliza lo mejor de la arquitectura MVC y la realiza de modo que el desarrollo de aplicaciones sea rápido y sencillo. En el controlador se encuentran las acciones (Actions), las cuales son el núcleo de la aplicación, pues contienen toda la lógica de la aplicación. Estas acciones utilizan el modelo que a su vez utiliza una capa de abstracción de base de datos en este caso el ORM Doctrine y precisan las variables para la vista, la cual tiene las plantillas necesarias para mostrar la información al usuario y como mínimo una plantilla global (layout) que encapsula el código HTML común de las demás plantillas (46).

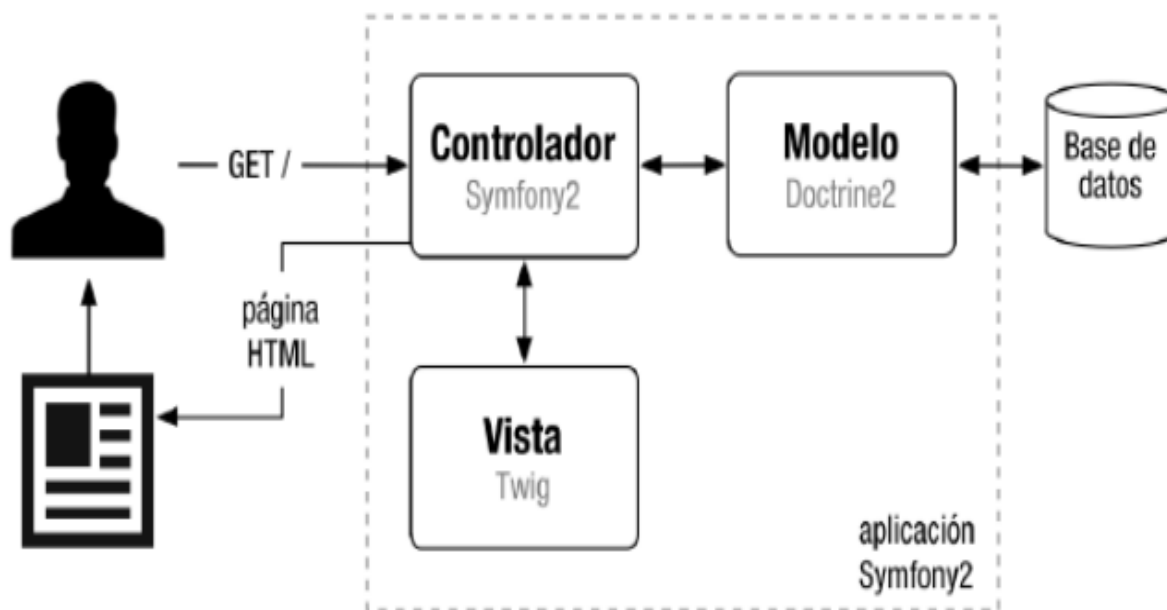


Ilustración 10: Modelo Vista Controlador

2.11 Conclusiones parciales

Tras realizar un estudio de los principales conceptos de Dominio se logra desarrollar un análisis crítico de los procesos que conforman el componente de detección de plagio. Se presenta un listado de RF y RNF, donde se recogen las principales funcionalidades del componente a diseñar. Las descripciones de las historias de usuario, diagramas de análisis y diseño construidos servirán de apoyo en el desarrollo de las funcionalidades a implementar para dar respuesta al problema planteado.

Capítulo 3: Implementación y prueba.

3.1 Introducción

El presente capítulo centra su objetivo en describir los elementos necesarios para la implementación, partiendo de los resultados obtenidos en las etapas de análisis y diseño. Para que el desarrollo del componente de software sea de forma exitosa tiene que estar acompañado de diferentes pruebas que demuestren que lo que se implementa está correcto y que esté acorde con lo que quiere el cliente.

3.2 Modelo de Implementación

La implementación empieza con el resultado del diseño, se lleva a cabo después la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Afortunadamente, la mayor parte de la arquitectura del sistema es capturada durante el diseño, siendo el propósito principal de la implementación el desarrollar la arquitectura y el sistema como un todo (47).

De forma más específica, los propósitos de la implementación son (47):

- Planificar las integraciones de sistema necesarias en cada iteración. Se sigue para ello un enfoque incremental, lo que da lugar a un sistema que se implementa en una sucesión de pasos pequeños y manejables.
- Distribuir el sistema asignado a componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa fundamentalmente en las clases activas encontradas durante el diseño.
- Implementar las clases y subsistemas encontrados en el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.

El **modelo de implementación** describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. Describe también cómo se organizan los componentes de acuerdo a los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros (47).

3.2.1 Diagrama de despliegue

Los Diagramas de Despliegue Representan la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Muestran la vista de despliegue estática de una arquitectura y se relacionan con los componentes ya que, por lo común, los nodos contienen uno o más componentes (48).

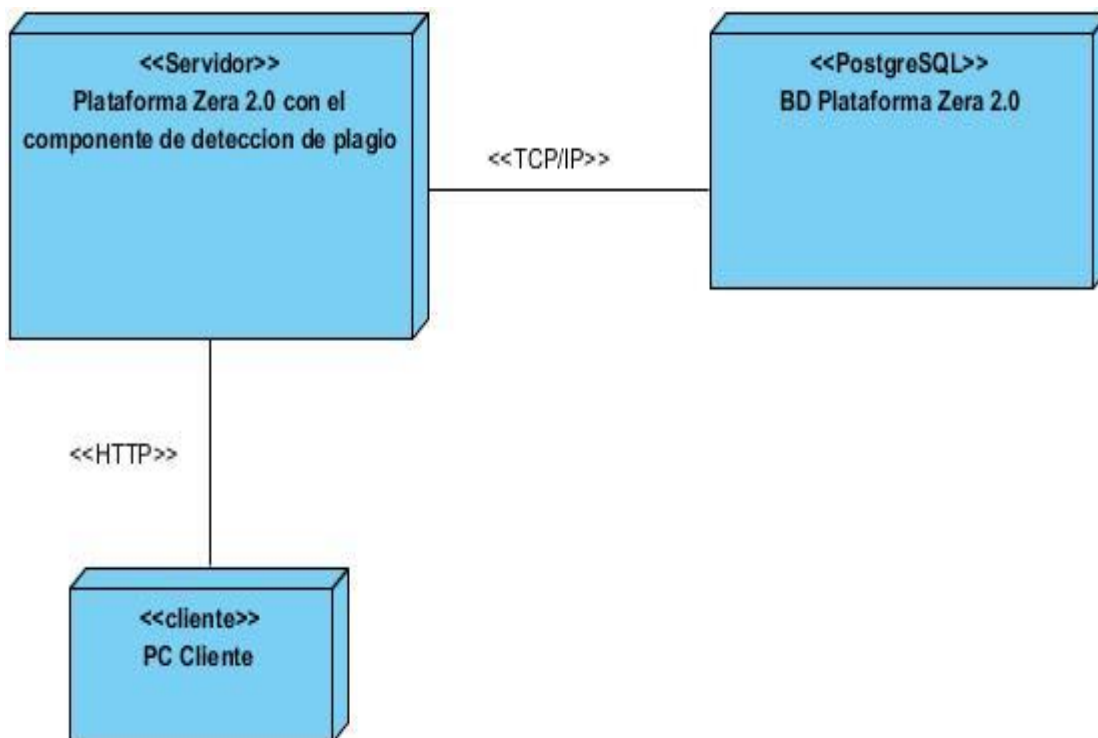


Ilustración 11: Diagrama de despliegue.

3.2.2 Diagrama de componentes

Muestra la organización y las dependencias entre un conjunto de componentes. Cubren la vista de la implementación estática y se relacionan con los diagramas de clases ya que en un componente suele tener una o más clases, interfaces o colaboraciones (48).

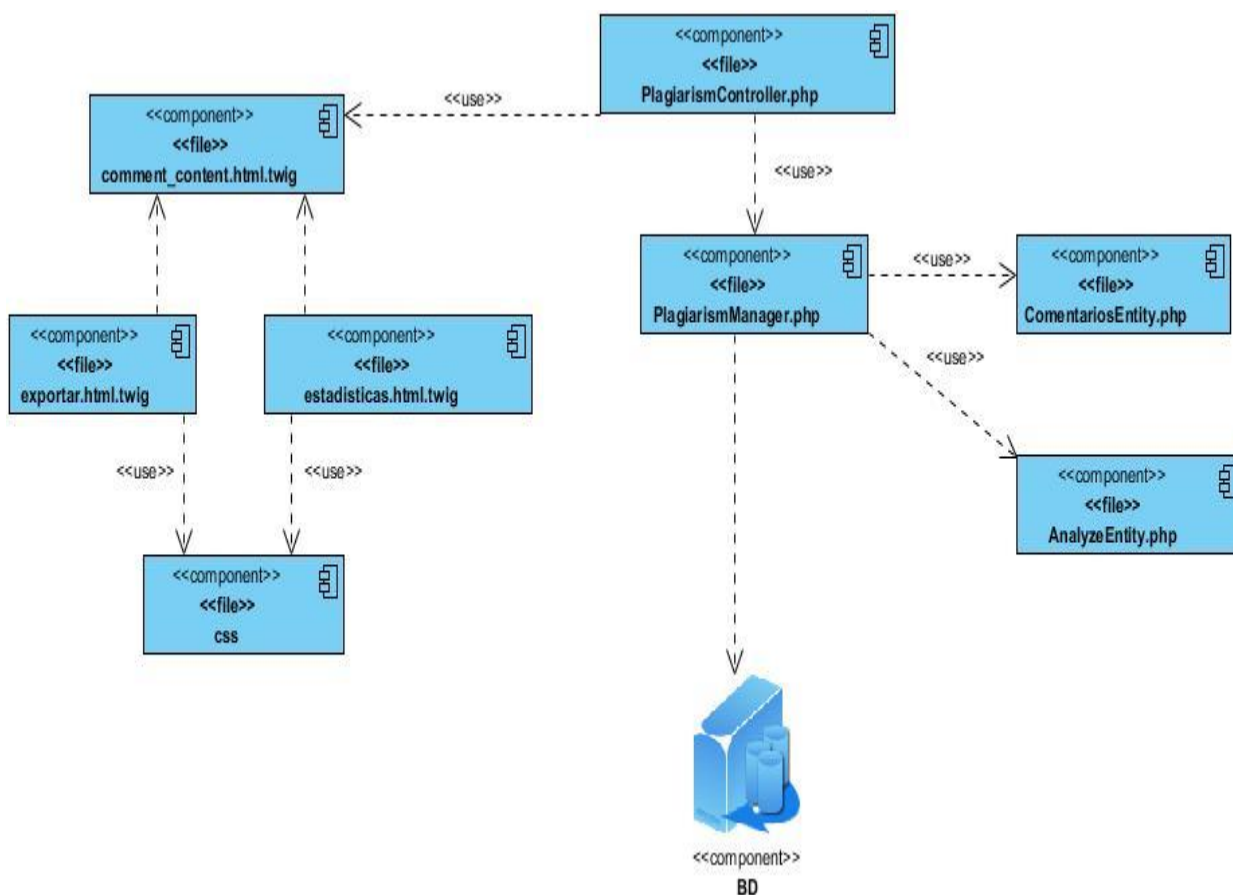


Ilustración 12: Diagrama de componentes.

3.3 Pruebas de software

El software se prueba para descubrir errores que se cometieron de manera inadvertida conforme se diseñó y construyó, por lo que es de vital importancia el desarrollo de las pruebas de software, este es un elemento de un tema más amplio que usualmente se conoce como verificación y validación. La verificación se refiere al conjunto de tareas que

garantizan que el software implementa correctamente una función específica. La validación es un conjunto diferente de tareas que aseguran que el software que se construye sigue los requerimientos del cliente. El objetivo de las pruebas del software es descubrir errores (42).

3.4 Nivel de prueba

Pruebas de sistema: Es una serie de diferentes pruebas cuyo propósito principal es ejercitar por completo el sistema basado en computadora. Aunque cada prueba tenga un propósito diferente, todo él funciona para verificar que los elementos del sistema se hayan integrado de manera adecuada y que se realicen las funciones asignadas (42).

Pruebas unitarias: En este nivel se verifican por separados las piezas de software en un funcionamiento aislado, estas piezas pueden ser módulos individuales, subprogramas y componentes. Pueden llevarse a cabo con herramientas de depuración, acceso al código fuente y pueden participar en esta de forma opcional los programadores (49).

Las pruebas de unidad no se realizan mediante el uso de una herramienta informática pues solo se efectuaron revisiones del código por parte de los desarrolladores. Son las pruebas de sistema las que permitirán comprobar el cumplimiento de los requisitos funcionales y no funcionales por estar enfocadas a verificar cómo trabaja el software una vez concluido su desarrollo.

3.4.1 Tipo de prueba: Prueba funcional

Su objetivo es asegurar el cumplimiento de los requisitos funcionales, entrada de datos, procesamiento y obtención de resultados. La principal intención de estas pruebas es medir la correspondencia entre la arquitectura de información propuesta y las funciones que realmente fueron implementadas (42).

Para realizar este tipo de pruebas es necesario tener definidos los requerimientos a verificar con los casos de prueba afiliados a cada uno de ellos. Estos serán los encargados de verificar la aplicación implementada, se utilizará el método de caja negra.

3.4.2 Método de Caja Negra

Este método se refiere a las pruebas que se enfocan en los requerimientos funcionales del software; es decir, permiten derivar conjuntos de condiciones de entrada que revisarían por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca. Estas pruebas permiten encontrar errores en las siguientes categorías (42):

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a base de datos externas.
- Errores de comportamiento o rendimiento.
- Errores de inicialización o determinación.

3.5 Diseño de casos de prueba.

Dentro del método de caja negra se utiliza la técnica “Partición Equivalente” siendo considerada como una de las más efectivas en la evaluación de los valores válidos, inválidos y los que no es necesario proporcionar un valor del dato en las entradas existentes en la aplicación. La partición de equivalencia divide el dominio de entrada de un programa en clase de datos de los que pueden derivarse casos de prueba (42).

El diseño de casos de prueba para la partición de equivalencia se basa en una evaluación de clases de equivalencia (por sus siglas CE) para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Por lo general una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana. Estas clases pueden definirse de acuerdo a los siguientes lineamientos (42):

1. Si una condición de entrada especifica un rango, se define una CE válida y dos inválidas.
2. Si una condición de entrada requiere un valor específico, se define una CE válida y dos inválidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se define una CE válida y una inválida.
4. Si una condición de entrada es booleana, se define una clase válida y una inválida.

A continuación se muestra la descripción de un caso de prueba, los demás se recogen en los epígrafes de los anexos:

Tabla 9: CP-Eliminar el resultado de análisis de plagio y volver a analizar el comentario.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1 Opción Editar comentario	Se selecciona la opción Editar comentario	Permite: -Enviar comentario. -Cancelar	Foro/Tema/ Listado comentarios / Editar comentario
EC 1.1 Opción Enviar comentario	Selecciona la opción Enviar comentario y automáticamente se borran las estadísticas de plagio del anterior comentario existentes en la base de datos.	Elimina los resultados. Regresa al listado de comentarios actualizado.	Foro/Tema/ Listado comentarios / Editar comentario / Enviar comentario / Listado comentarios/
EC 1.2 Opción Cancelar	Selecciona la opción cancelar.	Regresa a la vista anterior	Foro/ Tema/ Listado comentarios / Editar comentario / Cancelar/ Listado comentarios/

Tabla 10: CP-Analizar similitud del comentario enviado con otros comentarios existentes.

Escenario	Descripción	Respuesta del sistema	Flujo Central
-----------	-------------	-----------------------	---------------

EC 1 Opción enviar comentario	Se selecciona la Opción enviar comentario y los algoritmos implementados se ejecutan automáticamente para realizar este.	Almacena en la base de datos todo este análisis.	Foro/Tema/ Listado comentarios/ Enviar Comentario
--	--	--	---

Tabla 11: CP-Mostrar los comentarios similares a un comentario específico.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1 Opción Análisis de plagio	Se selecciona la opción Análisis de plagio.	Muestra los comentarios similares a ese comentario.	Foro/Tema/ Listado comentarios / Análisis de plagio /

Tabla 12: CP-Mostrar el resultado del análisis de cada algoritmo en un comentario.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1 Opción Análisis de plagio	Se selecciona la opción Análisis de plagio.	Muestra el análisis de plagio de un comentario por cada algoritmo utilizado.	Foro/Tema/ Listado comentarios / Análisis de plagio

Tabla 13: CP-Exportar resultado de comparación de un comentario en formato PDF.

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC 1 Opción Exportar a PDF.	Se selecciona la opción Exportar a PDF.	Exporta el análisis de plagio de un comentario por cada algoritmo utilizado en formato PDF.	Foro/ Tema/ Listado comentarios / Análisis de plagio / Exportar a PDF

3.6 Resultados de las pruebas

Con el fin de detectar la mayor cantidad de no conformidades posibles en las funcionalidades se realizaron dos iteraciones de prueba. A continuación se muestran las tablas que contienen las no conformidades identificadas por cada requisito funcional en cada iteración. Estas no conformidades se clasifican en Significativa y No significativa y su estado puede ser Resuelta o Pendiente.

Tabla 14: No conformidades encontradas en la Iteración 1.

Iteración 1				
	Requisito funcional	Descripción	Clasificación	Estado
1	Analizar similitud del comentario enviado con otros comentarios existentes.	No guarda los datos en la base de datos.	Significativa	Resuelta
2	Eliminar el resultado de comparación de plagio y volver a analizar el comentario.	No sobrescribe los datos anteriormente guardados en la base de datos.	Significativa	Pendiente
3	Mostrar los comentarios similares a un comentario específico.	Muestra una tabla vacía en caso de que sea No plagiado el comentario.	No significativa	Pendiente

Tabla 15: No conformidades encontradas en la Iteración 2.

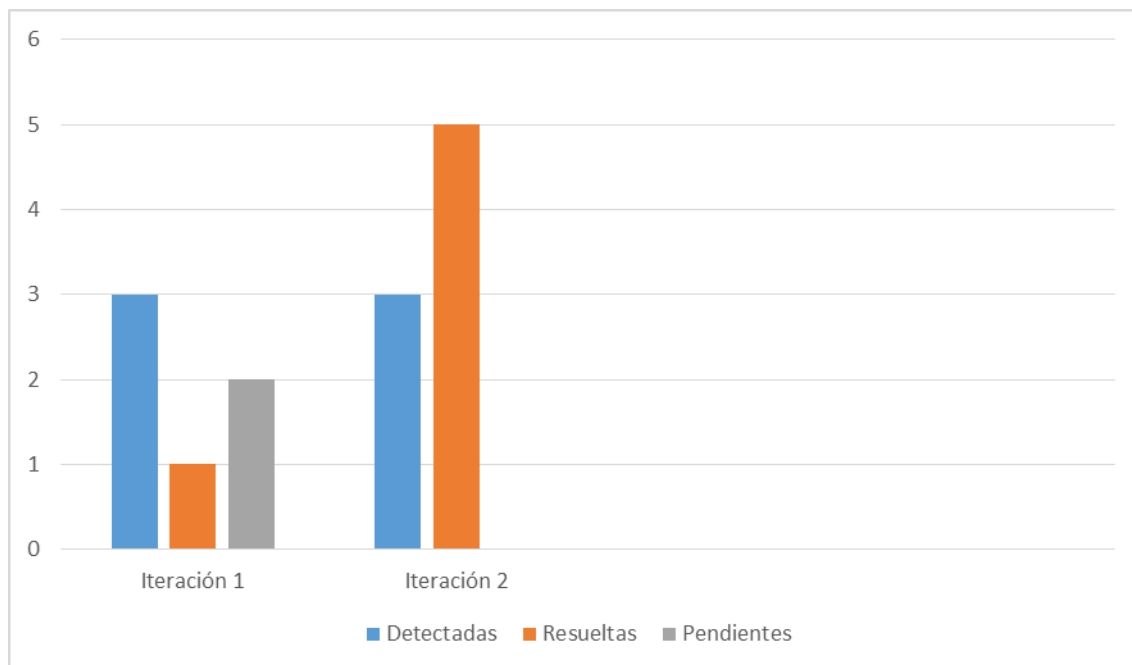
Iteración 2				
	Requisito funcional	Descripción	Clasificación	Estado
1	Mostrar el resultado del análisis de cada algoritmo en un comentario.	No se muestra el análisis hecho por el algoritmo N-gramas.	Significativa	Resuelta
2		Se muestra el análisis por algoritmo en formato de porciento.	No significativa	Resuelta
3	Exportar resultado de comparación de un comentario en formato PDF.	No se exportan los comentarios similares.	Significativa	Resuelta
Pendientes de la Iteración Anterior				
4	Eliminar el resultado de comparación de plagio y volver a analizar el comentario.	No sobrescribe los datos anteriormente guardados en la base de datos.	Significativa	Resuelta
5	Mostrar los comentarios similares a un comentario específico.	Muestra una tabla vacía en caso de que sea No plagiado el comentario.	No significativa	Resuelta

Para obtener la cantidad total de no conformidades en cada iteración y su clasificación se realiza la siguiente tabla:

Tabla 16: Cantidad total de no conformidades en cada iteración.

Requisitos funcionales	Iteración 1			Iteración 2		
	NC	S	NS	NC	S	NS
Analizar similitud del comentario enviado con otros comentarios existentes.	1	1	-	-	-	-
Eliminar el resultado de comparación de plagio y volver a analizar el comentario.	1	1	-	1	1	-
Mostrar los comentarios similares a un comentario específico.	1	-	1	1	-	1
Mostrar el resultado del análisis de cada algoritmo en un comentario.	-	-	-	2	1	1
Exportar resultado de comparación de un comentario en formato PDF.	-	-	-	1	1	-
TOTAL	3	2	1	5	3	2

A continuación se muestra un gráfico donde se puntualiza por iteraciones el total de no conformidades identificadas, el total de no conformidades resueltas y la cantidad de no conformidades pendientes.



3.7 Conclusiones parciales

En este capítulo se modelaron los diagramas correspondientes al componente de detección de plagio de la plataforma Zera 2.0: el Diagrama de Despliegue el cual representa la distribución física por nodos de la aplicación y el Diagrama de Componentes que refleja la dependencia de los mismos con la base de datos. Además, las pruebas realizadas validaron que las funcionalidades desarrolladas satisfacen los requisitos especificados, dejando listo el componente con vista a su utilización.

Conclusiones Generales

Con el desarrollo de la presente investigación se ha dado respuesta a los objetivos específicos propuestos, llegando a las siguientes conclusiones:

- Durante el desarrollo de este trabajo se ha realizado una amplia investigación del estado del arte de las diferentes herramientas y algoritmos para la detección de plagio en textos, existentes en la actualidad, seleccionando a partir de éstos los algoritmos que conforman dicho componente.
- Con la identificación de las principales clases y funcionalidades a implementar se logró un mejor entendimiento del componente a desarrollar.
- La confección de los principales artefactos dentro de la etapa diseño, correspondientes a los flujos de trabajo de la metodología utilizada sirvieron de apoyo en la implementación de las funcionalidades definidas en la investigación.
- Con la implementación de los requisitos definidos dio como resultado un componente capaz de detectar plagio en los comentarios enviados por los estudiantes en los foros de la plataforma educativa Zera 2.0.
- Los métodos de validación definidos permitieron identificar y erradicar los errores del componente, permitiendo comprobar el correcto funcionamiento del mismo, demostrándose el cumplimiento de los objetivos trazados en la investigación.

Recomendaciones

Una vez concluido el presente trabajo se recomienda:

- Para hacer más fiable la clasificación de los resultados implementar los algoritmos Dot Plot y Karp Rabin - Greedy String Tiling que soportan reordenamiento de las palabras.
- Implementar funcionalidades que permitan extraer la información de documentos de distintos formatos, para después realizar la comparación de textos haciendo uso de los algoritmos ya implementados.
- Generalizar la detección de plagio en la Plataforma Educativa Zera 2.0, para también detectar plagio en otros tipos de actividades que respondan los estudiantes.

REFERENCIAS BIBLIOGRÁFICAS

1. Educación Médica Superior - Impacto de las tecnologías de la información y las comunicaciones en la educación y nuevos paradigmas del enfoque educativo. [online]. [Accessed 4 February 2016]. Available from: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412011000100009
2. Definición de e-Learning. [online]. [Accessed 4 February 2016]. Available from: <http://www.e-abclearning.com/definicion-e-learning>
3. ¿Qué es una plataforma de e-Learning? [online]. [Accessed 4 February 2016]. Available from: <http://www.e-abclearning.com/queesunaplataformadeelearning>
4. ¿Qué es una plataforma de e-Learning? [online]. 4 February 2016. [Accessed 4 February 2016]. Available from: <http://www.e-abclearning.com/queesunaplataformadeelearning>
5. Las TIC y su utilización en la educación : 37 Plataformas virtuales educativas gratuitas. [online]. [Accessed 4 February 2016]. Available from: <http://tics-ti.blogspot.com/2014/05/30-plataformas-virtuales-educativas.html>
6. ¿Qué es un curso MOOC? - UAB Barcelona. [online]. [Accessed 4 February 2016]. Available from: <http://www.uab.cat/web/estudiar/mooc/-que-es-un-curso-mooc-1345668281247.html>
7. Foro. [online]. [Accessed 4 February 2016]. Available from: <http://es.slideshare.net/michelledominguez5095/foro-32823969>
8. 1. ¿Qué es el plagio? |. [online]. [Accessed 4 February 2016]. Available from: http://www3.uah.es/bibliotecaformacion/BECO/plagio/1_qu_es_el_plagio.html

9. ARMANDO SOTO RODRÍGUEZ. *El plagio y su impacto a nivel académico y profesional* [online]. Revista electrónica semestral, ISSN-1659-4142. Available from: http://www.ugr.es/~plagio_hum/Documentacion/06Publicaciones/ART003.pdf
10. LUIS ALBERTO BARRÓN CEDEÑO. *Detección automática de plagio en texto* [online]. November 2008. Available from: <https://riunet.upv.es/bitstream/handle/10251/12186/tesisMaster.pdf?sequence=1>
11. VICTORIA ELIZALDE. *Estudio y desarrollo de nuevos algoritmos de detección de plagio* [online]. June 2011. Available from: <http://www.dc.uba.ar/inv/tesis/licenciatura/2011/elizalde>
12. ALBERTO BARRÓN-CEDEÑO, MARTA VILA and PAOLO ROSSO. *Detección automática de plagio: de la copia exacta a la paráfrasis* [online]. Available from: http://users.dsic.upv.es/~proso/resources/BarronEtAl_JLF10.pdf
13. GRETHELL CASTILLO REYES, YANISLEY GONZÁLEZ GONZÁLEZ and GUILLERMO LUZUA FARIAS. *Técnica de clasificación bayesiana para identificar posible plagio en información textual* [online]. December 2014. Revista Cubana de Ciencias Informáticas. Available from: [http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path\[\]=759&path\[\]=300](http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path[]=759&path[]=300)
14. Detector de plagio. Detección de plagio preciso con reportes en tiempo real. [online]. [Accessed 23 March 2016]. Available from: <https://es.unplag.com/>
15. El detector de plagio académico Unplag se integra a la perfección. [online]. [Accessed 23 March 2016]. Available from: <https://es.unplag.com/academic-plagiarism-checker/>
16. Características - MoodleDocs. [online]. [Accessed 19 May 2016]. Available from: <https://docs.moodle.org/all/es/Caracter%C3%ADsticas>

17. Moodle.org: Old modules and plugins. [online]. [Accessed 23 March 2016]. Available from: <https://moodle.org/mod/data/view.php?d=13&rid=2141&filter=1>
18. Plagiarism Prevention Turnitin - MoodleDocs. [online]. [Accessed 23 March 2016]. Available from: https://docs.moodle.org/26/en/Plagiarism_Prevention_Turnitin
19. 9 plataformas MOOC para masificar el aprendizaje y transformar la formación en línea. [online]. [Accessed 4 February 2016]. Available from: <http://www.americalearningmedia.com/edicion-016/191-tester/2367-9-plataformas-mooc-para-masificar-el-aprendizaje-y-transformar-la-formacion-en-linea>
20. Hacia un código de ética para la educación virtual, basado en las cinco mentes del futuro - Revista Mexicana de Bachillerato a Distancia. [online]. [Accessed 4 February 2016]. Available from: <http://bdistancia.ecoesad.org.mx/?articulo=hacia-un-codigo-de-etica-para-la-educacion-virtual-basado-en-las-cinco-mentes-del-futuro>
21. Lección 6 - Definición de Ingeniería de Software. [online]. [Accessed 4 February 2016]. Available from: http://datateca.unad.edu.co/contenidos/301404/301404_ContenidoEnLinea/leccin_6__definicion_de_ingeniera_de_software.html
22. Sistema para la detección de plagio y validación de estructura en los documentos científicos emitidos en el Centro de Información Científico Técnica (CICT) del ISMMM. [online]. [Accessed 9 February 2016]. Available from: <http://caribeña.eumed.net/sistema-para-la-deteccion-de-plagio-y-validacion-de-estructura-en-los-documentos-cientificos-emitidos-en-el-centro-de-informacion-cientifico-tecnica-cict-del-ismmm/>
23. TAMARA RODRÍGUEZ SÁNCHEZ. *Metodología de desarrollo para la Actividad productiva de la UCI*. 6 March 2015.

24. Qué es HTML. [online]. [Accessed 9 February 2016]. Available from:
<http://www.desarrolloweb.com/articulos/que-es-html.html>
25. Definición de CSS - ¿Qué son las hojas de estilo o cascading style sheets?
[online]. [Accessed 9 February 2016]. Available from:
<https://www.masadelante.com/faqs/css>
26. Introducción - JavaScript | MDN. [online]. [Accessed 9 February 2016]. Available
from: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>
27. Bootstrap Get Started. [online]. [Accessed 19 May 2016]. Available from:
http://www.w3schools.com/bootstrap/bootstrap_get_started.asp
28. Introducción a jQuery. [online]. [Accessed 19 May 2016]. Available from:
<http://www.desarrolloweb.com/articulos/introduccion-jquery.html>
29. PHP: ¿Qué es PHP? - Manual. [online]. [Accessed 9 February 2016]. Available
from: <http://php.net/manual/es/intro-what-is.php>
30. - Symfony. [online]. [Accessed 9 February 2016]. Available from:
http://symfony.com/legacy/doc/gentle-introduction/1_4/en/01-Introducing-Symfony
31. NetBeans IDE - Overview. [online]. [Accessed 9 February 2016]. Available from:
<https://netbeans.org/features/index.html>
32. Unified Modeling Language (UML). [online]. [Accessed 9 February 2016]. Available
from: <http://www.uml.org/>
33. El Lenguaje de Modelado Unificado (UML). [online]. [Accessed 9 February 2016].
Available from: <http://www.docirs.com/uml.htm>
34. Visual Paradigm frequently asked questions. [online]. [Accessed 9 February 2016].
Available from: <http://www.visual-paradigm.com/support/faq.jsp>

35. Qué es PostgreSQL y cuáles son sus ventajas. [online].
[Accessed 9 February 2016]. Available from: <https://platzi.com/blog/que-es-postgresql/>
36. pgAdmin: Feature. [online]. [Accessed 9 February 2016]. Available from:
<http://www.pgadmin.org/features.php>
37. About the Apache HTTP Server Project - The Apache HTTP Server Project.
[online]. [Accessed 9 February 2016]. Available from:
http://httpd.apache.org/ABOUT_APACHE.html
38. GRETHELL CASTILLO REYES, YANISLEY GONZÁLEZ GONZÁLEZ and
GUILLERMO LUZUA FARIAS. *Técnica de clasificación bayesiana para identificar posible
plagio en información textual* [online]. December 2014. Revista Cubana de Ciencias
Informáticas. Available from:
[http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path\[\]=759&path\[\]=300](http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path[]=759&path[]=300)
39. SOMMERVILLE, Ian. *Ingeniería del software. Séptima Edición* [online]. Available
from: <http://technologycompartida.blogspot.com>
40. 3. Técnicas para Identificar Requisitos Funcionales y No Funcionales -
Metodología Gestión de Requerimientos. [online]. [Accessed 23 March 2016]. Available
from: <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>
41. ¿Qué son las historias de usuario?: 7 preguntas y respuestas - La Oficina de
Proyectos de Informática. [online]. 24 April 2013. [Accessed 19 May 2016]. Available from:
<http://www.pmoinformatica.com/2013/04/que-son-las-historias-de-usuario-7.html>
42. PRESSMAN, Roger. *Ingeniería del software. Un enfoque práctico. Séptima edición.*
43. 2.1 DEFINICIÓN DE UN MODELO DE DATOS. | Tombasededatos's Blog. [online].
[Accessed 19 May 2016]. Available from:
<https://tombasededatos.wordpress.com/2010/08/28/2-1-definicion-de-un-modelo-de-datos/>

44. RUBENFA. Patrones de diseño: qué son y por qué debes usarlos. *Genbeta Dev* [online]. 14 July 2014. [Accessed 19 May 2016]. Available from: <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos> Artículo que explica qué son los patrones de diseño y por qué todo desarrollador debería utilizarlos...
45. LARMAN, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*.
46. POTENCIER, Fabien and ZANINOTTO, Francois. *Symfony la guía definitiva* [online]. Available from: www.librosweb.es
47. JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James. *El proceso unificado de desarrollo de software*.
48. ALARCÓN, Raúl. *DISEÑO ORIENTADO A OBJETOS CON UML* [online]. Grupo EIDOS Consultoría y Documentación Informática, S.L., 2000. Available from: www.grupoeidos.com/www.eidos.es
49. JURISTO, Natalia, M. MORENO, Ana and VEGAS, Sira. *TÉCNICAS DE EVALUACIÓN DE SOFTWARE*. 17 October 2006.