TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE ACAPULCO

SISTEMA PARA VERIFICAR LA AUTENTICIDAD DE LOS TRABAJOS ENTREGADOS EN FORMATO DIGITAL PARA OBTENER EL GRADO DE LICENCIATURA EN EL INSTITUTO TECNOLOGICO DE ACAPULCO

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:
MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:
ING. CRISOL ANGELINA MENDIOLA PIZA

DIRECTOR DE TESIS:
M.T.I. ELOY CADENA MENDOZA

CO-DIRECTOR DE TESIS:

M.T.I. JUAN MIGUEL HERNANDEZ BRAVO

ACAPULCO, GRO., DICIEMBRE 2019

El presente trabajo de tesis fue desarrollado en la *División de Estudios de Posgrado e Investigación* del *Instituto Tecnológico de Acapulco*, perteneciente al Programa Nacional de Posgrados de Calidad (PNPC-CONACYT).

Con domicilio para recibir y oír notificaciones en Av. Instituto Tecnológico de Acapulco s/n, Crucero del Cayaco, Acapulco, Guerrero, México. C.P. 39905.

Becario:	Crisol Angelina Mendiola Piza.
CVU:	851402.
Núm. de apoyo:	627002.
Grado:	Maestría

Descargo de Responsabilidad Institucional

El que suscribe C. Crisol Angelina Mendiola Piza alumna de la Maestría en Sistemas Computacionales , con el número de Control G17320001 declara que el presente documento intitulado “Sistema para verificar la autenticidad de los trabajos entregados en formato digital para obtener el grado de licenciatura en el instituto tecnológico de Acapulco, que fue desarrollado bajo la supervisión y dirección del asesor M.T.I. Eloy Cadena Mendoza ,es un trabajo propio y original, el cuál no ha sido utilizado anteriormente en institución alguna para propósitos de evaluación, publicación y/o obtención de algún grado académico.

Además, se han recogido todas las fuentes de información utilizadas, las cuales han sido citadas en la sección de referencias bibliográfica de este trabajo.

Nombre: Ing. Crisol Angelina Mendiola Piza
Ingeniera en Sistema Computacionales

Acapulco, Gro., Diciembre 2019

Agradecimientos

A Conacyt por el apoyo económico a lo largo de dos años en este posgrado y al Instituto Tecnológico de Acapulco por creer en nosotros para ser parte de esta segunda generación de la Maestría.

A mi Director de Tesis. M.T.I. Eloy Cadena Mendoza, por su valiosa asesoría en el trascurso de esta maestría, en especial al final de esta etapa donde me costó trabajo dar una correcta redacción y finalmente gracias por creer en mí y defender este proyecto cuando solo era una propuesta que después de estos años logra convertirse en una Tesis donde se plasma el desarrollo del proyecto.

A mis compañeros de generación con los que conviví y aprendí algo de cada uno al final del día, con ellos me di cuenta que puedo exigirme más de lo que ya soy, gracias por no dejarme sola al final y brindarme su apoyo y ayuda en esta última etapa, siempre tendrán mi apoyo y amistad incondicional abejorros programadores de la vida, gracias por ser como una segunda familia en estos dos años.

Dedicatorias

A mi madre por ser siempre mi ejemplo de vida y pilar en mi familia, por brindarme mi espacio todo este tiempo que me veía que llegaba cansada a casa y solo subía a preguntarme si necesitaba algo, gracias por ser el mejor ejemplo en mi vida y sobre todo gracias por estar ahí cuando más lo necesitaba y cuando creía estar sola.

A mi primer Jefe en Computo el Ing. José Francisco Gazga Portillo, quien fue la primera persona en creer en mi aun cuando yo no lo hacía, a enseñarme todo lo que se hoy y ayudarme a formar profesionalmente desde mis prácticas profesionales en Computo en mi Licenciatura, a pesar de que nosotros con mis chamacos hacíamos los trabajos en red o telefonía usted siempre dio la cara por nosotros cuando algo hacíamos mal, gracias por ser un excelente Jefe y Líder

A mi Jefa la M.C. Laura Sánchez Hernández por brindarme la oportunidad de conocer un nuevo ambiente laboral como coordinadora en la Institución y por creer en mi como la persona indicada para ese puesto que desempeñe por cuatro años y sobre todo gracias por dejarme seguir creciendo profesionalmente cuando le dije que quería estudiar un posgrado y tendría que dejar el puesto, usted es y será un ejemplo a seguir para mí

A mis compañeras de trabajo en el tiempo que estuve en la coordinación las cuales siempre me ayudaban en los temas que desconocía y en procesos internos que se llevan en la

oficina, siempre fueron un gran apoyo y fortaleza para mí en ese tiempo, gracias por formar parte del equipo y su amistad, Tania, Roxana, Verito y a ti Francis en especial porque siempre fuiste mi apoyo cuando tenía tantas dudas y guía en mis problemas.

A mi equipo de Servicios Escolares que siempre me ha orientado en mis procesos, tato de la coordinación o procesos propios, gracias por ayudarme cuando se me cargaba el trabajo en coordinación y veían como esta y llegaban simplemente a decirme en que te ayudamos Cris, siempre estarán en mis pensamientos Naty, Lety, Emma, Brillito,Olfi, Gabriel, Doña Mary, Tomas y al Ing. Javier Sánchez Padilla.

A mis hermanas Pamela y Berenice, a pesar que no somos hermanas de sangre ustedes se han ganado ese lugar en mi corazón, por estar ahí siempre que lo necesite y brindarme apoyo incondicional cuando sentía que no podía más, gracias por demostrarme que aún hay personas que son diferentes y que están ahí cuando menos lo imaginas y sobre todo cuando más las necesitas y son una alegría en tu vida, las quiero.

A los Huguitos, quienes aportaron cada uno sus asesorías en temas que desconocía y me tuvieron la paciencia de explicarme y quedarse hasta tarde mientras estaba trabajando y no me fuera sola de la oficina, Raúl, Beto y Rogelio.

A mi persona especial que ha estado conmigo desde que decide empezar este posgrado , aunque no estas cerca siempre me has escuchado y ayudado cuando tenía

problemas o quería irme por un camino fácil, siempre me orientaste a hacer las cosas correctas y nunca lo hiciste fácil para mí , siempre buscaste una forma de como yo aprendiera a hacer las cosas o aprender a mi modo y sacar el trabajo por mí misma, gracias por no dejar que fuera una más del montón y sobre todo gracias por seguir apoyándome para superarme en todo lo que quiero hacer, gracias por soportarme ni cuando yo podía conmigo misma . Te quiero Mitzunari.

Resumen

Una búsqueda por similitud consiste en recuperar todos los objetos dentro de una base de datos que sean parecidos o relevantes en una determinada consulta. Se ha avanzado en numerosas estructuras métricas que funcionan como índices y realizan un procesamiento de datos, con el fin de disminuir las evaluaciones de distancia en el momento de la búsqueda. Es necesario alimentar una gran cantidad de información y contar con las tecnologías y herramientas necesarias para obtener resultados óptimos en menor tiempo y bajo costo. Este proyecto aborda una problemática sobre la similitud detectada en los trabajos de titulación de los egresados del Instituto Tecnológico de Acapulco. Se propone desarrollar un sistema de búsqueda de similitudes, comparando el nuevo trabajo con los ya existentes en formato electrónico PDF. Esta comparación arrojará como resultado un porcentaje de similitud del nuevo archivo a comparar contra los existentes. Con este porcentaje arrojado, los involucrados pueden decidir cómo se va a proceder con el nuevo trabajo recibido. Se presenta un marco teórico con los temas relacionados a lenguajes de programación, sistemas gestores de base de datos y archivos electrónicos, para conocer lo que tienen en el mercado estas herramientas.

Palabras clave: Búsqueda por similitud, espacios métricos, porcentaje de similitud

Abstract

A similarity search consists in recovering all the objects within a database that are similar or relevant in a given query. There have been advances in numerous metric structures that function as indexes and perform a data processing, in order to reduce the distance evaluations at the time of the search. It is necessary to feed a large amount of information and have the necessary technologies and tools to obtain optimal results in less time and at low cost. This project addresses a problem about the similarity detected in the graduation work of graduates of the Technological Institute of Acapulco. It is proposed to develop a similarity search system, comparing the new work with the existing ones in PDF electronic format. This comparison will result in a similarity percentage of the new file to be compared against the existing ones. With this percentage thrown in, those involved can decide how to proceed with the new work received. It presents a theoretical framework with topics related to programming languages, database management systems and electronic files, to know what these tools have on the market.

Keywords: Search by similarity, metric spaces, percentage of similarity

Índice de Contenido

Índice de figuras

Índice de tablas

Capítulo 1 Introducción

1.1 Antecedentes del problema

El proceso de titulación comienza una vez que el egresado cumple con el 100% de los créditos de su plan de estudio y acreditación de un programa de lengua extranjera. Cumpliendo estos requisitos el egresado comienza su proceso de titulación con el Departamento de Servicios Escolares integrando su expediente para ese propósito, pagos y documentación requerida en original y copias. Teniendo su copia del expediente recibido se dirige a la División de Estudios Profesionales para presentar una propuesta de titulación.

Inicialmente el Instituto Tecnológico ofreció las carreras de Nivel Medio Superior Técnico en Aire Acondicionado y Refrigeración, Técnico en Mantenimiento Mecánico, Técnico Laboratorista Químico, Técnico en Obras Arquitectónicas y Técnico en Administración de Personal. A partir de 1983 se comienza a ofertar el Nivel Superior con las carreras de Ingeniería Electromecánica en Planta y Mantenimiento y Licenciatura en Relaciones Comerciales para dar inicio al primer Plan de Estudios a Nivel Superior con el plan de estudios 1975 para las carreras de Ingeniería Electromecánica en Planta y Mantenimiento y Licenciatura en Relaciones Comerciales.

Cada opción de titulación depende del plan de estudios al que pertenezca el egresado. En los últimos tres planes de estudios son en los que se comienza a incrementar la matrícula. Debido a esto se incrementa el número de egresados que llegan al proceso

de titulación. Los tres planes de estudios mencionados son los siguientes: plan de estudios 1993 con las siguientes opciones de titulación: a) Tesis Profesional, b) Libros de Texto o Prototipos didácticos, c) Proyectos de Investigación, d) Diseño o Rediseño de Equipo, Aparato o Maquinaria, Curso Especial de Titulación, e) Examen Global por Áreas de conocimiento, f) Memoria de Experiencia Profesional, g) Escolaridad por promedio, h) Escolaridad por Estudios de Postgrado. Dentro del segundo plan de estudios 2004-2009 se encuentran las siguientes formas de titulación: a)Tesis Profesional, b)Proyectos de Investigación, c)Exámenes por áreas de conocimiento, d)Escolaridad por promedio, e)Informe de residencia Profesional. A partir de esta retícula plan de estudios 2004 se descarta la opción de titulación por Escolaridad por Estudios de Postgrado. Actualmente el plan de estudios vigente es el 2010 llamado por el Modelo por Competencias; las formas de titulación son: a) Informe de Residencia Profesional, b) Proyecto de Investigación y/o Desarrollo Tecnológico, c) Informe de Estancia, d) Tesis o Tesina y e) Examen por áreas de conocimiento CENEVAL. Esta última opción no necesita entregar un engargolado para la asignación de sus asesores debido a que se presenta una copia de la constancia de testimonio *satisfactorio* o *sobresaliente* como comprobante de la acreditación del examen. Para cada opción de titulación se maneja un normativo sobre la estructura de los trabajos profesionales que se entregan en cuatro engargolados/ o en formato electrónico a la División de Estudios Profesionales para que se turnen los trabajos a las academias correspondientes en espera de la asignación de asesores y revisores de dichos trabajos.

Siguiendo al proceso de titulación, una vez entregados los cuatro engargolados, al Departamento Académico le corresponde asignar a los revisores del trabajo. Se

menciona en el manual de lineamientos TecNM Capítulo 14. Lineamiento para la titulación Integral, Sección 14.4 Políticas de Operación. Apartado 14.4.3 Del Jefe de Departamento Académico el cual nos dice que se asignarán asesores a partir de que el estudiante cursa la asignatura del Taller de Investigación II. A partir del plan de estudios 2010 se realiza una modificación al mapa curricular en donde se cambia de semestre la asignatura de Taller de Investigación II, que se encontraba en el séptimo semestre y ahora se recorre al noveno semestre para que el estudiante comience a trabajar en su proyecto para titulación. El Departamento Académico emitirá un oficio de asignación de asesores para que ellos junto con el estudiante trabajen en revisiones de su trabajo de tesis.

Terminando estas revisiones se emite la liberación del proyecto de titulación integral a la División de Estudios Profesionales junto con el informe del proyecto de titulación integral en formato digital a la División de Estudios Profesionales y Centro de Información, así como lo menciona el Capítulo 14. Lineamiento para la titulación Integral, Sección 14.4 Políticas de Operación, Apartado 14.4.1.5 De los requisitos. Se menciona este último lineamiento en especial debido a que el presente trabajo se enfoca en la búsqueda de similitudes de trabajos digitales ya existentes en la Tesiteca, debido a que es el lugar donde se archivan los documentos físicos desde que comenzaron a llevarse a cabo las titulaciones en la Institución.

En el 2005 se acepta el primer trabajo en formato digital junto con el empastado de la carrera de Arquitectura. A partir esta fecha en adelante se pide a los egresados se entregue al Departamento de la División de Estudios Profesionales los trabajos en físico y

en un disco con el archivo en digital en formato PDF. Todo este histórico mencionado de trabajos de titulación en digital se le puede facilitar al alumnado si llega a solicitar algún archivo con los encargados de la Tesiteca.

1.2 Planteamiento del problema.

La gráfica mostrada en la tabla1.1 representa el histórico de las titulaciones que se han realizado desde el año 2012 al 2017, mostrando en ella el total de alumnos titulados con cada una de las opciones de titulación que cuenta la Institución.

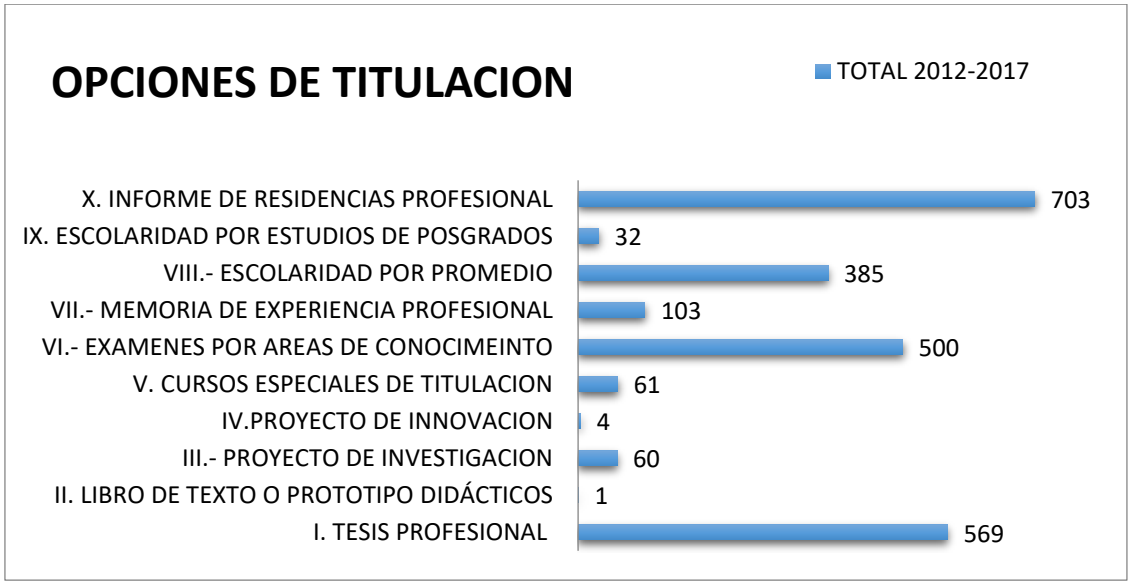


Tabla 1-1Total de alumnos titulados

En la opción de Residencias Profesionales encontramos que se tiene el mayor número de alumnos titulados. Las Residencias Profesionales pueden seleccionarse cuando el alumno cumple con el 90% de sus créditos aprobados y puede seleccionar de forma simultánea con las asignaturas pendientes en caso de que el alumno aun tenga asignaturas

pendientes para concluir con su retícula. El alumno, previo a seleccionar la asignatura comienza a buscar un proyecto que desarrollará en el transcurso de su Residencia Profesional. En el banco de proyectos del Instituto Tecnológico es donde se ofertan de forma semestral los proyectos que ofrecen las empresas que tienen un convenio establecido con la Institución para que los alumnos puedan realizar sus Residencias Profesionales en ellas. El departamento de Gestión Tecnológica y Vinculación es el encargado de gestionar dicho convenio entre la empresa y la Institución con un tiempo de validez de un año; esto permitirá que de forma constante las empresas están solicitando estudiantes como residentes para desarrollar sus proyectos. Generalmente estas empresas o instituciones con las que se tienen los convenios establecidos son únicas en el estado por ejemplo IMSS, TELMEX, LAS BRISAS; CAPAMA, CFE; manejando en muchas ocasiones un mismo plan de trabajo para un proyecto ya existente, sin renovarlo en el banco de proyectos que se oferta en la institución semestre tras semestre.

Lo anterior tiene como consecuencia que los nuevos alumnos que llegan a la empresa a realizar sus Residencias estén trabajando sobre un mismo proyecto que otros compañeros ya habían trabajado en semestres anteriores. Al terminar las Residencias los estudiantes pueden titularse por la opción de Memoria de Residencias Profesionales con este mismo proyecto, ocasionando que estos trabajos sean similares debido a que es un proyecto sobre el que se trabaja semestre tras semestre. Una vez que el alumno tiene la empresa y un proyecto para trabajar presentará un engargolado con el anteproyecto a su coordinador de carrera; este documento describe el proyecto y las actividades que realizará en su estancia dentro de la empresa. Incluso en esta parte se puede notar como a

pesar de que el título del proyecto no es el mismo, al revisar las actividades son muy parecidas o son las mismas a un proyecto de la empresa presentado anteriormente por otro alumno en un semestre pasado. Al concluir las residencias profesionales el alumno presenta un informe final a su coordinador, este documento es un reporte técnico que contiene el desarrollo y las actividades que realizó en el proyecto.

Más adelante este mismo trabajo se puede utilizar como la forma de titulación X, Informe de Residencias Profesionales. El estudiante deberá de trabajar sobre su reporte técnico con sus revisores asignados por la academia para presentar este trabajo en un informe de residencias profesionales y así poder titularse mediante esta opción. Se mencionó en un inicio que esta opción de titulación es la que cuenta con un mayor número de titulados en la institución, pero también es donde se pueden detectar más similitudes de trabajos desde el momento en que el alumno entrega su reporte final de Residencias debido a que se está trabajando sobre un mismo proyecto y el reporte que entregan es similar al de los compañeros que ya anteriormente habían realizado su Residencias Profesional en la misma empresa sobre un mismo proyecto. Toda esta serie de eventos que se están arrastrando desde el momento en que el alumno selecciona un proyecto generará como resultado final que el alumno al momento de titularse mediante la opción X, Informe de Residencias Profesionales, entregue un trabajo similar al de sus compañeros que ya se han titulado mediante esta opción.

La segunda opción de titulación con un mayor número de titulados es opción I, Tesis Profesional. Este tipo de trabajo consta de una propuesta concreta, desarrollada

mediante una metodología de investigación; dicha propuesta deberá ser original, en este último término me refiero a que la tesis pueda ser sometida o demostrada mediante pruebas y razonamientos apropiados para demostrar su originalidad. Al realizar una tesis o algún proyecto de investigación se recomienda al estudiante llevar a cabo su registro del tema de la tesis a su nombre. Si el alumno no realiza este paso se queda expuesto a que más adelante alguien más pueda copiar su trabajo ya que no tiene un registro previo de derechos. Retomando la primera opción de titulación que fue Informe de Residencias Profesionales, existen casos en los que el proyecto que el alumno desarrolló en sus Prácticas Profesionales lo transformen a una tesis ya que está genera un valor curricular mayor al alumno y también al asesor que será su director de tesis.

Pero con esta última observación se recae en el mismo inconveniente de titularse por Informe de Residencias Profesionales ya que los egresados siguen trabajando sobre los mismos proyectos, con la diferencia que ahora cambian de Opción de titulación de Informe de Residencias Profesionales a Tesis Profesional, pero en su desarrollo siguen trabajando sobre la copia de un proyecto ya desarrollado teniendo como única variante la opción de titulación.

1.3 Objetivos

1.3.1 Objetivo general

Detectar similitudes en los escritos de los trabajos de titulación a nivel licenciatura del Instituto Tecnológico de Acapulco.

1.3.2 Objetivos específicos

- Desarrollar un sistema que realice la comparación de trabajos electrónicos en formato PDF.

- Alertar mediante un porcentaje de similitud a los interesados.

- Conocer las características de las versiones Acrobat 7.0, Acrobat 8.0, Acrobat 9.0, Acrobat 9.1 y Acrobat X (10) de los archivos electrónicos en formato PDF.

- Encontrar un algoritmo óptimo para la comparación de los trabajos existentes.

1.4 Hipótesis

Proporcionar una herramienta de apoyo para detectar similitudes de trabajos nuevos con los ya existentes de las carreras de licenciatura en el Instituto Tecnológico de Acapulco con la finalidad de ayudar a disminuir la copia de reporte de los trabajos derivados de los proyectos afines. Se desarrollará un sistema que realice la comparación del nuevo trabajo entregado contra los ya existentes. Dicho sistema comenzara a analizar el texto desde el título hasta el marco teórico; limitando un porcentaje de similitud

tolerable hasta un valor determinado, en cuanto se detecte que el porcentaje es mayor a esto se sigue realizando la comparación del demás contenido del archivo contra el existente para poder obtener un porcentaje final de similitud de las partes que conforman dicho documento.

1.5 Justificación

1.5.1 Académica

Los departamentos académicos están involucrados como parte del proceso de titulación debido a que es ahí donde se encuentran los asesores, los cuales juegan un papel importante ya que se necesita de su firma de liberación para poder llegar a la parte final de la obtención del grado. Antes de llegar a esta parte, el trabajo de cada uno de los asesores es encargarse de revisar el trabajo de sus asesorados, tomando en cuenta diversos criterios dentro de la revisión, por mencionar algunos son: la estructura, coherencia y un punto importante es la originalidad del trabajo, donde se demuestre que es trabajo propio de cada uno y revisar una correcta referencia académica de trabajos relacionados, artículos y libros.

1.5.2 Institucional

Los egresados que genera una institución educativa forman parte de la comunidad profesional. En las empresas, como parte de los requisitos les piden tener un Título y Cédula profesional para poder postularse a un puesto de trabajo. Que deberán desarrollar profesionalmente bajo un juramento de ética profesional, hecho en su institución. Por lo que un egresado está representando a una Institución.

1.6 Alcance y limitaciones

1.6.1 Alcance

Este proyecto realizará la comparación de los trabajos de titulación de los egresados de licenciatura del Instituto Tecnológico de Acapulco, obtenidos del año 2005 al 2017 en formato electrónico PDF.

Se desarrollará un sistema informático para buscar la similitud de los trabajos que presentan los egresados, para obtener el grado académico de licenciatura. Como resultado esta búsqueda se obtendrá un porcentaje de similitud, detectada mediante la comparación de este nuevo trabajo con los existentes del periodo ya mencionado.

1.6.2 Limitaciones

- Se iniciará con el año 2005 a realizar la comparación de trabajos nuevos contra esta fecha de inicio.

- Se trabajará con archivos electrónicos en formato PDF de las siguientes versiones: Acrobat 7.0, Acrobat 8.0, Acrobat 9.0, Acrobat 9.1 y Acrobat X (10).

- Estos archivos en formato electrónico PDF provienen de las siguientes opciones de titulación de licenciatura, que generan un archivo electrónico junto con su empastado; a) Memoria de Residencia Profesional y b) Tesis.

Capítulo 2 Marco Teórico

2.1 Estado del arte

En el artículo Procesamiento de búsquedas por similitud. Tecnologías de paralelización e indexación de: Eder dos Santos, Albert Aníbal Osiris Sofía, Roberto Uribe Paredes. (Año 2015, Volumen 7, No.2). Procesamiento de búsquedas por similitud. Tecnologías de paralelización e indexación (Paginas 111-138). Obtenido del Departamento de Ingeniería en Computación de la Universidad de Magallanes, Chile. https://dialnet.unirioja.es/servlet/articulo?codigo=5179331.

Este artículo aborda la búsqueda por similitud y la implementación de estructuras métricas sobre entornos paralelos. La búsqueda por similitud consiste en la búsqueda de objetos más similares a través de una búsqueda por rango o de vecinos más cercanos en un espacio métrico utilizando la siguiente formula: definiendo un espacio métrico como un conjunto X con una función distancia d: X2→R, tal que $\forall x , y , z \in X$, se deben cumplir las propiedades de: positividad (d(x,y)>=0 y d(x,y)=0 ssi x=y), simetría (d(x,y)=d(y,x)) y desigualdad triangular (d(x,y)+d(y,z)>=d(x,z)). Existen distintas estructuras para buscar en espacios métricos, las cuales pueden ocupar funciones discretas o continuas de distancia. Estas estructuras se basan en la búsqueda de pivotes y otras en "*clustering".* Para el caso de encontrar las estructuras métricas en "*clustering",* utilizan el diagrama de Voronoi para dividirlas de la siguiente forma: la subdivisión del plano en n áreas, una por cada centro ci del conjunto {c1, c2, ..., cn}, tal que q ∈al área ci sí y sólo sí

la distancia euclidiana d (q, ci) < d (q, cj) para cada cj, con j ≠ i. En los métodos basados en pivotes, se selecciona un conjunto de pivotes y se pre calculan las distancias entre los pivotes y todos los elementos de la base de datos, de la siguiente forma:

Sea {p1, p2, ..., pk} ∈ X un conjunto de pivotes. Para cada elemento y de la base de datos Y, se almacena su distancia a los k pivotes (d(x, p1), ..., d(x, pk)). Dada una consulta q y un rango r, se calcula su distancia a los k pivotes (d(q, p1), ..., d(q, pk)).

Si para algún pivote pi se cumple que | d(q, pi) – d(x, pi) | > r, entonces por desigualdad triangular se tiene que d(q, x) > r; por lo tanto, no es necesario evaluar explícitamente d (x,q). Todos los objetos que no se puedan descartar por esta regla deben ser comparados directamente con la consulta q.

Se trabajaron con plataformas de memoria compartida utilizadas para los experimentos para recaudar la información para este informe. Una de ellas OpenMP es una interfaz de programación de aplicaciones (API) para la programación multiproceso de memoria compartida en múltiples plataformas. La segunda son las Unidades de procesamiento grafico; este tipo de dispositivos permite aumentar la capacidad de procesamiento respecto de las CPU. En los experimentos realizados se seleccionan distintos espacios métricos con la finalidad de clasificar dichos espacios, se utiliza la función de distancia utilizada para la búsqueda por similitud, de esta manera es posible identificar dos grupos de espacios métricos el primer conjunto corresponde al diccionario de distintos idiomas y el segundo grupo contiene la base de datos de vectores de distintas dimensiones que representan a los distintos objetos. Como plataforma de evaluación

experimental, se ha trabajado con los espacios métricos mencionados en un entorno "*multicore"* y un entorno GPU, utilizando la estructura GMS basada en distintas cantidades de pivotes y el algoritmo de búsqueda por fuerza. Como resultado con respecto al tiempo de procesamiento para un espacio métrico de palabras utilizando el algoritmo de búsqueda conocido como Fuerza Bruta y la optimización lograda por todas las modalidades de paralelismo comparadas con el procesamiento secuencial se observan los siguientes resultados mostrados en la Figura 2.1 en Unidad de procesamiento gráfico (GPU) y Unidad de procesamiento gráfico con memoria compartida (GPU SM):

Opción	Tiempo s.		
		GPU	GPUSM
GPU	647,30	--	*0,09*
GPU SM	56,48	11,46	--

Figura 2.1 Comparación de tiempo en una búsqueda (Eder dos Santos 2015)

Conclusión: Su aporte de este trabajo es el manejo en que ellos buscan obtener sus espacios métricos y tener una mejor distribución en la búsqueda en la base de datos, y conocer el rendimiento de tiempo arrojado utilizando distintas estrategias de procesamiento paralelo, analizando el rendimiento de la búsqueda en ambientes con GPU (CUDA) y multi-core (OpenMP), concluyendo que. El procesamiento con GPU resulta más eficiente a medida que aumenta el rango de búsqueda y consecuentemente la cantidad de evaluaciones de distancia realizadas.

En el artículo Indexación y búsqueda en base de datos de: Anabella Bautista, Andrés Pascal y Juan Pablo Nuñez. (12 de Mayo del 2015). Indexación y búsqueda en

base de datos (Paginas 1-5). Obtenido del Departamento de Ingeniería en Sistemas de Información, de la Universidad Tecnológica Nacional entre ríos, Argentina. http://hdl.handle.net/10915/45629.

Se propone para gestionar objetos con alguna referencia espacial el uso de las bases de datos espaciales, haciendo referencia a que un dato puede ser en su forma más simple que un punto, este objeto puede tener valor en ciertos atributos, pero también una ubicación espacial. Para tratar este tipo de datos existen aplicaciones para los modelos de base de datos espaciales las más destacadas son los sistemas de información geográfica (SIG). Otro concepto interesante del artículo es poder asociar tiempos de vigencia a objetos almacenados, por ello surgen las Bases de Datos Temporales, que permiten manejar internamente una o más dimensiones temporales. Este modelo utiliza la siguiente función para construir un espacio métrico-temporal: es un par (U, d), donde U = O x N x N, y la función d es de la forma d : O x O ^ R+. Se implementan índices métrico-temporales en memoria secundaria , proponiendo los siguientes índices FHQT-Temporal,Historical -FHQT y Pivot-FHQT, tomando como base el índice para espacios métricos *"Fixed Height Queries Tree"* utilizando funciones de distancias discretas.Para las pruebas de estas bases de datos espaciales en este artículo de menciona la colaboración de Grupo de Estudios de Calidad y Medio Ambiente de la Regional Concepción del Uruguay de la UTN en la implementación de un Sistema de información Geográfica para el Municipio de la ciudad de Urdinarrain, con ellos se desarrolló una aplicación que permitirá gestionar los datos obtenidos a partir del levamiento de comercios, asociando cada uno de ellos a su posición geográfica, para permitir un

posterior análisis de distribución de comercios por rubro, zona geográfica y otros factores de interés para el caso particular.

Conclusión: Se observa como las base de datos temporales se utilizaron para manejar de forma interna más de una dimensión temporal, implementando los índices métrico-temporales FHQT-Temporal, Historical -FHQT y Pivot-FHQT, por su óptimo desempeño en este tipo de bases, ya que ellos utilizan funciones de distancia discretas.

En el artículo Búsqueda en grandes volúmenes de datos de: Luis Britos, María E. Di Gennaro, Verónica Gil-Costa, Fernando Kasián . (Abril 2016). Indexación y Búsqueda en grandes volúmenes de datos (Paginas 283-287). Obtenido del Centro de Investigación Científica y de Educación Superior Ensenada, México http://hdl.handle.net/10915/52900

Esta trabajo realiza su diseño de índices con el árbol de aproximación espacial distante DiSAT que muy eficiente en cuanto al número de cálculos de distancias realizados tanto en construcción como en búsquedas, debido a que se trabajó con una cantidad de datos masivos de internet, fue necesario almacenarlos en memoria secundaria, para este caso ellos diseñaron e implementaron la siguientes estructuras tructura

-tree y el DSACL+-tree, que al optimizarlas resultaron ser igual de competitivas con otras estructuras conocidas como el M-tree y DSA-tree y DSA+-tree, esperando en trabajos a futuro logar una implementación paralela eficiente de estos índices .Se trabajó

con un sistema de recuperación de imágenes basada en el contenido (CBIR) con una plataforma base de rayo de puerta programable (FPGA) para al trabajar en conjunto con ellos acelerar la búsqueda de similitud mediante los beneficios de las herramientas de síntesis de alto nivel. Al terminar de trabajar con esto se obtiene una versión se memoria secundaria (LPA). Su sistema de base de datos seleccionado fue PostgreSQL debido a que particularmente permite realizar búsqueda por similitud sobre algunos atributos que se necesitan indexar para la búsqueda mediante el método de k-vecinos más cercanos , pero más adelante se trabaja con la extensión de PostgreSQL que incluya más consultas por similitud sobre distintos tipos de datos de los que ellos manejaron, ya que en esta extensión se puede realizar este tipo de consultas y es necesario cambiarla de la que se ocuparon en las pruebas.

Conclusión: El árbol de aproximación espacial distante DiSAT para el diseño de los índices brinda un mejor resultado cuando se maneja un gran volumen de datos, en cuanto a la realización de las distancias en las búsquedas por similitud. En cuanto a su gestor de base de datos que utilizan PostgreSQL se menciona como característica relevante e importante para este tipo de búsquedas que puede manejar los atributos necesarios a indexar para ello mediante la búsqueda de k- vecinos, que otros gestores no poseen esta característica.

En el artículo Filtrado de información para la búsqueda de respuestas de: Luis Britos, María E. Di Gennaro, Veronica Gil-Costa, Fernando Kasián. (Septiembre 2016).

Filtrado de información para la búsqueda de respuestas (Paginas 145-152). Obtenido del Departamento de sistemas y lenguaje natural y sistemas de información, universidad de Alicante. http://www.sepln.org/revistaSEPLN/revista/37/19.pdf

Se ha realizado una aproximación para el filtrado de información en un sistema de (BR), basado en la clasificación de los documentos/pasajes en relevantes o no relevantes. Ejecutando un sistema RI como primer paso, para con el obtener la lista de los documentos que el sistema devuelva y a partir de estos seleccionar los que superan un determinado umbral, esto se convertiría en el segundo paso que es seleccionar los documentos relevantes. Estos documentos se van a distribuir de una forma más condensada en posiciones más altas y a partir de un determinado punto hacer una distribución esta distribución más dispersa. Para calcularlo utilizan el teorema del valor medio y se obtuvo como resultado que más del 50% de los documentos que se clasificaron en relevantes encontraron encima de ese punto. Para evaluar este método propuesto y compararlo contra otros métodos se usaron las siguientes herramientas: sistema de Recuperación de la Información (IR). (Llopis y Noguera, 2005), que es un sistema de PR que establece como baseline un número fijo de frases para este caso 8 frases y como medidas de similitud se usa okapi, la medida del coseno y dfr. En los experimentos se ha evaluado la propuesta del Teorema del Valor Medio (MVT) y se ha comparado con el método QALC y con un sistema de referencia (Baseline), el cual devuelve un número fijo de pasajes (400 frases) para cada pregunta. El sistema de BR con el Baseline ha obtenido un Rango Reciproco Medio (MRR) medio de 0.4406 s, mientras que con el método MVT, el

sistema de BR obtiene un MRR medio de 0.4717 s. Se concluye que el método MVT mejora los resultados con respecto al Baseline en el sistema de BR en un 7.05 %.

Conclusión: Se realiza un limpieza previa de los documentos con los que se van a realizar las comparaciones para eliminar los espacios vacíos que puedan tener o valores atípicos, para el momento de realizar la búsqueda sea más eficiente ya que anteriormente ya pasaron por un pre procesamiento los archivos. Se mencionan dos herramientas para evaluar el método propuesto las cuales son el sistema IR-n y el sistema PR, estos sistemas se pueden tomar también para evaluar los métodos con los que se esté trabajando y obtener una tabla de comparaciones en cuanto a los resultados de tiempo arrojados con cada uno y finalmente poder seleccionar el que nos maneja un menor costo de tiempo, es decir seriá nuestra herramienta evaluadora.

En el artículo Implementación de un digesto digital paralelo para búsquedas por similitud sobre documentos: Solar, Roberto, Uribe Paredes, Roberto , Gesto, Esteban ,Osiris, Sofía 2008. (Octubre 2008). Implementación de un digesto digital paralelo para búsquedas por similitud sobre documentos (Paginas 1-11). Obtenido Red de Universidades con Carreras en Informática (RedUNCI). http://hdl.handle.net/10915/21972

Este trabajo presenta una solución eficiente y de bajo costo de un motor de búsqueda en paralelo, presentando una alternativa para consultas en un digesto digital

institucional, la búsqueda por documentos por similitud. Para el proyecto se trabajó con dos estructuras métricas el árbol geométrico de acceso cercano al vecino (GNAT) y su evolución (evolución del árbol geométrico de acceso cercano al vecino (EGNAT) debido a que tienen buen desempeño en espacios métricos de alta dimensión, su estructura es basada en clustering y usan los diagramas de Varonoi para dividir el espacio. Se implementó un Digesto Digital con la información de una de las universidades a la que pertenecen los autores, debido a que presenta un volumen de datos atractivos para la realización de las pruebas, más adelante se planea utilizar un volumen de al menos 500,000 páginas de documentos que corresponderán a la información de los últimos 10 años. Durante una etapa de pre procesamiento de documentos, se realizan procesos de eliminación de palabras vacías para obtener los vectores representativos de la dimensión 9,341, de la dimensión original de 18,000. Su motor de búsqueda se implementa sobre un *"cluster"* de PCs, donde se realizaron pruebas utilizando 2,4 y 8 máquinas en el *"cluster"* para tener una visión de la disminución de los costos de procesamiento, cada máquina del *"cluster"* tiene las siguientes características: procesador Intel core 2 Duo de 2.2 GHz, memoria RAM de 1 Gbyte, discos duros de 7200rpm Sata II. Los resultados obtenidos de esta implementación paralela fueron evaluados por la aplicación *"Speed-up"* que se define como la proporción de tiempo que toma solucionar un problema sobre un procesador y el tiempo requerido para solucionar el mismo problema sobre un computador paralelo con p procesadores. En los resultados finales se observa que al implementar una estructura EGNAT existe una considerable disminución en los costos de tiempo, porque en evaluaciones de distancia ambas estructuras tiene un mismo desempeño al recuperar un 10% de la base de datos. En una versión original el digesto

almacenaba el texto de los documentos en tablas almacenadas en un servidor de base de datos y distribuida en *"cluster"* de PC, ya con esta nueva propuesta se agrega la posibilidad de realizar búsquedas por similitud o aproximada de documentos, mediante la nueva estructura métrica que arrojó mejores resultados.

Conclusión: La estructura métrica EGNAT que trabaja en espacios métricos de altas dimensiones reduce de forma considerable los costos de tiempos en cuanto a la realización de búsquedas, para clasificar los espacios métricos se optó la utilización de los diagramas de Varonoi debido a que proceso un gran volumen de datos de manera más eficiente. Otro punto importante es que se obtienen mejores resultados con un pre procesamiento de documentos mediante una limpieza para obtener finalmente vectores representativos de la dimensión. Ocupa como herramienta de evaluación el Speed -up para calcular los tiempos de la búsquedas con los diferentes procesadores y las estructuras que se manejaron, para con estos resultados obtenidos determinar cuál será el óptimo.

En el artículo Evaluación de estructuras métricascon Unidades de Procesamiento Gráfico de Propósito General de: Sofia, Albert Osiris. (15 de diciembre del 2015). Evaluación de estructuras métricas con Unidades de Procesamiento Gráfico de Propósito General (Paginas 1-6). Obtenido de la Revista Tecnica Admnistrativa, Buenos Aires Argentina www.cyta.com.ar/ta1404/v14n4a3.htm

A fin de optimizar este procesamiento de búsqueda por similitud se han desarrollado numerosas estructuras métricas, que funcionan como índices y realizan un preprocesamiento de los datos a fin de disminuir las evaluaciones de distancia al momento de la búsqueda. Existen una serie de tecnologías para realizar implementaciones de procesamiento paralelo. Se incluyeron en el trabajo las más vigentes las tecnologías basadas en arquitecturas multi-CPU (multi-core) y GPU / multi-GPU, que son interesantes debido a las altas prestaciones y los bajos costes involucrados. El artículo aborda la búsqueda por similitud y la implementación de estructuras métricas sobre entornos paralelos. En cuanto a búsquedas por similitud en espacios métricos utilizan el algoritmo de los k vecinos más cercanos. Para las estructuras métricas de búsquedas se van por estructuras son GNAT, M-Tree, SAT, Slim-Tree y EGNAT. Se mencionan las plataformas paralelas de memoria compartida sobre las cuales se puede implementar estas estructuras métricas enfocándose inicialmente en esa plataforma de memoria distribuida que usan bibliotecas de alto nivel como MPI o PVM, y memoria compartida usando directivas de OpenMP. En los experimentos llevados a cabo por el grupo de investigación, se seleccionan distintos espacios métricos. Con la finalidad de clasificar dichos espacios, se utiliza la función de distancia utilizada para la búsqueda por similitud. De esta manera, es posible identificar dos grupos de espacios métricos. El primer grupo corresponderá a diccionarios de distintos idiomas y el segundo es de espacios métricos que contienen la base de datos de vectores de distintas dimensiones. Como plataforma de evaluación experimental, se ha trabajado con los espacios métricos mencionados en un entorno multicore y un entorno GPU, utilizando la estructura GMS basada en distintas cantidades de pivotes y el algoritmo de búsqueda por fuerza bruta. Finalmente, el

hardware utilizado corresponde a un Intel® Core™ i7-2600 CPU @3.40GHz de 4 núcleos y soporte de Hyper-Threading, 12GB de memoria principal y dos tarjetas Nvidia EVGA DDR5 de 384 cores CUDA y 1 GB de memoria global cada una. La codificación de las estructuras métricas y de los algoritmos de búsqueda se ha realizado utilizando el lenguaje C (gcc 4.3.4), ejecutadas sobre una plataforma Linux Ubuntu 12.04 LTS (Precise Pangolin); para la paralelización, se ha adoptado CUDA SDK v3.2 para el caso de las aplicaciones GPU, y se ha utilizado la librería OpenMP para la paralelización multi-CPU. Los resultados obtenidos en cuanta eficiencia de procesadores fueron los siguientes:

- Secuencial: utilización de 1 sólo core en su capacidad máxima.
- Multi-core: utilización de los 4 cores en su máxima capacidad con OpenMP.
- Hyper-Threading(TM): utilización de los 4 cores en su máxima capacidad en modo Hyper-Threading con OpenMP.

En cuanto a la comparación de Procesamiento Secuencial contra Procesamiento Paralelo se demuestra que la mayor diferencia con el uso de memoria compartida GPU SM respecto a un procesamiento secuencia es menor, los datos arrojados al ser analizados con el Speedup fueron los siguientes :GPU 64730S, GPU SM 5649S, MC 555509S y SEC 340015S.

Conclusion : Se mencionan tecnologías vigentes basadas en arquitecturas multi-CPU (multi-core) y GPU / multi-GPU que aportan bajo costo y pueden realizar un

procesamiento en paralelo. Nos muestra bibliotecas de alto nivel para trabajar con memoria compartida las cuales con MPI o PVM y las directivas de OpenMP. En este trabajo se conoció la eficiencia de los diferentes procesadores que se manejaron dejando seleccionando a Hyper-Threading(TM) que fue el que utiliza su capacidad máxima para este tipo de procesamientos.

En el artículo Búsquedas por similitud en PostgreSQL de: Kasián, Fernando, Reyes, Nora Susana. (Octubre 2012). Búsquedas por similitud en PostgreSQL Búsquedas por similitud en PostgreSQL (Paginas 1-10). Obtenido de: Red de Universidades con Carreras en Informática, Buenos Aires Argentina (RedUNCI). http://hdl.handle.net/10915/23754

Este trabajo propone desarrollar un gestor de bases de datos, conteniendo datos no estructurados y que sea capaz de responder las operaciones por similitud más comunes sobre estos tipos de datos, basándonos para ello en: PostgreSQL. Utiliza las consultas por proximidad o similitud en espacios métricos empleando los algoritmos de k-Consulta vecino más cercano y Consulta de rango. Se selecciona a PostgreSQL por su fiabilidad, integridad de datos y correcto desempeño, como así también por su alta portabilidad a los principales sistemas operativos Linux, Unix y Windows. Pero en especial por ofrecer soporte para una amplia variedad de índices, para los que implementa diferentes estructuras de almacenamiento como B-tree, R-tree, Hash o GiST . Para propuesta incorpora las búsquedas por similitud como parte integral del motor PostgreSQL,

proporcionando la funcionalidad y los comandos necesarios para que esto suceda. Para las consultas por similitud básicas: consulta por rango y consulta de k vecinos más cercanos, se indexa la base de datos, dependiendo de la dimensionalidad intrínseca de la misma:

- Con un índice basado en pivotes
- Con un índice basado en particiones compactas.

A la hora de implementar las consultas como el *join* por similitud, uno de los principales aspectos que se tomó en cuenta fue cómo aplicar el criterio de semejanza o distancia en los operadores de Join y Group by. Dependiendo de la situación en la que se encuentren de: un índice para join, en caso de que sea posible preprocesar las bases de datos para indexarlas conjuntamente con el fin de construir el índice, si ambas bases de datos se han indexado separadamente, algoritmos que calculan el join o si ambas bases de datos se han indexado separadamente, algoritmos que calculan el join.

Conclusión: En este trabajo se vuelve a seleccionar el gestor de base de datos PostgreSQL para la realización de esta búsqueda en grandes similitudes, al ser encontrado este dato se toma en cuenta para ser el posible candidato a utilizar en nuestro proyecto ya que se observa que en varios trabajos sobre esta búsqueda lo utilizan por cierta características propias que maneja el, las cuales se mencionan en la síntesis de este mismo artículo.

En el artículo Elaboración de Estrategias Paralelas para Búsquedas por Similitud en Espacios Métricos de: Norma Beatriz Pérez, Mario Berón. (Abril 2013). Elaboración de Estrategias Paralelas para Búsquedas por Similitud en Espacios Métricos (Paginas 1-10). Obtenido de: Red de Universidades con Carreras en Informática (RedUNCI), Buenos Aires Argentina. http://hdl.handle.net/10915/27289

En la paralelización de estructuras de datos para la búsqueda por similitud se clasificaron los algoritmos de indexación en dos grupos uno que se basa en "*clustering"* o particiones compactas y pivotes. Después de analizar y comparar a estos dos grupos con sus respectivos algoritmos se definio al D-Index secuencial como el método más rápido de acceso disponible para resolver búsquedas por similitud debido a que posee las siguientes características:

- Representa uno de los avances más recientes en el escenario de los espacios métricos.
- Une la familia de algoritmos basadas en clustering y pivotes.
- Se basa en: la definición de una función que divide a los objetos y el establecimiento de una jerarquía de buckets que almacenan estos objetos.

En cuanto a los modelos de computación paralela se selecciona a la infraestructura Watershed, que es una estructura que asocia al modelo de Filter-Stream por las siguientes características brindadas: permite el diseño y programación de aplicaciones on-line y off-line distribuidas y posee un mayor poder cómputo al posibilitar resolver consultas en

tiempos aceptables. Para los algoritmos de búsqueda por similitud paralelos se definieron en dos partes, la primera la construcción de índices que es donde se define la construcción de la estructura de los datos que almacena los índices para facilitar la búsqueda, ocupando algoritmos como: Algoritmo Naive_build, Algoritmo Local_build y Algoritmo Global_build. El segundo es un grupo de búsqueda utilizando los siguientes algortimos: Algoritmo Naive_search, Algoritmo Local_search y Algoritmo Global_search.

Conclusión: Se selecciona al algoritmo D-Index secuencial como el método más rápido en el acceso de búsqueda después de ser comparados con aquellos algoritmos basados en "clusterins" o pivotes, en este artículo es la primera vez que da a conocer un algoritmo que sea diferente a los mencionados anteriormente que recaen en estos dos grupos de "clusterins" o pivotes. El artículo presenta también una nueva infraestructura Watershed debido a que permite el diseño y programación de manera on-line y off-line distribuidas, por otro lado en cuanto a su rendimiento de cómputo fue el que brindó un mejor rendimiento de respuestas en cuanto a costo de tiempo.

En el artículo Métodos de acceso por similitud de: Edgar L. Chávez, Norma E. Herrera, Carina M. Ruano, Ana V. Villega. (Abril 2013). Métodos de acceso por similitud (Paginas 59-64). Obtenido de: Red de Universidades con Carreras en Informática (RedUNCI), Buenos Aires Argentina. http://hdl.handle.net/10915/21225

De la familia de estructuras FQ se emplea para este trabajo la FQtrie, logrando una implementación eficiente no solo en términos de la cantidad de evaluaciones de la

función distancia sino también en tiempo extra de CPU. Se seleccionan pivotes para la búsqueda que permitan filtrán una mayor cantidad de objetos, mientras mayor sea esta cantidad nos convertirá en menor la cantidad de evaluaciones de la función que se deba realizar. Como resultado de la selección de estos pivotes se logra indexar un determinado espacio métrico de manera más amplia. Estos pivotes se seleccionan durante la búsqueda, que sean efectivos para el *"query"*. Se utilizan tablas lookup para mejorar el desempeño en CPU y del *"scan*" secuencial. Al manejar la estructura FQtrie para la implementación en espacios métricos se modifica para que sea eficiente en su manejo de disco, esto se logra particionado el espacio métrico a manera que el índice de cada una de las partes entren en memoria principal, para que la búsqueda se resuelva separadamente en cada uno de estos índices, tal operación se hará en memoria principal y paralelo. Al particionar se logra la posibilidad de que los elementos a trabajar puedan ser de mayor tamaño de una página de disco, trabajando más adelante con apuntadores a los objetos reales.

Conclusión: Se utilizaron pivotes en determinadas cantidad de bits como la mejor opción para poder determinar los espacios métricos en los que se trabajará, esto permite establecer una relación entre el pivote y el espacio que se le debe asignar al mismo. Esta selección de pivotes se realiza con la estructura FQtrie.

En el artículo Desarrollo de una herramienta para el análisis y representación semántica de colecciones documentales a través del indicador de Frecuencia de término - Frecuencia inversa del documento (TF-IDF) de: Vuotto, Andrés y Fernández, Gladys Vanesa. (18 de Enero del 2013). Desarrollo de una herramienta para el análisis y

representación semántica de colecciones documentales a través del factor TF-IDF (Paginas 1-9). Obtenido de: Universidad Nacional de Mar del Plata. Facultad de Humanidades. Departamento de Ciencia de la Información; Argentina.. http://humadoc.mdp.edu.ar:8080/xmlui/handle/123456789/631

El trabajo utiliza el repositorio institucional de la Facultad de Humanidades de la UNMdP sobre el cual se desarrollara la aplicación, estructurando este trabajo sobre la representación y visualización de las distintas colecciones del RI por medio de palabras clave que aportan los autores en el proceso de autoarchivo de sus producciones académicas, aprovechando la riqueza y fidelidad (en términos de representación semántica) del lenguaje libre que aportado en la descripción de los objetos digitales generando representaciones apartir del indicador Frecuencia de término - Frecuencia inversa del documento (TF-IDF). Se aplicó una representación semántica de las colecciones implicadas, para identificar el peso de cada palabra clave en la relación a la comunidad de términos en que participa. El TF-IDF permite la construcción de modelos de recuperación avanzados de tipo vectorial, dando como resultado una matriz de datos con tantas columnas como términos y tantas filas como documentos. Este cálculo de representación vectorial ocupa dos sub-factores los cuales son: el factor del termino de frecuencia TF (calcula la capacidad de representación del termino en un documento o colección a través de la obtención de su frecuencia de aparición) cuya fórmula es $Tf(n)=\sum D1(n)$ y el factor IDF de cada termino de colección (calcula la capacidad discriminatoria del termino respecto a la colección) con la siguiente fórmula del indicador de frecuencia $IDF(n)= \log10\ N/DF(n)+1$. Se utiliza la herramienta DSpace que

es el sistema de información con arquitectura de repositorio digital que captura, almacena, ordena y preserva el material de investigación digital de Facultad de Humanidades de la Universidad Nacional de Mar del Plata. Esta herramienta está desarrollada bajo la filosofía del opensource por lo cual es gratuita y se puede personalizar según las necesidades. Su base de datos seleccionada fue PostgreSQL para almacenar las palabras clave, debido a que su esquema de trabajo de este proyecto se basa en primero la extracción de palabras clave en un lenguaje libre que estas depositadas en el repositorio por medio del proceso de autoarchivo. Se desarrolló un algoritmo propio con la siguiente secuencia de subrutinas:

- Acopio de listado de palabras claves y todos los demás metadatos necesarios
- Procesamiento y tratamiento previo de los textos
- Cálculo de un valor tf-idf para cada palabra incluida
- Determinación de los diferentes umbrales de presentación, generando un núcleo de las palabras claves con mayor valor TF-IDF y por consiguiente participantes del gráfico resultante.
- Construcción de las imágenes del dominio consultado.

Como lenguaje de programación utilizaron PHP. Como gestores de bases de datos se utilizó el nativo del sistema Dspace, postgreSQL; y se reforzó el aspecto de gestión de datos con MySQL. La interface se trabajó en HTML5, en combinación con CSS3 y el uso

de la herramienta Google Chart para la confección de grafos que trabajen la tecnología SVG para el desarrollo de espacios vectoriales en entornos web.

Conclusión: En este artículo se desarrolló de manera propia un algoritmo con rutinas propias que se describieron anteriormente, lo único que toma ya establecido es la función para los cálculos de los valores TF-IDF. Otro dato relevante para mi trabajo a realizar es que se detecta por tercera vez en estos artículos que toman como gestos de base de datos a PostgreSQL reforzado con la gestión de datos de MySQL. Al observar que varios autores trabajan con PostgreSQL me está indicando que posee las características necesarias para aplicar las búsquedas por similitud en documentos electrónicos y podría ser el mejor candidato a seleccionar en mi trabajo, por eso se tendría que investigar más a fondo para ampliar el panorama que nos ofrece y conocer que limitantes nos encontramos.

2.2 Lenguajes de programación

Los lenguajes de programación son idiomas artificiales diseñados para expresar cálculos y procesos que serán llevados a cabo por ordenadores. Un lenguaje de programación está formado por un conjunto de palabras reservadas, símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. El proceso de programación consiste en la escritura, compilación y verificación del código fuente de un programa.

2.2.1 Java

Java es un lenguaje de programación desarrollado por Sun Microsystems. Java es un lenguaje muy valorado porque los programas Java se pueden ejecutar en diversas plataformas con sistemas operativos como Windows, Mac OS, Linux o Solaris. Con Java se buscó diseñar un lenguaje que permitiera programar una aplicación una sola vez que luego pudiera ejecutarse en distintas máquinas y sistemas operativos. Para conseguir la portabilidad de los programas Java se utiliza un entorno de ejecución para los programas compilados. Este entorno se denomina Java Runtime Environment (JRE). Es gratuito y está disponible para los principales sistemas operativos. Esto asegura que el mismo programa Java pueda ejecutarse en Windows, Mac OS, Linux o Solaris. Los programas Java son portables, es decir, independientes de la plataforma, porque pueden ejecutarse en cualquier ordenador o dispositivo móvil, independientemente del sistema operativo que tengan instalado: Un programa Java puede ejecutarse en un ordenador de mesa, un ordenador portátil, una tableta, un teléfono, un reproductor de música o en cualquier otro dispositivo móvil con cualquier sistema operativo. (Martínez, p.2)

2.2.1.1 La plataforma de java

Los programas Java se compilan a un lenguaje intermedio, denominado Bytecode. Este código es interpretado por la máquina virtual de Java del entorno de ejecución (JRE)

y así se consigue la portabilidad en distintas plataformas. El JRE es una pieza intermedia entre el código Bytecode y los distintos sistemas operativos existentes en el mercado. Un programa Java compilado en Bytecode se puede ejecutar en sistemas operativos como Windows, Linux, Mac Os, Solaris, BlackBerry OS, iOs o Android utilizando el entorno de ejecución de Java (JRE) apropiado. Una de las características más importantes de los lenguajes de programación modernos es la portabilidad. La plataforma de desarrollo de Java, denominada Java Development Kit (JDK), se ha ido ampliando y cada vez incorpora a un número mayor de programadores en todo el mundo. Java es un lenguaje, una plataforma de desarrollo, un entorno de ejecución y un conjunto de librerías para desarrollo de programas sofisticados. Las librerías para desarrollo se denominan Java Application Programming Interface (Java API).

2.2.1.2 Entornos de desarrollo para java

Existen distintos entornos de desarrollo de aplicaciones Java. Este tipo de productos ofrecen al programador un entorno de trabajo integrado para facilitar el proceso completo de desarrollo de aplicaciones, desde el diseño, la programación, la documentación y la verificación de los programas. Estos productos se denominan Integrated Development Environment (IDE). Existen entornos de distribución libre como: NetBeans, Eclipse o BlueJ. Entre los productos comerciales están JBuilder o JCreatorPro. Para utilizar un entorno de desarrollo es necesario instalar el Java Runtime Environment (JRE) apropiado para el sistema operativo. (Guevara, págs. 1-9)

2.2.2 C#

C# es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET La sintaxis y estructuración de C# es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son comparables con los de Visual Basic.

2.2.2.1 Características de C#

A continuación, se mencionan de manera resumida las características principales del lenguaje de programación C# que son propias de la plataforma. NET:

Modernidad: Incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits, la inclusión de una instrucción como “foreach” que permita recorrer colecciones con facilidad y es ampliable

a tipos definidos por el usuario, la inclusión de un tipo básico “string” para representar cadenas o la distinción de un tipo “bool” específico para representar valores lógicos.

Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del Common Type System (CTS) que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# no admiten ni funciones ni variables globales, sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. (Guevara, págs. 1-9)

Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros). (Guevara, págs. 1-9)

Gestión automática de memoria: Como ya se mencionó anteriormente todo lenguaje de .NET tiene a su disposición un recolector de elementos no utilizados conocido como Common Language Runtime (CLR). Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos.

Instrucciones seguras: Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de asignación (=); y todo caso de un "switch" ha de terminar en un "break" o "goto" que indique cuál es la siguiente acción que realizará, lo que evita la ejecución accidental de casos y facilita su reordenación.

Sistema de tipos unificado: A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada System.Object, por lo que dispondrán de todos los miembros definidos en esta clase, es decir, serán objetos.

Extensibilidad de tipos básicos: C# permite definir, a través de estructuras, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro "ref".

Extensibilidad de operadores: Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia

de Java, C# permite redefinir el significado de la mayoría de los operadores -incluidos los de conversión, tanto para conversiones implícitas como explícitas- cuando se apliquen a diferentes tipos de objetos. (Seco, 2002, págs. 22-27)

2.2.3 PHYTON

Python es un lenguaje de programación creado por Guido Van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses "Monty Python". Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio la primera vez que se ejecuta, generando archivos versión ya compilada (. pyc) o archivo ya optimizado (.pyo), que son los que se ejecutarán en sucesivas ocasiones.

El intérprete de Python está disponible en multitud de plataformas como UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS. por lo que si no utilizamos librerías específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios. La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en Python parecen pseudocódigo. Por este motivo se trata además de uno de los mejores lenguajes para comenzar a programar. Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y

todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles. (Duque, págs. 7-8)

2.2.4 Lenguaje C

El lenguaje C es un lenguaje para programadores en el sentido de que proporciona una gran flexibilidad de programación y una muy baja comprobación de incorrecciones, de forma que el lenguaje deja bajo la responsabilidad del programador acciones que otros lenguajes realizan por sí mismos. C no comprueba que el índice de referencia de un vector no sobrepase el tamaño de este; que no se escriba en zonas de memoria que no pertenecen al área de datos del programa. El lenguaje C es un lenguaje estructurado, en el mismo sentido que lo son otros lenguajes de programación tales como el lenguaje Pascal, el Ada o el Modula-2, pero no es estructurado por bloques, o sea, no es posible declarar subrutinas dentro de otras subrutinas, a diferencia de como sucede con otro lenguaje estructurado como Pascal, ALGOL, PL/I y Ada. Además, el lenguaje C no es rígido en la comprobación de tipos de datos, permitiendo fácilmente la conversión entre diferentes tipos de datos y la asignación entre tipos de datos diferentes.

Todo programa de C consta, básicamente, de un conjunto de funciones, y una función llamada "*main"*, la cual es la primera que se ejecuta al comenzar el programa, llamándose desde ella al resto de funciones que compongan nuestro programa. Desde su creación, surgieron distintas versiones de C, que incluían unas u otras características,

palabras reservadas, etc. Este hecho provocó la necesidad de unificar el lenguaje C, y es por ello por lo que surgió un standard de C, llamado ANSI-C, que declara una serie de características por ejemplo acceso a memoria de bajo nivel, núcleo de lenguaje simple, conjunto reducido de palabras claves, punteros a funciones y variables estáticas , que debe cumplir todo lenguaje C. Por ello, y dado que todo programa que se desarrolle siguiendo el estándar ANSI de C será fácilmente portable de un modelo de ordenador a otro modelo de ordenador, y de igual forma de un modelo de compilador a otro, en estos apuntes explicaremos un C basado en el estándar ANSI-C.

C es un lenguaje sensible al contexto, debido a que diferencia entre mayúsculas y minúsculas, y, por tanto, diferencia entre una palabra escrita total o parcialmente en mayúsculas y otra escrita completamente en minúsculas. En el lenguaje C, un identificador es cualquier palabra no reservada que comience por una letra o por un subrayado, pudiendo contener en su interior letras, números y subrayados. La longitud máxima de un identificador depende del compilador que se esté usando, pero, generalmente, suelen ser de 32 caracteres, ignorándose todos aquellos caracteres que compongan el identificador y sobrepasen la longitud máxima. El lenguaje C suele ser sensible al contexto, un identificador escrito como "*esto_es_un_ident*" y otra vez como "*Esto_Es_Un_Ident*" será interpretado como dos identificadores completamente distintos. (Esteban, 2018)

2.3 Sistemas gestores de base de datos

A medida que se fueron introduciendo las líneas de comunicación, los terminales y los discos, se fueron escribiendo programas que permitían a varios usuarios consultar los mismos ficheros en línea y de forma simultánea. Más adelante fue surgiendo la necesidad de hacer las actualizaciones también en línea. A medida que se integraban las aplicaciones, se tuvieron que interrelacionar sus ficheros y fue necesario eliminar la redundancia. El nuevo conjunto de ficheros se debía diseñar de modo que estuviesen interrelacionados; al mismo tiempo, las informaciones redundantes, que figuraban en los ficheros de más de una de las aplicaciones, debían estar ahora en un solo lugar. Estos conjuntos de ficheros interrelacionados, con estructuras complejas y compartidos por varios procesos de forma simultánea, recibieron al principio el nombre de Bando de Datos, y después, a inicios de los años setenta, el de Base de Datos (BD). El programa de gestión de ficheros era demasiado elemental para dar satisfacción a todas estas necesidades. (Abraham Silberschatz, 2002, págs. 1-21)

El tratamiento de las interrelaciones no estaba previsto, no era posible que varios usuarios actualizaran datos simultáneamente. Debido a la utilización de estos conjuntos de ficheros en los programas que se utilizaban las aplicaciones eran excesivamente complejas, por ello a mediados de los años setenta salieron en el mercado programas más sofisticado-conocidos como sistemas gestores de base de datos (SGBD).

Un SGBD consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una

base de datos de manera que sea tanto práctica como eficiente. Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos. (Abraham Silberschatz, 2002, págs. 1-21)

2.3.1 Tendencias actuales

Los SGDB relacionales están en plena transformación para adaptarse a tres tecnologías de éxito reciente, fuertemente relacionadas: la multimedia, la orientación a objetos e internet y la web. A lo largo de los años que han trabajado con BD de distintas aplicaciones, las empresas han ido acumulando gran cantidad de datos de todo tipo. Si estos datos se analizan convenientemente pueden dar información valiosa. Por lo tanto, se trata de mantener una gran BD con información proveniente de toda clase de aplicaciones de la empresa e, incluso, de fuera. Los datos de este gran almacén se obtienen por una replicación más o menos elaborada de las que hay en las BD que se utilizan en el trabajo cotidiano de la empresa.

2.3.2 Objetivos y servicios de los SGBD

Los SGBD que actualmente están en el mercado pretenden satisfacer un conjunto de objetivos directamente deducibles de lo que hemos explicado hasta ahora.

Consultas no predefinidas y complejas: Los usuarios podrán hacer consultas de cualquier tipo y complejidad directamente al SGBD. El SGBD tendrá que responder inmediatamente sin que estas consultas estén preestablecidas; es decir, sin que se tenga que escribir, compilar y ejecutar un programa específico para cada consulta.

Flexibilidad e independencia: La complejidad de las BD y la necesidad de irlas adaptando a la evolución del Sistema de información (SI) hacen que un objetivo básico de los SGBD sea dar flexibilidad a los cambios. Interesa obtener la máxima independencia posible entre los datos y los procesos usuarios para que se pueda llevar a cabo todo tipo de cambios tecnológicos y variaciones en la descripción de la BD, sin que se deban modificar los programas de aplicación ya escritos ni cambiar la forma de escribir las consultas o actualizaciones directas. (Abraham Silberschatz, 2002, págs. 1-21)

Problemas de la redundancia: En el mundo de los ficheros tradicionales, cada aplicación utilizaba su fichero. Sin embargo, puesto que se daba mucha coincidencia de datos entre aplicaciones, se producía también mucha redundancia entre los ficheros. Ya hemos dicho que uno de los objetivos de los SGBD es facilitar la eliminación de la redundancia.

Integridad de datos: Nos interesará que los SGBD aseguren el mantenimiento de la calidad de los datos en cualquier circunstancia. Cuando el SGBD detecte que un programa quiere hacer una operación que va contra las reglas establecidas al definir la BD, no se lo deberá permitir, y le tendrá que devolver un estado de error. Al diseñar una BD para un SI concreto y escribir su esquema, no sólo definiremos los datos, sino también las reglas de integridad que queremos que el SGBD haga cumplir. Aparte de las reglas de integridad que el diseñador de la BD puede definir y que el SGBD entenderá y hará cumplir, el mismo SGBD tiene reglas de integridad inherentes al modelo de datos que utiliza y que siempre se cumplirán. Son las denominadas reglas de integridad del modelo. Las reglas definibles por parte del usuario son las reglas de integridad del usuario.

Concurrencia de usuarios: Un objetivo fundamental de los SGBD es permitir que varios usuarios puedan acceder concurrentemente a la mismo BD. Cuando un usuario o más de uno están actualizando los datos, se pueden producir problemas de interferencia que tengan como consecuencia la obtención de datos erróneos y la pérdida de integridad de la BD. Para tratar los accesos concurrentes, los SGBD utilizan el concepto de transacción de BD, concepto de especial utilidad para todo aquello que hace referencia a la integridad de los datos. (Abraham Silberschatz, 2002, págs. 1-21)

Seguridad: Actualmente, en el campo de los SGBD, el término seguridad se suele utilizar para hacer referencia a los temas relativos a la confidencialidad, las autorizaciones, los derechos de acceso. Los SGBD permiten definir autorizaciones o derechos de acceso a diferentes niveles: al nivel global de toda la BD, al nivel entidad y al

nivel atributo. Estos mecanismos de seguridad requieren que el usuario se pueda identificar. Se acostumbra a utilizar códigos de usuarios (y grupos de usuarios) acompañados de contraseñas, pero también se utilizan tarjetas magnéticas, identificación por reconocimiento de la voz. (Abraham Silberschatz, 2002, págs. 1-21)

2.3.1 MYSQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. Está desarrollado en su mayor parte en ANSI C y C++. Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una interfaz de programación de aplicaciones específica.

2.3.1.1 Características

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello,

atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad. Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre.

Búsqueda e indexación de campos de texto: MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, se necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central

en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos: Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta: Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y

ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones.

Las siguientes características son implementadas únicamente por MySQL:

- Permite escoger entre múltiples motores de almacenamiento para cada tabla. En MySQL 5.0 éstos debían añadirse en tiempo de compilación, a partir de MySQL 5.1 se pueden añadir dinámicamente en tiempo de ejecución:
  - Los hay nativos como MyISAM, Falcon, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example
  - Desarrollados por compañeros como solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (antes Brighthouse), Kickfire, XtraDB, IBM DB2. InnoDB estuvo desarrollado así pero ahora pertenece también a Oracle.
  - Desarrollados por la comunidad como memcache, httpd, PBXT y Revision.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo. (MySQL, 2018)

2.3.2 POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona,

sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PostgreSQL Global Development Group (PGDG).

Algunas de sus principales características

Alta concurrencia: Mediante un sistema denominado Acceso concurrente multiversión (MVCC), PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente.

Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas con una variedad de funciones asociadas.
- Direcciones de Protocolo de internet (IP) en las versiones IPv4 e IPv6.
- Bloques de direcciones estilo Classless Inter-Domain Routing (CIDR).
- Direcciones Media Access Control (MAC).

"Arrays": Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL.

Algunos de los lenguajes que se pueden usar son los siguientes:

- Un lenguaje propio llamado PL/PgSQL similar al PL/SQL de oracle.
- C.
- C++.
- Java PL/Java web.
- PL/Perl.
- plPHP.
- PL/Python.
- PL/Ruby.
- PL/sh.
- PL/Tcl. (PostgreSQL, 2018)

2.3.3 Microsoft SQL Server

Microsoft SQL Server es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado por la línea de comandos o mediante la interfaz gráfica de "*Management Studio*" es Transact-SQL (TSQL). Dentro de los competidores más destacados de SQL Server están: Oracle, MariaDB, MySQL, PostgreSQL. SQL Server ha estado tradicionalmente disponible solo para sistemas operativos Windows de Microsoft, pero desde 2017 también está disponible para Linux y Docker containers.34 Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor.

2.3.3.1 Características

Este sistema incluye una versión reducida, llamada Motor de escritorio de Microsoft (MSDE) con el mismo motor de base de datos, pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita. Es común desarrollar proyectos completos empleando Microsoft SQL Server y Microsoft Access a través de los llamados Proyecto de acceso a datos (ADP). De esta forma se completa la base de datos Microsoft SQL Server, con el entorno de desarrollo, a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows. Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos.

2.3.3.2 Programación

Su lenguaje de cómputo estandarizado (T-SQL) es el principal medio de interacción con el Servidor, el cual permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de base de datos, inserción y modificación de datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias en T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente. Cliente Nativo de SQL, es la biblioteca de acceso a datos para los clientes de Microsoft

SQL Server versión 2005 en adelante. Implementa de forma nativa soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos. Cliente Nativo de SQL se utiliza como extensión de SQL Server "*plug-ins"* para otras tecnologías de acceso de datos, incluyendo ADO u OLE DB. Cliente Nativo de SQL puede también usarse directamente, pasando por alto las capas de acceso de datos. (Microsoft SQL Server, 2018)

2.3.4 Oracle

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional, el sistema de gestión de bases de datos relacionales de objetos (ORDBMS) fue desarrollado por Oracle Corporation. Su dominio en el mercado de servidores empresariales había sido casi total hasta que recientemente tiene la competencia del Microsoft SQL Server y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

Oracle nace en 1977 bajo el nombre Laboratorio de Desarrollo de Software (SDL). En 1979 SDL cambia su nombre a Relational Software, Inc. (RSI) La función de

SDL Fue motivada principalmente por un estudio sobre los SGBD (Sistemas Gestores de Base de Datos) de George Koch. En la actualidad, Oracle (Nasdaq: ORCL) Todavía encabeza la lista. La tecnología de Oracle se encuentra casi en todas las industrias alrededor del mundo y en las oficinas de 98 de las 100 empresas fortune 100. Oracle es la primera compañía que desarrolla e implementa software para bases de datos cien por ciento activo por internet atreves de su amplia línea de productos: Base de datos, Aplicaciones comerciales y herramientas de desarrollo de aplicaciones además de un soporte de decisiones.

Características. La base de datos de Oracle 10G Standar Edition es compatible con medianas industrias. Esto incluye Real Aplication Clúster, para crear protección en contra de fallos de hardware. Es muy sencillo de instalar y configurar, y viene con su propio software de clustering.

La base de datos Oracle 10g Standar Edition, proporciona una rápida instalación sin contratiempos tanto en un único servidor como en un ambiente de clúster. La base de datos está preconfigurada lista para ser usada en producción, completa con espacio automatizado, administración de almacenamiento y de memoria, Back-up y recuperación automatizada y administrador de estadísticas automatizado.

La consola de Enterprise Manager 10g data base control provee una interface web que te enseña el estado actual de la base de datos y del ambiente del clúster y permite la administración de la base de datos desde cualquier browser conectado a su sistema

La base de datos Oracle Standar Edition toma ventaja también de la solución de clusterware, apartando la complejidad de tener que instalar y configurar clusterware de terceras personas. Los procedimientos almacenados pueden ser escritos en java PL SQL o utilizando .Net CLR Support en Oracle Database 10g Release Dos.

La base de datos Oracle Standar Edition, usa las mismas gestiones de concurrencia que son usadas por la base de datos de Oracle Enterprise Edition, asegurando así el máximo rendimiento para todas las cargas de trabajo. (Oracle Database, 2018)

2.4 Archivos electrónicos

Archivo digital o electrónico son términos de moda, pero por su uso tan diverso no sólo resulta excesivamente genéricos, sino ambiguos. Se emplean indistintamente para indicar la reunión de documentos creados mediante medios informáticos, sea tanto un conjunto de ficheros generados por una aplicación como una colección de documentos textuales e icónicos digitalizados, en muchos casos accesibles a través de internet. Informatización de archivos es un término que se utiliza con mayor precisión para denotar la aplicación de las tecnologías de la información al desarrollo de las actividades administrativas y de gestión documental en los archivos, automatizando el mayor número de procesos y de tareas posibles. Sistemas integrados para la gestión de archivos y

edición electrónica de instrumentos de descripción y de documentos son expresiones que pertenecen al mismo marco conceptual. (Angel, 2018)

2.4.1 Formato doc

Microsoft Word utiliza un formato nativo cerrado y muy utilizado, comúnmente llamado DOC (abreviatura de documento), con extensión de archivo .doc. Por la amplísima difusión del Microsoft Word, este formato se ha convertido en un estándar de facto en el que transferir textos con o sin formato, o hasta imágenes, siendo preferido por muchos usuarios antes que otras opciones como archivos de texto (TXT) para el texto sin formato o Joint Photographic Experts Group (JPG) para gráficos; sin embargo, este formato posee la desventaja de tener un mayor tamaño comparado con algunos otros. Por otro lado, la Organización Internacional para la Estandarización ha elegido el formato de documento abierto (ODF) como estándar para el intercambio de texto con formato, lo cual ha supuesto una desventaja para el formato .doc.

2.4.1.1 Archivos Word

Microsoft Word es un procesador de texto creado por Microsoft, y actualmente integrado en la suite ofimática Microsoft Office. Originalmente desarrollado por Richard Brodie para el ordenador de IBM con el sistema operativo DOS en 1983. Se crearon versiones posteriores Apple Macintosh en 1984 y Microsoft Windows en 1989, siendo

esta última versión la más difundida en la actualidad, llegando a ser el procesador de texto más popular.

2.4.1.2 Formato RTF

El formato Rich Text Format (RTF) surgió como acuerdo para intercambio de datos entre Microsoft y Apple en los tiempos en que Apple dominaba el mercado de los ordenadores personales. Las primeras versiones del formato .doc de word eran derivadas del RTF. Incluso ahora hay programas de Microsoft como el wordpad que usan el RTF como formato nativo. El RTF también tiene extension .doc al igual que los sucesivos formatos de word que se han presentado. Puede ser por tanto considerado un segundo formato nativo.

2.4.1.3 Otros formatos

Word tiene un mecanismo similar al de los “plug-ins” para entender otros formatos. Fue desarrollado en los tiempos en que Word Perfect era el estándar de facto para quitarle cuota de mercado. Se basa en instalar una librería dinámica (DLL) para implementar el formato. Microsoft incluso publicó un "Converter SDK" Software Development Kit para permitir a los usuarios que escribieran soporte para formatos no soportados. (Stallman, 2018)

2.4.2 PDF

El formato de documento portátil (PDF) se utiliza para presentar e intercambiar documentos de forma fiable, independiente del software, el hardware o el sistema operativo. Inventado por Adobe, PDF es ahora un estándar abierto y oficial reconocido por la Organización Internacional para la Estandarización (ISO). Los archivos PDF pueden contener vínculos y botones, campos de formulario, audio, vídeo y lógica empresarial. También se pueden firmar de manera electrónica y se visualizan fácilmente con el software gratuito Acrobat Reader DC. (Archivos PDF, 2018)

2.4.2.1 Características de los archivos PDF

Cuenta con dos versiones: Acrobat Reader disponible para Mac, Windows y Unix, la cual es totalmente gratuita y puede ser descargada a través de la red. Esta versión como su nombre lo indica sirve únicamente para visualizar el contenido de un documento PDF en pantalla. Si lo que se desea es crear un documento PDF, entonces se tiene que adquirir el software completo de Adobe Acrobat. Lo anterior no significa que no se pueda explorar o trabajar con archivos PDF creados en versiones recientes si nuestro ordenador cuenta con algún sistema operativo de los mencionados, ya que bastaría con instalar el Acrobat versión 5 o anterior o la versión Reader para visualizarlos en pantalla.

No requiere de mucho espacio de alojamiento, ya que generalmente los archivos en formato PDF, son divididos en secciones que no rebasan 1MB, de hecho, el peso promedio de un documento es de 200 KB a 400 KB. Esto varía dependiendo del número de páginas que tenga nuestro documento y del tipo de original. Un PDF generado a partir de imágenes o de un documento impreso resulta más pesado que uno creado con alguna aplicación de Microsoft Windows. Este es uno de los motivos por el que este formato es muy utilizado en Internet, ya que el tiempo de espera en la descarga no excede los 15 segundos. Puede ser visualizado, explorado e incluso imprimirse desde cualquier sistema operativo, ya sea Windows 98, Me, Xp, Mac, Unix o Linux.

La seguridad es un aspecto muy importante para Acrobat, ya que a un documento PDF se le pueden establecer permisos de impresión, edición, copiar o inclusive apertura mediante la implementación de una contraseña.

Posee las características de todo documento electrónico como: la posibilidad de incluir hipervínculos, botones, animaciones, formularios, videos o gráficos, lo cual, aplica tanto en archivos creados a partir de una aplicación de Microsoft Windows o Linux para los que fueron generados a partir de un conjunto de imágenes o un documento impreso. (Martínez, 2005)

2.5 Modelo de Desarrollo de Software

Un proceso de software es una serie de actividades relacionadas que conduce a la elaboración de un producto de software. Estas actividades pueden incluir el desarrollo de software desde cero en un lenguaje de programación estándar como Java o C. Existen

diferentes procesos de software los cuales deben contener cuatro actividades fundamentales: especificación del software, diseño e implementación del software, validación del software, evolución del software. Un modelo de proceso de software es una representación simplificada de este proceso. Cada modelo del proceso representa a otro desde una particular perspectiva, ofrece sólo información parcial acerca de dicho proceso. (Kendall, 2011)

2.5.1 Modelo de prototipos

El modelo de prototipos permite que todo el sistema, o algunas de sus partes, se construyan rápidamente para comprender con facilidad y aclarar ciertos aspectos en los que se aseguren que el desarrollador, el usuario, el cliente estén de acuerdo en lo que se necesita así como también la solución que se propone para dicha necesidad y de esta forma minimizar el riesgo y la incertidumbre en el desarrollo. Este modelo se encarga del desarrollo de diseños para que estos sean analizados y prescindir de ellos a medida que se adhieran nuevas especificaciones, es ideal para medir el alcance del producto, pero no se asegura su uso real. Este modelo se encarga principalmente de ayudar al ingeniero de sistemas y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

Ciclo de Vida de un Sistema basado en Prototipo

Un prototipo de pantallas muestra la interfaz de la aplicación, su cara externa, pero dicha interfaz está fija es decir estática, no procesa datos. El prototipo no tiene desarrollada una lógica interna, sólo muestra las pantallas por las que irá pasando la futura aplicación. El prototipo funcional evolutivo desarrolla un comportamiento que satisface los requisitos y necesidades que se han entendido claramente. Realiza un proceso real de datos, para contrastarlo con el usuario. Se va modificando y desarrollando sobre la marcha, según las apreciaciones del cliente. Esto ralentiza el proceso de desarrollo y disminuye la fiabilidad, puesto que el software está constantemente variando, a la larga genera un producto más seguro, en cuanto a la satisfacción de las necesidades del cliente.

Ventajas del Modelo de Prototipo.

Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida. También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

Desventajas del Modelo de Prototipo.

Su principal desventaja es que una vez que el cliente ha dado su aprobación final al prototipo y cree que está a punto de recibir el proyecto final, se encuentra con que es necesario reescribir buena parte del prototipo para hacerlo funcional, porque lo más seguro es que el desarrollador haya hecho compromisos de implementación para hacer que el prototipo funcione rápidamente. Es posible que el prototipo sea muy lento, muy grande, no muy amigable en su uso, o incluso, que esté escrito en un lenguaje de programación inadecuado. En la siguiente Figura 2.2 se muestra el diagrama del modelo de prototipo:

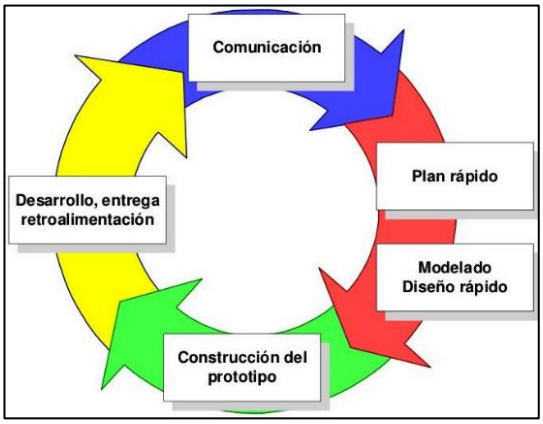


Figura 2.2 Modelo de Prototipo (Kendall, 2011)

2.5.1.1 Modelo en espiral

El modelo espiral es un modelo meta del ciclo de vida del software donde el esfuerzo del desarrollo es iterativo, tan pronto culmina un esfuerzo del desarrollo por ahí mismo comienza otro; en cada ejecución del desarrollo se sigue cuatro pasos principales:

1. Determinar o fijar los objetivos: En este paso se definen los objetivos específicos para posteriormente identifica las limitaciones del proceso y del sistema de software, además se diseña una planificación detallada de gestión y se identifican los riesgos.

2. Análisis del riesgo: En este paso se efectúa un análisis detallado para cada uno de los riesgos identificados del proyecto, se definen los pasos a seguir para reducir los riesgos y luego del análisis de estos riesgos se planean estrategias alternativas.

3. Desarrollar, verificar y validar: En este tercer paso, después del análisis de riesgo, se eligen un paradigma para el desarrollo del sistema de software y se lo desarrolla.

4. Planificar: En este último paso es donde el proyecto se revisa y se toma la decisión si se debe continuar con un ciclo posterior al de la espiral. Si se decide continuar, se desarrollan los planes para la siguiente fase del proyecto.

Con cada iteración alrededor de la espiral, se crean sucesivas versiones del software, cada vez más completas y, al final, el sistema de software ya queda totalmente funcional. Un modelo espiral comienza con la determinación de los objetivos tanto funcionales como de rendimiento. Después se enumeran algunas formas posibles de alcanzar estos objetivos identificando las fuentes de riesgos posibles. Luego se continúa con el siguiente paso que es resolver estos riesgos y llevar a cabo las actividades de desarrollo, para finalizar con la planificación del siguiente ciclo de la espiral

(Sommerville, 2011). A continuación en la Figura 2.3 se muestra el diagrama de las etapas del modelo de espiral:

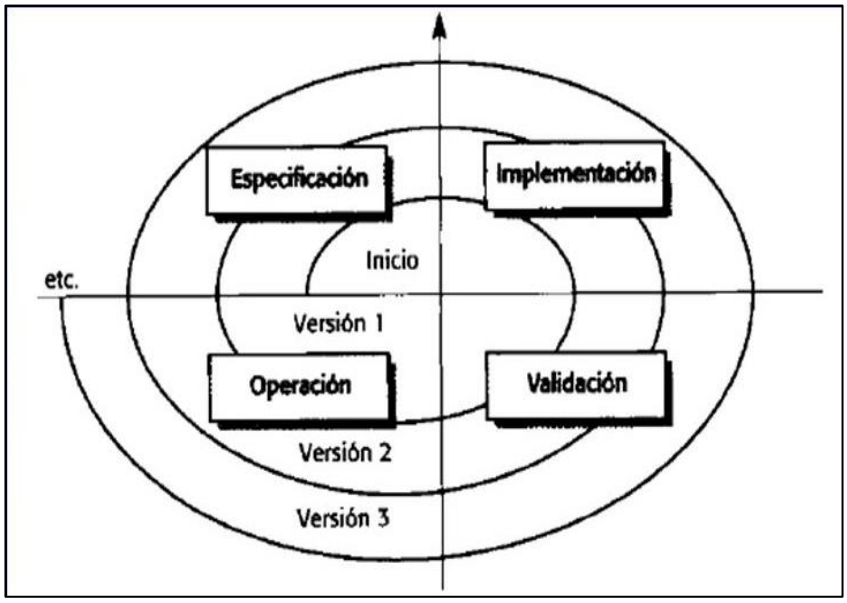


Figura 2.3 Modelo de Espiral (Sommerville, 2011)

2.5.1.2 RUP (Proceso Unificado de Rational)

Proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es garantizar la producción de alta calidad software que satisfaga las necesidades de sus usuarios finales, dentro de un horario predecible y presupuesto.

Características esenciales

Este proceso de software RUP tiene tres características esenciales: Dirigido por los Casos de Uso, Centrado en la arquitectura, y es iterativo e incremental.

1. Proceso dirigido por Casos de Uso: En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba.

2. Proceso centrado en la arquitectura: En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

3. Proceso iterativo e incremental: La estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración, un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales, del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada de etapas.Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas),

también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

4. Estructura Dinámica del proceso: RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable

RUP describe cómo utilizar de forma efectiva procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software, conocidos como “mejores prácticas”. Una parte importante de estas "mejores prácticas" es: la administración de requerimientos, que consiste en definir, organizar y documentar las especificaciones funcionales y sus limitantes, así como las restricciones; dar seguimiento y documentar decisiones y alternativas tomadas; y capturar y comunicar con facilidad los requerimientos del negocio. Las nociones de "casos de uso" y escenarios utilizados en el proceso de desarrollo son una excelente forma para capturar los requerimientos funcionales.

Este es uno de los flujos de trabajo más importantes, porque en él se establece que tiene que hacer exactamente el sistema que se construya. Los requerimientos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requerimientos que se especifiquen.

Los requerimientos se dividen en dos grupos. Los requerimientos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de Casos de Uso. Los requerimientos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad. (Sommerville, 2011)

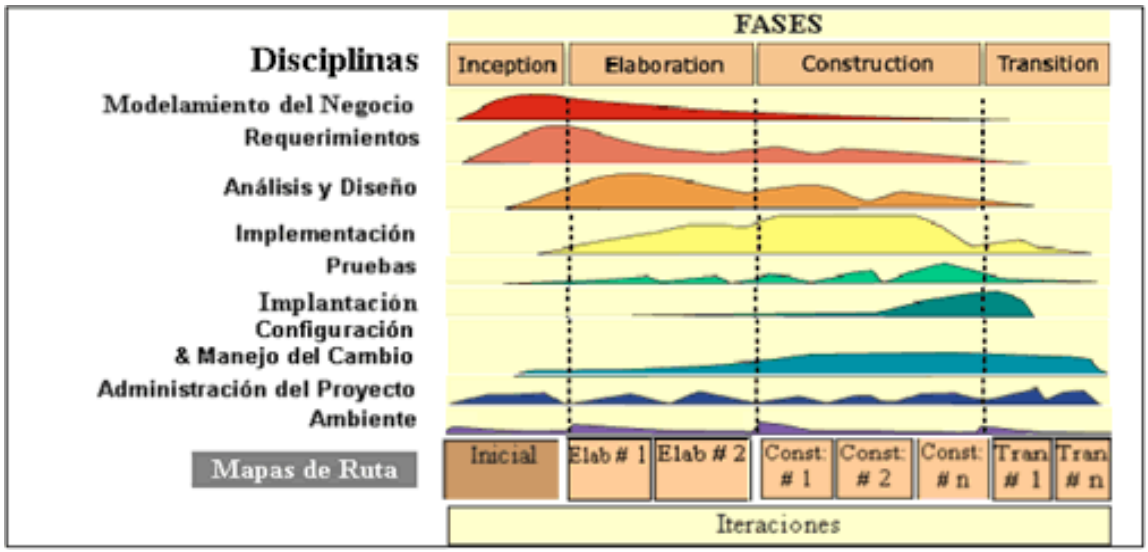


Figura 2.4 Fases RUP (Sommerville, 2011)

2.5.1.3 Modelo en cascada o Clásico (modelo tradicional)

Éste toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y, luego, los representa como fases separadas del proceso, tal como especificación de requerimientos, diseño de software, implementación, pruebas.

Desarrollo incremental: Este enfoque vincula las actividades de especificación, desarrollo y validación. El sistema se desarrolla como una serie de versiones (incrementos), y cada versión añade funcionalidad a la versión anterior. Las principales etapas del modelo en cascada reflejan directamente las actividades fundamentales del desarrollo:

1. Análisis y definición de requerimientos. Los servicios, las restricciones y las metas del sistema se establecen mediante consulta a los usuarios del sistema. Luego, se definen

con detalle y sirven como una especificación del sistema.

2. Diseño del sistema y del software El proceso de diseño de sistemas asigna los requerimientos, para sistemas de hardware o de software, al establecer una arquitectura de sistema global. El diseño del software implica identificar y describir las abstracciones

fundamentales del sistema de software y sus relaciones.

3. Implementación y prueba de unidad Durante esta etapa, el diseño de software se realiza como un conjunto de programas o unidades del programa. La prueba de unidad consiste en verificar que cada unidad cumpla con su especificación.

4. Integración y prueba de sistema Las unidades del programa o los programas individuales se integran y prueban como un sistema completo para asegurarse de que se

cumplan los requerimientos de software. Después de probarlo, se libera el sistema de software al cliente.

5. Operación y mantenimiento, ésta es la fase más larga del ciclo de vida, donde el sistema se instala y se pone en práctica. El mantenimiento incluye corregir los errores que no se detectaron en etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema e incrementar los servicios del sistema conforme se descubren nuevos requerimientos.

En principio, el resultado de cada fase consiste en uno o más documentos que se autorizaron ("firmaron"). La siguiente fase no debe comenzar sino hasta que termine la fase previa. En la práctica, dichas etapas se traslapan y se nutren mutuamente de información. Durante el diseño se identifican los problemas con los requerimientos. En la codificación se descubren problemas de diseño, y así sucesivamente. El proceso de software no es un simple modelo lineal, sino que implica retroalimentación de una fase a otra. Entonces, es posible que los documentos generados en cada fase deban modificarse para reflejar los cambios que se realizan.

Debido a los costos de producción y aprobación de documentos, las iteraciones suelen ser onerosas e implicar un rediseño significativo. Por lo tanto, después de un pequeño número de iteraciones, es normal detener partes del desarrollo, como la especificación, y continuar con etapas de desarrollo posteriores. Los problemas se dejan para una resolución posterior, se ignoran o se programan. (Kendall, 2011)

2.5.1.4 Scrum

La metodología SCRUM, es una metodología ágil y flexible utilizada para la gestión de proyectos. Fue desarrolla por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, al analizar el desarrollo de proyectos de las principales empresas tecnológicas: Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M y Hewlett-Packard. Scrum descompone la organización en pequeños equipos auto-organizados. Cada equipo desarrolla los proyectos en base a entregas parciales *sprints*, con el objetivo de alinear expectativas con el cliente y aumentar el valor que se ofrece a los mismos.

Funcionamiento

El cliente/sponsor o "Product Owner" define los requisitos del sistema a desarrollar «Product Backlog«, siempre bajo la figura de un asistente de supervisión o *Scrum Master*. Se descomponen estos requisitos en varios paquetes de trabajo más manejables *Sprint Backlog*, que puede ir de 2 a 4 semanas de trabajo por paquete, esta descomposición se realiza en una reunión o *Sprint planning meeting* que puede durar hasta 8 horas y donde se define el alcance. El equipo de trabajo auto organizado tiene una reunión diariamente *Daily Scrum* durante unos 15 minutos, en esta reunión cada uno expone que hizo, que va a hacer y que problemas se ha encontrado y se debate entre todos como como realizar las tareas.

Cuando termina un sprint se realiza una reunión o *Sprint Review* donde se presenta el producto resultante del *Sprint Backlog*, también puede realizarse una reunión retrospectiva *Sprint Retrospective* de hasta 3 horas, en la que se evalúan las técnicas y habilidades empleadas para valorar si pueden mejorarse y aplicarse para los siguientes *Sprint*. Repitiéndolo para cada *Sprint Backlog* obtendríamos el producto final como una sucesión de pequeños incrementos. (Calvo, 2015)

Reuniones prescritas

- Sprint planning meeting: Reunión de Planificación de Sprint.
- Daily Scrum: Reunión de seguimiento diaria.
- Sprint Review: Reunión de revisión.
- Sprint Retrospective: Reunión de retrospectiva.

Roles

- Product Owner: cliente o sponsor
- ScrumMaster: supervisor que asiste todo el proceso.
- Miembros del equipo de desarrollo.

2.6 Costos

2.6.1 Visual Studio Community 2017

Un completo IDE extensible y gratuito con todas las características para crear aplicaciones modernas para Windows, Android e iOS, además de aplicaciones web y servicios en la nube.

Novedades de Visual Studio 2017

Este nuevo instalador diseñado desde cero brinda las siguientes ventajas:

- Reducir al mínimo el consumo de memoria de Visual Studio.
- Instalar más rápidamente con menos impacto en el sistema y desinstalar de una forma más limpia.
- Facilitar la selección y la instalación únicamente de las características que necesitas.

Productividad

Las mejoras realizadas en la navegación de código, IntelliSense, la refactorización, las correcciones de código y la depuración permiten ahorrar tiempo y esfuerzo en las tareas cotidianas, independientemente del lenguaje o la plataforma. Para aquellos equipos que adoptan DevOps, Visual Studio 2017 simplifica el bucle interior y

agiliza el flujo de código con características totalmente nuevas en tiempo real, como, por ejemplo, las pruebas unitarias dinámicas y la validación de dependencias de arquitectura en tiempo real.

Aspectos fundamentales renovados

Hay un enfoque renovado para mejorar la eficacia de las tareas esenciales que debes abordar diariamente. Esto incluye una instalación ligera y modular totalmente nueva adaptada a tus necesidades, un IDE más rápido desde el inicio hasta el apagado y una nueva forma de ver, editar y depurar cualquier código sin proyectos ni soluciones. Visual Studio 2017 te ayuda a centrarte en la visión general.

Desarrollo de Azure simplificado

Un conjunto integrado de herramientas de Azure que te permiten crear con facilidad aplicaciones principalmente destinadas a la nube con tecnología de Microsoft Azure. Visual Studio 2017 facilita la configuración, la compilación, la depuración, el empaquetado y la implementación de aplicaciones y servicios en Microsoft Azure directamente desde el IDE.

Desarrollo móvil de alta calidad

Con herramientas avanzadas de depuración y generación de perfiles, y características de generación de pruebas unitarias, Visual Studio 2017 con Xamarin agiliza y simplifica más que nunca los procesos de compilación, conexión y ajuste de las aplicaciones móviles para Android, iOS y Windows. También puedes optar por

desarrollar aplicaciones móviles mediante el desarrollo de bibliotecas multiplataforma de Apache Cordova o Visual C++ en Visual Studio.

Costo de Visual Studio Community 2017: Gratuito

2.6.2 PostgreSQL

Es un potente sistema de base de datos relacional de objetos abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. PostgreSQL viene con muchas características destinadas a ayudar a los desarrolladores para crear aplicaciones, administradores para proteger la integridad de los datos y crear entornos tolerantes a fallas, y ayudarlo a administrar sus datos sin importar cuán grande o pequeño sea el conjunto de datos. Además de ser de código abierto y gratuito, PostgreSQL es altamente extensible. Por ejemplo, puede definir sus propios tipos de datos, desarrollar funciones personalizadas e incluso escribir códigos de diferentes lenguajes de programación sin recompilar su base de datos.

PostgreSQL intenta cumplir con el estándar SQL donde dicha conformidad no contradice las características tradicionales o podría llevar a decisiones arquitectónicas deficientes. Muchas de las funciones requeridas por el estándar SQL son compatibles, aunque a veces con una sintaxis o función ligeramente diferente. Se pueden esperar más movimientos hacia la conformidad a lo largo del tiempo. A partir del lanzamiento de la versión 10 en octubre de 2017, PostgreSQL cumple con al menos 160 de las 179

características obligatorias para SQL: conformidad con el estándar 2011, donde a partir de este momento, ninguna base de datos relacional cumple totalmente con este estándar.

PostgreSQL se lanza bajo la licencia PostgreSQL, una licencia liberal de código abierto, similar a las licencias BSD o MIT, Sistema de gestión de bases de datos PostgreSQL . Se otorga permiso para usar, copiar, modificar y distribuir este software y su documentación para cualquier propósito, sin cargo, y sin un acuerdo por escrito, siempre que el aviso de copyright anterior y este párrafo y los dos párrafos siguientes aparezcan en todas las copias.

PostgreSQL Versión 10.5: Gratuito

Capítulo 3 Metodología

Se seleccionó la metodología de desarrollo ágil *Scrum* debido a que tiene como base la creación de ciclos breves en el desarrollo conocidos como iteraciones y que en *Scrum* se llamaran *Sprints*. Debido a la creación de estos ciclos en el proyecto podemos delimitar el tiempo que llevara cada una de las fases del proyecto y poder hacer una planeación correcta del tiempo respecto al desarrollo correcto del sistema. Dentro de estos ciclos esta la creación de los mini ciclos en el ciclo principal que nos sirven de apoyo para la revisión diaria de estas iteraciones e ir trabajando de forma correcta en el desarrollo de cada etapa. El flujo de trabajo de este modelo se muestra en la Figura 3.1.

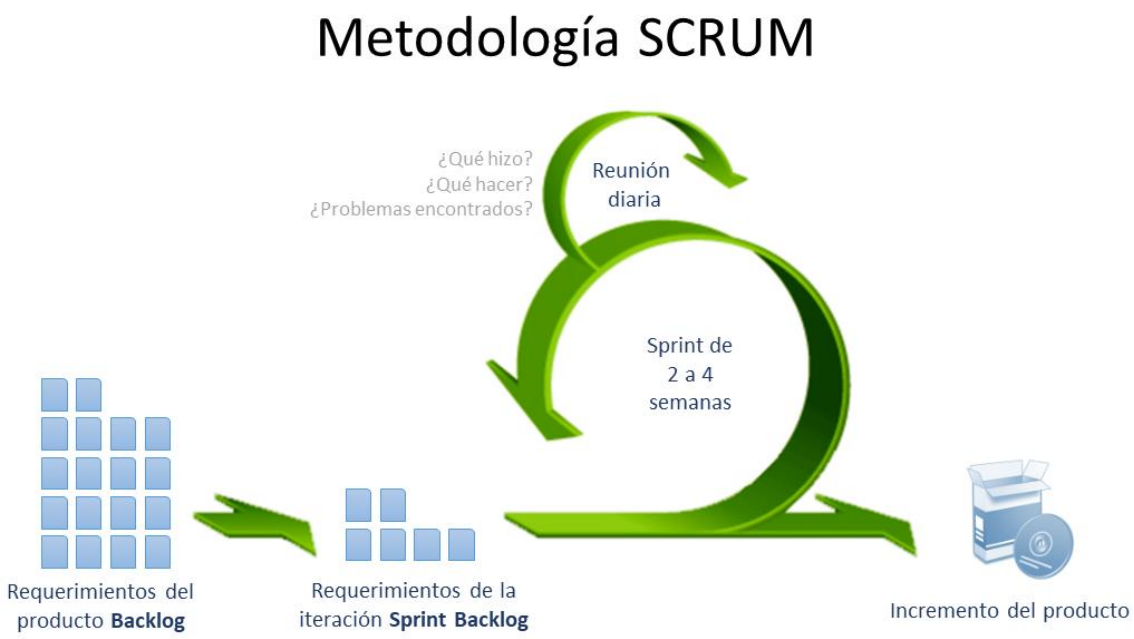


Figura 3. 1 Modelo de Scrum (Calvo, 2015)

3.1 Definición de los requerimientos

En esta sección se muestra el levantamiento de requerimientos y su respectivo análisis de cada uno.

3.1.1 Características de los usuarios

Es importante mencionar qué tipo de usuarios serán los que van a manipular el sistema y qué perfil a fin deben tener.

Tipo de usuario	Coordinador
Formación	Licenciaturas a fin
Actividades	Facilitar el proceso de búsqueda en la recepción del nuevo trabajo recibido en la coordinación de titulación de la División de Estudios Profesionales.

Tipo de usuario	Jefe del departamento
Formación	Licenciatura a fin
Actividades	Facilitar el proceso de búsqueda en la recepción del nuevo trabajo recibido en la coordinación de titulación de la División de Estudios Profesionales en caso de que no esté el titular de esta área.

3.1.2 Requerimientos Funcionales

En los requerimientos funcionales encontraremos las funciones que definirán al sistema. A continuación de enumera cada uno con un identificador único por ejemplo Requerimiento Funcional uno (RF01).

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Autentificación de Usuario.
Características:	Los usuarios deberán identificarse para acceder a la página principal del sistema.
Descripción del requerimiento:	El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF04 • RNF05
Prioridad del requerimiento: Alta	
Identific	RF02

ación del requerimiento:	
Nombre del Requerimiento:	Registrar Usuarios.
Características:	Los usuarios deberán registrarse en el sistema para acceder al sistema y dependiendo sus privilegios son las acciones que podrá realizar.
Descripción del requerimiento:	El sistema permitirá al usuario (administrador o coordinador) registrarse. El usuario debe suministrar datos como: Usuario y Password.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF04 • RNF05
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF03

Nombre del Requerimiento:	Consultar Información de archivos cargados
Características:	El sistema ofrecerá al usuario información general acerca de los archivos cargados en la base de datos de los trabajos de titulación.
Descripción del requerimiento:	**<u>Consultar archivos cargados:</u>** Muestra información general sobre los trabajos, por ejemplo: ID del archivo, autor, año, carrera, número de control estudiante y sinodales
Requerimiento NO funcional:	• RNF01 • RNF02
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Consultar Información.
Caracte	El sistema ofrecerá al usuario el porcentaje de similitud

rísticas:	arrojado y a qué documento pertenece.
Descripción del requerimiento:	**<u>Consultar los filtros de los trabajos:</u>** Muestra a los usuarios el porcentaje de similitud del filtro seleccionado con respecto a la búsqueda que realizó en el sistema.
Requerimiento NO funcional:	• RNF01 • RNF02
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Modificar.
Características:	El sistema permitirá al administrador dar de alta o modificar privilegios del usuario.
Descripción del requerimiento:	Permite al administrador modificar datos de los usuarios con sus cuentas creadas.

Requerimiento NO funcional:	• RNF01 • RNF02 • RNF05
Prioridad del requerimiento: Alta	

3.1.3 Requerimientos No Funcionales.

Los requerimientos no funcionales son aquellos requerimientos que nos mostrarán las características generales del sistema. A continuación se enumera cada uno con un identificador único por ejemplo Requerimiento No Funcional uno (RNF01).

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentara una interfaz de usuario sencilla y amigable, para que sea de fácil manejo a los usuarios del sistema.
Descripc	El sistema debe tener una interfaz de uso intuitiva y

ión del requerimiento:	sencilla.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Ayuda en el uso del sistema.
Características:	La interfaz del usuario deberá de presentar un sistema de ayuda para que a los mismos usuarios del sistema se les facilite el trabajo en cuanto al manejo del sistema. Deberá estar ubicado en una sección de ayuda dentro de su menú.
Descripción del requerimiento:	La interfaz debe estar complementada con un sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas y esto le podrá servir como un manual de usuario).
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Mantenimiento.
Características:	El sistema deberá de tener un manual de instalación y manual de usuario para facilitar los mantenimientos que serán realizados por el administrador.
Descripción del requerimiento:	El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible.
Prioridad del requerimiento: Alta	
Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Desempeño

Características:	El sistema garantizará a los usuarios un desempeño en cuanto a los datos almacenado en el sistema ofreciéndole una confiabilidad a ésta misma.
Descripción del requerimiento:	Garantizar el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF05
Nombre del Requerimiento:	Seguridad en información
Características:	El sistema garantizará a los usuarios una seguridad en cuanto a la información que se procese en el sistema.
Descripción del	Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales como documentos,

requerimiento:	archivos y/o contraseñas.
Prioridad del requerimiento: Alta	

3.1.4 Requerimientos del Software

- PC de desarrollo
- ✓ Sistema Operativo Windows 10
- ✓ Enterprise Archited 8.0
- ✓ SQL Server
- ✓ Visual Studio 2018
- ✓ Adobe Reader versión X

3.1.5 Requerimientos del Hardware

- PC de desarrollo
- ✓ Procesador i5
- ✓ 8 Gigas de Ram

3.2 Análisis y diseño

La documentación mostrada en este apartado muestra los diferentes diagramas diseñados para cada uno de los procesos relevantes del proceso y del sistema. En casa uno se da una breve descripción de ellos.

3.2.1 Modelado de negocios

La descripción de los procesos del modelado de negocio para el proceso de titulación con el diagrama explotado de búsqueda mostrado más adelante son los siguientes:

- **Concluir sus créditos:** El alumno concluye el total de los créditos en su avance reticular, incluyendo su servicio social y residencia profesional.

Para integrar el expediente de titulación el alumno también debe liberar la lengua extranjera. Entrega a servicios escolares la documentación solicitada para abrir su expediente de titulación. Lleva una copia del expediente recibido junto con su propuesta de titulación en 4 tantos a la división de estudios profesionales.

- **Entregar expediente de titulación:** El alumno prepara la documentación solicitada por el departamento de servicios escolares en original y dos copias. El encargado da apertura al expediente de titulación coteja los documentos y prepara un juego para el departamento de servicios escolares y genera un expediente de recibido para el alumno.

- **Recibir propuesta de titulación:** El alumno entrega a la división de estudios profesionales la copia del expediente de titulación emitido por el departamento de servicios escolares junto con sus 4 engargolados de la propuesta de titulación. Una vez que el trabajo se detecta que no tiene similitud con otro anterior es aceptado por la

división de estudios y genera un oficio para el departamento académico correspondiente para que le asignen los revisores del trabajo.

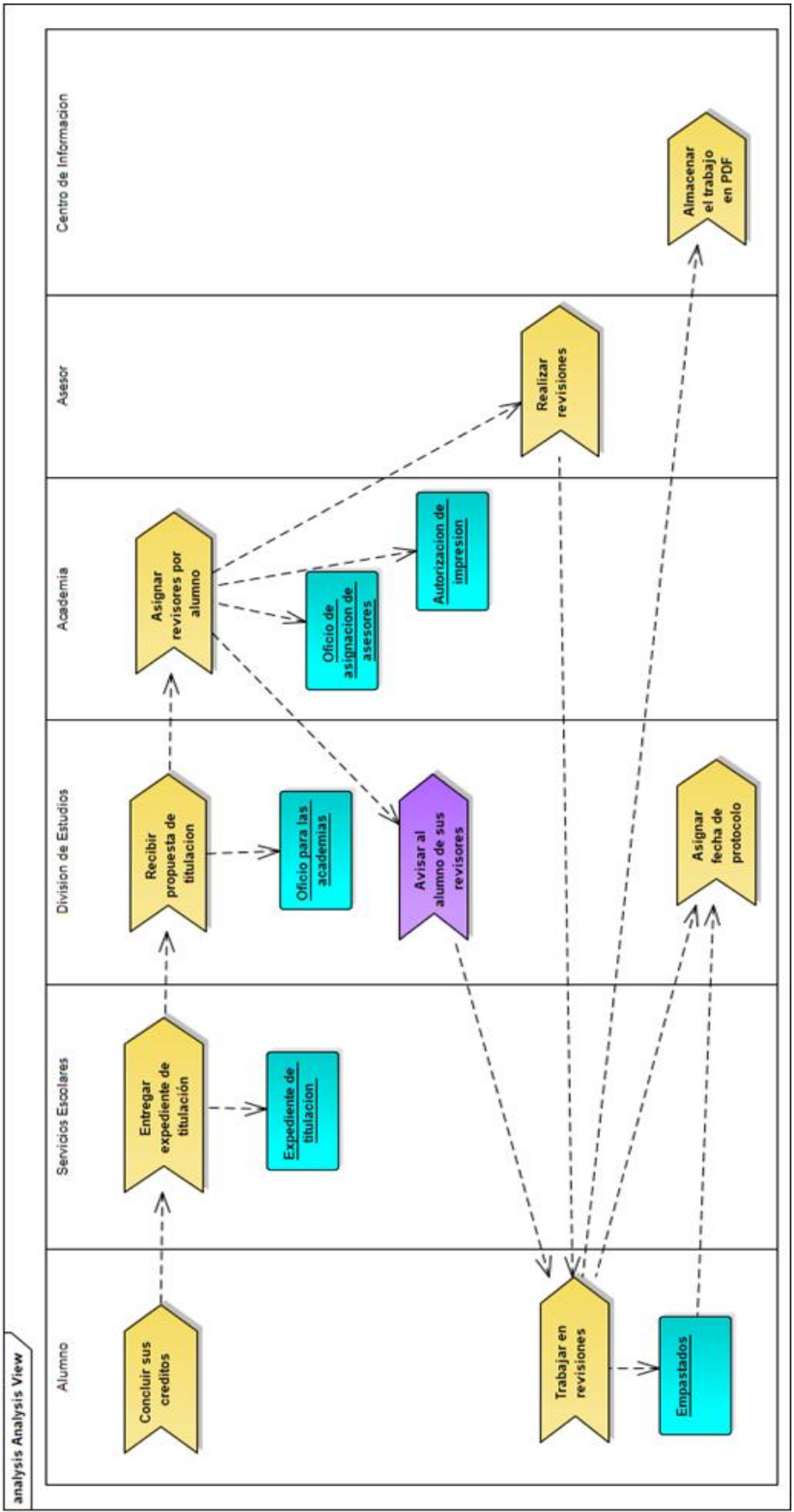
- **Asignar revisores por alumno:** La academia recibe los trabajos de titulación y asigna a 4 revisores del trabajo de acuerdo con el perfil del tema es como se asigna a los asesores. Realiza el oficio de asignación de asesores para entregarlo a la división de estudios profesionales en respuesta al anterior oficio de solicitud de asesores y este le informe al alumno quienes serán sus revisores y comience a trabajar con los asesores en las revisiones.

- **Realizar revisiones:** Una vez que se le asigna al alumno sus asesores se comienzan a trabajar en las revisiones de su trabajo de titulación junto con los asesores sobre la propuesta de titulación aceptada.

- **Almacenar trabajos en PDF**: En el centro de información se encuentra la Tesiteca que es el lugar donde se tienen almacenados todos los trabajos de titulación en empastados con sus respectivos formatos en electrónico en PDF. El centro de información es quien alimentara la base de datos de los formatos electrónicos en PDF que el sistema utilizara para realizar la búsqueda. La Tesiteca cuenta con formatos en electrónico desde el año 2005 a la fecha.

- **Trabajar en revisiones:** El alumno es informado por parte de la división de estudios quienes son sus asesores para trabajar con ellos en la revisión del documento.

El alumno termina con todas las revisiones y atiende las observaciones que le dan los docentes. El alumno se dirige al departamento académico para solicitar la fecha de presentación de predefensa y se le autorice la impresión.



2 Diagrama del modelado de negocio)

- **Asignar fecha de protocolo:** La división de estudios le presentara una fecha para su toma de protesta en la sala de titulación.

Diagrama del proceso de búsqueda

Descripción del proceso de búsqueda explotado dentro del modelado de negocio para el proceso de titulación:

- **Entrega el trabajo de titulación:** El alumno presenta el trabajo de titulación a la coordinación de titulación de forma impresa en cuatro tantos.

- **Recibir el trabajo:** Una vez que la coordinadora recibe el trabajo, empieza la búsqueda en el sistema para detectar las posibles similitudes con los trabajos almacenados en la base de datos, mediante los siguientes filtros:

  - Título
  - Objetivos
  - Planteamiento del problema
  - Antecedentes
  - Marco teórico

- **Iniciar la búsqueda:** El sistema es alimentado con los campos necesarios y comienza a realizar el análisis mediante la búsqueda de similitudes en la base de datos que se tiene de trabajos de titulación en formato electrónico PDF, entregados del 2005 al 2018.

Arroja como resultado un porcentaje de similitud para saber qué decisión tomar con respecto a él.

- **Atender el aviso:** Se le comunica al alumno que el trabajo que ha entregado es parecido a uno ya existente por lo que deberá volver a trabajarlo y así modificarlo o seleccionar otra forma de titulación ya que su trabajo inicial se ha detectado como copia de uno existente.

- **Concluir proceso:** Debido a que el porcentaje arrojado no es alarmante se toma que el trabajo presentado es original y se sigue dando seguimiento al proceso de titulación del estudiante.

- **Rango de porcentajes:** SI el porcentaje arrojado es mayor al 60 % es un posible copiado de un trabajo anterior. NO es >30% el trabajo no entra en el rango de copiado de uno anterior.

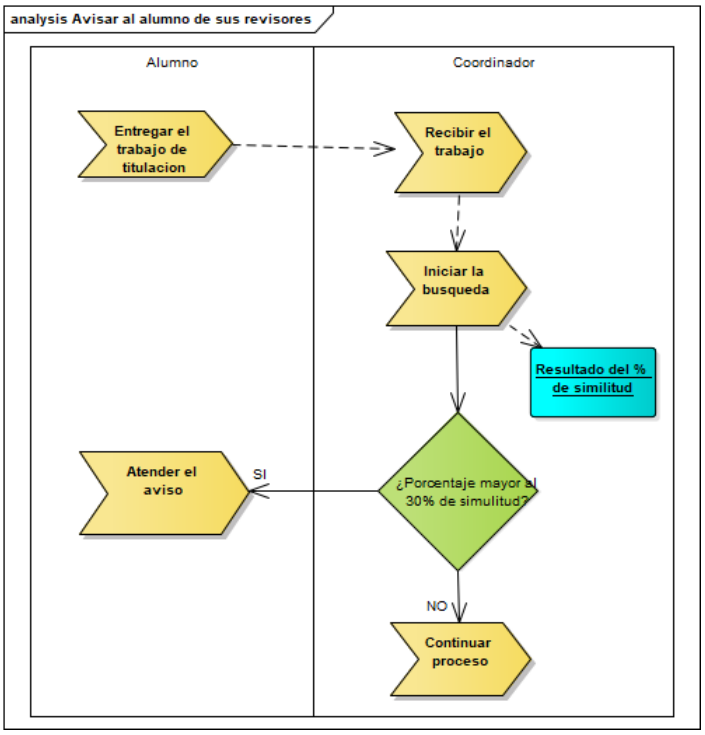


Figura 3.3 Diagrama del proceso de búsqueda

3.2.2 Diagrama de secuencia

A continuación se describe el diagrama de secuencia del proceso de búsqueda del sistema:

- **Usuario:** El usuario ha iniciado sesión, una vez hecho esto se dirige a solicitar al sistema acceder a la página de búsquedas.

La página de búsqueda manejara diversos filtros, los cuales son:

- ✓ -título
- ✓ -objetivos
- ✓ -justificación
- ✓ -introducción
- ✓ -antecedentes
- ✓ -planteamiento del problema

- **Base de datos**: En esta base de datos se estarán almacenando todos los archivos en electrónico para realizar las comparaciones y los procesos almacenados para realizar estas comparaciones.

- **Conexión de la base de datos:** El gestor confirma la conexión a ConexionBD, quien, a su vez confirmada al procesador de resultados, que se encargará de generar el *script* de consulta a enviar a la Base de Datos directamente. Cuando el procesador de resultados recibe los resultados del gestor de la base de datos, los procesa y genera la página de resultados que devuelve al usuario.

- **Filtro de Búsqueda:** La petición anterior mediante el método *post* envía la solicitud al procesador de resultados para este punto, ésta solicita conectarse a la base de datos mediante la clase ConexionBD, que se conecta directamente con el gestor de bases de datos.

- **Sesión principal:** el menú principal atiende esta petición y responde enviando la página. Allí el usuario especifica las palabras de búsqueda (en este caso primero se atiende a seleccionar el filtro del tema), y envía la petición a Index.

- **Petición de la búsqueda:** El usuario inicia la sesión y se dirige a la ventana principal donde seleccionará con qué filtro de búsqueda trabajará.

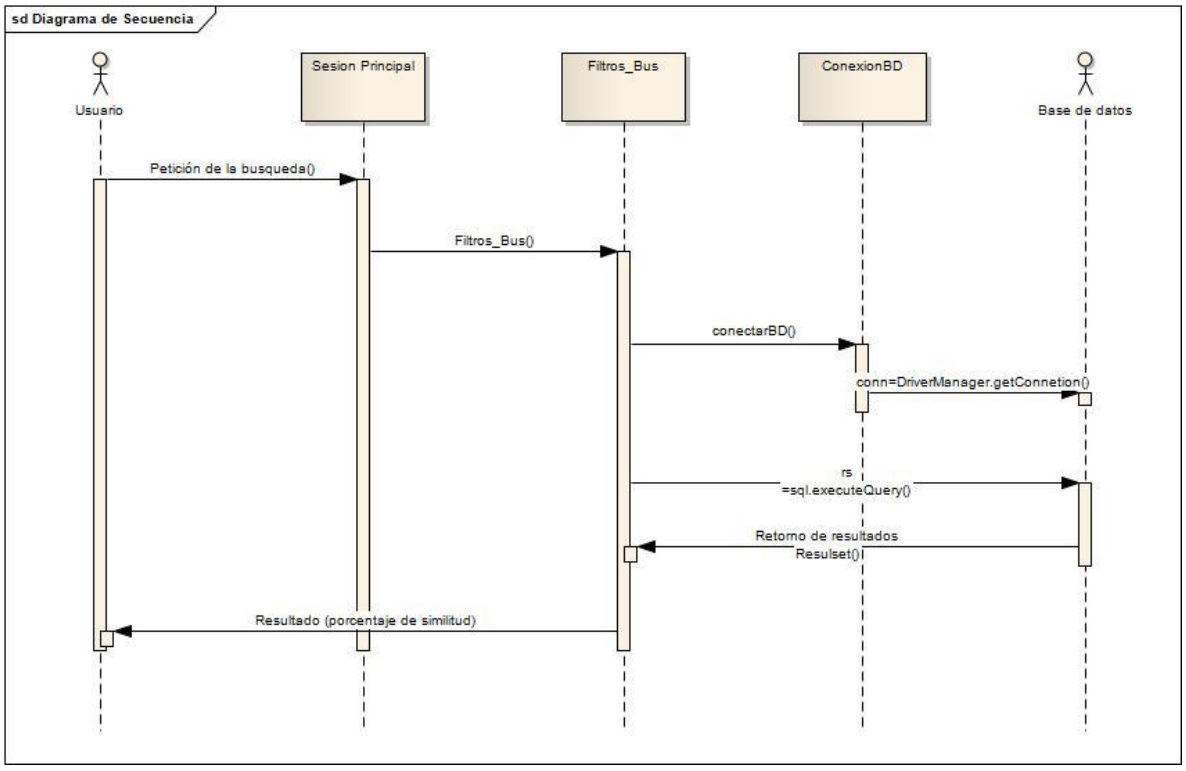


Figura 3.4 Diagrama de secuencia del sistema

3.2.3 Casos de uso

Para el modelado de caso de uso se elaboró un diagrama para el proceso de recepción de expediente y la búsqueda del trabajo en el sistema, dicho diagrama se muestra en la figura 3.5. La descripción de los actores es la siguiente:

- **Estudiante:** Persona egresada de la institución que cumple con todos los requisitos para integrar su expediente de titulación. El egresado previamente entrega un expediente de documentos a servicios escolares.
- **Coordinador:** Persona encargada recibir el expediente de titulación junto con el archivo electrónico de la Tesis o Memoria de Residencia, esto será dependiendo la forma de titulación que el egresado eligió previamente. Esta persona será la encargada de interactuar con el sistema.

Descripción de los casos de uso mencionados en el diagrama:

- **Entregar expediente:**

1. El egresado pasa a servicios escolares para entregar los documentos necesarios e iniciar el proceso de integración de expediente de titulación.
2. Servicios Escolares le entrega al estudiante una copia de este expediente generado.

3. El estudiante entrega esta copia del expediente junto con un archivo electrónico de la Tesis o Memoria de Residencia a la coordinación de Titulación.

- **Ingresar tesis al sistema:**

1. El coordinador valida sus datos de inicio de sesión en el sistema.
2. Ingresa al menú principal y selecciona el filtro en donde desea iniciar a buscar contra los archivos.

- **Realizar la búsqueda:**

1. El sistema tiene el filtro inicial y comienza la búsqueda.
2. Se termina la búsqueda y se obtiene el porcentaje de resultado.

- **Informar resultados al estudiante:**

1. El porcentaje arrojado de la búsqueda es mayor (alarmante) en todo el documento, se le informa al estudiante de una posible duplicidad y debido a esto no se le aceptara por el momento ese documento hasta que tenga las correcciones pertinentes.
2. El porcentaje arrojado es menor, el coordinador acepta el expediente de titulación y su archivo en electrónico.

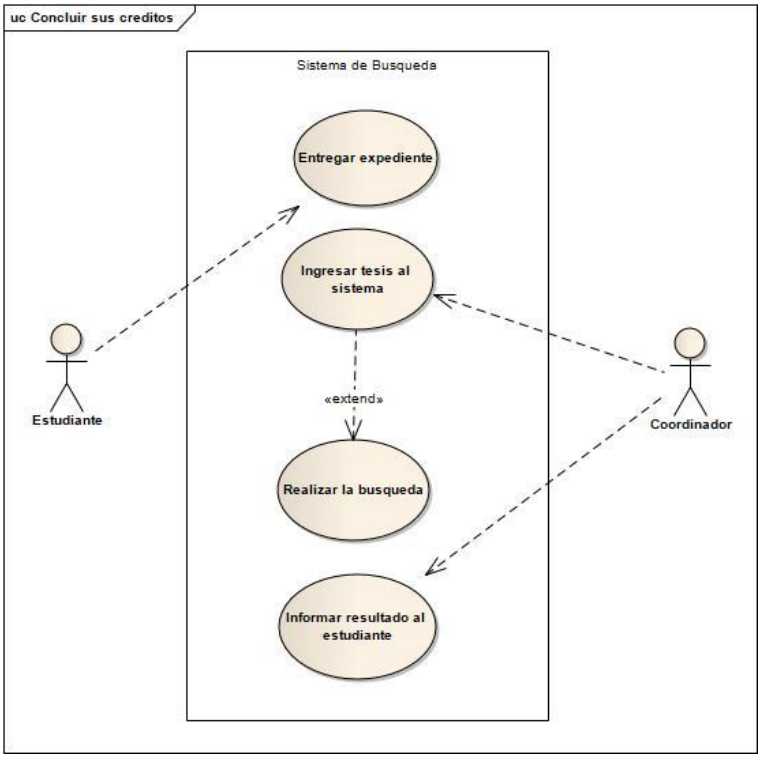


Figura 3.5 Diagrama de clase

3.4 Implementación

Se comenzaron a realizar pruebas con dos algoritmos diferentes para tener un comparativo de la eficiencia de cada uno. Estos dos algoritmos fueron el algoritmo *Knuth-Morris-Pratt* (KMP) implementado en una aplicación de consola con el lenguaje de programación C++ y el segundo es *Diff-Match-Patch* que es una librería de uso libre bajo la licencia de apache, esta librería implementa el algoritmo *diff de Myer*. Para esta segunda prueba se utilizó el lengua de programación C# y con el Visual Studio se creó una interfaz más de acuerdo a la que vamos a desarrollar en el proyecto.

3.4.1 Comparación de algoritmos

Este primer algoritmo KMP es algoritmo de búsqueda secuencial de texto llamado algoritmo de fuerza bruta, el cual recorrerá el texto carácter a carácter, buscando coincidencias con los caracteres de las palabras buscada en cada uno de los párrafos. Se sitúa el patrón en la primera posición y se compara carácter por carácter hasta encontrar un valor o llegar al final del patrón, se pasa a continuación a la siguiente posición y se repite este proceso. Termina hasta alcanzar el final del texto, no existe un pre procesamiento del patrón. En la figura 3.6 se muestra el ejemplo de cómo funciona el análisis de este algoritmo al comparar la cadena inicial contra un patrón determinado. (Búsqueda secuencial de texto).

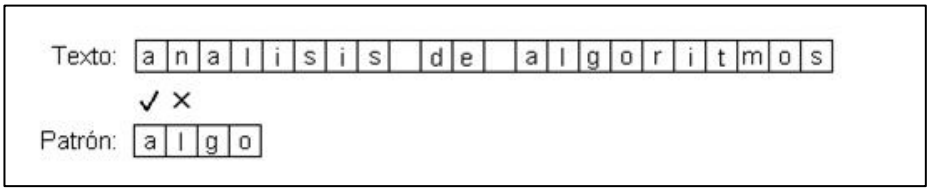


Figura 3.6 Alineación del patrón con la primera posición del texto (Busqueda secuencial de texto)

Se comparan los caracteres uno a uno hasta que se acabe el patrón, si se detiene la búsqueda por una discrepancia, se desliza el patrón a una posición hacia la derecha y se intenta calzar el patrón nuevamente. Esto se puede observar en la figura 3.7 (Búsqueda secuencial de texto).

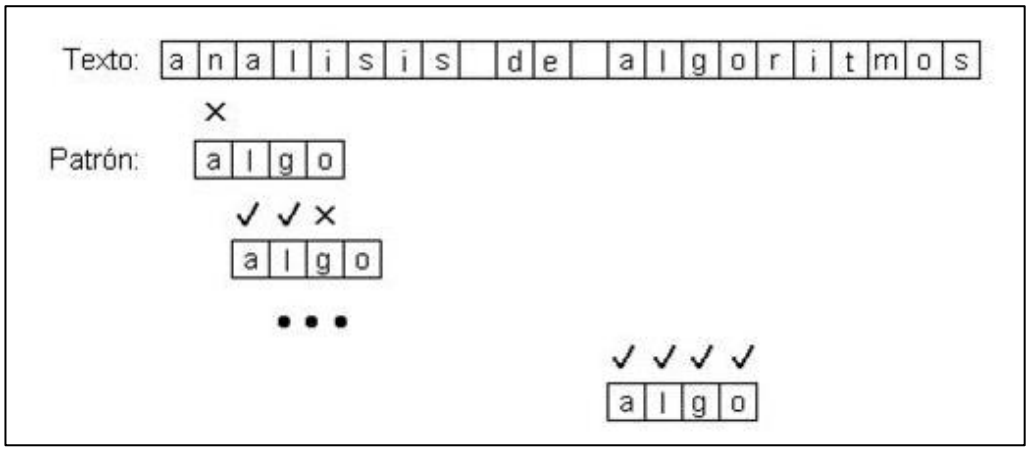


Figura 3.7 Se intenta seguir buscando el patrón nuevamente (Búsqueda secuencial de texto)

Con este algoritmo se creó una aplicación para comprobar su funcionamiento y eficiencia al procesar la búsqueda en el texto, los resultados obtenidos fueron los siguientes:

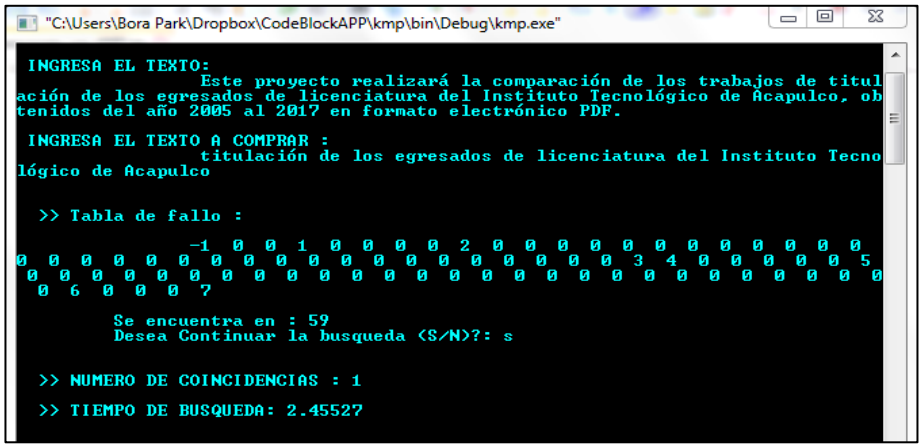


Figura 3.8 Búsqueda de similitud con un párrafo

En la figura 3.9 se ingresa un texto regular de tres renglones y posteriormente se procede a ingresar un párrafo para compararlo contra el original, se observa en la tabla

de fallas que donde se marca un índice de 1 es que se encontró una palabra similar en esa posición, pero a partir de la posición 59 es donde se localizó la cadena completa de similitud , como resultado final nos arroja que si se encontró una similitud en este texto, una vez localizado nos pregunta si deseamos seguir buscando más similitudes, de lo contrario salimos del programa.

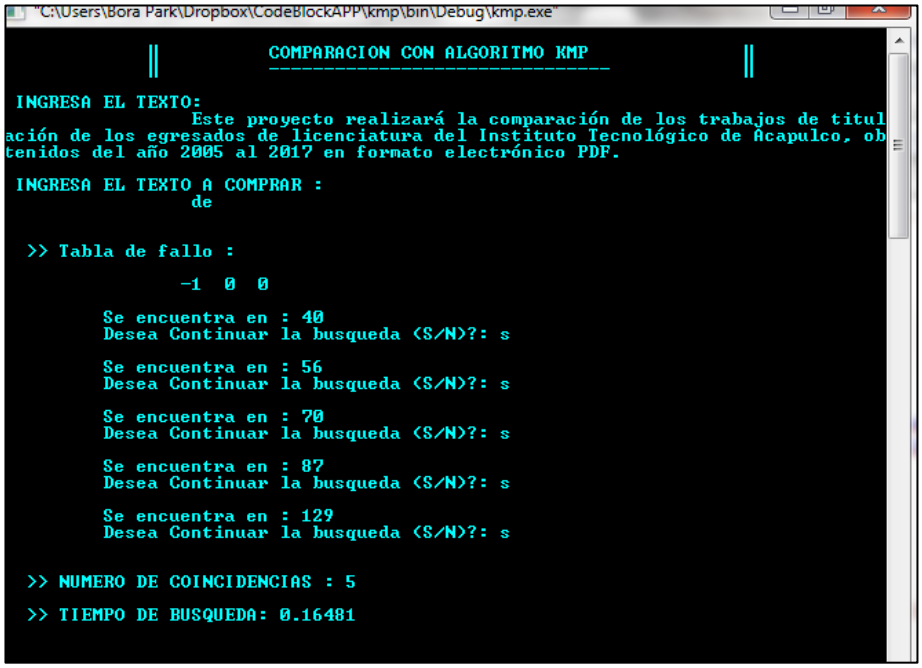


Figura 3.9 Búsqueda de similitud con una palabra

En esta segunda prueba sobre esa misma aplicación se ingresa solo una palabra para revisar los ciclos de las comparaciones de las cadenas y tras encontrar una nos arroje en que marcador está localizado este carácter, si se encuentra una similitud podemos decidir si seguir buscando más en todo el texto o detener la búsqueda tras encontrar las primeras, al final de la corrida como resultado se muestra el número de coincidencias de esta palabra en todo el texto ingresado. Este resultado se muestra en la Figura 3.9. En

ambas pruebas se tiene también el conteo del tiempo de ejecución que tarda este algoritmo en realizar la búsqueda.

Para esta segunda prueba se utilizó una biblioteca de alto rendimiento que se puede utilizar en varios lenguajes de programación para manipular texto sin formato. Las bibliotecas Diff Match y Patch ofrecen algoritmos robustos para realizar las operaciones necesarias para sincronizar texto sin formato. Originalmente construida en 2006 para potenciar Google Docs, esta biblioteca ahora se encuentra disponible en C ++, C #, Dart, Java, JavaScript, Lua, Objective C y Python.

Este algoritmo funciona mediante la búsqueda de forma recursiva en el centro del partido de dos secuencias, con la secuencia de comandos de edición más pequeña. Una vez hecho esto sólo el partido inicial se memoriza, y las dos subsecuencias anteriores y posteriores que se comparan de nuevo de forma recursiva hasta que no hay nada más para comparar. Para encontrar la pareja central se realiza haciendo coincidir los extremos de sub-secuencias medida de lo posible, y en cualquier momento no es posible, aumentar el guion de edición 1, explorando cada posición más alejada alcanzado hasta allí para cada diagonal y ver hasta nuevo partidos pueden ir, si un partido encuentra una palabra del otro extremo, el algoritmo acaba de encontrar el partido central.

El resultado de la implementación de este algoritmo se muestra en la Figura 3.10, con la ayuda de esta librería ingresamos un párrafo de mayor tamaño que en la aplicación anterior. Una vez ingresado el párrafo inicial ingresamos el siguiente texto a comparar, se tiene como resultado de la búsqueda dos secciones una donde se muestra el texto que se encontró igual y el que no es igual, mostrando este resultado en cada uno de los apartados de la aplicación. Si el texto es completamente igual la parte que muestra el texto diferente quedará en blanco y todo el texto que es igual se mostrará completamente.

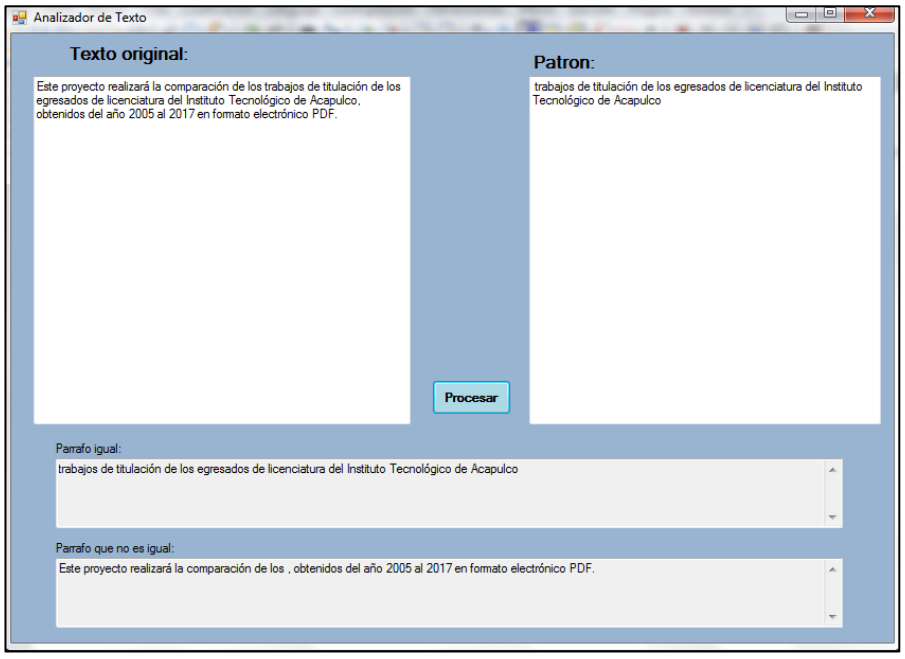


Figura 3.10 Implementación de la librería diff-match-patch

Capítulo 4 Desarrollo del proyecto

4.1 Creación de los Sprint

En este capítulo se explica el desarrollo del proceso que se llevó a cabo para realizar el sistema de información para detectar la similitud de las nuevas Tesis y Memorias de Residencias Profesionales entregadas en formato electrónico PDF contra las ya existentes.

Una vez que se tienen los requerimientos funcionales y no funcionales, se crea un listado de las actividades que se van a desarrollar y se anotan en la herramienta "Sprintometer" para tener un control del seguimiento de las actividades que se están desarrollando para la creación del sistema. Se crea el proyecto en la herramienta "Sprintometer" de tipo *"scrum"* y se le asigna el nombre del proyecto como se muestra en la figura 4.1:

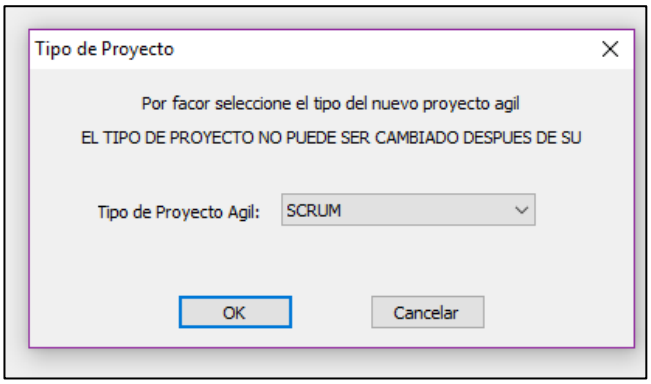


Figura 4.1Creación del proyecto.

Teniendo el proyecto creado con su respectiva unidad de estimación para el trabajo, en este caso *Horas Normales*, debido a que se le estarán dedicando de 5 a 8 diarias en el transcurso de los doce meses que se calendarizó para terminar la codificación, se dan de alta los recursos humanos para el proyecto y posteriormente poder capturar el listado de actividades. En la figura 4.2 se muestra la ventana del recurso humano asignado a cada actividad con sus respectivas horas de trabajo:

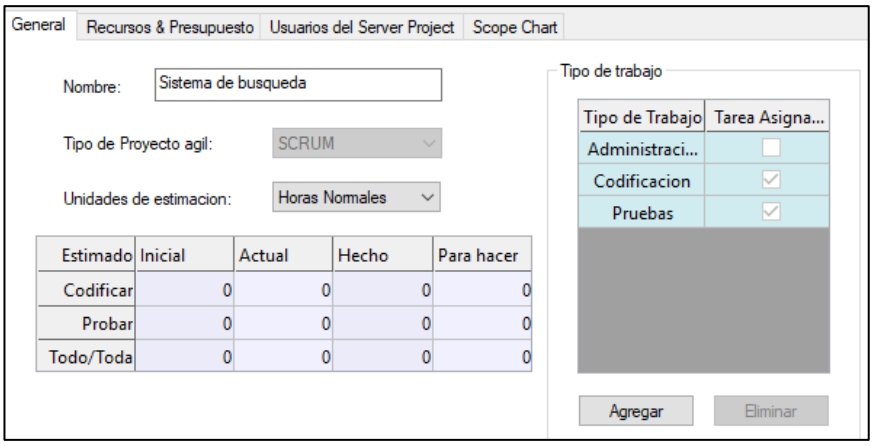


Figura 4.2 Asignación de horas y actividades

Después de crear los recursos humanos se capturan los "*sprint",* que son cada uno de los ciclos o iteraciones que vamos a tener dentro de este proyecto del tipo "*Scrum*", en cada uno se anotarán sus respectivas historias de usuarios. En cada una de ella se anota una descripción breve del desarrollo de la actividad así como el tiempo que se le está asignando a cada una, al finalizar el ciclo completo con las historias de usuarios asignadas en cada uno se realiza una prueba de este ciclo y al ejecutarse de manera correcta se sigue con el siguiente ciclo y así sucesivamente hasta terminar todos los ciclos. En la figura 4.3, se pueden ver todas las historias y "*sprints*" creados para este proyecto:

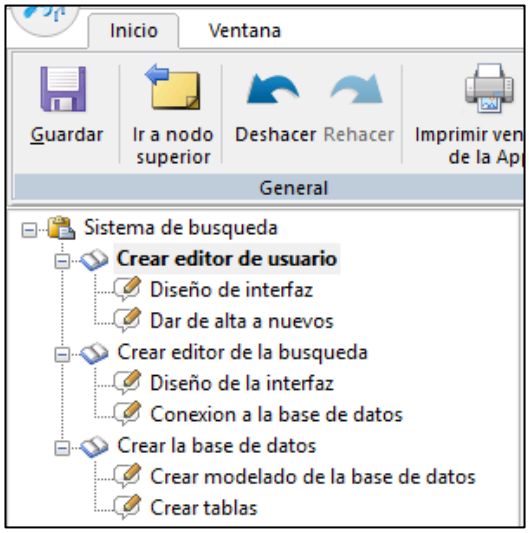


Figura 4.3 Lista de actividades creadas

4.2 Diseño y creación de la base de datos

Una vez que se han capturado todas las actividades en "Sprintometer" se procede a llevarlas a cabo. La primera de ellas es la creación de la base de datos que es parte esencial de este trabajo, debido a que en ella contendrá los documentos y procedimientos

almacenados que darán funcionalidad al algoritmo de *"Miyer*" para la detección de la similitud de los textos. Se optó por "*Mongodb"* debido a que mostró mejor eficiencia en el manejo de los archivos y esta base de datos se especializa en datos documentales generando archivos del tipo "*json"*. La ventaja de utilizar archivos de este tipo es que nos facilita mostrar o enviar información de una forma más rápida a los sistemas para ser interpretadas por otros sistemas, gracias a que es un formato independiente a cualquier lenguaje de programación.

4.2.1 Instalación de MongoDB

Como manejador de base de datos se optó por "*MongoDB"*, debido a que es una base de datos distribuida, basada en documentos, por lo que almacena datos en forma de documentos tipo "*json"*. Debido a esta característica nos ayudará a tener un mejor procesamiento de la información almacenada de los archivos en nuestra base de datos, logrando así un mejor tiempo de ejecución en las consultas de las comparaciones con los documentos almacenados en ella.

Una vez que se instaló el programa, se crean las carpetas de almacenamiento y configuración de *"MongoDB"*. Para dar permiso a la escritura y lectura de estas carpetas se ejecutan diversos comandos desde la terminal de *windows*, para este caso creamos una

carpeta de almacenamiento en una ruta segura, con el siguiente comando mostrado en la figura 4.4 a continuación:

```
D:\mongodb\data\db
```

Figura 4.4 Código de la creación de la carpeta segura

Posteriormente para iniciar el servicio del servido de *"MongoDB"*, se utiliza "*mongod"* como parte de la sintaxis de ejecución en la misma terminal que tenemos abierta anteriormente en la creación de la carpeta segura, en la Figura 4.5 se muestra el código de esta ejecución:

```
C:\mongodb\bin\mongo.exe
```

Figura 4.5Código para iniciar los servicios

4.2.2.1 Creación de la base de datos en MongoDB

La creación de la base de datos en esta herramienta fue de la forma siguiente; para la tabla alumnos se requieren datos generales de los alumnos como su número de control, apellido paterno, apellido materno, nombres. Correo electrónico, carrera a la que pertenece y un teléfono de contacto. La figura 4.6 muestra el código utilizado para la creación de la tabla *alumnos*:

```
alumnos.no_de_control,
alumnos.apellido_paterno,
alumnos.apellido_materno,
alumnos.nombre_alumno,
alumnos.correo_electronico,
carreras.nombre_carrera,
alumnos_generales.telefono
```

Figura 4.6 Código de la creación tabla alumnos

4.2.2.2 Almacenamiento de tesis

En esta sección se muestra como se están almacenando las Tesis o Memorias de Residencias Profesionales, por ejemplo: "_id" se deduce que es la clave primaria, "filename" quiere decir que es toda la ruta o nombre completo del archivo en formato *pdf*, aquí son "collections" en vez de tablas, y las filas se denominan "documentos". Se presenta a continuación en la figura 4.7 el código de cómo se está almacenando la información de los archivos electrónicos de cada una de las tesis almacenadas, guardando cada una con un identificador único y apuntando a la ruta donde estará almacenado el archivo:

```
Collection Fs.files //.> aquí se almacenan las tesis
{
    "_id" : ObjectId("5da3e462666fce5254cbd131"),
    "length" : NumberLong(2610079),
    "chunkSize" : 261120,
    "uploadDate" : ISODate("2019-10-14T02:58:42.716Z"),
    "md5" : "d66a1a81f3559056cef31c0770f2985a",
    "filename" : "C:\\Rosh\\Maestria\\Recursos\\Guias y tesis ejemplo\\Tesis MSC Generación 2016\\Tesis MSC -
    Cesar Javier Jimenez Rodriguez - DESARROLLO DE SISTEMA DE INFORMACIÓN.pdf"
}
```

Figura 4.7 Código de cómo se almacenan las tesis en la base de datos

En esta parte se muestra el código de cómo se almacenan la información y el estatus de los usuarios que están creados en el sistema. Los datos que se utilizan para esta tabla son datos generales del usuario como el nombre de usuario, apellido paterno, apellido materno, nombres, correo electrónico y para la información de su acceso su nombre de usuario, contraseña y el rol que estará desempeñando en el sistema. Se presenta a continuación en la figura 4.8 el código de almacenamiento de los datos de usuarios:

```
Collection auth_usuarios  //.> aquí se almacenan los usuarios que accederán al sistema
{
    "_id" : ObjectId("5da3da5a666fce40049d2f26"),
    "nombre_usuario" : "Crisol",
    "apellido_paterno" : "Mendiola",
    "apellido_materno" : "Piza",
    "correo_electronico" : "crisol@mendiola.com",
    "username" : "crisol_mendiola",
    "password" : "+NZqCaWQm7BfSeXqdUFJh6sD7Nm+sosv",
    "password_salt" : "259kLbg94lPg1nZJjSFGP7XAQALR/59d",
    "status" : {
        "_id" : ObjectId("5d8293304a88e78e66914ecd"),
        "nombre_status" : "Activo"
    }
```

Figura 4.8 Código del almacenamiento de los datos de usuarios

Como se mencionó en el capítulo 1 los archivos electrónicos con los que se trabajará para llevar a cabo la comparación son en formato “PDF”, para que la base de datos del sistema pueda analizar la información es necesario extraer el contenido de las Tesis o Memorias de Residencias Profesionales, la información que se extraerá es desde los datos generales del autor del trabajo hasta el contenido completo de dicho trabajo, que comprenderá la bibliografía y anexos. En la figura 4.9 se muestra el código como se realiza la extracción de dicha información y es almacenada en un identificador único con toda la información en un nuevo formato del tipo “json”.

```
{
    "_id" : ObjectId("5da68a44666fce13a412deab"),
    "titulo_tesis" : "mi tesis",
    "opcion_titulacion" : "Tesis profesional",
    "fecha_impreso" : ISODate("2019-10-16T03:11:00.294Z"),
    "alumnos" : [
        {
            "numero_control" : "G17320006 ",
            "nombre_completo" : "HUGO ROJAS SALGADO",
            "carrera" : "MAESTRIA EN SISTEMAS COMPUTACIONALES",
            "especialidad" : null,
            "tipo_autor" : "Autor"
        }
    ],
    "archivo_tesis_id" : ObjectId("5da68a36666fce13a412de9a"),
    "contenido_tesis" : "1 \r\nImplementación de Servicios Digitales E1 (VoIP) Utilizando la Plataforma Cisco Unified
    \r\nCommunications Manager 11 \r\nWilliams Nava Díaz \r"
}
```


Figura 4.9 Código de cómo se extrae la información de los trabajos

4.3 Descripción de las entradas de datos del sistema

Una vez creada e instalada la base de datos procedemos a ligar nuestra interfaz elaborada en "*Visual Studio 2018",* las ventanas presentadas a continuacion son las encargadas de ingresar los datos de los usuarios e ingresar la informacion de las Tesis o Memorias de Resiencias Profesionales para comenzar la comparacion.

4.3.1 Creación del Login

El sistema será utilizado por dos tipos diferentes de usuarios con diferentes privilegios:

Administrador: Usuario que cuenta con todos los privilegios del sistema, las acciones que puede realizar son las siguientes: modificar datos del registro las Tesis o Memorias de Residencias Profesionales almacenadas, registrar nuevos usuario o elimarnarlos y realizar la comparacion de los trabajos.

Coordinador: Usuario que se encargara de registrar los datos las Tesis o Memorias de Residencias Profesionales para ser almacenadas en la base de datos y llevar a cabo el proceso de la búsqueda de similitudes.

Una vez que se ejecuta el programa la primera ventana que se presenta es el *login*, en la cual el usuario ingresará sus credenciales para tener acceso a la interfaz principal del sistema. Los datos de los usarios deben estar registrados previamente desde la sesion de un administrador. En la figura 4.10 se muestra el formulario del *login* de la aplicación .

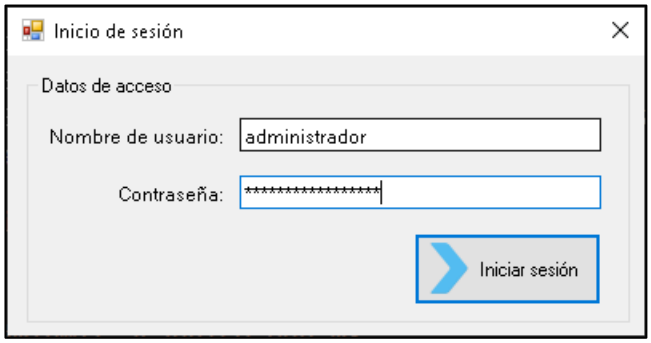


Figura 4.10 Login de la aplicación

4.3.2 Registro de usuarios

En este módulo del sistema se lleva a cabo la creacion de los nuevos usuarios que estarán interactuando con el sistema. En la ventana principal encontramos este módulo en la pestaña del lado derecho en el menu de configuración, como se muestra en la figura 4.11, al dar clic en configuracion se muestra la opción de que el usuario pueda ingresar al menú principal para iniciar la búsqueda o si es administrador pueda agregar a un usuario

nuevo, también dentro de esta sección de *agregar* está la opción para revisar los estatus de los usuarios registrados y si desea eliminar uno puede hacerlo desde esta misma opción .

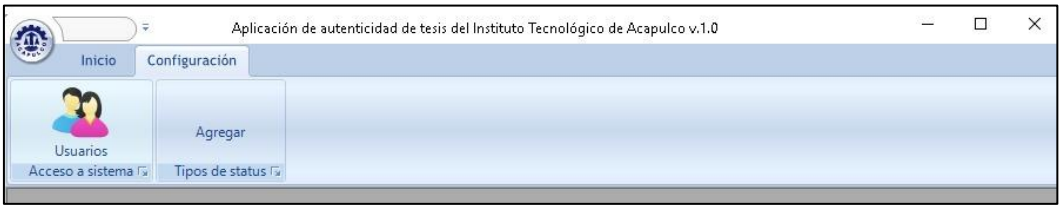


Figura 4.11 Módulo de usuario

Los datos que se le van a solicitar son los que se muestran en la figura 4.12, los cuales son datos básicos del usuario (datos generales) y los datos de usuario y contraseña para iniciar sesión (datos de acceso). Para crear un nuevo usuario llenamos los campos antes mencionados y damos clic en guardar. Del lado derecho se muestran los usuarios registrados y al momento de registrar uno nuevo si la operación fue de forma exitosa se puede vizualizar este dato registrado de forma correcta en la lista que se encuentra en el lado derecho como se muestra a continuación:

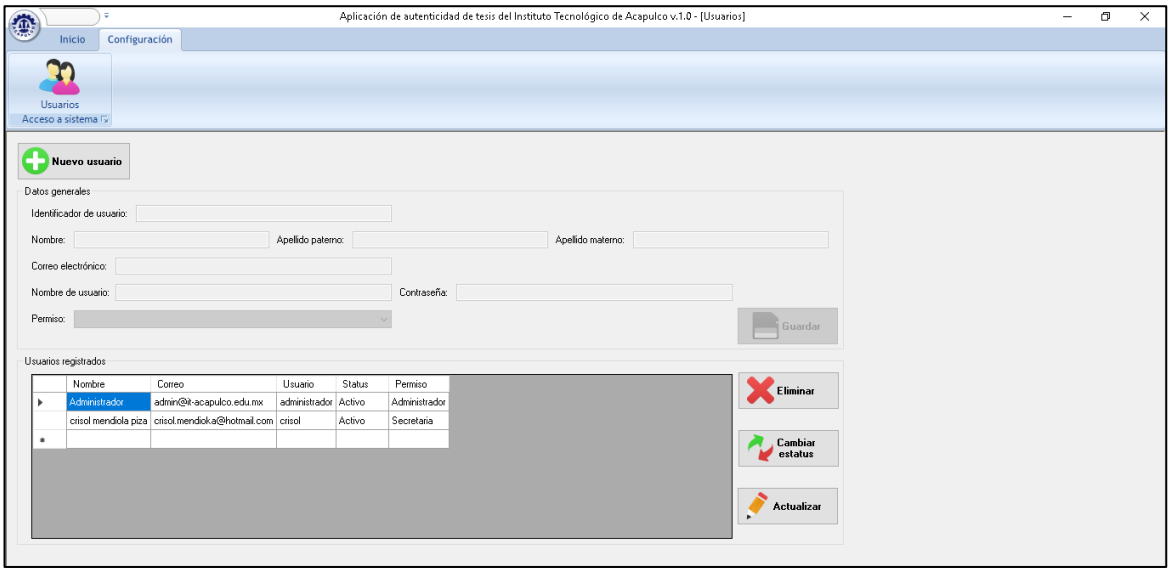


Figura 4.12 Registro de usuarios

4.3.3 Registro de archivo nuevo

El usuario a iniciado sesión se muestra la ventana principal que es la encargada de introducir los datos del nuevo documento como los de su autor (alumno), debido a que el sistema está conectado con el SII del Instituto Tecnológico de Acapulco. Una vez que se ingresa el número de control el sistema nos arrojará la información completa del alumno. En la siguiente sección (asesores) agregamos el nombre del director de Tesis o Memoria de Residencias Profesionales, dependiendo la opción de titulación que esté presentando el alumno y por último en la última sección se carga y guarda la nueva tesis. En la figura 4.13 se muestran las secciones anteriormente mencionadas.

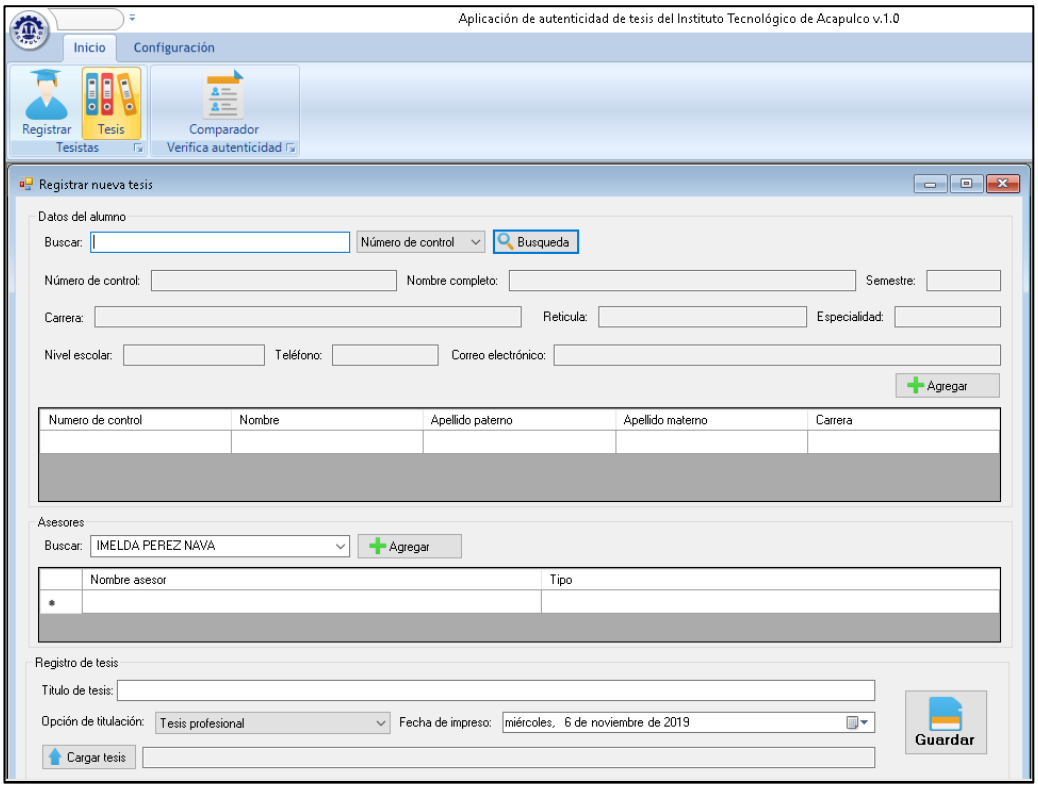


Figura 4.13 Registro de las Tesis o Memorias de Residencias

En la figura 4.14 se muestran un registro exitoso de un nuevo archivo, las tres secciones de datos a registrar son las siguientes :

Sección del alumno: Es la encargada de registrán los datos generales del alumno (nombre, número de control, carrera, retícula, telefono de contacto y correo electrónico).

Seccion de asesores: En esta parte se registran los nombres completos de los revisores del trabajo a registrar.

Seccion de registro de tesis : Esta última sección contiene los datos de la tesis a registrar para ser almacenada en la base de datos. Los datos que se requieren son el título, la opción de titulación y la fecha de impresión del empastado.

En la parte inferior localizamos el boton de *cargar archivo* que es con el cual podemos seleccionar la ubicación de donde se encuentra el trabajo electronico, una vez llenados los datos de las tres secciones mencionadas y seleccionada la ubicación del archivo, damos clic en guardar y si todo el proceso fue correcto se mostrará una ventana emergente de que el archivo fue guardado de manera exitosa.

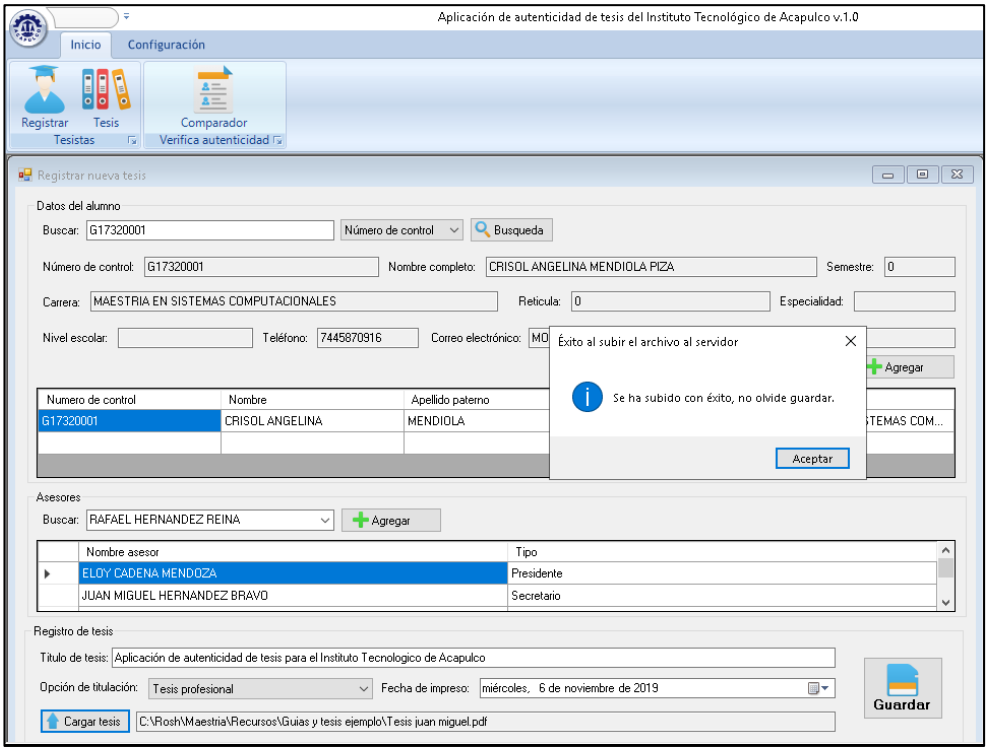


Figura 4.14 Registro exitoso de una tesis

4.4 Implementacion de la librería Diff Mach Pach

Como se menciona en el Capítulo 3 sección 3.4.1 se selecciona el algoritmo de *Myer* para llevar a cabo la comparación por ser más eficiente. Para el uso de este algoritmo de comparación en el proyecto se ocupa la biblioteca de uso libre de *Diff Mach Pach* . Esta biblioteca implementa el algoritmo *diff* de *Myer*, que generalmente se considera el mejor *diff* de propósito general. Una capa de aceleraciones pre-diff y limpiezas *post-diff* rodean el algoritmo *Diff*, mejorando tanto el rendimiento como la calidad de salida y la limpieza de los datos en el reprocesamiento de ellos. En el proyecto se crea esta clase llamada *DiffMatchPatch* que contienen las funciones para llevar a cabo las secuencias del proceso del algoritmo descritas en el siguiente Capítulo 5 sección 5.1 , donde se explica cómo funciona este algoritmo para realizar la comparación de texto. La figura 4.15 muestra la clase ya creada en el proyecto.

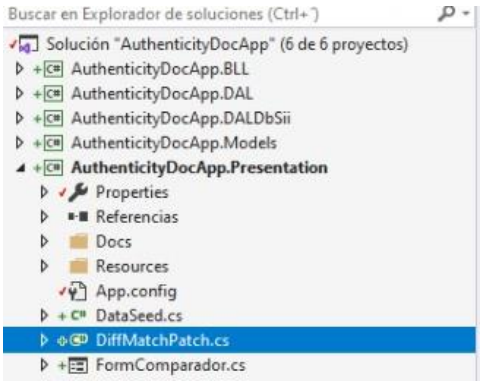


Figura 4.15 Clase DiffMachPach

Las partes principales de este algoritmo son el *Diff*, *mach* y el *pach* que se encargan de hacer la limpieza, procesamiento y la comparación del texto. En la figura 4.16 se muestra

parte del código del *Mach* , que es el encargado de buscar en el texto la mejor coincidencia difusa en un bloque de texto sin formato contra todos los documentos almacenados.

```
start_loc = match_main(text,
    text1.Substring(0, this.Match_MaxBits), expected_loc);
if (start_loc != -1) {
  end_loc = match_main(text,
      text1.Substring(text1.Length - this.Match_MaxBits),
      expected_loc + text1.Length - this.Match_MaxBits);
  if (end_loc == -1 || start_loc >= end_loc) {
    // Can't find valid trailing context.  Drop this patch.
    start_loc = -1;
  }
}
} else {
  start_loc = this.match_main(text, text1, expected_loc);
}
if (start_loc == -1) {
  // No match found.  :(
  results[x] = false;
  // Subtract the delta for this failed patch from subsequent patches.
```

Figura 4.16 Código del mach

Dado un texto para buscar, un patrón para buscar y una ubicación esperada la función del pach es, devolver la ubicación que coincida más cercana. La función pach buscará la mejor coincidencia según el número de errores de caracteres entre el patrón y la coincidencia potencial, así como la distancia entre la ubicación esperada y la coincidencia potencial. Parte de este código se muestra en la Figura 4.17.

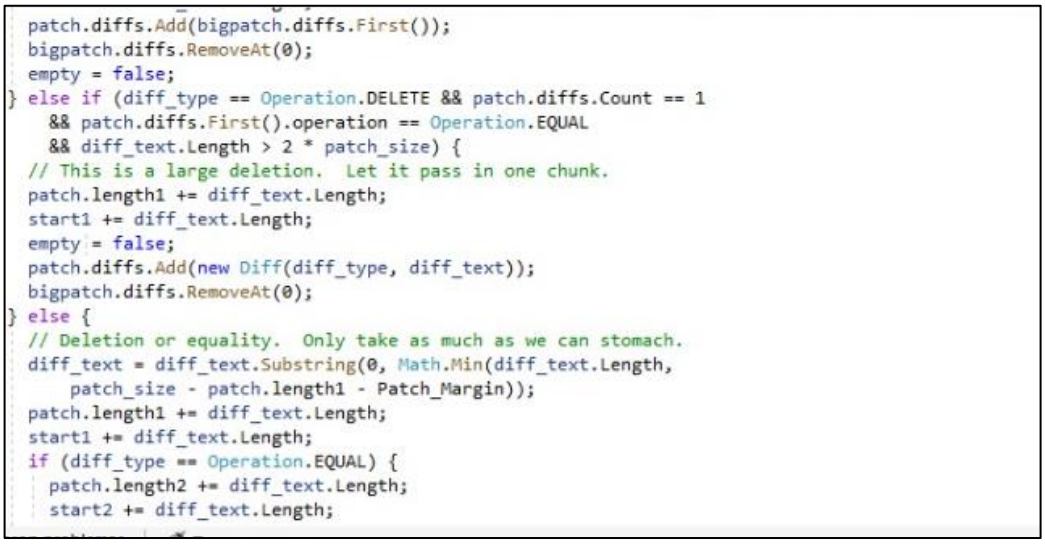
```
1 referencia
public string patch_addPadding(List<Patch> patches) {
  short paddingLength = this.Patch_Margin;
  string nullPadding = string.Empty;
  for (short x = 1; x <= paddingLength; x++) {
    nullPadding += (char)x;
  }

  // Bump all the patches forward.
  foreach (Patch aPatch in patches) {
    aPatch.start1 += paddingLength;
    aPatch.start2 += paddingLength;
  }

  // Add some padding on start of first diff.
  Patch patch = patches.First();
  List<Diff> diffs = patch.diffs;
  if (diffs.Count == 0 || diffs.First().operation != Operation.EQUAL) {
    // Add nullPadding equality.
```

Figura 2.17 Código del pach

Por último se tiene el *Diff* que es el encargado de calcular las diferencias dentro del texto y las distancias que se tienen entre ellas para poder realizar la comparación final entre ambos extremos de la cadena de caracteres tomada como patrón de partida. Esta misma sección brinda una limpieza de datos final para poder calcular el porcentaje correcto de la similitud encontrada. La figura 4.18 nos muestra una parte de dicho código.



```
  patch.diffs.Add(bigpatch.diffs.First());
  bigpatch.diffs.RemoveAt(0);
  empty = false;
} else if (diff_type == Operation.DELETE && patch.diffs.Count == 1
    && patch.diffs.First().operation == Operation.EQUAL
    && diff_text.Length > 2 * patch_size) {
  // This is a large deletion.  Let it pass in one chunk.
  patch.length1 += diff_text.Length;
  start1 += diff_text.Length;
  empty = false;
  patch.diffs.Add(new Diff(diff_type, diff_text));
  bigpatch.diffs.RemoveAt(0);
} else {
  // Deletion or equality.  Only take as much as we can stomach.
  diff_text = diff_text.Substring(0, Math.Min(diff_text.Length,
      patch_size - patch.length1 - Patch_Margin));
  patch.length1 += diff_text.Length;
  start1 += diff_text.Length;
  if (diff_type == Operation.EQUAL) {
    patch.length2 += diff_text.Length;
    start2 += diff_text.Length;
```

Figura 4.18 Codifo del Diff

Capítulo 5 Resultados y conclusión

5.1 Resultados

Se logró desarrollar un sistema que realiza la comparación de archivos electrónicos con el uso de herramientas gratuitas disponibles en internet como se mencionó en el capítulo 3. Para el desarrollo de la interfaz gráfica se utilizó *Visual Studio Comunity* en el lenguaje de programación *C#,* debido a que nos brindó mejor manipulación de la plataforma *.NET* para trabajar en conjunto con las característica de: reutilización de código de la librería *Diff Macth Patch*, eliminación de elementos dañados

de otros lenguajes (exportación de archivos tipo *json*), adaptación fácil de lenguajes de programación y la seguridad que tiene su mecanismo de control de acceso en los datos. Se selecciona a Mongo DB como el gestor de base de datos por la fácil manipulación de datos documentales en archivos del tipo *json*, ya que otros gestores no brindaban el manejo y almacenamiento de este tipo de archivo, se tenía que hacer una doble conversión para almacenar la información de las tesis.

Para obtener el porcentaje de similitud del nuevo documento contra los ya existentes se tiene que seguir el siguiente proceso descrito a continuación. Se deben cargar previamente las Tesis y Memorias de Residencias Profesionales en el sistema, este procedimiento se describe en el Capítulo 4. A continuación se establecen los filtros para delimitar los archivos en los que se comenzará a realizar la comparación, se selecciona la carrera y opción de titulación con la que se desea realizar la comparación del nuevo archivo y se da clic en el ícono de buscar para poder ejecutar el filtrado de estas opciones. En la Figura 5.1 se muestra el resultado de las tesis cargadas hasta el momento que corresponden a los datos seleccionados previamente.

Figura 5.1 Listado de trabajos registrados (Fuente: Elaboración propia)

.

Como se puede apreciar en la Figura 5.2, se encuentra una sección del lado izquierdo llamada patrón de búsqueda, en la cual se copiará el texto a comparar del nuevo documento electrónico, previamente a realizar esta búsqueda se debe seleccionar la carrera y la opción de titulación para delimitar el rango de la búsqueda. Una vez que se ingresa el texto a comparar damos clic en el botón *buscar* para que comience el proceso de la comparación. Del costado derecho en esa misma ventana se observa el resultado marcado en morado, que indica el texto igual que se encontró, mientras que el color verde indica el texto que no es similar. En la parte inferior se muestra un listado de tesis con el porcentaje de similitud detectadas en ellas con respecto al texto que se copió previamente en la sección de patrón de búsqueda .

La Figura 5.2 se muestra el listado de las tesis con el porcentaje del resultado de las similitudes detectadas en los documentos. En el ejemplo antes mencioando se cargarón 15 tesis para realizar la comparacion de los documentos.

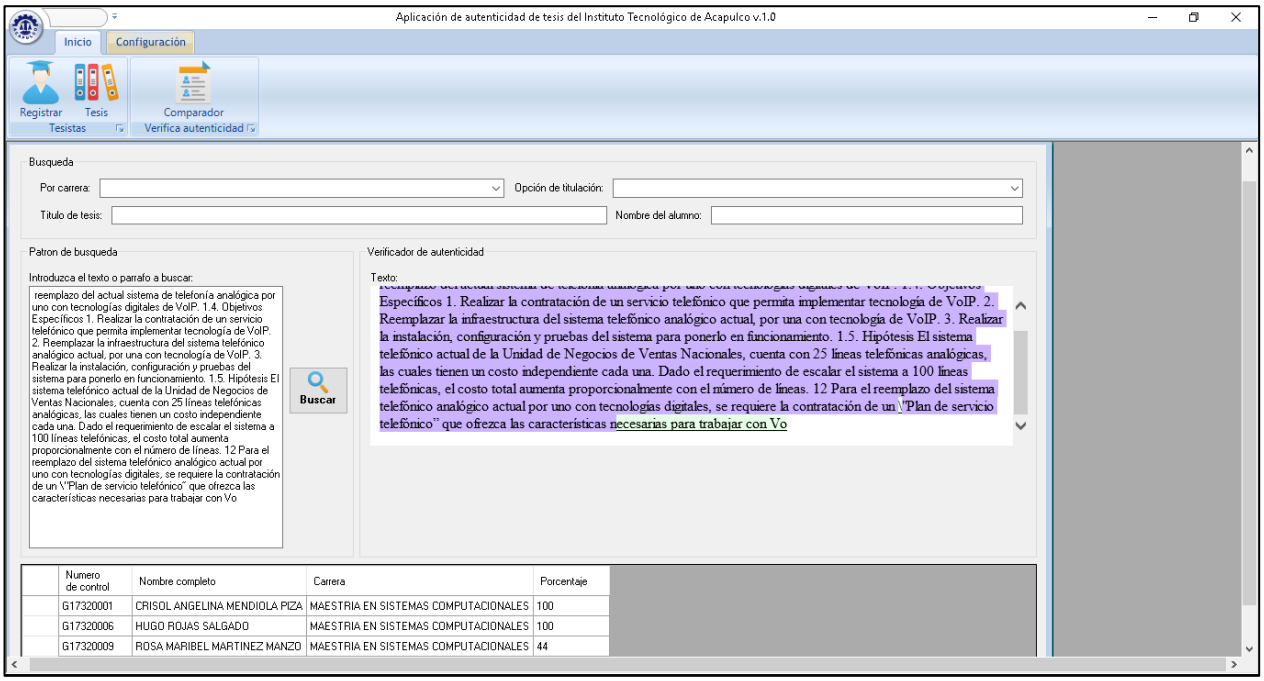


Figura 5.2 Resultado del porcentae de la búsqueda (Fuente: Elaboración propia)}

Para encontrar el algoritmo óptimo en la comparación de texto, se realizarón pruebas de los siguientes algoritmos KMP. Myer y la librería Diff Macth Patch descritos en el subtema de 3.4 “Implementación” del presente trabajo de tesis. Las pruebas consistían en el tiempo de ejecución respecto a la cantidad máxima de caracteres que podía comparar una vez que se insertaba el patrón de búsqueda. Finalmente se decide trabajar con el algoritmo de Myer apoyado con la librería de Diff Mach Pach, que brinda un menor tiempo de ejecución, debido a que se utiliza esta librería de código libre como motor de búsqueda en la base de datos documental.

A continuación se describe el proceso que utiliza el Algoritmo de *Myer para realizar la comparación de los documentos. U*na vez que se ingresa el texto a comprar (patrón de búsqueda) se realiza un primer proceso de limpieza que se llama prueba de igualdad, en el cual se busca que la secuencia de texto se encuentre igual en al menos uno de los documentos contra los que se está comparando. Si esta prueba es nula se garantiza que en la siguiente secuencia de búsqueda habrá una diferencia, y se elimina el caso de que se encuentre un documento igual, a continuación el porcentaje de igualdad se calculará con tres iteraciones posteriores que realiza el algoritmo. En la segunda iteración descarta la posibilidad de igualdad completa empezando a trabajar en los prefijos y subfijos en común, con esta iteración se busca que las cadenas de caracteres compartan alguna subcadena en común. Estas cadenas pueden estar al inicio o al final, el algoritmo divide la cadena total a la mitad y ahí se indica una bandera igual a Cero que es el punto de partida en el cual se toma como marcado para buscar entre los extremos de las cadenas de ida y vuelta.

En esta tercera secuencia de iteración es cuando se utiliza la librería de *GNU Diff* que se encarga de hacer la coincidencia lineal para los prefijos y sufijos. Con esta librería se aplica una limpieza del texto para las siguientes iteraciones de comparación. Después de realizarla se obtienen nuevos textos ya sin los prefijos y sufijos, y se comienzan a realizar las inserciones de comparación entre las cadenas de textos resultantes. Como última secuencia del algoritmo se generan nuevas ediciones resultantes en las que se realiza una comparación de diferencia final.

Una vez que se encontró el algoritmo óptimo mencionado se implementa la librería de Licencia GNU Diff Macth Patch para realizar la limpieza de los datos que se almacenaron en la base de datos. La librería de *Diff Macth Patch* se encarga de realizar la limpieza posterior al procesamiento inicial para generar un menor número de ediciones requeridas y así se pueda realizar más rápido el procesamiento de la comparación del texto. Gracias a este paso se elimina una sobrecarga de almacenamiento temporal de los nuevos archivos generados y con este proceso mejorar el tiempo de ejecución total.

Conclusiones

En la implementación del algoritmo de *Myer* con la ayuda de la librería de *Mach Diff Pach* en el sistema de búsqueda se demostró la eficiencia del algoritmo con respecto a la búsqueda de similitud y los tiempos de respuesta de la ejecución de la búsqueda. Mientras más pequeño sea el rango de la búsqueda el tiempo de respuesta se reduce, el ejemplo mencionado en la Figura 5.2 en el que se analizaron 15 trabajos su tiempo de respuesta fue de 35 segundos, mientras que en otro análisis realizado en 100 trabajos el tiempo de respuesta que nos arrojó fue de 2.5 minutos, debido a que era un mayor contenido a analizar y como se menciona en la sección 5.1 en análisis de las secciones copiadas a comparar se realizan sobre el contenido completo de las tesis, no solo una sección particular, la búsqueda se realiza sobre el contenido completo de todo el trabajo almacenado en la base de datos.

Se demuestra que con la ayuda de herramientas y el uso de librerías ya existentes se puede desarrollar un nuevo sistema de búsqueda de similitudes enfocado al algoritmo de *Myer* y gracias al uso de esta librería mencionada anteriormente se simplifica el tiempo de codificación. Debido a que la librería de *Mach Diff Pach* está de forma libre para desarrollar en diferentes lenguajes utilizamos esta librería basada en *C#* y solo se le hicieron modificaciones mínimas a este código existente para lograr implementar el patrón del algoritmo de Myer como motor de búsqueda y método de limpieza de datos en el sistema. Al implementar este algoritmo hacemos uso de patrones de búsqueda de *minería de datos* y así cumplir también con el propósito de esta maestría

profesionalizante respecto a la línea de investigación sobre la que estuve trabajando en toda mi estancia.

Al momento de recopilar los trabajos electrónicos de las Tesis y Memorias de Residencias Profesionales se observó que no se tiene un control u orden de dichos documentos, los documentos no se encontraban solo en una carpeta si no que el jefe encargado del área tuvo que revisar en todo el equipo para proporcionarme un material más completo con el cual trabajar. A pesar de ello los documentos que me proporcionó el área no es total de los empastados que se encuentran en físico, esto se debe a que en el área no cuenta con un personal específico encargado de atender la Tesiteca y se tiene mucho trabajo pendiente por realizar, parte de ese trabajo es continuar guardando en digital los trabajos de Tesis y Memoria de Residencias que se están en físico pero no se encuentran almacenados en electrónico en las carpetas de la PC.

Trabajo a futuro

El área de aplicación de este sistema de búsqueda con la implementación del algoritmo de *Myer* es muy amplia, debido a la rapidez con la que responde el tiempo de ejecución del programa. Este sistema se podría implementar en conjunto con los repositorios institucionales de los demás Tecnológicos para realiza una búsqueda externa de los trabajos y no solo con trabajos internos a nivel licenciatura con los que se estuvo trabajando para el desarrollo de este proyecto.

Es de gran importancia seguir realizando desarrollos de sistemas propios con la ayuda de estas herramientas ya existentes en el mercado y así reducir el tiempo de trabajo que se invierte en la codificación, debido a que podemos reutilizar código ya existente y adaptarlo a las necesidades del sistema. Se podría aplicar el uso de las tecnologías emergentes para dar más alcance al desarrollo del sistema que se desarrolló en este trabajo y llevarlo a búsquedas en tiempo real con archivos existentes en internet, como lo hacen otros sistemas de búsqueda que tienen costo en el mercado. La ventaja del sistema que se desarrollo es que tiene un costo bajo debido a que las herramientas para el desarrollo son de licencia gratuita y el único costo que se genera el del recurso humano encargado de la codificación del sistema.

Referencias bibliográficas

Silberschatz, et al., Fundamentos de Base de Datos, 4 ed., 2001.

Archivos PDF. (16 de Abril de 2018). Obtenido de Adobe Acrobat: https://acrobat.adobe.com)

Microsoft SQL Server. (1 de Mayo de 2018). Obtenido de wikipedia: https://es.wikipedia.org/wiki/Microsoft_SQL_Server

MySQL. (15 de Abril de 2018). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/MySQL

Oracle Database. (1 de Mayo de 2018). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Oracle_Database

PostgreSQL. (10 de Abril de 2018). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/PostgreSQLhttps://es.wikipedia.org/wiki/PostgreSQL

Abraham Silberschatz, H. F. (2002). Fundamentos de base de datos. España: McGraw-Hill.

American Psychological Association. (2010). Manual de Publicaciones de la American Psychological Association (6 ed.). (M. G. Frías, Trad.) México, México: El Manual Moderno.

Angel, E. N. (12 de Abril de 2018). Los archivos de los documentos electronicos. Obtenido de El profesional de la informacion: http://elprofesionaldelainformacion.com

Duque, R. G. (s.f.). Python para todos. España: Creative Commons Reconocimiento 2.5.

Esteban, E. V. (2 de Mayo de 2018). Lenguaje de programación C. Obtenido de https://informatica.uv.es/estguia/ATD/apuntes/laboratorio/Lenguaje-C.pdf

Guevara, J. M. (s.f.). Fundamentos de programación en Java. Madrid España: G-Tec.

Kendall, K. E. (2011). Analisis y Diseño de Sistemas. Mexico: Pearson Education.

Martínez, J. A. (Octubre de 2005). ¿Qué es Acrobat. Obtenido de Revista Digital Universitaria: http://www.revista.unam.mx

Seco, J. A. (2002). El lenguaje de programación C#.

Stallman, R. (15 de Abril de 2018). Microsoft Word. Obtenido de Enciclopedia Universal de Enseñanza Libres: http://enciclopedia.us.es/index.php/Microsoft_Word

Ian Somerville (2005). Ingeniería del Software .Pearson Educación.

DELÉGLISE, D. (2013). MySQL 5 (versiones 5.1 a 5.6): Guía de referencia del desarrollador.

Barcelona: Ediciones ENI.

Yera, Á. C. (2007). Diseño y programación de Bases de Datos. Madrid: Vision Libros.

Ceballos.S.F. (2013) Visual C#. Interfaces Gráficas y Silverligth. Ra-Ma

Adobe Reader. (Octubre de 2015). Caracteristicass Acrobat. Obtenido de la pagina oficial de Adobe Reader : https://acrobat.adobe.com/mx

Oracle. (2011). Java. Obtenido de la página oficial de Java: https://www.java.com

Eder dos Santos, Albert Aníbal Osiris Sofía, Roberto Uribe Paredes. (Año 2015, Volumen 7, No.2). Procesamiento de búsquedas por similitud. Tecnologías de paralelización e indexación (Paginas 111-138). Obtenido del Departamento de Ingeniería en Computación de la Universidad de Magallanes, Chile. https://dialnet.unirioja.es/servlet/articulo?codigo=5179331.

Anabella Bautista, Andrés Pascal y Juan Pablo Nuñez. (12 de Mayo del 2015). Indexación y búsqueda en base de datos (Paginas 1-5). Obtenido del Departamento de Ingeniería en Sistemas de Información, de la Universidad Tecnológica Nacional entre ríos, Argentina. http://hdl.handle.net/10915/45629.

Luis Britos, María E. Di Gennaro, Veronica Gil-Costa, Fernando Kasián. (Septiembre 2016). Filtrado de información para la búsqueda de respuestas (Paginas 145-152). Obtenido del Departamento de sistemas y lenguaje natural y sistemas de información, universidad de Alicante. http://www.sepln.org/revistaSEPLN/revista/37/19.pdf

Solar, Roberto, Uribe Paredes, Roberto , Gesto, Esteban ,Osiris, Sofía 2008. (Octubre 2008). Implementación de un digesto digital paralelo para búsquedas por similitud sobre documentos (Paginas 1-11). Obtenido Red de Universidades con Carreras en Informática (RedUNCI). http://hdl.handle.net/10915/21972

Sofia, Albert Osiris. (15 de diciembre del 2015). Evaluación de estructuras métricas con Unidades de Procesamiento Gráfico de Propósito General (Paginas 1-6). Obtenido de la Revista Tecnica Admnistrativa, Buenos Aires Argentina www.cyta.com.ar/ta1404/v14n4a3.htm

Kasián, Fernando, Reyes, Nora Susana. (Octubre 2012). Búsquedas por similitud en PostgreSQL Búsquedas por similitud en PostgreSQL (Paginas 1-10). Obtenido de: Red de Universidades con Carreras en Informática, Buenos Aires Argentina (RedUNCI). http://hdl.handle.net/10915/23754

Norma Beatriz Pérez, Mario Berón. (Abril 2013). Elaboración de Estrategias Paralelas para Búsquedas por Similitud en Espacios Métricos (Paginas 1-10). Obtenido de: Red de Universidades con Carreras en Informática (RedUNCI), Buenos Aires Argentina. http://hdl.handle.net/10915/27289

Edgar L. Chávez, Norma E. Herrera, Carina M. Ruano, Ana V. Villega. (Abril 2013). Métodos de acceso por similitud (Paginas 59-64). Obtenido de: Red de Universidades con Carreras en Informática (RedUNCI), Buenos Aires Argentina. http://hdl.handle.net/10915/21225

Andrés y Fernández, Gladys Vanesa. (18 de Enero del 2013). Desarrollo de una herramienta para el análisis y representación semántica de colecciones documentales a través del factor TF-IDF (Paginas 1-9). Obtenido de: Universidad Nacional de Mar del Plata. Facultad de Humanidades. Departamento de Ciencia de la Información; Argentina.. http://humadoc.mdp.edu.ar:8080/xmlui/handle/123456789/631

Anexo

A 1 Licencia apache algoritmo Myer

Licencia Apache

versión 2.0

http://www.apache.org/licenses/

TÉRMINOS Y CONDICIONES DE USO

Reproducción y distribución 1. Definiciones. "Licencia" significará los términos y condiciones de uso, la reproducción y distribución tal como se define por las secciones

1 a 9 de este documento. "Licenciante" significa el propietario del copyright o entidad autorizada por el propietario de los derechos que concede la Licencia. "Entidad jurídica", la unión de la entidad que actúe y todas las demás entidades que controlan, son controladas por, o está bajo control común con dicha entidad. Para los fines de esta definición, "control" significa (i) la potencia, directa o indirecta, para hacer que la dirección o la gestión de dicha entidad, ya sea por contrato o de otra manera, o (ii) la propiedad del cincuenta por ciento (50%) o más de las acciones en circulación, o (iii) la propiedad beneficiosa de dicha entidad. "Usted" (o "su") significa una persona física o jurídica que ejerza permisos concedidos por esta licencia. formulario "fuente" se entiende la forma preferida para hacer modificaciones, incluyendo pero no limitado a código de software de código, fuente de documentación y archivos de conFiguración. formulario de "objeto" significará cualquier forma resultante de la transformación mecánica o la traducción de una forma Fuente, incluyendo pero no limitado a código objeto compilado, documentación generada y las conversiones a otros tipos de medios. "Trabajo" se entenderá la obra de autor, ya sea en fuente u objeto, hecho disponible bajo la Licencia, como se indica por un aviso de copyright que se incluye en o unido a la obra (un ejemplo se proporciona en el Apéndice más adelante). "Obra derivada" significa cualquier trabajo, ya sea en la fuente u objeto, que se basa en (o derivado de) el trabajo y para el cual las revisiones editoriales, anotaciones, elaboraciones u otras modificaciones representan, en su conjunto, una obra original de la autoría. A los efectos de esta licencia, Trabajos derivados no incluir obras que se puedan separar de, o simplemente enlazan (o se unen por su nombre) a las interfaces de, la Obra y Obras derivadas de la misma. "Contribución" se refiere a cualquier obra de autor, incluida la versión original del trabajo

y las modificaciones o adiciones a esa obra o las obras derivadas de los mismos, que se presenta intencionadamente al Concedente para su inclusión en el trabajo por el propietario del copyright o por un individuo o persona jurídica autorizada a presentar en nombre del propietario del copyright. A los efectos de esta definición, "presentado" significa cualquier forma de comunicación electrónica, verbal o escrita enviada al licenciante o sus representantes, incluyendo pero no limitado a la comunicación en listas de correo electrónico, sistemas de control de código fuente, y los sistemas de seguimiento de incidencias son gestionados por, o en nombre de, el licenciante con el propósito de discutir y mejorar el trabajo, pero con exclusión de comunicación que está claramente marcada o designada por escrito por el titular de los derechos de lo contrario "No es una contribución". "Contribuyente" significará licenciador y cualquier persona física o jurídica en nombre de los cuales una contribución ha sido recibida por el licenciador y posteriormente incorporado en el Trabajo. 2. Concesión de licencia de copyright. Sujeto a los términos y condiciones de esta Licencia, cada Colaborador le otorga a Usted una licencia perpetua, mundial, no exclusiva, sin cargo, libre de regalías, licencia de copyright irrevocable para reproducir, preparar trabajos derivados de, mostrar públicamente, ejecutar públicamente, sublicenciar y distribuir el trabajo y tales obras derivadas de fuente u objeto. 3. Concesión de licencia de patente. Sujeto a los términos y condiciones de esta Licencia, cada Colaborador le otorga a Usted una licencia mundial, no exclusiva, sin cargo perpetua, libre de regalías, irrevocable (salvo lo dispuesto en esta sección) licencia de patente para fabricar, encargar, uso, oferta para vender, vender, importar y transferir el trabajo, cuando dicha licencia se aplica sólo a aquellos reivindicaciones de patente a licencia por dicho Colaborador que se ha infringido necesariamente por su

contribución (s) solo o por la combinación de su contribución (s) con la obra a la que se presentó dicha contribución (s). Si instituir un litigio de patentes contra cualquier entidad (incluyendo una reconvención o contrademanda en un juicio), alegando que la Obra o una Contribución incorporado en el Trabajo constituye una infracción de patente directa o contribuyente, a continuación, cualquier patente concedida a Usted bajo esta Licencia para ese obra resuelve a partir de la fecha de presentación de dicho litigio. 4. La redistribución. Usted puede reproducir y distribuir copias de la obra o las obras derivadas de los mismos en cualquier medio, con o sin modificaciones, y en la fuente u objeto, siempre que se cumplan las siguientes condiciones: (a) Debe dar a cualquier otro receptor de la obra o Derivadas una copia de esta licencia; y (b) Debe hacer que los archivos modificados lleven anuncios prominentes indicando que ha modificado los archivos; y (c) Usted debe conservar, en forma Fuente de cualquier obra derivada que distribuya, todos los derechos de autor, patentes, marcas, y avisos de atribución de la forma fuente de la obra, con excepción de las notificaciones que no pertenezcan a ninguna parte de la Trabajos derivados; y (d) Si el trabajo incluye un "aviso" archivo de texto como parte de su distribución, entonces cualquier Obra derivada que distribuya debe incluir una copia legible de los avisos de atribución contenidas dentro de dicho archivo de AVISO, excluyendo aquellos avisos que no pertenezcan a ninguna parte de la Obra Derivada, en al menos uno de los siguientes lugares: dentro de un archivo de texto AVISO distribuido como parte de las obras derivadas; dentro de la forma Fuente o documentación, si se proporciona junto con las obras derivadas; o, dentro de una pantalla generada por las obras derivadas, siempre y dondequiera que normalmente aparecen los avisos de terceros. El contenido del archivo AVISO son sólo para fines informativos y no modifican la

Licencia. Puede añadir sus propios avisos de atribución en obras derivadas que distribuya, junto o como una adición al texto de aviso del trabajo, siempre que tales avisos de atribución adicionales no pueden interpretarse como una modificación de la Licencia. Usted puede añadir su propia declaración de derechos de autor de sus modificaciones y puede proporcionar términos de licencia adicionales o diferentes y condiciones de uso, reproducción o distribución de sus modificaciones, ni de los Trabajos derivados en su totalidad, siempre y cuando su uso, reproducción y distribución del trabajo, además de cumplir con las condiciones establecidas en esta licencia. 5. Presentación de las contribuciones. A menos que indique expresamente lo contrario, cualquier Contribución de forma intencionada para su inclusión en el trabajo por usted al licenciante estará bajo los términos y condiciones de esta licencia, sin ningún tipo de condiciones adicionales. No obstante lo anterior, aquí no sustituyen o modifican los términos de cualquier contrato de licencia independiente que pueda haber ejecutado con el Concedente respecto a tales contribuciones. 6. Marcas. Esta licencia no otorga permiso para usar los nombres comerciales, marcas registradas, marcas de servicio o nombres de productos de la licenciador, excepto que se requiera para su uso razonable y usual en la descripción del origen de la obra y reproducir el contenido del archivo AVISO. 7. Exclusión de garantías. A menos que lo requiera la ley aplicable o se acuerde por escrito, Licenciante proporciona el trabajo (y cada Colaborador proporciona a sus contribuciones) en un "AS IS", SIN GARANTÍAS O

CONDICIONES DE CUALQUIER TIPO

Expresa o implícita, incluyendo, sin limitación, cualquier garantía o condiciones de título, no infracción, comerciabilidad o IDONEIDAD PARA UN FIN DETERMINADO. Usted es el único responsable de determinar la conveniencia de utilizar o redistribuir el trabajo y asumir todos los riesgos asociados a su ejercicio de permisos bajo esta licencia. 8. Limitación de responsabilidad. En ningún caso y bajo ninguna teoría legal, ya sea en agravio (incluyendo negligencia), contrato, o de otro modo, a menos que la legislación aplicable exija (como actos deliberados y negligencia grave) o acordado por escrito, Se ningún Colaborador será responsable ante usted por daños, incluyendo daños directos, indirectos, especiales, accidentales o consecuentes de cualquier carácter que surge como resultado de esta Licencia o por el uso o la imposibilidad de utilizar el trabajo (incluyendo, pero no limitado a daños por pérdida de buena voluntad, paro, falta de equipo o mal funcionamiento, o cualquier otro daño o pérdida comercial), incluso si dicho colaborador ha sido advertido de la posibilidad de tales daños. 9. La aceptación de la garantía o responsabilidad adicional. Mientras redistribución de la obra o trabajos derivados de éstos, usted puede optar por ofrecer, y cobrar una tarifa por, la aceptación de apoyo, garantía, indemnización u otras obligaciones y / o derechos pasivos consistentes con esta licencia. Sin embargo, en la aceptación de tales obligaciones, Es posible actuar sólo en su propio nombre y bajo su responsabilidad, no en nombre de cualquier otro colaborador, y sólo si está de acuerdo en indemnizar, defender y mantener cada contribuyente a salvo de toda responsabilidad incurrida por, o reclamaciones interpuestos contra, dicho Colaborador en razón de su aceptación de cualquier garantía o responsabilidad adicional.

FIN DE LOS TÉRMINOS Y CONDICIONES

Apéndice: Cómo aplicar la licencia Apache a su trabajo. Para solicitar la licencia Apache a su trabajo, coloque el siguiente aviso repetitivo, con los campos encerrados por corchetes "[]" reemplazado con su propia información de identificación. (No incluya los corchetes!) El texto debe estar encerrada en la sintaxis de comentario apropiado para el formato de archivo. También se recomienda que un nombre de archivo o de clase y descripción del propósito incluirse en la misma "página impresa" como el aviso de copyright para facilitar su identificación dentro de los archivos de terceros. Los derechos de autor [aaaa] [nombre del propietario del copyright] licenciado bajo la licencia Apache, versión 2.0 (la "Licencia"); no se puede utilizar este archivo salvo en cumplimiento de la Licencia. Usted puede obtener una copia de la Licencia en http://www.apache.org/licenses/LICENSE-2.0 menos que sea requerido por la ley aplicable o se acuerde por escrito, el software distribuido bajo la Licencia se distribuye "tal cual", SIN GARANTÍAS o CONDICIONES DE CUALQUIER TIPO, ya sea expresa o implícita. Consulte la Licencia para los permisos idioma específico que rige y limitaciones de la Licencia. como el aviso de copyright para facilitar su identificación dentro de los archivos de terceros. Los derechos de autor [aaaa] [nombre del propietario del copyright] licenciado bajo la licencia Apache, versión 2.0 (la "Licencia"); no se puede utilizar este archivo salvo en cumplimiento de la Licencia. Usted puede obtener una copia de la Licencia en http://www.apache.org/licenses/LICENSE-2.0 menos que sea requerido por la ley aplicable o se acuerde por escrito, el software distribuido bajo la Licencia se distribuye "tal cual", SIN GARANTÍAS o CONDICIONES DE CUALQUIER TIPO, ya