DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN SISTEMAS E INFORMÁTICA

AUTOR: JESSICA CAROLINA BALAREZO GALARZA

ANÁLISIS DE FACTIBILIDAD Y SELECCIÓN DE UN FRAMEWORK DE BÚSQUEDA GLOBAL PARA SU IMPLEMENTACIÓN EN EL SISTEMA GESTOR FIDUCIA FONDOS JEE DE LA EMPRESA GESTORINCSA S.A.

DIRECTOR: ING. CORAL, HENRY

CODIRECTOR: ING. SALVADOR, SANTIAGO

SANGOLQUÍ, SEPTIEMBRE DE 2014

CERTIFICACIÓN

En mi calidad de Director del trabajo de investigación: “ANÁLISIS DE FACTIBILIDAD Y SELECCIÓN DE UN FRAMEWORK DE BÚSQUEDA GLOBAL PARA SU IMPLEMENTACIÓN EN EL SISTEMA GESTOR FIDUCIA FONDOS JEE DE LA EMPRESA GESTORINCSA S.A.”, elaborado por la señorita Jessica Carolina Balarezo Galarza, egresada de la Carrera de Ingeniería en Sistemas e Informática, **Certificamos** que fue dirigida observando los aspectos técnicos y reglamentarios de la norma vigente.

Por lo tanto autorizamos su presentación ante los organismos pertinentes.

_____	_____
ING. HENRY CORAL DIRECTOR DE TESIS	ING. SANTIAGO SALVADOR CORDINADOR DE TESIS

DECLARACIÓN DE RESPONSABILIDAD

Yo Jessica Carolina Balarezo Galarza

DECLARO QUE:

La presente tesis de grado titulada “ANÁLISIS DE FACTIBILIDAD Y SELECCIÓN DE UN FRAMEWORK DE BÚSQUEDA GLOBAL PARA SU IMPLEMENTACIÓN EN EL SISTEMA GESTOR FIDUCIA FONDOS JEE DE LA EMPRESA GESTOR INC.” ha sido desarrollada bajo una exhaustiva investigación, respetando el contenido de las fuentes en las que se han consultado y se cita en los pies de página. Este trabajo es de mi autoría y me responsabilizo del contenido y la veracidad del documento.

Sangolquí, 24 de Septiembre del 2014

_____

Jessica Carolina Balarezo Galarza

AUTORIZACIÓN

Autorizo a la Universidad de las Fuerzas Armadas – ESPE la publicación en la Biblioteca Virtual de la Institución del trabajo ANÁLISIS DE FACTIBILIDAD Y SELECCIÓN DE UN FRAMEWORK DE BÚSQUEDA GLOBAL PARA SU IMPLEMENTACIÓN EN EL SISTEMA GESTOR FIDUCIA FONDOS JEE DE LA EMPRESA GESTOR INC., cuyo contenido es de mi responsabilidad y autoría.

Sangolquí, 24 de Septiembre del 2014

_____

Jessica Carolina Balarezo Galarza

DEDICATORIA

A mi familia por ser una parte fundamental en mi vida y brindarme su apoyo y ayuda incondicional para cumplir una meta más en mi vida.

AGRADECIMIENTO

Agradezco a todos quienes me apoyaron y contribuyeron para lograr uno más de mis objetivos profesionales, a mis amigos y amigas por estar siempre ahí y compartir momentos agradables en toda la carrera universitaria y a mis profesores por compartir sus conocimientos y experiencias académicas.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

RESUMEN

Con el tiempo el almacenamiento de documentos en repositorios relacionales (bases de datos) de las empresas fue creciendo constantemente; sin embargo al momento de buscar información dentro de cada uno de ellos se volvió una tarea compleja que muchas veces era imposible de realizar debido a las limitaciones tecnológicas. Por esta razón muchas empresas se vieron en la necesidad de adaptar sus sistemas empresariales transaccionales con sistemas de búsqueda documental los cuales tienen la capacidad de acceder a estos repositorios y extraer la información de cada documento para ejecutar los procesos de búsqueda. Actualmente existen diferentes herramientas en el mercado de licenciamiento privado o público para ser integradas en diferentes plataformas de software de las empresas. El presente artículo detalla la implementación de un sistema de Búsqueda Documental genérico el cual puede ejecutar de forma independiente búsquedas de texto en documentos de formato propietario como Microsoft Office, PDF, entre otros y que se encuentran almacenados en una base de datos relacional. El producto final desarrollado y que es la parte central de esta tesis de grado se llama “Gestor Search”; una aplicación web desarrollada bajo los estándares de la plataforma JEE y que integra el uso de las librerías Apache Tika y Apache Lucene, para cumplir con este propósito, siendo una aplicación de búsqueda empresarial que cuenta con las funcionalidades de indexación, búsqueda y acceso a los documentos.

PALABRAS CLAVE: BÚSQUEDA, TIKA, LUCENE, DOCUMENTAL, REPOSITORIOS, BASE DE DATOS.

ABSTRACT

Eventually storing documents in relational repositories (databases) of companies grew steadily; however when seeking information within each a complex task that often became impossible to perform was due to technological limitations. For this reason many companies saw the need to adapt their systems with transactional enterprise information retrieval systems which have the ability to access these repositories and extract information from each document to run the search processes. Currently exist several tools on the market for private or public licensing that can be integrated into the various software platforms companies. This paper details the implementation of a generic search Documentary system which can run independently text searches on documents proprietary format as Microsoft Office, PDF, among others that are stored in a relational database. The final product developed that is the central part of this article is called "Gestor Search"; a web application developed under the standards of the JEE platform and integrates the use of libraries and Apache Tika Apache Lucene, to accomplish this purpose. Gestor Search is an application enterprise search functionality has indexing, search and access to documents.

KEYWORDS: BÚSQUEDA, TIKA, LUCENE, DOCUMENTAL, REPOSITORIOS, BASE DE DATOS.

CAPÍTULO 1
INTRODUCCIÓN

1.1 Título

Análisis de Factibilidad y selección de un “framework” de Búsqueda Global para su implementación en el Sistema Gestor Fiducia Fondos de la Empresa Gestor.

1.2 Introducción

Cada día la búsqueda de la información en las empresas es más grande por la cantidad de repositorios que en ellas existen. Por eso se ha visto la necesidad de investigar nuevas tecnologías que ayuden a mejorar el desempeño del sistema Gestor Fiducia Fondos JEE y poder implementarlas. Estas nuevas tecnologías deben ser fáciles, rápidas y muy confiables hacia el usuario que las necesita diariamente.

Por lo general las personas están acostumbradas a realizar las búsquedas con palabras claves o frases para obtener los documentos que coincidan con dicha búsqueda, un ejemplo son los buscadores más populares usados por todos como son Google o Yahoo. Por esto se han creado “frameworks” de búsqueda global que tienen las mismas características solo con la mínima diferencia que se lo utilizará para búsquedas en intranet, siendo compatibles con los documentos más utilizados.

Los resultados de las búsquedas realizadas se presentan en una página html con el título del documento, el formato y con un link que lo direccionará al documento original guardado en la base de datos presentándolo como PDF o con la descarga automática del documento si es con el formato .doc, .docx, .xls, etc.

1.3 Planteamiento del Problema

Actualmente la búsqueda de información como archivos de Word, Excel, PowerPoint, PDF, archivos de texto, etc, almacenados en múltiples repositorios lleva demasiado tiempo, haciendo que la búsqueda sea prolongada por la masiva documentación que se tiene. Lo anterior por mucho tiempo se ha vuelto un problema muy importante y primordial para todos, generando inconvenientes al momento de buscar un documento.

El desconocimiento de las nuevas tecnologías se debe a la poca o ninguna información ofrecida por los creadores acerca de su uso.

Las nuevas tecnologías en un principio pueden llegar a ser muy costosas, impidiendo la adquisición de las mismas para pequeñas o medianas empresas. Por eso se debe realizar un estudio de factibilidad tanto económico y operativo de las nuevas tecnologías para el beneficio de la empresa y de los usuarios.

Por lo tanto la selección de un motor de búsqueda como Oracle Secure Enterprise Search y Lucene es indispensable para el sistema Gestor Fiducia Fondos JEE.

1.4 Objetivos

1.4.1 Objetivo General

Analizar la factibilidad y selección de un “framework” de búsqueda global mediante la implementación en el Sistema Gestor Fiducia Fondos para mejorar su funcionamiento dentro de la empresa Gestor.

1.4.2 Objetivos Específicos

- Facilitar las búsquedas de documentos en el sistema Gestor Fiducia Fondos.

- Presentar un cuadro comparativo de los “frameworks” seleccionados y escoger el mejor según los resultados obtenidos.

- Diseñar el proceso de solicitud, indexación, búsqueda y respuesta de los documentos almacenados en la base de datos de acuerdo al “framework” seleccionado.

- Realizar un control de versionamiento del sistema de acuerdo al “framework” seleccionado.

1.5 Justificación

Debido a la necesidad de la implementación de un “framework” en el sistema Gestor Fiducia Fondos para la búsqueda de documentos almacenados en la base de datos, es necesaria la investigación de nuevas tecnologías, que sean fáciles, rápidas, confiables y seguras, facilitando el trabajo diario. Por estas razones el presente estudio pretende dar la solución con la utilización de un “framework” de búsqueda global, brindando una mejor calidad de servicio a sus empleados e indirectamente a sus usuarios, con los resultados que se obtengan se resolverán los problemas planteados, contribuyendo con el mejoramiento del sistema y del desempeño laboral.

Oracle Secure Enterprise Search provee a los usuarios un acceso seguro a todas las fuentes de datos, sitios web, servidores de archivos, sistemas de gestión de contenido, sistemas de planificación de recursos empresariales y gestión de relaciones con clientes, sistemas de inteligencia de negocio y bases de datos de la organización. Así se mejora el acceso a la

información empresarial, al mismo tiempo protege los datos delicados ante los usuarios no autorizados.

Con Lucene se puede usar en cualquier aplicación que requiera indexado y búsqueda de texto completo, la indexación consiste en analizar y extraer toda la información disponible que pueda representarse en forma textual creando los índices que se utilizarán en la búsqueda, este API puede ser usado para su utilidad en la implementación de motores de búsquedas. Lucene es una librería, una herramienta de desarrollo, no es una aplicación de búsqueda.

1.6 Alcance

El presente proyecto de tesis contendrá un estudio teórico completo para tomar la decisión correcta y así elegir cuál es el “framework” de búsqueda global adecuado para el sistema Gestor Fiducia Fondos de acuerdo a las necesidades planteadas. Para ello se ha escogido dos “frameworks” que se han considerado importantes para implementar, como son: Oracle Secure Enterprise Search y Lucene.

Primero se investigarán las tecnologías antes mencionadas y luego se realizará la selección correcta e integrarla en el sistema Gestor Fiducia Fondos dando una mayor facilidad y seguridad para todos quienes lo utilizan.

Se realizará un diseño del funcionamiento del “framework” seleccionado implementado en el sistema Gestor Fiducia Fondos desde la solicitud del usuario, la conexión con el “framework”, la conexión con la base de datos y la respuesta de la búsqueda, para entender mejor el funcionamiento.

CAPÍTULO 2

LA EMPRESA GESTOR Y SU PRODUCTO GESTOR FIDUCIA FONDOS JEE

2.1 La Empresa Gestor

2.1.1 Historia

GESTORINCSA S.A. es una empresa líder en el desarrollo de soluciones tecnológicas ofreciendo software de calidad, en sus inicios fue solo para el mercado Nacional en la Industria Financiera de Banca de Inversión y Fideicomisos.

En el año 2000 la empresa ganó espacio en el mercado internacional empezando por los países centroamericanos como Guatemala y México, actualmente cuenta con 59 clientes y se encuentra en los siguientes países: Argentina, Colombia, Costa Rica, Ecuador, Guatemala, Honduras, Nicaragua, México, Panamá, Perú, Uruguay, República Dominicana y Venezuela.

La empresa con más de 16 años de experiencia, fundada en el mes de junio de 1997, ofrece un producto que reúne las últimas metodologías y definiciones de negocio de los siguientes tipos de negocios (productos) que ofrecen las instituciones dedicadas a la banca de inversión como: Fondos de Inversión, Ahorro y Pensión, Titularización, Portafolios, Casa de Bolsa y Wealth/Asset Management.

2.1.2 Misión y Valores

MISIÓN

Generar WOWs.[1]

VALORES

La Empresa Gestor ha considerado 5 valores importantes para su desempeño en el ámbito laboral como son:

- **Integridad:** Cumplimos nuestra misión corporativa actuando con honestidad, transparencia, respeto y justicia, encaminando nuestras acciones al crecimiento sostenible de nuestros grupos de interés.

- **Excelencia:** Entregamos lo que ofrecemos y agregamos valor más allá de lo esperado, forjándonos como líderes con vocación innovadora en todo lo que hacemos**.**

- **Compromiso:** Sentimos como propios los objetivos de la organización y cumplimos nuestros compromisos profesionales y personales enfocándonos en asumir retos desafiantes**.**

- **Trabajo en Equipo:** El éxito de Gestor para alcanzar metas y objetivos compartidos, se basa en la cooperación y coordinación del trabajo conjunto con colaboradores y clientes**.**

[1]Nota de Autor: De acuerdo a los ejecutivos de Gestor, un "WOW" está identificado con la expresión de: satisfacción, sorpresa y gusto que un cliente puede tener al recibir un servicio, utilizar una funcionalidad o al conocer el producto Gestor; de acuerdo a esto se podría entender que la misión de Gestor es generar satisfacción y sorpresa entre sus clientes.

- **Proactividad:** Actuamos con criterio de antelación, determinando las causas y efectos de nuestras decisiones y asumiendo con coraje y visión los retos y los resultados. Estamos abiertos a la innovación que nos conduzca al mejoramiento continuo de nuestros procesos, productos y servicios.

2.1.3 Organización de la empresa

Para una mejor organización la Presidencia Ejecutiva de Gestor ha visto la necesidad de hacer una división entre las diferentes unidades de la empresa, para esto ha creado dos grupos principales, el primero agrupa a todas las unidades que son parte fundamental del negocio de la empresa, es decir las unidades encargadas de la comercialización, desarrollo, soporte y gestión de proyectos de implantación relacionados al producto Gestor Fiducia Fondos y a los nuevos productos que forman parte del ecosistema Gestor; el segundo agrupa a las unidades encargadas de soportar el negocio de la compañía y asegurar el normal desenvolvimiento de la misma. La Figura 2.1 muestra la estructura organizacional de Gestor, con sus dos principales grupos y las unidades que los conforman.

Gestor cuenta con las siguientes unidades Estratégicas y Corporativas: Unidad Estratégica de Comercialización, Unidad Estratégica de Servicio al Cliente, Unidad Estratégica de Consultoría, Unidad Estratégica de Desarrollo, Unidad Estratégica de Formación, Unidad Corporativa de Talento Humano, Unidad Corporativa de Sistemas, Unidad Corporativa de Productividad y Calidad, Unidad Corporativa de Administración y Finanzas y Unidad Corporativa de Marketing y Comunicaciones.

Actualmente Gestor cuenta con dos oficinas en el país una en la Ciudad de Quito y otra en Guayaquil.

Figura 2. 1 Estructura Organizacional de Gestor.

2.2 Gestor Fiducia Fondos JEE

2.2.1 Introducción

Gestor contempla una sola solución global, según el área de Banca de Inversión a atender el sistema se divide en cuatro subsistemas dependiendo su uso y modularidad: Gestor Fiducia, Gestor Fondos, Gestor Fondos de Retiro y Pensiones, Gestor Valores y Gestor Inversiones.

2.2.2 Ventajas de Gestor Fiducia Fondos JEE

- **Enfoque:** Brinda una solución Fiduciaria construida de manera modular y cubre la totalidad del Negocio Fiduciario, con los módulos que se utilizan por demanda y necesidad.

- **Flexibilidad y Control:** Creación flexible de Productos Fiduciarios (Negocio) de manera dinámica sin restricción de tipos de productos, con el control sobre el negocio y toda su operación, y altamente paramétrico.

- **100% Operación:** Cubre el Front y Back Office del negocio de punta a punta, sin áreas desatendidas y no contempladas en la operación, incluyendo el proceso contable automático y flexible.

- **Integridad:** Existe la integridad natural entre módulos, procesos y funcionalidades, contemplando un solo input.

- **Canales/Autoservicio:** Posibilidad de incorporar Canales de Interacción con el Cliente, vía Internet, IVR, Cajeros, Call Center.

- **Históricos:** Manejo ordenado y bajo integridad de históricos de información, en los distintos módulos y funcionalidades del sistema.

- **Estados de Cuenta:** Manejo flexible de estados de cuenta, con detalle de información requerido y posibilidad de generación a fechas anteriores.

- **Inteligencia de Negocios:** Modelo de Datos Documentado e Integral, utilizando herramientas de explotación de información.

- **Tecnología:** Solución WebEnabled (acceso vía browser), bajo tecnología Oracle y con un Road Map sustentando su evolución hacia una arquitectura orientada a servicios.

- **Política de Sustentabilidad:** Los productos de Gestor se basan en procesos y políticas claramente definidas. La empresa Gestor mantiene una relación a largo plazo con sus clientes mediante un plan de sustentabilidad que incorpora los avances tecnológicos, operacionales y funcionales en sus productos, basados en un enfoque total de versión única.

2.2.3 Características Tecnológicas del producto Gestor Fiducia Fondos

El sistema cuenta con las siguientes características tecnológicas más actualizadas del mercado:

- Sistema integrado modular.
- Navegación flexible e intuitiva.
- Web-Enabled: acceso vía Browser.
- Soporte Business Intelligence: permite utilización de cubos de información para generación de reportes gerenciales, de negocio y operativos.
- Esquema Single-Deployment con Administración centralizada, basado en servidor de aplicaciones.
- Ayuda en línea.
- Control total del usuario final.
- Altamente parametrizable.
- Multiempresa, Multimoneda, Multiportafolio y Multiagencia.
- Sólidos Controles de Seguridad:
    - Varios niveles de acceso de acuerdo al perfil del usuario.
    - Auditoría completa de cambios.
- Sistema Operativo de Red: UNIX, Windows, Linux.
- Plataforma Actual: Base de Datos ORACLE 11g, Java JEE 6 y servidor de Aplicaciones Weblogic 12c. y Websphere 8.5
- Road Map: Proceso de evolución hacia una Arquitectura Orientada a Servicios, bajo esquema de versión única.

2.2.4 Enfoques de Producto Gestor

Gestor Fiducia: Cubre el área de fideicomisos (inversión, inmobiliario, de garantía, administración y pagos, titularización incluyendo la administración en sí, del activo a titularizar, encargos fiduciarios, etc.), tanto

en la administración de los fideicomisos como tales, así como de la propia operación del fiduciarios.

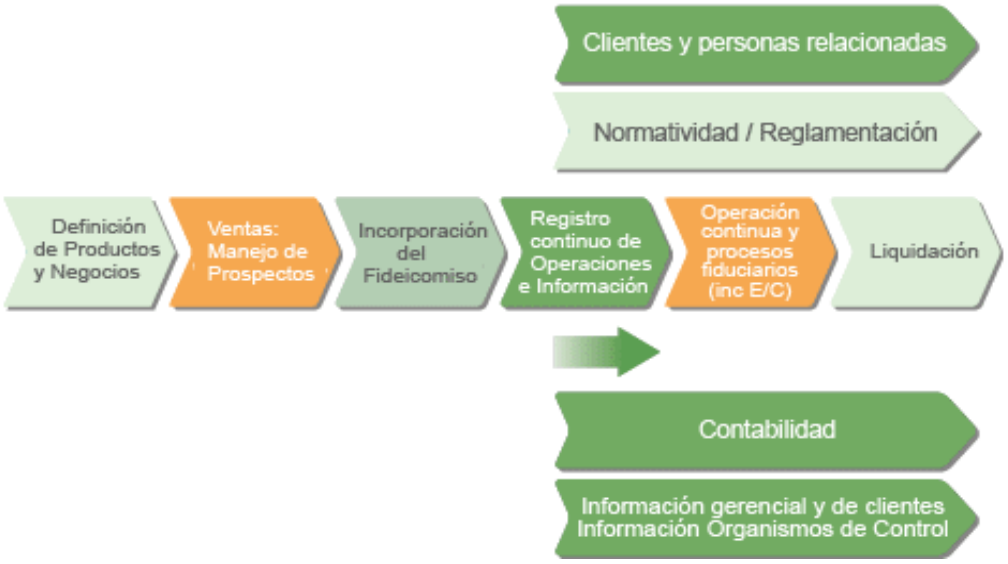

Figura 2. 2 Enfoque de la Solución de Gestor Fiducia.

Gestor Fondos: cubre el área de fondos de inversión, fondos mutuos, así como el fondo de pensiones y retiros. Cubre tanto la administración del pasivo como del activo del fondo, y la propia administración de la administradora.

Figura 2. 3 Enfoque de la Solución de Gestor Fondos.

Gestor Fondos de Retiro y Pensiones: cubre tanto la administración del Pasivo como del Activo de un Fondo de Retiro o Pensiones, y la propia administración de la Administradora.

Figura 2. 4 Enfoque de la Solución de Gestor Fondos de Retiro y Pensiones.

Gestor Valores: cubre tanto el área de órdenes, manejo de cuentas individuales, administración de portafolio (con sus distintas áreas y back office), intermediación, manejo total de inversiones (mercado de dinero, mercado de capitales y derivados) y custodia de los valores.

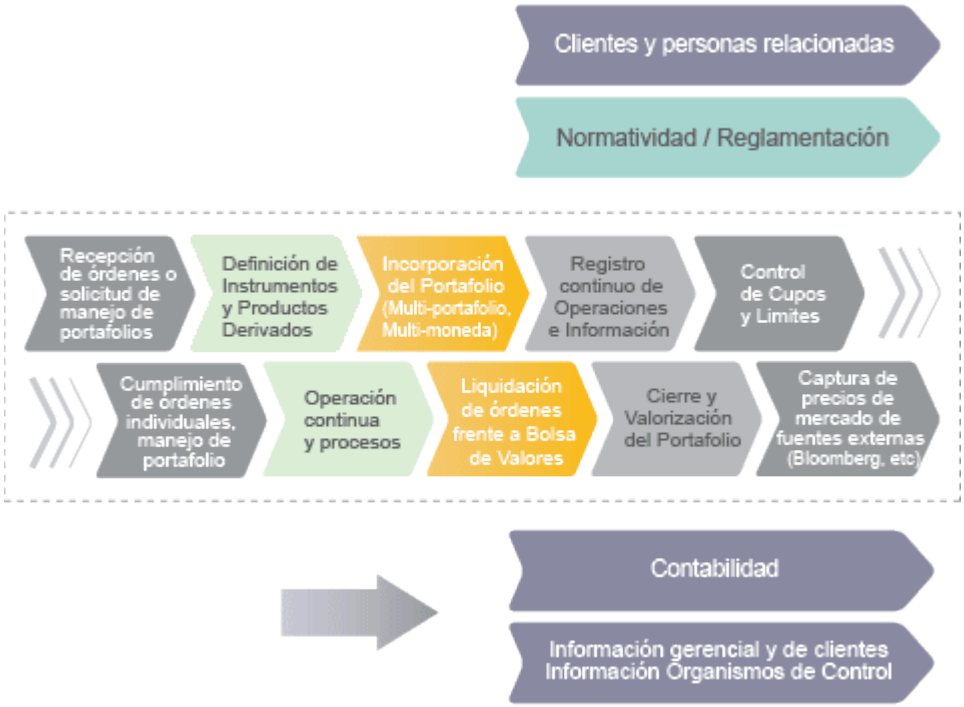

Figura 2. 5 Enfoque de Solución de Gestor Valores.

Gestor Inversiones: cubre tanto el front office, back office, tesorería y contabilidad de la administración de portafolios, incluyendo políticas de inversión, control de riesgos, manejo de liquidez, mesa de dinero, mesa de capitales y mesa de derivados, análisis de simulaciones y proyecciones, así como el manejo de caja (tesorería), con sus procesos operativos y la contabilización.

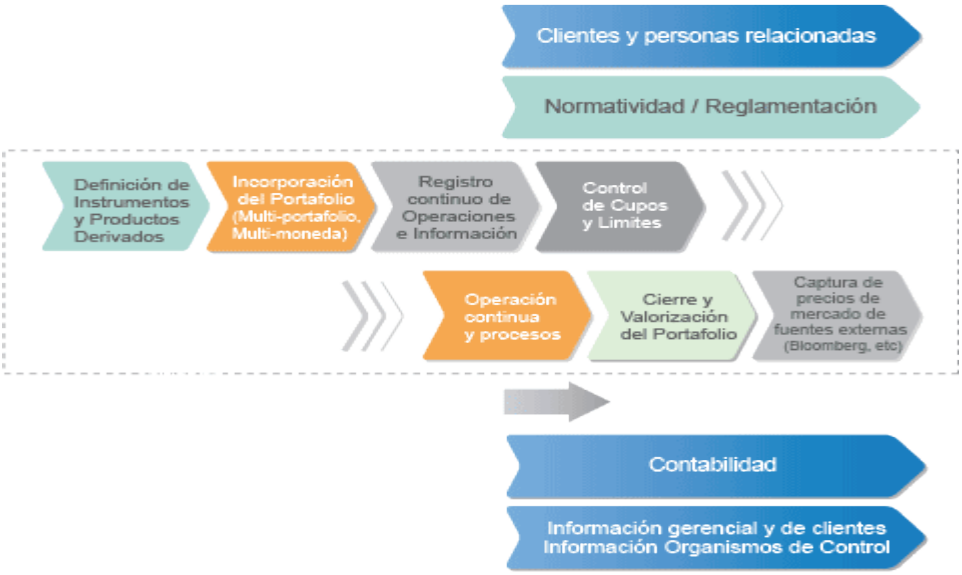

Figura 2. 6 Enfoque de la Solución de Gestor Inversiones.

2.2.5 Módulos y funcionalidades asociadas del Sistema

Figura 2. 7 Módulos del sistema Gestor Fiducia Fondos.

PERSONAS

- Módulo central donde se registran todas las personas o entidades que interactúan con el sistema en el manejo fiduciario y de fondos de inversión.
- Controla la información global de los clientes y las personas relacionadas de manera integrada.

CONTABILIDAD

- Interactúa con los otros módulos generando de manera automática los movimientos contables correspondientes.
- Contabilidad independiente opcional por cada producto, fideicomiso, encargo fiduciario o fondo de inversión (patrimonio autónomo), contemplando cierre masivo o individual por negocio.
- Plan de cuentas parametrizable por producto.
- Manejo multi-moneda, tomando en cuenta ajuste por diferencial cambiario y ajuste por inflación.

PRODUCTO

- Actúa como punto inicial en la creación y operación de los fideicomisos/fondos.
- Apoya en la parametrización de controles, normas del negocio y transacciones de ingreso y egreso.
- Acepta información masiva a través de Planillas, Archivos de Texto, etc.
- Posibilita el establecimiento de fórmulas de VANU (Valor de la Unidad de Participación).
- Interactúa con otros módulos para particularizar las características de cada fideicomiso o fondo.
- Permite acceso a información de clientes en lo referente a compromisos o asuntos pendientes, relacionando tareas, fechas y responsables.

AUDITORÍA

- Genera una bitácora de auditoría de todos los campos del sistema registrando usuario, equipo, fecha, hora, valor anterior y valor nuevo para las operaciones de ingreso, cambio o eliminación de datos.

SEGURIDADES

- Controles de acceso por usuario, pantalla y/o producto fiduciario en base a perfiles y horarios de trabajo.

PARTÍCIPES

- Permite el registro de Fideicomitentes/Partícipes y Beneficiarios así como el ingreso de Clientes que aportan a un fideicomiso o fondo (Cuentas individuales de aportación).
- Parametriza las transacciones de aporte o retiro, estableciendo controles de horario, control de firmas, co-partícipes, montos máximos, retenciones, disponibilidad, etc.
- Posibilita manejar subcuentas para aportes de partícipe, patronales, voluntarios, extraordinarios y otros.
- Permite la generación de transacciones en volumen, vía archivo parametrizable, para clientes corporativos.

CAJA BANCOS

- Cumple con la función operativa de registro y control de ingresos y egresos, permitiendo diferenciarlos por tipos.
- Conciliación bancaria automática.
- Generación de Comprobantes de Retención y Comprobantes de Ingreso (Facturas), con sus procesos de control y consulta.

TITULARIZACIÓN

- Administración de pasivo-patrimonio a través de sus sub-módulos: Libro de Accionistas, Dividendos y Emisión de Deuda Primaria (Captaciones).
- Manejo del activo a titularizar a través de los módulos de Crédito, Inversiones, Bienes, Presupuesto, Caja y Bancos, según el tipo de activo a titularizar.
- Permite registrar las transacciones de “Emisión Primaria” realizadas respecto a la propiedad de los diferentes títulos de participación que se emiten.
- Registra las transacciones de “Mercado Secundario” que se realicen con los títulos valores llevando un histórico de la propiedad de cada uno de ellos y permitiendo el agrupamiento o división según las negociaciones que se efectúen.
- Posibilita la generación de dividendos registrando dividendo acción y/o dividendo efectivo.
- Admite que los patrimonios autónomos generen títulos de deuda según los diferentes productos que se desee crear.

GARANTÍA

- Posibilita generar, administrar, consultar y controlar los Certificados de Garantía con sus respectivas condiciones y características totales.
- Verifica los certificados emitidos frente a los valores de los activos registrados y sus diferentes porcentajes de garantía, así como la amortización de capital de los créditos.
- Registra el crédito relacionado a la garantía en el Módulo de Crédito, sin contabilizarlo dentro del fideicomiso, para controlar el avance de los pagos.

CRÉDITO

- Permite el registro y administración de cartera como un activo del patrimonio, así como créditos de compra de bienes en Fideicomisos Inmobiliarios o de Administración y adelantos de dinero vía crédito en Fondos o Fideicomisos de Inversión. Además registra los créditos de Fideicomisos en Garantía.
- Almacena las tablas de amortización y/o plan de pagos incluyendo intereses, moras, dividendos y moneda de las obligaciones, así como valores adicionales al crédito.
- Controla el cumplimiento de pagos y calcula mora/multas en caso de existir.
- Ajusta la capacidad de garantías (Certificados de Garantía) con base a la amortización de capital.

BIENES

- Registra el aporte o inclusión de diferentes tipos de bienes al patrimonio autónomo del producto.
- Registra para cada bien sus avalúos, seguros, impuestos, descripciones técnicas, documentación legal y de propiedad, propietarios y porcentajes de participación.
- Soporte a imágenes que permitan identificar y registrar el bien a través de medios gráfico y fotográfico.
- Mantiene el detalle de los bienes, permitiendo una rápida generación del inventario a vender y una integración hacia los presupuestos establecidos.

INVERSIONES

- Incluye: Políticas de Portafolio, Control de Riesgos, Manejo de Liquidez, Mesa de Dinero, Mesa de Capitales, Mesa de Cambios/Divisas y Derivados.
- Permite la generación de captaciones como Tesorería Pasiva (Emisión de Obligaciones).
- Manejo del flujo de fondos individual y consolidado. Inversiones individuales y consolidadas.
- Proyección del flujo de caja entre períodos de tiempo y planificación de las necesidades de liquidez de los fondos.
- Utilización en línea de herramientas financieras para apoyo en cálculo y negociación.
- Realiza el control de los títulos entregados en custodia.

MENSAJERÍA

- Permite manejar y controlar diversidad de compromisos de la administración fiduciaria a través de un utilitario que recuerda actividades, controla su lectura y confirma su ejecución.
- Permite registro de actividades de manera directa, así como generación automática desde los demás módulos del sistema especificando fecha, periodicidad, responsables y mensaje a recordar.

COMITÉS

- Maneja la formación de comités, sus convocatorias, actas y resoluciones.
- Registra las resoluciones de comités fiduciarios, señalando responsables, fechas límites y seguimiento necesario.

- Interrelación con Mensajería para el control de cumplimiento de las tareas establecidas en los comités.

CONTRATOS

- Posibilita almacenar e imprimir formatos de contratos por tipo de fideicomiso.
- Permite manejar una base de datos de cláusulas para la mejor administración y generación de contratos.
- Permite generar tareas controladas por Mensajería, relacionadas a las cláusulas.

HONORARIOS / COMISIONES

- Registro de una tabla universal de comisiones, tomando en cuenta condiciones particulares y específicas de cada fideicomiso o fondo.
- Se administra a través de fórmulas que permiten conformar las diferentes comisiones diseñadas directamente por el usuario final.

PROYECTOS / PRESUPUESTOS

- Permite registrar y controlar un presupuesto sobre cualquier proyecto y efectuar comparaciones con el movimiento real
- Alimentación de datos reales a través de la contabilidad así como control de movimientos de caja vs. presupuesto.

PROCESOS LEGALES

- Permite el registro de procesos legales tanto de la Administradora como de sus patrimonios administrados (fideicomisos o fondos), estableciendo el detalle del proceso y su control.

COMPROMISO DE PAGO

- Registro y manejo de contratos con terceros y sus respectivos desembolsos de acuerdo a los términos acordados. Control de calendarios, montos, impuestos, retenciones y excepciones, así como documentos, avisos y notificaciones (integración con Mensajería).

FONDO ROTATIVO

- Permite la entrega de fondos de obra o de contrato para que terceros administren gastos del fideicomiso y una vez reportados, se registre el detalle de lo gastado para su control y reembolso.

PUNTO DE EQUILIBRIO

- Posibilita el registro de proyectos de los fideicomisos, sus fases, etapas y puntos de equilibrio tanto cuantitativos (en base a fórmulas) como de cumplimiento de actividades.

LIQUIDACIÓN

- Controla diversos pendientes antes de concretar el estado de liquidación a un fideicomiso: Cuentas por pagar, Cuentas por cobrar, Garantías, Actividades y documentos, etc.

FRONT OFFICE

Manejo de Ordenes de Clientes

- Registro de órdenes de compra y venta de valores, para intermediación de valores o para portafolio administrado.

- Registro de Liquidaciones parciales o totales de las órdenes de clientes.

Mercado de Dinero (Renta Fija)

- Funcionalidad completa para el registro y control de las inversiones en instrumentos de renta fija.
- Registro de operaciones de Compra/Venta de instrumentos y el cobro de interés y/o capital a su vencimiento.
- Manejo de instrumentos del mercado local e internacional, tales como: Depósitos a Plazo, Papeles Comerciales, Activos Titulizados, Corporativos, Gobierno, Bonos del Tesoro, Bonos Hipotecarios, Bonos Soberanos, Otros Bonos, etc.
- Control en línea de Cupos y Límites.
- Herramientas de apoyo como Simulaciones, Carteras Tipo, Benchmark.

Mercado de Capitales (Renta Variable)

- Funcionalidad completa para el registro y control de las inversiones en instrumentos de renta variable.
- Registro de operaciones de Compra/Venta de instrumentos, suscripción y cobro de dividendos.
- Manejo de instrumentos del mercado local e internacional, tales como: Acciones y Valores registrados en Bolsa, Acciones Comunes y Preferentes, Cuotas de participación en Fondos Mutuos y de Inversión, Certificados de Suscripción Preferente, etc.
- Control en línea de Cupos y Límites.
- Herramientas de apoyo como Simulaciones, Carteras Tipo, Benchmark.

Derivados

- Registro y control de Derivados Estándares, OTC, Forwards y Swaps.
- Registro de operaciones de Compra/Venta de Opciones de Compra (call), Compra/Venta de Opciones de Venta (Put), Compra/Venta de Futuros.
- Liquidación y Margen.

Divisas

- Registro y seguimiento de transacciones de compra/venta de divisas.
- Soporte multi-moneda y multi-agencia.
- Control de línea de Cupos y Límites.

Tesorería

- Registro de transacciones de ingreso a cuentas bancarias.
- Registro de transacciones de egreso de cuentas bancarias.
- Registro de transferencias entre cuentas bancarias.

CAPÍTULO 3

MARCO TEÓRICO

3.1 Framework

3.1.1 Definición

Un “framework”[2] es un conjunto de librerías que no hacen parte de una aplicación específica y que han sido creados para ser utilizados por cualquier aplicación.

3.1.2 Características

Un “framework” o marco de trabajo se caracteriza por ser robusto y sus componentes pueden ser usados para varios propósitos ya que están probados para funcionar en cualquier aplicación que se necesite.

Puede ser capaz de soportar aplicaciones de cualquier tipo y tamaño y contiene componentes que son utilizados en aplicaciones completas, en el caso que no existieran, el “framework” posee parámetros definidos para crear componentes que se integren fácilmente con el resto de funcionalidades, esto quiere decir que es extensible.

3.2 Definición de búsqueda

La búsqueda es la capacidad de realizar consultas sobre los datos almacenados en una base de datos para obtener documentos que contengan información valiosa para el usuario.

[2]Framework: Marco de Trabajo.

Cuando los usuarios interactúan con un sistema de información, ellos necesitan acceder a la información. Los sistemas de información modernos tienden a dar a los usuarios acceso a más y más datos. Saber exactamente dónde encontrar lo que se está buscando es el caso extremo de búsqueda. Antes de saber dónde buscar, es necesario tener una idea de lo que se está buscando.

3.3 Gestión Documental

La práctica de la gestión documental está dada por un conjunto definido de normas técnicas y algunas prácticas generales que son usadas para:

- La administración del flujo de documentos de todo tipo en una organización.
- La recuperación dela información.
- Especificar en qué tiempo deben almacenarse los documentos.
- La eliminación de los documentos que ya no sirven.
- La conservación de los todos documentos que se consideren valiosos indefinidamente.

3.3.1 Componentes

3.3.1.1 Base de Datos

Las bases de datos hoy en día son un componente indispensable en las organizaciones de todos los países y en la administración de sus diferentes áreas, una de esas áreas es la gestión documental que están sustituyendo a los documentos físicos y son un soporte de información valiosa para la organización.

3.3.1.2 Hardware

Dispositivos de Digitalización y escáneres: Los documentos físicos son preparados para ser transformados en documentos digitales por medio de dispositivos especializados, los cuales serán guardados en un repositorio de la organización.

Servidores: Son los que contienen todos los documentos previamente digitalizados. Los usuarios finales ingresaron a un servidor para acceder a la información digitalizada para la revisión o modificación.

3.3.1.3 Software

Gestores Documentales: Son aplicaciones especializadas para el proceso de la gestión documental que maneja una empresa. Existen otras soluciones informáticas para la administración de archivos digitales, gestión documental y la administración de todo tipo de bibliotecas.

3.3.1.4 Redes

Los usuarios por medio de las redes pueden acceder a toda la información que se encuentra almacenada en los servidores. Existen dos tipos de redes en las cuales se pueden acceder a la información como son las locales y por medio del Internet.

3.3.1.5 Usuarios

Mediante una cuenta los usuarios pueden acceder a los documentos digitalizados en el sistema de gestión documental y así pueden realizar las consultas electrónicamente de acuerdo a los permisos que se le haya asignado al usuario que exista en el sistema.

3.3.1.6 Administradores

Los administradores de los sistemas son los encargados de realizar la codificación e indexación en las bases de datos de los servidores, identifica donde se encuentran ubicados físicamente los documentos originales y asignan lógicamente a cada documento claves para que puedan acceder a ellos con seguridad.

3.3.2 Ventajas

- Rápido y fácil acceso
    - Acceso inmediato la documentación almacenada.
    - Reduce el tiempo de consultas.
    - Resuelve el problema de localización.
    - Posee control total de toda la documentación e información.
    - Se puede compartir la información entre los diferentes usuarios.
    - Se facilita la rápida distribución o envío de documentos.

- Ahorro de materiales
    - Se puede ahorrar en la impresión de documentos (copias, impresiones, etc.).
    - Se ahorra el espacio físico.

- Alta seguridad y fiabilidad
    - Para documentos de gran valor o confidenciales posee una custodia de alta seguridad.
    - Sustituye los documentos impresos por réplicas electrónicas.
    - No se duplican los documentos.

3.3.3 Desventajas

- La empresa que desee este sistema debe estar dispuesta a invertir una gran suma monetaria.

- Los protocolos de seguridad deben ser configurados correctamente y no deben estar debajo del nivel aceptable porque las personas no autorizadas pueden obtener ilícitamente los datos más sensibles desde una computadora o servidor.

- Se debe definir planes de redundancia (es decir, siempre se debe hacer respaldos en un servidor secundario que esté localizado en diferente lugar) en caso de robo, incendio o alguna catástrofe natural y evitar perder la información, para después poder recuperar los datos.

- Se corre el riesgo de violar alguna ley o algunos otros reglamentos de rastreo y resguardo de la información.

3.4 Algoritmos de búsqueda

Un algoritmo de búsqueda es un algoritmo para encontrar un elemento con propiedades especificas entre una colección de elementos. Los elementos pueden ser almacenados individualmente como registros en una base de datos, o pueden ser elementos de un espacio de búsqueda definidos por una fórmula o procedimiento matemático, tales como las raíces de una ecuación con variables enteras o una combinación de los dos.

Hay dos tipos de algoritmos de búsqueda: algoritmos que no hacen suposiciones sobre el orden de la lista y los algoritmos que asumen que la lista ya está en orden.

3.4.1 Búsqueda Lineal

El algoritmo de búsqueda más simple es la búsqueda lineal. En la búsqueda lineal, se fija en cada elemento de la lista, y a la vez, encuentra un elemento que coincida con el término de búsqueda o una vez que se ha llegado al final de la lista. El "valor de retorno" es el índice en el que se encontró el término de búsqueda, o de algún indicador de que el término de búsqueda no se encontró en la lista.

3.4.1.1 Algoritmo

Para (cada elemento de la lista) {
    Compara el término de búsqueda para el elemento actual.
    Si lo encuentra,
        Guarda el índice del elemento que coincide.
        Sale de la condición.
}

Retorna el índice del elemento que coincide, o retorna -1 si no se encuentra el elemento.

3.4.1.2 Implementación en Java

```
public int sequentialSearch(int item, int[] list) {
    // Si el índice todavía es -1 al final de todo este método, el elemento
    no está en esta lista
    Int index = -1;

    // bucle a través de cada elemento de la matriz. Si encontramos el
    término de búsqueda, salir del bucle.
```

```
        for (int i=0; i<list.length; i++) {
                if (list[i] == item) {
                        index = i;
                        break;
                }
        }
        Return index;
}
```

3.4.1.3 Rendimiento de la búsqueda Lineal

Cuando se comparan los algoritmos de búsqueda, sólo se toma en cuenta el número de comparaciones, ya que no se cambia ningún valor durante la búsqueda. A menudo, cuando se compara el rendimiento, se debe tomar en cuenta tres casos:

- Mejor caso: ¿Cuál es el menor número de comparaciones necesarias para encontrar un artículo?
- Peor de los casos: ¿Cuál es el mayor número de comparaciones necesarias para encontrar un artículo?
- Caso medio: En promedio, ¿cuántas comparaciones se tarda en encontrar un elemento en la lista?

Para la búsqueda lineal, los casos serían los siguientes:

- Mejor caso: El mejor de los casos se produce cuando el término de búsqueda se encuentra en la primera ranura de la matriz.
- Peor de los casos: El peor caso se produce cuando el término de búsqueda se encuentra en la última ranura de la matriz, o no está en la matriz. Si la matriz tiene N elementos, entonces se necesita N comparaciones en el peor de los casos.

- Caso media: En promedio, el término de búsqueda será en algún lugar en medio de la matriz. El número de comparaciones sería aproximadamente N / 2.

En ambos el peor de los casos y el caso promedio, el número de comparaciones es proporcional al número de elementos de la matriz, N. Por lo tanto, decir que en estos dos casos que el número de comparaciones es orden N o 0 (N) para acortar. Para el mejor de los casos, se dice que el número de comparaciones es para 1 o 0 (1), para abreviar.

3.4.2 Búsqueda Binaria

La búsqueda lineal funciona bien en muchos casos, sobre todo si no se sabe si la lista está en orden. Su único inconveniente es que puede ser lento. Si N, el número de elementos en la lista, es de 1.000.000, entonces puede tomar mucho tiempo, en promedio, para encontrar el término de búsqueda en la lista (en promedio, que se llevará 500.000 comparaciones).

La búsqueda binaria explota el ordenamiento de una lista. La idea detrás de la búsqueda binaria es que cada vez que hacemos una comparación, se elimina la mitad de la lista, hasta que encontramos el término de búsqueda o determinamos que el término no está en la lista. Hacemos esto al mirar el punto medio de la lista, y determinar si nuestro término de búsqueda es más alto o más bajo que el punto medio.

Si es inferior, se elimina la mitad superior de la lista y se repite la búsqueda empezando por el punto medio entre el primer elemento y el elemento central. Si es superior, se elimina la mitad inferior de la lista y se repite la búsqueda empezando en el punto medio entre el punto medio y el último elemento.

3.4.2.1 Algoritmo

Inicializa las variables primero = 1, último = N, mid = N/2
Mientras (no encuentre el elemento y primero sea menor que último) {
    Compara los términos de búsqueda con el elemento mid
    Si coincide
        Guarda el índice
        Finaliza
    Caso contrario Si el término de búsqueda es menos que el elemento mid,
        Asigna a la variable último = mid-1
    Caso contrario
        Asigna a la variable primero = mid+1
        Asigna a la variable mid = (primero + último)/2
}
Retorna el índice del elemento coincidente, o retorna -1 si no se encuentra elemento.

3.4.2.2 Implementación en Java

```
public int binarySearch(int item, int[] list) {
    // Si index = -1 cuando se termina el procedimiento, no se encontró el término de búsqueda en la matriz
    Int index = -1;

    // Establece los índices de inicio y final de la matriz, los cuales cambiarán a medida que se reduce la búsqueda
    int low = 0;
    int high = list.length-1;
    int mid;
    // Continua para buscar el término de búsqueda hasta encontrarlo o hasta que nuestros marcadores estén ' bajos ' y ' altos'
```

```
        while (high>= low) {
                mid = (high + low)/2;// calcular el punto medio de la matriz
                actual
                if (item<list[mid]) { // valor se encuentra en la mitad inferior
                        high = mid - 1;
                } else if (item >list[mid]) {
                        // valor está en la mitad superior
                        low = mid + 1;
                } else {
                        // encontrado ! rompe el bucle
                        index = mid;
                        break;
                }
        }
        return index;
}
```

3.4.2.3 Rendimiento de la Búsqueda Binaria

El mejor caso para búsqueda binaria todavía ocurre cuando se encuentra el término de búsqueda en el primer intento. En este caso, el término de búsqueda sería en el medio de la lista. Al igual que con la búsqueda lineal, el mejor de los casos para la búsqueda binaria es 0 (1), ya que tiene exactamente una comparación para encontrar el término de búsqueda en la lista.

El peor de los casos para la búsqueda binaria se produce cuando el término de búsqueda no está en la lista, o cuando el término de búsqueda es un elemento lejos de la mitad de la lista, o cuando el término de búsqueda es el primero o el último elemento de la lista.

El promedio de los casos se produce cuando el término de búsqueda es en cualquier otro lugar en la lista. El número de comparaciones es más o menos el mismo que para el peor de los casos, por lo que también es 0 (logN). En general, en cualquier momento un algoritmo implica dividir una lista en la mitad y el número de operaciones es 0 (log N).

3.5 Indexación

En la Figura 3.1 se puede ver el proceso de indexación que consiste en analizar y extraer de entre toda la información disponible, la verdaderamente relevante. Posteriormente, con esa información se crea el índice a partir del cual se realizarán las búsquedas. El índice es una estructura de datos que permite el acceso rápido a la información, algo similar semánticamente a lo que podría ser el índice de un libro.

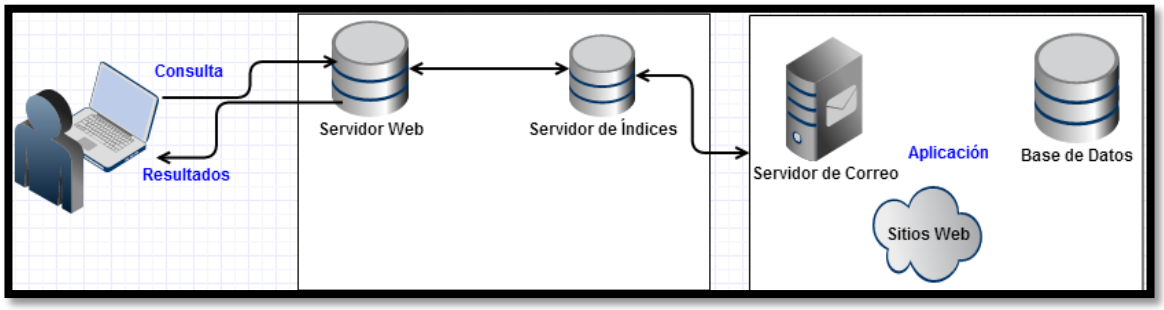

Figura 3. 1 Proceso de Indexación

3.5.1 Índices invertidos

Muchos motores de búsqueda incorporan un índice invertido en la evaluación de una consulta de búsqueda para localizar rápidamente los documentos que contengan las palabras en una consulta y luego clasificar los documentos por fecha. Debido a que el índice invertido almacena una lista de los documentos que contienen cada palabra, el motor de búsqueda puede utilizar el acceso directo para encontrar los documentos asociados a

cada palabra de la consulta con el fin de recuperar los documentos que coinciden con rapidez.

La siguiente tabla es una ilustración simplificada de un índice invertido:

Tabla 3. 1

Índice Invertido

Palabra	Documentos
Los	Documento 1, Documento 2
Carros	Documento 3, Documento 1

Este índice sólo puede determinar si existe una palabra dentro de un documento en particular, ya que almacena ninguna información con respecto a la frecuencia y la posición de la palabra, por lo que se considera que es un índice booleano. Este índice determina qué documentos coincide con una consulta pero no clasifica los documentos coincidentes. En algunos diseños el índice incluye información adicional, como la frecuencia de cada palabra en cada documento o las posiciones de una palabra en cada documento.

La información de posición permite que el algoritmo de búsqueda identifique la palabra más próxima para apoyar la búsqueda de frases; la frecuencia se puede utilizar para ayudar en el ranking de la relevancia de los documentos a la consulta. Tales temas son el foco central de la investigación de la recuperación de información.

El índice invertido es una matriz dispersa, ya que no todas las palabras están presentes en cada documento. Para reducir los requisitos de memoria de almacenamiento de ordenador, se almacena de manera diferente a partir de una matriz de dos dimensiones. El índice es similar a las matrices de documentos empleados por el análisis semántico latente.

3.5.2 Fusión de Índices

El índice invertido se llena a través de una fusión o reconstrucción. La reconstrucción es similar a una fusión, pero elimina primero el contenido del índice invertido.

La arquitectura puede ser diseñada para apoyar la indexación gradual, donde una combinación identifica el documento o los documentos que se agregan o se actualizan y luego analiza cada documento en palabras. Para mayor precisión técnica, una combinación fusiona documentos recientemente indexados, por lo general reside en la memoria virtual, con el caché de índice que reside en los discos duros de una o más computadoras.

Después del análisis, el indexador agrega el documento referenciado a la lista de documentos para las palabras adecuadas. En un motor de búsqueda más grande, el proceso de búsqueda de cada palabra en el índice invertido (con el fin de informar de que se produjo dentro de un documento) puede ser demasiado lento, por lo que este proceso se suele dividir en dos partes, el desarrollo de un índice hacia adelante y un proceso que ordena el contenido del índice hacia adelante en el índice invertido. El índice invertido se llama así porque es una inversión del índice hacia adelante.

3.5.3 Índice hacia adelante

El índice hacia adelante almacena una lista de palabras para cada documento. La siguiente tabla 3.2 es una forma simplificada del índice hacia adelante:

Tabla 3. 2

Índice hacia adelante

Documento	Palabras
Documento 1	Los,perros,ladran
Documento 2	El,gato,con,botas
Documento 3	Los,planetas

El fundamento de la elaboración de un índice hacia adelante es que a medida que los documentos se están analizando, es mejor almacenar de inmediato las palabras por documento. La delimitación permite el procesamiento del sistema asíncrono, lo que evita en parte el cuello de botella de actualización del índice invertido. El índice de avance está ordenado para transformarlo a un índice invertido.

El índice hacia adelante es esencialmente una lista de pares formados por un documento y una palabra, recopilado por el documento. Convertir el índice de interés con un índice invertido es sólo una cuestión de la clasificación de los pares de las palabras. En este sentido, el índice invertido es un índice de avance de palabras ordenados.

3.5.4 Compresión

Generar o mantener un índice del motor de búsqueda a gran escala representa un almacenamiento significativo y el desafío de procesamiento. Muchos motores de búsqueda utilizan una forma de compresión para reducir el tamaño de los índices en el disco.

Se necesita 8 bits (o 1 byte) para almacenar un solo carácter. Algunas codificaciones utilizan 2 bytes por carácter.

El número medio de caracteres en cualquier palabra en una página, puede estimarse en 5.

Ante este escenario, un índice sin comprimir (suponiendo un no fusionado, simple, índice) por 2 mil millones de páginas web tendría que almacenar 500000000000 entradas de palabras. A 1 byte por carácter, o 5 bytes por palabra, esto requeriría 2.500 gigabytes de espacio de almacenamiento, más de espacio libre en disco promedio de 2 ordenadores personales. Este requisito de espacio puede ser aún más grande para una arquitectura de almacenamiento distribuido de alta disponibilidad. Dependiendo de la técnica de compresión elegida, el índice puede ser reducido a una fracción de este tamaño. La desventaja es el tiempo y la potencia de procesamiento requerida para llevar a cabo la compresión y la descompresión.

Cabe destacar que los diseños de motores de búsqueda a gran escala incorporan el costo de almacenamiento, así como los costos de la electricidad para alimentar el almacenamiento. Por lo tanto la compresión es una medida de costo.

3.6 Oracle Secure Enterprise Search

Oracle Secure Enterprise Search es una aplicación independiente de búsqueda integrada. Las personas mejoran su productividad a través de la búsqueda en múltiples repositorios de forma simultánea. Oracle Secure Enterprise Search se integra con modelos de seguridad populares para mantener su contenido y la seguridad del índice de búsqueda también rastrea el contenido no estructurado y estructurado y devuelve resultados de alta calidad integrados desde los repositorios de intranet y extranet según su configuración. (Oracle, Oracle Secure Enterprise Search 11g, s.f.)

3.6.1 Características de Oracle Secure Enterprise Search (SES)

A continuación se lista las características claves de Oracle Secure Enterprise Search:

- Capacidad para buscar y localizar contenidos públicos, privados y compartidos a través de Intranet servidores web, bases de datos, archivos en el disco local o en servidores de archivos, correo electrónico IMAP, sistemas de gestión de documentos, aplicaciones y portales.
- Una interfaz simple, búsqueda intuitiva que lleva a una excelente experiencia de usuario.
- Excelente calidad de búsqueda, con los temas más relevantes para una consulta que se muestra en primer lugar, aun cuando la consulta se extiende por diversas fuentes de datos públicos o privados.
- Alta seguridad en el rastreo, indexación y búsqueda en los diferentes repositorios.
- Facilidad de administración y mantenimiento aprovechando su experiencia en TI existentes.
- Oracle posee KWIC (palabra clave en contexto) en el resultado de la búsqueda con una restricción en la búsqueda de cuatro mil caracteres. Es decir, si la palabra clave buscada aparece en los primeros cuatro mil caracteres de un documento, aparece en el resultado de la búsqueda. Si la palabra clave aparece después de los cuatro mil caracteres del documento, no se muestra ninguna palabra clave.
- Los caracteres no alfanuméricos en una consulta se consideran como separadores de términos; por ejemplo, #, %, /, etc. Los caracteres internacionales no se consideran caracteres especiales.
- La información de una empresa se puede transmitir a través de las páginas Web, bases de datos, servidores de correo, repositorios de documentos, servidores de archivos y computadoras de escritorio.

Oracle SES busca en todos tus datos a través de la misma interfaz, está totalmente globalizado y trabaja con muchos idiomas, incluyendo chino, japonés, coreano, árabe y hebreo. (Oracle, Características de Oracle Secure Enterprise Search, s.f.)

3.6.2 Requisitos de Software y Hardware

Requisitos de Software

Las versiones compatibles de los navegadores para la administración y consulta de la herramienta de Oracle SES son:

- Firefox 3.x
- Internet Explorer 7.x, 8.x
- Safari 4.x

Oracle SES 11.1.2.2 está certificado para funcionar en los siguientes sistemas operativos de Windows:

- Windows Server 2003 (64-bit).
- Windows Server 2003 R2 (64-bit).
- Windows XP Profesional x64.
- Windows Server 2008 R1 SP1 o service pack más alto (64-bit).

Estas son las únicas versiones y distribuciones soportadas. Oracle SES 11.1.2.2 no se debe instalar en otras versiones de Windows.

Requisitos de Hardware

Oracle SES requiere un mínimo de 2 gigabytes de espacio en disco. Esto incluye 1 gigabyte para instalar y aproximadamente 0,5 gigabytes para crear el índice inicial de Oracle SES.

Los requisitos adicionales de Oracle SES se basan en la cantidad de datos que se necesita para buscar. A continuación se muestran algunos ejemplos de configuración:

Para indexar 100.000 documentos:

- 4 GB de espacio en disco.
- 2 GB de RAM.

Para indexar 1.000.000 documentos:

- 20 GB de espacio en disco.
- 6 GB de memoria RAM.

3.6.3 Búsqueda Segura

Gran parte de la información dentro de una organización es de acceso público, cualquier persona puede verlo. Por lo tanto, es relativamente fácil para un rastreador encontrar e indexar esa información.

Sin embargo, existen otras fuentes que están protegidas. Estas fuentes protegidas podrían ser visibles únicamente por ciertos usuarios o grupos de usuarios. Por ejemplo, aunque los usuarios pueden buscar en sus propias carpetas de correo electrónico, no debería ser capaz de buscar cualquier otro e-mail.

Para las fuentes protegidas, los rastreadores de Oracle SES indexan cualquier documento con la lista de control de acceso apropiado. Cuando los usuarios finales realizan una búsqueda, sólo los documentos que tienen privilegios para ver se devuelven. (Oracle, Búsqueda segura, s.f.)

3.6.3.1 Caché de autorización de usuario

La caché de autorización del usuario (UAC) es un tipo de fuente que puede rastrear e informar sobre la caché de autorización del usuario como los grupos y los valores accesibles de los atributos de seguridad del usuario. Esta información almacenada en la caché se utiliza en tiempo de consulta para construir un filtro de seguridad. Consultar una memoria caché local es mucho más rápida que la recuperación de la información de autorización de un repositorio externo y sistemas de identidad, por lo que reduce significativamente el tiempo para construir los filtros de seguridad para el usuario actual. Como resultado, los usuarios pueden iniciar sesión en Oracle SES mucho más rápidamente. Además, puede configurar orígenes de UAC para rastrear la información de autorización del usuario fuera de línea, lo que reduce la carga en los depósitos de destino en el momento de la consulta.

Se puede utilizar el UAC para las fuentes que se basan en cualquiera de los modelos de seguridad soportados en Oracle SES:

- Seguridad basada en identidad: UAC está habilitado para Oracle Internet Directory, Active Directory y Lotus Notes.
- Basada en atributos de seguridad: UAC está habilitado para Oracle Content Database y Servidor de contenido de fuentes de Oracle. Este tipo de seguridad es también llamado el definido modelo de seguridad por el usuario.

El rastreador almacena la siguiente información en la caché de autorización del usuario:

- Los grupos de usuarios: La lista de los grupos que pertenece un usuario.
- Los valores de atributo de usuario: Los valores de una lista especificada de atributos para las fuentes de datos particulares. Los valores pueden ser valores individuales o series de valores.

3.6.4 Búsqueda federada

Oracle SES puede buscar varias instancias con sus propios repositorios de documentos e índices. Proporciona un marco unificado para buscar en los diferentes repositorios que se rastrean, indexado y mantenido por separado.

La búsqueda federada permite a una sola consulta funcionar en todas las instancias de Oracle SES. Sumando los resultados de la búsqueda para mostrar una lista de resultados unificada para el usuario. Las credenciales de usuario se pasan junto con la consulta de modo que cada punto final de la federación puede autenticar al usuario en contra de su propio repositorio de documentos.

La Figura 3.2 muestra la arquitectura de federación y dos opciones para un usuario final para conectarse a través de un navegador para Oracle SES.

- **Opción 1:** permite a los usuarios conectar sus navegadores directamente a Oracle SES, utiliza la interfaz gráfica de usuario final.
- **Opción 2:** recupera los resultados de Oracle SES a través de Servicios Web después del post-procesamiento, como cambiar la apariencia o la incrustación de los resultados en una página. Para esta opción, el navegador se conecta a las aplicaciones remotas que

se conectan a la API de servicios Web. (Oracle, Búsqueda Federada de Oracle Secure Enterprise Search, s.f.)

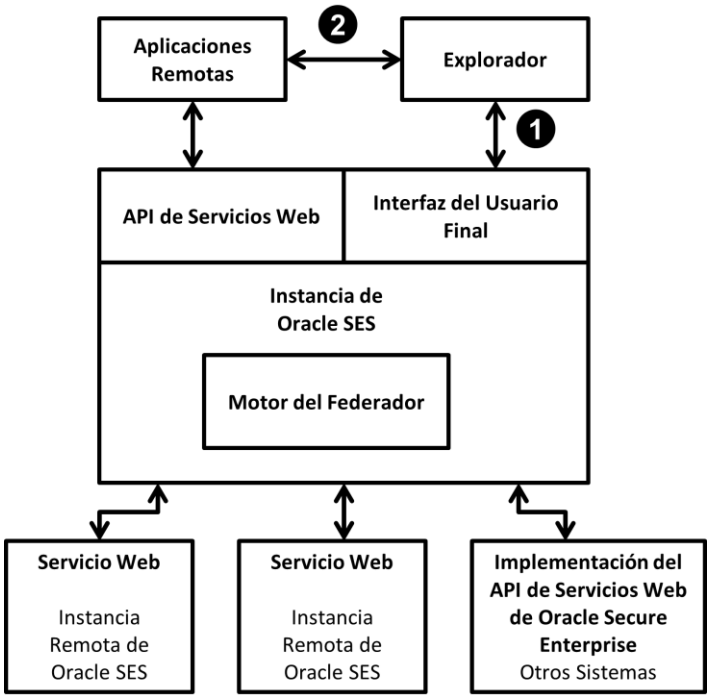

Figura 3. 2 Arquitectura de la Búsqueda Federada

3.6.5 Ventajas

Oracle Secure Enterprise Search brinda las siguientes ventajas principales al utilizarlo:

Máximo nivel de seguridad

Múltiples capas de seguridad para imponer el cumplimiento de la normativa, la protección de IP y preservar la privacidad en el acceso a la información.

La gama más amplia de fuentes de datos empresariales

Acceso inmediato a más aplicaciones que cualquier otro motor de búsqueda e interfaces basadas en estándares para conectarse a fuentes de datos personalizadas.

La rentabilización más rápida

La estética y el comportamiento familiar de la búsqueda en web aseguran una rápida adopción entre los usuarios. La administración basada en web facilita el ajuste de los resultados para proporcionar la información adecuada a los usuarios.

Hot-pluggable

Oracle Secure Enterprise Search posee el sistema de directorio virtual compatible con los sistemas de gestión de Oracle y de terceros, mientras que la búsqueda federada[3] integra los resultados de los motores de búsqueda de terceros.

Buscar en todos los repositorios

Oracle Secure Enterprise Search buscará en todos los repositorios incluyendo extranet, intranet, servidores de archivos, documentos y sistemas de gestión de contenidos, servidores de correo electrónico y bases de datos.

Oracle Secure Enterprise Search también es compatible con fuentes remotas utilizando el marco de la federación, y las fuentes de encargo usando la API de plugins de búsqueda.

[3] Búsqueda Federada: permite establecer relaciones de búsqueda con otras fuentes.

3.6.6 Estructura

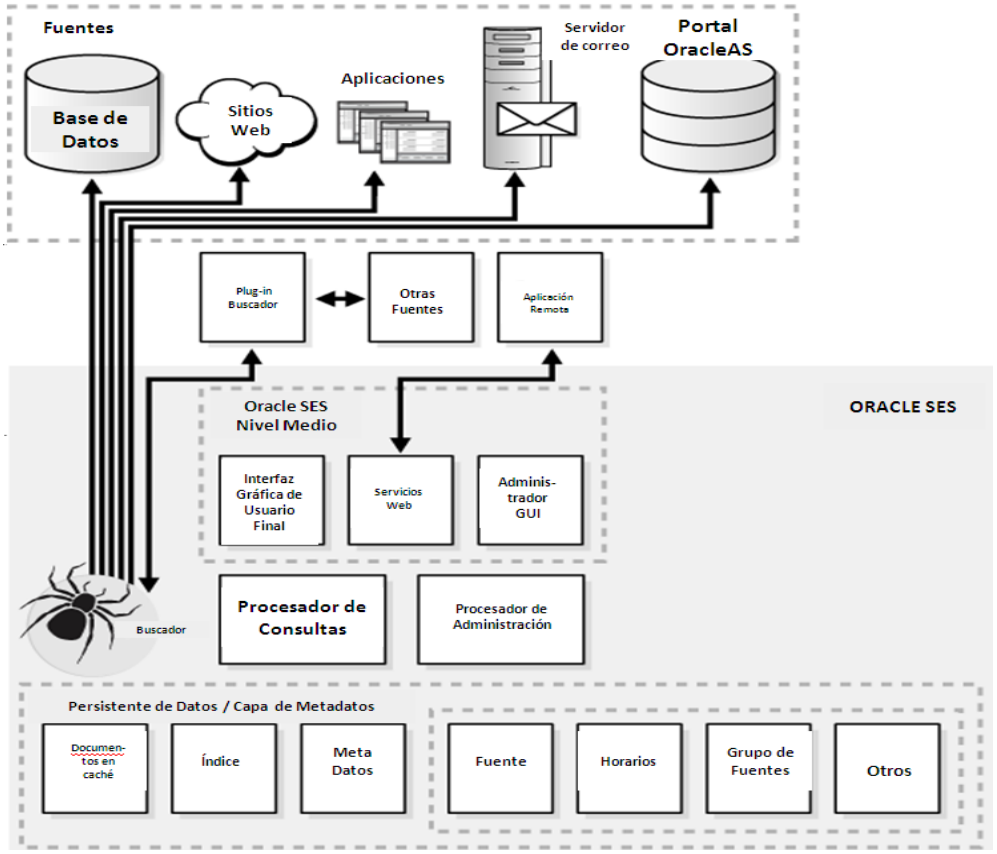

Figura 3. 3 Recopilación de información para Oracle SES[4]

En la Figura 3.3 se puede observar las fuentes remotas en la parte superior izquierda.

Los orígenes remotos son una base de datos, sitios web, aplicaciones, un servidor de correo electrónico, y un servidor de Portal. Un rastreador recopila estas fuentes y las envía a un rastreador plug-in. El buscador o crawler de plug-in puede ir y volver a las fuentes y aplicaciones remotas. El nivel intermedio incluye la interfaz gráfica de usuario final, servicios Web y la herramienta de administración.

[4] Vishwanath Sreeraman, D. C., Cathy Shea, K.R., Michele Cyran, (2013). Oracle® Secure Enterprise SearchAdministrator's Guide, 11g Release 1 (11.1.2.0.0). Estados Unidos: U.S. Government.

En la parte inferior de la gráfica se encuentra la capa de metadatos donde se incluye documentos en caché, el índice, los metadatos, el origen, horarios, grupo de origen, entre otros.

3.6.7 Tipos de Fuentes

Una colección de información se denomina fuente. Cada fuente tiene un tipo que identifica el lugar donde se almacena la información, tal como en un sitio Web o en una tabla de base de datos. Oracle SES proporciona varios tipos de origen incorporados y una arquitectura para la adición de nuevos tipos.

Además, Oracle SES provee acceso a más de terceros repositorios de datos que cualquier otro motor de búsqueda empresarial, sin necesidad de generar ningún código adicional. Si bien estas fuentes de datos se clasifican en tipos definidos por el usuario de origen, que están disponibles como los tipos de fuentes incorporadas.

Oracle SES también proporciona fuentes de autorización de caché para facilitar el acceso a datos seguros. (Oracle, Tipos de Fuentes, s.f.)

3.6.7.1 Fuentes incorporadas

- **Web**: Representa el contenido de un sitio web específico. El rastreo de las fuentes web facilitan el mantenimiento de sitios web específicos.
- **Table**: representa el contenido de una tabla o vista de base de datos Oracle.
- **Expediente**: Es el conjunto de documentos que se pueden acceder a través del protocolo de sistema de archivos.

- **E-mail**: Deriva el contenido de los correos electrónicos enviados a una determinada dirección de correo electrónico. Cuando Oracle SES arrastra una fuente de e-mail, recoge el correo electrónico de todas las carpetas creadas en la cuenta de correo electrónico, incluyendo Borradores, Enviados y Papelera de e-mails.
- **Lista de envío**: Deriva el contenido de los e-mails enviados a una lista de correo específica.
- **Portal OracleAS**: Le permite buscar a través de múltiples repositorios de Portal, como páginas Web, archivos en el disco, y las páginas de otras instancias de Portal.
- **Federados**: Le permite compartir contenido a través de múltiples instancias de Oracle SES.

3.6.8 Búsqueda Básica

La búsqueda en general no es sensible a mayúsculas/minúsculas. Se puede ordenar la forma en la que se presentan los resultados utilizando las listas Agrupar por y Ordenar por.

Oracle SES aplica semejanzas al término de consulta. Esto amplía el término a otros que comparten la misma raíz. Por ejemplo, la consulta [bancos] devuelve documentos que contengan la palabra bancos, banca o banco. Oracle SES utiliza la semejanza basada en el idioma de la consulta. La ampliación de semejanzas implícita se aplica a la búsqueda de términos únicos, a la búsqueda por proximidad y a la búsqueda de método abreviado para el atributo STRING. No se aplica a la búsqueda de frases y se puede desactivar delimitando el término entre comillas.

Oracle SES también realiza la ampliación de palabras alternativas implícitas. Cuando se amplía un par de palabras alternativas, Oracle SES muestra un mensaje de aclaración. La ampliación de palabras alternativas se aplica sólo a un único término y a las frases posibles. Por ejemplo, la

consulta [buscar empresa] podría tener la ampliación implícita de palabras alternativas para el término empresa, el término buscar y "frase buscar empresa", el que se corresponda con las palabras alternativas más largas que estén registradas.

Los resultados pueden incluir la versión HTML almacenada en caché del documento y enlaces a páginas que se enlazan hacia/desde ese resultado. Los enlaces situados sobre el campo de texto son grupos de orígenes. Los grupos de orígenes son grupos de documentos creados por el administrador de búsqueda, que se pueden buscar de forma conjunta. Al hacer clic en un grupo de orígenes, se restringe la búsqueda a ese grupo. (Oracle, Oracle SecureEnterprseSearch, Búsqueda Básica, 2006,2008)

3.6.9 Oracle Secure Enterprise Search APIs

Oracle Secure Enterprise Search proporciona varias API. Por ejemplo, con la API de servicios Web, puede integrar las capacidades de búsqueda de Oracle SES en su aplicación de búsqueda. Usando la administración del API, puede manejar varias instalaciones de Oracle SES más fácilmente que utilizando una interfaz gráfica.

3.6.9.1 API de servicios web

El API de servicios Web se utiliza para integrar las capacidades de búsqueda de Oracle SES en su aplicación de búsqueda. Oracle SES proporciona bibliotecas Java del servidor proxy. Puede utilizar las bibliotecas de Java o crear proxies, basado en el Lenguaje de Descripción de Servicios Web publicados (WSDL), para acceder a los servicios web de Oracle SES. Oracle SES proporciona dos API de servicios web:

- Consultas en el API de servicios Web: Permite realizar consultas de búsqueda, por ejemplo buscar los “beneficios de oracle" y devolverá

todos los documentos. También puede personalizar la puntuación por defecto de Oracle SES para crear una lista de resultados de búsqueda más relevantes para la empresa o configurar la agrupación de aplicaciones personalizadas.

- Administración del API de servicios Web: Permite realizar diversas tareas administrativas, por ejemplo, iniciar o detener una programación de rastreo, comprobar el estado del horario, obtener el nivel de índice de fragmentación estimado y llevar a cabo la optimización de índices.

El API de servicios web proporciona los siguientes beneficios:

- Las aplicaciones pueden ser desplegadas en cualquier ordenador que se conecta al servidor de Oracle SES a través de un protocolo de Internet estándar.
- El protocolo de servicios web está basado en XML, que permite la integración fácil de la aplicación.

Oracle SES también proporciona los proxies de cliente Java para la clasificación y analizar mensajes de servicios Web SOAP (Protocolo de Acceso a Objetos Simples). Las aplicaciones del cliente pueden utilizar la biblioteca en lugar de crear peticiones SOAP y analizar las respuestas SOAP por sí mismos para acceder a los servicios web de Oracle SES.

3.6.9.2 Conceptos de servicios web

Oracle SES Servicios Web consiste en una llamada a un procedimiento remoto (RPC) de la interfaz de Oracle SES que permite a la aplicación cliente invocar operaciones sobre Oracle SES en la red.

La aplicación cliente utiliza la especificación WSDL[5] publicado por Oracle SES URL de servicios Web para enviar un mensaje de solicitud de uso de Protocolo de Acceso a Objetos Simples (SOAP). El servidor responde a la aplicación cliente con un mensaje de respuesta SOAP.

3.6.9.2.1 Servicios Web

Un servicio Web es una aplicación de software identificado por un URI[6] (Identificadores Uniformes de Recursos), cuyas interfaces y encuadernación son capaces de ser definido, descrito y descubierto por los artefactos XML.

Un servicio Web es compatible con las interacciones directas con otras aplicaciones de software utilizando mensajes basados en XML y productos basados en Internet.

Un servicio web hace lo siguiente:

- Expone y describe a sí mismo: un servicio web define su funcionalidad y atributos para que otras aplicaciones puedan entenderlo. Al proporcionar un archivo WSDL de un servicio Web hace su funcionalidad disponible para otras aplicaciones.
- Permite a otros servicios localizarlo en la red: un servicio web puede ser registrado en un registro UDDI[7] para que las aplicaciones puedan localizarlo.
- Se puede invocar: Después de que un Servicio Web ha sido localizado y examinado, la aplicación remota puede invocar el servicio utilizando un protocolo estándar de Internet.

---

[5] WSDL: Un propósito general del lenguaje XML para describir los detalles de la interfaz, enlaces de protocolos y la implementación de los servicios Web.
[6] URI: Es una cadena de caracteres compactos para la identificación de un recurso abstracto o físico.
[7] UDDI: (Descripción Universal, Descubrimiento e Integración) define una manera de publicar y descubrir información acerca de los servicios Web.

- Los Servicios Web son de cualquier solicitud y de respuesta o el estilo de una vía, y se puede utilizar cualquier comunicación sincrónica o asincrónica. Sin embargo, la unidad fundamental de intercambio entre clientes de servicios Web y servicios Web, de cualquier estilo o tipo de comunicación, es un mensaje.

3.2.9.2.1 Protocolo de Acceso a Objetos Simples (SOAP)

El Protocolo de Acceso a Objetos Simples (SOAP) es un protocolo ligero basado en XML para el intercambio de información en un entorno distribuido descentralizado. SOAP soporta diferentes estilos de intercambio de información, incluido el intercambio orientado a RPC[8] y orientado a mensajes. Estilo RPC permite el intercambio de información para el procesamiento de petición-respuesta, donde un punto final recibe un mensaje orientado al procedimiento y responde con un mensaje de respuesta correlacionada. Orientado a mensajes de intercambio de información apoya a las organizaciones y aplicaciones que deben intercambiar mensajes u otros tipos de documentos en los que se envía un mensaje, pero el emisor no puede esperar o esperar una respuesta inmediata. Orientado a mensajes de intercambio de información también se conoce como el estilo de intercambio de documentos.

SOAP tiene las siguientes características:

- Protocolo de independencia.
- Lenguaje independiente.
- Plataforma e independencia del sistema operativo.
- Apoyo a la incorporación de mensajes XML SOAP de archivos adjuntos (utilizando la estructura MIME multipart).

[8] RCP: es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

3.2.9.2.2 Lenguaje de Descripción de los Servicios Web (WSDL)

El lenguaje de descripción de servicios Web (WSDL) es un formato XML para describir servicios de red que contengan información RPC y orientado a mensajes.

Los programadores o herramientas automatizadas de desarrollo pueden crear archivos WSDL para describir un servicio y puede hacer que la descripción disponible este en Internet.

Del lado del cliente programadores y herramientas de desarrollo pueden utilizar las especificaciones publicadas WSDL para obtener información acerca de los servicios Web disponibles para construir y crear proxies o plantillas de programas que acceden a los servicios disponibles.

3.2.9.3 Arquitectura de los Servicios Web

Oracle SES Servicios Web funciona con Oracle WebLogic Server. La implementación, configuración y despliegue de servicios Web de Oracle SES sigue los procedimientos y las normas proporcionadas por Oracle WebLogic Server.

Oracle SES WSDL define las operaciones y mensajes para los servicios Web de Oracle SES. El intercambio de mensajes de Oracle SES Servicios Web es estilo RPC, en el que el contenido del cuerpo del mensaje SOAP se ajustan a una estructura que especifica un procedimiento, e incluye un conjunto de parámetros o una respuesta con un número y cualquier parámetro adicional.

Oracle SES mensajes SOAP utiliza el enlace HTTP que está incrustado en un mensaje SOAP en el cuerpo de una solicitud HTTP y un mensaje SOAP se devuelve en la respuesta HTTP.

La siguiente Figura 3.4 muestra la arquitectura de Oracle SES Servicios Web:

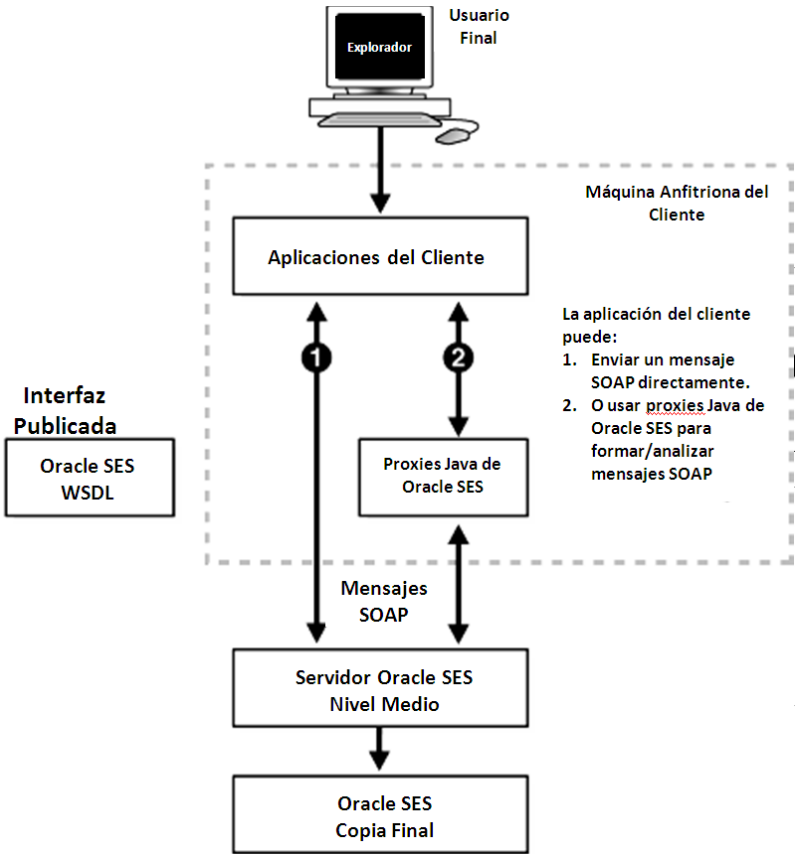

Figura 3. 4 Arquitectura de los Servicios Web[9]

3.7 Apache Lucene

Apache Lucene(TM) tiene un alto rendimiento, posee todas las funciones de un motor de búsqueda de texto escrito completamente en Java y tiene versiones en otros lenguajes como Perl, Python o .NET. Se trata de una tecnología adecuada para casi cualquier aplicación que requiera la búsqueda de texto, especialmente multiplataforma. Lucene ofrece dos servicios principales: la indexación de texto y búsqueda de texto. Estas dos operaciones son relativamente independientes entre sí. (Foundation, 2011-2012)

[9]Vishwanath Sreeraman, D. C., Cathy Shea, K.R., Michele Cyran, (2013). Oracle® Secure Enterprise SearchAdministrator's Guide, 11g Release 1 (11.1.2.0.0). Estados Unidos: U.S. Government

3.7.1 Historia

Lucene fue originalmente escrito por Doug Cutting[10], inicialmente disponible para su descarga desde el sitio web de Source Forge. Se unió a la familia de productos Jakarta de Apache Software Foundation, en septiembre de 2001 y se convirtió en su propio proyecto Apache de nivel superior en febrero de 2005. Ahora cuenta con un número de sub-proyectos, que se puede ver en http://lucene.apache.org.

Con cada lanzamiento, el proyecto ha contado con una mayor visibilidad, atrayendo a más usuarios y desarrolladores. En marzo de 2009, Lucene versión 2.4.1 había sido liberada. La siguiente Tabla 3.3 muestra la historia de la liberación de Lucene.

Tabla 3. 3

Historial de versiones de Lucene

Versión	Fecha de lanzamiento	Hitos
0.01	Marzo 2000	Liberación del primer código abierto (SourceForge)
1.0	Octubre 2000	
1.01b	Julio 2001	Última liberación de SourceForge
1.2	Junio 2002	Primera liberación de Apache Jakarta
1.3	Diciembre 2003	Formato de índice compuesto, mejoras de QueryParser, búsqueda remota, puntuación extensible del API, posicionamiento simbólico
1.4	Julio 2004	Clasificación, consultas, vectores
1.4.1	Agosto 2004	Corrección de errores para clasificar el rendimiento
1.4.2	Octubre 2004	Optimización de IndexSearcher
1.4.3	Noviembre 2004	
1.9.0	Febrero 2006	Campos binarios almacenados, DateTools, NumberTools, RangeFilter, RegexQuery, Requiere Java 1.4
1.9.1	Marzo 2006	Corrección de errores en BufferedIndexOutput
2.0	Mayo 2006	Eliminación de métodos obsoletos
2.1	Febrero 2007	Eliminar/Actualizar documento en

CONTINÚA

[10] Lucene es el segundo nombre de la esposa de Doug, es también el primer nombre de su abuela materna.

		IndexWriter, Bloqueo de simplificaciones, mejoras de QueryParser, contrib/benchmark
2.2	Junio 2007	Mejoras de rendimiento, Consultas de funciones, cargas útiles, Campos Pre-analizados, políticas de eliminación personalizada
2.3.0	Enero 2008	Mejoras de rendimiento, fusionar y combinar las políticas personalizada de los horarios, el fondo se fusiona de forma predeterminada, herramienta para detectar corrupción de los índices, IndexReader.reopen
2.3.1	Febrero 2008	Corrección de errores de la 2.3.0
2.3.2	Mayo 2008	Corrección de errores de la 2.3.1
2.4.0	Octubre 2008	Mejoras de rendimiento adicionales, semántica transaccional (rollback, commit), método expungeDeletes, eliminar por consulta en IndexWriter
2.4.1	Marzo 2009	Corrección de errores de la 2.4.0

Doug Cutting sigue siendo una fuerza importante detrás de Lucene, y muchos más desarrolladores se han unido al proyecto con el tiempo. El equipo central de Lucene incluye media docena de desarrolladores activos, tres de los cuales son autores del libro Manning Lucene in Action 2nd Edition. Además de los desarrolladores oficiales de proyectos, Lucene tiene una comunidad de usuarios técnicos bastante grande y activos que con frecuencia contribuye parches, correcciones de errores y nuevas características. (Erik Hatcher, 2004)

3.7.2 Puertos de Lucene: Perl, Python, C + +, NET, Ruby, PHP

Una manera de juzgar el éxito del software libre es por el número de veces que ha sido portado a otros lenguajes de programación. Usando este indicador, Lucene es un gran éxito. Aunque Lucene está escrito en Java, hay puertos de Lucene y enlaces en muchos otros entornos de programación, incluyendo Perl, Python, Ruby, C / C + +, PHP y C # (. NET).Con el fin de entender exactamente cómo Lucene encaja en una aplicación de búsqueda,

incluyendo lo que Lucene puede y no puede hacer, a continuación se presenta la arquitectura de una aplicación moderna “típica” de búsqueda.

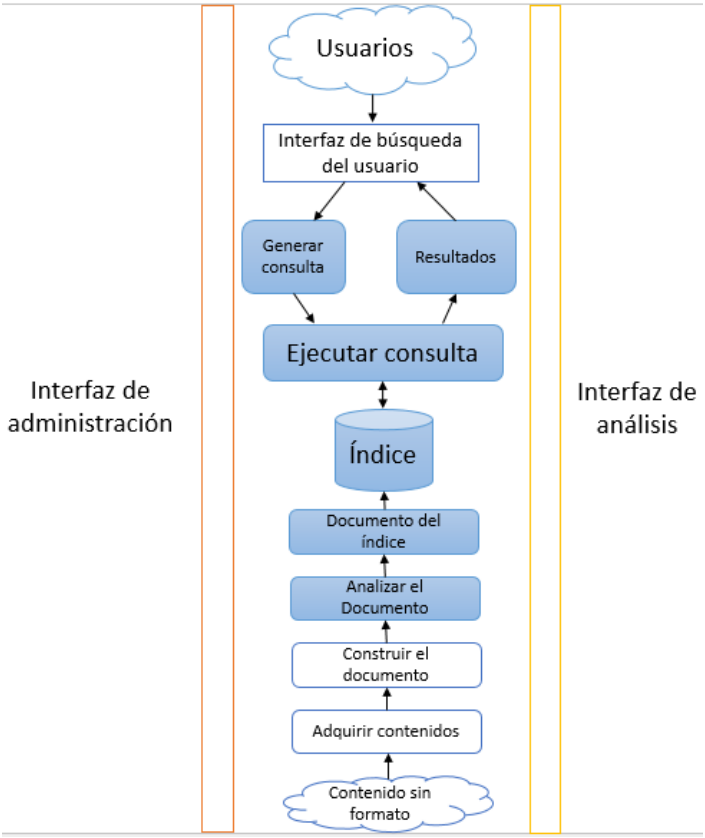

Figura 3. 5 Componentes típicos de la aplicación de búsqueda; los componentes sombreados muestran que partes ocupa Lucene

3.7.2.1 Componentes de la indexación

Un índice es como una estructura de datos que permite un rápido acceso aleatorio a palabras almacenadas en su interior. El concepto detrás de esto es análogo a un índice al final de un libro, que le permite localizar rápidamente las páginas que tratan sobre ciertos temas. En el caso de Lucene, un índice es una estructura de datos especialmente diseñado, típicamente almacenados en el sistema de archivos como un conjunto de archivos de índice.

3.7.2.1.1 Adquirir contenido (Acquire Content)

El primer paso, en la parte inferior de la figura 3.5, es adquirir contenido. Este proceso, que se refiere a menudo como un crawler (buscador) o araña, recoge el contenido que necesita ser indexado. Eso puede ser trivial, por ejemplo, si desea indexar un conjunto de archivos XML que residen en un directorio específico del sistema de archivos o si todo su contenido reside en una base de datos bien organizada. Puede ser terriblemente complejo y desordenado, si el contenido se esparce en todo tipo de lugares (sistemas de archivos, sistemas de gestión de contenidos, Microsoft Exchange, Lotus Domino, varios sitios web, bases de datos, archivos XML locales, etc).

Autorización, que significa sólo se permiten ciertos usuarios autenticados para ver ciertos documentos, pueden complicar la adquisición de contenidos, ya que puede requerir el acceso "súper usuario" en la adquisición de los contenidos.

Lucene, al ser una biblioteca de búsqueda de núcleo, no proporciona ninguna funcionalidad para apoyar la adquisición de contenidos. Esto es totalmente fuera de su aplicación, o una pieza separada de software. Hay una serie de rastreadores de código abierto, por ejemplo:

- Nutch tiene un buscador de alta escala que es la adecuada para descubrir contenidos rastreando sitios Web.
- Grub (http://www.grub.org), un popular rastreador web de código abierto.
- Heritrix, rastreador de código abierto de Archivo de Internet.
- Proyecto Apache Droids.
- Aperture (http://aperture.sourceforge.net) tiene capacidad para el rastreo de sitios web, sistemas de archivos y los buzones de correo y la extracción y el texto de indexación.

3.7.2.2 Construir Documento (Build Document)

Una vez que se obtenga el contenido en bruto que necesita ser indexado, debe traducir el contenido a las "unidades" (generalmente llamados "documentos") utilizadas por el motor de búsqueda. El documento consiste típicamente en varios campos denominados por separado con los valores, por ejemplo, el título, cuerpo, resumen, autor, url, etc.

3.7.2.3 Analizar Documento (Analyze Document)

No hay índices de los motores de búsqueda de texto directamente: más bien, el texto debe ser dividido en una serie de elementos atómicos individuales llamados tokens. Esto es lo que sucede durante el paso de "Analyze document". Cada ficha corresponde aproximadamente a una "palabra" en el idioma, y este paso determina cómo los campos de texto en el documento se dividen en una serie de tokens.

3.7.2.4 Índice de Documentos (Index Document)

Durante el paso de la indexación, se añade el documento al índice. Lucene, ofrece todo lo necesario para este paso, y trabaja con un poco de magia bajo un sorprendentemente simple API. Es importante recordar que la indexación es algo así como un "mal necesario" que se debe realizar con el fin de proporcionar una buena experiencia de búsqueda: se debe diseñar y personalizar el proceso de indexación en la medida en que mejora la experiencia de búsqueda de los usuarios.

3.7.3 Requisitos del sistema

Apache Lucene corre en Java 6 o superior. Con todas las versiones de Java, se recomienda no utilizar opciones experimentales -XX JVM. También

se recomienda siempre usar la última versión de su Java VM, porque los errores pueden afectar a Lucene.

3.7.4 Características

Lucene posee las siguientes características de gran alcance a través del API:

- Potente, preciso y eficiente algoritmo de búsqueda.
- Búsqueda por Ranking
  Retorna los mejores primeros resultados.
- Muchos tipos de consultas poderosas entre las cuales se encuentran: consultas de frases, consultas de proximidad, rangos y más.
- Búsqueda por campos
  La búsqueda por campos se puede realizar mediante título, autor, contenidos, entre otros.
- Búsqueda por rangos de fecha.
- Se puede ordenar por cualquier campo.
- Múltiples índices de búsqueda con resultados combinados.
- Permite la actualización y búsqueda simultánea.
- Puede indexar y realizar búsquedas sobre cualquier dato que pueda ser convertido a texto desde páginas web hasta documentos Microsoft Word pasando por archivos PDF.

3.7.5 Búsqueda con Lucene

Lucene puede analizar, indexar y buscar cualquier documento que se puede convertir en texto, por lo que no es un limitante para las páginas web. Las etapas básicas en la búsqueda convencional son:

- Crawling (Buscador)
- Parsing

- Análisis
- Indexación
- Búsqueda

Crawling o Buscador se refiere al proceso de recopilación de los documentos en los que se quiere habilitar la funcionalidad de búsqueda. Puede no ser necesario si los documentos existen o ya se han recogido.

Parsing es necesario para la transformación de los documentos (XML, HTML, Word, PDF) en una estructura común que representará a los campos de indexación en forma puramente textual.

3.7.6 Integración de Lucene

Lucene permite añadir capacidades de indexación y búsqueda en sus aplicaciones. También puede indexar y consultar los datos que se pueden convertir a un formato de texto. Como se puede ver en la siguiente figura:

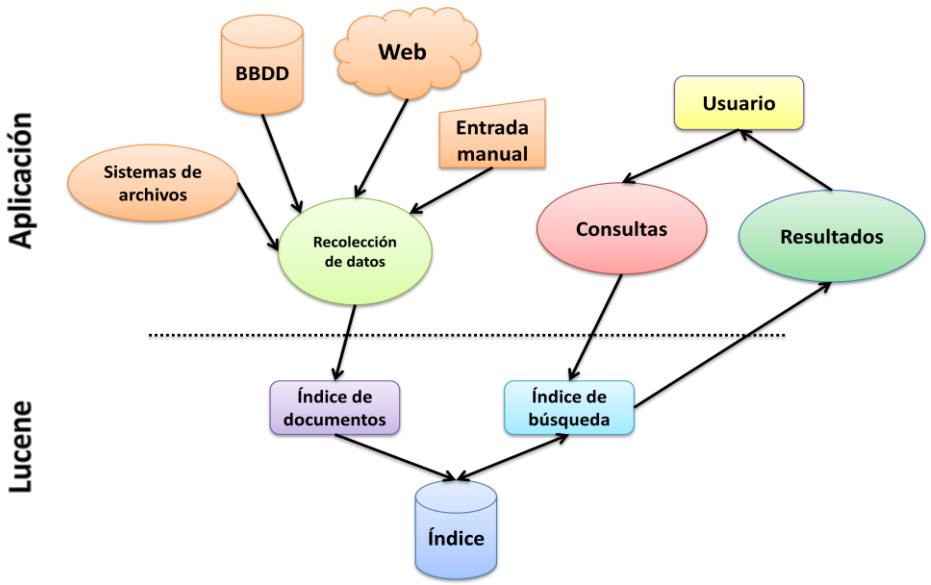

Figura 3. 6 Integración con Lucene

3.7.6.1 Clases para indexación

Las clases básicas importantes de Lucene para el proceso de indexación son:

- **IndexWriter**

  Es el componente central del proceso de indexado. Esta clase permite crear un nuevo índice o usar uno ya creado y añadirle documentos. Da permisos de escritura en el índice pero no de lectura o búsqueda.

  Para crear un índice, lo primero que se debe hacer es crear un objeto IndexWriter. El objeto IndexWriter se utiliza para crear el índice y para agregar nuevas entradas de índice. Se puede crear un IndexWriter de la siguiente manera:

```
IndexWriter indexWriter = new IndexWriter("index-directory", new
StandardAnalyzer(), true);
```

  El primer parámetro es el directorio en el que se creará el índice de Lucene, en este caso es el índice de directorio.

  El segundo parámetro especifica el "document parser" o "document analyzer", que se utilizará cuando Lucene indexa sus datos.

  El tercer parámetro indica a Lucene crear un nuevo índice, si el índice no se ha creado en el directorio todavía.

- **Directory**

  La clase Directory representa la ubicación de un índice en Lucene. Esta a su vez utiliza subclases FSDirectory para guardar los índices en el sistema de archivos. Esta es la clase que más se usa para el almacenamiento de índices. La clase IndexWriter hace uso de FSDirectory cuando necesita recibir como parámetro el directorio donde se almacenarán los índices. Otra subclase llamada RAMDirectory, a diferencia de FSDirectory, esta se usa para

almacenar los índices en memoria, es recomendable cuando se crean índices pequeños o si se realizan pruebas de indización o búsqueda.[11]

- **Analyzer**

  El analizador especificado en el constructor del *IndexWriter es* el encargado de extraer los tokens del texto que van a ser indexados y eliminar el resto. Hay muchas implementaciones de esta clase que realizan distintos filtros.

  El trabajo del Analyzer es "analizar" cada campo de datos en "tokens" indexables o palabras clave. La siguiente Tabla 3.4 muestra algunos de los más interesantes:

Tabla 3. 4

Analizadores de Lucene

Analyzer	Descripción
StandardAnalyzer	Un sofisticado analizador de propósito general.
WhitespaceAnalyzer	Un analizador muy sencillo que simplemente separa tokens utilizando el espacio en blanco.
StopAnalyzer	Elimina las palabras comunes inglesas que no son generalmente útiles para la indexación.
SnowballAnalyzer	Un interesante analizador experimental que trabaja en las raíces de palabras (Por ejemplo una búsqueda sobre lluvia también debe devolver entradas con llover, llovió, y así sucesivamente).

Existe una serie de analizadores específicos del idioma, incluyendo analizadores para alemán, holandés ruso, francés, y otros.

Cuando se crea un “IndexWriter”, se debe especificar qué “Analyzer” se usará para el índice.

- **Document**

  Representa una colección de campos. De algún modo es un documento virtual que se quiere hacer recuperable. Los campos

[11] Tomado de Otis Gospodneticet.al., "Lucene in action", *Manning* , 2005, pp. 44, 45, 46,53, 54,55, 56, 57, 58, 59, 349, 352, 353, 355.

almacenan toda la información del documento, por ejemplo: autor, título o tema son indexados y almacenados como campos separados de un documento.[12]

- **Field**

  Corresponden a porciones de información que van a ser requeridas por el índice durante la búsqueda. Consisten en pares de nombre y valor y pueden ser de 4 tipos:

  - **Keyword**

    No es analizado pero es indexado y almacenado. Se utiliza para valores que deben ser preservados como URLs, rutas de archivos o fechas.
  - **UnIndexed**

    No es analizado ni indexado pero si almacenado y se utiliza para valores que se quieren mostrar cuando se realiza una búsqueda pero que nunca son buscados directamente.
  - **UnStored**

    El contrario de "UnIndexed". Este campo es analizado e indexado pero no almacenado en el índice. Se utiliza para grandes cantidades de texto que no deben ser recuperadas en su forma original como los cuerpos de páginas web.
  - **Text**

    Es analizado e indexado y es contra lo que se busca.

Todo campo tiene un nombre y valor, que son pasados como argumentos al definir el tipo de campo a crear. Estos son algunos ejemplos de cómo utilizar la clase **Field** y, según su tipo si cada campo es analizado, indexado o guardado en el momento de crear el índice:

[12] Tomado de Otis Gospodneticet.al., "Lucene in action", Manning , 2005, pp. 44, 45, 46,53, 54,55, 56, 57, 58, 59, 349, 352, 353, 355.

Tabla 3. 5

Tipos de Campos en Lucene[13]

Método Field	Analizado	Indexado	Guardado	Ejemplos
Field.Keyword(String, String) Field.Keyword(String, Date)		X	X	Teléfonos y números de seguridad social, URLs, nombres de personas, fechas.
Field.UnIndexed(String, String)			X	Tipo de documento(PDF,HTML,etc.), si no son usados como un criterio de búsqueda.
Field.UnStored(String, String)	X	X		Títulos de documentos y contenidos.
Field.Text(String, String)	X	X	X	Títulos de documentos y contenidos.
Field.Text(String, Reader)	X	X		Títulos de documentos y contenidos.

3.7.6.2 Clases para la búsqueda

Para la búsqueda se debe conocer las siguientes clases básicas:

- **IndexSearcher**

  "IndexSearcher" es en la búsqueda lo que "IndexWriter" es en la indexación. Es la clase principal que abre el índice para buscar en él y ofrece varios métodos de búsqueda, lo que hace esta clase es pasar como parámetro el "query" o consulta y regresar un objeto hits.

[13]Tomado de OTIS GOSPODNETIC Y ERIK HATCHER. "Lucene in Action". Manning Publications (2004). GOS o 05:1 1.Ex. [Citado en pág. 21,22.]

- **Term**

  Un término es la unidad básica para la búsqueda. Similar al objeto Field, consiste de un par de elementos: el nombre del campo y su valor.

- **Query**

  Lucene tiene diferentes subclases de “Query”, la más utilizada es “TermQuery” por los métodos que ella contiene.

- **QueryParser**

  La clase “QueryParser” es utilizada para construir un analizador que puede buscar a través de un índice. Un “Query” es una serie de cláusulas. Una cláusula puede ser un término que indica todos los documentos que contienen un término en particular o de una consulta anidada entre paréntesis. Una consulta anidada se puede utilizar con un (+) o (-) de prefijo para requerir cualquiera de un conjunto de términos.

- **TermQuey**

  Es el tipo de Query más básico soportada por Lucene, se utiliza para hacer coincidir documentos que tienen valores específicos.

- **Hits**

  La clase Hits almacena los puntos de referencia a los resultados de la búsqueda, es decir todos los documentos encontrados que se relacionan con el “Query”.[14]

3.7.7 Indexación

El núcleo principal de todos los motores de búsqueda es el concepto de indexación. La indexación en definitiva se puede definir como el tratamiento de los datos originales en una muy eficiente búsqueda de referencia cruzada con el fin de facilitar la búsqueda rápida. Un ejemplo de indexación sería si una persona quiere buscar una palabra o una frase entre un gran número de

[14]Tomado de Otis Gospodneticet.al., "Lucene in action", *Manning*, 2005, pp. 44, 45, 46,53, 54,55, 56, 57, 58, 59, 349, 352, 353, 355.

archivos, el método más simple sería la de analizar de forma secuencial cada archivo de la palabra o frase determinada. La principal desventaja de este enfoque es que no funciona bien con conjuntos de archivos más grandes o los casos en que los archivos son de gran tamaño, por lo tanto, para buscar grandes cantidades de texto rápidamente, primero es necesario el índice del texto y convertirlo en un formato que le permitirá a la persona buscar con rapidez, lo que elimina el proceso de exploración secuencial lento. Este proceso de conversión se llama indexación y su salida se llama un índice.

3.7.7.1 Estructura de un Índice

Un índice de Lucene se almacena en un único directorio del sistema de archivos en un disco duro. Los elementos básicos de un índice de Lucene son segmentos, documentos, campos y términos. Un índice de Lucene se compone de uno o más segmentos. Cada segmento contiene uno o más documentos. Cada documento tiene uno o más campos, y cada campo contiene una o más términos. Cada término es un par de cadenas que representan un nombre de campo y un valor. Un segmento consta de una serie de archivos como se puede ver en la Figura.

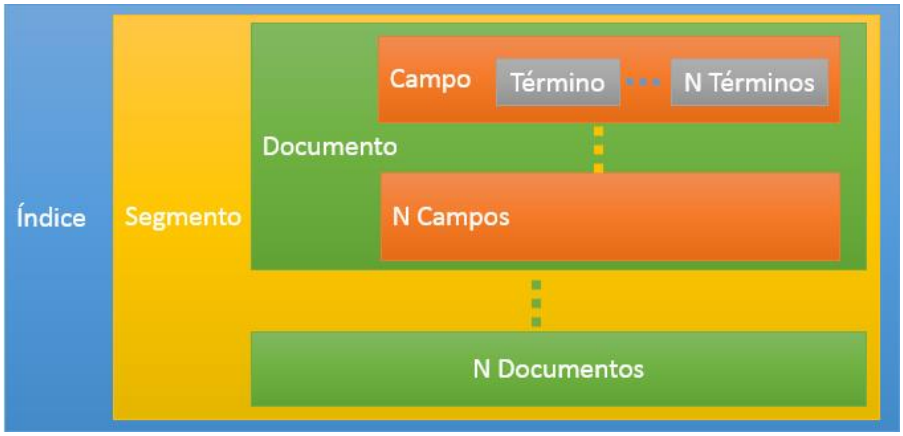

Figura 3. 7 Estructura de un Índice

3.7.7.2 Factores que afectan la velocidad de indexación

El cuello de botella de una aplicación típica de indexación de texto es el proceso de escritura de los archivos del índice en un disco. Cuando se añaden nuevos documentos a un índice de Lucene, se almacenan inicialmente en la memoria en lugar de escribir de inmediato al disco. La forma más sencilla de mejorar el rendimiento de indexación de Lucene es ajustar el valor de la variable de instancia Factor del “IndexWriter”. Este valor le dice a Lucene cuántos documentos almacena en la memoria antes de escribirlos en el disco, así como la frecuencia que combina varios segmentos. El valor por defecto es 10; Lucene almacenará 10 documentos en la memoria antes de escribirlos en un único segmento en el disco.

El valor del Factor de combinación de 10 también significa que una vez que el número de segmentos en el disco ha alcanzado la potencia de 10, Lucene se fusionará estos segmentos en un solo segmento. Cuando el valor del Factor de combinación se establece a 10, un nuevo segmento se crea en el disco por cada 10 documentos agregados al índice. Cuando se añade el décimo segmento de tamaño 10, todos los 10 se fusionarán en un solo segmento de tamaño 100.Cuando se han añadido 10 segmentos de tamaño 100, éstas serán combinadas en un solo segmento que contiene 1.000 documentos, y así sucesivamente. Por lo tanto, en cualquier momento, habrá no más de 9 segmentos en cada potencia de 10 del tamaño del índice.

3.7.8 Opciones de campo

El campo es tal vez la clase más importante al indexar documentos: es la clase real que tiene cada valor para ser indexados. Cuando se crea un campo, hay numerosas opciones especificadas para controlar exactamente lo que Lucene debe hacer con ese campo una vez añadido el documento al índice. Las opciones se dividen en varias categorías independientes, que se

cubren en cada subsección siguiente: indexación, almacenamiento y vectores de términos.

A continuación se enumeran las combinaciones comunes de opciones de campo.

3.7.8.1 Opciones de campo para la indexación

Las opciones para el control de indexación (Field.Index. *) como el campo de texto fue buscado a través del índice invertido. Estas son las opciones:

- **Index.ANALYZED:** Se utiliza el analizador para romper el valor del campo en una secuencia de símbolos separados y hacer que cada una de las secuencias se pueden buscar.
- **Index.NOT_ANALYZED:** Indexar el campo, pero no analizar la cadena. En su lugar, tratar el valor entero del campo como un solo símbolo y hacer ese símbolo de búsqueda. Esto es útil para los campos que desea buscar, pero no debe ser dividido, como URLs, rutas del sistema de archivos, fechas, nombres de personas, números de Seguro Social, números de teléfono, etc. Esto es especialmente útil para permitir "coincidencia exacta" de la búsqueda. Se tiene indexado la ruta del sistema de archivos en el índice usando esta opción.
- **Index.ANALYZED_NO_NORMS:** una variante avanzada de Index.ANALYZED no almacena información de normas en el índice. Las normas son un registro de información de aumento en el índice, pero puede ser consumida la memoria cuando se busca.
- **Index.NOT_ANALYZED_NO_NORMS:** al igual, pero también no almacena Normas.
- **Index.NO:** no se tiene el valor de este campo disponible para la búsqueda en todos.

3.8 Apache Tika

Apache Tika es una librería que detecta y extrae los metadatos y el contenido de texto estructurado de documentos en diferentes formatos (Microsoft Word, Excel, Power Point, PDF, txt, etc) mediante bibliotecas analizadoras existentes. Tika es un proyecto de la Fundación de Software Apache, y anteriormente fue un sub-proyecto de Apache Lucene. (Chris A. Mattmann, 2012)

3.8.1 Historia

La figura 3.8 muestra una línea de tiempo del desarrollo de Tika, desde la propuesta hasta el máximo nivel del proyecto. La idea de Tika fue originalmente propuesta en el proyecto de Apache Nutch. Nutch se describe mejor como un marco de código abierto para la búsqueda en la Web a gran escala.

El proyecto comenzó como una idea original de Doug Cutting (el padre de los proyectos de Lucene y Hadoop, un asistente general de búsqueda de código abierto), que estaba frustrado con las empresas de búsqueda comerciales y la naturaleza propietaria de sus algoritmos de clasificación y sus características. En el sitio habitual de Nutch Sourceforge.net declaró:

"Nutch provee una alternativa transparente a los motores de búsqueda comerciales. Sólo los resultados de búsqueda de código abierto pueden ser completamente confiables. Todos los principales motores de búsqueda tienen fórmulas propietarias de ranking, resultados y no revelan por qué una página determinada ocupa los primeros lugares en la búsqueda. Además,

algunos buscadores deciden qué sitios a indexar basados en los pagos, en lugar de los méritos de los mismos sitios."[15]

Nutch creció rápidamente a partir de un esfuerzo inicial en un marco establecido, con el mundo académico que abarca la participación comunitaria (el departamento de Servicios Web centroamericanos de Oregon State); la industria, por ejemplo, en el Archivo de Internet (una empresa sin fines de lucro enfocada en archivar digitalmente la web); gobierno (con algunos de los esfuerzos de búsqueda de la ciencia planetaria y la investigación sobre el cáncer realizado por su servidor en la NASA); y docenas de otras entidades comerciales y esfuerzos.

Finalmente, Nutch alcanzó sus límites superiores en la escalabilidad, alrededor de 100 millones de páginas web, un factor de 40 menor que la de los motores de búsqueda comerciales como Google. Alrededor del mismo tiempo, el equipo de computación grid de Yahoo! entró en escena y comenzó a evaluar Nutch, pero la limitación de la escalabilidad es un problema que necesita ser resuelto.

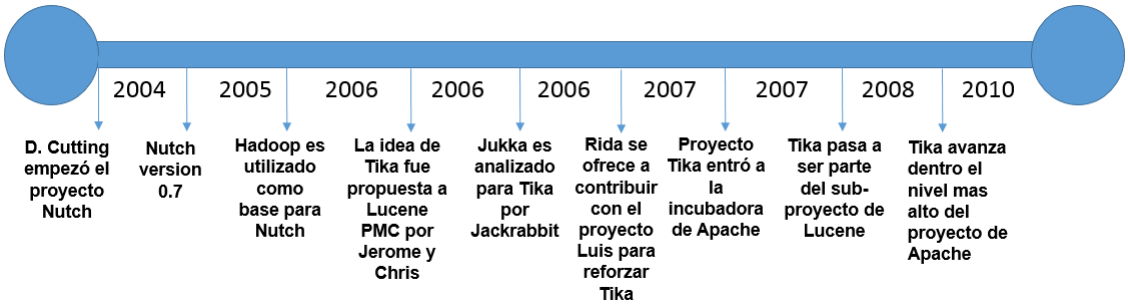

Figura 3. 8 Línea de tiempo visual de la historia de Tika.[16]

3.8.2 Formatos de documento admitidos

A continuación se muestra todos los formatos de documentos soportados por Apache Tika 0.6 y la forma en que se analiza por Tika.

[15] Tomado de Chris A. Mattmann y Jukka L. Zitting, ¨Tika in Action¨, Manning, 2012, pp 15.
[16] Tomado de Chris A. Mattmann y Jukka L. Zitting, ¨Tika in Action¨, Manning, 2012.

- **Lenguaje de Marcas de Hipertexto (HTML)**

  El Lenguaje de Marcas de Hipertexto (HTML) es la lengua franca de la web. Tika utiliza la biblioteca “TagSoup” para soportar prácticamente cualquier tipo de HTML que se encuentra en la web. La salida de la clase “HtmlParser” se garantiza que sea bien formado y válido para XHTML y diversas heurísticas que se utilizan para prevenir cosas como scripts de línea de desorden del contenido del texto extraído.

- **XML y formatos derivados**

  El formato de lenguaje de marcado extensible (XML) es un formato genérico que puede ser utilizado para todo tipo de contenido. Tika tiene analizadores personalizados para algunos vocabularios XML ampliamente usados como XHTML, OOXML y ODF, pero la clase DcXMLParser por defecto simplemente extrae el contenido del texto del documento y hace caso omiso de cualquier estructura XML. La única excepción a esta regla son los elementos de metadatos Dublin Core que se utilizan para los metadatos del documento.

- **Formatos de documentos de Microsoft Office**

  Microsoft Office y algunas aplicaciones relacionadas producen documentos en el formato genérico OLE 2 Documento Compuesto y Office Open XML (OOXML). El formato anterior OLE 2 se introdujo en Microsoft Office versión 97 y fue el formato por defecto hasta la versión de Office 2007 y el nuevo formato OOXML basado en XML. Las clases “OfficeParser” y “OOXMLParser” utilizan las bibliotecas Apache POI para que admita texto y los metadatos de extracción de ambos documentos OOXML y OLE2.

- **Formato OpenDocument**

  El formato OpenDocument (ODF) se utiliza sobre todo como el formato por defecto de la suite ofimática OpenOffice.org. La clase

OpenDocumentParser soporta este formato y el formato anterior OpenOffice 1.0 en el que se basa la FOD.

- **Formato de Documento Portátil**

  La clase PDFParser analiza los documentos del Formato de Documento Portátil (PDF) usando la librería Apache PDFBox.

- **Formato de publicación electrónica**

  La clase EpubParser soporta el formato de publicación electrónica (EPUB) utilizado por muchos libros digitales.

- **Formato de texto enriquecido**

  La clase RTFParser utiliza la característica javax.swing.text.rtf estándar para extraer el contenido de texto desde los documentos de Texto Enriquecido (RTF).

- **Formatos de compresión y empaquetado**

  Tika utiliza la biblioteca Commons Compress para soportar varios formatos de compresión y empaquetado. La clase PackageParser y sus subclases primero analizan la compresión de nivel superior o Formato de empaquetamiento y luego pasan los flujos de documentos descomprimidos a una segunda etapa de análisis mediante la instancia del analizador especificado en el contexto de análisis.

- **Formatos de texto**

  Extrayendo el contenido del texto a partir de archivos de texto plano parece una tarea sencilla hasta que empiece a pensar en todas las posibles codificaciones de caracteres. La clase TXTParser utiliza la codificación de código de detección del proyecto UCI para detectar automáticamente la codificación de caracteres de un documento de texto.

- **Formatos de fuentes y Sindicación**

  La clase FeedParser soporta los formatos de sindicación RSS y Atom feed.

- **Formatos de audio**

  Tika puede detectar varios formatos comunes de audio y extraer metadatos de ellos, incluso la extracción de texto es compatible con algunos archivos de audio que contienen la letra u otro contenido textual. Los metadatos extraídos incluyen velocidades de muestreo, canales, información de formato, artistas, títulos, etc. Las clases AudioParser y MidiParser utilizan características javax.sound estándar para procesar formatos de audio simples. La clase Mp3Parser añade soporte para el formato MP3 ampliamente utilizado, mientras que las clases VorbisParser y FlacParser proporcionan un apoyo similar para el Ogg Vorbis y FLAC, y MP4Parser ofrece para audio MP4.

- **Formatos de imagen**

  La clase ImageParser utiliza la característica javax.imageio estándar para extraer metadatos sencilla de formatos de imagen compatibles con la plataforma Java. Los metadatos de imagen más compleja se encuentra disponible a través de la clase y las clases JpegParser TiffParser que utiliza la biblioteca de metadatos-centrifugadora para compatible con Exif extracción de metadatos de imágenes JPEG y TIFF.

- **Formatos de vídeo**

  Actualmente Tika sólo soporta el formato de vídeo Flash mediante un algoritmo de análisis sencillo implementado en la clase FLVParser. La familia MP4 de formatos de vídeo (MP4, QuickTime, 3GPP, etc) es apoyado por la clase MP4Parser, que extrae los metadatos en el vídeo, junto con el flujo de audio (si está presente).

- **Archivos de clase de Java y archivos**

  La clase ClassParser extrae nombres de clase y firmas de los métodos de los archivos de clases de Java, y la clase ZipParser soporta también archivos jar.

- **El formato mbox**

  El MboxParser puede extraer mensajes de correo electrónico desde el formato mbox utilizado por muchos archivos de correo electrónico y buzones de correo al estilo Unix.

- **Formatos CAD**

  El DWGParser puede extraer metadatos simple desde el formato DWG[17] de CAD.

- **Los formatos de fuente**

  La clase TryeTypeParser puede extraer metadatos simples desde el formato de fuentes TrueType.

- **Programas y bibliotecas ejecutables**

  El ExecutableParser puede extraer información de los metadatos en las plataformas, arquitecturas y tipos de una gama de formatos ejecutables y bibliotecas, tales como archivos ejecutables de Windows y los programas de Linux / BSD y bibliotecas.

[17] **DWG** es un formato de archivo informático de dibujo computarizado, utilizado principalmente por el programa AutoCAD; producto de la compañía AutoDesk.

CAPÍTULO 4

4.1 Implementación de Oracle Secure Enterprise Search (SES) en el sistema Gestor Fiducia Fondos JEE

La integración entre Oracle SES y la aplicación de Gestor Fiducia Fondos JEE se realiza mediante una funcionalidad la cual contiene la página de búsqueda de Oracle SES para esto primero se identificó cuáles son los repositorios donde se encuentra la información que se necesita para la búsqueda de archivos. El repositorio de Gestor está diseñado de tal forma que toda la información necesaria para la búsqueda se almacena en un conjunto específico de tablas de la base de datos.

Los documentos que necesitamos indexar se encuentra almacenada en la tabla "DOC_DOCUMENTO_VERSION", esta información se guarda desde diferentes funcionalidades del sistema Gestor Fiducia Fondos JEE, después de haber identificado la tabla se escoge que tipo de origen se quiere crear para la indexación, en Oracle SES existen varias opciones para indexar repositorios ya sea base de datos, una tabla, páginas web, un archivo, E-mail, entre otras, la opción más adecuada para este caso es la de Tabla, otra alternativa es la de Base de Datos pero por el número de tablas que existen en el repositorio, al momento de la indexación tomaría más tiempo y se requeriría de mayor capacidad en hardware y software.

3.8.2.1 Proceso de solicitud, indexación y búsqueda

La siguiente figura 4.1 muestra el proceso de solicitud, indexación y búsqueda del Oracle SES integrado con la aplicación del sistema Gestor.

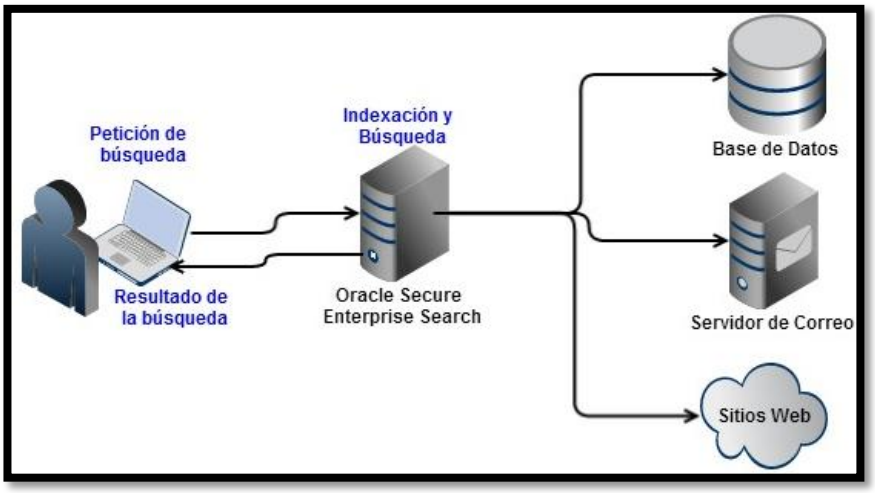

Figura 4. 1 Proceso de solicitud, indexación y búsqueda

3.8.2.2 Creación del Origen

Una vez definido lo que se necesita para la configuración del origen, se escoge el tipo de origen que es “Tabla” y se lo crea. En la siguiente pantalla, en la sección General, en “SourceName” se define un nombre que lo identifique, después en la sección de Información de Base de Datos se ingresa la información del repositorio Gestor como se puede observar en la Figura 4.2, en el campo “Database Host Name” se ingresa la dirección IP donde se encuentra la Base de datos, en “TCP Port Number” se ingresará el puerto que utiliza la Base de Datos, en “UserName” y “Password” el nombre y la contraseña que se conecta a la Base de Datos.

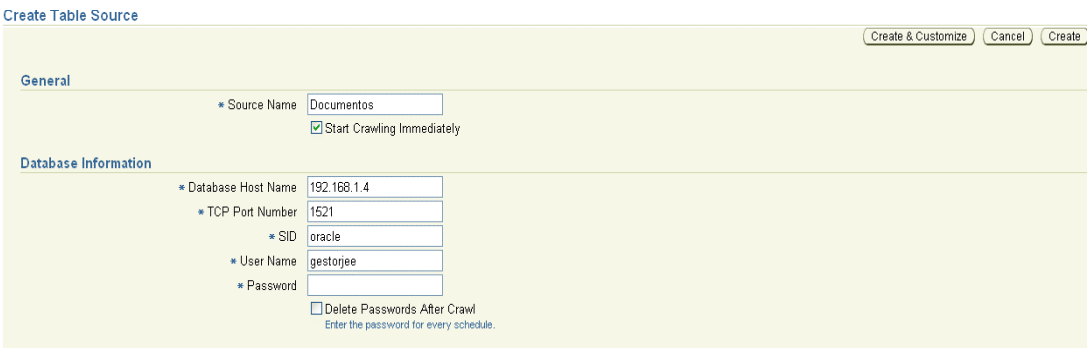

Figura 4. 2 Ingreso de datos en la sección General e Información de Base de Datos

En la sección de “Información de la tabla (TableInformation)” se ingresa el esquema y el nombre de la tabla donde se encuentra los archivos y después se selecciona la opción “LocateTable” para localizar la tabla desde la base de datos, una vez localizada y cargados los campos se selecciona la clave primaria de la tabla que es “NUM_LICENCIA, COD_DOCUMENTO, VERSION”, después en la siguiente opción se escoge la columna que contiene los archivos que se llama “ARCHIVO” y el tipo de contenido que es Binario como se puede ver en la Figura 4.3:

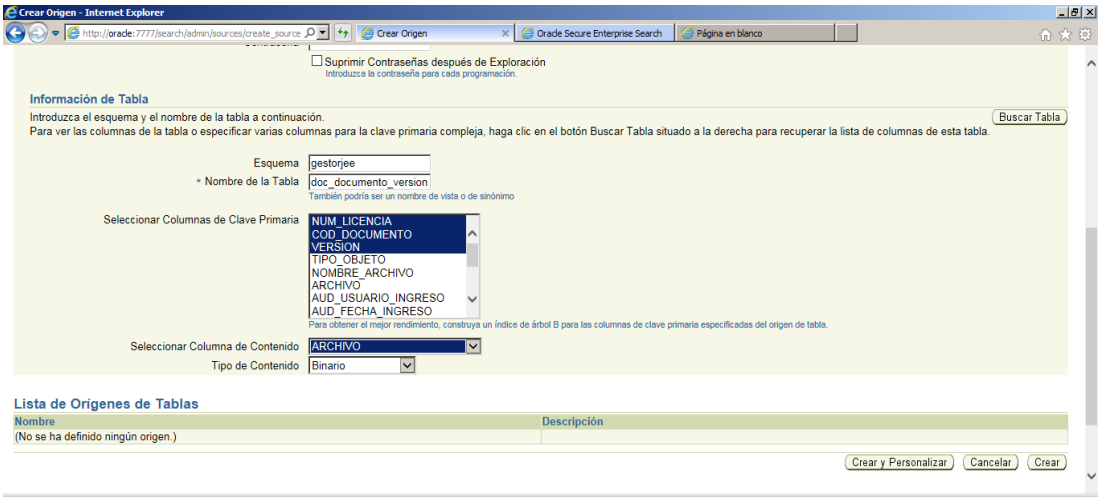

Figura 4. 3 Ingreso de la tabla donde se encuentra los documentos

Una vez seleccionada la tabla se vuelve a ingresar la contraseña para la conexión con la base de datos. Si se desea agregar más columnas al origen se escoge la opción “Crear y Personalizar” para que sean mapeadas como lo demuestra la Figura 4.4, en este caso no es necesario agregar más columnas ya que solo se cuenta con una.

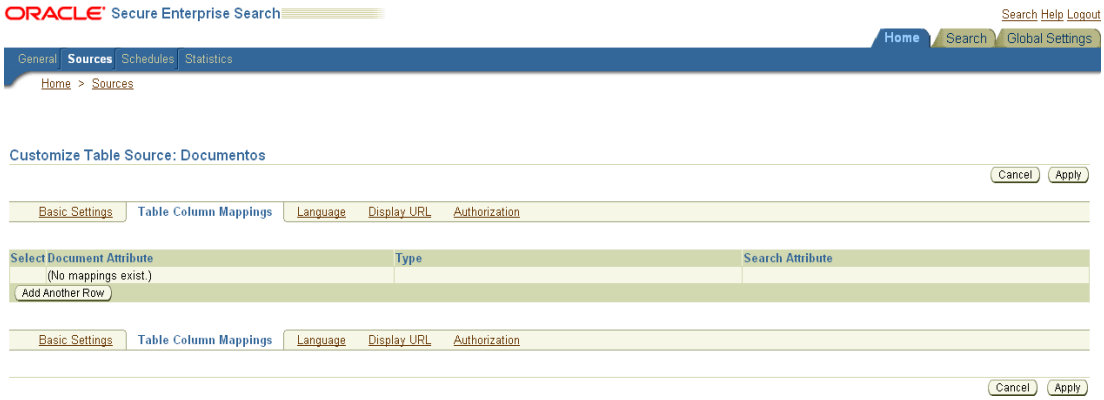

Figura 4. 4 Asignaciones de columnas de tablas

En la pestaña “Lenguaje” que se muestra en la Figura 4.5 se puede definir cuál es el lenguaje del contenido que va a ser indexado, por defecto aparecerá en inglés y se la cambia a español, la otra opción es el lenguaje de la columna que puede contener una tabla, esta opción especifica el código de idioma (ISO-639-1). El idioma especificado en esa columna anulará el idioma predeterminado, esta opción no se la toma en cuenta ya que en la tabla no se tiene una columna especificada para el idioma del contenido.

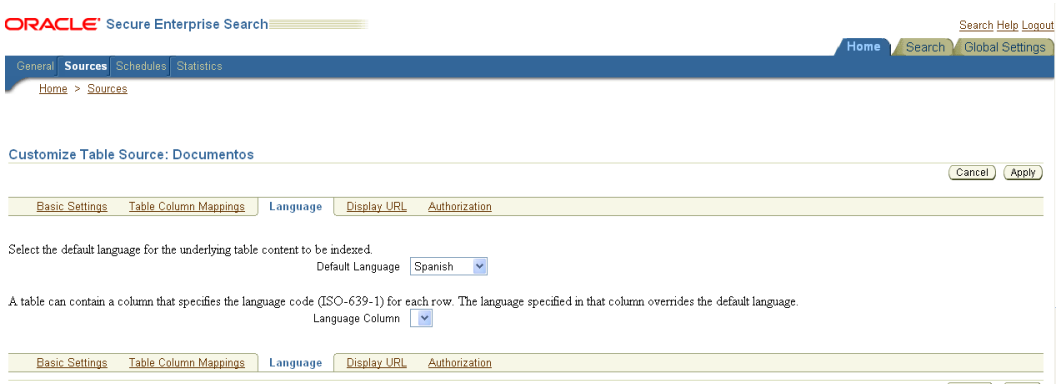

Figura 4. 5 Lenguaje para el contenido de la tabla subyacente para ser indexados

La siguiente opción es la “Visualización URL” como se muestra en la Figura 4.6, se puede utilizar una plantilla o utilizar la URL por defecto para visualizar los resultados después de la búsqueda.

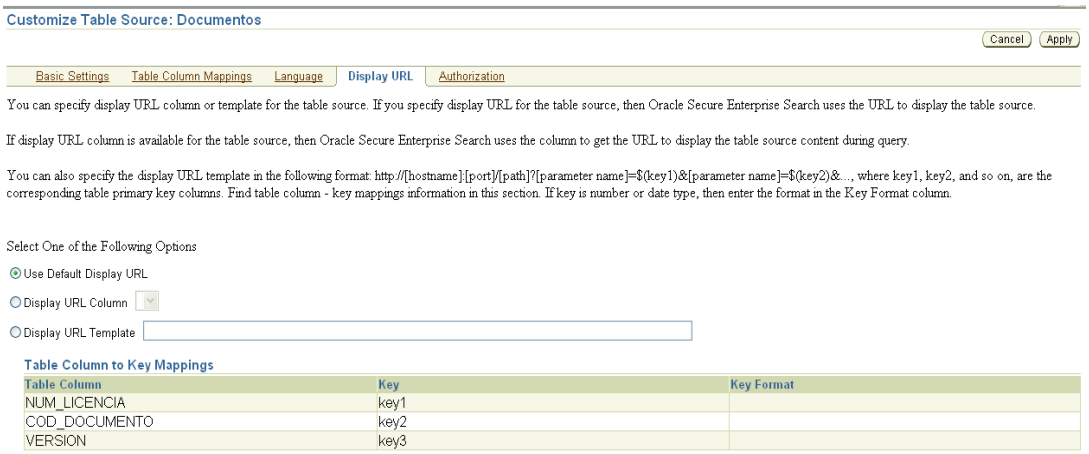

Figura 4. 6 Visualizar URL

En la sub pestaña de “Autorización” como se puede ver en la figura 4.7 existen dos opciones, la primera aparece por defecto y la segunda se puede hacer mediante un plug-in en el cual se obtiene los parámetros por medio de la opción “Obtener Parámetros”, no se configura ninguna autorización ya que todos los usuarios tendrán acceso.

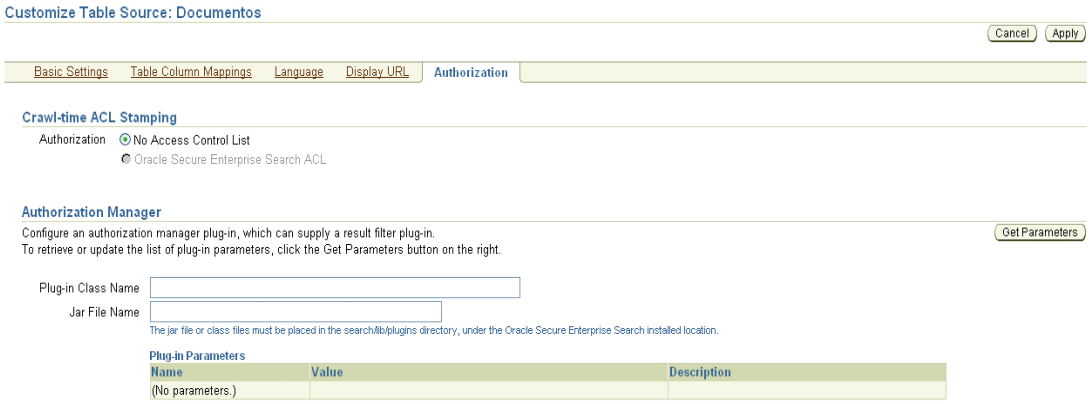

Figura 4. 7 Autorización

3.8.2.3 Creación de la Calendarización

En la sub pestaña “Calendarización (Schedules)”, se pueden crear las calendarizaciones que se desee siempre y cuando existan orígenes creados. Cuando se crea un origen se crea por defecto la calendarización la cual tiene el mismo nombre del origen, también se puede visualizar el estado de la

calendarización, el nombre del origen al que pertenece, el tipo de origen, un archivo de registro del resultado de la calendarización, la siguiente calendarización por ejecutar y las opciones de editar y eliminar como se muestra en la Figura 4.8.

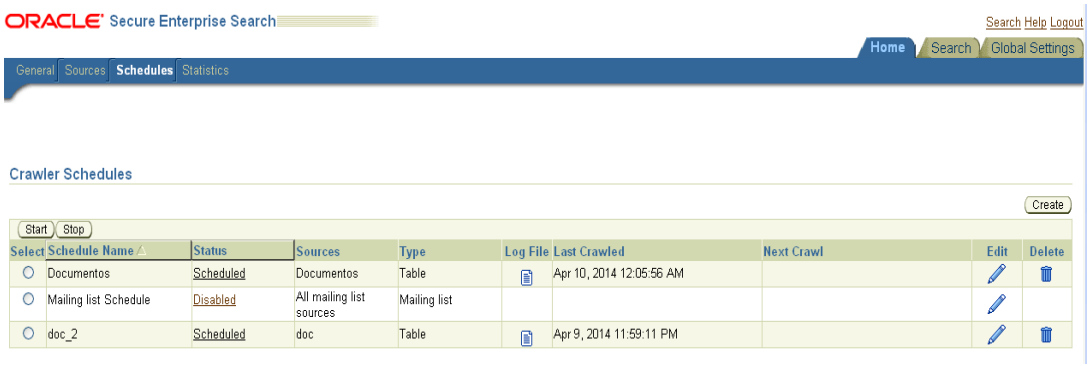

Figura 4. 8 Calendarizaciones (Schedules) de indexación

3.8.2.4 Administración de la Calendarización

Cuando se da clic en el estado de la calendarización como se puede ver en la Figura 4.9, se especifica el estado de la sincronización, con el nombre del origen, la fecha con hora de la última indexación, una opción para deshabilitar la calendarización, otra opción para ejecutar nuevamente la calendarización para que realice la indexación y el registro que se genera al finalizar la indexación.

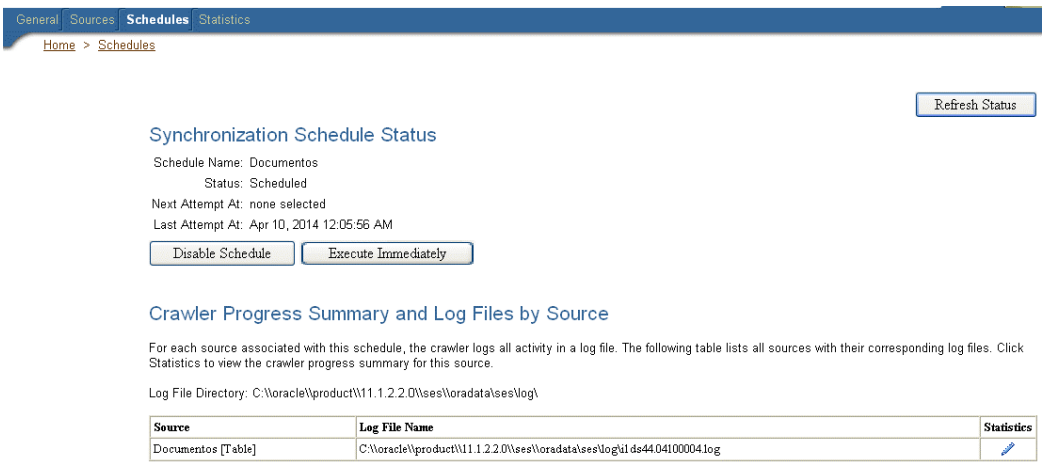

Figura 4. 9 Estado de la sincronización de la calendarización (Schedule)

Al empezar la sincronización de la calendarización se genera un archivo de registro que se puede ver el resumen de la indexación de los archivos, en la Figura 4.10 se muestra el progreso de la calendarización en el cual consta el tipo de origen, el nombre del origen, el tiempo en el que empezó, finalizó y transcurrió la sincronización, el tiempo total de la indexación, el tamaño total de documentos indexados, cuantos documentos fueron encontrados, indexados y cuantos documentos no fueron indexados.

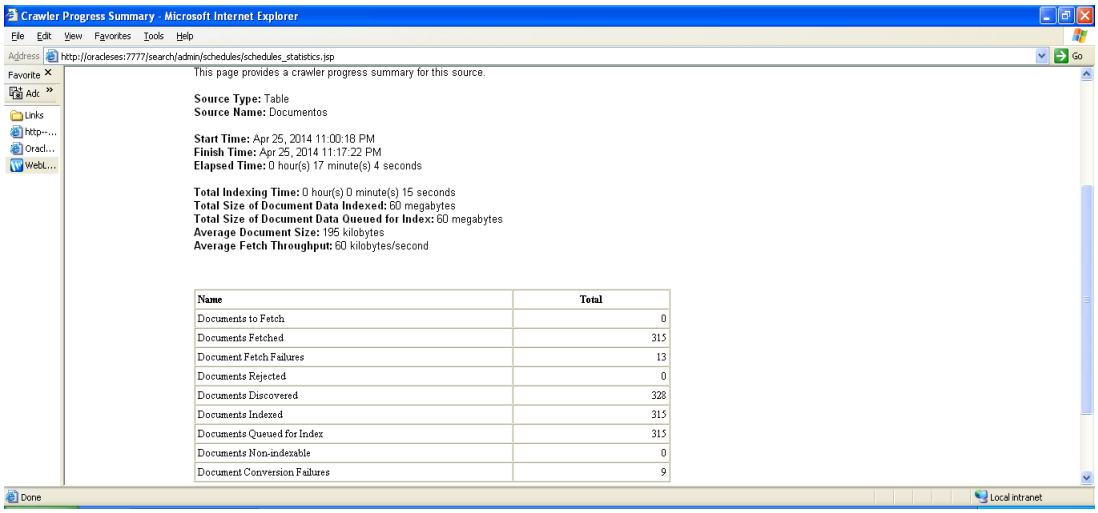

Figura 4. 10 Resumen de Progreso del Rastreo de Documentos

3.8.2.5 Creación de la funcionalidad para la búsqueda documental

Oracle SES cuenta con un API de Web Services, el cual se realizó la implementación por todos los métodos para ser integrado con la aplicación pero no se obtuvo buenos resultados ya que el Web Service está desarrollado bajo una tecnología anterior.

En la aplicación de Gestor Fiducia Fondos JEE se crea una página xhtml para la búsqueda documental, en la cual se agrega un “iframe” para integrar la pantalla de búsqueda del Oracle SES como se puede ver en la Figura 4.11.

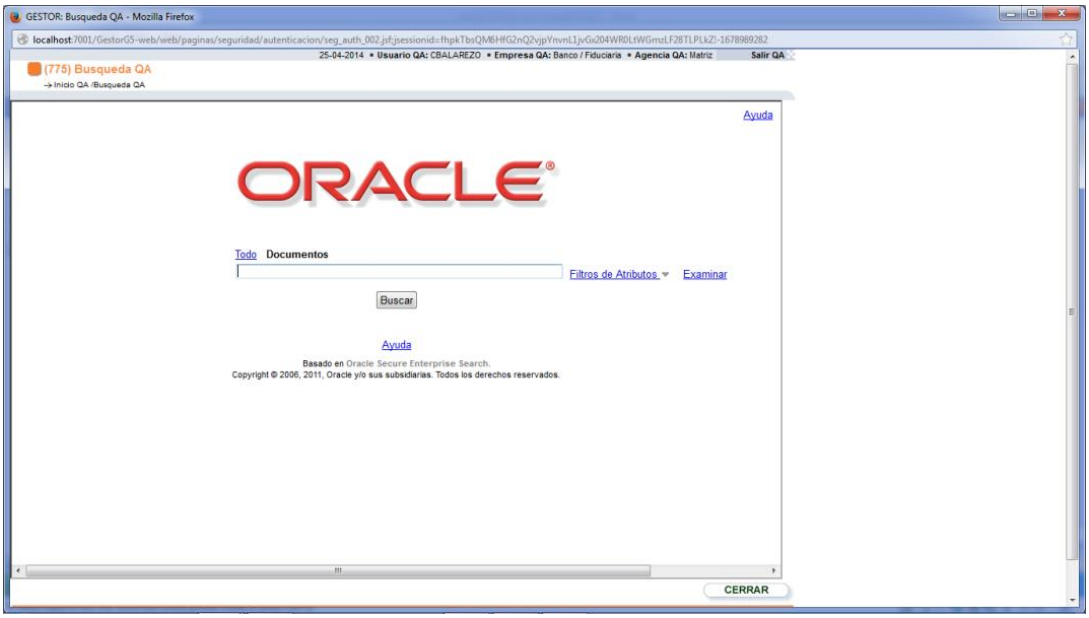

Figura 4. 11 Integración de Oracle SES con el sistema Gestor Fiducia Fondos JEE

En la Figura 4.12 se puede ver el resultado de la integración de la página de búsqueda del Oracle SES con la aplicación de Gestor, se ingresa una palabra clave “hibernate” para que realice la búsqueda de los documentos que contengan esa palabra.

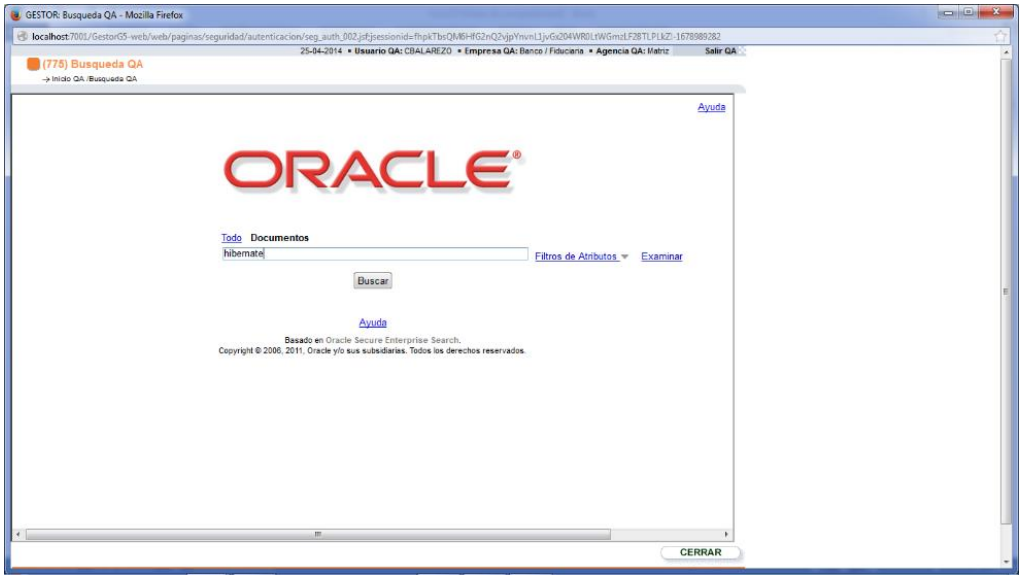

Figura 4. 12 Búsqueda en el grupo de Documentos

En la Figura 4.13 se muestra el resultado de la búsqueda con la lista de los documentos que contiene esa palabra y en la Figura 4.14 se muestra el documento abierto de la búsqueda.

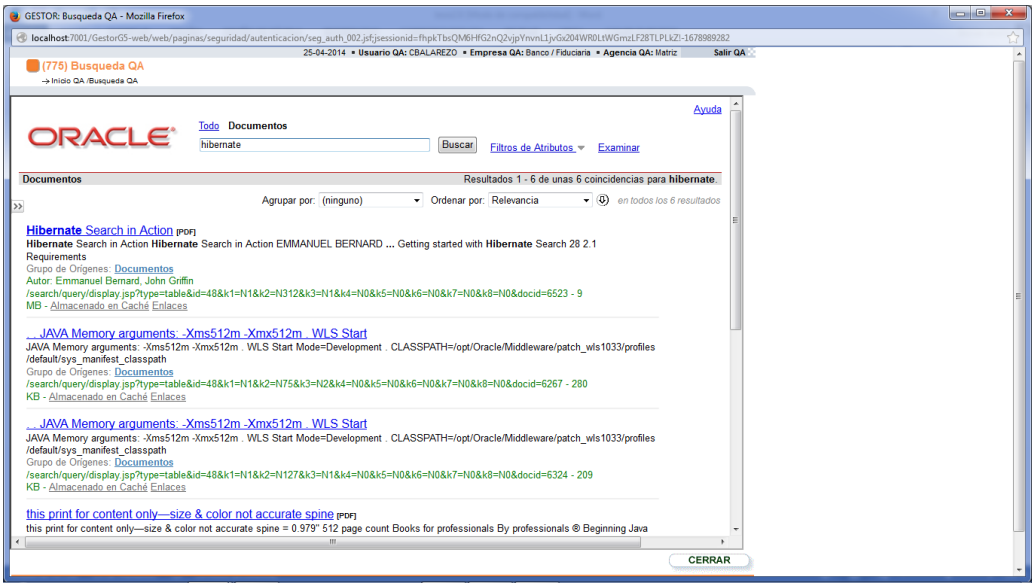

Figura 4. 13 Resultado de la búsqueda

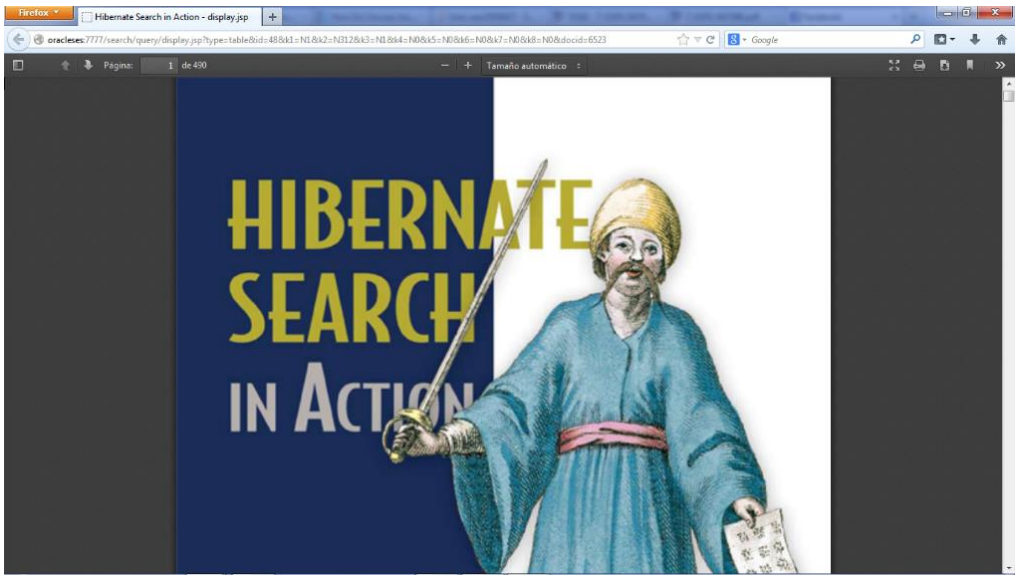

Figura 4. 14 Visualización del documento en PDF

4.2 Implementación de la Aplicación de Búsqueda utilizando Apache Lucene y ApacheTika

La aplicación que se ha desarrollado para demostrar el uso de las librerías Apache Lucene y Apache Tika, ha sido denominada Gestor Search, ya que en el proceso de definición, desarrollo y pruebas de la misma, se vio la factibilidad de crear un posible producto que forme parte del ecosistema Gestor.

3.8.2.6 Arquitectura de la Aplicación Gestor Search

Gestor Search es una aplicación empresarial construida bajo la plataforma JEE, que utiliza principalmente las librerías de los productos Apache Lucene y Apache Tika; para implementar el concepto de “Búsqueda Empresarial” en los diferentes documentos que se encuentran almacenados en la Base de Datos del sistema Gestor Fiducia Fondos JEE.

La aplicación desarrollada está constituida por dos funcionalidades principales: Funcionalidad de Indexación de Documentos y Funcionalidad de Búsqueda de Documentos, cada una de ellas cuenta con algunas funcionalidades internas.

La arquitectura lógica de la aplicación desarrollada se la describe en la Figura 4.15.

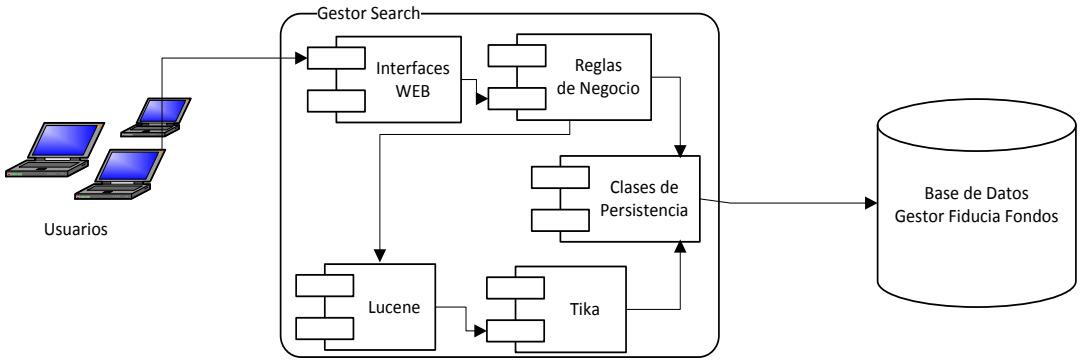

Figura 4. 15 Arquitectura de Gestor Search

En la figura se detallan las diferentes capas lógicas de la aplicación, entre las que tenemos:

- **Interfaces Web**

  En esta capa se encuentran las páginas JSF con sus respectivas clases “BanckingBean” para dar soporte a los servicios de indexación y búsqueda.

- **Reglas de Negocio**

  En esta capa se encuentra la implementación central de la aplicación, las clases de esta capa son las encargadas de recibir las llamadas desde la capa de “Interfaces Web” y procesar las solicitudes de búsqueda o indexación; para esto se sirven de la capa de persistencia y la utilización de las librerías de Lucene y Tika.

  A continuación se describe detalladamente la implementación de las funcionalidades de Indexación y Búsqueda de la aplicación Gestor Search.

3.8.2.7 Funcionalidad de Indexación

Esta funcionalidad es la más compleja de la aplicación y de la que se depende para la posterior implementación de la búsqueda. Apache Lucene es una librería que realiza la indexación en base los archivos de texto, sin embargo, los archivos de texto plano no son un estándar en las empresas en la actualidad, por lo que es necesario en primer lugar extraer el texto de los documentos que se encuentran en formato propietario como (PDF, archivos

de Microsoft Office, archivos de OpenOffice, etc.); para que este texto sea utilizado por Lucene y pueda ser indexado.

La extracción de texto desde los archivos en formato propietario se la realizó utilizando la librería Apache Tika, esta herramienta gratuita distribuida bajo la licencia Apache, permite extraer el texto desde un sin número de orígenes, tal como se explicó anteriormente en el marco teórico.

Para el proceso de extracción de texto, Apache Tika, lee un documento almacenado en disco, sin embargo, el sistema Gestor Fiducia Fondos JEE, como la mayoría de sistemas transaccionales guarda los documentos en la propia Base de Datos, por lo que fue necesario desarrollar una pequeña funcionalidad, para primero extraer el documento desde la base de datos y pasarlo a Apache Tika.

El esquema general de la funcionalidad de indexación se lo presenta en la Figura 4.16 que representa al diagrama de las clases desarrolladas en la implementación de la funcionalidad.

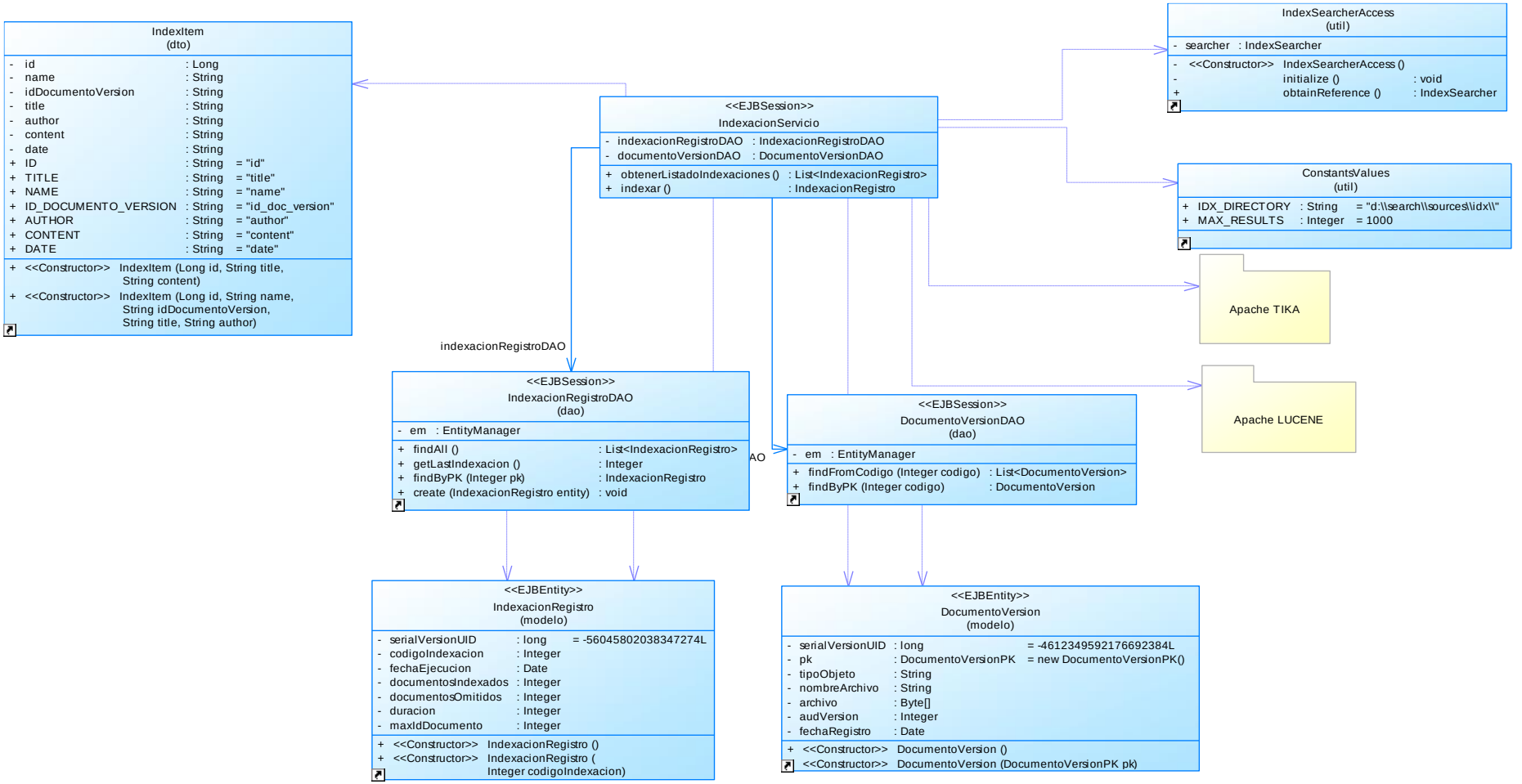

Figura 4. 16 Diagrama de Clases de Indexación

Como se puede apreciar el núcleo de la funcionalidad de Indexación es la clase ***IndexacionServicio***; esta clase define el método ***indexar***(); el cual es el encargado de implementar todo el proceso de indexación.

El proceso de indexación implementado es incremental, eso significa que cada vez que se ejecuta la indexación de archivos primero se busca en la base de datos información relacionada al último proceso de indexación para obtener el ID del último documento indexado, con esta información el proceso indexa solamente los nuevos archivos, es decir, los archivos cuyo ID sea mayor al obtenido; de esta forma, el proceso de indexación no tarda en su ejecución y tampoco se adueña de recursos que pueden afectar al rendimiento de todo el sistema.

Al encontrarse los archivos almacenados en la Base de Datos, se los extrae de la misma creando instancias de la clase ***DocumentoVersion***, el proceso de indexación toma estas instancias y las procesa; como parte del proceso la librería Apache Lucene crea una instancia de la clase ***IndexItem*** por cada documento procesado, en estos nuevos objetos se agrega la información correspondiente a Título del Documento, Autor del Documento y Fecha de Creación del Documento, cabe señalar que esta información no está disponible para archivos de texto plano.

El resultado de cada proceso de indexación se asigna en una instancia de la clase ***IndexacionRegistro***, el cual al finalizar el proceso es registrado en la Base de Datos, para ser utilizado la próxima vez que se ejecute el proceso; adicionalmente esta información también se presenta en la interfaz del usuario desde donde se invoca el proceso de indexación.

3.8.2.8 Funcionalidad de Búsqueda

Esta funcionalidad es la encargada de buscar en los archivos de índices generados la cadena de texto solicitado.

La librería Apache Lucene es quien implementa esta funcionalidad, sin embargo, es necesario incluir cierta lógica para obtener los fragmentos de texto coincidentes de cada documento encontrado y resaltar las palabras que coinciden con la cadena de búsqueda enviada; adicionalmente se debe presentar un link para descarga de cada uno de los archivos que forman parte del resultado en la interfaz de usuario.

El esquema general de la funcionalidad de búsqueda se lo presenta en la Figura 4.17 que representa al diagrama de las clases desarrolladas en la implementación de la funcionalidad.

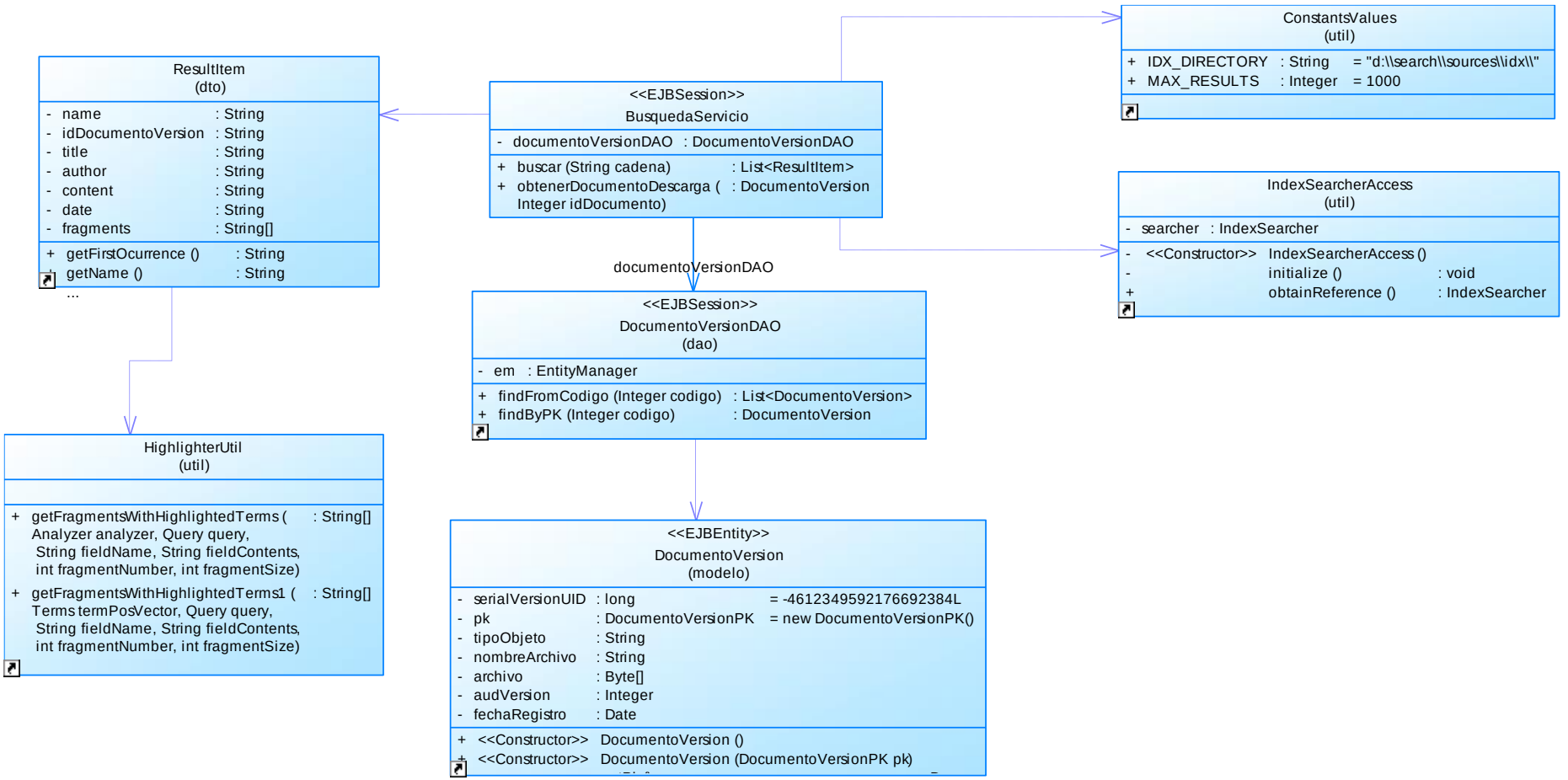

Figura 4. 17 Diagrama de Clases de Búsqueda

De igual forma que en la funcionalidad de indexación, la funcionalidad de búsqueda tiene una clase principal que es la clase ***BusquedaServicio***, como se puede apreciar en la figura 4.17.

El proceso de búsqueda recibe desde la interfaz de usuario, una cadena de búsqueda, la cual contiene el texto que el usuario desea buscar en los documentos indexados. La clase ***BusquedaServicio*** expone el método ***buscar(String cadena)*** que internamente implementa el proceso de búsqueda, para esto, en primer lugar se utiliza la librería Lucene, quien es la encargada de buscar la cadena de texto en los archivos de índices anteriormente generados, y retorna los archivos de resultado en orden de importancia; para esto utiliza una serie de algoritmos estándares de la industria, que fueron descritos en el marco teórico de este documento de tesis.

Una vez que se han explorado los índices y se ha obtenido los documentos que coinciden con la cadena de búsqueda, se utiliza una funcionalidad propia de Lucene, la cual se encarga de encontrar los primeros diez párrafos en los cuales se encuentran términos coincidentes con la cadena de búsqueda.

Con estos resultados, se generan instancias de la clase **ResultItem** incluyendo además la información necesaria para generar el vínculo de descarga del documento en caso de que el usuario así lo requiera; estas instancias son enviadas a la capa de presentación (interfaz de usuario) para ser desplegados.

3.8.2.9 Integración de GestorSearch con el sistema Gestor Fiducia Fondos

La integración de la aplicación Gestor Search con el sistema Gestor Fiducia Fondos, se la ha realizado por medio de la inclusión de dos

funcionalidades, las cuales tienen internamente elementos de tipo “iframe” los cuales hacen llamado a la funcionalidad de indexación y a la funcionalidad de búsqueda, de esta forma se logra una independencia de Gestor Fiducia Fondos con la aplicación Gestor Search, permitiendo a esta última su propia evolución, disminuyendo al máximo la cohesión entre las dos aplicaciones.

En la figura 4.18 se muestra la interfaz de indexación con el resultado de cuantos documentos se indexaron, la fecha, duración. En la figura 4.19 se muestra la pantalla de búsqueda se ingresa cualquier palabra y en la figura 4.20 se muestra el despliegue de los resultados de búsqueda, como se puede observar en esta última imagen, la interfaz de usuario de despliegue de resultados es similar a la del buscador Google para que el usuario sienta familiaridad al momento de utilizar esta funcionalidad.

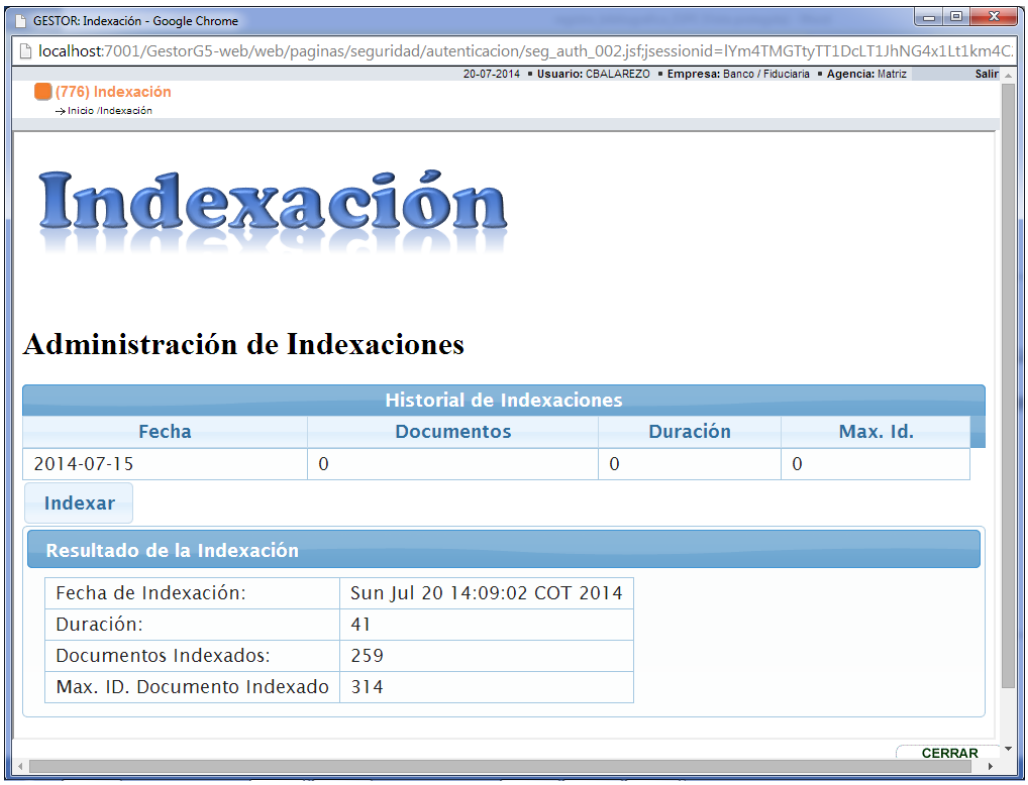

Figura 4. 18 Administración de Indexaciones

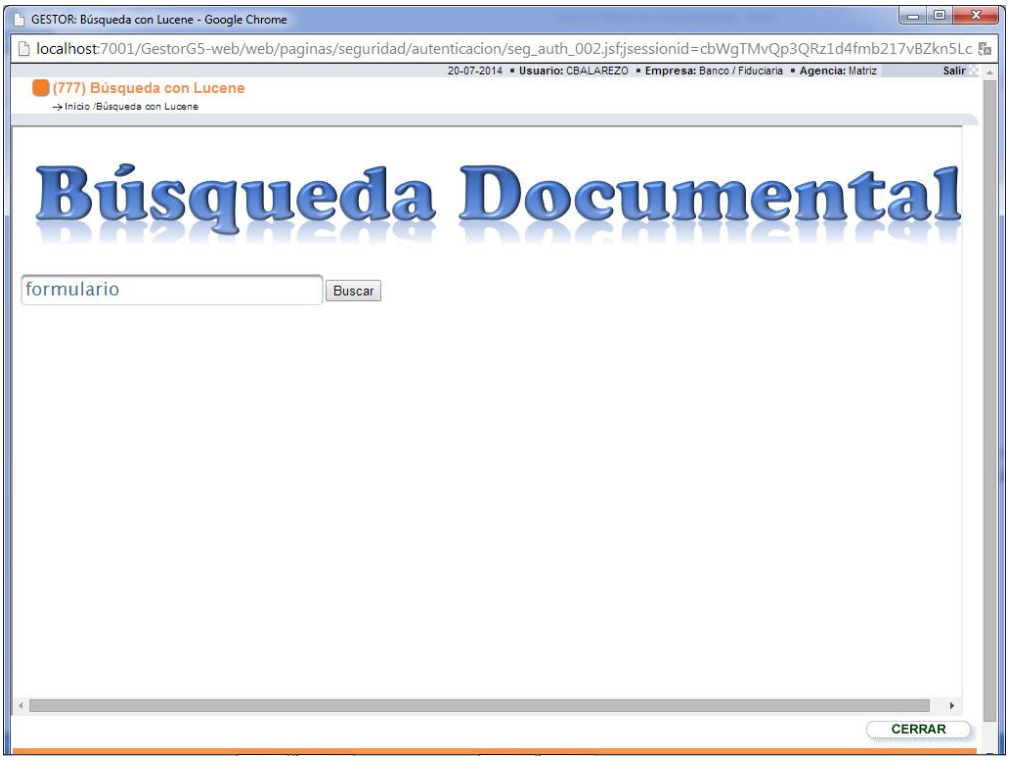

Figura 4. 19 Página de búsqueda documental con Lucene

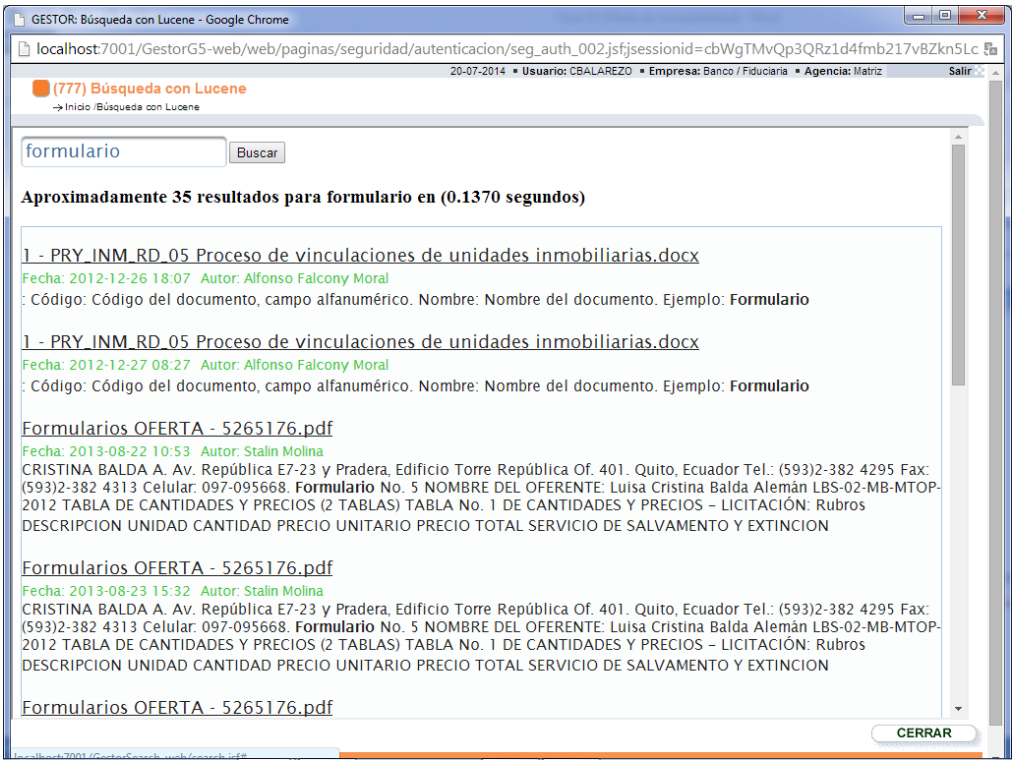

Figura 4. 20 Resultados de la búsqueda

4.3 Selección de la Herramienta de Búsqueda Empresarial

4.3.1 Definición de Parámetros de Evaluación

En los capítulos anteriores se ha realizado la descripción de las dos posibles soluciones a ser implementadas como herramienta de Búsqueda Empresarial en el sistema Gestor Fiducia Fondos JEE, la primera opción corresponde al producto propietario de Oracle Corporation llamado “Oracle Secure Enterprise Search” y la segunda opción a la aplicación desarrollada llamada “Gestor Search” una aplicación desarrollada como producto de investigación de esta tesis de grado, la cual utiliza librerías de licencia Open Source Software como Apache Lucene y Apache Tika.

Las dos posibles soluciones brindan similar funcionalidad de acuerdo a las pruebas realizadas, sin embargo, la empresa GestorInc. S.A. ha definido que para la selección de una de las dos se deberá tomar en cuenta los siguientes parámetros: estabilidad, costos, facilidad de uso y escalabilidad.

A continuación se describe el significado que da GestorInc. S.A. a cada uno de los parámetros mencionados y su relación con el objetivo final de este trabajo de investigación que es la selección de una herramienta de Búsqueda Empresarial para su implementación en el sistema Gestor Fiducia Fondos JEE.

- **Estabilidad**

  Este parámetro se refiere a la medición de la ejecución normal de la aplicación de Búsqueda Empresarial y sus funcionalidades asociadas como el proceso de Indexación, la funcionalidad de Búsqueda y Despliegue de Resultados. La normal ejecución de la aplicación es clave para GestorInc. S.A., ya que de esto depende seguir manteniendo la confianza ganada en estos 17 años de existencia de la empresa con sus clientes.

- **Costos**

  La idea principal de ofrecer la aplicación de Búsqueda Empresarial como un beneficio adicional para los clientes del sistema Gestor Fiducia Fondos JEE es la de poder promocionar en la misma, los diversos servicios y nuevos productos de Gestor, utilizando formas similares a las del buscador “Google”. Para mantener este esquema lo ideal es que los clientes tengan que realizar una mínima inversión económica, Gestor antes que beneficiarse de la venta del producto lo que desea es ofrecer un nuevo servicio a sus clientes y promocionarse con los mismos.

- **Facilidad de Uso**

  Al ser un servicio “gratuito” que espera brindar Gestor a sus clientes, es de suma importancia el que la instalación, capacitación y mantenimiento de la aplicación necesiten la mínima cantidad de recursos, lo ideal es que no se necesite personal especializado (recursos de consultoría de Gestor) para su instalación, configuración y ejecución.

- **Escalabilidad**

  Este parámetro de evaluación calificará la capacidad de la aplicación de Búsqueda de ejecutarse en diferentes arquitecturas de hardware y software; como se ha explicado anteriormente se desea que la aplicación pueda ser desplegada en cualquier arquitectura de los clientes, sin que estos tengan que incurrir en gastos adicionales.

4.4 Proceso de Evaluación

Una vez que se ha descrito los parámetros de evaluación y su relación con el objetivo final de este trabajo de investigación, es necesario calificar a cada uno de estos con un factor de importancia; el cual ayudará en la elaboración de la tabla final de calificaciones de cada proveedor.

El factor de importancia será un valor de 1 a 3; siendo 3 el valor que se dará a los parámetros más importantes en la evaluación; luego de conversar con el personal de las unidades de Soporte y Sistemas de Gestor se ha elaborado la Tabla 4.1, en la que se muestra la calificación de importancia a cada uno de los parámetros anteriormente detallados:

Tabla 4. 1

Factor de Importancia de los parámetros de evaluación

Parámetro	Factor Importancia
Estabilidad	3
Costos	3
Facilidad de Uso	1
Escalabilidad	2

La calificación para cada uno de los parámetros será del 1 al 4; siendo 4 la máxima calificación que cada aplicación de Búsqueda Empresarial puede tener en un determinado parámetros y 1 la mínima calificación.

De acuerdo a las características detalladas al inicio del capítulo las dos aplicaciones de Búsqueda y de lo expuesto en el proceso de selección se ha procedido a realizar la siguiente tabla de evaluación de cada una de las aplicaciones.

Oracle Secure Enterprise Search

La Tabla 4.2 muestra la calificación para cada uno de los parámetros de la aplicación Oracle Secure Search.

Tabla 4. 2

Calificaciones detallas de Oracle Secure Search

Parámetro	Calificación	Factor	Total
Estabilidad	2	3	6
Costos	1	3	3
Facilidad de Uso	3	1	3
Escalabilidad	3	2	6
Total			18

Oracle Secure Entreprise Search ha obtenido calificaciones bajas en los parámetros Costos y Estabilidad, debido a que al ser un producto de Oracle Corporation, tiene un costo aproximado de $30.000,00 USD por procesador lo cual obliga al cliente a invertir en la compra de esta licencia para su utilización; adicionalmente al momento de realizar las pruebas de uso con esta aplicación se encontró diversos problemas, relacionados con su instalación y ejecución; principalmente con el proceso de Indexación de los archivos.

Gestor Search

La Tabla 4.3 muestra la calificación para cada uno de los parámetros de la aplicación Gestor Search.

Tabla 4. 3

Calificaciones detallas de Gestor Search

Parámetro	Calificación	Factor	Total
Estabilidad	4	3	12
Costos	4	3	12
Facilidad de Uso	3	1	3
Escalabilidad	4	2	8
Total			**35**

Gestor Search ha obtenido la máxima calificación en los parámetros de Estabilidad, Costos y Escalabilidad debido a que no se debe pagar ningún costo por concepto de licencia de las librerías Apache Tika y Apache Lucene; adicionalmente en las pruebas realizadas no se presentaron problemas de “Estabilidad” es decir la aplicación se ejecutó normalmente; al ser una aplicación desarrollada siguiendo los estándares de la plataforma JEE la misma no presenta dificultades para ejecutarse en cualquier servidor de aplicaciones certificado.

Resumen Final

La Tabla 4.4 muestra la evaluación final de las aplicaciones de Búsqueda Empresarial.

Tabla 4. 4

Resultados Finales.

Proveedor	Puntaje Final
Oracle Secure Enterprise Search	18
Gestor Search	35

Una vez que se ha realizado la calificación respectiva a cada uno de los parámetros de selección y se ha utilizado el “Factor de Importancia” para obtener los puntajes finales, se ha seleccionado a la aplicación “Gestor Search” como la aplicación a ser implementada para solventar la Búsqueda Empresarial en el Sistema Gestor Fiducia Fondos.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Terminado el desarrollo de la aplicación Gestor Search utilizando las librerías Apache Lucene y Tika y su integración con el sistema Gestor se ejecutaron pruebas en las que se obtuvo buenos resultados en tiempos de búsqueda, menores a 2 segundos utilizando un conjunto de 500 archivos los cuales aproximadamente representan 10GB de espacio en disco.

- Para la empresa GestorInc. S.A. fue una gran ayuda la implementación de esta aplicación ya que se consiguió facilitar el proceso de búsqueda que los clientes realizan sobre los documentos almacenados en el repositorio del sistema Gestor G5 Trust.

- La ventaja de la implementación de la aplicación Gestor Search es su compatibilidad con el sistema Gestor tanto en requisitos de hardware y software, por esta razón los clientes no se ven en la obligación de incurrir en costos relacionados a la adquisición de una infraestructura.

5.2 Recomendaciones

- Para la implementación de Gestor Search con Apache Lucene y Tika no se debe adquirir un equipo dedicado ya que estará integrado en el mismo sistema y así facilitará su manejo.

- Se recomienda que la aplicación Gestor Search incluya nuevas funcionalidades como el acceso a documentos de acuerdo al rol de cada usuario para que sea vista como un producto comercial por los clientes.

- La funcionalidad de indexación de documentos no está diseñada para realizarse de forma automática cuando se registran nuevos archivos, ya que si se lo hace puede llegar a saturar la base de datos; por este motivo se recomienda el desarrollo de una funcionalidad de Calendarización Automática, similar a la que tiene el producto Oracle Search.

- Se debe tener en cuenta las nuevas versiones de Apache Lucene y Tika para que la aplicación “Gestor Search” no quede obsoleta, esté actualizada y se pueda desarrollar nuevas funcionalidades.

Bibliografía

Chris A. Mattmann, J. L. (2012). *Tika in Action.* Shelter Island, NY 11964: Manning Publications Co.

Erik Hatcher, O. G. (2004). *Lucene in Action.* Canada: Manning Publications.

Foundation, T. A. (2011-2012). *The Apache Software Foundation*. Obtenido de http://lucene.apache.org/core/

López, C. P. (2005). *Oracle 10g Administración y análisis de bases de datos.* México D.F.: Alfaomega Grupo Editor.

Marlene Theriault, A. N. (2002). *Manual de Seguridad de Oracle.* Madrid-España: McGraw-Hill.

Oracle. (2006 - 2007). *Oracle Secure Enterprise Search Java API Reference*. Obtenido de http://docs.oracle.com/cd/E10502_01/doc/search.1018/e10465/toc.htm

Oracle. (2006). *Oracle Secure Enterprise Search APIs*. Obtenido de http://docs.oracle.com/cd/B19306_01/search.102/b32259/api.htm#i1004617

Oracle. (2006,2008). *Oracle SecureEnterprseSearch, Búsqueda Básica*. Obtenido de http://ses.gddc.pt/search/query/ohw/help?navId=2&navSetId=_&locale=es&vtTopicFile=query_cs_help_es/query_cs_basic.html

Oracle. (2009). *Developer Tutorial: Building Custom Plug-Ins*. Obtenido de http://st-curriculum.oracle.com/tutorial/SESDevTutorial/index.htm

Oracle. (2009). *Tutorial de Administración de Oracle Secure Enterprise Search*. Obtenido de http://st-curriculum.oracle.com/tutorial/SESAdminTutorial/index.htm

Oracle. (s.f.). *Administración de la consola del Oracle Weblogic Server en Oracle SES*. Obtenido de http://docs.oracle.com/cd/E21698_01/admin.1122/e21605/tuning009.htm

Oracle. (s.f.). *Administración de Oracle SES*. Obtenido de http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/start002.htm

Oracle. (s.f.). *Búsqueda Federada de Oracle Secure Enterprise Search*. Obtenido de http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/over005.htm

Oracle. (s.f.). *Búsqueda segura*. Obtenido de http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/oessecurity002.htm#BGBCHBGA

Oracle. (s.f.). *Características de Oracle Secure Enterprise Search*. Obtenido de http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/over005.htm

Oracle. (s.f.). *Estructura, componentes y administración de Oracle Secure Enterprise Search*. Obtenido de http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/over003.htm#i1005612

Oracle. (s.f.). *Glosario de la Administración de Oracle SES*. Obtenido de http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/glossary.htm#CHDEHJGB

Oracle. (s.f.). *Informe técnico de Oracle Secure Enterprise Search*. Obtenido de http://www.oracle.com/technetwork/search/oses/overview/ses-technical-whitepaper-11-131406.pdf?ssSourceSiteId=ocomlad

Oracle. (s.f.). *Instalación de Oracle Secure Enterprise Serach*. Obtenido de http://docs.oracle.com/cd/B32393_01/doc/install.1018/b32263/toc.htm

Oracle. (s.f.). *Oracle Secure Enterprise Search 11g*. Obtenido de http://www.oracle.com/technetwork/search/oses/overview/index.html

Oracle. (s.f.). *Oracle Secure Enterprise Search Database*. Obtenido de http://www.oracle.com/lad/products/database/secure-enterprise-search/index.html

Oracle. (s.f.). *Tipos de Fuentes*. Obtenido de http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/over002.htm#BGBEDJFE

Oracle. (s.f.). *Tipos de fuentes de Oracle Secure Enterprise Search*. Obtenido de

http://docs.oracle.com/cd/E25054_01/doc.1111/e17332/over002.htm#BGBEDJFE

VishwanathSreeraman, D. C. (2010). *Guía para el Administrador de Oracle Secure Enterprise Search 11g Release 1 (11.1.2.0.0).* Estados Unidos.