

RESUMEN

Las actividades tendientes a la búsqueda de recursos naturales, siguen teniendo gran relevancia para la economía de países como el nuestro y concretamente las orientadas a la ubicación y exploración de yacimientos petrolíferos.

Se toma como caso de estudio el de la empresa, como la que en nuestro país se dedica a este tipo de actividades y, en donde la información recopilada y que es resultante de los proyectos de exploración es de vital importancia como parte del proceso de producción de petróleo. Dicha información es captada en formatos Excel y requiere ser cargada en una base de datos MySQL, por lo que surge la problemática de extraer, transformar y cargar información de libros Excel a un ambiente de base de datos.

Por lo antes expuesto, el presente trabajo de tesis está orientado hacia el tratamiento corporativo de datos y la recuperación dinámica de información, en un esquema donde distintas entidades interactúan como parte de un ambiente integrado de generación de información, para el proceso de toma de decisiones.

Para solucionar la problemática antes mencionada, se consideró la tecnología surgida del proyecto Jakarta POI API, la cual permite acceder al manejo de archivos basados en el formato de Documento Compuesto OLE 2 de Microsoft; entre los que se incluyen los de tipo Excel (.XLS); y es mediante la combinación de sus paquetes POIFS y HSSF, en conjunción con Java, que se pueden leer y escribir archivos como los antes mencionados. Se considera que el uso de dicha API conjuntamente con el esquema Java es, tecnológicamente hablando, una combinación poderosa y representa una solución viable, a un bajo costo y de fácil integración con el entorno. La integración de lo anterior, con un nuevo ambiente de captura, ágil y estructurado, para captar en Excel la información de los rubros relacionados con un proyecto de exploración, es parte de la solución.

La aplicación que en este trabajo se presenta, aporta entornos innovadores para el soporte de los requerimientos de proceso y actualización de información, orientados a la toma de decisiones y, conlleva un punto de referencia al favorecer la generación de nuevas soluciones.

Las pruebas realizadas se basan en la simulación de periodos de exploración, vía la captación de datos mediante el capturador Excel, en donde todos los datos captados son ficticios y sin relación con algún proyecto real.

Para evaluar los alcances de la parte dinámica y analizar los resultados obtenidos, se utilizaron los libros generadores previamente en Excel con la base de datos de prueba, previamente establecida.

ABSTRACT

Natural resources searching efforts, still have great relevance to the economies of countries like ours, and specifically targeted to the exploration and location of oil fields.

A case study is the company, such as in our country engages in such activities, where the exploration projects information results are gather and become as main part of the production process oil. This information is collected in Excel formats and needs to be loaded into a MySQL database, so, the problem arises to extract, transform and load Excel information into a database.

As above, this thesis work is focused towards the treatment of corporate data and dynamic information retrieval, in a scheme where different entities interact as part of an integrated environment to generate information for decision-making process.

To the problem mentioned above, the Jakarta POI API project technology was included from as a option solution, which one allows access to the management of files based on the Microsoft OLE 2 Compound Document format including the Excel type (.XLS) and to uses a combination of its packages POIFS and HSSF in conjunction with Java, it can read and write files as above. The use of the Jakarta POI API with Java technology is, technologically speaking, a powerful combination and represents a viable solution, at a low cost and easy environment integration. Combining the above, implemented with the new flexible and structured Excel data capture tool for exploration projects, is a solution's approach.

The implementation in this thesis work, provides innovative environments for the information updating process aimed to the decision making support and is a reference to facilitate the generation of new solutions.

The assesment was based on simulation exploration periods, via the Excel data capture, where all data collected were fictitious and unrelated to any real project.

To assess the extent of the dynamic part and analyze the results, were used the Excel books that previously had been generated and the data base already exist.

ÍNDICE

Resumen	i
Abstract.....	iii
Índice.....	v
Lista de figuras.....	vii
Lista de tablas	ix
Glosario.....	x
CAPÍTULO 1. INTRODUCCIÓN	1
1.1 Definición del problema.....	1
1.2 Objetivos.....	2
1.2.1 Objetivo general	2
1.2.2 Objetivos específicos.....	2
1.3 Justificación	2
1.4 Alcances y limitaciones.....	3
1.5 Beneficios esperados	4
1.6 Organización del documento	5
CAPÍTULO 2. ANTECEDENTES.....	9
2.1 Marco teórico.....	9
2.1.1 Los archivos Excel.....	9
2.1.1.1 Documento tipo Libro	9
2.1.1.2 Documento tipo Hoja	10
2.1.2 POI API de Apache Jakarta.....	11
2.1.2.1 El modelo de documento Excel.....	11
2.2 Estado del arte.....	14
2.2.1 Tecnologías para el manejo de formatos tipo Excel	14
2.2.1.1 Paquetes Java para leer y escribir archivos de documentos tipo Excel	16
2.2.1.2 El Api Java Excel	17
2.2.1.3 El Api JExcel.....	18
2.2.2 Casos de éxito basados en el uso del POI API de Jakarta	19
CAPÍTULO 3. ANÁLISIS Y DISEÑO	23
3.1 Introducción.....	23
3.2 Análisis de requerimientos.....	24
3.2.1 Especificación inicial	24
3.2.2 Interfaces gráficas.....	25
3.2.3 Datos de entrada	25
3.2.3.1 Recursos Excel.....	25
3.2.3.2 Usuarios de la aplicación	26
3.2.3.3 Referencia de recursos Excel	26
3.2.3.4 Costos totales.....	26
3.2.4 Soporte tecnológico requerido	26
3.3. Perfiles de usuario	27
3.3.1 Perfil usuario administrador.....	27
3.3.2 Perfil usuario básico	27
3.4 Diseño de la aplicación.....	27
3.4.1 Objetivos	27
3.4.2 Diseño de interfaces	28
3.5 Requerimientos de desarrollo.....	29
3.5.1 Módulo de captura Excel	29

3.5.2 Inicio de sesión	29
3.5.3 Administración de usuarios	30
3.5.4 Módulo de recursos Excel	30
3.5.5 Módulo de transmisión de información.....	30
3.5.5.1 Lectura de archivos Excel	30
3.5.5.2 Procesamiento de información	31
3.5.5.3 Actualización de base de datos	31
3.5.6 Consulta de información actualizada	31
3.5.7 Cierre de sesión	31
3.6 Arquitectura de la aplicación	31
3.7 Casos de uso	32
3.7.1 Actores y sus casos de uso	32
3.7.2 Escenarios del actor “usuario administrador”	34
3.7.2.1 Administrar datos de usuarios.....	34
3.7.2.2 Administrar recursos Excel.....	35
3.7.2.3 Transmitir información	35
3.7.2.4 Consultar información actualizada	37
3.7.3 Escenarios del actor “usuario básico”	38
3.7.3.1 Agregar recursos Excel	38
3.7.3.2 Consultar recursos Excel	38
3.7.4 Diagrama de secuencia del acceso a la aplicación	39
3.8 Diagrama relacional	40
3.9 Paquetes de clases que integran la aplicación	42
3.10 Diagramas de clases.....	43
3.10.1 Descripción de clases.....	43
3.10.1.1 La clase Main	43
3.10.1.2 La clase Consultar.....	44
3.10.1.3 Diagrama de clases de la aplicación	45
3.10.1.4 La clase FileUpload	47
CAPÍTULO 4. DESARROLLO DE LA APLICACIÓN	49
4.1 Introducción.....	49
4.2 Esquema del flujo de la operación	49
4.3 Descripción del ambiente de operación	50
4.4 Soporte de la operación.....	51
4.5 Funcionalidad de la aplicación	53
4.5.1 Acceso a la aplicación	54
4.5.2 Verificación de usuarios	55
4.5.3 Módulos del entorno de operación del usuario administrador .	55
4.5.3.1 Usuarios y recursos Excel.....	55
4.5.3.2 Actualización	57
4.5.3.3 Consulta	58
4.5.4 Ambiente del usuario básico.....	58
4.5.4.1 Módulo de registro de recursos Excel.....	59
4.6 Localización de la aplicación	59
CAPÍTULO 5. PRUEBAS Y RESULTADOS.....	63
5.1. Captura de datos	63
5.2. Pruebas iniciales	66
5.3. Recuperación y carga de recursos Excel.....	67
5.4 Transmisión de datos de recursos Excel.....	74
5.5. Consulta a la base de datos de prueba	79

5.6. Módulo de usuarios.....	80
CAPÍTULO 6. CONCLUSIONES	83
6.1. Análisis de resultados	83
6.2. Logros alcanzados.....	84
6.3. Contribuciones	86
6.4. Trabajo futuro	87
BIBLIOGRAFÍA	89
APÉNDICES	91
Apéndice A Código para generar un documento tipo Excel	91
Apéndice B Manual de uso del módulo de captura para proyectos de exploración de yacimientos petrolíferos.....	95
Apéndice C Manual de uso de la aplicación para la manipulación y recuperación dinámica de información utilizando el Jakarta POI API.....	105
Apéndice D Tablas que integran la base de datos de prueba	115
Apéndice E Requerimientos para el desarrollo y soporte de la aplicación	116
Apéndice F Procedimientos en Java y JSP	117

LISTA DE FIGURAS

Figura 2.1.1	1.1 Estructura general de un documento tipo LIBRO	10
Figura 2.1.1.2.1	Estructura general de un documento tipo HOJA	10
Figura 2.1.2	Resumen del diagrama de clases de HSSF usermodel	12
Figura 2.1.2.1	Estructura del modelo del documento Excel	12
Figura 2.1.2.2	Libro Excel generado mediante el POI API de Jakarta	15
Figura 2.2.1.2	Componentes principales del Api Java Excel.....	17
Figura 3.6.1	Arquitectura de la aplicación	32
Figura 3.7.1.1	Diagrama de contexto de la aplicación para la manipulación y recuperación dinámica de información	33
Figura 3.7.2.2	Diagrama de casos de uso para la administración de usuarios y de recursos Excel	36
Figura 3.7.2.3	Diagrama de casos de uso de la transmisión de información ...	37
Figura 3.7.2.4	Diagrama de casos de uso para la consulta de la base de datos.....	37
Figura 3.7.3.2	Diagrama de casos de uso del usuario básico para las opciones de agregar y consultar recursos Excel	39
Figura 3.7.4.1	Diagrama de secuencia del acceso a la aplicación de acuerdo al tipo de usuario	40
Figura 3.8	Diagrama entidad-relación	41
Figura 3.9	Diagrama de paquetes de clases que integran la aplicación	42
Figura 3.10.1.1	Diagrama de clases de la clase Main.....	44
Figura 3.10.1.2	Diagrama de clases de la clase Consultar	45
Figura 3.10.1.3	Diagrama de clases de las clases que forman parte de la aplicación	46
Figura 4.1	Flujo general de la operación	50
Figura 5.1.1	Opciones del menú del capturador Excel.....	64
Figura 5.1.2	Hoja de totales del capturador Excel.....	65
Figura 5.2.1	Interfaz gráfica para el ingreso a la aplicación	66
Figura 5.3.1	Interfaz gráfica para el módulo de registro de recursos Excel.....	67
Figura 5.3.2	Interfaz gráfica que muestra lista de archivos a elegir	68
Figura 5.3.3	Archivo seleccionado generado por el capturador Excel	68
Figura 5.3.4	Archivo seleccionado que será enviado para su registro en la base de datos de prueba.....	69
Figura 5.3.5	Indica al usuario que la actualización fue exitosa	69
Figura 5.3.6	El archivo elegido ha sido cargado y puede ser consultado	70
Figura 5.3.7	Recursos Excel registrados en la base de datos de prueba	71
Figura 5.3.8	Indica que se realizará la acción de Borrar registro	72
Figura 5.3.9	Confirmar la baja de la elección realizada	72
Figura 5.3.10	Informa al usuario de la eliminación de la referencia elegida.....	72
Figura 5.3.11	Ha sido realizada la actualización y eliminada la referencia del registro elegido.....	73
Figura 5.4.1	Menú de operación de la aplicación para el usuario administrador.....	74
Figura 5.4.2	Interfaz que permite elegir el portafolio de datos a procesar	75
Figura 5.4.3	Archivos disponibles para ser procesados	76
Figura 5.4.4	Despliegue de datos del portafolio elegido	76
Figura 5.4.5	Valores importados del portafolio de información	77
Figura 5.4.6	Se indica al usuario que la información fue actualizadda.....	78

Figura 5.4.7 Aviso del por qué no se realizó la actualización respectiva	79
Figura 5.5.1 Resultao de la consulta a la base de datos de prueba	79
Figura 5.6.1 Interfaz gráfica del módulo de usuarios	80
Figura A.1 Uso del Jakarta POI API de Apache	91
Figura B.1 Plantilla original Excel de captura	95
Figura B.2 Módulo de captura para proyectos de exploración	97
Figura B.3 Totales del módulo de captura para proyectos de exploración	98
Figura B.4 Opciones de inversión operacional.....	99
Figura B.5 Hoja de captura correspondiente a servicios generales	100
Figura B.6 Parte final de la hoja correspondiente a servicios generales.....	101
Figura B.7 Pantalla de captura de años y nombre para la opción elegida	102
Figura B.8 Hoja de estudios de identificación y evaluación de Sistemas Generadores	103
Figura B.9 Se aprecian los totales por cada objeto para la opción elegida	103
Figura C.1 Inicio de sesión de usuario	105
Figura C.2 Módulo de recursos Excel (versión limitada)	106
Figura C.3 Entorno de operación del sistema	107
Figura C.4 Interfaz del módulo de recursos Excel.....	108
Figura C.5 Interfaz del módulo de usuario	109
Figura C.6 Alta de usuario.....	109
Figura C.7 Modificación de datos del usuario	110
Figura C.8 Eliminación del registro del usuario	110
Figura C.9 Proceso de lectura y transformación de información residente en archivos de recursos Excel.....	111
Figura C.10 Datos ha ser ingresados en las tablas de costos	112
Figura C.11 Resultado del proceso de actualización	112
Figura C.12 Aviso de proceso previamente actualizado	113
Figura C.13 Resultado de la consulta a la base de datos ya actualizada	113
Figura C.14 Interfaz de acceso al sistema para una nueva sesión	114

LISTA DE TABLAS

Tabla 2.1.2. Paquetes que integran el módulo HSSF	11
Tabla 2.2.1.1. Paquetes Java para leer y escribir archivos de documentos tipo Excel.....	16
Tabla 2.2.2.1. Casos de éxito basados en el uso del POI API de Jakarta	19
Tabla 3.7.1. Casos de uso de cada actor.....	34
Tabla 3.7.2.1. Flujo de eventos del caso de uso administrar datos de usuarios	34
Tabla 3.7.2.2. Flujo de eventos del caso de uso administrar recursos Excel ...	35
Tabla 3.7.2.3. Flujo de eventos del caso de uso transmitir información	36
Tabla 3.7.2.4. Flujo de eventos del caso de uso consultar información actualizada	37
Tabla 3.7.3.1. Flujo de eventos del caso de uso agregar recursos Excel	38
Tabla 3.7.3.2. Flujo de eventos del caso de uso consultar recursos Excel	38
Tabla 4.5.1. Opciones por tipo de usuario.....	53
Tabla 4.5.1.1. Forma Home2.html.....	54
Tabla 4.5.1.2. Forma myMenu.jsp.....	55
Tabla 4.5.2.1. Forma IndexSistema.jsp.....	55
Tabla 4.5.3.1.1. Forma Administradores_ABC2.jsp	55
Tabla 4.5.3.1.2. Forma Alta_Administrador.jsp	56
Tabla 4.5.3.1.3. Forma BorraAdministrador.jsp.....	56
Tabla 4.5.3.1.4. Forma ModifAdministrador.jsp.....	56
Tabla 4.5.3.1.5. Forma archivos.jsp	56
Tabla 4.5.3.1.6. Forma Examinar	57
Tabla 4.5.3.1.7. Forma subeArchivo2.jsp.....	57
Tabla 4.5.3.1.8. Forma verArchivos.jsp.....	57
Tabla 4.5.3.1.9. Forma BorraMaterial.jsp	57
Tabla 4.5.3.2.1. Forma ctotalac_1.jsp	57
Tabla 4.5.3.2.2. Forma ctotalac_4.jsp	58
Tabla 4.5.3.2.3. Forma GuardaDatos2.jsp	58
Tabla 4.5.3.3.1. Forma cdtablas.jsp	58
Tabla 4.5.4.1.1. Forma archivos_usuario.jsp	59
Tabla 4.5.4.1.2. Forma Examinar.....	59
Tabla 4.5.4.1.3. Forma subeArchivo2.jsp.....	59
Tabla 5.3.1 Proceso de carga al servidor de archivos de recursos Excel	70
Tabla 5.4.1 Proceso de actualización de la base de datos de prueba, de las tablas de costos de Inversión operacional, Sísmica 2D y 3D y, Estudios de Campo y Gabinete.....	78
Tabla D.1. Tablas que integran la base de datos de prueba.....	115
Tabla E.1. Requerimientos de software	116
Tabla E.2. Requerimientos de hardware	116

GLOSARIO

Ambiente: Es el espacio en el cual se encuentra situada y desenvuelve una aplicación y está representado, por una serie de elementos como características o recursos de dicho ambiente.

API: (por sus siglas en inglés *Application Programming Interface*). Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos API comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

Aplicación: Software que realiza una función particular para el usuario.

Argumento: Parte de una función que identifica los datos sobre los cuales se debe operar.

Autenticación: Capacidad de probar que una entidad, por ejemplo un usuario o una computadora es quien dice ser.

Base de Datos: Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior.

Bean: Un componente software que tiene la particularidad de ser reutilizable y evita el programar distintos componentes uno a uno, puede representar desde un botón, un grid de resultados, un panel contenedor o un simple campo de texto, hasta otras soluciones más complejas como conexiones a bases de datos, etc.

BOF: (por sus siglas en inglés *Begin Of File*), indica el inicio de los registros de un archivo.

CSS: (por sus siglas en inglés *Cascading Style Sheets*), lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un documento de su *presentación*.

Clase: En programación orientada a objetos, un tipo de datos definido por el usuario que especifica un conjunto de objetos que comparten las mismas características. Un miembro de la clase (objeto) es un "ejemplo" o caso de la clase. Las clases concretas están diseñadas para citar como ejemplos, mientras que las clases abstractas, para pasar las características por herencia.

Cliente/Servidor: Modelo lógico de una forma de proceso cooperativo, independiente de plataformas hardware y sistemas operativos. El concepto se refiere más a una filosofía que a un conjunto determinado de productos. Generalmente, el modelo se refiere a un puesto de trabajo o cliente que accede mediante una combinación de hardware y software a los recursos situados en una computadora denominado servidor.

Código de Acceso: Combinación de letras, números y signos que debe introducirse para obtener acceso a un programa o partes de un programa determinado, una terminal ó computadora personal, un punto en la red, etc.

Código Fuente: Programa en su forma original, tal y como fue escrito por el programador, el código fuente no es ejecutable directamente por el computador, debe convertirse en lenguaje de maquina mediante compiladores, ensambladores o interpretes.

Comando: Instrucción dirigida a una computadora que invoca la ejecución de una secuencia de instrucciones programada previamente.

Componente: Una unidad de composición de aplicaciones que posee un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes, de forma independiente en tiempo y espacio.

Conectividad: Referente al grado en que los dispositivos de hardware, el software, y las bases de datos se pueden relacionar funcionalmente entre sí.

Controlador: Software o pequeños programas necesarios para aprovechar al máximo los dispositivos, accesorios o periféricos distintos que se le pueden conectar a una computadora. Por otro lado también es utilizado para poder lograr conectividad con diversas aplicaciones, principalmente base de datos.

Conexión: Circuito virtual de transporte que se establece entre dos programas de aplicación con fines de comunicación.

CONNECTION: objeto que se utiliza para establecer una conexión o enlace a la base de datos.

CSV (del inglés *comma-separated values*), estos archivos son un tipo de documento para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: cinco, país, plata...) y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles. El formato **CSV** no indica un juego de caracteres concreto, ni cómo van

situados los bytes, ni el formato para el salto de línea. Estos puntos deben indicarse muchas veces al abrir el fichero, por ejemplo, con una hoja de cálculo.

Dato: Se le considera la unidad mínima y básica de toda información; es un elemento susceptible de observación directa ya que no dice algo por si mismo, es decir, los datos no son significativos como tales si no son procesados y convertidos para integrarse a toda la información.

DBMS:(por sus siglas en inglés *Data Base Management Systems*). El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

DCL: (por sus siglas en inglés *Data Control Language*) es proporcionado por el sistema de gestión de base de datos e incluye una serie de comandos SQL que permiten al administrador controlar el acceso a la información contenida en la base de datos. Algunos ejemplos de comandos incluidos son GRANT y REVOKE y las tareas sobre las que se pueden conceder o denegar permisos relacionados pueden ser CONNECT e instrucciones del tipo DML

DDL: (por sus siglas en inglés *Data Definition Language*), instrucciones del tipo DDL son utilizadas para la creación de las estructuras de datos en una base de datos y de todos sus componentes: tablas, índices, relaciones, disparadores (triggers), admite instrucciones para definición como CREATE, DROP y ALTER así como GRANT y REVOKE, los cuales sirven para otorgar permisos o quitarlos, ya sea a usuarios específicos o a un rol creado dentro de la base de datos.

DML: (por sus siglas en inglés *Data Manipulation Language*), las instrucciones DML son aquellas utilizadas básicamente para insertar, borrar, modificar y consultar los datos de una base de datos: SELECT, INSERT, UPDATE, DELETE y USAGE

Dirección IP: (por sus siglas en inglés *Internet Protocol*). Protocolo de comunicaciones para Internet. Representación o notación de la dirección de un equipo conectado a Internet o a una red con el protocolo TCP/IP.

Dirección URL: (por sus siglas en inglés *Uniform Resource Locator*). Abreviatura de localizador uniforme de recursos. Es la dirección que especifica la ubicación electrónica de un sitio (*site*) o un recurso (un archivo) específico de Internet. Una dirección URL consta normalmente de cuatro partes: protocolo, servidor (o dominio), ruta de acceso y recurso.

EOF: (por sus siglas en inglés *End-Of-File*, abreviación de fin de archivo) es un indicador o marca de que no hay más información que recuperar de una fuente de

datos. La fuente de datos puede ser un archivo o un flujo de datos (stream). Conceptualmente en el caso de un archivo indica que se llegó al final del mismo, en caso de un stream es que se finalizó la transmisión o transferencia de datos.

Excel: Microsoft Office Excel, es una aplicación para manejar hojas de cálculos. Este programa fue y sigue siendo desarrollado y distribuido por Microsoft, y es utilizado normalmente en tareas financieras y contables.

Firewall: Equipo de hardware o software utilizado en las redes de computadoras para prevenir algunos tipos de comunicaciones prohibidos por la política de red.

HSLF: (por sus siglas en inglés *Horrible Slide Layout Format*), paquete de clases del Jakarta POI-API-JAVA que proporciona los mecanismos para leer archivos Microsoft PowerPoint.

HSSF: (por sus siglas en inglés *Horrible SpreadSheet Format*), paquete de clases del Jakarta POI-API-JAVA que proporciona los mecanismos para leer archivos Microsoft EXCEL y sus hojas de cálculo de archivos , así como para crear, modificar, leer y escribir archivos con extensión .XLS

HTML: (por sus siglas en inglés *HyperText Markup Language*) es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas WEB, HTML es uno de los formatos más populares que existen para la construcción de este tipo de documentos.

HTTP: (por sus siglas en inglés *HyperText Transfer Protocol*). Protocolo de transferencia de hipertexto. Es el protocolo en que se basa la tecnología de *World Wide Web*, consistente en un conjunto de reglas que gobiernan el software que transporta los documentos HTML a través de Internet.

HWPF: (por sus siglas en inglés *Horrible Word Processor Format*), paquete de clases del Jakarta POI-API-Java que proporciona los mecanismos para leer archivos Microsoft WORD. hojas de cálculo, así como para crear, modificar, leer y escribir archivos con extensión .XLS

Información: Es la emisión o procesamiento de los datos, lo que puede proporcionar un conocimiento o la comprensión de ciertos factores.

Interfaz de usuario: Engloba la forma en la que el operador interactúa con la computadora, los mensajes que éste recibe en pantalla, las respuestas de la computadora a la utilización de periféricos de entrada de datos, etc.

Internet: Es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP. También se usa este

nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que la Internet, independientemente de su extensión o de que sea pública o privada.

Interoperabilidad: Característica de las computadoras que permite su interconexión y funcionamiento conjunto de manera compatible. Esto no siempre es posible, debido a los diferentes sistemas operativos y arquitecturas de cada sistema, pero los esfuerzos de estandarización están permitiendo que cada vez sean más las computadoras capaces de interoperar entre sí.

Intranet: Red privada dentro de una organización. Las *Intranet* suelen utilizar protocolos de Internet para entregar contenido. A menudo se protegen contra al acceso desde Internet mediante servidores de seguridad.

Jakarta POI-API Java: proyecto integrado por varias API para manipulación de varios tipos de archivos basados en el formato de Documento Compuesto OLE 2 de Microsoft manipular, utilizando Java puro. En concreto, se pueden leer y escribir ficheros MS Excel utilizando Java.

Java: Lenguaje de programación desarrollada por Sun Microsystems, en conjunto con varios *frameworks* conforma una plataforma de software. Esta plataforma ha sido desarrollada de tal manera que los programas desarrollados para ella puedan ejecutarse de la misma forma en diferentes tipos de arquitecturas y dispositivos computacionales.

JavaScript: es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas WEB.

JDBC: (por sus siglas en inglés *Java Database Connectivity*). Conectividad Java a Bases de Datos es un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

JSP: (por sus siglas en inglés: *Java Server Pages*), tecnología Java que permite generar contenido dinámico para WEB, en forma de documentos HTML, XML o de otro tipo.

JVM: (por sus siglas en inglés *Java Virtual Machine*), programa nativo, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java;.pieza fundamentales de este último; cuando se escribe una aplicación Java, ésta será ejecutada en una máquina virtual Java en concreto, que es la que en última instancia convierte de código bytecode a código nativo del dispositivo final.

Lenguaje de Programación: Técnica estándar de comunicación para ordenarle instrucciones a la computadora. Un lenguaje le dá la capacidad al programador de especificarle a la computadora, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano.

Metodología: Se entiende que una metodología integra el proceso para desarrollar de manera organizada una aplicación. Su función consiste en identificar las etapas necesarias para la definición de los requerimientos de un proyecto y la consecución de éste; amalgama las acciones y medios que permiten la transformación de los requerimientos del cliente o usuario, a una definición objetiva de los mismos, toda vez que han sido cumplimentados.

Middleware: Término común que se aplica al software que intercambia información en forma transparente entre aplicaciones y bases de datos. El *middleware* es un mecanismo de conexión abstracta entre el software de aplicación y la base de datos y oculta al programador de aplicaciones, los elementos específicos que dependen de la implementación.

Modelo Relacional: Su idea fundamental es el uso de «relaciones». Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados «tuplas»; considera a la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Modelo-Vista-Controlador. Patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones WEB, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Multiplataforma: Término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

MySQL: Gestor de base de datos relacional, sencillo de usar y de gran rapidez; uno de los motores de base de datos más usados en Internet, es del tipo Open Source.

Navegador WEB: Visor o examinador o “browser”. Programa cliente y herramienta básica de navegación para buscar los diferentes recursos de Internet.

Los más usados son Netscape Navigator, Microsoft Internet Explorer, Firefox, Opera, Mozilla y Safari.

OLE DB: (por sus siglas en inglés *Object Linking and Embedding Data Base*). Parte central de la tecnología de conectividad de base de datos de Microsoft.

Página Web: Documento el cual está basado en el lenguaje HTML, la cual permite desplegar información de diversos tipos.

Paquete: Uno o más programas diseñados para realizar una tarea de procesamiento particular.

Plataforma: Se define como el funcionamiento, estructura y diseño de computadoras. Dentro de este término se engloban aspectos como formato de instrucción, modo de direccionamiento y conjunto de instrucciones.

POI: (por sus siglas en inglés Poor Obfuscation Implementation Project). El proyecto POI consiste en varias API para manipular varios formatos de archivos basados en el formato de Documento Compuesto OLE 2 de Microsoft, utilizando Java puro

POIFS: (por sus siglas en inglés Poor Obfuscation Implementation File System) paquete de clases del Jakarta POI-API-JAVA, se utiliza si se tiene un documento escrito en el Formato de Documento Compuesto OLE 2, probablemente escrito utilizando MFC, que se necesite leer en Java. De manera alterna, se utiliza para escribir en el Formato de Documento Compuesto OLE 2 cuando se necesita interoperar con programas ejecutándose en la plataforma Windows.

Protocolo: Sistema de reglas o estándares para comunicarse a través de una red, en especial a través de la Internet. Los equipos y las redes interactúan de acuerdo con los protocolos que determinan el comportamiento, que cada lado espera del otro en la transferencia de información.

Red de Área Local (LAN-Local Area Network) Conexión entre computadoras mediante un medio de transmisión dentro de una distancia inferior a los 10 kilómetros. Son utilizadas principalmente en edificios donde se encuentran oficinas, universidades, hospitales y centros de investigación (entre otros).

Red de Área Ampla (WAN-Wide Area Network): Interconexión de computadoras en áreas geográficas distantes; dicha interconexión necesita contar con mecanismos de acceso especiales, los cuales se enmarcan en lo que se conoce como intercomunicación remota, requiriendo para ello el empleo de microondas, fibra óptica, cable submarino y satélites.

ResultSet: Es la representación en memoria de una de las tablas de la base de datos, se puede entender como una tabla virtual donde todos los procesos que se operan en dicha tabla (insertar registros, eliminar registros, etc) se realizan realmente contra un **RESULTSET** y no provocan ningún cambio en la tabla física hasta que se indique lo contrario.

Rol: Es el papel o acción que debe tomar un componente para alcanzar sus objetivos.

Router (enrutador o encaminador): Es un dispositivo hardware o software de interconexión de redes de computadoras que opera en la capa tres (nivel de red) del modelo OSI. Este dispositivo interconecta segmentos de red o redes enteras. Hace pasar paquetes de datos entre redes tomando como base la información de la capa de red.

Servlet: son objetos que corren dentro del contexto de un contenedor de servlets (ej: TOMCAT, es decir un servidor), su uso más común es generar páginas WEB de forma dinámica a partir de los parámetros de la petición que envíe el navegador WEB.

Servidor de aplicaciones: es un servidor en una red de computadoras que ejecuta ciertas aplicaciones. Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios del uso de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Si bien el término es aplicable a todas las plataformas de software, hoy en día el término *servidor de aplicaciones* se ha convertido en sinónimo de la plataforma J2EE de Sun Microsystems.

Servidor WEB: Programa que implementa el protocolo HTTP. Se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP y de acuerdo a las mismas enviar un resultado.

Sistema Operativo: Conjunto de programas que permiten al usuario, interactuar con un computadora.

Sitio WEB: Colección de páginas *WEB* relacionadas, que residen en el mismo servidor y están interconectadas entre sí mediante vínculos de hipertextos que facilitan el desplazarse ordenadamente a través de las páginas que lo forman.

Sitios Dinámicos: Un sitio *WEB* que se genera de forma dinámica en base a las solicitudes enviadas por el usuario.

Software: Parte lógica de la computadora, esto es, el conjunto de instrucciones (programas) que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es por tanto un campo de estudio de la informática.

SQL: (por sus siglas en inglés *Structured Query Language*). Lenguaje declarativo de acceso a Bases de Datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

Statement: Este objeto y sus dos métodos *executeQuery* (para select de sql) y *executeUpdate* (para insert, update y delete de sql) son los métodos que se utilizan para comunicarse con la tabla física en disco.

Tarea: Una tarea es una acción que puede ser llevada a cabo por uno o más componentes en la búsqueda del logro de los objetivos del sistema.

TOMCAT: (también llamado **Jakarta TOMCAT** o **Apache TOMCAT**), un contenedor de servlets (objetos que corren dentro del contexto de un servidor de aplicaciones y extienden su funcionalidad) desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. TOMCAT implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems.

UML: (por sus siglas en inglés *Unified Modeling Language*). Lenguaje para el modelado, es un lenguaje para la especificación, visualización, construcción y documentación de los componentes de un sistema.

Usuario: Individuo que interactúa con la computadora o bien con la aplicación.

WWW: (por sus siglas en inglés *World Wide Web*). La *Web* o WWW, es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador *Web* para extraer elementos de información (llamados "documentos" o "páginas *Web*") de los servidores *Web* (o "sitios") y presentar los resultados en la pantalla del usuario.

XAMPP: El nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos), **A** *Apache*, **M** *MySQL*, **P** *PHP*, **P** *Perl*; es un servidor independiente de plataforma, software libre, que consiste principalmente de la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas.



CAPÍTULO 1. INTRODUCCIÓN

En este Capítulo se describe la problemática a tratar, se indican el objetivo general y los objetivos específicos relacionados con la misma. Se establece la justificación de este trabajo y por último se define la organización del documento de la tesis.

1.1 Definición del problema

En la actualidad existe una gran necesidad de integración y conjunción de información en las empresas, la emanada de su operación; entre ellas, se encuentran las que orientan sus actividades a tareas relacionadas con la búsqueda y extracción de recursos naturales no renovables, como es el caso del petróleo y sus derivados, es decir, proyectos de exploración de yacimientos petrolíferos. Esto último es una de las actividades de la empresa que se encarga de realizar dichos proyectos en nuestro país y que, para este trabajo, será considerada como caso de estudio.

Se requiere que la información producto de dicha exploración sea capturada en archivos Excel y posteriormente se utilice para actualizar los rubros correspondientes en una base de datos corporativa, la cual es utilizada como parte de los elementos de apoyo relacionados con el proceso de toma de decisiones.

De lo anterior, se deriva que el problema en cuestión es extraer, transformar y cargar información de libros Excel a un ambiente de base de datos.

Lo aquí descrito refiere la problemática de una empresa, cuyo esquema de trabajo está orientado hacia el tratamiento corporativo de datos y su recuperación dinámica, en un entorno donde distintas entidades interactúan como parte de un ambiente integrado de generación de información para el proceso de toma de decisiones.

La solución que se presenta para dicha problemática es una aplicación, acorde a los requerimientos de información empresarial, que permita la transferencia de datos contenidos en formatos Excel a una base de datos corporativa.

Por lo antes mencionado, se entiende que los modelos y métodos existentes deben ser actualizados para satisfacer las demandas actuales. Tecnología reciente sugiere elaborar nuevas estrategias acordes a la solución de las necesidades presentes y futuras.

Para atender estas necesidades se requiere innovar y generar nuevas formas de procesamiento de datos que incluyan un manejo práctico y específico de la

información requerida; la conformada por la combinación Excel-Java-Base de Datos representa una nueva opción.

Por la naturaleza y características de los problemas comentados, se ha encontrado que la tecnología del JAKARTA POI API ha surgido como una forma viable para apoyar la solución de la problemática aquí presentada, donde esta API se vuelve de uso común, día con día, para la solución de problemas específicos de esta índole.

1.2 Objetivos

El desarrollo de este trabajo contempla los siguientes objetivos:

1.2.1 Objetivo General

Presentar una solución basada en el uso del JAKARTA POI API y tecnología Java, que permita agilizar el proceso de lectura, extracción y transformación de información residente en archivos Excel, así como el traslado de la misma y su carga a una base de datos.

1.2.2 Objetivos Específicos

- Diseñar y desarrollar una herramienta de captura (Capturador), de documentos Excel (soportada por VBA y macros en Excel).
- Utilizar el Capturador de documentos Excel para registrar la información correspondiente, contabilizarla y generar los totales de las diferentes categorías.
- Diseñar y desarrollar una aplicación que permita resolver el esquema Excel-Java-Base de Datos.
- Establecer un entorno WEB que permita utilizar en él, las interfaces gráficas y la aplicación desarrollada.
- Utilizar para el desarrollo de la solución en su parte de tratamiento dinámico de datos, herramientas del tipo Open Source (software libre) como el propio Jakarta POI API, la tecnología Java con sus variantes, el servidor WEB TOMCAT y un manejador de base de datos MySQL.

1.3 Justificación

Las empresas están conformadas por entidades activas, las cuales realizan una serie de operaciones, este hecho conlleva la consolidación de estas últimas,

soportado por el registro y manipulación dinámica de datos y, mediante el uso de herramientas adecuadas.

Es por esto que los ambientes de proceso de datos tienden a ser complejos, descentralizados y cambiantes, por lo que se requiere que incorporen flexibilidad y modularidad, mediante el uso de nuevas tecnologías.

La intermitencia en las aplicaciones tradicionales para atender y resolver, de manera ya no se diga integral, sino adecuada, las necesidades en materia de datos e información, está caracterizadas por aspectos como:

- La naturaleza de la empresa y, la agilidad y pertinencia en la toma de decisiones.
- El reto organizacional que propicie una rápida adaptación al cambio.
- Contar con una adecuada infraestructura de proceso de datos y manejo de información.
- Adolecer de una adecuada planeación para la integración de nuevas tecnologías a los diferentes esquemas administrativos y de operación.
- Propiciar la generación repetitiva, en materia de manejo de información, de “nuevos” métodos y procedimientos.
- El temor al uso de tecnologías (aplicaciones y soluciones) basadas en Software Libre.

Lo anterior, puede generar un aletargamiento y peligrosa conformidad en relación a la obtención oportuna de datos e información, de los diferentes entornos que integran la operación de la empresa.

Es en esta etapa donde surge la combinación Excel-Java-Base de Datos, que permite atender este tipo de esquemas a un costo menor y apoyada en el uso de tecnologías de tipo Open Source (software libre).

1.4 Alcances y limitaciones

El trabajo que aquí se presenta, está enfocado a solucionar el problema de extraer, transformar y cargar información de libros Excel a una base de datos MYSQL.

Dicha información, se relaciona con el proceso de toma de decisiones en una empresa que realiza trabajo de exploración para la detección de yacimientos petrolíferos y la cual basa este proceso, en un esquema que, para dicha tarea, considera dos fases:

- La primera a partir del análisis de la información vertida en formatos Excel, resultante de la actividad de exploración.

- La segunda considera la manipulación y recuperación de dichos datos y su traslado e incorporación a una base de datos MySQL, donde una vez actualizada, se permita consultar la información correspondiente.

Por lo anterior se tiene un entorno de solución que contempla:

- Para la captación de la información vertida en formatos Excel, se requiere el diseño y construcción de un esquema de captura, mediante el uso de Visual Basic for Applications y macros Excel.
- Para la manipulación y recuperación dinámica de dicha información, se considera el desarrollo de la aplicación con base en el uso de la tecnología emanada del Jakarta POI API y, de Java propiamente.

La tecnología basada en el API antes citado, provee métodos adecuados para el diseño y desarrollo de aplicaciones orientadas a la recuperación y tratamiento de información residente en formatos tipo Excel y, al analizar sus propiedades (manejo de clases y métodos) y su interacción con el entorno de Java, se establece la viabilidad de su utilización, ya que procura el desarrollo de aplicaciones modulares y reutilizables, es decir, se puede innovar.

En el caso de Excel, su funcionamiento básico es a nivel de libros, hojas y celdas; pero al establecer su relación con el ámbito del POI API, se pueden representar instancias y objetos, lo cual resulta funcional.

Lo aquí mencionado propicia un ámbito de:

- procesamiento masivo
- posibles ambientes descentralizados de suministro de datos y
- una mayor confiabilidad en el control y proceso de la información que se utilice para actualizar la base de datos, al ser resuelto el esquema Excel-JAVA-Base de Datos.

La limitante en el caso del trabajo que aquí se presenta, es que está enfocado exclusivamente, a solucionar el problema de extraer, transformar y cargar información de libros Excel a una base de datos de prueba y que el mismo se ha desarrollado en el entorno de WINDOWS XP, MYSQL y TOMCAT.

1.5 Beneficios esperados

El desarrollo de este trabajo es tendiente a solucionar la problemática antes descrita, al establecer un entorno de aplicación que permita conjuntar el uso de herramientas de captura, manejo y administración de datos para resolver el esquema Excel-Java-Base de Datos.

Propiciar la transmisión de información específica y homogénea y, evitar con ello, la duplicidad de criterios en el manejo de la misma, genera expectativas confiables. Lo anterior es una característica de este desarrollo, soportada por las interfaces gráficas de usuario, como parte de la utilidad de un entorno WEB, diseñadas para permitir el visualizar y revisar, en cada etapa, la información que se está procesando y en su caso, decidir lo conducente, relacionado con cada resultado parcial, antes de finalizar el proceso correspondiente.

La utilización de una solución de bajo costo, como la aquí presentada, que por estar basada en el uso de tecnología de software libre, permite simular diversas situaciones relacionadas con información producto de la exploración, que se traducen en aportes para el proceso de toma de decisiones, sin que necesariamente éstas resulten onerosas.

Al contar con un ambiente de operación donde el proceso principal integra las tareas de recuperación de información de los recursos Excel y su manejo dinámico, así como la actualización de la base de datos de prueba y que dicho proceso sea competencia del servidor, garantiza una adecuada integridad y consistencia de la información, y libera al cliente de la responsabilidad de residir la base de datos en su entorno, lo que propicia la confiabilidad de la administración y resguardo de esta última.

El hecho de utilizar herramientas basadas en arquitecturas ágiles, con adecuados niveles de adaptabilidad y reutilización, como es caso del POI API de Apache Jakarta, permite emplear otras tecnologías actuales como Java, que conjuntamente con la antes citada, procuran el desarrollo de aplicaciones con una mayor flexibilidad y modularidad, así como una mejor integración y funcionalidad con la infraestructura operativa ya existente en la empresa.

Esto último, permite considerar la portabilidad de la aplicación en distintas plataformas (bajo el concepto de Java), lo que finalmente representa un valor agregado en el procesamiento de la información.

También se agiliza el proceso de captura, mediante la elaboración de una aplicación que permite, además de captar la información correspondiente a los rubros inherentes a los estudios de exploración, generar totales de manera automática, lo que significa datos concretos para efecto de toma de decisiones. Dicha captura se basa en un esquema funcional establecido conjuntamente con el usuario, durante el desarrollo e implementación de la misma.

1.6 Organización del Documento

El contenido restante de la tesis está estructurado de la siguiente manera:

En el Capítulo 2 se presenta el marco teórico relacionado con los conceptos generales del manejo de datos mediante archivos de tipo Excel, así como las características principales del POI API y su esquema de manejo de datos; como parte del estado del arte se citan algunas aplicaciones existentes para el manejo de archivos Excel y su ámbito de aplicación y finalmente se presentan algunos casos de éxito basados en el uso del API antes citado.

En el Capítulo 3 se describen los alcances del capturador EXCEL, se identifican y definen los actores participantes en un esquema de generación y proceso de información. Así mismo, se definen los procesos que permitirán la interacción entre las diferentes entidades participantes en el ámbito de estudio y se presenta un esquema de la arquitectura considerada para esta aplicación y basada en un patrón de arquitectura de software denominado Modelo-Vista-Controlador (MVC).

En el Capítulo 4 se consideran las herramientas y criterios utilizados para el desarrollo, así como los módulos e interfaces que integran la aplicación, incluyendo el entorno WEB que soporta la operación.

En el Capítulo 5 y de acuerdo al caso de estudio para el uso de la aplicación, se presentan las pruebas de funcionamiento efectuadas y los resultados obtenidos.

En el Capítulo 6 se realiza el análisis de resultados, se comentan los logros alcanzados, las contribuciones que este trabajo aporta y los trabajos futuros.

Finalmente en el acápite de Apéndices se tienen los siguientes:

- Apéndice A, muestra el código para generar un documento tipo Excel, mediante el uso del Jakarta POI API y sus bibliotecas HSSF y POIFS
- Apéndice B, muestra el Manual de uso del módulo de captura de datos relacionados con proyectos de exploración de yacimientos petrolíferos
- Apéndice C, presenta el Manual de uso de la aplicación para la manipulación y recuperación dinámica de información utilizando el Jakarta POI API y tecnología Java.
- Apéndice D, muestra las Tablas que integran la base de datos de prueba.
- Apéndice E, presenta los requerimientos en materia de software y hardware para el desarrollo y soporte de la aplicación.
- Apéndice F, se tienen procedimientos Java y JSP desarrollados como parte de la solución propuesta.

Resumen

En las organizaciones productivas de bienes y servicios el procesamiento de datos a nivel corporativo es una necesidad que se incrementa de manera gradual, su importancia estriba en el aporte de información global, producto de la operación de cada una de las entidades que las integran, como soporte al proceso de toma de decisiones. El manejo actual de los volúmenes de información no se concibe sin herramientas informáticas que propicien su adecuado tratamiento.

El caso de la problemática que en este capítulo se menciona, considera la integración de diversas tecnologías para ofrecer una nueva opción de solución, identificada ésta como una aplicación para la manipulación y recuperación dinámica de información utilizando el Jakarta POI API. Esta aportación se justifica por el beneficio que ofrece para el manejo de la combinación Excel-Java-MySQL en un ambiente ágil y funcional.

En el Capítulo dos se mencionan algunos conceptos sobre el manejo de documentos Excel, sobre Jakarta POI API y, se presentan algunas soluciones relacionadas que han sido desarrolladas alrededor del mundo, incluyendo algunos casos de éxito al utilizar dicho API.

página en blanco

CAPÍTULO 2. ANTECEDENTES

En este Capítulo se presenta el marco teórico relacionado con los conceptos generales del manejo de datos mediante archivos de tipo Excel y con las características principales del POI API y como parte del estado del arte, se analizan tecnologías existentes y su ámbito de aplicación.

2.1 Marco Teórico

El *marco teórico* o conceptual [29, Scherba] es el grupo central de conceptos y teorías que se utilizan para formular y desarrollar un argumento (o tesis) y se refiere a las ideas básicas que forman la base para los argumentos; en este caso, aplica para los archivos de tipo Excel y para su manejo mediante el Jakarta POI API.

2.1.1 Los Archivos Excel

Los documentos Excel [28, Rentz] se consideran archivos con Formato Microsoft, basados en el formato de Documento Compuesto OLE 2 de Microsoft; son conocidos como archivos de tipo libro que puede incluir múltiples hojas ya sea vacías o con información.

2.1.1.1 Documento tipo LIBRO

Definición. Un documento tipo libro puede tener diferentes hojas y es posible combinar hojas con diferente contenido dentro del libro; por ejemplo, hojas de cálculo con valores y fórmulas, hojas de gráficos con múltiples esquemas gráficos e imágenes y las que se combinan también con macros o módulos de tipo Visual Basic.

Se podría pensar que las hojas de cálculo y de gráficas, son como páginas en un libro Excel; cuando se crea un libro para un proyecto, las hojas que contiene están siempre integradas en él.

Cada libro tiene sus propias características establecidas como Propiedades. La extensión de este tipo de documentos es “.XLS”. Es de hacerse notar que una hoja es parte de un documento tipo HOJA, así como lo es el que este último sea parte de un documento tipo LIBRO, véase la Figura 2.1.1.1.1.

Es posible que un documento tipo LIBRO tenga solo una hoja y aún así será considerado como tal.

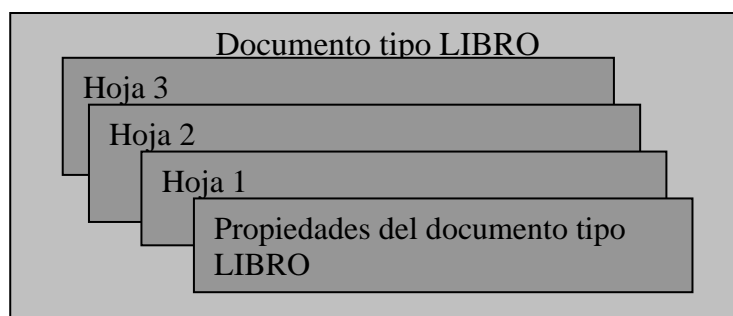


Fig. 2.1. 1.1 .1 Estructura general de un documento tipo LIBRO.

La estructura que contiene información en un documento tipo LIBRO de Excel, está conformada en diferentes registros. Para mantener la consistencia de dicha estructura es de gran relevancia mantener el orden y ubicación de dichos registros.

2.1.1.2 Documento tipo HOJA

Definición. Un documento de este tipo básicamente consiste de renglones y columnas. Pueden existir diferentes tipos de hojas; el caso de una hoja normal puede contener valores y fórmulas, gráficas, imágenes o incluso macros. Por estar contenido en un LIBRO, la extensión para este tipo de documentos también se considera “.XLS”. La Figura 2.1.1.2.1 muestra la estructura general de un documento tipo Hoja.

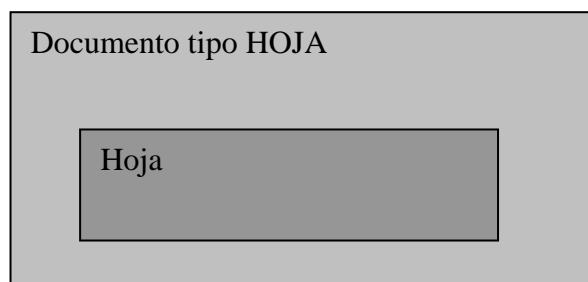


Figura 2.1.1.2.1 Estructura general de un documento tipo HOJA

Los documentos de tipo Hoja también contienen una estructura y sus registros están acotados básicamente por los de tipo BOF (Begin-Of-File) y EOF (End-Of-File). La estructura incluye información acerca de la hoja, las dimensiones, la vista, la lista de fuentes que contiene, lista de nombres y referencias externas, así como el formato de las celdas, el ancho de los renglones y el alto de las columnas, entre otros.

2.1.2 POI API de Apache Jakarta

POI API [27, POI] es un API que permite manipular diversos formatos basados en el formato OLE 2 COMPOUND DOCUMENT de Microsoft, con base en Java puro.

Este API propicia la lectura y escritura de archivos Excel (HSSF); leer y escribir con limitaciones documentos Word (HWPF) y leer y escribir con limitaciones documentos de PowerPoint (HSLF). El grupo Apache Jakarta liberó la versión más reciente (3.0.2) del API. Para efecto de este trabajo, se hará énfasis en la parte de HSSF y la interacción con POIFS, necesario para generar o leer documentos .XLS o .DOC.

El módulo HSSF (por sus siglas en inglés: Horrible Spread Sheet Format) proporciona varios paquetes, véase Tabla 2.1.2.

Tabla 2.1.2 Paquetes que integran el Módulo HSSF	
Paquete	Propósito
Org.apache.poi.hssf.eventmodel	Maneja diferentes eventos generados en los procesos de lectura y escritura de documentos Excel.
Org.apache.poi.hssf.eventusermodel	Proporciona clases para el proceso de lectura de documentos Excel.
Org.apache.poi.hssf.record.formula	Contiene clase para el manejo de FORMULA utilizada en documentos Excel.
Org.apache.poi.hssf.usermodel	Contiene clases para crear documentos Excel.
Org.apache.poi.hssf.util	Es un entorno común de clases para manejar diferentes atributos de documentos Excel.

El uso de estos paquetes permite interactuar con documentos de tipo Excel ya existentes y también crear nuevos. En la Figura 2.1.2 se puede apreciar un extracto del diagrama de clases del módulo HSSF Usermodel [27, POI].

2.1.2.1 El Modelo de Documento Excel

Desde la perspectiva de HSSF (POI), los documentos tipo Excel se dividen en libros, que incluyen una o más hojas y éstas contienen renglones. Los renglones tienen varias celdas en las cuales se consideran diferentes atributos, como sus

dimensiones, bordes y el tipo de información que contienen, entre otros. La Figura 2.1.2.1 muestra la estructura correspondiente

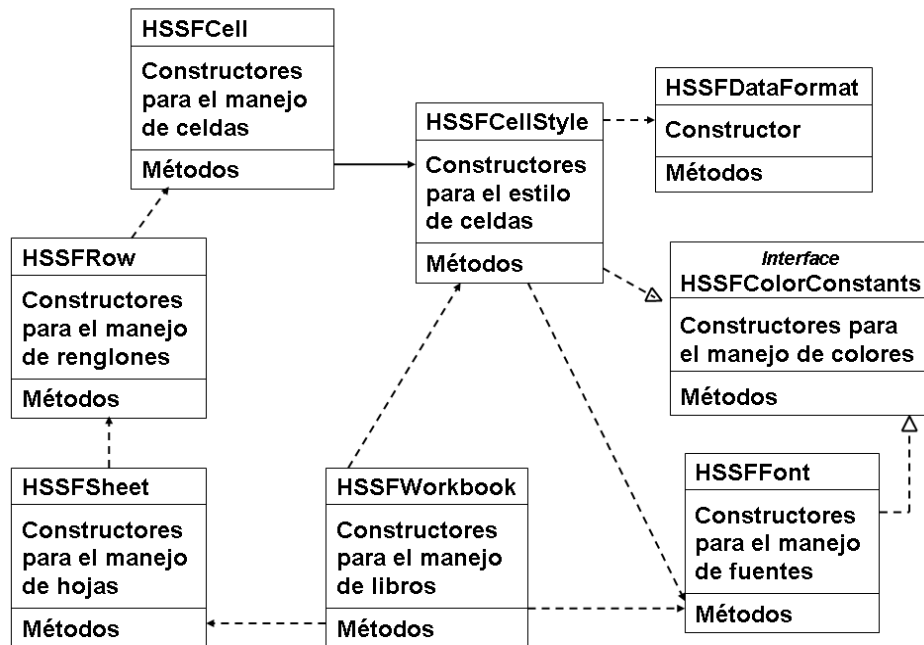


Figura 2.1.2 Resumen del diagrama de clases de HSSF usermodel by Matthew Young-The Apache POI Project

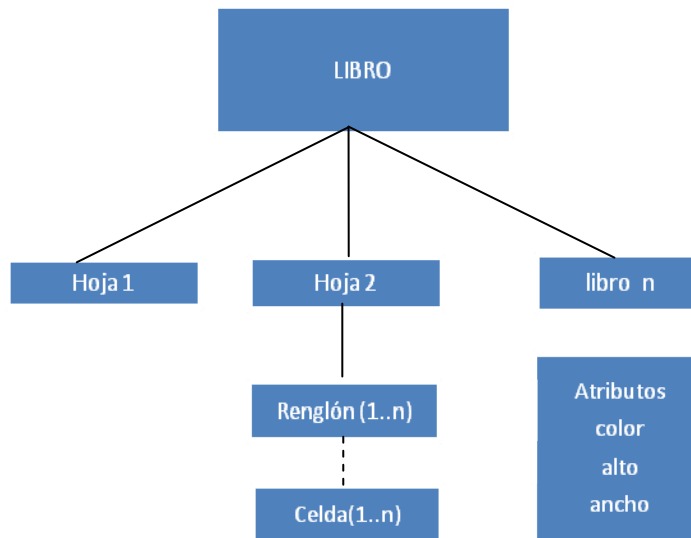


Figura 2.1.2.1 Estructura del modelo del documento Excel

De acuerdo a este modelo [20, Excel], donde se muestra la relación de objetos dentro de un documento tipo Excel, HSSF incluye objetos y clases para cada nivel, es decir, se puede leer un documento tipo Excel como un objeto Libro; por lo anterior, cuando se crea un documento Excel se crean objetos tipo Libro, compuestos a su vez de objetos tipo Hoja y éstos de objetos tipo Renglón y así de forma sucesiva.

Conceptos importantes:

- El libro se representa por la clase **HSSFWorkbook**.
- Cada hoja de cálculo del documento se representa por la clase **HSSFSheet** y se crea a partir de su clase **HSSFWorkbook**, por medio de: **objeto_HSSFWorkbook.createSheet("Nombre de la hoja")**.
- Antes de hacer referencia a una celda se debe referenciar su fila.
- La referencia a una fila se representa por medio de la clase **HSSFRow**
- La fila se obtiene a partir de la hoja (**HSSFSheet**).
- La celda **HSSFCell** se obtiene a partir de la fila (**HSSFRow**)

Generar un libro implica varias acciones, se crean las hojas y también los renglones, las celdas y sus características; todo esto para contener información diversa, como es la del tipo fórmula para un Total o una región para la captura de datos; finalmente, se debe guardar el documento creado.

Como un ejemplo práctico se describe, en el apéndice A, el código para generar un documento tipo Excel, que involucra lo siguiente:

- Utilizar el paquete HSSF del POI API
- Crear el encabezado
- Dejar dos renglones en blanco
- Insertar datos en las dos primeras celdas elegidas de cada renglón y desplegar en el último renglón un Total, de acuerdo a una fórmula que lo calcula.

Dicho código visto de manera sintetizada considera lo siguiente:

- Se crea un objeto **HSSFWorkbook** para el libro.
- Para las hojas se obtiene un objeto **HSSFSheet** desde el objeto anterior. Esta es la forma correcta, no existe un constructor de tipo public para el objeto **HSSFSheet**, lo cual tiene sentido si se considera que no hay un modelo de Excel para hojas que estén fuera del contexto de un libro.

- De la misma forma se obtienen objetos `HSSFRow` (desde `HSSFWorksheet`) para los renglones, así como `HSSFCell` (desde `HSSFRow`) para las celdas.
- Para los encabezados se define un área de renglones y celdas mediante el objeto `org.apache.poi.hssf.util.Region`. Haciendo énfasis en que dicha área se considera desde el valor cero (0).
- Mediante el método `setCellValue` del objeto `HSSFCell` se agrega valor a las celdas. Es importante mencionar que las celdas soportan todos los tipos de datos nativos de Java (`String`, `int`, `boolean`, etc.). El API también considera otros tipos como `Date` y `Calendar`; y los que se capturen mediante este método.
- Como parte del paquete `HSSF`, la clase `HSSFCellStyle` maneja los atributos que integran las celdas (color, fuente, altura, etc.). Para atributos como el color y la fuente, `HSSF` incluye objetos como `HSSFColour` y `HSSFFont`.
- Las celdas también soportan fórmulas, un ejemplo es el mostrado aquí:
`SUM(E6:E9)`
- Finalmente y para crear de manera física el documento Excel, se usa el objeto `OutputStream` y se pasa dicho objeto como parámetro mediante un llamado al método `write` del objeto `HSSFWorkbook`.

El resultado obtenido se puede apreciar en la Figura 2.1.2.2. Libro generado mediante el código antes descrito

2.2 Estado del arte

El estado del arte [30, U. de Antioquía] es una de las primeras etapas a considerar como parte de una investigación, ya que su relevancia consiste en “encontrar el rumbo y seguir la huella” del tema que se pretende investigar, permite determinar cómo ha sido tratado el tema, cómo se encuentra en el momento de realizar la propuesta de investigación y cuáles son las tendencias.

2.2.1 Tecnologías para el manejo de formatos tipo Excel

Los antecedentes relacionados con aplicaciones orientadas al manejo de información residente en archivos Excel, fuera del entorno de Microsoft, permiten establecer la importancia que actualmente se tiene al respecto.

En el siguiente apartado, 2.2.1.1, se mencionan algunas de las aplicaciones Java que existen para leer y escribir archivos de documentos tipo Excel.

The screenshot shows a Microsoft Excel spreadsheet with the following data:

Puntuación del Torneo	
Nombre	Puntos
Oscar	100
Rubén	200
César	350
Pedro	425
Total:	1075

Figura 2.1.2.2 Libro Excel generado mediante el POI API de Jakarta.

2.2.1.1 Paquetes Java para leer y escribir archivos de documentos tipo Excel

Aplicaciones [22, Java Libraries] de las que se tiene referencia hoy en día, se muestran en la Tabla 2.2.1.1.

Tabla 2.2.1.1 Paquetes Java para leer y escribir archivos de documentos tipo Excel		
Aplicación	Licencia	Descripción
Actuate Spreadsheet Engine	Comercial; versión de prueba de 30 días	Escribe archivos Excel (XLS).
ExcelReader	?	Artículo de JavaWorld sobre como leer archivos Microsoft Excel (XLS); incluye código y se requiere tener instalado el driver Microsoft ODBC y para Sun el driver ODBC-JDBC.
ExtenXLS	Comercial; versión de prueba de 30 días	Lee y escribe archivos Microsoft Excel (XLS).
JACOB project	LGPL*	Interfaz Java COM con ejemplos de enlace a Excel.
Java Excel API	LGPL	Lee archivos Excel (XLS).
Java to Excel conversion	?	Escribe archivos tipo SYLK, que Excel puede importar y exportar.
JExcel	Commercial	Componente Swing para integrar Excel vía JNI.
jXLS	LGPL	Crea archivos Excel mediante plantillas XLS.
POI	Apache Software License 1.1	Lee y escribe archivos con formato compuesto de Microsoft OLE 2. Esto incluye archivos MS Office (DOC, XLS, PPT) escritos con versiones de Office que fueron liberadas después de 1997.
Snowbound	Comercial	Lee archivos Excel.
SoftArtisans OfficeWriter	Comercial	Escribe archivos del tipo Word y Excel.
Vista JDBC	Comercial; versión de prueba de 15 días	Driver tipo JDBC, permite acceder a MS Excel (XLS).
xISQL	GPL**	Driver JDBC, permite el acceso a archivos MS Excel (XLS) y a archivos CSV.
* LGPL: Licencia Pública General Reducida de GNU (Lesser General Public License)		
** GPL: The GNU General Public License (GNU GPL o de manera simple GPL)		

A continuación, se hace una breve descripción de los API Java Excel y JExcel, mencionados en la Tabla 2.2.1.1, por considerar que son ejemplos representativos tanto de software libre como de un producto comercial, adicional a la realizada para el POI API.

2.2.1.2 El Api Java Excel

El API Java Excel [18, API Java Excel] es una biblioteca java que proporciona la capacidad, al igual que el POI API, de leer, escribir y modificar documentos tipo Microsoft Excel.

Se integra principalmente de los siguientes paquetes *jxl*; *jxl.demo*; *jxl.format* y *jxl.write*; de acuerdo a la estructura mostrada en la Figura 2.2.1.2

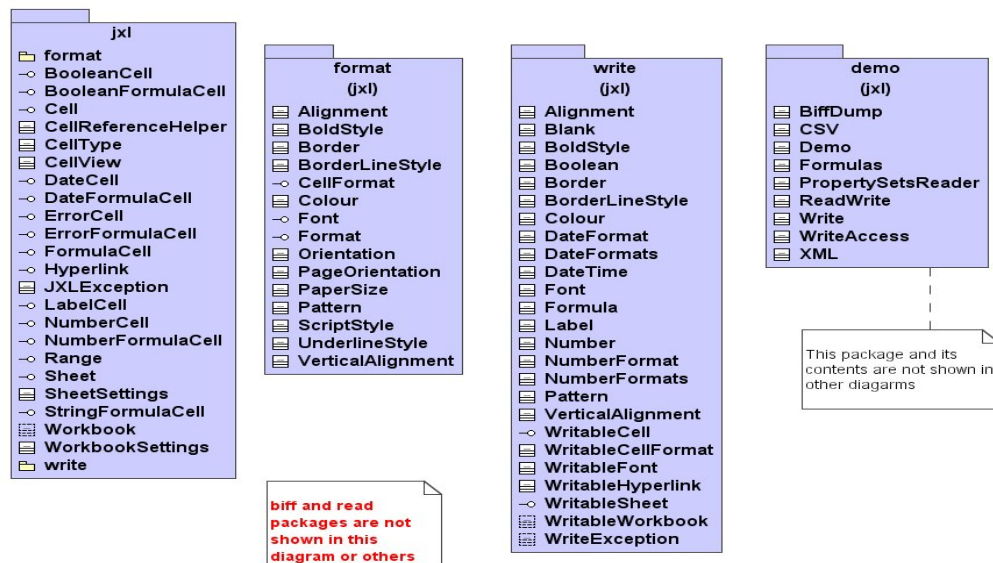


Figura 2.2.1.2. Componentes principales del Api Java Excel.

Un ejemplo de código es el siguiente:

```
// Crea el libro a ser escrito
WritableWorkbook libro = Workbook.createWorkbook(new
File("output.xls"));
// Crea una hoja
WritableSheet hoja = workbook.createSheet("HOJA UNO", 0);
//Agrega una etiqueta a una porción de la hoja
Label etiqueta = new Label(0, 2, "Una etiqueta de registro");
sheet.addCell(etiqueta);
//Agrega cantidades a una celda
Number numeros = new Number(3, 4, 3.1459);
sheet.addCell(numeros);
//Graba en el libro
libro.write();
//Cierra el documento
libro.close();
```

Aquí la diferencia es, por ejemplo, que una celda debe ser instanciada por uno mismo y después añadirla a la hoja y en el POI-API se llama a un método que realiza esto. Tal vez esto, al momento de desarrollar módulos de mayor complejidad, pudiera ser relevante.

El Api Java Excel requiere JDK para operar.

2.2.1.3 El API JExcel

Es una biblioteca [17, API JExcel] que integra documentos Microsoft Excel como parte de aplicaciones tipo Swing y accede a libros y hojas Excel desde programas Java.

Su funcionalidad se divide en:

- Automatizar una aplicación Excel. Esta es la funcionalidad principal y permite trabajar con aplicaciones Excel (libros, hojas, rangos de celdas y celdas).
- Integrar un libro Excel dentro de una aplicación Java tipo SWING. Mediante el componente JWorkbook se pueden integrar libros Excel en una aplicación tipo Java SWING como un componente ordinario de este último tipo.
- Manejo de eventos Excel del tipo libro u hoja. Las clases Workbook y Worksheet de JExcel permiten agregar “listeners” para eventos generados por libros u hojas y manejadores de eventos para controlar el desempeño de la aplicación.
- Trabajar con esquemas propietarios que permiten al usuario agregar funcionalidad basada en los alcances de esta herramienta.
- Permite imprimir documentos Excel.

Una consideración que no puede omitirse es que este producto es comercial y tiene un costo.

Cabe hacer la aclaración, solo para efectos didácticos, de que si no se desea trabajar directamente con la complejidad de estos paquetes, siempre estará la opción de intercambiar datos con archivos Excel, mediante una aplicación Java que lea y escriba archivos que tengan texto plano, con renglones que estén separados por coma (por sus siglas en inglés comma-separated values, csv) o por tab. Claro que este caso sería una aplicación en la que no se necesitarían las ventajas y bondades de los paquetes antes citados y en especial del POI API.

2.2.2 Casos de éxito basados en el uso del POI API de Jakarta

Finalmente, en la Tabla 2.2.2.1 se citan algunos casos exitosos basados en el uso del Jakarta POI API, en la que se indican el nombre de la aplicación, el nombre de la empresa respectiva y una breve descripción de la solución que aporta:

Tabla 2.2.2.1 Casos de éxito basados en el uso del POI API de Jakarta.		
Aplicación	Empresa	Descripción
Optimización de precios.	Sunshine Systems. www.sunshineys.com	Solución de informes basada en POI para un paquete software de optimización de precios que se usa en grandes cadenas de venta.
Informes de datos estadísticos financieros	Banco de Lituania. www.lbank.lt	Generación de informes de datos estadísticos financieros en formato Excel utilizando el API HSSF del proyecto Jakarta POI. El sistema está basado en Oracle JServer y emplea un Java store procedure que utilizando el API HSSF responde en formato XLS.
Sistema de Mantenimiento y Gestión de Instalaciones para la industria de las Telecomunicaciones	Edwards and Kelcey Technology. www.ekcorp.com	Originalmente la factura se hacía con una sencilla hoja CVS que era entonces marcada por cada cuenta y particularizada para cada cliente. Como el crecimiento ha sido consistente con la aplicación, aumentaron las necesidades de facturas que no necesitasen de ser retocadas a mano. POI proporcionó la solución a este problema, integrándose fácil y transparentemente en el sistema. Se utilizó POI HSSF para crear directamente las facturas desde el servidor en formato Excel 97 y ahora sirve más de 150 facturas diferentes cada mes.

Tabla 2.2.2.1 Casos de éxito basados en el uso del POI API de Jakarta.		
Aplicación	Empresa	Descripción
Exportador de datos a formato Excel	ClickFind Inc. www.clickfind.com	A sus clientes de investigación médica, proporcionar capacidad para exportar a Excel datos específicos desde su servicio WEB de recolección de datos electrónicos Data Collector 3.0.
ETL4ALL, (Extract/ Transform/Load)	IKAN Software NV. www.ikan.be	Herramienta de código abierto que virtualmente permite transferencia de datos desde y hacia cualquier origen de datos. Los usuarios pueden combinar y examinar datos almacenados en bases de datos relacionales, bases de datos XML, PDF, ficheros, EDI, CSV, etc.
Sistema de planeación para el fortalecimiento y desarrollo del posgrado en ciencias básicas, desarrollado para CONACYT.	Centro Mexicano de Estudios de Ingeniería para el Desarrollo, A.C. (CEMEID). www.cemeid.org.mx	Apoyar al estudio de las estructuras académicas del posgrado en ciencias básicas, sus avances y resultados, así como su conexión con las ciencias aplicadas, la tecnología y la innovación; parte de la solución se basa en importa archivos Excel.

Resumen

El revisar la información relacionada con el manejo de documentos Excel y con los alcances del Jakarta POI API, ha sido de gran utilidad para establecer la viabilidad del uso de esta API como parte del esquema de solución propuesto.

Como en este capítulo se ha mencionado, existen soluciones diversas para el tratamiento de información contenida en documentos Excel, así como para su traslado a bases de datos; no obstante, otra forma de solución, de igual importancia, es la del desarrollo de aplicaciones acordes a los requerimientos específicos.

Para la problemática considerada en este trabajo de tesis, se ha optado por esta última opción, mediante el desarrollo de una aplicación basada en la

composición del entorno Excel-POI API-Java-MySQL y soportada en un entorno WEB.

En el siguiente Capítulo se realiza el análisis de requerimientos para el desarrollo de la aplicación, en el que se consideran las especificaciones iniciales, así como las interfaces y los perfiles de usuario considerados; y se establecen los componentes que conforman la aplicación y su interacción, mostrado esto mediante un entorno de diagramación UML.

página en blanco

CAPÍTULO 3. ANÁLISIS Y DISEÑO

En el Capítulo anterior se presentó una descripción general de las características del formato tipo Excel así como del POI API y se hizo mención de algunos ambientes basados en otros desarrollos para el manejo de dichos formatos, haciendo énfasis en algunos casos de éxito soportados por el API ya mencionado.

En este capítulo se establecen las especificaciones iniciales de la aplicación, cómo llegan los datos de entrada, los perfiles de usuario y el soporte tecnológico requerido, los objetivos del diseño y se identifican los actores participantes, las diferentes instancias así como la interacción entre ellas, para establecer los requerimientos y arquitectura respectivos, que permitan alcanzar los objetivos de la aplicación.

3.1 Introducción

Es importante indicar que las empresas dedicadas al ámbito petrolero y en especial a la producción de petróleo, como es la de este caso de estudio, consideran, para esto último, las siguientes etapas:

- Exploración
- Perforación
- Extracción
- Refinación
- Distribución
- Venta

Es en dicha secuencia donde la primera etapa se vuelve de la mayor importancia, ya que consiste en buscar dónde hay petróleo en un territorio determinado y, la información obtenida y relacionada, entre otros, con rubros como los de Inversión Operacional, Sísmica 2D y 3D y, Estudios de Campo y de Gabinete, define la viabilidad y horizonte de proyectos de este tipo, en donde los resultados de éxito sugieren un incremento de reservas (es decir, petróleo con posibilidades de ser extraído); y es para esta etapa, que se requiere de una aplicación que permita registrar, actualizar y consultar dicha información.

El realizar de manera organizada una aplicación como la antes mencionada, implica el tener una metodología que integre el proceso del desarrollo.

La elaboración de dicha aplicación [8, Metzger], de manera general, involucra las etapas de análisis de requerimientos, diseño, desarrollo, pruebas y puesta en operación; y adicionalmente integra las tareas de mantenimiento y documentación.

Es en la etapa de análisis de requerimientos donde se establecen las condiciones de funcionalidad que debe satisfacer la aplicación; el diseño procura las especificaciones que satisfagan los requerimientos; el desarrollo considera la programación de lo diseñado y las pruebas tienden a verificar la adecuada funcionalidad de la aplicación. La documentación permite, entre otros, el contar con los manuales de operación respectivos

El Lenguaje Unificado de Modelado, UML [4, Jacobson] incorpora un conjunto de elementos gráficos que se integran para generar diagramas. El propósito de éstos es ofrecer soporte para modelar diferentes vistas de una aplicación. Cada una de ellas refleja un aspecto concreto de la misma. Así, cada diagrama es utilizado de acuerdo a la información que se quiere analizar u obtener. Este capítulo se apoya en algunos de los diagramas antes mencionados para establecer el modelado correspondiente para la elaboración de la aplicación.

3.2 Análisis de requerimientos

La aplicación que este trabajo sustenta, está orientada a satisfacer los requerimientos inherentes al registro de información generada por distintas instancias, producto de las actividades de exploración antes referidas, así como su manipulación y recuperación dinámica; lo anterior en un esquema de operación que permita a los usuarios administrar y consultar dicha información.

Para esto es necesario contar con elementos adicionales que, como parte de la aplicación, permitan controlar el acceso a la misma de acuerdo a un perfil de usuario, registrar y administrar información tanto de los usuarios correspondientes, como de los recursos Excel ha utilizar para el proceso de transmisión de información.

Con referencia en lo hasta aquí mencionado y como parte de las etapas que aborda este trabajo, a continuación se establece el análisis de requerimientos para la funcionalidad de la aplicación.

3.2.1 Especificación inicial.

Las funciones principales que se esperan de la aplicación son:

- Transportar datos entre Excel y MySQL.
- Actualizar información pertinente de los recursos Excel en una base de datos MySQL de prueba.
- Consultar la información actualizada de la base de datos.
- Administrar el acceso a la aplicación de acuerdo al perfil de usuario.
- Administrar el registro de los usuarios de la aplicación.
- Administrar el registro de los recursos Excel.

Por lo anterior, es necesario definir y agregar a la base de datos de prueba, las tablas que permitan incorporar la información de los usuarios y de los recursos Excel; así como de los costos totales relacionados con los rubros de Inversión Operacional, Sísmica 2D y 3D y Estudios de Campo y de Gabinete; se hace mención que como parte del caso de estudio, se cuenta ya con una base de datos de prueba, la cual ha sido modelada y establecida previamente por la empresa de considerada.

De forma adicional, la aplicación debe considerar los siguientes aspectos:

- Establecer el enlace con documentos tipo Excel (capturador), por medio del Jakarta POI API (enlace Excel-Java).
- Vía el enlace antes mencionado, acceder a la información en Excel.
- Establecer la comunicación entre Java y MySQL mediante el enlace JDBC respectivo.
- Interactuar de forma dinámica con MySQL mediante instrucciones SQL.
- Administrar la operación de la aplicación y los recursos inherentes, vía interfaces gráficas en un entorno WEB.

3.2.2 Interfaces gráficas.

Se deben tener interfaces gráficas que procuren:

- Seguridad, los usuarios ingresan a la aplicación mediante un identificador y su respectiva clave de acceso, registrados ambos en la base de datos; esto les dará acceso de acuerdo a su perfil de usuario.
- Administración de usuarios, se debe tener registrada la información correspondiente a los datos del usuario como el nombre, perfil de usuario, identificador (login) y su clave de acceso (password).
- Interacción con el usuario, en cada uno de los módulos que integran la aplicación.
- Mostrar al usuario, respectivamente, los resultados de los procesos de actualización, modificación o eliminación de registros.
- Opciones para el inicio y terminación de sesión.
- Consultas a la información.

3.2.3 Datos de entrada.

3.2.3.1 Recursos Excel

Tener en cuenta que, de origen, los datos de entrada están contenidos en hojas Excel [25, Macros en Excel] y manejan información sobre diversos rubros, entre

los que se consideran gastos y recursos prospectivos, además de costos, precios, premisas y planes de desarrollo. Estos datos son proporcionados por personas especializadas de diferentes instancias de la organización.

3.2.3.2 Usuarios de la aplicación

Para la administración de los usuarios es necesario contar con los siguientes datos:

- No. de registro
- Nombre
- Apellido paterno
- Apellido materno
- Identificador de usuario (login)
- Contraseña (password)
- Tipo de usuario (usuario administrador o usuario básico).

3.2.3.3 Referencia de recursos Excel

Para el registro de la referencia física de los recursos Excel y administración de los mismos, se requiere la siguiente información:

- No. de registro
- Nombre del archivo del recurso Excel

3.2.3.4 Costos totales

Para el registro de los costos totales relacionados con los rubros de de Inversión Operacional, Sísmica 2D y 3D y Estudios de Campo y de Gabinete, se requieren los siguientes datos:

- No. de registro
- Año inicial
- Año final
- Costo

No obstante que en este caso se trata de diferentes tablas, la definición de los datos para cada una de ellas es similar.

3.2.4 Soporte tecnológico requerido.

Con base en un enfoque tecnológico, un desarrollo como el que en este trabajo se presenta considera los siguientes elementos:

1. Contar con un enlace y conexión a una red pública (Internet) o privada.

2. Utilizar un esquema de acceso a través de una aplicación cliente o *Browser*.
3. Contar con una base de datos MySQL de prueba.
4. Incluir los mecanismos computacionales (POI API - JAVA) que permitan el enlace Excel-Java.
5. Incluir los enlaces Java-MySQL, como el Connector J, que permitan el enlace Java-MySQL.
6. Contar con un servidor WEB y un servidor de aplicaciones, que para efectos de este trabajo estarán contenidos en el mismo equipo.

3.3 Perfiles de usuario

De la solución propuesta en el capítulo anterior, se establece la funcionalidad de la aplicación, donde los usuarios tendrán solamente uno de dos perfiles:

- Usuario administrador.
- Usuario básico.

3.3.1 Perfil usuario administrador

El usuario con perfil de administrador tendrá la responsabilidad de los procesos para:

- Administrar
 - recursos Excel.
 - usuarios.
- Actualizar la base de datos.
- Consultar la base de datos.

3.3.2 Perfil usuario básico

El usuario básico accederá a los mecanismos que le permitan:

- cargar los recursos Excel en el servidor así como su referencia en la base de datos y
- consultar el contenido de dichos recursos.

Es necesario mencionar que en cualquiera de los dos perfiles, el usuario en turno no tiene la oportunidad de modificar la información del archivo Excel ya registrado, con lo que se garantiza, para este último, la integridad de los datos.

3.4 Diseño de la aplicación

3.4.1 Objetivos

Como parte de los objetivos del diseño, el desarrollo de este trabajo se fundamenta en la elaboración de una aplicación soportada en esquemas basados en una arquitectura cliente/servidor [9, Orfali], cuya funcionalidad permita alcanzar los siguientes objetivos:

- Habilitar el acceso a la aplicación a través de la WEB.
- Proveer una interfaz programática para el acceso, sin requerir de un cliente específico, lo que permitirá entrar a la misma desde otros entornos.
- Brindar la interfaz de usuario que proporcione acceso a la funcionalidad de la aplicación

Por lo anterior es necesario definir las interfaces gráficas que procuren la operación de la aplicación.

3.4.2 Diseño de interfaces

Se consideran tres interfaces principales:

- Para la presentación y acceso a la aplicación, esto último de acuerdo a los diferentes perfiles de usuario.
- Para el acceso al menú correspondiente al perfil de usuario administrador
 - Dentro de esta instancia, se debe contar con las interfaces para la navegación entre las diferentes opciones y su operación, a saber:
 - Módulo de administración de usuarios.
 - Módulo de administración de recursos Excel.
 - Módulo de actualización.
 - Módulo de consulta.
- Para el acceso al menú correspondiente al perfil de usuario básico.
 - Para este perfil de usuario se tendrán las interfaces que permitan la navegación entre las diferentes opciones y su operación, a saber:
 - Carga en el servidor de archivos de recursos Excel y actualización de su referencia en la base de datos
 - Consulta del contenido de los recursos cargados en el servidor.

Para cada interfaz se debe tener la posibilidad de rechazar o confirmar, según sea el caso, la acción realizada y recibir la notificación respectiva que así se

establezca; así como de poder salir de la sesión correspondiente en el momento que así se considere.

Siempre será posible ver los menús de cada perfil, independientemente de la opción que se esté operando y siempre será visible el nombre de la aplicación en cada página por la que se esté navegando.

3.5 Requerimientos de desarrollo

En esta sección se presenta la descripción de los módulos que requiere la integración de la aplicación para su operación, sin omitir la captura Excel la cual, no obstante ejecutarse fuera del entorno, es parte fundamental del planteamiento de este trabajo.

Con base en lo hasta aquí expuesto, se determina el desarrollo de una aplicación que considere lo siguiente:

- Iniciar y finalizar una sesión de acuerdo al perfil del usuario, si éste no está registrado no tendrá acceso, lo anterior como un aspecto de seguridad.
- Administrar el registro de usuarios (alta, baja, cambio y consulta).
- Administrar el registro de recursos Excel (alta, baja y consulta).
- Administrar el proceso de transmisión de información, el que considera la recuperación y tratamiento de datos contenidos en Excel, su tratamiento y su actualización en la base de datos de prueba.
- Propiciar la consulta de la información actualizada en la base de datos.

Por lo anterior, las opciones inherentes para cada uno de los aspectos referidos son establecidas y apoyadas mediante el correspondiente middleware y con tecnología Java, apoyada en especificaciones como JSP y Servlets [3, Hall]; esto último para el desarrollo del entorno dinámico.

El enfoque está centrado en cumplimentar las tareas mencionadas a través de tecnologías ágiles y confiables y, facilitar al usuario las etapas de proceso, actualización y consulta de información. Todo esto dentro de un ambiente WEB [2, Falkner] que procure la funcionalidad de la aplicación

3.5.1 Módulo de captura Excel

En el apéndice B se identifican los elementos que participan en un esquema de captura en el contexto de la presentación y desarrollo de este trabajo.

En ese mismo se menciona el antecedente sobre el por qué se propone una nueva herramienta de captura, su desarrollo, alcances y su entorno de operación; así como la manera de utilizarla y los resultados obtenidos.

3.5.2 Inicio de sesión

Permite al usuario el inicio de la sesión correspondiente de acuerdo a su identificador y contraseña respectivos.

3.5.3 Administración de usuarios

Este módulo permite verificar que el usuario esté registrado y que la información que se digita para tal efecto, se integre por los siguientes datos: su nombre, apellido paterno, apellido materno, nivel de usuario, identificador y contraseña. Estos dos últimos datos son necesarios para que se pueda acceder a la aplicación, de acuerdo al nivel de usuario.

En este módulo también existen, para el usuario administrador, las opciones de actualizar la información de cada usuario o eliminar el registro respectivo.

3.5.4 Módulo de recursos Excel

Este módulo se encarga principalmente de la administración de los recursos Excel y su referencia, contenida ésta en la tabla “archivos”. En esta parte se tiene un procedimiento para el registro de dichos recursos, los cuales se utilizan para actualizar las tablas de costos totales; cabe hacer notar que no se consideran archivos que no contengan el tipo de extensión correspondiente. El usuario debe utilizar la opción de recursos presentada en el Menú, ubicar el recurso Excel respectivo, seleccionarlo y activar la actualización para que el archivo se guarde en la carpeta del servidor y la referencia del mismo en la base de datos.

La tabla “archivos”, establecida en la base de datos, permite guardar la ruta fija del archivo Excel residente en el servidor; y así poder referenciar a este último por medio del identificador y nombre respectivos.

3.5.5 Módulo de transmisión de información

Este módulo se encarga del proceso que permite acceder a la información contenida en los archivos con extensión .XLS, hacer su tratamiento y actualizar la base de datos. Las tareas cubiertas en esta parte de la aplicación son:

- lectura de archivos Excel.
- procesamiento de información.
- actualización de la Base de Datos.

3.5.5.1 Lectura de archivos Excel

En esta sección el usuario indica el nombre del archivo que contendrá los datos para el proceso de actualización, los cuales una vez leídos son mostrados en

pantalla; para tal efecto basta que elija de la interfaz de selección, la referencia del archivo con la información a recuperar. Se procede a la lectura, extracción y presentación, vía la interfaz WEB, de los datos del archivo elegido. Aquí se utilizan las bondades del POI API.

3.5.5.2. Procesamiento de información

Después de mostrar en la etapa anterior los datos del libro Excel en turno, aquí se accede a la parte del proceso en el que se recopila la información específica, que es mostrada también vía la interfaz WEB y que será enviada al proceso de actualización de la base de datos.

3.5.5.3. Actualización de base de datos

Aquí se hace la conexión y el acceso a la base de datos de prueba mediante un enlace JDBC, en este caso el Connector/J [19, Connector/J] para Java-MYSQL, se reciben los datos para la actualización y por medio de instrucciones de tipo SQL se inserta la información en las tablas de costos.

3.5.6. Consulta de información actualizada

Al igual que en la etapa anterior, se realiza la conexión y el acceso a la base de datos respectivos; y por medio de instrucciones de tipo SQL se consultan las tablas de costos totales con la información ya actualizada, la cual es mostrada mediante la interfaz WEB.

3.5.7. Cierre de sesión

Esta opción permite al usuario el cierre de la sesión respectiva y el regreso a la interfaz de acceso a la aplicación.

Se hace mención de que a excepción del capturador Excel, lo antes descrito está soportado con herramientas tipo Software Libre (Open Source), especialmente en los alcances del Jakarta POI API y la tecnología Java.

3.6 Arquitectura de la aplicación

Al desarrollar aplicaciones con Beans y JSP's, se deben identificar y separar la lógica de presentación y la lógica de negocio, para lo cual se utiliza un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, dicho patrón se denomina arquitectura Modelo-Vista-Controlador (MVC) [21, Design Patterns].

De una manera generalizada, se menciona que los clientes, después del proceso de captura, se pueden comunicar con la aplicación mediante la WEB, donde el servidor [16, Apache Tomcat] considera el entorno requerido y, la lógica de negocios es proporcionada por los componentes de la aplicación. A su vez, éstos últimos realizan la validación del usuario y en su caso, establecen la conexión con el manejador de base de datos (MySQL) para realizar las operaciones respectivas, en un entorno de base de datos.

En la Figura 3.6.1 se aprecia el funcionamiento de la arquitectura considerada para esta aplicación, en la que se observa la secuencia de como un cliente a través de un navegador (browser) accede a la aplicación, hace una invocación a la misma al proporcionar su identificador (login) y su clave de usuario (password), los cuales son validados y registra recursos hacia, o los recupera de la base de datos.

Se hace mención de que para efectos de este trabajo, la aplicación se encuentra en el mismo servidor que la base de datos, por lo que la definición de la conexión hacia esta última es fija.

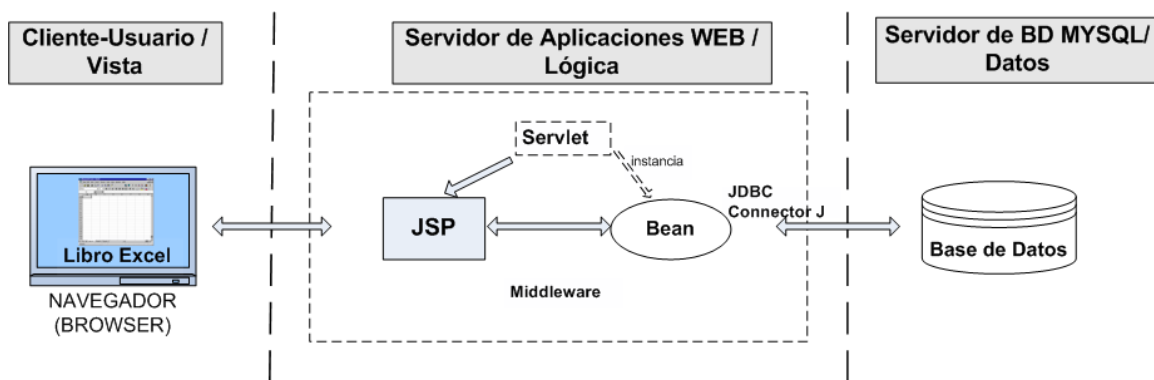


Figura 3.6.1. Arquitectura de la aplicación

Cabe recordar que una base de datos integra objetos (tablas, entre otros) que deben ser adaptados para registrar, actualizar y entregar información; y aquí se reflejan los alcances de esta arquitectura.

3.7 Casos de uso

Después de definir la arquitectura de la aplicación y tomando en cuenta los requerimientos establecidos, a continuación se muestran los diagramas UML [11, Schmuller] que se desarrollaron en soporte a la fase del análisis.

3.7.1 Actores y sus casos de uso

Para operar la aplicación se definen dos actores: usuario administrador y usuario básico, cuyos perfiles de usuario son:

- **Usuario Administrador:** persona que se encarga de la operación de la aplicación, teniendo como tareas la administración de usuarios, de recursos Excel y del registro de usuarios (incluidos los niveles de acceso), así como el proceso de transmisión de información y la consulta de la misma en la base de datos.
- **Usuario básico:** persona que opera, de manera limitada, el módulo de recursos Excel, al registrar y consultar los recursos correspondientes, sin poder darlos de baja.

La Figura 3.7.1.1 muestra el diagrama de contexto de la aplicación para la manipulación y recuperación dinámica de información (MRDI).

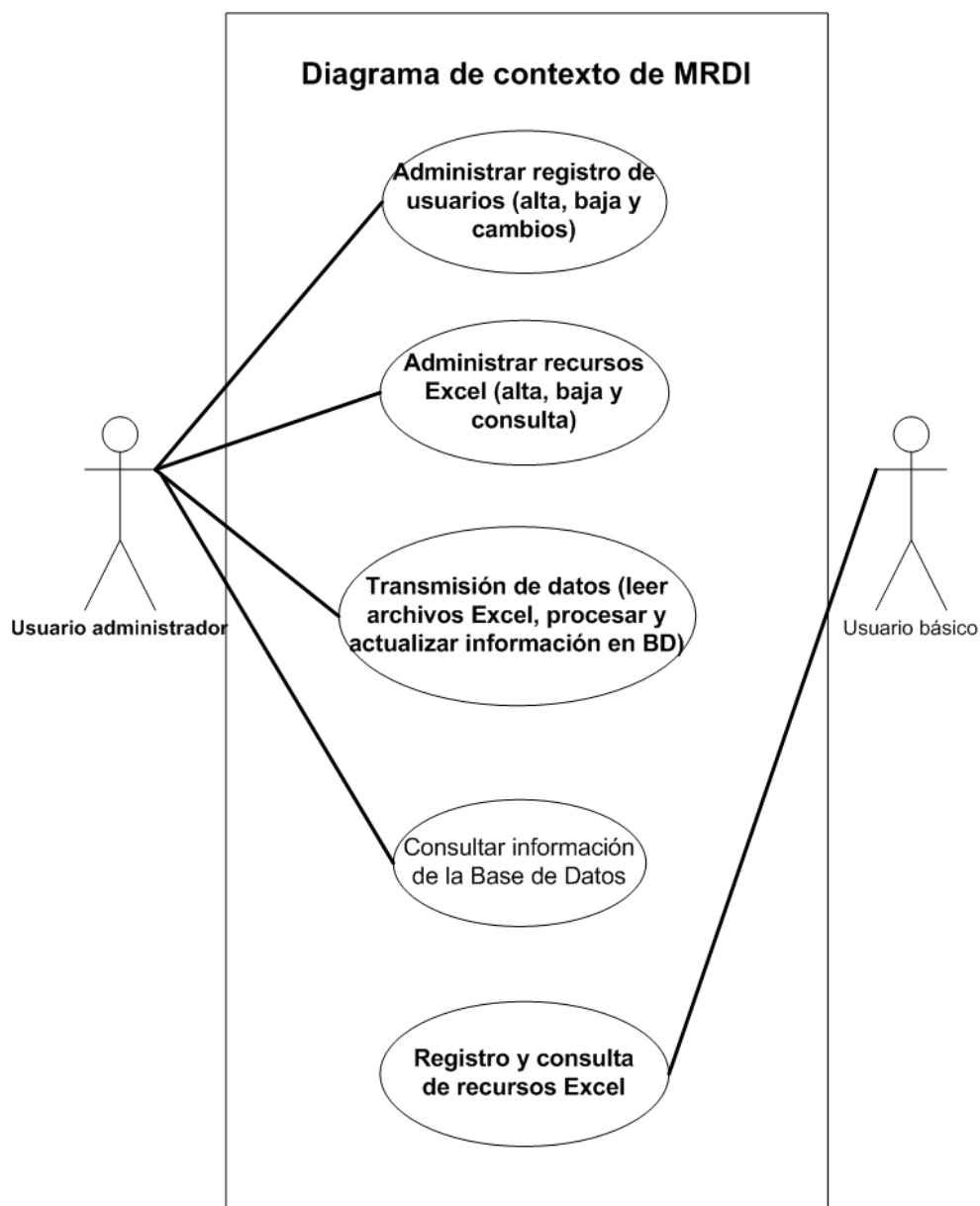


Figura 3.7.1.1. Diagrama de contexto de la aplicación para la manipulación y recuperación dinámica de información

En la Tabla 3.7.1 se muestran los casos de uso considerados para cada actor.

Tabla 3.7.1 Casos de uso de cada actor	
Actor	Casos de uso
Usuario Administrador	<ul style="list-style-type: none"> ▪ Administrar datos de usuarios ▪ Administrar recursos Excel ▪ Transmitir información ▪ Consultar información actualizada.
Usuario básico	<ul style="list-style-type: none"> ▪ Dar de alta recursos Excel ▪ Consultar recursos Excel ▪ Consultar base de datos actualizada

3.7.2 Escenarios del actor “Usuario administrador”

3.7.2.1 Administrar datos de usuarios

Descripción: Este caso de uso es empleado para detallar el registro y actualización de usuarios de la aplicación (altas, bajas y cambios), véase Tabla 3.7.2.1.

Tabla 3.7.2.1 Flujo de eventos del caso de uso Administrar datos de usuarios	
Acción del Actor	Respuesta de la Aplicación
1. Se identifica como usuario administrador.	2. Presenta el ambiente correspondiente de opciones.
3. Elige módulo de usuarios.	4. Muestra el entorno de usuarios.
5. Elige la opción de alta y proporciona la información del usuario a ser dado de alta.	6. Recibe y guarda la información del usuario. Realiza el registro.
7. Elige opción de cambios para el registro del usuario.	8. Muestra el entorno de cambios para el registro del usuario.
9. Realiza actualización de datos en el registro del usuario	10. Realiza la actualización e informa de los resultados.

Tabla 3.7.2.1 Flujo de eventos del caso de uso Administrar datos de usuarios	
11. Elige opción de baja del registro de usuario.	12. Muestra el entorno de bajas para el registro del usuario.
13. Realiza la baja del usuario.	14. Realiza la baja e informa de los resultados.

3.7.2.2 Administrar recursos Excel

Descripción: Este caso de uso se utiliza para detallar el registro, consulta y baja de los recursos Excel, véase Tabla 3.7.2.2.

Tabla 3.7.2.2 Flujo de eventos del caso de uso Administrar recursos Excel	
Acción del Actor	Respuesta de la Aplicación
1. Se identifica como usuario administrador.	2. Presenta el ambiente correspondiente de opciones.
3. Elige opción de alta de recursos Excel.	4. Muestra el entorno de alta de recursos Excel.
5. Busca y elige el archivo a ser registrado y lo envía para su ingreso.	6. Recibe y guarda la información del recurso. Realiza el registro.
7. Elige opción de consulta de recursos.	8. Muestra el recurso Excel elegido.
9. Consulta el recurso Excel.	10. Presenta el ambiente correspondiente de opciones.
10. Elige la opción de baja de recursos.	11. Muestra el entorno de baja de recurso.
12. Elige el recurso a dar de baja.	13. Realiza la baja del recurso.

En la Figura 3.7.2.2, se aprecia el diagrama de casos de uso correspondiente a las opciones de administrar usuarios y administrar recursos Excel.

3.7.2.3 Transmitir información

Descripción: Este caso de uso se utiliza para detallar la transmisión de información (recuperación de recursos Excel, procesamiento de la información y su actualización en la base de datos).

En la Tabla 3.7.2.3, se aprecia el flujo de eventos del caso de uso Transmitir información.

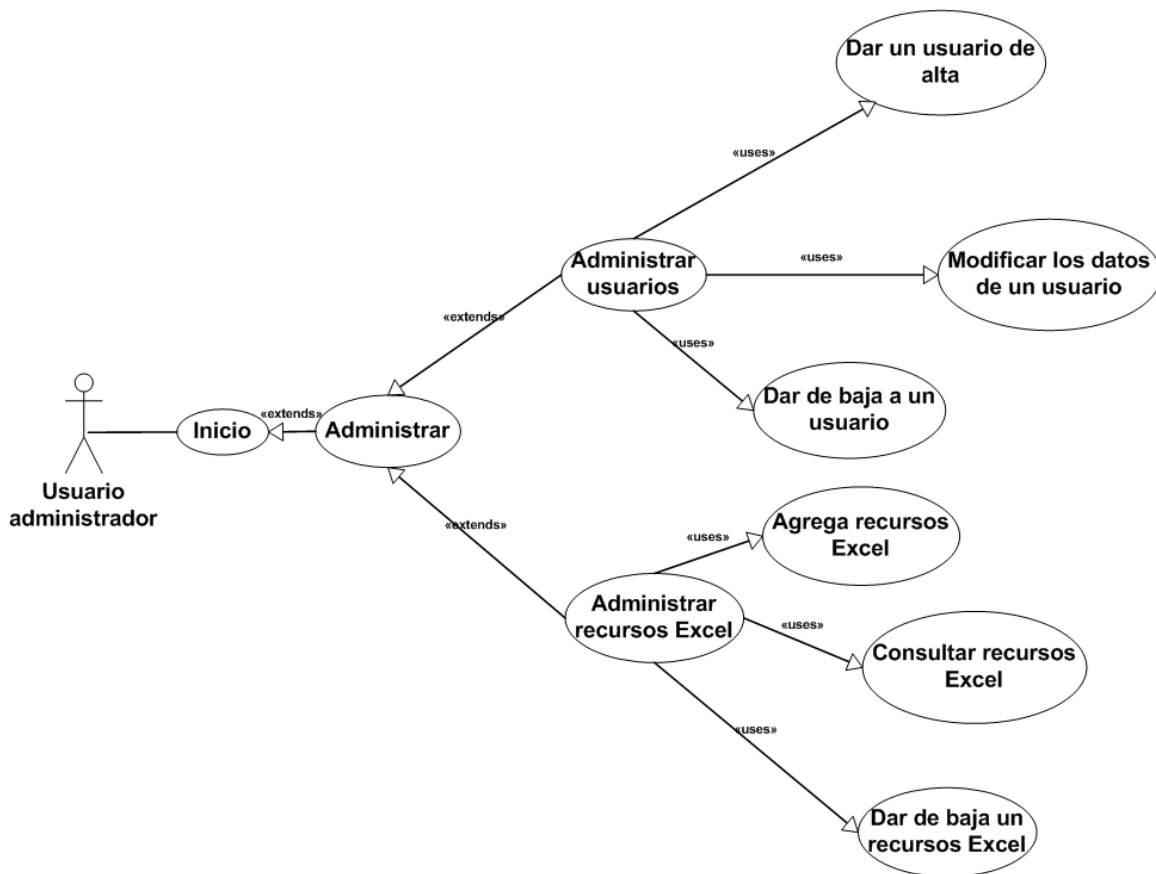


Figura 3.7.2.2. Diagrama de casos de uso para la administración de usuarios y de recursos Excel.

Tabla 3.7.2.3 Flujo de eventos del caso de uso Transmitir información	
Acción del Actor	Respuesta de la Aplicación
1. Se identifica como usuario administrador.	2. Presenta el ambiente correspondiente de opciones.
3. Elige opción de actualización	4. Muestra lista de archivos de recursos Excel ya registrados.
5. Selecciona recurso Excel.	6. Lee la información ha ser procesada y muestra datos del libro.
7. Visualiza información y elige la	8. Procesa información, actualiza la base

Tabla 3.7.2.3 Flujo de eventos del caso de uso Transmitir información	
Acción del Actor	Respuesta de la Aplicación
opción de procesar.	de datos y emite aviso del resultado de la actualización.
9. Visualiza resultados de la actualización de la base de datos.	10. Presenta el ambiente correspondiente de opciones.

La Figura 3.7.2.3 muestra el diagrama de casos de uso de la transmisión de información (recuperación de recursos Excel, su proceso y actualización en la base de datos).

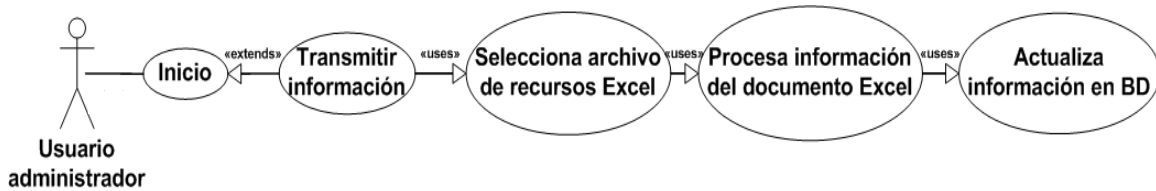


Figura 3.7.2.3. Diagrama de casos de uso para la transmisión de información.

3.7.2.4 Consultar información actualizada

Descripción: Este caso de uso se utiliza para detallar la consulta de información actualizada en la base de datos, el cual se describe en la Tabla 3.7.2.4.

Tabla 3.7.2.4 Flujo de eventos del caso de uso Consultar información actualizada	
Acción del Actor	Respuesta de la Aplicación
1. Se identifica como usuario administrador.	2. Presenta el ambiente correspondiente de opciones.
3. Selecciona módulo de consulta y efectúa la consulta.	4. Muestra información de la base de datos ya actualizada.
5. Visualiza resultados de la consulta.	6. Presenta opciones de la aplicación.

La Figura 3.7.2.4 muestra el diagrama de casos de uso para la consulta de la base de datos.



Figura 3.7.2.4. Diagrama de casos de uso para la consulta de la base de datos.

3.7.3 Escenarios del actor “Usuario básico”

3.7.3.1 Agregar recursos Excel

Descripción: Este caso de uso se utiliza para detallar el registro y consulta de recursos Excel, véase Tabla 3.7.3.1.

Tabla 3.7.3.1 Flujo de eventos del caso de uso Agregar recursos Excel	
Acción del Actor	Respuesta de la Aplicación
1. Se identifica como usuario básico.	2. Presenta el ambiente correspondiente de opciones.
3. Elige módulo de recursos Excel.	4. Muestra el entorno de alta de recursos Excel.
5. Busca y elige el archivo a ser registrado y lo envía para su ingreso.	6. Recibe y guarda la información del recurso. Realiza el registro.
7. Elige opción de consulta de recursos Excel.	8. Muestra lista de recurso Excel existentes.
9. Elige y consulta el recurso Excel.	10. Muestra la información del archivo elegido.

3.7.3.2. Consultar recursos Excel

Descripción: Este caso de uso se utiliza para detallar la consulta de información actualizada en la base de datos, véase Tabla 3.7.3.2.

Tabla 3.7.3.2 Flujo de eventos del caso de uso Consultar recursos Excel	
Acción del Actor	Respuesta de la Aplicación

1. Se identifica como usuario básico.	2. Presenta el ambiente correspondiente de opciones.
3. Elige módulo de recursos Excel.	4. Muestra lista de los archivos existentes ya cargados en la base de datos.
5. Selecciona recurso Excel de la lista.	6. Muestra la información correspondiente
7. Visualiza resultados de la consulta.	8. Presenta opciones de la aplicación.

La Figura 3.7.3.2 muestra el diagrama de casos de uso del usuario básico para las opciones de agregar y consultar recursos Excel.

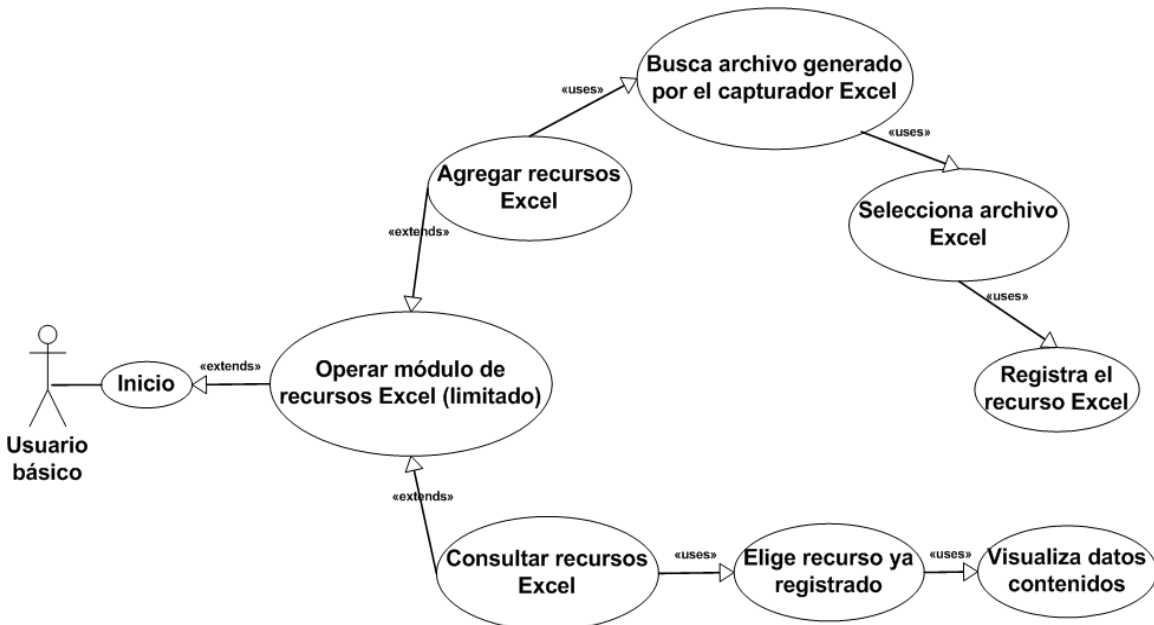


Figura 3.7.3.2 Diagrama de casos de uso del usuario básico para las opciones de agregar y consultar recursos Excel

3.7.4 Diagrama de secuencia del acceso a la aplicación

Para operar la aplicación de acuerdo a las opciones ya mencionadas, es preciso ubicar primero al tipo de usuario, con base en su identificador y clave de acceso, es por esto que en el siguiente diagrama de secuencia del acceso a la aplicación, se muestra la trayectoria ordenada de eventos que surgen entre la interacción del actor y la aplicación. Véase Figura 3.7.4.1.

De forma concisa, la descripción del diagrama considera las siguientes etapas:

1. Se presenta la interfaz principal al usuario.
2. El usuario proporciona su identificador y clave de acceso.
3. Se reciben los datos enviados (identificador y la clave correspondientes) y se turnan para su validación
4. Se valida que los datos estén registrados y se regresa el tipo de usuario
5. Recibe la información verificada y turna al ambiente correspondiente, según el tipo de usuario.

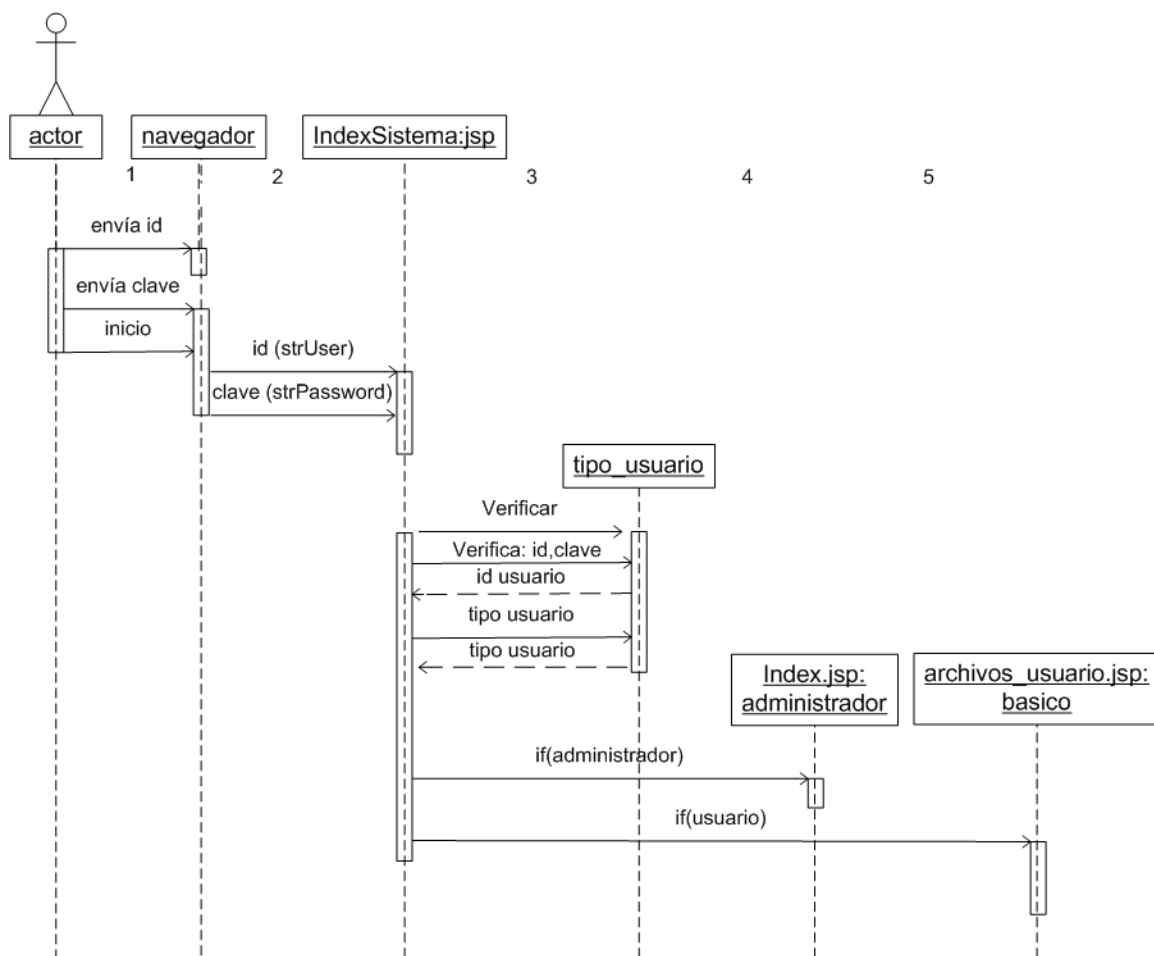


Figura 3.7.4.1. Diagrama de secuencia del acceso a la aplicación de acuerdo al tipo de usuario.

3.8 Diagrama relacional

Para el caso de la administración de usuarios y de recursos Excel, se definieron dos tablas: usuario y archivos, las cuales incluyen las estructuras correspondientes.

Se hace notar que se ha designado un número de proyecto ficticio, que en este caso se asigna directamente al insertar los datos de actualización para cada una de las tablas. En la práctica, el esquema de operación considera el dato previamente actualizado por el usuario en la tabla Simulable.

En el apéndice D se muestran las tablas que conforman la estructura original de la base de datos de prueba.

3.9 Paquetes de clases que integran la aplicación

Los paquetes que forman parte de la aplicación para la manipulación y recuperación dinámica de información (MRDI), están conformados por las diferentes clases [1, Deitel] que otorgan la funcionalidad necesaria a la aplicación.

El diagrama correspondiente se muestra en la Figura 3.9.

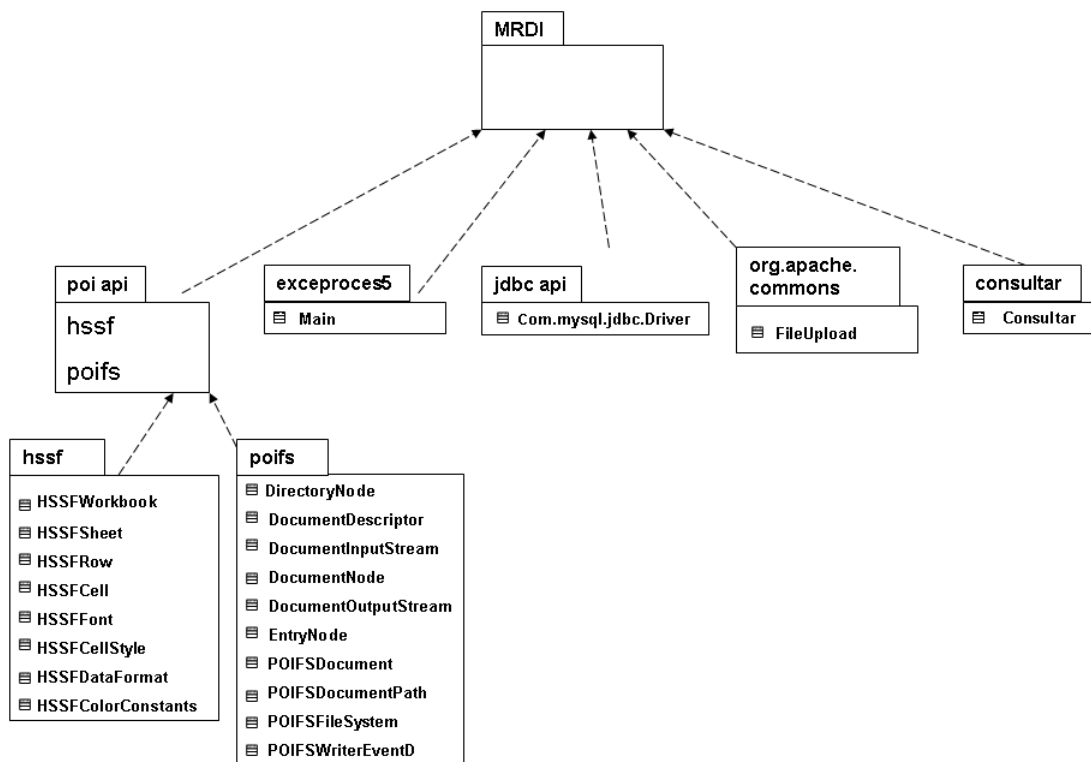


Figura 3.9 Diagrama de paquetes de clases que integran la aplicación

Descripción:

- **mrldi:** aplicación que contiene los paquetes que la integran y, sus respectivas clases.
- **poi-api:** contiene las clases que hacen posible el manejo de los libros, hojas, files y celdas de un documento Excel.
- **poifs:** contiene las clases, que permiten acceder a los documentos Excel, donde la principal clase para este trabajo es POIFSFileSystem.
- **hssf:** contiene las clases que permiten acceder a la información del documento Excel a nivel de libro, hoja, fila o celda.
- **exceproces5:** contiene la clase mediante la cual es posible el tratamiento de la información contenida en los recursos Excel.
- **jdbc api:** el api JDBC consiste de un conjunto de clases e interfaces que permiten a cualquier programa Java acceder a sistemas de bases de datos, de forma homogénea. En el caso del manejador de bases de datos MySQL, el Connector/J es el driver JDBC oficial que contiene la clase que permite el enlace entre Java y la Base de datos MySQL.
- **org.apache.commons:** contiene la clase que permite hacer la carga de los recursos Excel al servidor.
- **consultar:** contiene la clase que propicia la consulta a la información de las tablas actualizadas

3.10 Diagramas de clases

A continuación se describen en forma general las clases que se generan dentro de la aplicación, en este caso la clase Main, contenida en el paquete exceproces5; de la cual se genera el diagrama correspondiente y se hace también una breve descripción de la clase FileUpload y su utilidad; las clases que son parte de paquetes establecidos pueden ser consultadas en los enlaces respectivos.

3.10.1 Descripción de la clase

3.10.1.1 La clase Main

Esta clase se desarrolló con la idea de apoyar el procesamiento de información relacionada con los eventos de:

- Acceder a los libros Excel
- Leer la información contenida en sus hojas
- Dar un tratamiento específico a dicha información para un posterior proceso, como el de actualizar con ella una base de datos.

Así mismo para permitir:

- Realizar la conexión a una base de datos MySQL
- Finalizar dicha conexión y

Consultar la información correspondiente a la tabla relacionada con la referencia de los archivos Excel cargados al servidor.

Dicha clase incluye los métodos:

- public static double [] suma(String archivo)
- public void connect()
- public void disconnect()
- public ResultSet VerArchivo()

En la Figura 3.10.1.1 se muestra el diagrama de clases para la clase Main.

Nombre de la clase	Main
Atributos	<ul style="list-style-type: none"> ◆ String msg; ◆ String dia = ""; ◆ String mes = ""; ◆ String annio = ""; ◆ String fecha = ""; ◆ Connection con; ◆ Statement stmt; ◆ ResultSet rs = null; ◆ String error; ◆ double costos2dy3d = 0; ◆ double costocyg = 0; ◆ double costotiop = 0; ◆ String tabla1 = "totalcostosdeinversiondeestudioscyg"; ◆ String tabla2 = "totalcostosdeinversiondesismica2dy3d"; ◆ String tabla3 = "totalcostosdeinversionoperacional";
Métodos	<ul style="list-style-type: none"> ◆ public static double [] suma(String archivo) ◆ public void connect() ◆ public void disconnect() ◆ public ResultSet VerArchivo()

Figura 3.10.1.1 Diagrama de clases de la clase Main.

3.10.1.2 La clase Consultar

Esta clase propicia la consulta de la información de las tablas de costos que se actualizan como parte del proceso Excel-Java-MySQL.

Así mismo permite conectarse con la base de datos de prueba, así como finalizar dicha conexión.

La clase incluye los métodos:

- public void connect()
- public void disconnect()
- public ResultSet VerDatos()
- public ResultSet VerDato()
- public ResultSet VerData()

En la Figura 3.10.1.2 se muestra el diagrama de clases para la clase Consultar.

Nombre de la clase	Consultar
Atributos	<ul style="list-style-type: none"> ◆ String msg = ""; ◆ Connection con = null;
Métodos	<ul style="list-style-type: none"> ◆ public ResultSet VerDatos(String clave) ◆ public ResultSet VerDato(String clave) ◆ public ResultSet VerData(String clave) ◆ public void disconnect() ◆ public ResultSet VerArchivo()

Figura 3.10.1.2 Diagrama de clases de la clase Consultar.

3.10.1.3 Diagrama de clases de la aplicación

El diagrama de clases de una aplicación permite organizar las clases y su clasificación, así como los métodos que se utilizan en cada uno de ellos.

En la figura 3.10.1.3 se muestra el diagrama de clases de las clases que forman parte de la aplicación.

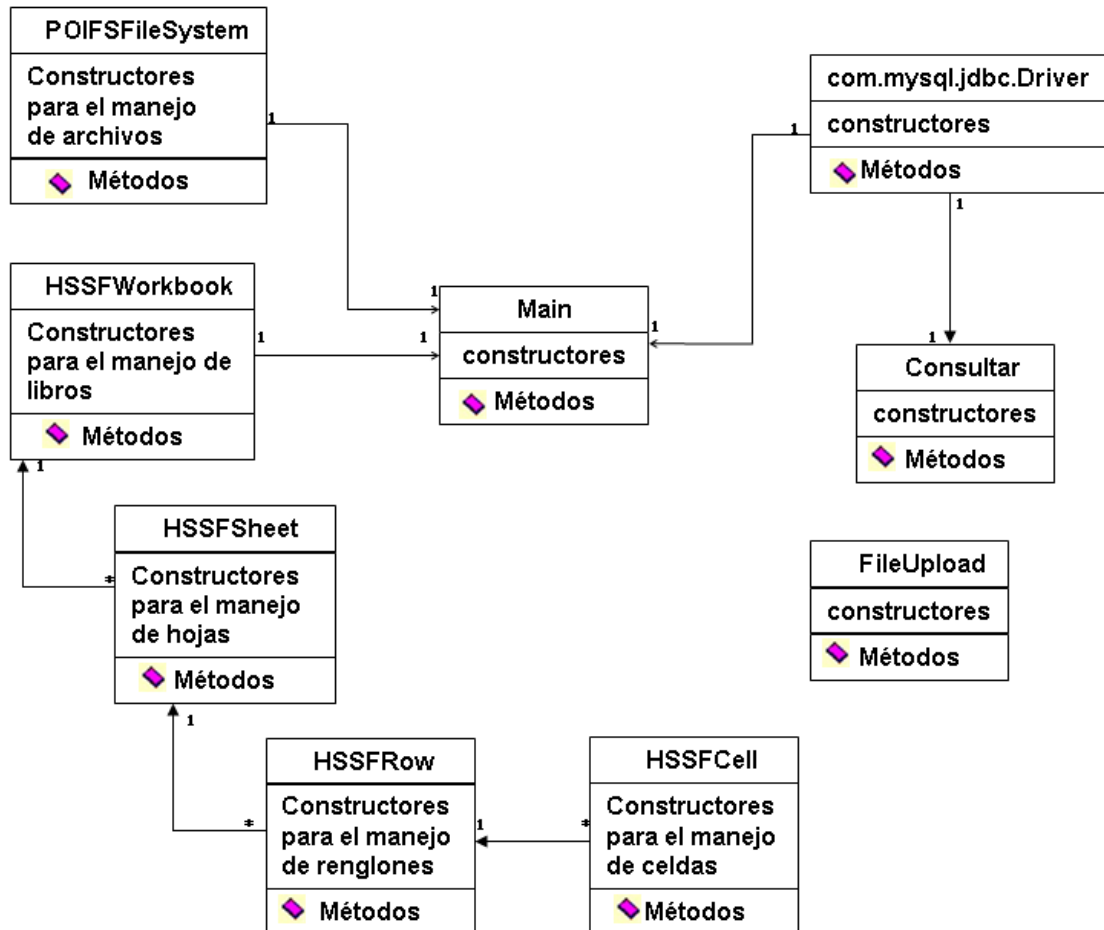


Figura 3.10.1.3 Diagrama de clases de las clases que forman parte de la aplicación.

Descripción de clases:

- Main:** Invoca a la clase POIFSFileSystem y a su método que permite el enlace con el documento Excel y junto con los métodos de la clase HSSF accede a la información de dicho documento; lo anterior en el contexto del método SUMA, mediante el que se realiza el tratamiento de la información contenida en las hojas, renglones y celdas y deja un vector con la información requerida; con el método CONNECT se establece el enlace entre el entorno JAVA y la base de datos MySQL y, el método

DISCONNECT finaliza dicho enlace; así mismo, mediante el método VerArchivo se consulta a la tabla “archivos” contenida en la base de datos de prueba, para verificar la referencia de los archivos Excel cargados en el servidor.

- **POIFSFileSystem:** contiene entre otros, el método POIFSFileSystem que permite establecer el enlace con el archivo del documento Excel.
- **HSSFWorkbook:** permite acceder al entorno del libro Excel.
- **HSSFSheet:** permite acceder al entorno del documento hoja, del libro Excel.
- **HSSFRow:** proporciona las opciones de manejo de información a nivel de renglones de la hoja Excel.
- **HSSFCell:** permite el acceso a la información de los renglones de la hoja Excel a nivel de celda.
- **Com.mysql.jdbc.Driver:** se encarga de establecer el enlace entre el entorno Java y la base de datos MySQL.
- **Consultar:** contiene los métodos que permiten acceder a la consulta de las tablas respectivas en la base de datos.
- **FileUpload:** como se detalle a continuación, permite cargar o residir información del cliente en el servidor, en este caso los archivos Excel.

3.10.1.4 Clase FileUpload

Esta clase forma parte del paquete Org.Apache.Commons [31, Upload] y se menciona como parte de este trabajo al permitir seleccionar los archivos que se cargan al servidor, en este caso los archivos Excel.

La clase **FileUpload** mediante un control de cuadro de texto y un botón de búsqueda, permite al usuario seleccionar un archivo en el cliente y cargarlo en el servidor Web. El usuario elige el archivo que desea cargar y señala su ruta de acceso completa en el equipo local en el cuadro de texto del control, por ejemplo:

```
("C:\\Archivos de programa\\Apache Software
Foundation\\Tomcat5.5\\webapps\\refinacion\\archivosExcel\\"+itemName);
```

También puede seleccionar el archivo haciendo clic en el botón **Examinar** y seleccionarlo después en el cuadro de diálogo **Elegir archivo**.

Una vez que el usuario ha seleccionado el archivo para cargarlo, el control **FileUpload** no lo guarda automáticamente en el servidor, es necesario establecer y proporcionar explícitamente un control o un mecanismo que permita al usuario enviar el archivo especificado. Es decir, que proporcione, por ejemplo, un botón donde el usuario pueda hacer clic para cargar el archivo. El código que se escriba para guardar el archivo especificado tiene que llamar al método `SaveAs`, que es el que guarda el contenido de un archivo en una ruta especificada del servidor.

Resumen

En este capítulo se han establecido los requerimientos y su análisis para el desarrollo de la aplicación al definir las especificaciones iniciales, los perfiles de usuario soportados y las interfaces consideradas para la operación, así mismo los componentes que la integran y la interacción entre ellos.

Se han definido los casos de uso que muestran los alcances del entorno, así como los paquetes y sus clases que propician la generación del código Java respectivo.

En el siguiente Capítulo se describe de forma detallada la aplicación para la manipulación y recuperación dinámica de información, en la que se establece su funcionalidad así como los módulos que la integran y sus alcances.

CAPÍTULO 4. DESARROLLO DE LA APLICACIÓN

En consecución de las etapas para el desarrollo de una aplicación como la que aquí se presenta, en este Capítulo se describe detalladamente el entorno de la aplicación, así como la infraestructura que permite su desarrollo.

4.1. Introducción

La función principal de la aplicación que aquí se presenta, sobre la cual son vertidas las ideas formuladas en este trabajo, es la de lograr la interacción de la plataforma POI API con Excel y, conjuntamente con la combinación Java y MySQL [26, Manual MySQL], apoyar la obtención de información oportuna y confiable, orientada a la toma de decisiones.

En primera instancia se dará una breve descripción del ambiente de operación, desde el punto de vista de funcionamiento, con el fin de comentar cómo se integra.

Se describirán algunos de los aspectos importantes de la plataforma utilizada, resaltando las ventajas que ello aporta a la consecución de la aplicación y, se referirán los diferentes elementos que integran esta última y sus respectivas funcionalidades.

Por cada elemento, se describirán los aspectos respectivos y la interacción de sus alcances entre sí y con la aplicación; y en el apéndice E se especifican los recursos de hardware y de software requeridos para la elaboración del trabajo aquí presentado.

El tratar de resolver un problema específico, en este caso el mencionado en el objetivo, involucra de manera directa los aspectos propios del manejo del formato Excel, el código Java y el manejo de bases de datos e interfaces, entre otros.

4.2 Esquema del flujo de la operación

En su forma más genérica, el esquema del flujo de la operación considera la recopilación de datos y su captura (eventos que se realizan fuera de la aplicación), el registro de éstos, su recuperación y tratamiento y, el proceso de la carga y actualización de la información en la base de datos, así como su respectiva consulta, se puede apreciar en la Figura 4.1.

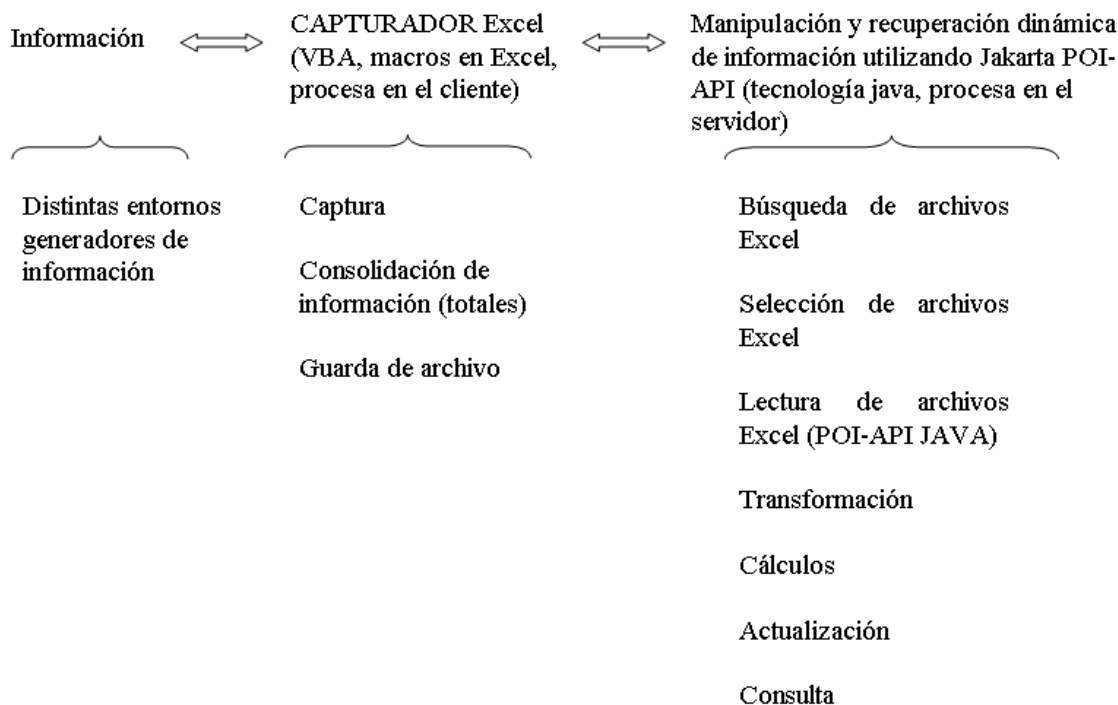


Figura 4.1 Flujo general de la operación.

4.3 Descripción del ambiente de operación

La aplicación desarrollada cuenta además de las clases Java necesarias, como se mencionó en el apartado anterior, con el entorno necesario que da funcionamiento a un ambiente WEB (la interfaz de usuario), que incluye las formas de acceso a la misma, así como del proceso que permite la actualización de la base de datos de prueba [14, Acceso a Bases de datos].

La interfaz WEB se ejecuta en el servidor correspondiente; y para efectos de ejemplo, este servidor será también el de aplicaciones.

Lo antes mencionado liberará al cliente de proceso adicional, pues solo tiene que subir los recursos Excel y no necesita la base de datos en su equipo.

Se tiene una interfaz de control que se encarga de indicarles a los demás componentes el inicio de la operación en alguna de sus dos modalidades: usuario administrador y usuario básico.

El proceso de la interfaz, para cada una de las opciones arriba referidas se basa en la verificación del nivel del usuario y conforme al mismo, corresponde la asignación de las operaciones inherentes.

Los módulos que integran la aplicación son los de Usuarios, Recursos Excel, Actualización de Base de Datos, así como el de Consulta a la Base de Datos.

El entorno de operación para el usuario administrador, considera las cuatro opciones arriba mencionadas.

Por su parte, para el usuario básico, solo se le presenta la interfaz de Registro de recursos Excel, en una versión limitada, donde únicamente tiene oportunidad de ingresar nuevos registros y en su caso consultar los materiales Excel ingresados, pero sin posibilidad de modificarlos o eliminarlos.

En la operación de la aplicación se tienen elementos llamados comunes por ser referidos en diversas instancias, como son los de Ingreso (inicio de sesión) y de Fin de Sesión, cuya tarea principal de este último, es la de permitir salir del ambiente correspondiente.

Un elemento importante para la operación de la aplicación, es la búsqueda de recursos, y para realizar esto se filtra que el archivo generado por el capturador Excel y que ha de ser almacenado en el servidor sea con extensión .XLS.

Aspectos adicionales tendientes a optimizar el uso de la aplicación se reflejan en su entorno de operación, al mostrar datos concretos y opciones de elección específicas, aunado a los mensajes de verificación o en su caso de alerta de las decisiones de operación realizadas.

Cabe hacer notar que en todo el proceso de la aplicación en sus distintas vertientes, los diferentes usuarios no están en posibilidades de modificar de manera alguna la información recibida de origen (emanada del capturador Excel).

4.4 Soporte de la operación

Las aplicaciones que se apoyan en un ambiente de operación WEB [10, Rodríguez] deben ser soportadas por elementos de un adecuado nivel de confiabilidad, por esto es que el servidor WEB debe estar siempre disponible para la aplicación y es deseable que su configuración esté enfocada a garantizar un adecuado uso de sus recursos, evitar errores y atender peticiones. Siempre debe poder satisfacer los requerimientos en forma aceptable.

La elaboración de la aplicación está basada en una arquitectura cliente/servidor, la cual se implementó en un entorno Java, dado que es una herramienta que incluye los elementos necesarios para el desarrollo de esta clase de aplicaciones. Se utilizó específicamente el lenguaje Java y algunas de sus variantes (JSP, Servlets y JavaScript [7, López]), que en conjunto, conforman un

esquema práctico, sencillo y ampliamente utilizado, mediante el que se generan herramientas seguras y de gran versatilidad.

Así mismo, se ha empleado el entorno de desarrollo integrado (IDE, por sus siglas en inglés) NETBEANS en su versión 5.5.

Para operar una aplicación WEB realizada en Java, es necesario contar con un contenedor, en este caso se tiene el entorno de TOMCAT 5.5 [16, Apache Tomcat], que es una plataforma del tipo software libre y que considera una amplia compatibilidad para los estándares respectivos. Algunas de las características claves para su utilización se mencionan a continuación:

- Es un servidor WEB con soporte de JSP [24, JSP] y servlets.
- Ha sido desarrollado y es actualizado por miembros de la Apache Software Foundation y voluntarios independientes.
- Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*.
- La jerarquía de directorios incluye, entre otros, el directorio que contiene las aplicaciones WEB, en este caso WEBAPPS [6, Kurniawan].

Cada uno de los módulos accede a la base de datos de prueba MySQL que proporciona la información a la aplicación. La conexión hacia este manejador de base de datos se realiza mediante el estándar MySQL Connector/J, el cual es una variante JDBC Tipo 4 para establecer la comunicación con servidores del tipo MySQL. Este conector es un desarrollo en Java puro para el protocolo de MySQL y no reside en la parte del cliente.

La aplicación de Java debe tener acceso a un controlador (*driver*) JDBC adecuado. Este controlador es el que implanta la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC [23, JDBC] y la base de datos real. De una manera muy simple, al usar JDBC se puede:

- Establecer una conexión a una fuente de datos (por ejemplo, una base de datos).
- Enviar consultas y comandos a la fuente de datos.
- Procesar los resultados

La referencia de conexión es:

(“jdbc:mysql://localhost: port/DBase”, userName, password)

Si fuese necesario que la aplicación se conectara a otra base de datos, del mismo tipo, sería suficiente con cambiar el nombre de ésta, el nombre del usuario, así como de la contraseña en la estructura de la declaración de conexión; para la conexión a otro manejador se elegiría el conector respectivo.

Un ejemplo de código es el siguiente:

```
public void connect() throws ClassNotFoundException, SQLException,
Exception{
// Define la variable de conexión
    Connection con = null;

    String msg = "";

    try {
// Define el registro del driver a utilizar
        Class.forName("com.mysql.jdbc.Driver").newInstance();

        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BDprueba","root
","pass");
    } catch (ClassNotFoundException cnfe) {
        msg = "ClassNotFoundException: no se localizo el driver.";
        throw new ClassNotFoundException(msg);
    } catch (SQLException cnfe){
        msg = "SQLException: No se pudo conectar con la BD";
        throw new SQLException(msg);
    } catch (Exception e){
        msg = "Exception: Un Error desconocido ha ocurrido.";
        throw new Exception(msg);
    }
}
```

4.5 Funcionalidad de la operación

En forma más específica y de acuerdo a la Tabla 4.5.1, se hace mención de las opciones que se tienen por cada tipo de usuario:

Tabla 4.5.1. Opciones por tipo de usuario			
Usuario administrador - Operación completa.			
<u>Módulo</u>	Funciones	<u>Módulo</u>	Funciones
<u>Acceso</u>	<ul style="list-style-type: none"> Identificador 	<u>Proceso</u>	<ul style="list-style-type: none"> Recibir información Procesar datos Mostrar resultados Enviar información Cerrar sesión

<u>Usuarios</u>	<u>Usuarios Alta</u> <ul style="list-style-type: none"> • Seleccionar opción • Capturar datos • Actualizar B. de D. • Consultar <u>Usuarios Baja</u> <ul style="list-style-type: none"> • Seleccionar opción • Ver registros • Elegir registro • Borrar registro • Consultar <u>Usuarios Cambios</u> <ul style="list-style-type: none"> • Seleccionar opción • Ver registros • Elegir registro • Modificar registro • Consultar • Cerrar Sesión 	<u>Actualización</u>	<ul style="list-style-type: none"> • Recibir información • Conectar con Base de Datos • Armar INSERT • Actualizar Base de Datos • Mostrar resultados • Cerrar Sesión
	<u>Recursos</u>	<u>Recursos Alta</u> <ul style="list-style-type: none"> • Buscar • Seleccionar • Envío de archivos elegidos • Registro de recursos (alta) • Visualización de archivos • Cerrar Sesión 	
	<u>Recursos Baja</u> <ul style="list-style-type: none"> • Buscar • Seleccionar • Borrar archivos elegidos • Cerrar sesión 	<u>Consulta</u>	<ul style="list-style-type: none"> • Conectar con Base de Datos • Armar SELECT • Consultar • Mostrar resultados • Cerrar Sesión
<u>Usuario básico – sólo Recursos Excel.</u>			
<u>Módulo</u>	<u>Función</u>		
<u>Acceso</u>	<ul style="list-style-type: none"> • Identificador 		
<u>Alta</u>	<ul style="list-style-type: none"> • Buscar • Seleccionar • Envío de archivos • Registro de recursos (alta) • Visualización de archivos • Cerrar Sesión 		

4.5.1. Acceso a la aplicación

En las Tablas 4.5.1.1 y 4.5.1.2 se describen las formas WEB que fueron desarrolladas para dar funcionalidad a los módulos de Acceso y Menú Principal de la aplicación.

Tabla 4.5.1.1 Forma Home2.html	
Nombre de la Forma:	Home2.html
Acceso	
Descripción:	Página principal del acceso a la aplicación, contiene el enlace

Tabla 4.5.1.1 Forma Home2.html	
Nombre de la Forma:	Home2.html
	correspondiente para el procedimiento de verificación de tipo de usuario, el cual permite acceder a la navegación de las demás secciones. (Usuario administrador, Usuario básico)

Tabla 4.5.1.2 Forma myMenu.jsp	
Nombre de la Forma:	myMenu.jsp
	Menú Principal
Descripción:	Página principal del módulo de operación, contiene el enlace correspondiente a las demás opciones de navegación (Módulo de usuarios, Módulo de recursos Excel, Módulo de Actualización, Módulo de consulta, Cierre de sesión).

Estas formas como se indica, son las que se encargan de la interfaz de ingreso a la aplicación y de las opciones de operación para el usuario administrador.

4.5.2. Verificación de usuario

Aquí se describe de manera breve el proceso de la verificación de usuarios. Ver Tabla 4.5.2.1.

Tabla 4.5.2.1 Forma IndexSistema.jsp	
Nombre de la forma	IndexSistema.jsp
	Verifica usuario
Recibe parámetros de	Identificador y contraseña del usuario
Regresa:	Tipo de usuario
Descripción:	Verifica que un usuario esté registrado en la base de datos vía los parámetros requeridos y, en base al tipo de usuario permite el acceso a la sección correspondiente (usuario básico o usuario administrador); en caso de que el usuario no esté registrado, la aplicación restringe el acceso.

4.5.3. Módulos del entorno de operación del usuario administrador

A continuación se describen las formas implementadas para el funcionamiento del ambiente del usuario administrador.

4.5.3.1. Usuarios y recursos Excel.

Se describen en las Tablas 4.5.3.1.1 a 4.5.3.1.4 las formas WEB desarrolladas para dar funcionalidad a las opciones del módulo de usuarios.

Tabla 4.5.3.1.1 Forma Administradores_ABC2.jsp	
Nombre de la Forma:	Administradores_ABC2.jsp

Tabla 4.5.3.1.1 Forma Administradores_ABC2.jsp	
Nombre de la Forma:	Administradores_ABC2.jsp
Menú Usuarios	
Descripción:	Página principal del módulo de usuarios, contiene las ligas correspondientes a la navegación de las demás opciones: alta, baja o modificaciones al registro del usuario. Estas operaciones así como la de baja del registro, de recurso Excel, sólo puede realizarlas el usuario administrador.

Tabla 4.5.3.1.2 Forma Alta_Administrador.jsp	
Nombre de la Forma:	Alta_Administrador.jsp
Alta de Usuarios	
Descripción:	Presenta al usuario administrador la forma de registro de un nuevo usuario.

Tabla 4.5.3.1.3 Forma BorraAdministrador.jsp	
Nombre de la Forma:	BorraAdministrador.jsp
Baja de Usuarios	
Descripción:	Presenta al usuario administrador la forma para dar de baja a un usuario.

Tabla 4.5.3.1.4 Forma ModifAdministrador.jsp	
Nombre de la Forma:	ModifAdministradores.jsp
Actualizar el registro del Usuario	
Descripción:	Presenta al usuario administrador la forma para modificar el registro de un usuario.

El uso de las formas antes descritas, propician la operación del módulo de usuarios, incluyendo las opciones de alta y baja de usuarios, así como de cambios en su registro.

En las Tablas 4.5.3.1.5 a 4.5.3.1.9, se describen las formas correspondientes para la funcionalidad del módulo de registro de recursos Excel, también utilizando tecnología Java y JSP.

Tabla 4.5.3.1.5 Forma archivos.jsp	
Nombre de la Forma:	archivos.jsp
Menú de Recursos Excel	
Descripción:	Contiene las ligas correspondientes a la navegación de las demás opciones de este menú (altas, bajas y consulta) de recursos; no puede haber opción de modificaciones para dichos recursos.

Tabla 4..5.3.1.6 Forma Examinar	
Nombre de la Forma:	Examinar
Búsqueda de recursos Excel	
Descripción:	Interfaz gráfica donde el usuario puede buscar y elegir los archivos con extensión .XLS ha ser utilizados en el proceso de alta de recursos Excel a la Base de Datos.

Tabla 4.5.3.1.7 Forma subeArchivo2.jsp	
Nombre de la Forma:	subeArchivo2.jsp
Alta de recursos Excel	
Descripción:	Presenta al usuario la forma donde puede ejecutar el proceso de alta de recursos Excel a la Base de Datos.

Tabla 4.5.3.1.8 Forma verArchivos.jsp	
Nombre de la Forma:	verArchivos.jsp
Consulta de recursos Excel	
Descripción:	Presenta la forma donde puede consultar los archivos Excel, cuya referencia ha sido actualizada en la Base de Datos.

Tabla 4.5.3.1.9 Forma BorraMaterial.jsp	
Nombre de la Forma:	BorraMaterial.jsp
Baja de recursos Excel	
Descripción:	Presenta la forma donde se puede realizar la baja del registro de recursos Excel.

Mediante la aplicación de las formas antes mencionadas se logra, por parte del usuario administrador, la operación del módulo de registro de recursos Excel

4.5.3.2. Actualización

Las formas WEB para el funcionamiento del ambiente del proceso de actualización de la base de datos de prueba se describen a continuación. Ver las Tablas de la 4.5.3.2.1 a 4.5.3.2.3.

Tabla 4.5.3.2.1 Forma ctotalac_1.jsp	
Nombre de la Forma:	ctotalac_1.jsp
Lectura de Recursos Excel	
Descripción:	Página donde se elige el portafolio de los recursos generados por el CAPTURADOR Excel y mediante el uso del POI API se recupera y realiza el tratamiento de la información correspondiente a los totales de los rubros capturados. En este entorno se muestran los resultados de la lectura y la recuperación de la información.

Tabla 4.5.3.2.2 Forma ctotalac_4.jsp	
Nombre de la Forma:	ctotalac_4.jsp
Preparación de datos para el proceso de actualización	
Descripción:	Preparación y transferencia de la información recuperada del libro Excel, para el proceso que actualizará las tablas de costos totales de inversión operacional, estudios de campo y gabinete , así como de estudios de sísmica 2D y 3D en la base de datos de prueba. El usuario puede visualizar los resultados del proceso referido.

Tabla 4.5.3.2.3 Forma GuardaDatos2.jsp	
Nombre de la Forma:	GuardaDatos2.jsp
Actualiza la Base de Datos de prueba	
Descripción:	Toma los totales resultantes del proceso anterior, habilita la conexión con la Base de Datos, elabora las instrucciones del tipo INSERT TO (SQL) para insertar la información y ejecutar la actualización de la Base de Datos de prueba. Presenta los resultados de este proceso, donde indica si fue exitoso o ya se había ejecutado la actualización correspondiente.

Las formas antes descritas permiten ejecutar el proceso de actualización a la base de datos de prueba.

4.5.3.3. Consulta

Tener la certeza de que el proceso que conlleva la actualización de la información se haya realizado de forma correcta es prioritario. Por ello es que se instrumentó una opción para realizar dicha consulta, la Tabla 4.5.3.3.1 describe de manera breve la forma correspondiente.

Tabla 4.5.3.3.1 Forma cdtablas.jsp	
Nombre de la Forma	cdtablas.jsp
Consulta las tablas de costos totales de la Base de Datos de prueba	
Descripción	Conecta con la base de datos de prueba, prepara instrucciones para consultar (SELECT), ejecuta la consulta y muestra los resultados. Se muestra de manera integrada la actualización por cada rubro.

4.5.4. Ambiente del usuario básico

A continuación se describen las formas implementadas para el funcionamiento del ambiente del usuario básico.

Se hace mención de que la descripción previa, realizada en los apartados de Acceso a la aplicación y de Verificación de Usuarios, también aplica en este caso, ya que dichos entornos son comunes para los perfiles de usuario definidos.

El usuario básico o no administrador tiene la opción de acceso al módulo de recursos Excel en una versión limitada, es decir, donde únicamente tiene la oportunidad de ingresar nuevos registros y en su caso consultar los materiales Excel ingresados, pero sin posibilidad de eliminarlos.

4.5.4.1. Módulo de registro de recursos Excel

En las Tablas 4.5.4.1.1 a 4.5.4.1.3 se describen las herramientas que componen esta Forma.

Tabla 4.5.4.1.1 Forma archivos_usuario.jsp	
Nombre de la Forma:	archivos_usuario.jsp
Menú de Recursos Excel (versión limitada)	
Descripción:	Contiene las ligas correspondientes a la navegación de las demás opciones (altas y consulta de recursos).

Tabla 4.5.4.1.2 Forma Examinar	
Nombre de la Forma:	Examinar
Búsqueda de recursos Excel	
Descripción:	Presenta al usuario la interfaz gráfica donde puede buscar y elegir los archivos con extensión .XLS ha ser utilizados para el proceso de alta de recursos Excel a la Base de Datos.

Tabla 4.5.4.1.3 Forma subeArchivo2.jsp	
Nombre de la Forma:	subeArchivo2.jsp
Alta de recursos Excel	
Descripción:	Presenta al usuario la forma donde puede ejecutar el proceso de alta de recursos Excel a la Base de Datos.

Con las formas antes descritas, el usuario básico tiene la posibilidad de ingresar la referencia de recursos Excel en la tabla “archivos” y cargar los recursos en el servidor

4.6 Localización de la aplicación

El desarrollo de aplicaciones Java en el entorno WEB, considera que el programador pueda adicionar mejoras o nuevas opciones, sin que esto afecte la operación.

De la misma manera los usuarios pueden acceder a la aplicación, de acuerdo a su perfil de usuario, desde su sitio de operación. Esta es una de las funcionalidades que permite el ambiente WEB. En el caso de esta aplicación, cuando el usuario básico tenga el acceso correspondiente, podrá registrar nuevos recursos Excel sin necesidad de trasladarse físicamente o enviarlos mediante correo electrónico o alguna otra forma poco segura.

Simplemente bastará con ingresar a la aplicación (desde el entorno del cliente), elegir la opción de los recursos a ingresar y ejecutar. La referencia será almacenada en la base de datos y los archivos en el servidor (no residen físicamente del lado del cliente), lo que permitirá al usuario administrador procesar la información en ellos contenida y establecer el tiempo en que residirán ahí.

En este punto, la opción de Recursos Excel, toma un aspecto relevante, pues propicia que los usuarios del captador Excel programen sus actividades al respecto y puedan subir los libros correspondientes al entorno de la aplicación, esto en concordancia con el usuario administrador y con los manejadores de la base de datos.

La aplicación por parte del cliente solo requiere los accesos correspondientes, pues todo el proceso corre en la parte del servidor.

Al obtener la ubicación (URL) de la aplicación, se accede al proceso dinámico. Esta operación se realiza con el apoyo del servidor WEB y el de aplicaciones, soportada por la estructura de comunicaciones existente en la empresa.

Resumen

Se eligió la plataforma Jakarta POI API ya que contiene los paquetes y clases que permiten interactuar con documentos Excel y acceder a su información inclusive a nivel de celdas y que en combinación con la tecnología Java ha permitido establecer los medios para satisfacer la necesidad Excel-Java-MySQL a un bajo costo, al considerar que la misma es del tipo software libre (por sus siglas en inglés Open Source).

La aplicación se dividió en cuatro módulos básicos: Usuarios, Recursos Excel, Actualización y Consulta de la base de datos; y dos adicionales para las opciones de inicio y fin de sesión, la primera desde la interfaz gráfica de inicio, que es una página HTML y, la segunda desde el entorno de operación de la aplicación.

Las interfaces de cada uno de los módulos permiten su operación y enlace con las distintas opciones que los conforman dentro del entorno WEB, creado para tal efecto.

Las interfaces están desarrolladas mediante JSP y Java y facilitan al usuario seleccionar las opciones existentes y navegar y operar las mismas.

La aplicación está concebida como una interfaz entre Excel y MySQL. La conectividad con la base de datos de prueba se desarrolló mediante JDBC y sus clases Java, que permiten la conexión a través de su manejador desarrollado en Java puro y que incluye el protocolo respectivo para MySQL. La seguridad de la conexión para el cliente estaría considerada a nivel de red por el firewall que controla los acceso externos y, al nivel de la base de datos, en este caso, por el registro de las sesiones (identificador y contraseña) para cada usuario.

La aplicación considera las funciones que permiten el acceso a los documentos Excel, la recuperación y tratamiento de su información, la consolidación de ésta en datos globales y la actualización de los mismos en la base de datos de prueba.

Mediante la conexión ya mencionada, se realizan las instrucciones SQL adecuadas para la inserción y consulta de de información en la base de datos, apoyado esto en la funcionalidad de la interfaz JDBC.

Cabe hacer mención de la importancia de acordar, en la práctica, con el administrador de la base de datos a efecto de adoptar, en el futuro, otras formas de seguridad y manejo de concurrencia, establecidas en la empresa, basadas en criterios de acceso y administración de la información previamente establecidos.

En el Capítulo siguiente se describen las pruebas de funcionamiento efectuadas, los resultados obtenidos y su interpretación.

página en blanco

CAPÍTULO 5. PRUEBAS Y RESULTADOS

La consecución de las etapas de análisis y desarrollo se aprecia con la realización y presentación de las pruebas respectivas y la interpretación de sus resultados; en este Capítulo se describen las pruebas efectuadas tanto para la captura de datos como para su tratamiento dinámico mediante la aplicación que en este trabajo es presentada y, se muestran los resultados obtenidos.

La finalidad de las pruebas se orienta a las acciones que permitan verificar el correcto funcionamiento tanto del capturador Excel como de la aplicación, al operar esta última con base en la información emanada del proceso de captura correspondiente.

Se hace mención de que la información captada mediante el capturador Excel es totalmente ficticia y no tiene relación con proyecto alguno.

Cabe recordar que para la realización de este trabajo, se tomó como caso de estudio una empresa como es la que se tiene en nuestro país, dedicada, entre otras, a las actividades de exploración de recursos naturales y específicamente relacionadas con los proyectos de yacimientos petrolíferos.

Se considera la existencia de una base de datos de prueba a nivel corporativo, de la que en su momento la empresa en turno, ha realizado ya el modelado correspondiente y, a la que se han adicionado nuevas tablas relacionadas con la administración de usuarios, de recursos Excel y de costo totales.

Así mismo se ha establecido un número de proyecto ficticio para la elaboración de las pruebas de actualización.

5.1 Captura de datos

Con respecto a las opciones que contiene el capturador Excel se realizaron varias capturas de prueba, con datos ficticios, simulando diversas situaciones tanto para las categorías de inversión operacional, estudios de campo y gabinete, como para las de estudios de sísmica 2D y 3D.

Lo anterior, permitió crear varios libros, los que incluyen datos acumulados y el cálculo de totales por categoría y, de los que en el módulo de recursos Excel de la parte dinámica se aprecia su referencia y pueden ser consultados.

La figura 5.1.1 muestra el menú del capturador Excel, el cual integra además de una opción de recálculo, las categorías de:

- Inversión operacional
- Sísmica
 - Sísmica 2D

- Sísmica 3D
- Estudios
 - Estudios de campo
 - Estudios de gabinete.



Figura 5.1.1 Opciones del menú del capturador Excel.

Mediante la opción de Recalcular Totales, se permite el cálculo y despliegue de los totales resultantes de la información capturada, Por considerar su importancia, en la Figura 5.1.2 se muestra la disposición de esta hoja y la forma de presentar dichos totales.

	T	U	V	W	X	Y	Z	AA	AB	AC
	2026	2027	2028	2029	2030	2031	2032	2033	2034	Total
Totales										
Sismica 2D										
5 Kilómetros	0	0	0	0	0	0	0	0	0	50
6 Millones de pesos	0	0	0	0	0	0	0	500	0	660
Sismica 3D										
9 Kilómetros cuadrados	0	0	0	0	0	0	0	0	0	90
10 Millones de pesos	0	0	0	0	0	0	0	0	0	90
12 Total de inversión	0	0	0	0	0	0	0	500	0	750
Estudios de Campo										
16 Número	0	0	0	0	0	0	0	0	10	140
17 Millones de pesos	0	0	0	0	0	0	0	0	120	460
Estudios de Gabinete										
20 Número	0	0	0	0	0	0	0	0	0	800
21 Millones de pesos	0	0	0	0	0	0	0	0	0	510
23 Total de estudios	0	0	0	0	0	0	0	0	10	1940
24 Total de millones de pesos	0	0	0	0	0	0	0	0	120	990
Inversión Operacional										
28 Número	0	0	0	0	0	0	0	0	2690	3485
29 Millones de pesos	0	0	0	0	0	0	0	0	3060	3655
Ir al menú										

Figura 5.1.2 Hoja de totales del capturador Excel.

Se hace énfasis que en el apéndice B se detalla el manual de uso de esta herramienta de captura.

Se menciona también que, además de que las capturas realizadas fueron de gran importancia para comprobar los alcances del capturador Excel en un ambiente considerado real, lo es también el hecho de su utilidad para verificar que el diseño y operación de la solución dinámica en el módulo de registro de recursos Excel, fuese acorde para acceder, leer y recuperar, de manera satisfactoria, los datos residentes en dichos libros.

5.2 Pruebas iniciales

Como parte de las pruebas iniciales de la funcionalidad de la aplicación se comprobó la validación del acceso a la misma.

Ésta se realiza a través de la interfaz gráfica de presentación y mediante el formulario que solicita los datos de identificador y contraseña del usuario; se turnan éstos al procedimiento que verifica la existencia del registro respectivo y que en su caso canaliza, de acuerdo al perfil del usuario, a la sesión respectiva. La Figura 5.2.1 muestra la interfaz gráfica para el ingreso a la aplicación e inicio de sesión de acuerdo al perfil del usuario.

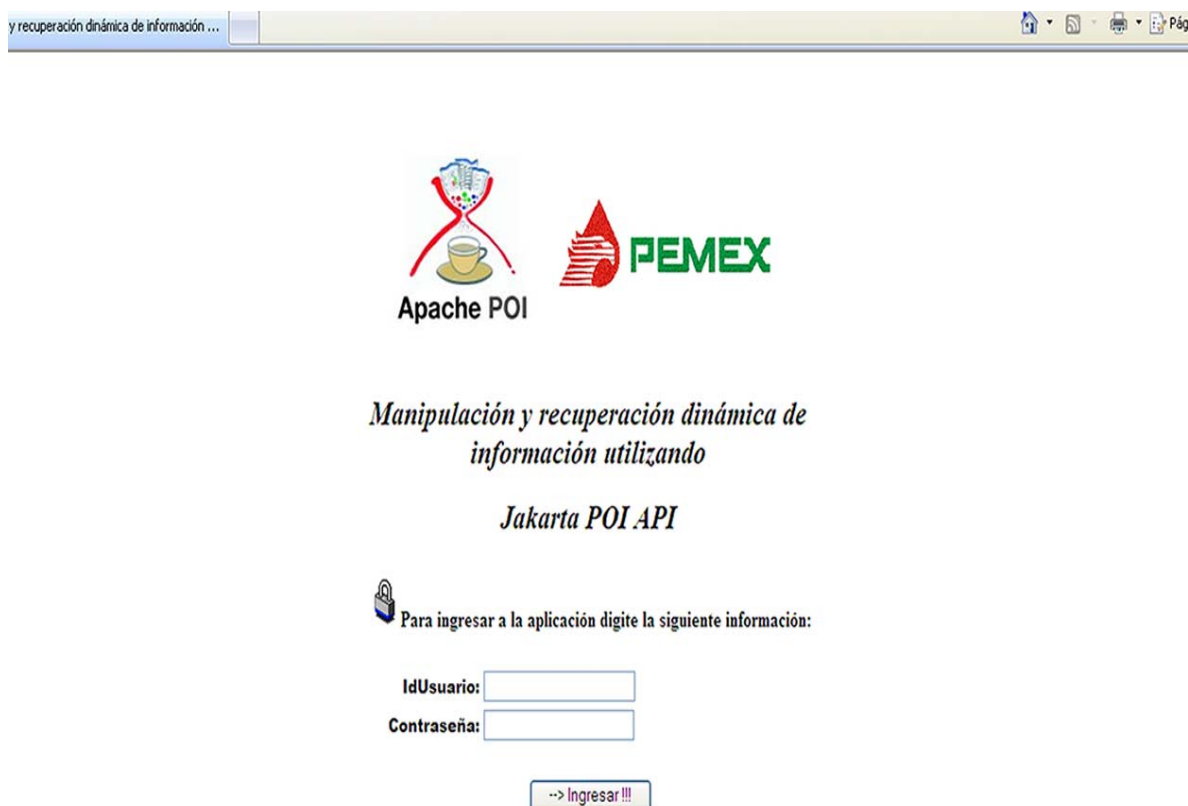


Figura 5.2.1 Interfaz gráfica para el ingreso a la aplicación.

5.3 Recuperación y carga de recursos Excel

En cuanto a la recuperación de recursos Excel, las pruebas realizadas contemplaron el verificar que el proceso de examinar, elegir y actualizar la tabla “archivos” en la base de datos fuese exitoso.

Además se verificó el cumplimiento para el módulo del cliente de:

- Establecer comunicación vía WEB.
- Efectuar la conexión respectiva con la base de datos
- Cargar el archivo correspondiente al servidor
- Actualizar la base de datos.

A este punto se hace notar que ningún usuario tiene la posibilidad de alterar la información de origen, solo el usuario administrador puede borrar el registro de la tabla “archivos”, pero no puede modificar el contenido del archivo cargado.

Aquí se establece la importancia de esta etapa, ya que de su adecuado funcionamiento depende la correcta recuperación de información de los formatos Excel, utilizando precisamente el Jakarta POI API.

Como parte del proceso antes referido, la Figura 5.3.1, muestra que desde el módulo de registro de recursos Excel, se puede realizar el proceso de examinar, seleccionar y enviar la referencia del archivo en turno a la tabla “archivos y, efectuar la actualización respectiva en la base de datos.



Figura 5.3.1 Interfaz gráfica para el módulo de registro de recursos Excel.

Al activar la opción de Examinar, se aprecia en la Figura 5.3.2 la pantalla Elegir archivo, en la que se muestran, entre otros, los archivos generados por el

capturador Excel, como es el denominado capturador_prueba1.xls, el cual será elegido para su incorporación al servidor.

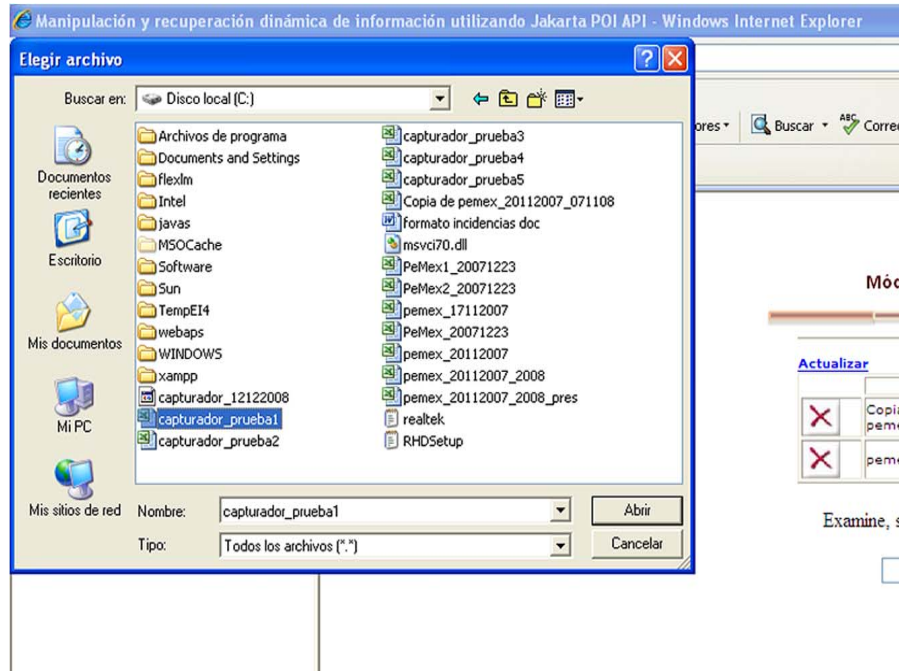


Figura 5.3.2 Interfaz gráfica que muestra lista de archivos a elegir.

La Figura 5.3.3 muestra en el recuadro de Examinar ya señalado, el nombre del archivo elegido, por lo que se procede a su envío y actualización.

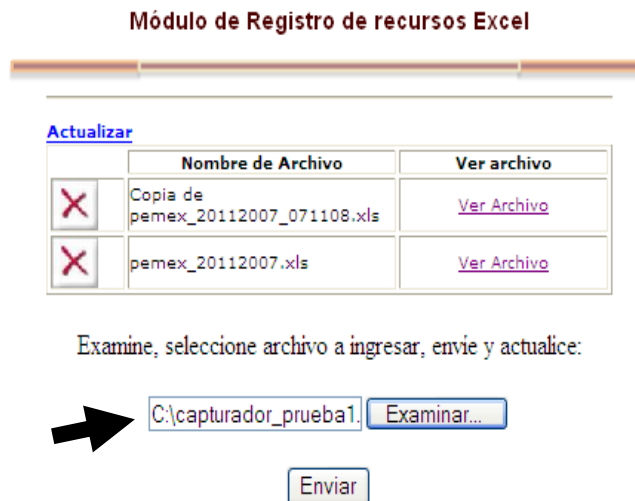


Figura 5.3.3 Archivo seleccionado generado por el capturador Excel.

Al accionar el botón Enviar, la aplicación informa al usuario que el archivo seleccionado será enviado para su registro en la base de datos, véase la Figura 5.3.4



Figura 5.3.4 Informa al usuario que el archivo seleccionado que será enviado para su registro en la base de datos de prueba.

Los resultados obtenidos se muestran en la Figura 5.3.5, que indica que la actualización fue realizada con éxito y que el archivo ya fue almacenado y puede ser utilizado por la aplicación.

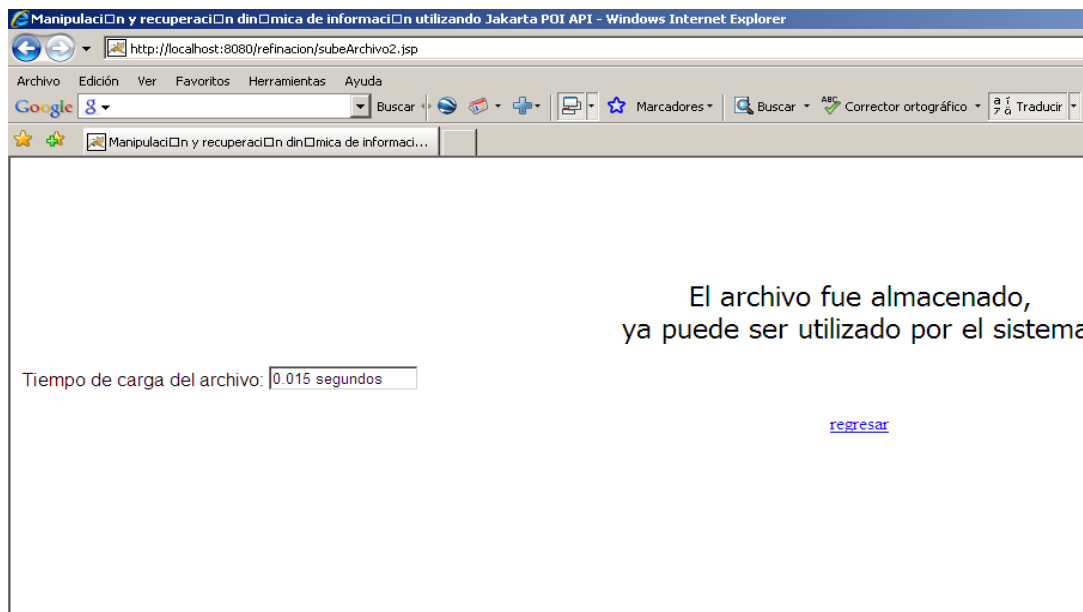


Figura 5.3.5 Se indica al usuario que la actualización fue exitosa.

Al final de este proceso y al activar la opción de actualizar, la aplicación indica mediante la interfaz gráfica, que el archivo elegido ha sido actualizado y la aplicación lo muestra ya como parte de los recursos que pueden ser revisados y posteriormente utilizados para actualizar la base de datos, véase la Figura 5.3.6.



Figura 5.3.6 Informa al usuario que el archivo elegido ha sido cargado y puede ser consultado.

Como parte de esta opción se realizó la siguiente prueba, la cual consistió en verificar el tiempo que tarda el proceso de carga del archivo Excel al servidor, tanto desde el entorno del computador que emula las veces de servidor WEB y de aplicaciones, como desde un equipo remoto con acceso a la aplicación. Para realizar esta prueba se implementó el código respectivo, adicionado al proceso de carga y mediante el cual se pudo cuantificar el lapso de este último.

La Tabla 5.3.1 muestra los resultados obtenidos, en donde se aprecian los tiempos registrados.

Tabla 5.3.1 Proceso de carga al servidor de archivos de recursos Excel.				
No.	Nombre de Archivo	Tamaño en KB	Tiempo de envío local (seg.)	Tiempo de envío LAN (seg.)
	Capturador_prueba1	1,851	0.015	0.016
	Capturador_prueba2	1,979	0.016	0.016
	Capturador_prueba3	1,995	0.016	0.018
	Capturador_prueba4	2,110	0.032	0.035
	Capturador_prueba5	1,853	0.015	0.016

Según los datos mostrados en la Tabla 5.3.1, el tiempo de carga de los archivos Excel al servidor, no resulta moroso y se considera aceptable como parte del proceso de actualización.

Este módulo permite también y, solo al usuario administrador, eliminar la referencia de un recurso Excel registrado en la base de datos de prueba, para lo cual debe hacer uso de las opciones correspondientes.

En la Figura 5.3.7, se aprecian tres referencias, que existen en la base de datos, de archivos generados por el capturador Excel y de los cuales el usuario ha elegido borrar el primero denominado “PeMex2_20071223.xls”.



Figura. 5.3.7. Referencia de recursos Excel registrados en la base de datos de prueba.

Se debe elegir la referencia del archivo a ser eliminado, mediante el icono:

Al ubicar el cursor sobre el icono de la referencia elegida, la interfaz gráfica muestra un cuadro de diálogo el cual indica que se realizará la acción de:

Borrar registro.

En la Figura 5.3.8 se aprecia lo antes descrito y, solo para efecto de este ejemplo, se indica el archivo a ser borrado mediante un icono del tipo flecha.



Figura 5.3.8 La interfaz gráfica muestra un cuadro de diálogo que indica que se realizará la acción de Borrar registro

Toda vez que ha sido activada la opción de borrar, se indica un mensaje preventivo al usuario, en el que se indica la leyenda: **Este archivo será borrado** como se ve en la Figura 5.3.9 antes mostrada.

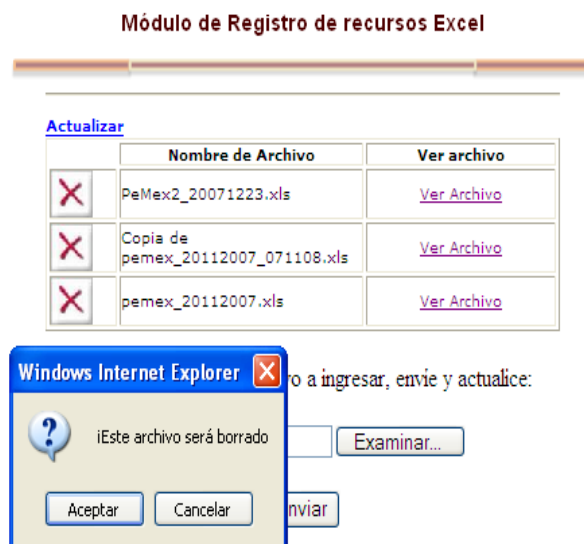


Figura 5.3.9 Indica al usuario el confirmar la baja de la elección realizada

Una vez que la acción ha sido aceptada, como se indica mediante el mensaje mostrado en la Figura 5.3.10, la aplicación elimina de la tabla “archivos” la referencia correspondiente.

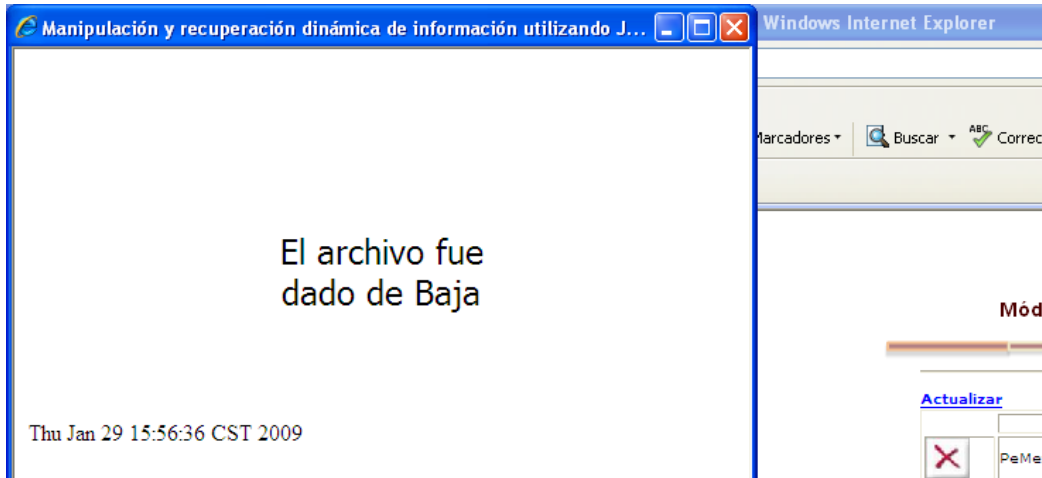


Figura 5.3.10 Informa al usuario de la eliminación de la referencia elegida

Finalmente, en la Figura 5.3.11 se aprecia que al activar Actualizar, la referencia ha sido eliminada del formulario y por lo tanto ya no puede ser consultada.

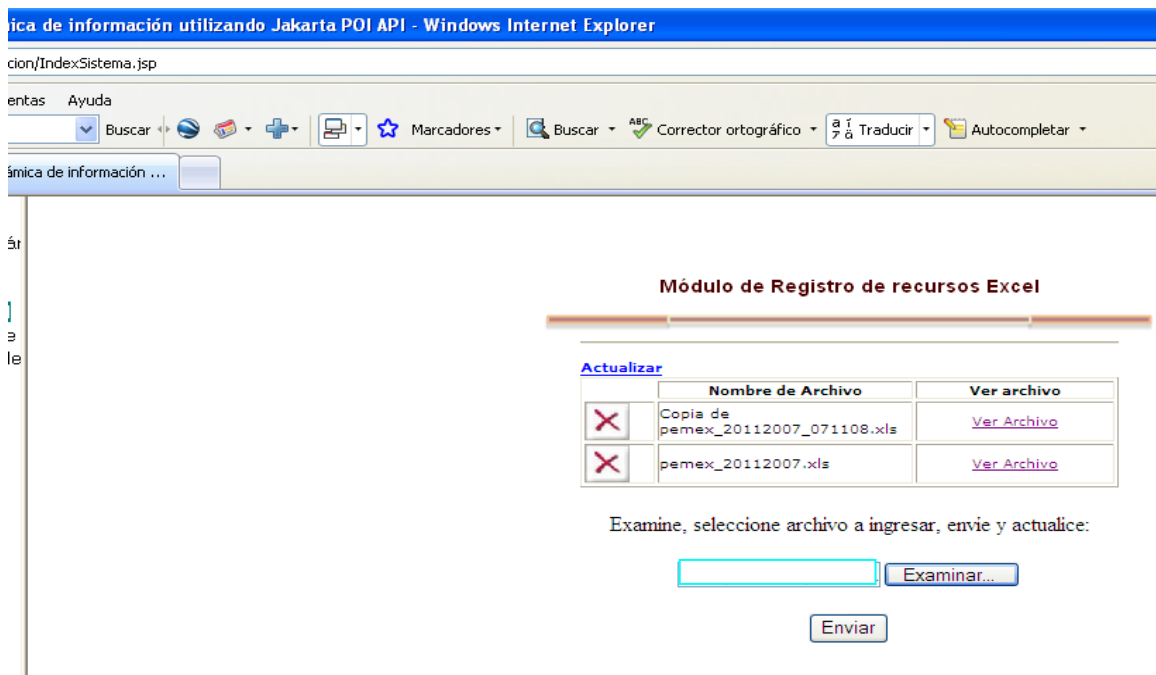


Figura 5.3.11 Ha sido realizada la actualización y eliminada la referencia del registro elegido.

5.4 Transmisión de datos de recursos Excel

Una vez realizada la selección y carga de archivos Excel al servidor, la siguiente prueba corresponde al proceso para la transmisión de los recursos Excel previamente cargados.

Para realizar esta prueba, es necesario elegir uno de los recursos Excel previamente cargados y ejecutar los pasos para dicho proceso, de acuerdo a las interfaces gráficas que se van mostrando al usuario.

Del menú izquierdo de la interfaz gráfica de operación presentado al usuario administrador, véase la Figura 5.4.1, se elige el módulo de actualización de la base de datos.

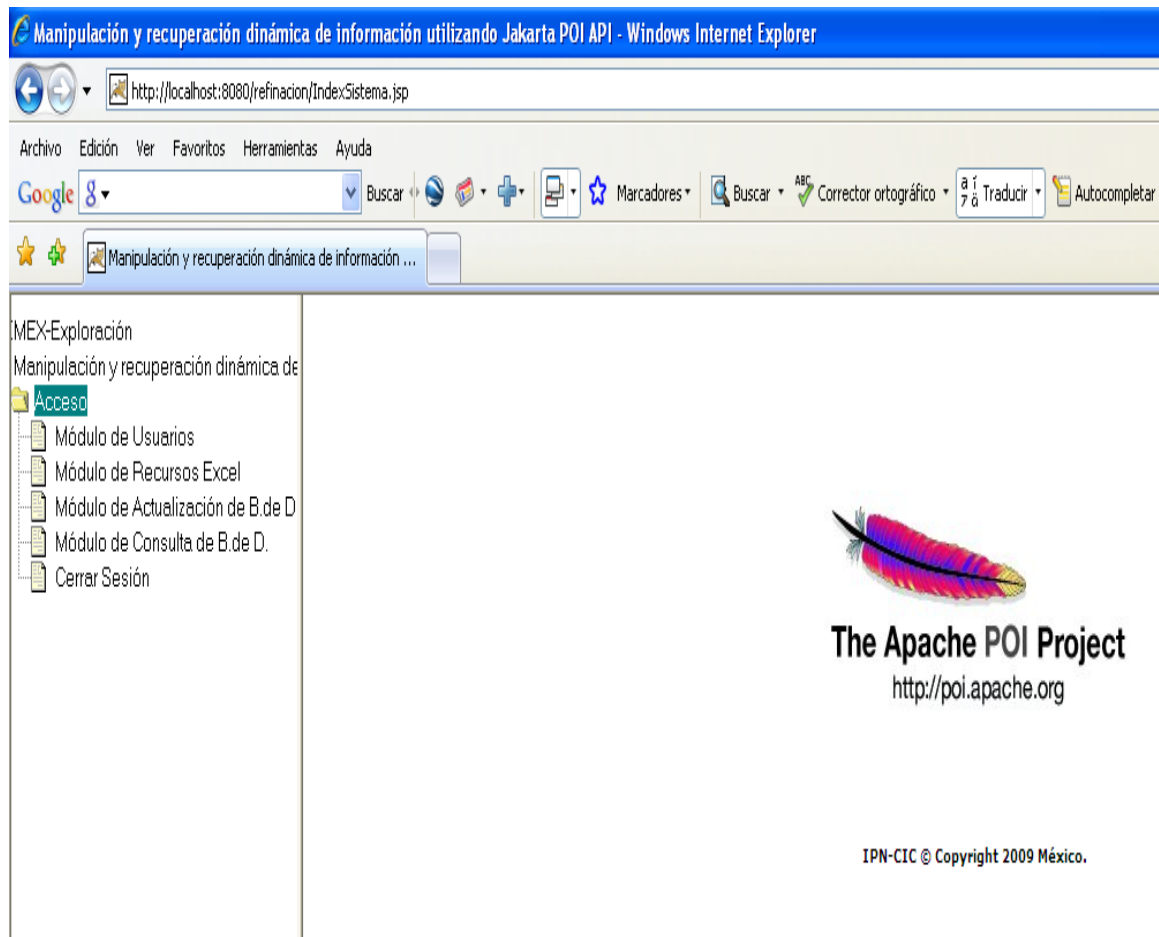


Figura 5.4.1 Menú de operación de la aplicación para el usuario administrador.

Al activar dicho módulo, se presenta la interfaz que permite elegir el archivo a procesar, como se muestra en la Figura 5.4.2.

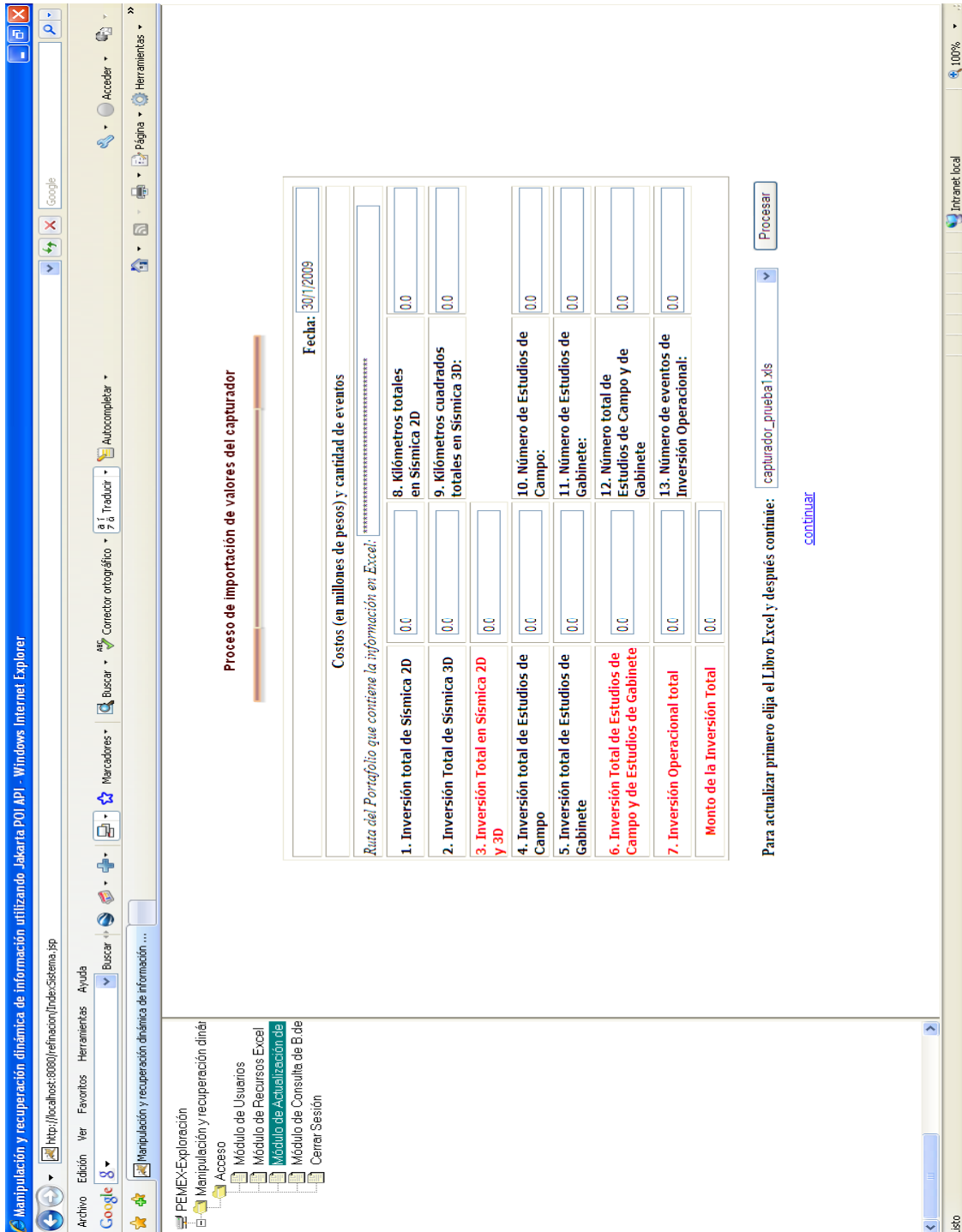


Figura 5.4.2 Interfaz que permite elegir el portafolio de datos a procesar.

Mientras no se elija un archivo y se inicie su proceso, la Tabla que muestra los datos estará con valores de cero (0.0).

Para dicha selección, se cuenta con un menú de archivos cargados y del cual se elige el que será procesado. En este caso se tienen los archivos cuya referencia se muestra en la figura 5.4.3, la cual es un extracto de la interfaz ya mostrada.

7. Inversión Operacional total	0.0	13. Número de eventos de Inversión Operacional:	0.0
Monto de la Inversión Total	0.0		

Para actualizar primero elija el Libro Excel y después continúe: [continuar](#)

Copia de pemex_20112007_071108.xls

- capturador_prueba1.xls
- Copia de pemex_20112007_071108.xls
- pemex_20112007.xls

Figura 5.4.3 Archivos disponibles para ser procesados.

Para este parte se ha elegido el archivo cuya referencia está sombreada y que se llama “Copia de pemex_20112007_071108.xls”, posteriormente se activa el botón Procesar, con lo que se despliegan los datos mostrados en la Figura 5.4.4.

Proceso de importación de valores del capturador

Fecha: 30/1/2009			
Costos (en millones de pesos) y cantidad de eventos			
Ruta del Portafolio que contiene la información en Excel: C:\Archivos de programa\Apache Software Foundation\Tomcat			
1. Inversión total de Sísmica 2D	660.0	8. Kilómetros totales en Sísmica 2D	50.0
2. Inversión Total de Sísmica 3D	90.0	9. Kilómetros cuadrados totales en Sísmica 3D:	90.0
3. Inversión Total en Sísmica 2D y 3D	750.0		
4. Inversión total de Estudios de Campo	480.0	10. Número de Estudios de Campo:	1140.0
5. Inversión total de Estudios de Gabinete	510.0	11. Número de Estudios de Gabinete:	800.0
6. Inversión Total de Estudios de Campo y de Estudios de Gabinete	990.0	12. Número total de Estudios de Campo y de Gabinete	1940.0
7. Inversión Operacional total	3655.0	13. Número de eventos de Inversión Operacional:	3485.0
Monto de la Inversión Total	5395.0		

Para actualizar primero elija el Libro Excel y después continúe: [continuar](#)

capturador_prueba1.xls

Figura 5.4.4 Despliegue de datos del Portafolio elegido

Se hace énfasis de que es en esta etapa donde se utiliza el Jakarta POI API mediante el paquete POIFS y su clase POIFSFileSystem para acceder a la información.

Así mismo, el paquete HSSF del mismo API, incluye las clases y métodos necesarios para que, mediante programación Java, se pueda acceder al libro recuperado al nivel de hoja, renglón y columna, logrando con esto obtener la información necesaria residente en el capturador Excel. En el apéndice F se muestra el procedimiento que realiza lo aquí descrito.

Ya que ha sido desplegada la información recuperada, al activar la opción de [continuar](#) se muestra la siguiente interfaz, Figura 5.4.5, en la que se detalla la información importada y que, en su caso, será actualizada en la base de datos de prueba; se menciona esto último, ya que si no se quiere realizar la actualización, el usuario puede optar por regresar a la interfaz anterior o cerrar la sesión, con lo que en cualquiera de las dos formas no se realizará la actualización respectiva.

Para realizar la actualización simplemente se elige el botón de Guardar datos globales y la aplicación efectúa lo procedente.

Proceso de importación a la base de datos

Valores importados del portafolio de información (Capturador en Excel)

Costos (en millones de pesos)		Fecha:	30/1/2009
1. Inversión Total en Sísmica 2D y 3D	750.0	2. Inversión Total en Estudios de Campo y de Gabinete	990.0
3. Inversión Operacional Total	3655.0	4. Monto Total de la Inversión	5395.0

[regresar](#)

Figura 5.4.5 Valores importados del portafolio de información

Cuando el proceso de actualización ha sido completado, se le informa al usuario mediante la interfaz gráfica respectiva, como se muestra en la Figura 5.4.6.

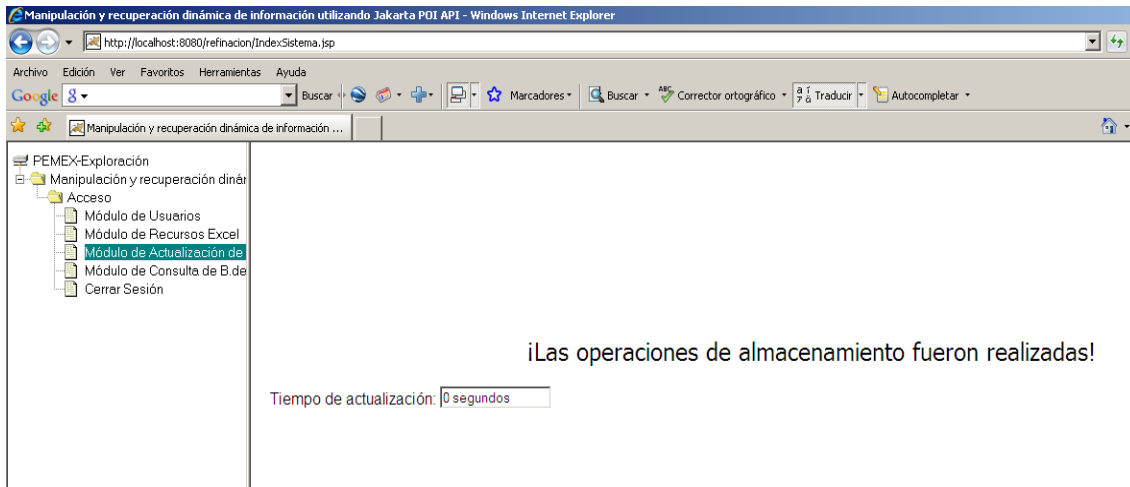


Figura 5.4.6 Se indica al usuario que la información fue almacenada.

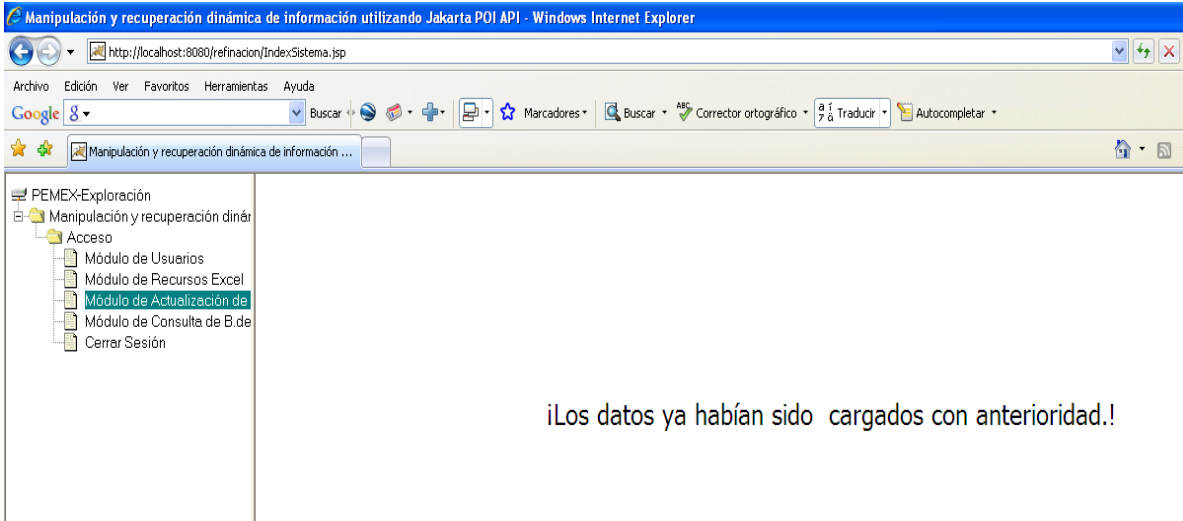
Así mismo y como parte de esta opción, se realizó la prueba de cuantificar el tiempo que tarda el proceso de actualización de la base de datos de prueba, en específico a las tablas de costos totales de los rubros de Inversión operacional, Sísmica 2D y 3D y, Estudios de Campo y Gabinete, tanto desde el entorno del computador que emula las veces de servidor WEB y de aplicaciones, como desde un equipo remoto con acceso a la aplicación, esto en modo administrador. Para realizar esta prueba se implementó el código respectivo, adicionado al proceso de actualización y mediante el cual se pudo cuantificar el lapso de este último.

La Tabla 5.4.1 muestra los resultados obtenidos, en donde se aprecian los tiempos registrados.

Tabla 5.4.1 Proceso de actualización de la base de datos de prueba, de las tablas de costos de Inversión operacional, Sísmica 2D y 3D y, Estudios de Campo y Gabinete.

No.	Nombre de Archivo	Tamaño en KB	Tiempo de envío local (seg)	Tiempo de envío LAN (seg)
	Capturador_prueba1	1,851	0.000	0.010
	Capturador_prueba2	1,979	0.010	0.010
	Capturador_prueba3	1,995	0.016	0.020
	Capturador_prueba4	2,110	0.012	0.030
	Capturador_prueba5	1,853	0.000	0.010

Continuando con la prueba de actualización y en el caso de que se elija un archivo Excel que ya ha sido procesado, la aplicación notifica al usuario de acuerdo a lo desplegado en la Figura 5.4.7.



¡Los datos ya habían sido cargados con anterioridad.!

Figura 5.4.7 Aviso del por qué no se realizó la actualización respectiva.

5.5 Consulta a la base de datos de prueba

En la consecución de la aplicación, se muestra en la Figura 5.5.1, los resultados de la consulta a la Base de Datos.

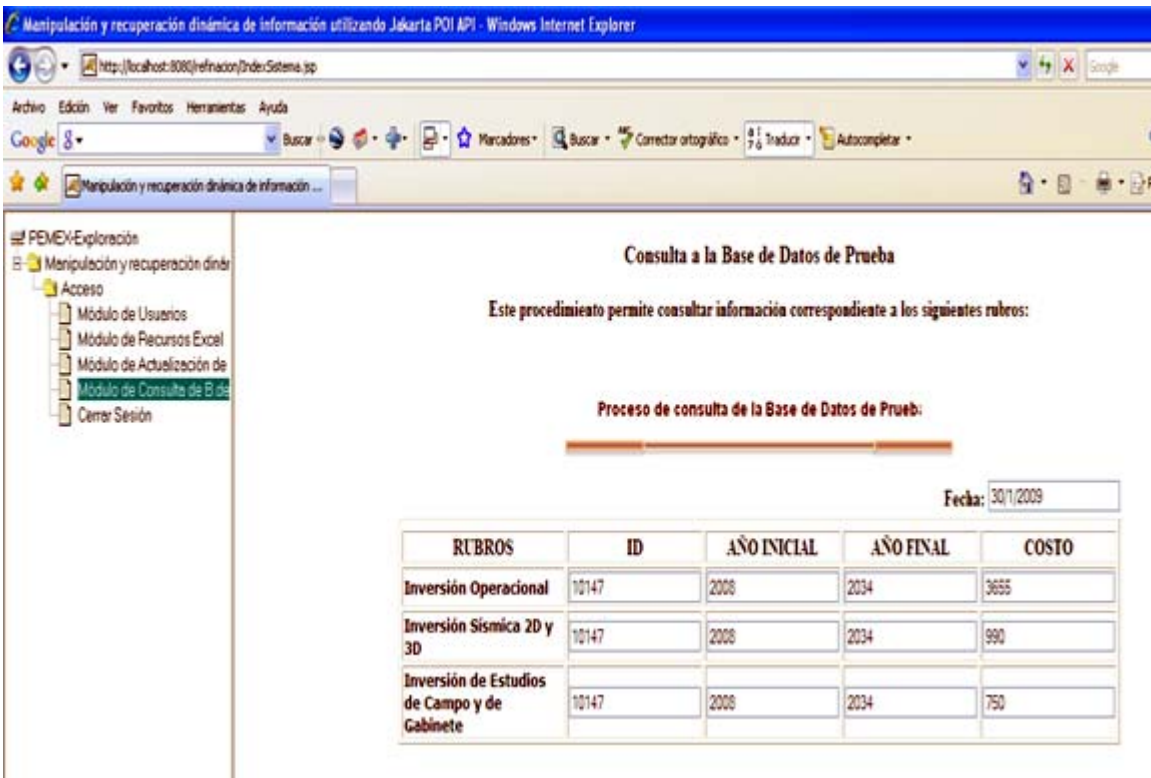





Figura 5.5.1 Resultado de la consulta a la base de datos de prueba.

Módulo de usuarios

La parte del registro y administración de los usuarios de la aplicación es por si misma importante, pero adiciona el aspecto de establecer el perfil correspondiente para cada uno.

Al seleccionar el módulo de usuarios, Figura 5.6.1, se presenta una interfaz gráfica en la que además de presentar los datos de los registros existentes, permite las siguientes operaciones mediante la selección del icono respectivo, a saber:

- Altas 
- Bajas 
- Cambios 

Lo anterior facilita al usuario el uso ágil y confiable del módulo y sus opciones.

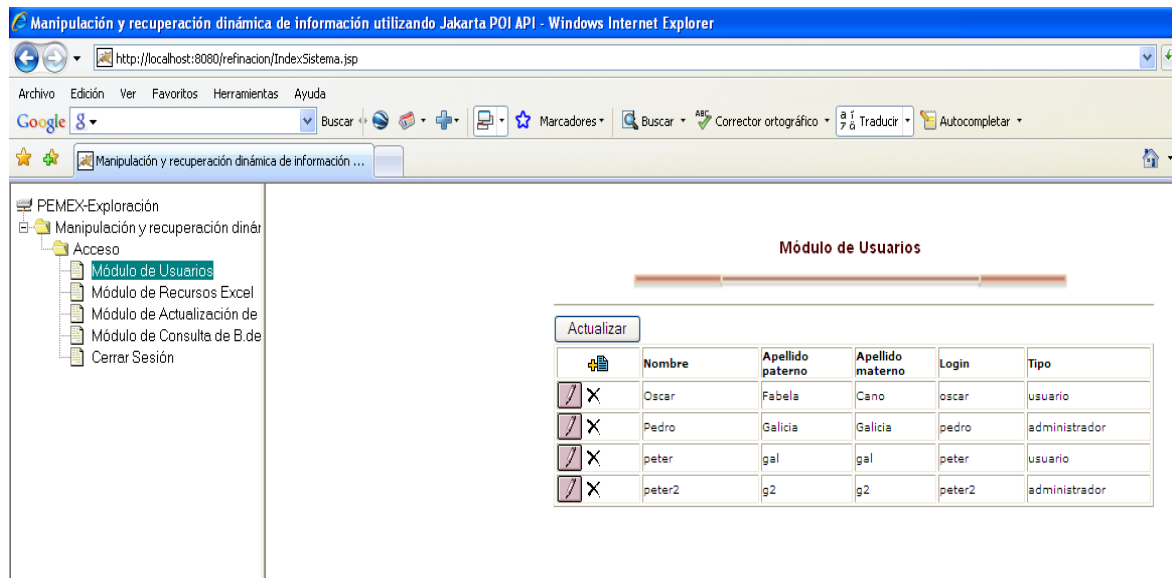


Figura 5.6.1 Interfaz gráfica del módulo de usuarios.

El apéndice C especifica con más detalle lo relacionado a la administración de usuarios y en donde se aprecia que el proceso respectivo es también exitoso.

El proceso de toma de decisiones relacionado con los proyectos de exploración de yacimientos petrolíferos tiene una base importante en el análisis de los costos y en la buena administración de los recursos inherentes, como son los humanos y los de información y, la aplicación aquí presentada proporciona al usuario, mediante los resultados mostrados información de primera mano, resultante del proceso de recuperar datos previamente captados vía un capturador Excel y actualizarlos en una base de datos corporativa MySQL [12, Schneider], orientados a soportar dicho proceso.

De lo descrito y mostrado en este apartado con respecto al funcionamiento de la aplicación, se puede considerar que el misma ha sido exitoso, toda vez que se ha logrado resolver mediante el uso del Jakarta POI API en conjunto con la tecnología Java, el esquema Excel-Java-Base de Datos.

Resumen

Los resultados obtenidos se engloban en el marco de la aplicación desarrollada, sus alcances y funciones, para lo cual se utilizó en una primera etapa un capturador Excel para concentrar la información relacionada con los proyectos de exploración de yacimientos petrolíferos, en una segunda etapa el uso de los mecanismos desarrollados en Java y apoyados en el uso del Jakarta POI API para el acceso a la información del libro Excel, su recuperación y tratamiento y, por último los procedimientos desarrollados en Java, combinados con instrucciones SQL y para el manejo del Connector/J (JDBC), para la actualización de la información en la base de datos de prueba.

El entorno WEB se logró mediante el desarrollo, en JSP [13, Tremblett], de las interfaces gráficas correspondientes y, soportadas éstas, mediante un navegador (en este caso Internet Explorer 7) y un servidor WEB (TOMCAT).

El análisis e interpretación de los resultados ha sido sobre la base del cumplimiento de los objetivos planteados. Para la captura de información, resultante de las actividades de exploración, se utilizó la herramienta desarrollada, es decir, el capturador Excel; se elaboró una aplicación que satisfizo la necesidad para la cual fue concebida, al lograr la correcta actualización de la información en la base de datos, así como su consulta, por parte de los diferentes usuarios.

Las interfaces se desarrollaron bajo el criterio de funcionalidad, confiabilidad y sencillez de uso.

El proceso conformado por las actividades de acceso, recuperación, extracción y tratamiento dinámico de la información, residente en documentos Excel, se desarrolló y fue puesto en operación de forma satisfactoria.

Se hace mención de que no obstante existir otras aplicaciones, tanto comerciales como de libre acceso, la aquí presentada para la manipulación y recuperación dinámica de información satisface una necesidad específica para la que fue concebida, sin tratar de resolver alguna otra que no fuese parte del requerimiento original.

CAPÍTULO 6. CONCLUSIONES

En este Capítulo se mencionan las conclusiones, las cuales consideran el análisis de resultados y logros obtenidos, así como las contribuciones del presente trabajo de tesis y se plantean los trabajos a futuro para la continuación y extensión de los alcances logrados.

6.1 Análisis de resultados

Los resultados obtenidos están estrechamente ligados a los objetivos planteados al inicio del presente documento, es decir:

- Con respecto al ambiente de captura de datos existente, relacionado con los proyectos de exploración de yacimientos petrolíferos, se realizó el análisis correspondiente y se consideró de manera particular la necesidad de contar, por parte del usuario, con una nueva herramienta integral, dinámica y de fácil manejo para la captura de datos, acorde a la plataforma Excel y que le permitiese de manera ágil la recuperación de la información generada.
- De acuerdo a lo estipulado en los objetivos de la tesis (Capítulo 1), se propuso una solución factible para la consecución de los resultados; se justificó la elaboración de este trabajo y se establecieron sus alcances y beneficios esperados.
- Se presentó el antecedente correspondiente (Capítulo 2), que consideró los aspectos del marco teórico y del estado del arte, en los que se mencionaron los criterios y estructura de los documentos basados en el manejo de formatos Excel y se describieron aspectos fundamentales del Jakarta POI API y su estructura, respectivamente, lo cual permitió conocer el entorno para desarrollar los algoritmos que permitiesen resolver el esquema Excel-Java-Base de datos; así mismo, se presentaron algunas opciones (ideas y trabajos existentes en la materia) tanto comerciales como de libre acceso relacionadas y, se describieron de manera breve algunos casos de éxito basados en el uso del API mencionado.
- Se presentó una solución (Capítulo 3) en la cual se identificaron los requerimientos de la aplicación y los diferentes componentes a desarrollar, se analizaron los diferentes casos de uso inherentes a la operación, así como el entorno relacional complementario para la base de datos de prueba existente. Una vez identificados los requerimientos y elementos participantes, se realizaron las etapas de análisis y diseño, orientadas al desarrollo de una aplicación basada en un patrón de arquitectura de software denominado Modelo-Vista-Controlador (MVC), el cual permite establecer la lógica de negocio y de control, así como de

presentación para el desarrollo respectivo, cuya operación se soporta vía un entorno WEB.

- Para efectos del desarrollo (Capítulo 4), se tomó en cuenta los perfiles de usuario establecidos, por lo que el núcleo de la aplicación estuvo compuesto básicamente de cuatro módulos para el usuario administrador: usuarios, recursos Excel, actualización y, consulta; relacionados éstos directamente con la administración del proceso de: lectura de información de documentos Excel, su extracción y transformación, así como su incorporación y actualización en una base de datos de prueba y, para el usuario básico solo se instrumentó el módulo de recursos Excel en una versión limitada

Se consideró el desarrollo de las interfaces que permitieron la operación de la aplicación; para el acceso se realizó la validación de los usuarios por medio de la interfaz correspondiente, integrada por un formulario en HTML, para captar el identificador y clave de acceso del usuario y de acuerdo a éstos, verificar su perfil y nivel de operación mediante el manejo de la sesión respectiva; así mismo, se generaron las interfaces para navegar por los módulos que integran la aplicación y, otra interfaz para acceder al módulo de registro de recursos Excel por parte del usuario básico; así mismo se integraron las interfaces para la presentación de resultados, incluyendo los enlaces y conexiones respectivos.

- Con respecto a las pruebas y resultados (Capítulo 5) primero se validó el acceso mediante la interfaz de ingreso y se probó que, si el usuario no estaba registrado en la base de datos, no podía acceder a la operación de la aplicación; así mismo, se probó el adecuado manejo de las sesiones, correspondientes a cada perfil de usuario. Mediante el módulo de usuarios se realizaron pruebas de altas, bajas y cambios, incluyendo la definición de su respectivo perfil. Con el módulo de recursos Excel se ingresaron movimientos de altas y bajas y, se realizó la consulta de los archivos cargados al servidor, para este módulo se instrumentó la cuantificación de los tiempos de carga como una forma de evaluar la adecuada operación de dicho módulo. Con respecto a la actualización de las tablas de costos totales, se realizaron pruebas del proceso de: lectura de información de documentos Excel, su extracción y transformación, así como de su actualización en la base de datos de prueba y, también se implementó la medición de los tiempos de actualización respectivos. Finalmente la consulta de la base de datos actualizada se realizó sin ningún problema, mostrándose los datos actualizados.

6.2 Logros alcanzados

Dentro de los logros alcanzados con el presente trabajo y de acuerdo a los requerimientos iniciales se concluye el hecho de que:

- Se estableció una plantilla básica de captura para los rubros relacionados con los proyectos de exploración de yacimientos petrolíferos y se utilizó como esquema para generar la herramienta de captura para archivos Excel; se hace mención de que los datos procesados, son los recursos de prueba generados al utilizar el CAPTURADOR Excel. Dichos datos no están relacionados con proyecto alguno, son totalmente ficticios.
- Los recursos de prueba generados se emplearon como parte del caso de estudio y para probar la operación de la aplicación.
- La definición de los alcances de la aplicación y su funcionalidad, fueron soportados en el análisis y diagramación UML, a través de los casos de uso y sus diagramas, así como de los diagramas de secuencia respectivos. Así mismo, ayudó a identificar los diferentes actores participantes y las diferentes interacciones existentes entre estos.
- Se diseñaron y desarrollaron algoritmos específicos, referidos en el Apéndice F, mediante el uso del POI API, de Java, así como de JSP, que permitieron la interacción Excel-Java-Base de datos
- Se ha logrado una infraestructura para los requerimientos de procesamiento de datos en el entorno del esquema Excel-Java-Bases de datos, soportada en el diseño y desarrollo de los algoritmos correspondientes, en un ámbito de operación integrado, incluido el ambiente WEB generado para tal efecto.
- El uso de la aplicación desarrollada y sus resultados, permite contar con una nueva estrategia para el logro de objetivos comunes: el determinar la viabilidad de nuevos proyectos o la continuación de los ya existentes, en este caso, con base en el análisis de costos totales.
- Apoyar de manera dinámica el proceso de toma de decisiones en el ámbito.
- La operación de la aplicación no está en desventaja con respecto a los procesos que ya pudiesen existir, ya que se basa en el entorno de la base de datos de prueba y su desarrollo está basado en un entorno de fácil escalabilidad.
- Tener otra opción para modelar acciones a bajo costo, con recursos de fácil disponibilidad y acordes a una rápida y adecuada integración con los procesos ya existentes.

Por lo aquí mencionado, se considera que se cumplieron los objetivos planteados en este trabajo de tesis.

6.3 Contribuciones

Las principales contribuciones de este trabajo de tesis, son las siguientes:

- Aplicación que implementa el esquema desarrollado en la presente tesis y que permite:
 - Ejecutar algoritmos para generar elementos de apoyo a los procesos de toma de decisiones y;
 - Adecuarse para ser utilizada en aplicaciones con criterios similares de proceso de datos para diferentes usuarios, gracias a la plataforma en que está desarrollada.
- Opciones de simulación a bajo costo, con recursos de fácil disponibilidad y acordes a una rápida y adecuada integración con los procesos ya existentes.
- Uso de una arquitectura basada en estándares ampliamente reconocidos, como es el patrón de arquitectura de software denominado Modelo-Vista-Controlador (MVC), lo que permite mayor versatilidad en caso de ajustes en la operación y propicia interoperabilidad con otras aplicaciones.
- Algoritmos y mecanismos soportados en la tecnología del Jakarta POI API y, complementados con las características de aplicaciones basadas en tecnología Java y JSP, lo que propicia que pueden ser tomados como punto de referencia.
- Aplicación que por sus características reside en el servidor y su administración y operación es competencia del administrador; el cliente, desde su entorno, simplemente sube información Excel sin que requiera que la base de datos resida en su equipo.
- Integridad y consistencia de datos, ya que, en ningún momento, se permite la posibilidad de modificación fuera de la responsabilidad del administrador de la base de datos.
- La aplicación desarrollada, contribuye a considerar una nueva opción para solucionar requerimientos de manejo y recuperación de información mediante un entorno como el aquí presentado.
- El desarrollo de este trabajo ha permitido establecer un criterio propio para el análisis y diseño de aplicaciones basadas en el esquema Excel-Java-Base de datos, conforme a la siguiente secuencia de actividades:
 1. Conocer y analizar las características de los documentos de tipo Excel y los requerimientos para su manejo.
 2. Ubicar las características de la información a ser procesada y sus

- requerimientos de uso.
3. Determinar las posibles herramientas que permitiesen establecer una adecuada solución, considerando las características de Excel en combinación con las del manejo y presentación dinámica de información.
 4. Presentar y utilizar un esquema de captura homogéneo y único (capturador Excel), basado en la naturaleza de la información y que mejorase los procedimientos existentes de captura.
 5. Elegir, entre las posibles opciones de software libre, la que permitiese aportar una solución adecuada, en este caso la del Jakarta POI API y la tecnología Java
 6. Proponer una solución acorde al análisis realizado, considerando el software para desarrollo previamente elegido.
 7. Desarrollar el entorno de recuperación de información para documentos del tipo Excel mediante el uso del API antes citado.
 8. Establecer los elementos de soporte, es decir: tipo servidor WEB, el entorno de base de datos y los elementos de enlace y conexión con este último.
 9. Establecer un esquema de interacción Excel-Java-Base de datos, soportado en la combinación Jakarta POI API, Java y elementos de enlace.
 10. Programar la parte dinámica de la solución, basada en la tecnología Java y JSP.
 11. Utilizar la información generada por el capturador EXCEL, conjuntamente con la base de datos de prueba, para simular un proceso de actualización real.
 12. Aplicar la propuesta de solución
 13. Evaluar los alcances del desarrollo.

De manera global, las contribuciones aquí mencionadas pueden ser vistas como la creación de una plataforma integradora, la cual permite:

- Manejo de información homologada, de diversas fuentes, de manera expedita.
- Tratamiento y actualización de dicha información.
- Análisis y evaluación colaborativa de los resultados del procesamiento de la información y,
- Acceso a la información, para los niveles participantes en la toma de decisiones, dentro de la organización.

6.4 Trabajo futuro

En base al trabajo realizado se puede plantear la siguiente expectativa:

- Considerar mejoras en la forma del manejo de la concurrencia, esto en coordinación con las políticas que se establezcan al respecto, por parte de las áreas involucradas en la empresa motivo del caso de estudio.

- Probar los alcances de la aplicación en otros entornos operativos y de navegación.
- Tener en cuenta las actualizaciones sucesivas del API utilizado, como una estrategia para mejorar la solución presentada.
- Continuar con la adquisición de nuevo conocimiento, relacionado con el uso de la tecnología Java y otras afines, con lo cual se estará en posibilidad de proponer criterios normativos y de decisión, acordes al entorno tecnológico.
- Sugerir el desarrollo de nuevas aplicaciones, de acuerdo con los alcances y logros obtenidos en este trabajo, como prototipos genéricos, que ayuden a la solución de problemas similares en el entorno empresarial.
- Analizar la pertinencia de migrar desarrollos de este tipo a entornos de mayor complejidad y alcance tecnológico.
- Comparar los alcances de esta aplicación, con entornos desarrollados mediante otras herramientas similares, del tipo no software libre.
- Propiciar el desarrollo de soluciones acordes a las necesidades actuales de las empresas, basadas éstas en el uso de mejores prácticas y, soportadas en la referencia tecnológica de vanguardia.

Resumen

El objetivo general de este trabajo de tesis, enfocado a implementar y presentar una solución acorde al proceso de manipulación y recuperación dinámica de información residente en archivos Excel, se ha logrado con la instrumentación de la aplicación aquí desarrollada, basada ésta en el uso de tecnologías como el Jakarta POI API, en combinación con Java e interfaces gráficas desarrolladas mediante JSP, integrando la operación de la misma en un entorno WEB.

Las conclusiones aquí descritas consolidan la expectativa presentada a través de este trabajo, pero son el punto de partida para mejorar los alcances del mismo.

BIBLIOGRAFÍA.

1. Deitel, Harvey M. Java: how to program / H.M. Deitel, P. J. Deitel -- 2a.Ed. -- Upper Saddle River, NJ : Prentice Hall 1998
2. Falkner, Jayson [et al]. Fundamentos de Desarrollo WEB con JSP. Editorial. Anaya Multimedia. 2002
3. Hall, Marty, More servlets and JavaServer pages / Marty Hall -- Upper Saddle River, NJ: Prentice Hall 2002
4. Jacobson, Ivan [et al]. El proceso unificado de desarrollo de software. Addison Wesley. 1999
5. James, Martin-1933-Computer data-base organization / James Martin -- 2a.Ed. -- Englewood Cliffs, N.J. : Prentice-Hall 1977
6. Kurniawan, Budi.How tomcat works: a guide to developing your own Java Servlet Container / Budi Kurniawan and Paul Deck. -- 2004
7. López, José. Domine JavaScript. 2a Ed. AlfaOmega. 2007.
8. Metzger, Philip. Administración de un proyecto de programación. Trillas 1978.
9. Orfali, Robert. Client / server survival guide / Robert Orfali, Dan Harkey, Jeri Edwards -- 3a.Ed. -- New York : John Wiley 1999
10. Rodríguez De.La.Fuente, Santiago [et al] Programación de aplicaciones WEB. Thomson.2003
11. Schuller, Joseph. Aprendiendo UML en 24 horas. Prentice Hall. 2000
12. Schneider, Robert D., MySQL database design and tuning / Robert D. Schneider. -- Indianapolis, Ind : MySQL Press 2005
13. Tremblett, Paul. Superutilidades para JavaServerPages. Editorial: McGraw-Hill. ISBN: 9701038282. Idioma: Español. Edición No. 1. Publicado: 2002.

Referencias Electrónicas

14. Acceso a Bases de Datos [JDBC]. Autor: Sun. Traductor: Juan Antonio Palos (Ozito). URL: <http://www.programacion.com/tutorial/jdbc/>

15. Apache Org. URL: <http://httpd.apache.org/docs/2.0/es/>
16. Apache Tomcat, URL: <http://tomcat.apache.org/>
17. API JExcel. URL: <http://www.teamdev.com/jexcel/index.jsf>
18. API Java Excel. URL: <http://www.andykhan.com/jexcelapi/index.html>
19. Connector J. URL: www.desarrolloweb.com/articulos/901.php
20. Excel. The Excel Document Model.
URL: <http://www.devx.com/Java/Article/17301>
21. Design Patterns: Model-View-Controller
URL: <http://java.sun.com/blueprints/patterns/MVC.html>
22. Java libraries to read/write Excel (.XLS) document files
URL: [http:// Schmidt.devlib.org/java/libraries-excel.html](http://Schmidt.devlib.org/java/libraries-excel.html)
23. JDBC, URL: <http://java.sun.com/products/jdbc/>
24. JSP, URL: <http://java.sun.com/products/jsp/>
25. Macros en Excel, Ramón Mendoza, URL:
<http://www.lawebdelprogramador.com/cursos/excel/index.php>
26. Manual MYSQL, URL: <http://www.webestilo.com/mysql/>
27. POI (Poor Obfuscation Implementation) Jakarta POI project.
<http://jakarta.apache.org/poi/>.
28. Rentz, Daniel [et al], OpenOffice.org's Documentation of the Microsoft Excel File Format. Excel Versions 2, 3, 4, 5, 95, 97, 2000, XP, 2003.
<http://sc.openoffice.org/excelfileformat.pdf>
29. Scherba, Julia, De Valenzuela. Ph.D.
URL: <http://www.unm.edu/~devalenz/handouts/MARCO.html>
30. Universidad de Antioquia. Seminario Estudios de Usuarios. Escuela Interamericana de Bibliotecología. Unidad 4. Organización y elaboración de proyectos de estudios de usuarios.
http://docencia.udea.edu.co/bibliotecologia/seminario-estudios-usuario/unidad4/estado_arte.html
31. Upload. FileUpload. URL: <http://commons.apache.org/fileupload/using.html>

APÉNDICES

Apéndice A. Código para generar un documento tipo Excel.

Ejemplo de código generado, el cual utiliza el POI API y sus bibliotecas de clases HSSF y POIFS.

Figura A.1 Uso del JAKARTA POI API de APACHE	
<pre> /* * Main.java * Created on 28 de junio de 2008, 09:19 PM * To change this template, choose Tools Template Manager * and open the template in the editor. */ package excread; import java.io.*; import org.apache.poi.hssf.record.formula.functions.*; import org.apache.poi.poifs.filesystem.*; import org.apache.poi.hssf.usermodel.*; import org.apache.poi.hssf.usermodel.HSSFCellStyle; import org.apache.poi.hssf.util.*; import org.apache.poi.hssf.util.HSSFColor; /** * @author */ public class Main { /** Se crea una nueva instancia de Main */ public Main() { } /** </pre>	

```

    * @param args the command line arguments

    */

    public static void main(String[] args) throws
    FileNotFoundException, IOException {

        /* Para crear un documento Excel se debe crear un nuevo
        libro: Se crea un nuevo libro */

        HSSFWorkbook workBook = new HSSFWorkbook();

        /* De esta manera podemos crear una nueva hoja dentro del
        libro: Se crea una nueva hoja llamada HOJA1 */

        HSSFSheet sheet = workBook.createSheet("HOJA1");

        /* Al llamar al método createRow, del objeto hoja podemos
        crear los renglones. No se debe olvidar que el índice de los
        renglones empieza en la posición cero(0). */

        /* ** El encabezado ** */

        /* Se crea un renglón de encabezado */

        HSSFRow headerRow = sheet.createRow((short) 0);

        /*Para crear el área de encabezado, se define la región entre
        el primer y tercer renglones; y de la primera a la decimal columna:
        Se define el área para el encabezado(renglón 1 al 3, Columna 1 a la
        10)*/

        sheet.addMergedRegion(new Region(0, (short) 0, 2, (short)
        10));

        /* Ahora se crea una celda para el dato a capturar.
        Se crea la celda para la información del encabezado */

        HSSFCell headerCell =

            headerRow.createCell((short) 0);

        /* Se añade el texto a esta celda */

        headerCell.setCellValue("PUNTUACIÓN DEL TORNEO");

        /* El texto indicado abarcará la región definida para el
        encabezado */

        /* Ya se ha mencionado que las celdas pueden tener
        características especiales, una de ellas es el estilo que se les
        pueda dar; mediante HSSF y el objeto HSSFCellStyle se puede hacer,
        un ejemplo de esto es el siguiente código */
    }

```

```
/* Se crea un estilo para la celda de encabezado */

        HSSFCellStyle headerStyle =
workBook.createCellStyle();

headerStyle.setAlignment(HSSFCellStyle.ALIGN_CENTER);

        headerCell.setCellStyle(headerStyle);

/* Las hojas, como ya se mencionó pueden contener tantos
renglones y celdas como se requieran, aplicarles estilos y
asignarles valores, esto último se logra empleando el método
setCellValue(). */

/* ** Entorno para el reporte de datos ** */

/* Se crean 6 renglones que se usaran para los contenidos de
las celdas así como el Total */

        HSSFRow row1 = sheet.createRow((short) 5);

        HSSFRow row2 = sheet.createRow((short) 6);

        HSSFRow row3 = sheet.createRow((short) 7);

        HSSFRow row4 = sheet.createRow((short) 8);

        HSSFRow row5 = sheet.createRow((short) 9);

        HSSFRow row6 = sheet.createRow((short) 10);

/* Después se crean 2 celdas para cada renglón y una más
para el texto del total*/

        HSSFCell c11 = row1.createCell((short) 0);

        HSSFCell c12 = row1.createCell((short) 4);

        HSSFCell c21 = row2.createCell((short) 0);

        HSSFCell c22 = row2.createCell((short) 4);

        HSSFCell c31 = row3.createCell((short) 0);

        HSSFCell c32 = row3.createCell((short) 4);

        HSSFCell c41 = row4.createCell((short) 0);

        HSSFCell c42 = row4.createCell((short) 4);

        HSSFCell c51 = row5.createCell((short) 0);
```

```

        HSSFCell c52 = row5.createCell((short) 4);

        HSSFCell c61 = row6.createCell((short) 0);

        HSSFCell c62 = row6.createCell((short) 3);

        HSSFCell total = row6.createCell((short) 4);

        /* Se agrega información a las celdas */

        c11.setCellValue("Nombre");

        c12.setCellValue("Puntos");

        c21.setCellValue("Óscar");

        c22.setCellValue(100);

        c31.setCellValue("Rubén");

        c32.setCellValue(200);

        c41.setCellValue("César");

        c42.setCellValue(350);

        c51.setCellValue("Pedro");

        c52.setCellValue(425);

        c62.setCellValue("Total: ");

        /* Una de las características de Excel es que las celdas
        pueden contener referencias a fórmulas matemáticas. El siguiente
        ejemplo asigna una celda para calcular un Total, para lo cual se le
        debe dar el tipo FORMULA. Se crea una fórmula para el cálculo del
        total */

        total.setCellType(HSSFCell.CELL_TYPE_FORMULA);

        total.setCellFormula("SUM(E6:E10)");

        /* Cuando el libro está completo, mediante el objeto
        FileOutputStream se puede guardar como un archivo y desplegar el
        contenido usando el método write */

        FileOutputStream stream = new

        FileOutputStream("C:\\Documents and
        Settings\\cic\\Mis documentos\\Libro3.xls");

        workbook.write(stream);

    } }

```


Apéndice B. Manual de uso del módulo de captura para proyectos de exploración de yacimientos petrolíferos

En este apartado se identifican los elementos que participan en un esquema de captura, en el contexto del desarrollo del presente trabajo. Se menciona el antecedente sobre el por qué se propone una nueva herramienta, su elaboración como parte de la solución, sus alcances y su entorno de operación; así como la manera de utilizarla y sus resultados.

B.1. Antecedentes.

Parte de la problemática a solucionar, en primera instancia, fue la de analizar las herramientas existentes para la captura de datos. Se encontró que ya existía en Excel (Figura B.1), una que incluía varias hojas con formatos e información diversa; no obstante, no emitía datos acumulados por cada hoja y más aún, no se tenía información global del libro, relacionada con rubros como los costos, entre otros.

	A	B	C	D	E	F	G	H
	Descripción	Clave de la Región a la que pertenece el proyecto. (Consignar el número asignado por MODEX en el combo Región)	Clave del Activo al que pertenece el proyecto. (Consignar el número asignado por MODEX en el combo Activo)	Clave del Proyecto Exploratorio. (Consignar el número asignado por MODEX en el combo Proyecto)	Clave del Proyecto Conceptual (Consignar el número de acuerdo a la tabla 1 del instructivo)	Nombre del proyecto integrado (Nombre asignado por el usuario al evaluar en MODEX)	Fecha de llenado del formato	Fecha de autorizació del proyec por la SHC
1								
2	Tipo de dato	NUMÉRICO	NUMÉRICO	NUMÉRICO	TEXTO	TEXTO	FECHA	FECHA
3	Formato	Entero	Entero	Entero	5 caracteres	100 caracteres	dd/mm/yyyy	dd/mm/yy
4	Restricción	Valor de catálogo	Valor de catálogo	No nula	No nula	No nula		
	Datos	1	1	1	402	CAMPECHE ORIENTE	07/11/2006	13/09/2006
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

Figura B.1. Plantilla original Excel de captura

Con base en el análisis anterior, en los nuevos requerimientos y en los esquemas operativos del usuario, se consideró el siguiente:

B.2 Objetivo

Proporcionar una herramienta basada en la utilización de hojas de cálculo Excel y soportada por un ambiente dinámico, estructurado y modular desarrollado mediante Visual Basic para Aplicaciones (VBA; macros), con un esquema claro y amigable de las opciones que contiene, agrupadas por categorías con sus respectivas variables, que permita un entorno operativo más accesible y versatilidad en su manejo; y que aporte datos globales de las variables cuantificables, relacionadas con la inversión operacional, con los proyectos de sísmica, así como con los estudios de campo y de gabinete, de los proyectos de exploración de yacimientos petrolíferos.

B.3. Entorno de Operación

Para usar el Módulo de Captura para Proyectos de Exploración, se requiere:

- Herramientas para el manejo de hojas de cálculo, en este caso Excel.
- Se utilizó Excel 2003 (Service Pack 3) para el desarrollo de las macros que integran el Módulo de Captura.

B.4. Uso del capturador Excel

El Módulo de Captura para Proyectos de Exploración (CAPTURADOR), está conformado por las siguientes categorías:

- Inversión Operacional
- Sísmica 2D y 3D
- Estudios de Campo
- Estudios de Gabinete

y de acuerdo a éstas, se establecen criterios de identificación y uso para su operación.

Añade para una mejor identificación de cada categoría y sus respectivas hojas un color específico, así como prefijos de identificación adicionales a los nombres de las pestañas e incorpora un ordenamiento por prefijo y ubicación.

Los prefijos para cada pestaña indican a qué categoría corresponde y van asociados al título de la hoja mostrado en la esquina superior izquierda; de esta forma el Módulo de Captura para Proyectos de Exploración ordena de manera alfabética las pestañas de las hojas, con base en los prefijos establecidos y el nombre asociado; evita así posibles “colisiones” con los nombres de las pestañas y permite la rápida localización e identificación de las mismas.

En la Figura B.2 se aprecian las opciones que conforman al Módulo de Captura para Proyectos de Exploración con sus correspondientes vertientes.



Figura B.2. Menú del módulo de captura para proyectos de exploración

De forma adicional al Menú mostrado en dicha figura, en la parte baja se muestran las pestañas con los nombres de las hojas restantes que conforman el libro y que se refieren a:

- Generalidades
- Información Clave
- Antecedentes-Metas-Solicitantes
- Metas Físicas Adicionales
- Oportunidades
- Total

Así mismo, en la parte superior de las opciones del Menú, se indica el rango de años correspondiente al Período de Evaluación.

El intervalo de años no podrá ser modificado en Inversión Operacional; no así en Sísmica y Estudios de Campo y de Gabinete, en los cuales sí podrá modificarse siempre que no rebase el período preestablecido antes mencionado.

Se incorpora también un elemento muy importante que es el botón para Recalcular Totales, mediante el que se permite obtener totales por categoría; para los casos de Inversión Operacional, Estudios de Campo y Estudios de Gabinete dichos totales se calculan en unidades o millones de pesos; y para los

casos de Sísmica 2D y Sísmica 3D son en kilómetros y en kilómetros cuadrados respectivamente. En caso de capturas nuevas o modificación de las existentes, es necesario oprimir dicho botón para que se actualicen los totales.

Es relevante mencionar que también con la pestaña de Total se tiene la posibilidad de hacer cálculos por cada una de las categorías; ya sea en forma parcial y de acuerdo a los años que se hayan capturado o por todo el período.

Por lo anterior, se considera la importancia de proteger todas las celdas de la página de Totales; en la Figura B.3 se aprecia la hoja correspondiente.

	2026	2027	2028	2029	2030	2031	2032	2033	2034	Total
Totales										
Sísmica 2D										
Kilómetros	0	0	0	0	0	0	0	0	0	0
Millones de pesos	0	0	0	0	0	0	0	500	0	600
Sísmica 3D										
Kilómetros cuadrados	0	0	0	0	0	0	0	0	0	0
Millones de pesos	0	0	0	0	0	0	0	0	0	0
Total de inversión	0	0	0	0	0	0	0	500	0	750
Estudios de Campo										
Número	0	0	0	0	0	0	0	10	0	1140
Millones de pesos	0	0	0	0	0	0	0	120	0	400
Estudios de Gabinete										
Número	0	0	0	0	0	0	0	0	0	800
Millones de pesos	0	0	0	0	0	0	0	0	0	510
Total de estudios	0	0	0	0	0	0	0	10	0	1940
Total de millones de pesos	0	0	0	0	0	0	0	120	0	990
Inversión Operacional.										
Número	0	0	0	0	0	0	0	2690	0	3485
Millones de pesos	0	0	0	0	0	0	0	3060	0	3655

Figura B.3. Totales del módulo de captura para proyectos de exploración

La Figura B.4 presenta la categoría de Inversión Operacional, que al igual que Sísmica 2D, Sísmica 3D, Estudios de Gabinete y Estudios de Campo, tienen sus propias opciones.

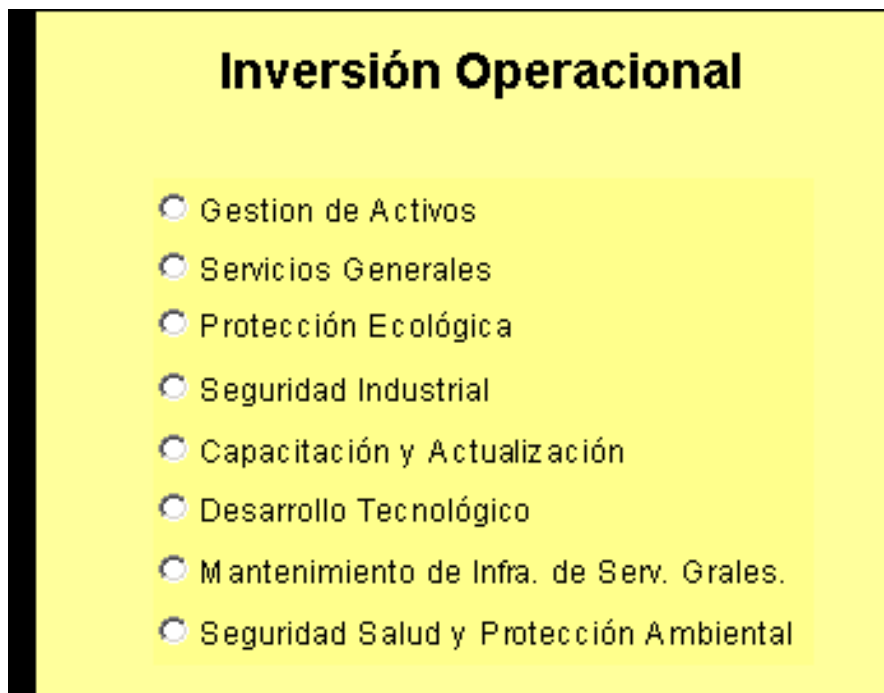


Figura B.4. Opciones de Inversión Operacional

Es conveniente mencionar que para poder capturar una categoría que no sea de Inversión Operacional es **obligatorio que el libro contenga al menos una hoja** de alguna de las opciones arriba presentadas; y se enfatiza en que por cada opción **sólo podrá activarse dicha hoja por una única vez**.

Para lo anterior, se elige la opción deseada y al dar click sobre la misma el Módulo de Captura para Proyectos de Exploración activa la hoja respectiva, como se muestra en la figura B.5 en este caso para Servicios Generales.

Ahí se indican el título, los proyectos relacionados, los rubros (objetos) que contiene, así como el área de captura que es la región rectangular en color verde claro (rango de celdas), que en este caso va del año 2008 hasta el año 2034; se hace mención que para dicha área se incluye la validación correspondiente que evita al usuario el poder asentar datos que no sean numéricos.

En la parte de total de millones de pesos acumula los montos correspondientes a cada año, relacionados con los objetos que incluyan importes.

Dicha hoja, como todas las que conforman el libro del Módulo de Captura para Proyectos de Exploración, tiene protegidos los paneles izquierdo y superior, así como los totales; y permite el recorrido en forma horizontal y vertical (mediante el desplazamiento o scroll) sin perder de vista los paneles antes mencionados (Títulos y Objetos).

A	B	C	D	E
Servicios Generales	Clave del Proyecto Exploratorio	Clave del Proyecto Conceptual	Nombre del Proyecto Integrado	
	202	505	Tabasco3	
Objeto	2008	2009	2010	2011
Casas y Edificios Habitacionales (número)				
Casas y Edificios Habitacionales (millones de pesos)				
Alojamiento Temporal (número)				
Alojamiento Temporal (millones de pesos)				
Edificios Administrativos (número)				
Edificios Administrativos (millones de pesos)				
Almacenes, Talleres y Laboratorios (número)				
Almacenes, Talleres y Laboratorios (millones de pesos)				
Cintoteca Regional de Exploración (número)				
Cintoteca Regional de Exploración (millones de pesos)				
Nucleoteca Regional (número)				
Nucleoteca Regional (millones de pesos)				
Casetas (número)				
Casetas (millones de pesos)				
Equipos de Comunicación (número)				
Equipos de Comunicación (millones de pesos)				
Otros Servicios Auxiliares (millones de pesos)				
Estudios (número)				
Estudios (millones de pesos)				
Otros Construcción para Servicios Auxiliares (millones de pesos)				
Total de Millones de Pesos	0.00	0.00	0.00	0.00
<input type="button" value="Ir al menú"/>				

Figura B.5 Hoja de captura correspondiente a Servicios Generales

Se observa que la pestaña correspondiente a Servicios Generales se compone del prefijo INV_Servicios Generales, la cual corresponde al título de la hoja y con el color de la opción, en este caso Inversión Operacional (color amarillo pálido).

Las demás pestañas proporcionan información de referencia, tanto para la persona que en ese momento utiliza el Módulo de Captura para Proyectos de Exploración como para que éste pueda incorporar de forma automática, datos de los proyectos, como parte de su entorno de operación.

En la Figura B.6, se aprecia como el Módulo de Captura para Proyectos de Exploración acumula, en este caso, por renglón y por columna los montos correspondientes a los objetos que manejan importes; y considera también el total de millones de pesos, por año y por objeto; y emite los totales para cada objeto.

Conviene aclarar que en las demás opciones del Módulo de Captura para Proyectos de Exploración, Sísmica y Estudios, no necesariamente se hará el acumulado del total de millones de pesos por cada año, esto depende de cada opción.

Objeto	2032	2033	2034	Total
Casas y Edificios Habitacionales (número)				0.00
Casas y Edificios Habitacionales (millones de pesos)		1.00	8.00	9.00
Alojamiento Temporal (número)				0.00
Alojamiento Temporal (millones de pesos)		1.00	9.00	10.00
Edificios Administrativos (número)				0.00
Edificios Administrativos (millones de pesos)		1.00	10.00	11.00
Almacenes, Talleres y Laboratorios (número)				0.00
Almacenes, Talleres y Laboratorios (millones de pesos)		1.00	9.00	10.00
Cintoteca Regional de Exploración (número)				0.00
Cintoteca Regional de Exploración (millones de pesos)		1.00	8.00	9.00
Nucleoteca Regional (número)				0.00
Nucleoteca Regional (millones de pesos)		1.00	7.00	8.00
Casetas (número)				0.00
Casetas (millones de pesos)		1.00	6.00	7.00
Equipos de Comunicación (número)				0.00
Equipos de Comunicación (millones de pesos)		1.00	5.00	6.00
Otros Servicios Auxiliares (millones de pesos)		1.00	4.00	5.00
Estudios (número)				0.00
Estudios (millones de pesos)		1.00	3.00	4.00
Otros Construcción para Servicios Auxiliares (millones de pesos)		1.00	2.00	3.00
Total de Millones de Pesos por Año	0.00	11.00	71.00	

Figura B.6. Parte final de la hoja correspondiente a Servicios Generales

Al elegir una categoría diferente de Inversión Operacional, es decir, Sísmica 2D, Sísmica 3D, Estudios de Campo o Estudios de Gabinete, se presentará, antes de desplegar la hoja, una pantalla adicional como la mostrada en la Figura B.7.

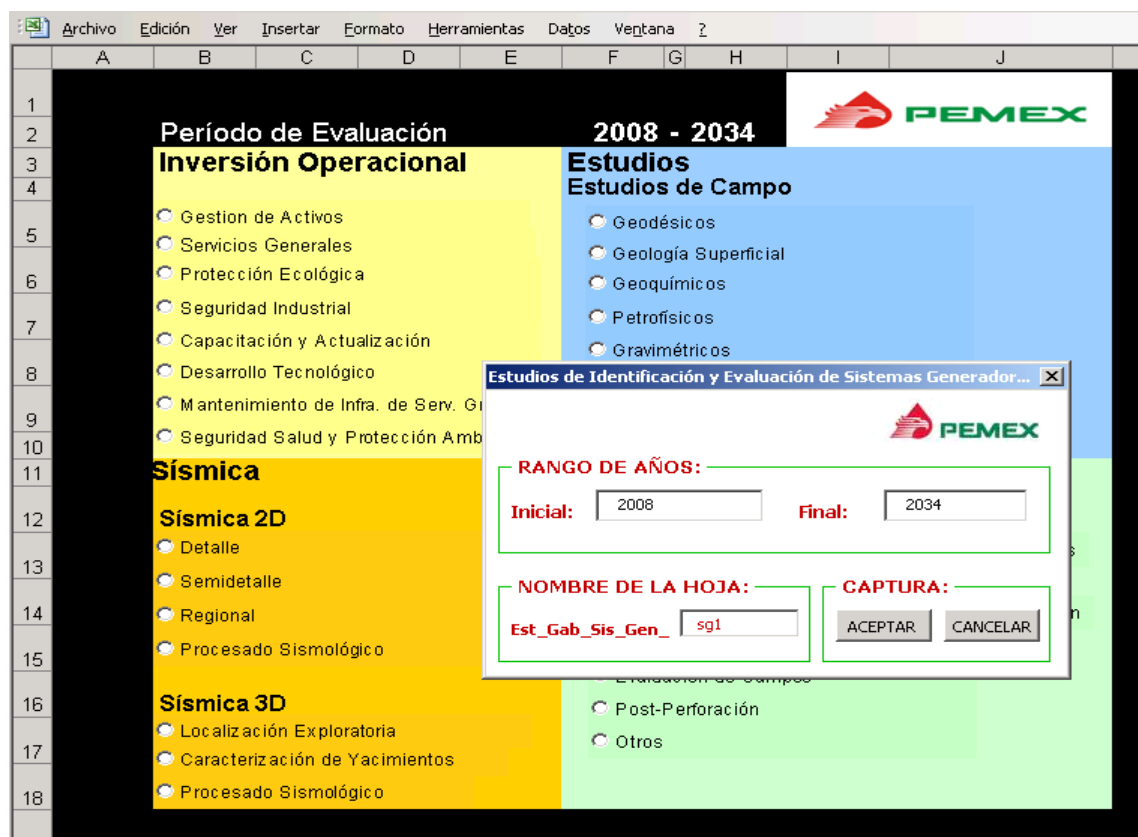


Figura B.7. Pantalla de captura de años y nombre para la opción elegida.

En dicha pantalla y en este caso para Estudios de Identificación y Evaluación de Sistemas Generadores, se pide indicar el rango de años o conservar el que aparece como propuesto; en caso de que se modifique el rango de años, éste deberá ser un subconjunto del período propuesto por el formulario; así mismo, se debe indicar el nombre de la hoja, en este caso es 'sg1', el que estará asociado al prefijo respectivo.

En la Figura B.8 se observa que el nombre de la pestaña incluye los caracteres 'sg1' y no se incluye un renglón de total de millones por año.

Así mismo, en la Figura B.9 se aprecia la suma que corresponde a cada objeto.

Hasta aquí se detalla el uso de este Módulo de Captura para Proyectos de Exploración. Para las otras categorías, se presentan las hojas respectivas de acuerdo a cada instancia y se tiene un tratamiento similar conforme a lo ya descrito.

	A	B	C	D	E
1	Estudios de Identificación y Evaluación de Sistemas Generadores	Clave del Proyecto Exploratorio	Clave del Proyecto Conceptual	Nombre del Proyecto Integrado	
2		25	402	Tabasco	
3		Objeto	2008	2009	
4	Estudios de Identificación y Evaluación de Sistemas Gene	1	2		
5	Estudios de Identificación y Evaluación de Sistemas Gene	1	2		
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					

Figura B.8. Hoja de Estudios de Identificación y Evaluación de Sistemas Generadores

	A	Y	Z	AA	AB	AC	AD	AE	AF
1	Estudios de Identificación y Evaluación de Sistemas Generadores								
2									
3		Objeto	2031	2032	2033	2034	Total		
4	Estudios de Identificación y Evaluación de Sistemas Gene				3	6			
5	Estudios de Identificación y Evaluación de Sistemas Gene				3	6			
6									
7									
8									
9									
10									

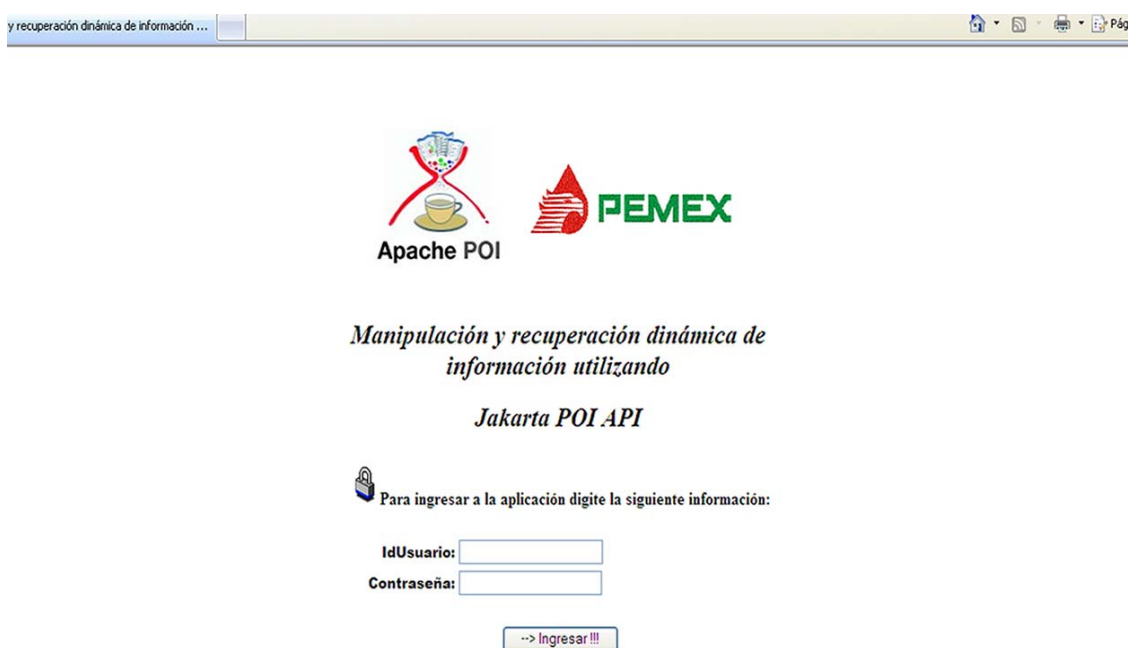
Figura B.9. Se aprecian los totales por cada objeto para la opción elegida

página en blanco

Apéndice C Manual de uso de la aplicación para la manipulación y recuperación dinámica de información utilizando el Jakarta POI API

En este apéndice se muestra el funcionamiento de la aplicación y cómo sería utilizado por un usuario administrador.

Para acceder al funcionamiento se pide indicar el identificador (LOGIN) y su correspondiente contraseña (PASSWORD), Figura C.1,



The screenshot shows a web browser window with the title "y recuperación dinámica de información...". The page features the logos for "Apache POI" and "PEMEX". Below the logos, the text reads: "Manipulación y recuperación dinámica de información utilizando Jakarta POI API". A lock icon is followed by the instruction: "Para ingresar a la aplicación digite la siguiente información:". There are two input fields: "IdUsuario:" and "Contraseña:". Below these fields is a button labeled "--> Ingresar !!!".

Figura C.1. Inicio de sesión de usuario

Para un usuario básico también se requieren los datos anteriores y solo puede acceder al módulo de Registro de recursos Excel (en su versión limitada), lo que se aprecia en la Figura C.2, pero sin la posibilidad de eliminar recursos ya registrados.

APÉNDICE C. MANUAL DE USO DEL SISTEMA DE MANIPULACIÓN

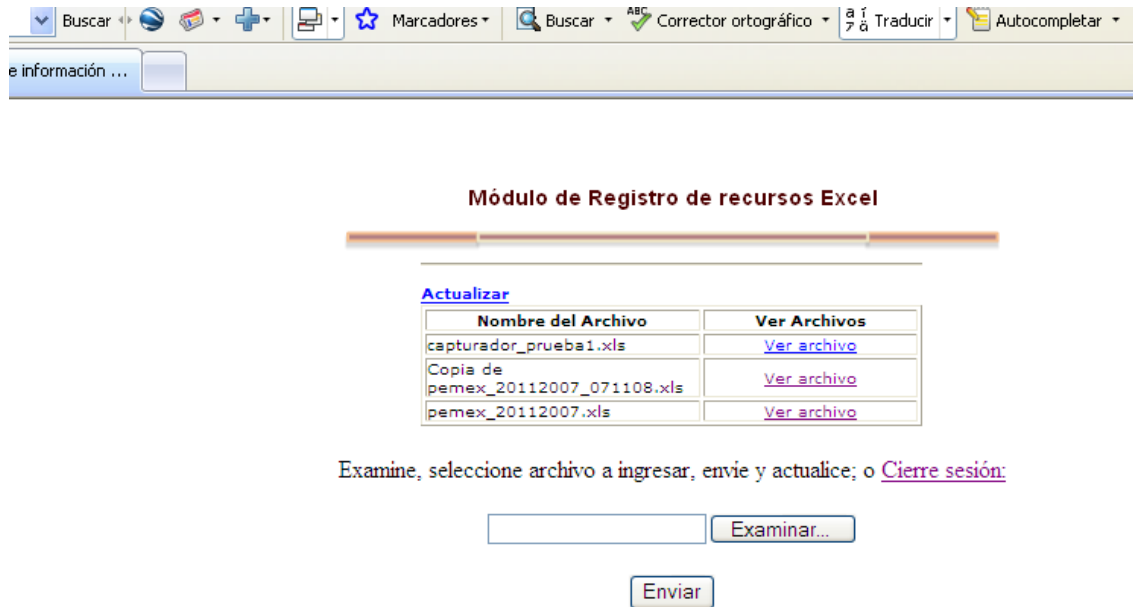


Figura C.2. Módulo de Recursos Excel (versión limitada)

Como se puede apreciar, la interfaz gráfica para este usuario no cuenta con el icono para eliminar registros.

Opciones para el Usuario Administrador

En el caso del usuario administrador, se tiene una aplicación que ofrece variedad de posibilidades, entre las que se encuentran los siguientes módulos:

- Usuarios
- Recursos
- Actualización
- Consulta

Para acceder a esta forma de operación, se requiere que un usuario administrador sea previamente registrado en la base de datos y validados su identificador y contraseña respectivos.

Ya que el usuario ha sido validado por la aplicación, puede ingresar al ambiente de operación, en donde podrá realizar acciones de ingreso, actualización y consulta, véase la Figura C.3.

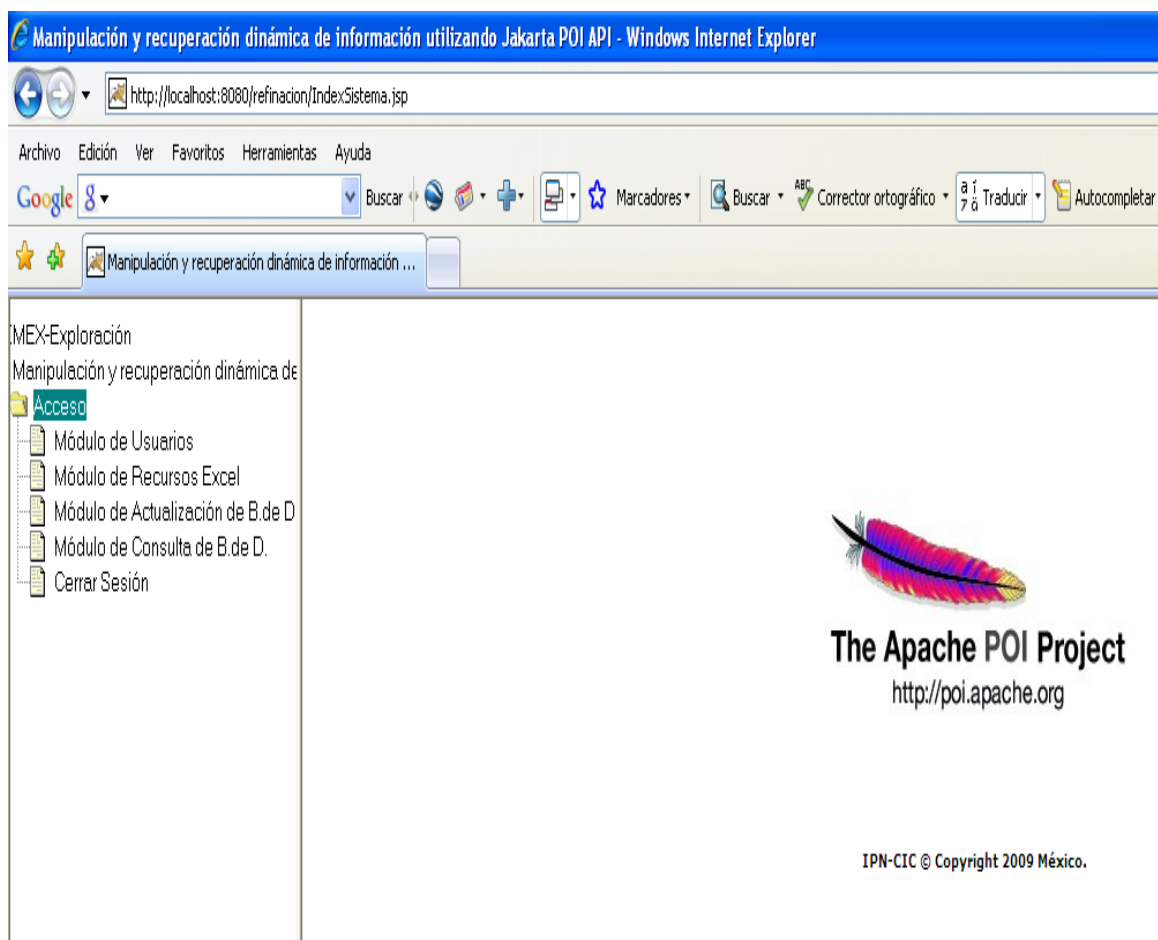


Figura C.3 Entorno de operación de la aplicación

Módulo de Recursos Excel

En esta sección, el usuario puede seleccionar la opciones de

- Alta
 - Examinar
 - Seleccionar
 - Enviar
 - Actualizar
- Baja
 - Actualizar

La Figura C.4 muestra la interfaz correspondiente y el formulario con las opciones ya mencionadas; mediante el que se le proporciona al usuario la interacción con este módulo.

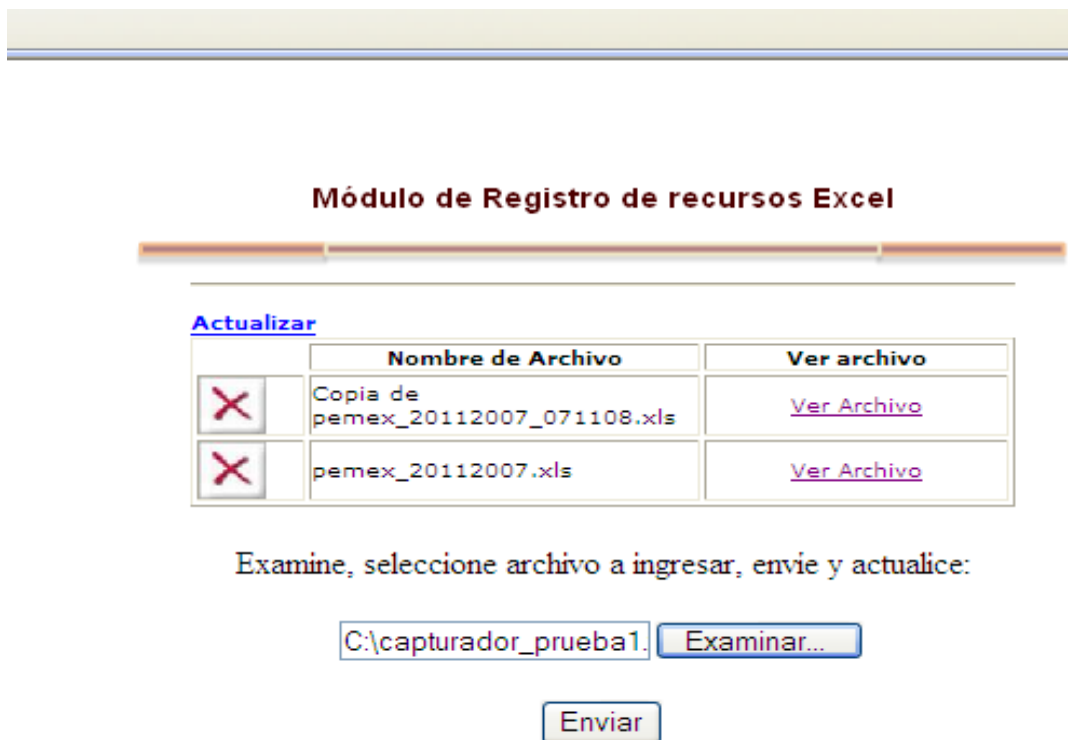


Figura C.4. Interfaz del módulo de recursos Excel

El formulario presenta un elemento de “Examinar” que es donde se busca la ubicación física del recurso en la computadora, se elige de acuerdo al tipo de documento Excel y se envía para activar el enlace con la base de datos y actualizar ésta. Para ello basta que el usuario dé click en el botón “actualizar”. En la opción de ver archivo se permite visualizar el libro correspondiente.

Para la opción de baja, simplemente acciona el icono respectivo, la aplicación informa que se dará de baja el registro y pide la confirmación correspondiente. Una vez aceptada la acción, se emite el aviso de la baja efectuada y con el botón “actualizar” se elimina la referencia del registro en el formulario.

Módulo de Usuarios

Para registrar un nuevo usuario, modificar su estatus o eliminar su registro, se debe elegir del menú principal esta opción.

En ella se muestra la interfaz donde será necesario escribir los datos correspondientes que solicita el formulario y al finalizar hacer click en el botón de actualizar, véase Figura C.5.

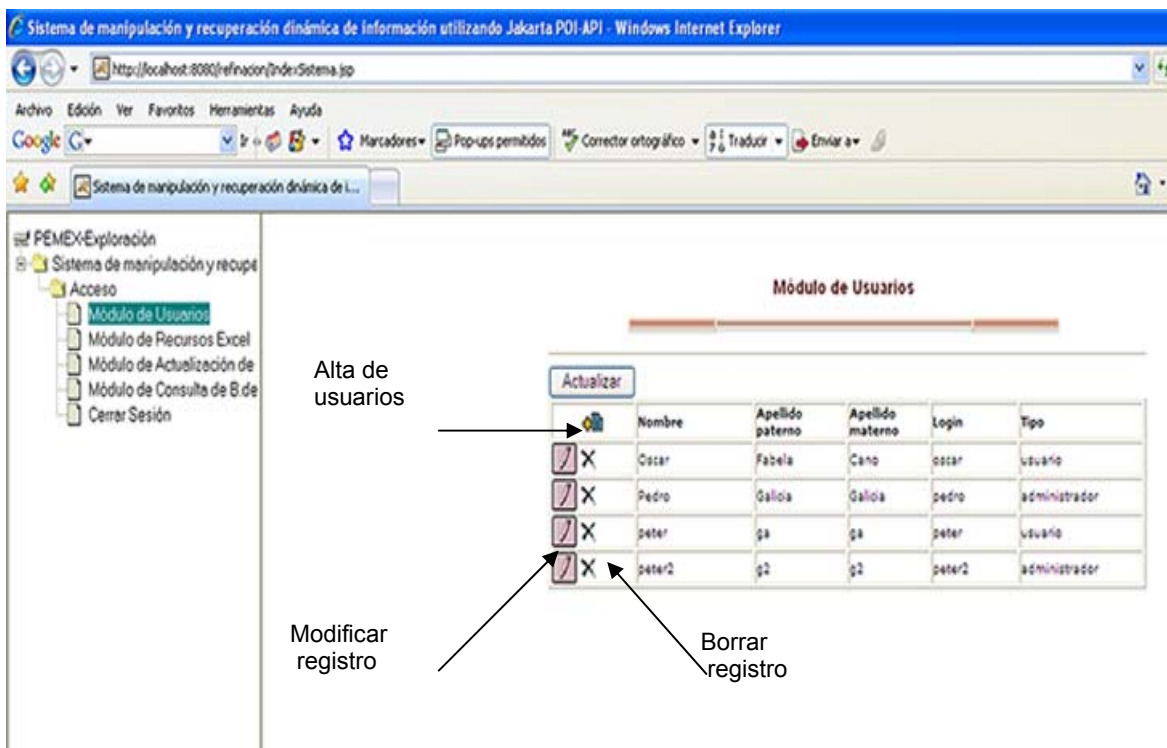


Figura C.5 Interfaz del Módulo de Usuarios

Alta, baja o cambios de registro de usuario.

Para trabajar con esta opción, el usuario administrador debe seleccionar Alta de usuarios (icono respectivo), le presentará el formulario respectivo, Figura C.6, en el que se asientan los datos solicitados y una vez terminado, basta dar click en el botón de actualizar para finalizar la operación.

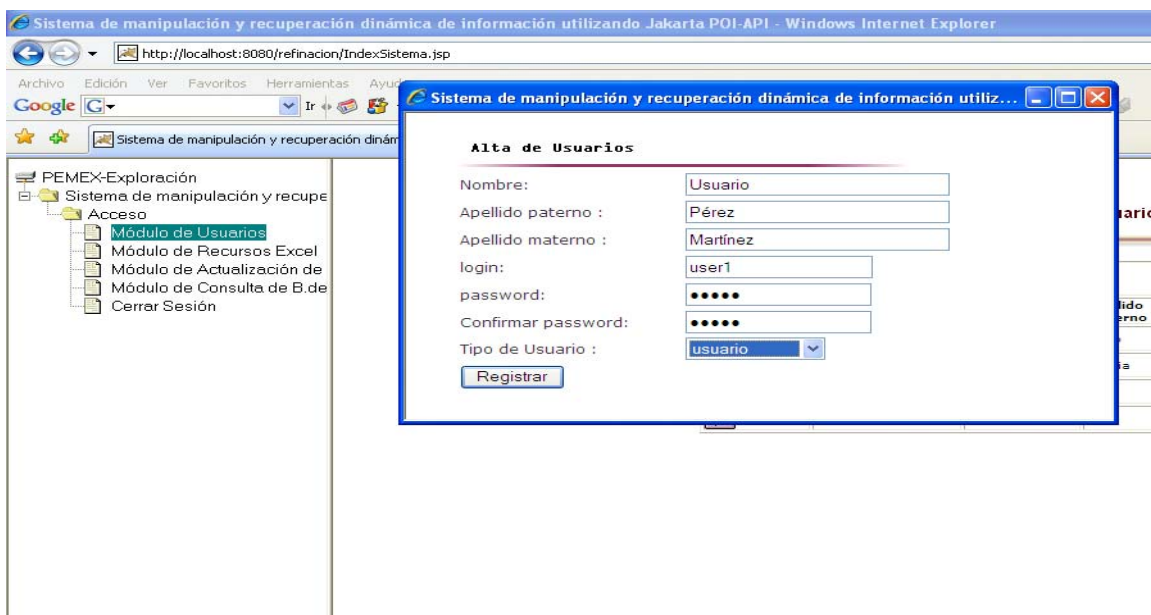


Figura C.6 Alta de usuario

Proceso similar es para la opción de cambios o modificaciones, Figura C.7;

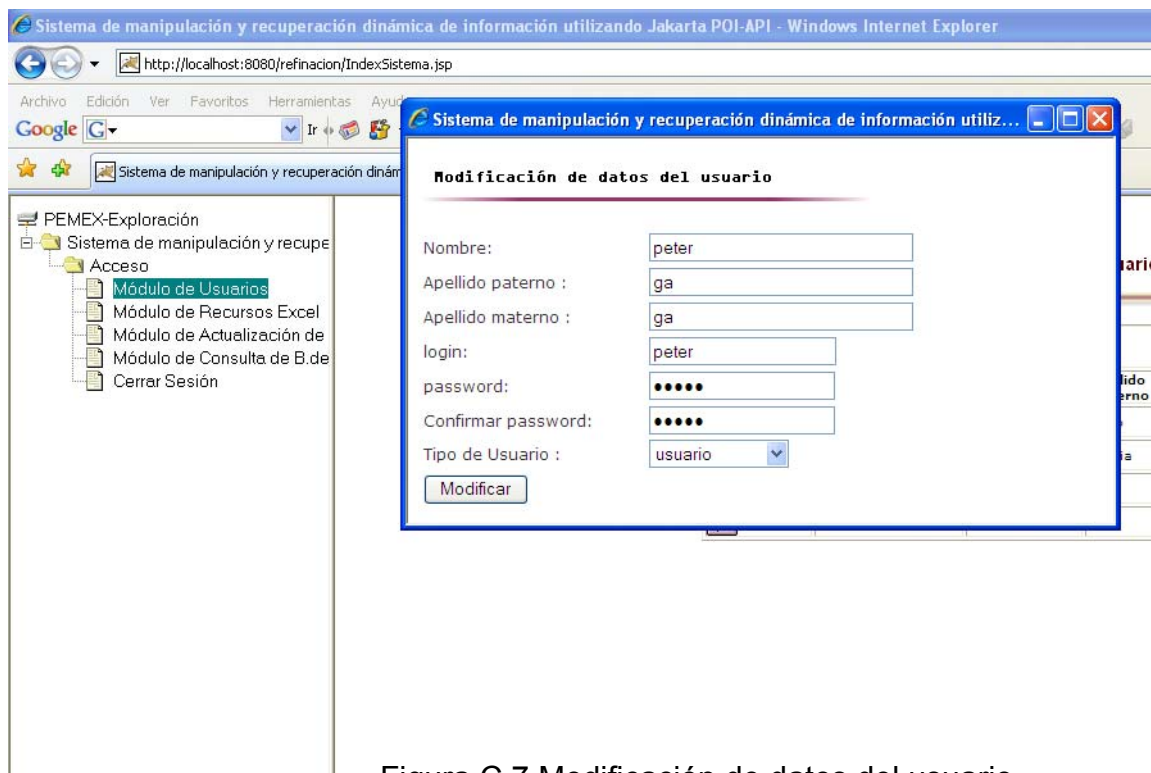


Figura C.7 Modificación de datos del usuario

y en el caso de baja (Figura C.8) la variante es que indicará que el registro será borrado.

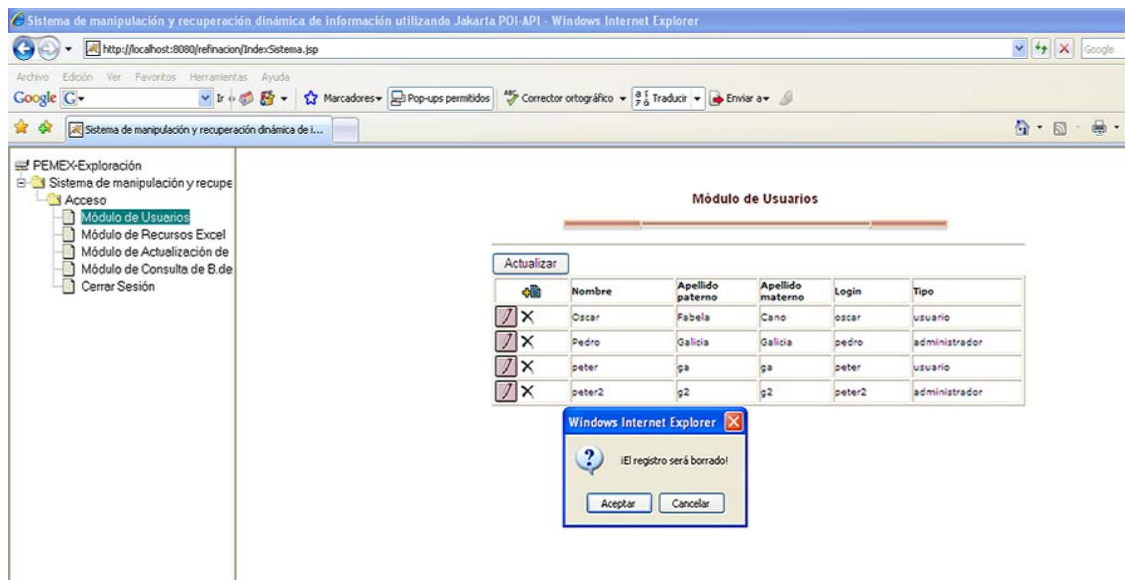


Figura C.8 Eliminación del registro del usuario

De origen la interfaz mostrará los registros existentes antes de cualquiera de las opciones elegidas en este módulo.

Módulo de Actualización

Este módulo permite ejecutar, mediante las interfaces respectivas, el proceso de lectura y transformación de la información residente en los libros Excel (generados por el CAPTURADOR), su traslado y actualización en la base de datos de prueba. Las actividades que un usuario administrador debe realizar para tal efecto, se detallan a continuación.

Para realizar una prueba de este módulo, se eligió uno de los recursos Excel incorporados y se realizó la lectura y selección de los datos; lo que se muestra, en la Figura C.9.

Proceso de importación de valores del capturador

Fecha: <input type="text" value="17/10/2008"/>	
Costos (en millones de pesos) y cantidad de eventos	
Ruta del Portafolio que contiene la información en Excel: <input type="text" value="C:\Archivos de programa\Apache Software Foundation\Tomcat\"/>	
1. Inversión total de Sísmica 2D	<input type="text" value="48.0"/>
2. Inversión Total de Sísmica 3D	<input type="text" value="36.0"/>
3. Inversión Total en Sísmica 2D y 3D	<input type="text" value="84.0"/>
4. Inversión total de Estudios de Campo	<input type="text" value="60.0"/>
5. Inversión total de Estudios de Gabinete	<input type="text" value="40.0"/>
6. Inversión Total de Estudios de Campo y de Estudios de Gabinete	<input type="text" value="100.0"/>
7. Inversión Operacional total	<input type="text" value="81.0"/>
Monto de la Inversión Total	<input type="text" value="265.0"/>
8. Kilómetros totales en Sísmica 2D	<input type="text" value="90.0"/>
9. Kilómetros cuadrados totales en Sísmica 3D:	<input type="text" value="66.0"/>
10. Número de Estudios de Campo:	<input type="text" value="12.0"/>
11. Número de Estudios de Gabinete:	<input type="text" value="120.0"/>
12. Número total de Estudios de Campo y de Gabinete	<input type="text" value="132.0"/>
13. Número de eventos de Inversión Operacional:	<input type="text" value="18.0"/>

Para actualizar primero elija el Libro Excel y después continúe:

[continuar](#)

Figura C.9 Proceso de lectura y transformación de información residente en archivos de recursos Excel

Al continuar la operación de la aplicación, se muestran los datos globales extraídos y procesados para actualizar las Tablas correspondientes, véase la Figura C.10.

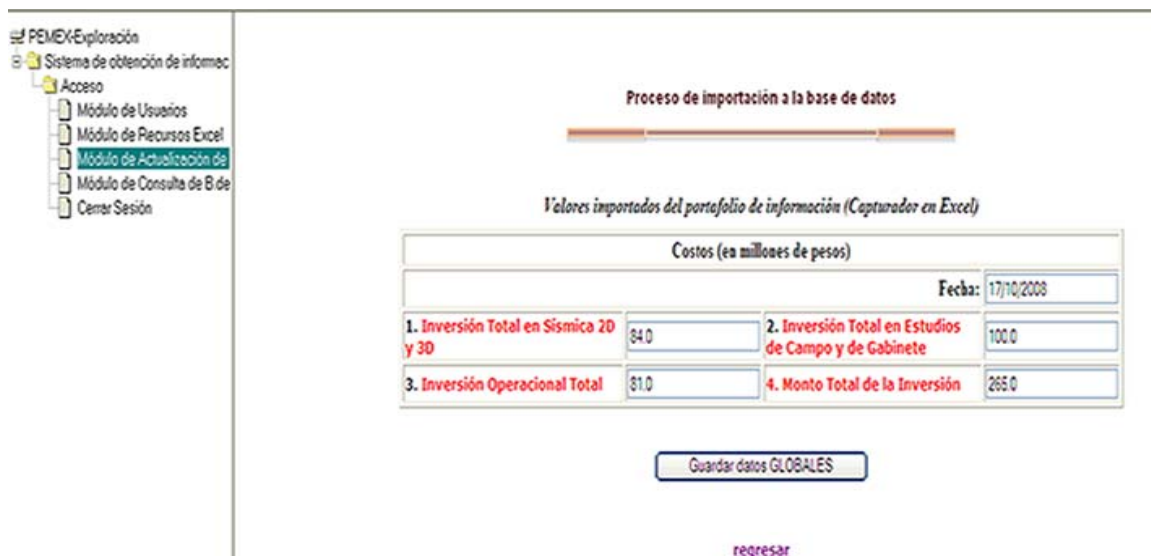


Figura C.10 Datos a ser ingresados en las Tablas de costos.

Finalmente y después de dar clic en el botón “Guardar datos globales”, mostrado en la figura anterior, en la Figura C.11 aparece la notificación de la actualización exitosa de la Base de Datos.

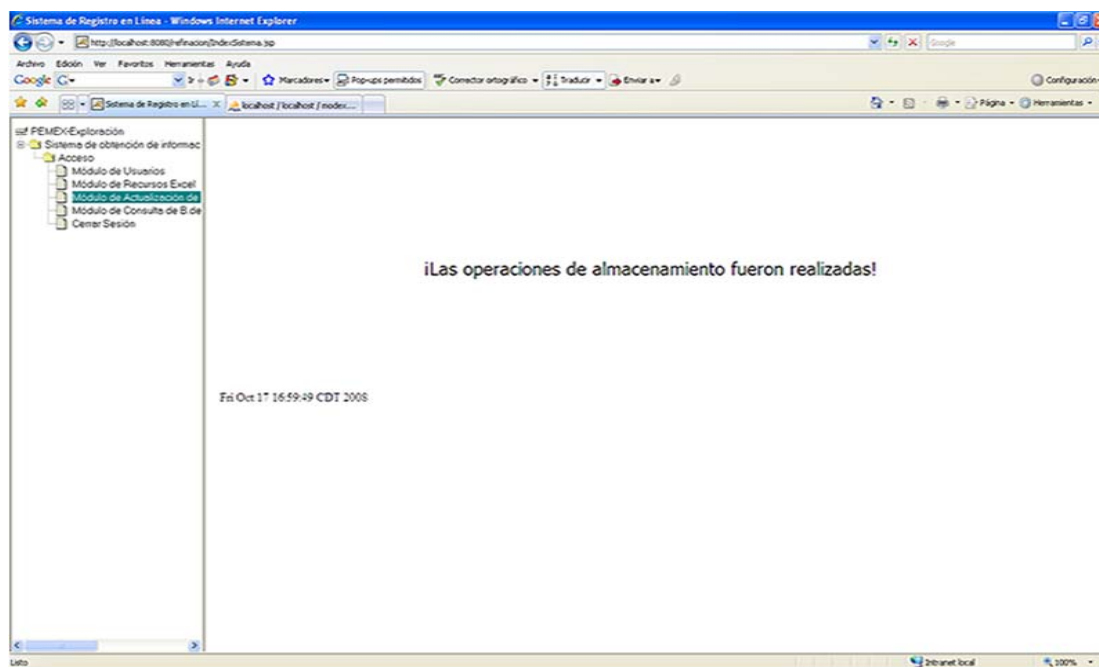


Figura C.11 Resultado del proceso de actualización

En el caso de que se elija un archivo cuya información ya ha sido previamente grabada en la base, se emitirá el resultado mostrado en la figura C.12.

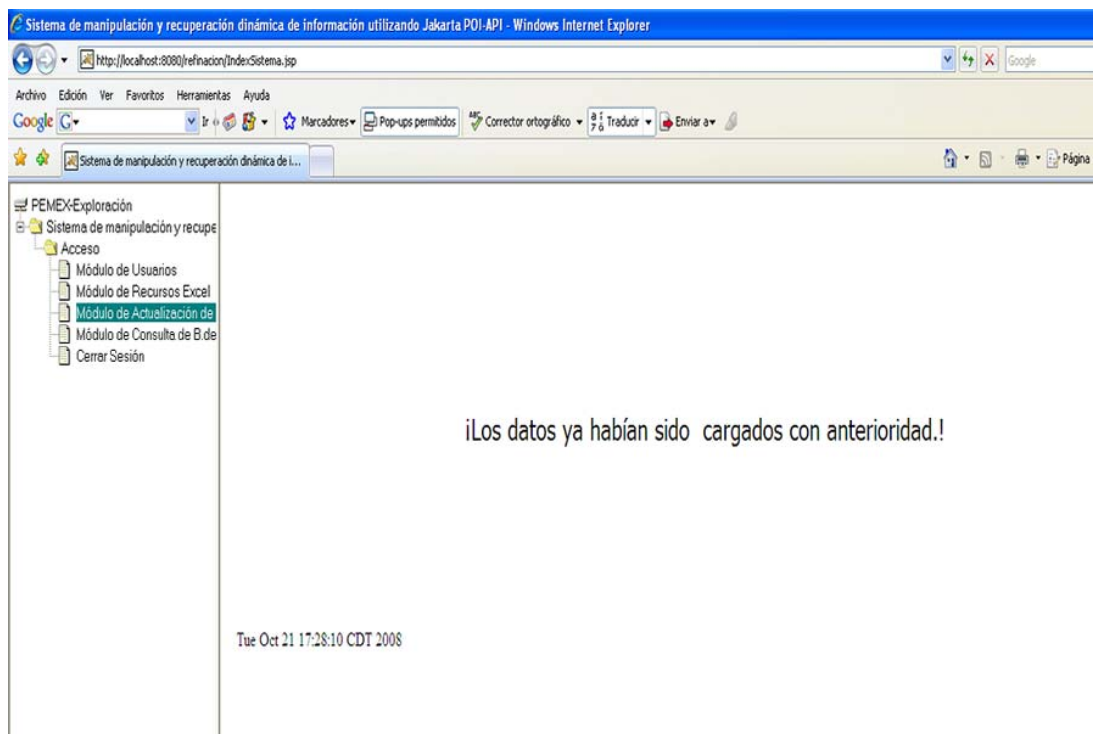


Figura C.12 Aviso de proceso previamente actualizado

Módulo de Consulta

Para ingresar a este módulo simplemente se elige del Menú principal y de manera automática mostrará los resultados de la actualización de la base de datos, como se muestra en la figura C.13.

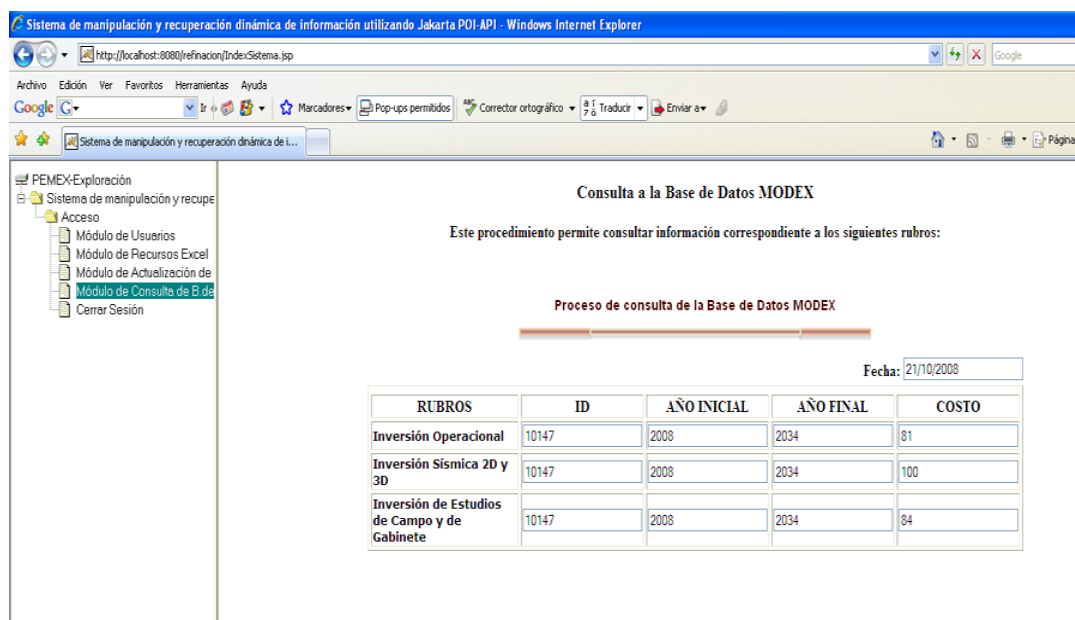


Figura C.13. Resultado de la consulta a la Base de datos ya actualizada.

La interfaz de datos de la consulta muestra la información referente al identificador del proyecto, el período respectivo (año inicial y final) así como los costos asociados a cada rubro.

Para terminar o cuando se desee salir del entorno de la operación, simplemente se elige Cerrar Sesión, con lo cual mostrará la pantalla de acceso y solicitará la información pertinente para una nueva sesión. Figura C.14.

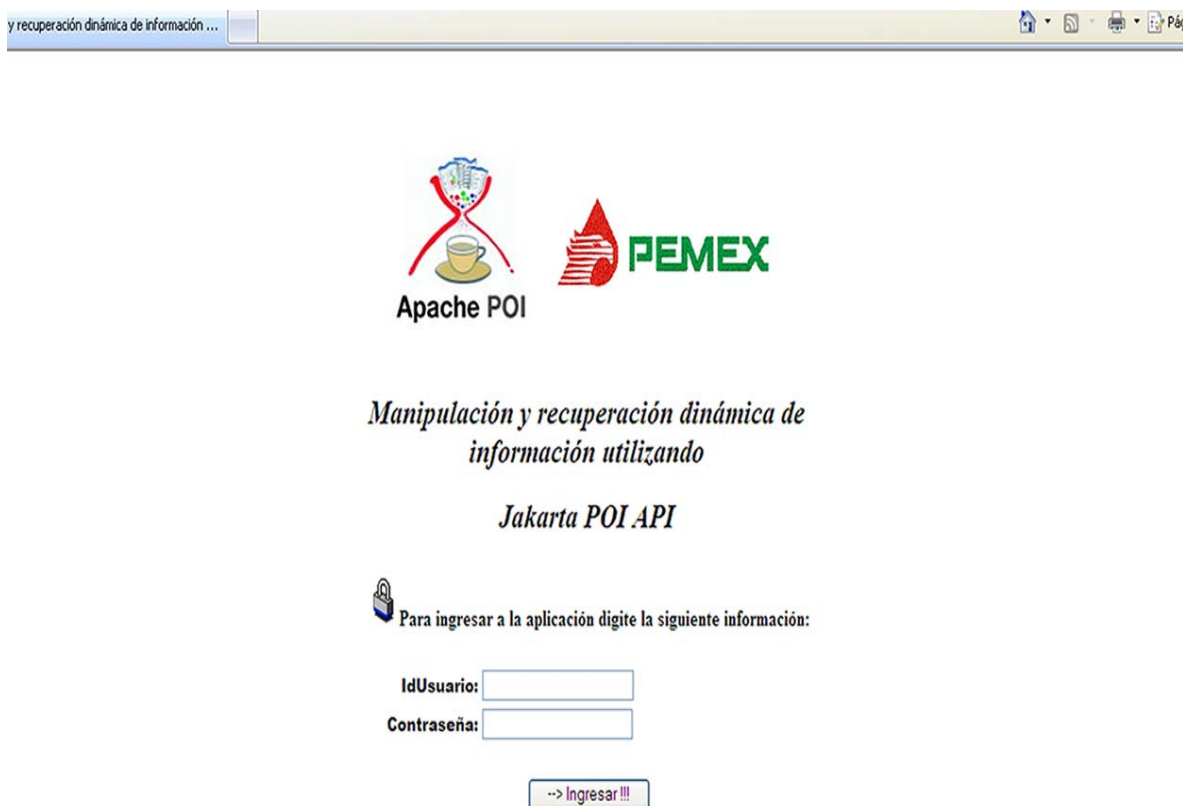


Figura C.14 Interfaz de acceso a la aplicación para una nueva sesión.

Apéndice D Tablas que integran la base de datos de prueba.

costofijo	costosdeinversion
costosdeinversionestudios	costosdeinversionestudioscampo
costosdeinversionestudiosgabinete	costosdeinversionoperacional
costosdeinversionsismica	costosdeinversionsismica2d
costosdeinversionsismica3d	costosdeinversionunidadoperacional
costovariable	distribuciones
distribuciones_arboles	distribuciones_buque_compra
distribuciones_buque_renta	distribuciones_costos_fijos
distribuciones_costos_total	distribuciones_costos_variables
distribuciones_costos_variables_compra_de_gas	distribuciones_costos_variables_interorganismos
distribuciones_costos_variables_mano_de_obra	distribuciones_costos_variables_materiales
distribuciones_costos_variables_reserva_laboral	distribuciones_costos_variables_servicios_generales
distribuciones_cubierta	distribuciones_flujos_descontado
distribuciones_flujos_inversiondescontada	distribuciones_flujos_total
distribuciones_ingresos_aceiteligero	distribuciones_ingresos_aceitepesado
distribuciones_ingresos_aceitesuperligero	distribuciones_ingresos_condensado
distribuciones_ingresos_gashumedo	distribuciones_ingresos_gaseco
distribuciones_ingresos_total	distribuciones_inversion_baterias
distribuciones_inversion_ductos	distribuciones_inversion_estaciondecompresion
distribuciones_inversion_infraestructuraexploratoria	distribuciones_inversion_infraestructurapozodesarrollo
distribuciones_inversion_lineadedescarga	distribuciones_inversion_manodeobraperforacion
distribuciones_inversion_operacionalfuturodesarrollo	distribuciones_inversion_perforacionpozodesarrollo
distribuciones_inversion_perforacionpozoexploratorio	distribuciones_inversion_plataformadedesarrollo
distribuciones_inversion_terminacionpozodesarrollo	distribuciones_inversion_terminacionpozoexploratorio
distribuciones_inversion_total	distribuciones_inversion_total_desarrollo
distribuciones_inversion_total_desarrollo_sin_operacional	distribuciones_inversion_total_inversionexploratoria
distribuciones_manifolds	distribuciones_numerodepozosdesarrolloperforados
distribuciones_numerodepozosdesarrolloterminados	distribuciones_numerodepozosefectivostrabajandoporann o
distribuciones_perfil_de_reservas	distribuciones_producciondiariapromedioeneseanno
distribuciones_risers	distribuciones_scontrol
distribucion_recursos	documentacion
objetivo	plandedesarrollo
preciosproducto	simulable
simulables_integrados	tipodecambio
variablesglobales	volumetriasoriginales

APÉNDICE E Requerimientos para el desarrollo y soporte de la aplicación.

Para el desarrollo y soporte de la aplicación se consideraron los siguientes requerimientos de software y de hardware, según las Tablas E.1 y E.2 respectivamente:

Tabla E.1. Requerimientos de software	
Componente	Requisitos
Base de datos tipo MYSQL	<ul style="list-style-type: none"> XAMPP (Manejador de base de datos MYSQL).
Navegador de Internet	<ul style="list-style-type: none"> Al menos Microsoft® Internet Explorer 6.01
Servidor WEB	<ul style="list-style-type: none"> Apache TOMCAT 5.5.
Enlace para conexión de bases de datos tipo MYSQL con desarrollos Java	<ul style="list-style-type: none"> JDBC (Connector/J) para MYSQL
API para el manejo de los documentos con formatos tipo Excel.	<ul style="list-style-type: none"> POI API de JAKARTA APACHE
Entorno integrado para desarrollo Java	<ul style="list-style-type: none"> NETBEANS 5.5
Máquina Virtual de Java	<ul style="list-style-type: none"> JDK 4.1 o superior.
Entorno para desarrollo de las interfaces visuales en Internet	<ul style="list-style-type: none"> Dreamweaver
Manejo de documentos Excel	<ul style="list-style-type: none"> MS OFFICE-Excel 2003
Diseño de diagramas UML	<ul style="list-style-type: none"> MS VISIO 2002
Entorno para el desarrollo del capturador Excel	<ul style="list-style-type: none"> Visual Basic for Applications (version 3 o superior)

Tabla E.2. Requerimientos de hardware	
Componente	Requisitos
Servidor de aplicaciones	Para efectos de este trabajo se utilizóDeseable que tenga la siguiente configuración: sistema operativo MSW-XP Professional 2002 SP3, procesador Intel Core 2 Duo E8400 a 3.0 Ghz, memoria de 2 Gb RAM de extensión física, disco duro de 250 Gb, tarjeta de red Ethernet 10/100, monitor LCD de 17”.

Apéndice F Procedimientos en Java y JSP

IndexSistema.jsp

```
<!--
Permite el inicio de sesión para el
usuario en turno, previo registro existente en
la base de datos y de acuerdo a su
nivel de acceso
-->
<html>
<head>
<title>Manipulación y recuperación dinámica de información utilizando
Jakarta POI API</title>
</head>
<body>

<%@ page session='true'%>
<%@ page import="java.sql.*" %>

<%

    String strUser = request.getParameter("User");
    String strPassword = request.getParameter("Password");

    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
    }

    catch (java.lang.ClassNotFoundException e)
    {
        e.printStackTrace();
    }

    Connection conn = null;
    Statement mandato= null;
    ResultSet resultado=null;

    int strId=0;

    try {

        conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/modex3","root","
");

        if (conn == null)
        {
            out.println("No me pude Conectar");
            return;
        }


```

```

        mandato = conn.createStatement();
        resultado = mandato.executeQuery("SELECT password FROM usuarios
where password='"+strPassword+"' and login = '"+strUser+"' and tipo =
'administrador' ");

        if (resultado.next())
        {
            HttpSession sesionOk = request.getSession();
            sesionOk.setAttribute("strUser",strUser);
        }
    %>

    <jsp:forward page="index.jsp" />
<%
    resultado.close();
    mandato.close(); } else {

        mandato = conn.createStatement();
        resultado = mandato.executeQuery("select password from usuarios
where
                                password = '"+strPassword+"' and login =
'"+strUser+"' and tipo =
                                'usuario' ");

        if (resultado.next())
        {
            HttpSession sesionOk = request.getSession();
            sesionOk.setAttribute("strUser",strUser);
        }
    %>

    <jsp:forward page="archivos_usuario.jsp" />

<%
    resultado.close();
    mandato.close(); } else {
    %>

<jsp:forward page="home2.html" />

<%} }

conn.close();

}

catch (SQLException e) {
e.printStackTrace();
}

%>

<br>
<br>

```



```
<br>
<br>
<br>
<br>
<p align="center"><a href="home2.html"><small><font
face="Arial">Regresar</font></small></a></p>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>

<%= new java.util.Date() %>
</body>
</html>
```

GuardaDatos2.jsp

```

<!--
Procedimiento que se
encarga de la actualización de
la Base de Datos de prueba.
-->

<html>
<head>
<title>Manipulación y recuperación dinámica de información utilizando
Jakarta POI API</title>
</head>
<body>
<%@ page language="java"%>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.*"%>
<%@ page import="java.io.*"%>
<%
        Connection conn      = null;
        Statement mandato    = null;
        ResultSet resultado  = null;

        String strRegistro = "";
        String strInicio   = "";
        String strFinal     = "";

        String costo1 = "";
        String costo2 = "";
        String costo3 = "";
        String costo4 = "";
        String nbase  = "";

        strInicio     = request.getParameter("ini");
        strFinal       = request.getParameter("fin");
        strRegistro    = request.getParameter("reg");
        costo1         = request.getParameter("valor1tsis");
        costo2         = request.getParameter("valor2tcyg");
        costo3         = request.getParameter("valor3tiop");

        // nbase       = request.getParameter("base");

try {

        Class.forName("com.mysql.jdbc.Driver").newInstance();
    }
catch (java.lang.ClassNotFoundException e) {
    e.printStackTrace();
}

try {

```

```

conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/modex3","root","
");
//conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/nbase","root","
");
if (conn == null) {
    out.println("No me pude Conectar");
    return;}

    try {
        mandato = conn.createStatement();
        mandato.execute("insert into
totalcostosdeinversiondeestudioscyg values ("+strRegistro+",
"+strInicio+", "+strFinal+", "+costo1+") ");
        mandato.close();

        mandato = conn.createStatement();
        mandato.execute("insert into
totalcostosdeinversiondesismica2dy3d values ("+strRegistro+",
"+strInicio+", "+strFinal+", "+costo2+") ");
        mandato.close();

        mandato = conn.createStatement();
        mandato.execute("insert into
totalcostosdeinversionoperacional values ("+strRegistro+",
"+strInicio+",
"+strFinal+", "+costo3+") ");
        mandato.close();

        %>

```

```

<br>
    <br>
<br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>

```

```

<center>
    <font face="Tahoma" size="5">¡Las operaciones de
almacenamiento </font>

```


cdtablas.jsp

```
<!--
Este procedimiento permite desplegar
de manera global la información previamente
actualizada de la base de datos de prueba.
-->

<%@ page language="java" import="java.sql.*, java.io.*, java.util.*,
consultar.*" errorPage="error.jsp" %>

<jsp:useBean id="Consultar" class="consultar.Consultar" />

<html>

<head>

<title>Manipulación y recuperación dinámica de información utilizando
Jakarta POI API</title>

</head>

<body>

<h3 align="center">Consulta a la Base de Datos MODEX</h3>
<h4 align="center">Este procedimiento permite consultar información
correspondiente a los siguientes rubros:</h4>

<%
String clave = "";

// Valores de Campo y Gabinete
String valor0    = "";
String valor1    = "";
String valor2    = "";
String valor3    = "";

// Valores de Sísmica 2D y 3D
String valor4    = "";
String valor5    = "";
String valor6    = "";
String valor7    = "";

// Valores de Inversión Operacional
String valor8    = "";
String valor9    = "";
String valor10   = "";
String valor11   = "";

String fecha = "";
String dia   = "";
String mes   = "";
String anno  = "";
```

```

ResultSet rs = null; // inversión operacional
ResultSet rs1 = null; // sísmica 2D y 3D
ResultSet rs2 = null; // estudios de campo y gabinete
%>

<%
Calendar c = new GregorianCalendar();
dia = Integer.toString(c.get(Calendar.DATE));
mes = Integer.toString(c.get(Calendar.MONTH)+1);
año = Integer.toString(c.get(Calendar.YEAR));
fecha = (dia + "/" + mes + "/" + año);

Consultar.connect();
rs = Consultar.VerDatos(clave);
rs1 = Consultar.VerDato(clave);
rs2 = Consultar.VerData(clave);

while (rs.next()){
    valor0 = rs.getString("id");
    valor1 = rs.getString("año_i");
    valor2 = rs.getString("año_f");
    valor3 = rs.getString("costo");}

while (rs1.next()){
    valor4 = rs1.getString("id");
    valor5 = rs1.getString("año_i");
    valor6 = rs1.getString("año_f");
    valor7 = rs1.getString("costo");}

while (rs2.next()){
    valor8 = rs2.getString("id");
    valor9 = rs2.getString("año_i");
    valor10 = rs2.getString("año_f");
    valor11 = rs2.getString("costo");}

Consultar.disconnect();
%>
<center>

<table border="0">
<tr>
<td height="69" align="center">

<object classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000 width=431
height=48

codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=        6,0,29,0>
    <param name="movie" value="flash/titulo.swf">
    <param name="quality" value="high">
    <param name="FlashVars" value="texto=Proceso de consulta de la Base
de Datos MODEX">
    <param name="PLAY" value="false">
    <param name="SCALE" value="exactfit">

```



```
        <td align="center"><input type="text" name="costo" value =
<%=valor7%>></td>
</tr>
<tr>
    <td><font COLOR="#000000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
        <b> Inversi&oacute;n de Estudios de Campo y de
Gabinete</b></font></td>
        <td align="center" ><input type="text" name="id" value =
<%=valor8%>></td>
        <td align="center"><input type="text" name="anio_i" value =
<%=valor9%>></td>
        <td align="center"><input type="text" name="anio_f" value =
<%=valor10%>></td>
        <td align="center"><input type="text" name="costo" value =
<%=valor11%>></td>
    </tr>
</table>

</body>
</html>
```


ctotalac_1.jsp

```

<%--
Procedimiento que consulta en la base de datos de prueba la referencia
del archivo Excel elegido, llama al bean que contiene el método que lee y
recupera la información de dicho archivo, presenta la información
extraída al usuario y envía datos específicos al siguiente procedimiento
--%>

<html>
<head>
<title>Manipulación y recuperación dinámica de información utilizando
Jakarta POI API</title>
</head>
<body>

<%@ page language="java" import="java.sql.*, java.io.*, java.util.*,
exceproces5.*"%>
<jsp:useBean id="Main" class="exceproces5.Main" />

<%@ page language = "java" import =
"org.apache.poi.poifs.filesystem.poifsfilesystem.*" %>
<%@ page language = "java" import = "org.apache.poi.hssf.usermodel.*"
errorPage = "error.jsp" %>

<%
double [] dats = new double [15];
for(int i =0;i<14;i++){
dats[i] = 0.0;}

// Valores a sumar
double kms2d    = 0.0;
double valors2d = 0.0;
double kms23d   = 0.0;
double valors3d = 0.0;
double numeca   = 0.0;
double valoreca = 0.0;
double numega   = 0.0;
double valorega = 0.0;
double numiop   = 0.0;

// Valores totales
double kmstotsis = 0.0;
double numtest   = 0.0;
double numtcyg   = 0.0;
double valorltsis = 0.0;
double valor2tcyg = 0.0;
double valor3tiop = 0.0;

// Suma de importes totales
double temp = 0.0;

// Año de inicio del proyecto
Double aniol = 0.0;

```

```

String dia, mes, annio;
%>
<%
String str      = "";
String base     = "";
String fecha    = "";
String esok     = "";
String archiv   = "";
String clave    = "";
String msg      = "";
String bien     = "";
String portafolio = "";

int in          = 1;
int anpro       = 0;
int anpro1      = 0;
try {

    Calendar c = new GregorianCalendar();
    dia = Integer.toString(c.get(Calendar.DATE));
    mes = Integer.toString(c.get(Calendar.MONTH)+1);
    annio = Integer.toString(c.get(Calendar.YEAR));
    fecha = (dia + "/" + mes + "/" + annio);

    portafolio = request.getParameter("archiv");

    if (portafolio == null) {
        portafolio = "*****";
    }

    clave = (request.getParameter("archiv"));

    esok = (request.getParameter("OK"));
    if (esok != null)
        dats = Main.suma(clave);
    valors2d = dats[0];
    valors3d = dats[1];
    valor1tsis = dats[2];
    valoreca = dats[3];
    valorega = dats[4];
    valor2tcyg = dats[5];
    valor3tiop = dats[6];
    kms2d = dats[7];
    kms23d = dats[8];
    numeca = dats[9];
    numega = dats[10];
    numtcyg = dats[11];
    numiop = dats[12];
    temp = dats[13];
    anio1 = dats[14];
    anpro = Double.valueOf( anio1 ).intValue();
    if (anpro==0) anpro=(c.get(Calendar.YEAR));
    anpro1 = anpro + 26;
}

```

```

%>

<center>

<table border="0">
<tr>
<td height="69" align="center">

<object classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000 width=431
height=48

codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.c
ab#version=          6,0,29,0>
    <param name="movie" value="flash/titulo.swf">
    <param name="quality" value="high">
    <param name="FlashVars" value="texto=Proceso de importación de
valores del capturador de Excel">
    <param name="PLAY" value="false">
    <param name="SCALE" value="exactfit">
    <embed
    src="flash/titulo.swf?texto=Valores importados del capturador de
Excel" width="431" height="48" align="absmiddle" quality="high"
    pluginspage="http://www.macromedia.com/go/getflashplayer"
    type="application/x-shockwave-flash" play="false"
scale="exactfit"></embed>
    </object>
</td>
</tr>
<br>
<tr>
<td>
<table align = "center" border=1 cellpadding=1 cellspacing=3 width="800">

    <tr>
    <td colspan="4" align="center">

    </td>
    </tr>
    <tr >
    <td colspan = 4 align="right">
    <b>Fecha: </b><input type = "text" value = <%=fecha%> width="90">
    </td>
    </tr>

    <tr>
    <td align = "center" colspan = 4><b>Costos (en millones de pesos) y
cantidad de eventos </b> </td><td width="0"></td>
    </tr>
    <tr>
    <td align="left" colspan = 4><em>Ruta del Portafolio que contiene la
informaci&oacute;n en Excel: </em>
    <b>

```

```

        <input type="text" maxlength="100" size="60" name = "clave"
value = "<%=portafolio%>"> </b> </td>

</tr>
<tr>
    <td width="292"><font COLOR="000000"
FACE="Verdana,Arial,Helvetica,Sans-serif" SIZE="2" align="left">
    <b>1. Inversi&oacute;n total de S&iacute;smica 2D </b></font></td>
    <td width="162" align="center"><input type="text" name="valors2d"
value = <%=valors2d%> ></td>
    <td width="162" align="left" ><font COLOR="000000"
FACE="Verdana,Arial,Helvetica,Sans-serif" SIZE="2" align="left">
    <b>8. Kil&oacute;metros totales en S&iacute;smica 2D </b></font></td>
    <td width="144" align="center"><input type="text" name="kms2d" value
= <%=kms2d%> > </td>
</tr>
<tr>
<td><font COLOR="000000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
    <b>2. Inversi&oacute;n Total de S&iacute;smica 3D </b></font></td>
    <td align="center"><input type="text" name="valors3d" value =
<%=valors3d%> > </td>
    <td align="left" ><font COLOR="000000"
FACE="Verdana,Arial,Helvetica,Sans-serif" SIZE="2" align="left">
    <b>9. Kil&oacute;metros cuadrados totales en S&iacute;smica 3D:
</b></font></td>
    <td align="center"><input type="text" name="kms23d" value =
<%=kms23d%> > </td>
</tr>
<tr>
<td><font COLOR="#FF0000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
    <b>3. Inversi&oacute;n Total en S&iacute;smica 2D y 3D</b></font></td>
    <td align="center"><input type="text" name="valorltsis" value =
<%=valorltsis%> > </td>

</tr>
<tr>
<td><font COLOR="#000000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
    <b>4. Inversi&oacute;n total de Estudios de Campo </b></font></td>
    <td align="center" ><input type="text" name="valoreca" value =
<%=valoreca%> > </td>
    <td align="left"><font COLOR="000000"
FACE="Verdana,Arial,Helvetica,Sans-serif" SIZE="2" align="left">
    <b>10. N&uacute;mero de Estudios de Campo:</b></font></td>
    <td align="center"><input type="text" name="numeca" value =
<%=numeca%> > </td>
</tr>
<tr>
<td ><font COLOR="#000000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
    <b>5. Inversi&oacute;n total de Estudios de Gabinete</b></font></td>

```

```

        <td align="center"><input type="text" name="valorega" value =
<%=valorega%> > </td>
        <td align="left" ><font COLOR="000000"
FACE="Verdana,Arial,Helvetica,Sans-serif" SIZE="2" align="left">
        <b>11. N&uacute;mero de Estudios de Gabinete: </b></font></td>
        <td align="center"><input type="text" name="numega" value =
<%=numega%> ></td>
    </tr>
    <tr>
        <td ><font COLOR="#FF0000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
        <b>6. Inversi&oacute;n Total de Estudios de Campo y de Estudios de
Gabinete </b></font></td>
        <td align="center"><input type="text" name="valor2tcyg" value =
<%=valor2tcyg%> ></td>
        <td align="left"><font COLOR="000000"
FACE="Verdana,Arial,Helvetica,Sans-serif" SIZE="2" align="left">
        <b>12. N&uacute;mero total de Estudios de Campo y de Gabinete
</b></font></td>
        <td align="center"><input type="text" name="numtcyg" value =
<%=numtcyg%> > </td>
    </tr>
    <tr>
        <td ><font COLOR="#FF0000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
        <b>7. Inversi&oacute;n Operacional total</b></font></td>
        <td align="center"><input type="text" name="valor3tiop" value =
<%=valor3tiop%> > </td>
        <td ><font COLOR="000000" FACE="Verdana,Arial,Helvetica,Sans-serif"
SIZE="2" align="left">
        <b>13. N&uacute;mero de eventos de Inversi&oacute;n Operacional:
</b></font></td>
        <td align="center"><input type="text" name="numiop" value =
<%=numiop%> > </td>
    </tr>
    <tr>
        <td align="center"> <font COLOR="#FF0000"
FACE="Verdana,Arial,Helvetica,Sans-serif" SIZE="2" align="left">
        <b>Monto de la Inversi&oacute;n Total</b></font></td>
        <td align="center"><input type="text" name="temp" value = <%=temp%> >
</td>
    </tr>
</table>
<%
}catch (Exception e){
    msg = "Exception: Un Error desconocido ha ocurrido.";
}
%>

</td>
</tr>
</table>

<br>

```



```

/*
 * Main.java
 *
 * Contiene el método suma, que recupera información de los
 * archivos del Capturador Excel, utilizando las bibliotecas
 * HSSF y POIFS del POI-API-JAVA.
 *
 * Se tiene también el método VerDatos que se utiliza de
 * prueba para el proceso de actualización y consulta de la
 * base de datos de prueba.
 *
 * Se incluye el método Connect, utilizado para establecer
 * la conexión con la base de datos de prueba.
 *
 * Se incluye el método disconnect, que se utiliza para
 * finalizar el enlace con la base de datos.
 *
 * Mediante el método VerArchivo, se consulta la tabla
 * archivos, de la base de datos de prueba, para acceder a
 * información relacionada con archivos del Capturador
 * Excel, previamente elegidos para su proceso.
 */

```

```
package exceproces5;
```

```

import java.io.*;
import java.util.*;
import java.sql.*;
import org.apache.poi.hssf.usermodel.*;
import org.apache.poi.poifs.filesystem.POIFSFileSystem;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFName;

```

```

/**
 *
 * @author PGG
 */

```

```

public class Main {
    String msg;
    String dia    = "";
    String mes    = "";
    String annio = "";
    String fecha = "";
    Connection con;

    Statement stmt;
    ResultSet rs = null;
    String error;
    double costos2dy3d = 0;
    double costocyg = 0;
    double costotiop = 0;
    String tabla1 = "totalcostosdeinversiondeestudioscyg";

```

```
String tabla2 = "totalcostosdeinversiondesismica2dy3d";
String tabla3 = "totalcostosdeinversionoperacional";

/** Creates a new instance of Main */
public Main() {
}

public static double [] suma(String archivo ) {

    // Número de año
    int    anio = 0;
    double anio1 = 0.0;

    // Número de renglón
    int numren = 0;

    // Número de renglones en la hoja
    int  reng = 0;
    int  numer = 0;

    // Valores a sumar
    double kms2d    = 0.0;
    double valors2d = 0.0;
    double kms23d   = 0.0;
    double valors3d = 0.0;
    double numeca   = 0.0;
    double valoreca = 0.0;
    double numega   = 0.0;
    double valorega = 0.0;
    double numiop   = 0.0;

    // Valores totales
    double kmstotsis = 0.0;
    double numtest   = 0.0;
    double numtcyg   = 0.0;
    double valor1tsis = 0.0;
    double valor2tcyg = 0.0;
    double valor3tiop = 0.0;

    // Suma de importes totales
    double temp = 0;
    double despl = 0.0;

    double [] dats = new double [15];

    HSSFRow anios;
    HSSFRow renglon;
    HSSFRow renglon1;
    HSSFCell celda;
    HSSFCell celda1;
    HSSFCell anioini;

    try{
        for(int m =0;m<=14;m++){
```



```

        dats[m] = 0;}

// Recibe el nombre del archivo a leer (formato Excel)
POIFSFileSystem fs = null;
fs = new POIFSFileSystem(new FileInputStream(archivo));
HSSFWorkbook libro = null;
libro = new HSSFWorkbook(fs);

// Ubica la hoja de totales
HSSFSheet hoja = libro.getSheetAt(6);

// ubica el año de inicio del proyecto
anios = hoja.getRow(3);
anioini = anios.getCell((short)1);
anio1 = anioini.getNumericCellValue();
//anio = Double.valueOf(anio1).intValue();

// Cálculo de acumulados
for(int i = 0; i < 29; i++) {
    renglon = hoja.getRow((short)i);
    // System.out.println("renglon: " +
renglon.getRowNum());
    if(renglon != null){
        numren = renglon.getRowNum();
        if(numren == 4){
            renglon1 = hoja.getRow((short)i+1);
            celda1 = renglon1.getCell((short)28);
            valors2d = celda1.getNumericCellValue();
            dats [0] = valors2d;
            // System.out.println("Inversión total de Sísmica
2D(millones de pesos): "+ celda1.getNumericCellValue());
            celda = renglon.getCell((short)28);
            kms2d = celda.getNumericCellValue();
            dats [7] = kms2d;
            // System.out.println("Kilometros totales en
Sísmica 2D: "+ celda.getNumericCellValue());
        } else{
            if(numren == 8){
                renglon1 = hoja.getRow((short)i+1);
                celda1 = renglon1.getCell((short)28);
                valors3d = celda1.getNumericCellValue();
                dats [1] = valors3d;
                // System.out.println("Inversión Total de
Sísmica 3D(millones de pesos): "+ celda1.getNumericCellValue());
                celda = renglon.getCell((short)28);
                kms23d = celda.getNumericCellValue();
                dats [8] = kms23d;
                // System.out.println("Kilometros cuadrados
totales en Sísmica 3D: "+ celda.getNumericCellValue());
            } else{
                if(numren == 11){
                    celda = renglon.getCell((short)28);

```

```

        valor1tsis = celda.getNumericCellValue();
        dats [2] = valor1tsis;
        // System.out.println("Inversión total en
Sísmica 2D y 3d(millones de pesos): "+ celda.getNumericCellValue());
    } else{
        if(numren == 15){
            renglon1 = hoja.getRow((short)i+1);
            celda1 = renglon1.getCell((short)28);
            valoreca =
celda1.getNumericCellValue();

            dats [3] = valoreca;
            // System.out.println("Inversión
total de Estudios de Campo(millones de pesos): "+
celda1.getNumericCellValue());

            celda = renglon.getCell((short)28);
            numeca = celda.getNumericCellValue();
            dats [9] = numeca;
            // System.out.println("Número de
Estudios de Campo: "+ celda.getNumericCellValue());
        } else{
            if(numren == 19){
                renglon1 =
hoja.getRow((short)i+1);

                celda1 =
renglon1.getCell((short)28);

                valorega =
celda1.getNumericCellValue();

                dats [4] = valorega;
                // System.out.println("Inversión
total de Estudios de Gabinete(millones de pesos): "+
celda1.getNumericCellValue());

                celda =
renglon.getCell((short)28);

                numega =
celda.getNumericCellValue();

                dats [10] = numega;
                // System.out.println("Número de
Estudios de Gabinete: "+ celda.getNumericCellValue());
            } else{
                if(numren == 23){
                    renglon1 =
hoja.getRow((short)i-1);

                    celda1 =
renglon1.getCell((short)28);

                    numtcyg =
celda1.getNumericCellValue();

                    dats [11] = numtcyg;
                    // System.out.println("Número
total de Estudios de Campo y de Gabinete: "+
celda1.getNumericCellValue());

                    celda =
renglon.getCell((short)28);

                    valor2tcyg =
celda.getNumericCellValue();

```



```

try {
    // Define el registro del driver a utilizar

    Class.forName("com.mysql.jdbc.Driver").newInstance();
    con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/modex3","root","
");

    } catch (ClassNotFoundException cnfe) {
        msg = "ClassNotFoundException: no se localizo el driver.";
        throw new ClassNotFoundException(msg);
    } catch (SQLException cnfe){
        msg = "SQLException: No se pudo conectar con la BD";
        throw new SQLException(msg);
    } catch (Exception e){
        msg = "Exception: Un Error desconocido ha ocurrido.";
        throw new Exception(msg);
    }
}

public void disconnect() throws SQLException {
    try {
        if(con != null) {
            con.close();
        }
    } catch (SQLException sqle){
        msg = ("SQLException: imposible cerrar la conexión con
laBD");
        throw new SQLException(msg);
    }
}

public ResultSet VerDatos(int idreg, double valor1tsis, double
valor2tcyg, double valor3tiop, int anio1, int anio2) throws SQLException,
Exception{

    try {

        if (con != null) {

            stmt = con.createStatement();
            stmt.execute("insert into
totalcostosdeinversiondeestudioscyg (id, anno_i, anno_f, costo)
values( "+idreg+", "+anio1+", "+anio2+", "+valor1tsis+" );");
            stmt.close();

        } else {
            error = "Exception: La conexión con BD se perdió.";
        }
        stmt = con.createStatement();
        String consulta = ("SELECT * FROM
totalcostosdeinversiondeestudioscyg");
        rs = stmt.executeQuery(consulta);
    }catch(Exception e) {

```

```
        msg = ("Exception = Algo está mal en el paso de parámetros.  
No se pudo realizar la operación, Verifique sus datos");  
        throw new Exception(msg);  
    }  
    return rs;  
}  
  
public ResultSet VerArchivo() throws SQLException, Exception{  
    ResultSet rsArchiv = null;  
  
    try {  
  
        String consulta = ("SELECT * FROM archivos;");  
        Statement stmt = con.createStatement();  
        rsArchiv = stmt.executeQuery(consulta);  
    }catch(Exception e) {  
        msg = ("Exception = No se pudieron recuperar los datos de los  
archivos de Excel. Verifique sus datos");  
        throw new Exception(msg);  
    }  
    return rsArchiv;  
}  
  
}
```