

**Universidad de las Ciencias Informáticas**  
**“Facultad 4”**



**Análisis y diseño del módulo de importación de  
ficheros y manejo de campos del software  
general de auditoría SOGAC.**

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**Autor(es):** Ana Beatriz Frómeta Luperon  
Ariel Antonio Milian González

**Tutor:** Lic. Guillermo Francisco Wood Fonseca

**Ciudad de la Habana**

**Julio 2008**

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Tutor: Lic. Guillermo Francisco Wood Fonseca

---

Ana Beatriz Frómeta Luperon

---

Ariel Antonio Milian González

## **DATOS DE CONTACTO**

Guillermo Francisco Wood Fonseca: Licenciado en Ciencias de la Computación, UH.

Diplomado en Economía, UH; Instructor Principal; NO Categoría Científica; 27 años de experiencia en el tema; 33 años de graduado en Informática.

## AGRADECIMIENTOS

*Todo lo bueno en el mundo es demasiado poco cuando se desea lo mejor para alguien a quien se estima, por esto simplemente agradezco a todo el que de una forma u otra ha hecho posible la realización de un sueño que lejos de ser una utopía es una realidad, especialmente a mis “segundos padres”, José y Anita que supieron mostrarme el camino correcto cuando mis padres estaban lejos. A mis padres y hermano, a Yoandy, a Leyri, a Jenny, a Zuzel, en fin a todos mis amigos y compañeros de estos 5 años de mi carrera.*

*También quiero agradecer a Tutor y a mi familia, a los padres de Leyri por su apoyo, dedicación y atención.*

*Y para terminar, quiero agradecer al amor y a la vida, que hicieron posible que este día se hiciera realidad.*

*Betty.*

*A los buenos amigos que he conocido aquí: a los que están, y a los que no están que podrían estar. Todos me han enseñado, de todos he aprendido.*

*Al tutor Wood, espejo de dedicación y energías.*

*Ariel.*

## DEDICATORIA

*A mis padres y a mi hermano,  
que son lo más querido entre  
mis seres queridos.*

*Ariel.*

*Dedico el presente trabajo:  
A mis padres, María Luperón y Hugo Frómeta  
que fueron mi guía y ejemplo.  
A mi novio, Yoandy que me brindó su apoyo y fuerzas  
para que viera realizados mis sueños.  
A mi hermano por sus consejos y confianza en mí.*

*Betty.*

## RESUMEN

Los procesos de informatización han impactado sobre las técnicas manuales y tradicionales de la auditoría. Como consecuencia lógica, han surgido las Técnicas de Auditoría Asistidas por Computadoras (TAAC). Una de las expresiones más conocidas de las TAAC son los softwares general de auditoría. El uso de programas con funcionalidades afines a estas técnicas, representa un beneficio para los auditores. En Cuba no existe un software de producción nacional de este tipo. Los que benefician a los auditores cubanos son pagados a altos precios en el mercado internacional.

Una de las funcionalidades básicas en un software de este tipo es la de importación de ficheros y manejo de los campos. En este trabajo, se ha realizado el análisis y diseño de un software que permita realizar estas operaciones. Utilizando la metodología del Proceso Unificado de Desarrollo (**RUP**), el lenguaje unificado de modelado (**UML**) y la herramienta **Rational Rose** se define el entorno del dominio del sistema, proponiendo un modelo de solución a partir de la identificación y la especificación de los requerimientos del software.

En tres capítulos, se explican las actividades realizadas en cada fase del trabajo para dar cumplimiento a los objetivos propuestos, presentando finalmente, el análisis y diseño obtenidos en la construcción del módulo de importación de ficheros y manejo de campos del Software General de Auditoría Cubano (**SOGAC**).

# ÍNDICE

DEDICATORIA.....	I
RESUMEN.....	II
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1 INTRODUCCIÓN.....	4
1.2 ESTADO DEL ARTE.....	4
1.2.1 ACL.....	4
1.2.2 IDEA.....	6
1.2.3 Crystal Reports.....	7
1.2.4 Agata Reports.....	8
1.2.5 Jasper Reports.....	9
1.3 ESTRUCTURAS DE ALMACENAMIENTO DE DATOS.....	10
1.4 FUNCIONAMIENTO DE LAS HERRAMIENTAS DE IMPORTACIÓN DE DATOS.....	11
1.5 METODOLOGÍA A UTILIZAR.....	12
1.5.1 El Proceso Unificado de Desarrollo de Software RUP.....	13
1.5.2 Características de RUP.....	14
1.5.2.1 Iterativo e Incremental.....	14
1.5.2.2 Dirigido por casos de uso.....	15
1.5.2.3 Centrado en la arquitectura.....	15
1.5.3 UML: Lenguaje Unificado de Modelado.....	16
1.6 HERRAMIENTAS UTILIZADAS: RATIONAL ROSE.....	18
1.7 TÉCNICA PARA VALIDACIÓN DE REQUISITOS: PROTOTIPO.....	19
1.8 CONCLUSIONES.....	19
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	20
2.1 INTRODUCCIÓN.....	20
2.2 PROPUESTA DE SOLUCIÓN.....	20
2.3 MODELO DEL DOMINIO.....	22
2.3.1 ¿Qué es un modelo del dominio?.....	22

2.3.2	¿Por qué hacer un modelo del dominio? .....	22
2.3.3	Definición de las entidades y conceptos principales.....	23
2.3.4	Diagrama del Modelo del Dominio. ....	24
2.4	ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE.....	24
2.4.1	Requerimientos Funcionales. ....	25
2.4.2	Requerimientos no funcionales. ....	26
2.5	DEFINICIÓN DE LOS ACTORES DEL SISTEMA.....	29
2.6	DIAGRAMA DE CASOS DE USO DEL SISTEMA. ....	29
2.6.1	Descripción de los Casos de Uso del Sistema. ....	31
2.7	CONCLUSIONES.....	54
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA .....		55
3.1	INTRODUCCIÓN.....	55
3.2	ANÁLISIS. ....	55
3.2.1	Diagrama de clases del análisis.....	56
3.3	DIAGRAMAS DE COLABORACIÓN DEL ANÁLISIS. ....	61
3.4	DISEÑO.....	61
3.4.1	Diagramas de clases del diseño.....	62
3.5	DIAGRAMAS DE SECUENCIA DEL DISEÑO. ....	71
3.6	DIAGRAMAS DE CLASES PERSISTENTES. ....	72
3.7	MODELO DE DATOS.....	73
3.8	CONCLUSIONES. ....	74
CONCLUSIONES .....		75
RECOMENDACIONES.....		76
BIBLIOGRAFÍA.....		77
GLOSARIO.....		95



# INTRODUCCIÓN

El uso creciente y acelerado de las nuevas tecnologías de la información y las comunicaciones (TIC) en los más diversos ámbitos de la vida personal y laboral, ha suscitado el interés global de las empresas de productos y servicios informáticos. En el contexto mundial de una economía con tendencia a agregar valor a los bienes y servicios, acorde al contenido en conocimientos que lo respalden, reemplazar viejos métodos y técnicas, por nuevas formas de capturar, guardar y presentar la información, que además sean más veloces y eficientes, encaminan los esfuerzos de la industria mundial del software. Los procesos de auditoría y control son de vital importancia en el enfoque empresarial.

Las Técnicas de Auditorías Asistidas por Computadoras (TAAC) van movidas por este interés global. A la vez que se reemplazan herramientas y técnicas tradicionales, se informatizan procedimientos manuales con las ventajas consiguientes. Con el empleo de las Técnicas de Auditorías Asistidas por Computadoras (TAAC), los beneficiarios - auditores en primera instancia, aunque podría decirse que el mundo empresarial en general - se favorecen en cuanto a la cobertura y alcance de la auditoría, menor tiempo de trabajo empleado y menor costo de las pruebas de control rutinarias. Se facilita además el registro de los datos, su interpretación mediante el vínculo con herramientas graficadoras, el acceso rápido y escalonado, la vinculación y comparación con otros datos y el filtrado de información de interés.

La importación de datos de fuentes externas al ambiente propio del software general de auditoría y el manejo de sus campos, es una funcionalidad esencial. Existe gran variedad de formatos que se utilizan con frecuencia para almacenar y gestionar datos de manera auxiliar, lo cual impone requerimientos excepcionales para la flexibilidad y versatilidad de herramientas de importación. En un software general de auditoría es de vital importancia desarrollar herramientas potentes para llevar a cabo estas funciones.

El software general de auditoría debe ser capaz de leer diferentes tipos de archivos comunes de forma automática, sin embargo en determinadas situaciones, el formato de los archivos de datos puede ser complejo o desconocido, en tal caso el software debe ser capaz de analizar estructuras de archivos de datos y construir una definición de registro, a través de un Asistente de Importación que pueda ser utilizada para interpretar e importar los datos. El software, debe ser capaz de ofrecer soluciones en los casos de importar ficheros mediante el ODBC (Open DataBase Connectivity) entre el sistema operativo y la base de datos, sin importar el Sistema Gestor de Base de Datos instalado, así como de importar ficheros de texto.

Los datos correspondientes deben ser leídos e interpretados en términos de campos dentro de registros, a tal efecto, en el proceso de importación de ficheros, debe ofrecerse también asistencia para realizar esta tarea. El criterio de importación, debe permitir filtrar los datos, importando sólo los registros que satisfacen un determinado criterio.

El hecho de ofrecer la funcionalidad de manejar los campos, creándolos, modificándolos o eliminándolos es una ventaja especial que debe ser tenida en cuenta en la implementación.

### **Situación Problemática:**

En Cuba se logran avances en la expansión del uso de las TIC, con una estrategia definida a nivel de gobierno, y la voluntad de poner las tecnologías en función de hacer más eficientes nuestras empresas y organismos. Numerosos procesos económicos y financieros se desarrollan en paralelo como parte de la dialéctica de la vida diaria del país. Estos procesos sin excepción son objeto de auditorías y controles constantes. Los profesionales cubanos de este sector no cuentan con un software general de auditoría de fabricación nacional, utilizan los que se comercializan en el mercado internacional, contando solo con unas pocas o viejas licencias (de 1997). Pagar las licencias de estos programas que se comercian en el mundo actualmente, sería muy costoso para la economía nacional. No conocemos software libre en esta categoría.

Modelar y desarrollar un software general de auditoría es una necesidad, adecuándolo a nuestras condiciones reales, que tenga como objetivo central, facilitar y hacer más eficientes los procesos de auditorías y control. En el proceso de desarrollo de un software de este tipo, es tarea primordial desarrollar un módulo importador de ficheros y manejador de campos. El proceso de importación de ficheros y manejos de campos en un software general de auditoría es una funcionalidad esencial, pues elimina barreras para el análisis de datos e información de fuentes externas al ambiente propio del programa.

En Cuba tampoco existe un software nacional que de forma general permita la importación de ficheros y manejos de campos, y los que se comercializan en el mundo son privados, caros y se acoplan a plataformas de desarrollo que también forman parte del software privado, por tanto no resulta factible su uso.

Después de un análisis de las situaciones explicadas, se identificó el **problema a resolver**: no está definida técnicamente, la forma estructural y funcional de importación de ficheros y manejo de campos del Software General de Auditoría Cubano (SOGAC).

Por tanto, nuestro **Objeto de Estudio** son los procesos de importación de ficheros y manejo de campos.

La **Idea a Defender** es: si se hace el análisis y diseño de un software que permita realizar los procesos de importación de ficheros y manejo de campos, contribuirá a la posterior implementación del Software General de Auditoría Cubano (SOGAC).

El **Campo de Acción** de este trabajo abarca los procesos de importación de ficheros y manejo de campos del software SOGAC.

El **Objetivo General** de esta investigación es obtener el análisis y diseño de un módulo de importación de ficheros y manejo de campos, requerido por el software SOGAC.

Acorde a esta propuesta, nos trazamos los **Objetivos Específicos**:

- Realizar un estudio sobre el marco teórico-conceptual de los mecanismos de importación de ficheros y manejo de campos en el software general de auditoría que conforman el estado del arte.
- Obtener el Análisis y Diseño de un software que permita importar ficheros y manejar campos en un software general de auditoría cubano (SOGAC).

Para cumplir estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **tareas**:

- Estudio y análisis del marco conceptual y el estado del arte de los software que se utilizan para importar ficheros y manejar campos.
- Definición de los diferentes artefactos que intervienen en la modelación del software.
- Realización de la captura de requisitos.
- Identificación y documentación de los casos de uso.
- Realización del Análisis y el Diseño.

El desarrollo y cumplimiento de las tareas propuestas, debe conducirnos a resolver el problema planteado.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

## 1.1 INTRODUCCIÓN

En este capítulo se brinda una explicación general de aspectos relacionados con aplicaciones de auditoría y control de gran uso y vigencia en el contexto informático actual. Tratamos además los procesos de importación y manejo de datos relacionados con tablas de base de datos, campos y registros.

También se ofrece una breve panorámica de las herramientas, técnicas y lenguajes de modelado utilizados para el desarrollo de nuestro proceso de análisis y diseño.

## 1.2 ESTADO DEL ARTE.

Los softwares de auditoría en los que se centra el estudio del estado del arte, son: ACL e IDEA en sus versiones recientes; ambos, son programas para la auditoría financiera y el análisis de datos. Estos dos softwares son los más usados en su categoría (software general de auditoría) a nivel mundial. Ambos, comparten funcionalidades afines a auditores y usuarios en general.

Además se analizarán los casos de Crystal Reports, Agata Reports y Jasper Reports que aunque no son software de auditoría, ofrecen funcionalidades de importación de datos y manejo de campos.

### 1.2.1 ACL.

**ACL** es un software para el análisis de datos, utilizado por auditores internos y externos en complejas organizaciones. Pertenece a la categoría de software privado; una licencia de **ACL** en su versión 9.1 cuesta 1 139,55 dólares en el mercado internacional.

Este software llega prácticamente a los datos de cualquier fuente, en cualquier sistema, mediante una interfaz de usuario consistente, ya sea que se encuentren en servidores o redes de computadoras.

Una de las principales funcionalidades que ofrece **ACL** es importar ficheros y manejar campos de datos.

**ACL** puede importar los datos accediendo según su tipo, de los siguientes ficheros:

- AS/400
- Microsoft Access, Excel, Word
- SAP
- DB2
- HTML
- COBOL

Una vez que **ACL** dispone de los ficheros que contienen la información necesaria para el trabajo, realiza su importación. Cuando importa el fichero, efectúa comprobaciones sobre su integridad.

En la importación de ficheros se tienen en cuenta los siguientes aspectos:

- Para tener adecuadamente localizados los ficheros de **ACL** se crea un directorio en el que se graban todos los ficheros que genera **ACL**. De esta forma en caso de querer copiarlo y transportarlo, se evitan posibles errores de funcionamiento. Es muy importante que todos los ficheros que va creando **ACL** estén en el mismo directorio a estos efectos.
- Una vez importado el fichero y con el fin de determinar si se ha realizado correctamente, se debe verificar que el número total de registros importados y el importe total de un campo con datos numéricos coincidan con los ficheros facilitados originales.
- Una vez importados los ficheros, se debe verificar que la información se encuentra en los formatos y campos que nos son útiles. A tales efectos se pueden hacer modificaciones de formato como por ejemplo:
  - Adecuación del formato del campo (Por ejemplo, al importar datos de Access, **ACL** le añade dígitos despreciables, podemos decirle el rango de los datos a importar y el formato numérico que deseamos).
  - Crear un nuevo campo que contenga los datos resultados de la aplicación de una ecuación a otros campos.

## 1.2.2 IDEA.

**IDEA** es un software de PC basado en la interrogación de ficheros, concebido para ser utilizado por auditores, contadores, investigadores y personal de seguridad informática esencialmente. Es distribuido por una compañía privada de desarrollo y comercialización de software; una licencia de **IDEA** para Windows 95/98/ME/NT/2000/XP cuesta 1900 dólares en el mercado internacional.

Entre las funcionalidades ventajosas que ofrece, se cuentan: permitir una mejor visión en la interpretación de los datos, por ejemplo análisis adicionales o creación de perfiles de datos, graficación, importar ficheros hacia su plataforma a la vez que permite manejar sus campos. También, analiza la información de diversas maneras y permite la extracción, el muestreo y la manipulación de datos para identificar errores, problemas, cuestiones específicas y tendencias.

Otra de sus ventajas principales el manejo de los campos, para seleccionar la información de manera coherente y segura. Cuenta con funciones para aritmética, fecha, hora y textos. Estas funciones le permiten ejecutar operaciones con fechas, cálculos financieros y estadísticos y búsqueda de textos. Permite también Secuenciar (indexar) los registros en la Ventana de Base de Datos.

**IDEA** permite como una de sus principales funcionalidades, importar datos, lo cual realiza de cuatro formas esencialmente:

- Cuando es un fichero conocido(AS 400, ASCII, dBASE, Microsoft Access, Microsoft Excel, Adobe PDF, SAP, XML).
- Cuando es un texto (ASCII o EBCDIC).
- Mediante el ODBC.
- Cuando es un fichero desconocido.

**IDEA** importa desde un amplio rango de tipos de ficheros, puede importar tantos archivos \*.mdb como \*.accdb además de:

- AS 400.
- ASCII o EBCDIC(Texto)
- dBASE
- Microsoft Access
- Microsoft Excel

- ODBC
- Adobe PDF
- SAP
- XML

En el momento de importar un fichero de texto es necesario, previo a la importación, definir su formato en **IDEA**. Para ello **IDEA** ofrece el Asistente de Importación que guiará al usuario a través de las diferentes etapas para definir e importar el archivo para su posterior análisis. Este asistente permitirá crear una definición de registro, guardarla e importar a través de él los ficheros.

Cuando se importa desde un EBCDIC longitud fija, **IDEA** identificará la longitud de cada registro, incluyendo los delimitadores de registro. **IDEA** excluirá automáticamente de la importación los delimitadores de registro.

**IDEA** proporciona dos opciones para secuenciar los registros: ordenar e indexar. Indexar ordena los registros en la base de datos activa, mientras ordenar, crea una nueva base de datos con los registros ordenados en base a la clave de orden ingresada. En el trabajo con los campos se pueden realizar las siguientes operaciones:

- **Crear Campos:** Durante este paso podrá agregar campos virtuales, editables.
- **Detalles de Campo:** En este paso podrá definir el nombre y los detalles de cada campo, y descartar determinados campos de la importación.
- **Paso Criterio de Importación:** En esta etapa del proceso de importación podrá ingresar una ecuación para filtrar los datos a ser importados en **IDEA**.

### 1.2.3 Crystal Reports.

**Crystal Reports** está diseñado para trabajar con una base de datos y ayudar en el análisis y la interpretación de la información importante. **Crystal Reports** facilita la creación de informes simples y dispone también de funciones poderosas necesarias para generar informes complejos o

especializados. Pertenece a la categoría de software privado y una licencia de **Crystal Reports Professional** cuesta 499 dólares.

Esta herramienta puede importar datos de una amplia gama de ficheros:

- ACCES (mdb),
- dBASE (5.0, III y IV)
- EXCEL (xls, .xlw)
- HTML
- XML
- PARADOX (3.x, 4.x, y 5.x)
- LOTUS (wk1, wk3, wk4)
- BTRIEVE (ddf)

**Crystal Reports** puede tomar también los datos directamente desde bases de datos mediante: SQL/ODBC o ADO. Además se puede importar un objeto con formato y basado en texto desde un archivo existente. Otra ventaja que ofrece es la manipulación de los datos y el manejo de los campos, se podrán insertar campos de fórmula, campos de enunciado, campos de parámetro, se puede además insertar campos (de un mismo tipo) en un objeto de texto. Los campos que son insertados en un objeto de texto, son acortados automáticamente (no tienen espacios en blanco al principio, ni al final).

### 1.2.4 Agata Reports.

**Agata Reports** es una aplicación liberada bajo la licencia GNU GPL, está escrita en PHP GTK y se soporta tanto en Windows como Linux.

Ofrece un conjunto de herramientas creadas para extraer lo más relevante de las bases de datos.

Permite extraer datos de:

- PostgreSQL
- MySQL
- Oracle
- DB2
- MS-SQL



- Informix
- Internase
- Sybase
- Frontbase

La información, para ser visualizada permite exportarla a:

- PostScript
- Texto
- HTML
- XML
- PDF
- CSV.

Esta es una herramienta potente y especializada en la importación desde las bases de datos.

### 1.2.5 Jasper Reports.

**Jasper Reports** es una herramienta gratuita y de código abierto que se compone de un conjunto de librerías Java para facilitar la generación de informes. Es capaz de extraer la información desde diferentes bases de datos instaladas, o ficheros (Excel y Acces).

Uno de los puntos fuertes de **Jasper Reports** es el apoyo creciente de la comunidad, se espera su desarrollo posterior a diferentes versiones más potentes.

Cada reporte que se genera tiene definido su diseño en un archivo XML, por convención su extensión es .jrxml, o sea la herramienta crea una extensión de ficheros propia. Es capaz de generar, compilar, modificar y exportar el fichero. Puede exportar a las siguientes extensiones:

- PDF
- XML
- HTML

- CSV
- XLS

Lo más interesante de esta herramienta es la creación de ficheros con extensión propia y la capacidad de exportar a diferentes ambientes.

### 1.3 ESTRUCTURAS DE ALMACENAMIENTO DE DATOS.

El almacenamiento de datos y el acceso a ellos, se ha diversificado en el propósito de realizar estos procedimientos de manera más eficiente y funcional, acorde a las necesidades de los usuarios.

“La sofisticación de la tecnología moderna de las bases de datos es el resultado de la evolución que a lo largo de varias décadas ha tenido lugar en el procesamiento de los datos y en la gestión de la información.” [1] Por esta razón, el desarrollo de herramientas que posibiliten el acceso a los datos reviste hoy vital importancia. Acoplar herramientas a sistemas y software de análisis y procesamiento de datos, que por ende permitan aumentar el alcance de acceso a datos de diversas fuentes es un procedimiento muy útil en la actualidad. Una herramienta de este tipo constituye el primer paso en la elaboración de un software general de auditoría. Pues el proceso de importación de datos al ambiente propio del software sería la funcionalidad básica que daría paso a las demás. Entender por tanto, como se estructuran los datos y se guardan en las bases de datos es esencial. “Una base de datos o banco de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada.” [2]

Los datos se estructuran a los más bajos niveles de las bases de datos, en término de definiciones abstractas, como tablas, campos y registros.

Las tablas en las bases de datos, se refieren al tipo de modelamiento de los datos, donde se guardan los datos recolectados por un programa. Su estructura general se asemeja a la vista general de un programa de hoja de cálculo. Tabla de una base de datos “es un conjunto de registros homogéneos con la misma estructura.” [3]

Las tablas se componen de dos estructuras:

- **Campo:** Corresponde al nombre de la columna. Debe ser único, además de tener un tipo de dato asociado.

- **Registro:** Corresponde a cada fila que compone la tabla.

Los campos son los distintos tipos de datos que componen la tabla, por ejemplo: nombre, apellido, domicilio. La definición de un campo requiere: el nombre del campo, el tipo de campo, la longitud del campo, su descripción, entre otros. A su vez, los registros constituyen la información que va contenida en los campos de la tabla, por ejemplo: el nombre del paciente, su apellido y la dirección de este. Generalmente los tipos de campos más importantes que se pueden almacenar son los siguientes: Texto (caracteres), Numérico (números), Fecha / Hora, Lógico (informaciones lógicas si/no, verdadero/falso).

Se puede hablar de propiedades características o campos característicos, y propiedades secundarias o campos secundarios según definan o complementen el elemento representado por el registro. Cuando se tienen varias tablas o ficheros con algún campo en común, entonces pueden relacionarse y constituirse una base de datos relacional.

En informática, y concretamente en el contexto de una base de datos relacional, un registro (también llamado fila o tupla) representa una unidad única de datos implícitamente estructurados en una tabla. En términos simples, una tabla de una base de datos puede imaginarse formada de filas y columnas o campos. Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura.

## **1.4 FUNCIONAMIENTO DE LAS HERRAMIENTAS DE IMPORTACIÓN DE DATOS.**

Las herramientas de importación de datos generalmente pueden trabajar archivos csv. "Csv es un formato estándar de archivos de texto que típicamente se compone de filas donde los campos de cada fila están entrecomillados y separados entre si por comas" [4]. Esto es configurable y por ejemplo podemos indicar que la separación entre campos se haga por tabulaciones, espacios en blanco o puntos y/o comas.

Este formato de archivo es muy fácil de editar y manipular, por ejemplo se puede utilizar una aplicación de hoja de cálculo como Excel para editar y guardar nuestros datos. Anteriormente tiene que haberse

conseguido exportar los datos desde la antigua aplicación a formato de Excel para poder abrirlo desde dichas aplicaciones.

Una vez que se tienen los archivos csv, debe lograrse que esos datos se amolden al diseño de tablas de la aplicación de destino (hacia la que realiza la importación). El objetivo es que las columnas de los archivos csv acoplen con los campos de la correspondiente tabla de nuestra aplicación, hacia donde se van a importar los registros.

Para lograr este objetivo, lo primero es tener definido el listado de columnas del archivo csv que se creará, y qué representa cada una de ellas. La primera fila representa la cabecera y la segunda el primer y único registro. Luego se tendría una serie de campos, uno por cada columna, separada por una tabulación. Generalmente se utilizan unas tablas intermedias para la importación. Deben saberse también, los campos que tienen estas tablas para poder mapear el archivo de destino. Con toda esta información, ya se está en disposición de realizar el mapeo. Lo ideal es que se cree un esquema de cómo se va a realizar para posteriormente, poder definir el formato de importación propio del archivo que se creará. Una vez que termine este proceso podrán verse todos los registros en la base de datos propia de la aplicación.

Lo esencial en este proceso es definir bien el mapeo de datos, la ventaja que ofrece este formato es que al trabajar con archivos de texto csv se pueden utilizar herramientas de manipulación de archivos.

## **1.5 METODOLOGÍA A UTILIZAR.**

Muchas organizaciones de desarrollo de software, desconocen la validez de la metodología aplicada durante el proceso de desarrollo, sin embargo, esta es una cuestión esencial si se quiere obtener el resultado esperado, y ganar en tiempo y calidad del producto final. Todo proceso de desarrollo de sistemas informáticos, es riesgoso y difícil de controlar. Si no se aplica una metodología adecuada, se tendrán clientes insatisfechos al final del trabajo realizado.

Una metodología de desarrollo de software ofrece un conjunto de técnicas y procedimientos que permiten conocer la organización de los elementos necesarios para definir un proyecto de software.

A veces, se realiza el diseño del software de manera rígida, con los requerimientos que el cliente solicita, de tal manera que cuando el cliente en la etapa final (etapa de prueba), solicita un cambio se hace muy difícil realizarlo, pues altera otras definiciones de otras fases, y es justo éste, uno de los factores que ocasiona un atraso en el proyecto, la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido. Para evitar estos incidentes, se debe seleccionar cuál metodología utilizar, teniendo en cuenta, la que se adapta más al medio en que se desarrolla y a los objetivos finales que queremos cumplir.

Decidimos aplicar el **Proceso Unificado Racional (RUP)** por sus siglas en inglés) como metodología para el desarrollo de este proyecto acompañado de la herramienta **Rational Rose** y el **Lenguaje Unificado de Modelado (UML)** por sus siglas en inglés) como lenguaje de modelado visual.

Con **RUP**, se logra una mayor productividad por parte del equipo de trabajo, ya que se definen claramente sus actividades, roles y sus responsabilidades. Está basado en componentes y utiliza el **Lenguaje Unificado de Modelado (UML)** por sus siglas en inglés) para la construcción de todos los esquemas necesarios durante el proceso de desarrollo de software.

### **1.5.1 El Proceso Unificado de Desarrollo de Software RUP.**

“El **Proceso Unificado** es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software.”[5] El logro de este objetivo final comprende la organización de un conjunto de actividades encaminadas a transformar los requisitos del sistema, como consecuencia de las peticiones de los usuarios, en un software.

**RUP** conjuntamente con el **Lenguaje Unificado de Modelado (UML)** constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. **RUP** utiliza **UML** para modelar los esquemas de un sistema, es por esto que **UML** es una parte esencial del proceso unificado.

**RUP y UML** están estrechamente relacionados entre sí, pues mientras el primero establece las actividades y los criterios para conducir un sistema desde su máximo nivel de abstracción (la idea del cliente), hasta su nivel más concreto (un programa ejecutándose en las instalaciones del cliente), el segundo ofrece la notación gráfica necesaria para representar los sucesivos modelos que se obtienen en el proceso de refinamiento.

## 1.5.2 Características de RUP.

**RUP** tiene tres características principales que rigen el proceso de desarrollo. Estas características, deben ser tenidas en cuenta durante todo el proceso de desarrollo.

### 1.5.2.1 Iterativo e Incremental.

El proceso de desarrollo de un software está decisivamente influenciado por el factor tiempo. Para optimizar el trabajo “es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento.” [6]

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas:

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** El resultado final de esta fase es un producto listo para su utilización. Se obtiene uno o varios release (candidato definitivo o candidato para el lanzamiento; se refiere a un producto final, preparado para lanzarse como versión definitiva) del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
- **Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Cada una de estas fases es a su vez dividida en una serie de iteraciones. Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema.

Cada una de estas iteraciones se divide en una serie de flujos de trabajo:

- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Despliegue e Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, entre otras) para entregar el software a los usuarios finales.
- **Gestión del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Gestión de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos y control de versiones.

Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

### 1.5.2.2 Dirigido por casos de uso.

Los casos de uso reflejan lo que los futuros usuarios necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

### 1.5.2.3 Centrado en la arquitectura.

La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. **RUP** se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura. El modelo de

arquitectura se representa a través de vistas (Vista de Casos de Uso, Vista Lógica, Vista de Procesos, Vista de Implementación y Vista de Despliegue) en las que se incluyen los diagramas de **UML**.

Como otra ventaja, el Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

### **1.5.3 UML: Lenguaje Unificado de Modelado.**

El **Lenguaje Unificado de Modelado (UML)** es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. **UML** “se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas”. [7]

**UML** está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el **UML** es un lenguaje, cuenta con reglas para combinar tales elementos.

Es importante recalcar que **UML** no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. **UML** es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. En lo que corresponde al desarrollo de programas, posee elementos gráficos para soportar la captura de requisitos, el análisis, el diseño, la implementación, y las pruebas. **UML** describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo, sus objetivos son ofrecer un lenguaje simple y legible que permitiera modelar aplicaciones en cualquier dominio, y generar de manera automática código fuente.

**UML** es un lenguaje capaz de abstraer cualquier tipo de sistema (informático o no), mediante los diagramas, esto es, mediante representaciones gráficas que contienen toda la información relevante del sistema. Para lograrlo, utiliza distintos tipos de diagramas, como los diagramas de implementación, diagramas de comportamiento o interacción, diagramas de casos de uso y diagramas de clases.

Las principales funciones de **UML** son:

- **Visualizar:** **UML** permite expresar de una forma gráfica un sistema de forma que otra persona lo puede entender.
- **Especificar:** **UML** permite especificar cuáles son las características de un sistema antes de su construcción.



- **Construir:** A partir de los modelos especificados se puede construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado, sirviendo para su futura revisión.

Un modelo **UML** está compuesto por tres clases de bloques de construcción. El primero de ellos proporciona los elementos, es decir, las abstracciones de cosas reales o ficticias como por ejemplo objetos y acciones; luego se encuentran las relaciones, esto es la relación entre los elementos (la cual puede ser de dependencia, asociación, generalización, entre otras); finalmente, se encuentran los diagramas, que son colecciones de elementos con sus relaciones.

Los elementos de **UML** pueden ser estructurales, de comportamiento de agrupación o de anotación:

- **Elementos Estructurales:**
  - **Actor:** "Algo" o "alguien" que puede interactuar con el negocio o sistema que se está desarrollando.
  - **Casos de Uso:** Descripción de un conjunto de secuencias de acciones que un negocio o sistema ejecuta y que produce un resultado observable de interés para un actor particular.
  - **Clase:** Corresponde a una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.
  - **Objetos:** Un objeto es una instancia de alguna clase.
- **Elementos de Comportamiento:**
  - **Mensajes:** Los mensajes se usan para especificar una comunicación entre objetos.
- **Elementos de Agrupación:**
  - **Paquete:** Sirve para organizar elementos en grupos. Un paquete es puramente conceptual (sólo existe en tiempo de desarrollo).

**UML** representa para los desarrolladores de aplicaciones y sistemas una serie de ventajas, al igual que para las organizaciones, entre estos beneficios se destacan:

- Produce un aumento en la calidad del desarrollo.

- Mejora en un 50% o más los tiempos totales de desarrollo.
- Permite especificar la estructura y el comportamiento del sistema.
- Permite dimensionar mejor los riesgos de un proyecto, tener un mejor rendimiento antes de construir el sistema.
- Facilita la documentación de las decisiones de la arquitectura del proyecto.
- Ofrece un mejor soporte a la planificación y control del proyecto.
- Ofrece mayor rigurosidad en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y viceversa.
- Soportado por muchas herramientas.
- 

## 1.6 HERAMIENTAS UTILIZADAS: RATIONAL ROSE.

El uso de la herramienta adecuada, que soporte y complemente el proceso de desarrollo, es muy importante. Definir el uso de una herramienta potente y que permita desarrollar todas las actividades definidas durante el proceso, es de vital importancia para el cumplimiento de los objetivos finales.

El **Rational Rose** es una herramienta que cubre todo el ciclo de vida de un proyecto, desde la concepción y formalización del modelo, construcción de los componentes, transición a los usuarios hasta la certificación de las distintas fases y entregables. Permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (Vista de Casos de Uso, Vista Lógica, Vista de Componentes y Vista de Despliegue) y utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. Proporciona mecanismos para realizar la denominada Ingeniería Inversa, a partir del código de un programa se puede obtener información de su diseño.

Es la **herramienta CASE** (Computer Aided Software Engineering) que comercializan los desarrolladores de **UML** y que soporta de forma completa la especificación del **UML**.

Como desventaja del **Racional** podemos decir que no trae muchos estereotipos definidos.

## 1.7 TÉCNICA PARA VALIDACIÓN DE REQUISITOS: PROTOTIPO.

Un prototipo es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso. Puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos a un complejo software.

### Ventajas del prototipo:

- Son útiles para comunicar, discutir y definir ideas entre los diseñadores y las partes responsables.
- Apoyan el trabajo evaluando productos, clarificando requisitos de usuario y definiendo alternativas.
- 

## 1.8 CONCLUSIONES.

En este capítulo se han esgrimido los elementos teóricos fundamentales que conducen la investigación. Abordamos en el estado del arte los principales software general de auditoría y herramientas de importación de ficheros más utilizadas en el mundo. Además se explica sobre las estructuras de almacenamiento de la información más comunes.

A partir de que explicamos cuál será la metodología que regirá todo el proceso de desarrollo, así como las herramientas y lenguajes de modelado que utilizaremos, se definieron las tecnologías y herramientas a utilizar, llegando a las siguientes conclusiones:

Usar el **Proceso Unificado de Rational (RUP)** como metodología para el desarrollo de este proyecto acompañado de la herramienta **Rational Rose** y **UML** como lenguaje de modelado visual.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### **2.1 INTRODUCCIÓN.**

Uno de los estados más importantes a la hora de construir un sistema, es saber qué se va a construir. Ninguna otra parte del trabajo conceptual es tan compleja y definitoria como establecer los requisitos técnicos detallados, incluyendo todas las interfaces con las personas, máquinas, y otros sistemas. El impacto negativo que pudiera tener esta parte del trabajo de ingeniería si no es realizado con eficiencia en el resto del proceso de desarrollo, será determinante en los factores tiempo y calidad.

Entonces, la tarea más importante que el Ingeniero de Software ejecuta llegado este punto, en función del cliente, es la extracción iterativa y el refinamiento de los requisitos del sistema, seguido a ello identifica los casos de uso del sistema, los cuales constituyen, precisamente, la guía del equipo de desarrollo que lleva adelante la implementación del mismo. Sin dudas, la efectiva identificación y especificación de los casos de uso del sistema, representa un factor determinante para lograr un producto exitoso.

En este capítulo, se realiza la descripción de las principales definiciones asociadas al dominio del problema y que son necesarias para entender el negocio y la propuesta de solución, llegando a elaborar el modelo del dominio. Además se definen los conceptos más importantes a manipular en el resto del documento y se determinan los requisitos tanto funcionales como no funcionales. Por otra parte, se tratarán los artefactos definidos en la metodología RUP referentes al flujo de trabajo de requerimientos: identificación y descripción de los actores del sistema, definición de los casos de uso del sistema (CUS), y se ilustra además el modelo de CUS. En este capítulo además se realiza la descripción de los casos de uso del sistema.

### **2.2 PROPUESTA DE SOLUCIÓN.**

Del análisis del estado del arte y las principales funcionalidades identificadas en los softwares de este tipo, se elaboró una propuesta de solución. El proceso de importación de ficheros y manejo de campos dentro de un software general de auditoría es un proceso complejo, y a la vez resulta un procedimiento fundamental para el funcionamiento del programa, pues lo primero es tener los datos en el formato de su plataforma, para posteriormente realizar los análisis e interpretaciones deseados por el usuario.

Así, se han identificado las cuatro formas principales en que se realizará la importación, serán analizadas de manera secuencial y jerárquica, comenzando por:

- Identificar si el fichero a importar es de extensión conocida, para ello, la plataforma debe contar con un archivo de registros con información sobre los ficheros más comunes desde los que se importan los datos en una auditoría. En este caso, se denominan ficheros de formatos predefinidos los siguientes: .xls, .dbf, mdb, pdf, xml.

En el primer caso, si se trata de importar un fichero mediante el ODBC, se seguirá un algoritmo diferente:

- Realizar la importación mediante el ODBC de la base de datos, en este caso, el software general de auditoría tomará directamente los datos de una base de datos especializada. Se establecerá la comunicación entre el sistema operativo y la base de datos mediante el ODBC o drivers de la base de datos. Las tablas serán importadas a la plataforma donde se manejarán los datos.
- Identificar si el fichero a importar es un fichero de datos no formateados, en esta categoría se definen: ASCII de longitud fija, ASCII delimitado, EBCDIC de longitud fija o un caso especial en que no se pueda determinar el tipo de archivo origen. En este caso, se tendrá en cuenta, la posible conversión de caracteres a símbolos ASCII.

Para el manejo de campos, se definieron las funcionalidades:

- Crear un nuevo campo.
- Modificar un campo.
- Eliminar un campo.

Con los campos, en el proceso de importación de datos no formateados se permitirá:

- Crear un delimitador de campos.
- Mover un delimitador de campos.
- Eliminar un delimitador de campos.

En esencia, dentro de estas variantes se puede dar respuesta al proceso de importación de ficheros hacia el ambiente propio del software general de auditoría, y el manejo de los campos en el ambiente propio del software, pues estarían tratados todos los posibles casos y variantes a enfrentar.

## **2.3 MODELO DEL DOMINIO.**

Este modelo, permite mostrar de manera visual los principales conceptos que se manejan en el dominio del software en desarrollo y de esta forma, utilizar un vocabulario común que ayude tanto a usuarios, clientes, desarrolladores e interesados de manera general, a entender la lógica de funcionamiento de los procesos relacionados con el negocio, logrando una captura correcta y consensuada de requisitos.

### **2.3.1 ¿Qué es un modelo del dominio?**

“Una de las primeras actividades centrales de un ciclo de desarrollo consiste en crear un modelo conceptual para los casos de uso, en el cual se explican a sus creadores los conceptos significativos en un dominio del problema” [8]. Tomando esta experiencia, el modelo del dominio define un modelo de clases común para todos los implicados en el desarrollo, representadas en objetos del dominio, sirve como interlocutor entre clientes y desarrolladores. El propósito fundamental de este modelo es generar una terminología común y sentar las bases del entendimiento del desarrollo y no para definir el sistema completo.

### **2.3.2 ¿Por qué hacer un modelo del dominio?**

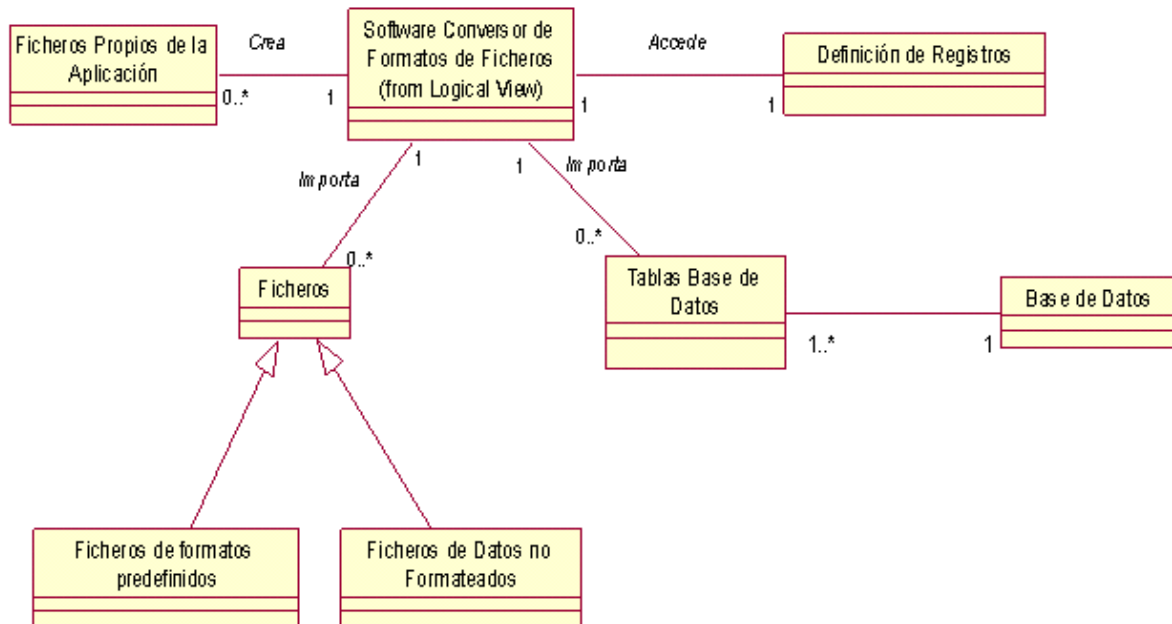
La estructuración de los procesos de importación de ficheros a software específicos es diversa. De tal manera, las formas y procedimientos que se establecen son variables. Los procesos de importación de ficheros en un software general de auditoría son procesos internos del sistema, por tanto, se están modelando procesos cuyo nivel de estructuración respecto a los usuarios es bajo.

Considerando tales escenarios y según **RUP**, cuando esto sucede, se recomienda desarrollar un modelo del dominio, seguido como alternativa al modelo de casos de uso del negocio, identificando a la vez los conceptos más significativos del problema en el mundo real , definiéndolos y tratando de unir y/o relacionar los mismos.

### 2.3.3 Definición de las entidades y conceptos principales.

Ficheros Propios de la Aplicación.	Son los ficheros que se crean dentro de la propia aplicación, y donde se guardan los datos importados para su posterior análisis.
Base de Datos	Es un conjunto de datos pertenecientes a un mismo contexto y almacenados de manera consistente y sistemáticamente para su posterior uso.
Definición de Registro.	Registro que contiene la estructura de los registros de otros ficheros mediante el cual se realiza el análisis de los archivos de datos a importar.
Software Conversor de Formatos de Ficheros.	Aplicación que realiza la conversión de formato de los ficheros.
Tablas de BD	Es una estructura contenedora de datos compuesta por registros (las filas de una tabla) y campos (las columnas de una tabla).
Ficheros	Estructura lógica representada por registros y campos donde se almacena algún tipo de información.
Ficheros de Formato Predefinidos.	Son los ficheros que se han definido en el Registro de Definiciones de la aplicación y que representan los tipos de ficheros más comunes que se encuentran en el proceso de importación.
Ficheros de Datos no Formateados.	Son los ficheros que no se encuentran definidos en el Registro de Definiciones de la aplicación.

### 2.3.4 Diagrama del Modelo del Dominio.



*Fig.2.1 Modelo del dominio.*

## 2.4 ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE.

Hacer una identificación exhaustiva y correcta de los requisitos fundamentales que debe cumplir el sistema es muy importante. “Un proyecto no puede ser exitoso sin una especificación correcta y exhaustiva de los requerimientos.” [9]

Una vez que se han definido los conceptos principales relacionados con el objeto de estudio y el dominio, se puede comenzar a analizar qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo. “Para crear una aplicación de software hay que describir el problema y las necesidades o requerimientos: en qué consiste el conflicto y qué debe hacerse.” [10]



Para ello, se enumeran a través de requerimientos funcionales y no funcionales, las acciones que el sistema deberá ser capaz de realizar, así como las exigencias de despliegue para su uso efectivo. Las acciones que podrán ejecutar el usuario y el ambiente externo en que se usará el software son explicadas a continuación.

### 2.4.1. Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. En este caso el proceso resulta complejo, pues se combina la captura de requisitos que ya son funcionalidades de software, o sea procesos que ocurren de manera digital, con la interpretación de cómo son ejecutados estos procesos. Se siguió la técnica de la entrevistas con el cliente para capturar sus exigencias de cómo deben mostrarse los resultados de los procesos ya automatizados, y se estudiaron los softwares del estado del arte, para mediante la ingeniería inversa también identificar requisitos. La ingeniería inversa es la técnica empleada para descubrir los principios tecnológicos detrás de un determinado objeto. En este caso, para descubrir las funcionalidades de un software. Con estas dos técnicas se identificaron los siguientes requisitos funcionales.

#### **RF1- Importar ficheros de formatos predefinidos.**

- 1.1- Seleccionar extensión del fichero a importar.
- 1.2- Mostrar datos a importar en la Ventana de Base de Datos.
- 1.3- Especificar fila que contiene el nombre de los campos.
- 1.4- Mostrar la base de datos importada.

#### **RF2- Cambiar configuración de la columna de datos.**

#### **RF3- Importar ficheros de datos no formateados.**

- 3.1- Determinar el tipo de archivo. (ASCII de longitud fija, ASCII delimitado, dBase, EBCDIC de longitud fija).
- 3.2- Mostrar datos a importar en la Ventana de Base de Datos.
- 3.3- Especificar longitud del registro.
- 3.4- Crear definición de registro.
- 3.5- Guardar definición de registro

**3.6-** Mostrar la base de datos importada.

**RF4-Editar Ecuaciones.**

**RF5-Importar Datos mediante ODBC.**

**RF6- Gestionar campos.**

**6.1-** Crear campos.

**6.2-** Modificar campos.

**6.3-** Eliminar campos.

**RF7-Gestionar delimitadores de campos.**

**7.1-** Crear línea de campo.

**7.2-** Quitar línea de campo.

**7.3-** Mover línea de campo.

**RF8- Mostrar estadísticas de campos.**

**RF9- Secuenciar los registros.**

**9.1-** Ordenar

**9.2-** Indexar

## 2.4.2 Requerimientos no funcionales.

Los requerimientos no funcionales aunque no aportan funcionalidades propiamente dichas dentro de la aplicación, son de vital importancia para una puesta en marcha exitosa del software, y para lograr que este responda a las expectativas del usuario. No se deben descuidar las exigencias en cuanto a este tipo de requerimientos que determinan el ambiente donde se realizará el despliegue del producto. Estos requerimientos, son cualidades o propiedades que el producto debe cumplir, las que conforman una aplicación entendible, confiable y fácil de utilizar por los usuarios.

**Apariencia o interfaz externa:**

- El sistema debe tener una interfaz entendible y amigable para los usuarios, lo que permitirá que pueda ser utilizado sin mucho entrenamiento por los mismos, con conocimientos básicos como operadores de computadoras y no como programadores.
- Estará diseñado para cualquier tipo de resolución.
- El sistema se mostrará como una aplicación de escritorio.

**Usabilidad:**

- El sistema debe estar disponible las 24 horas del día.
- El sistema podrá ser usado por cualquier persona que posea conocimientos mínimos de procesos de auditoría y control, y conocimientos medios en el manejo de las computadoras y las bases de datos.
- El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.

**Soporte:**

- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos.
- Se necesita una microcomputadora que soporte grandes volúmenes de datos y velocidad de procesamiento (128 Mb de memoria RAM).

**Seguridad:**

- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).
- Garantizar que la información sea editada únicamente por quien tiene derecho a editarla.
- Emplear técnicas para asegurar la integridad de los datos.
- Utilizar la seguridad integrada del sistema operativo en uso.

**Portabilidad:**

- El sistema debe ser multiplataforma, haciendo énfasis en Linux y Windows.

**Legales:**

- El empleo del sistema debe estar regido por un manual de normas y procedimientos que debe ser aprobado por la Dirección de la Universidad y estar basado en las disposiciones legales vigentes.

**Software:**

- Sistema operativo Linux, Windows 98 o superior.
- ODBC de los sistemas gestores de bases de datos instalado.

**Hardware:**

- Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.  
Capacidad de Disco Duro superior a 10 Giga bytes.

**Mantenimiento y actualización:**

- La capacidad de actualización es la habilidad para entregar nuevas versiones del producto a bajo costo a los clientes con un mínimo de tiempo de descarga o interrupción. Una característica clave para apoyar este objetivo es la descarga automática de parches o al equipo del usuario final.

## 2.5 DEFINICIÓN DE LOS ACTORES DEL SISTEMA.

Identificar quién o quiénes serán las entidades que interactuarán con nuestro sistema es el paso que sigue. “El actor es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso. Por lo regular estimula el sistema con eventos de entrada o recibe algo de él.” [11]

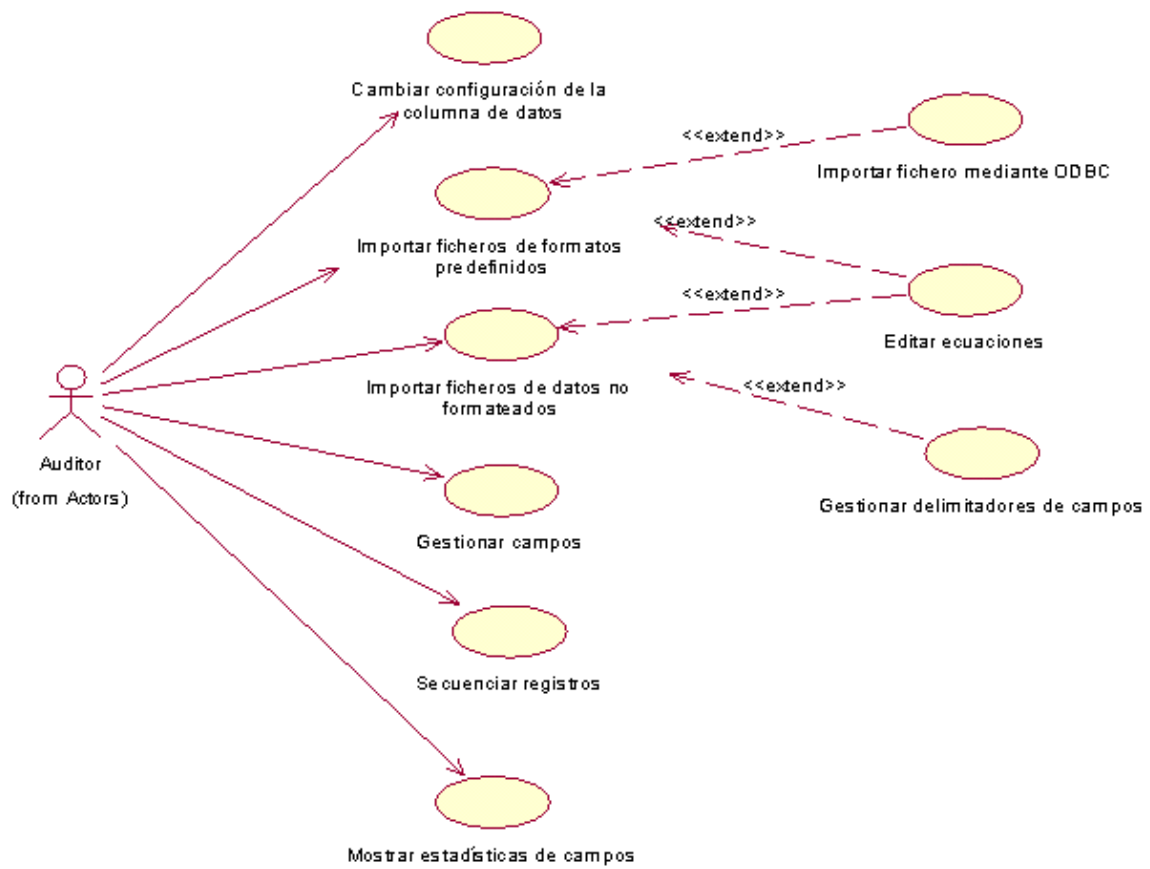
Actor.	Justificación.
Auditor	Es un usuario que accede al Sistema y puede importar datos solicitando la ejecución de las distintas funcionalidades.

## 2.6 DIAGRAMA DE CASOS DE USO DEL SISTEMA.

Un caso de uso describe un proceso que deberá convertirse en una funcionalidad. “Un proceso describe, de comienzo a fin, una secuencia de eventos, de las acciones y las transacciones que se requieren para producir u obtener algo de valor para una empresa o actor.” [12]


Un diagrama de casos de uso del sistema representa gráficamente las funcionalidades principales del sistema y su interacción con los actores.

A continuación quedaran representados en un diagrama de casos de uso del sistema, los casos de uso identificados.



**Fig.2.2** Diagrama de Casos de Uso del Sistema.

## 2.6.1 Descripción de los Casos de Uso del Sistema.

<b>CU-1</b>	
<b>Importar ficheros de formatos predefinidos.</b>	
<b>Actores</b>	Auditor.
<b>Propósito</b>	Realizar la importación.
<b>Resumen:</b> El caso de uso se inicia cuando el Auditor solicita realizar la importación de un fichero, seleccionando su ubicación en el explorador de archivos. Luego, selecciona la extensión del fichero (*.xls, *.dbf, *.mdb, *.pdf, *.xml) Este caso de uso termina cuando el fichero es importado satisfactoriamente, mostrando al Auditor la base de datos importada.	
<b>Referencias:</b> RF1	
<b>Precondiciones:</b> Debe existir al menos un fichero origen de extensión predefinida a importar.	
<b>Poscondiciones:</b> Se crea un fichero dentro de la aplicación con la base de datos importada.	
<b>Prototipo</b>	
<b>Pantalla 1</b>	
 A screenshot of a Windows-style dialog box titled "Importar ficheros formatos predefinidos.". The dialog box has a light gray background and a blue title bar. Inside, there is a group box labeled "Formatos" containing six radio button options: XLS (selected), DBF, MDB, PDF, XML, and ODBC. At the bottom of the dialog box, there are two buttons: "Importar" and "Cancelar".	
<b>Descripción.</b>	
<b>Flujo Normal de Eventos.</b>	

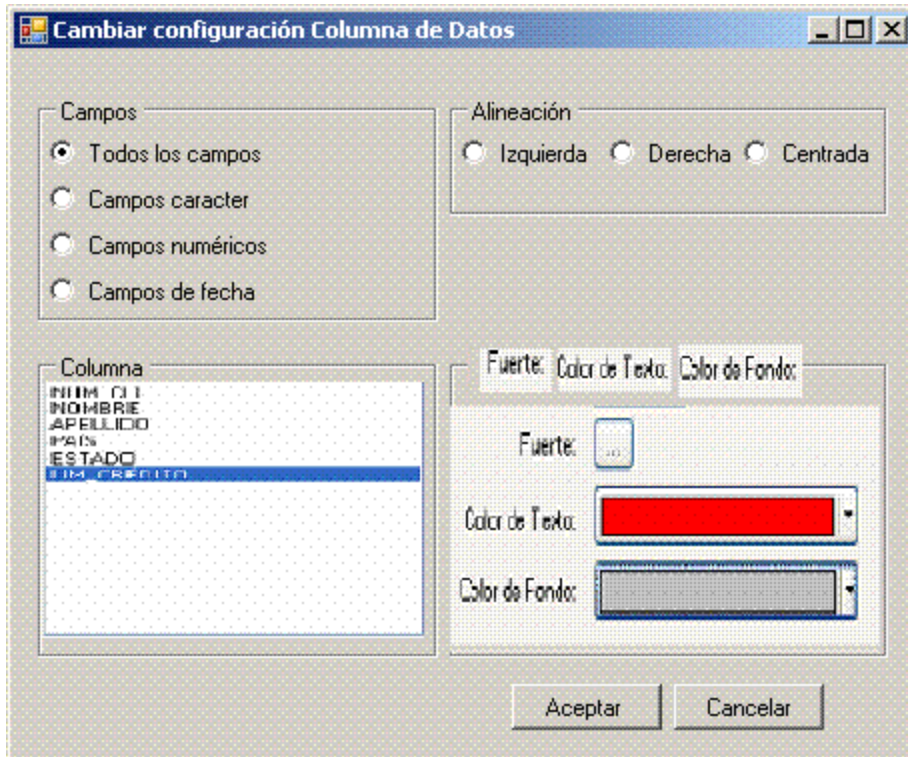
Acción del Actor.	Respuesta del Sistema.
1. El Auditor selecciona la opción Importar Fichero de formato predefinido.	2. El sistema muestra una ventana con las extensiones de ficheros definidas como conocidas (*.xls, *.dbf, *.mdb, *.pdf, *.xml).
3. El Auditor especifica la extensión del fichero a importar.	4. El sistema verifica que la definición de registro del fichero origen coincida con la definición de registro del sistema para ficheros de esa extensión.
	5. El sistema muestra una tabla con los campos importados.
6. El Auditor puede seleccionar una de las siguientes opciones: <ul style="list-style-type: none"> <li>• Importar datos mediante ODBC.</li> <li>• Editar ecuación.</li> <li>• Especificar fila que contiene el nombre de los campos.</li> </ul>	7. Si el Auditor seleccionó: <ul style="list-style-type: none"> <li>• “Importar datos mediante ODBC.” Ver descripción del caso de uso “Importar datos mediante ODBC.”</li> <li>• “Editar ecuación.” Ver descripción del caso de uso “Editar ecuación.”</li> <li>• “Especificar fila que contiene el nombre de los campos” Ver sección “Especificar fila que contiene el nombre de los campos.”</li> </ul>
<b>Sección Especificar fila que contiene el nombre de los campos.</b>	
Acción del actor	Respuesta del sistema
8. Especificar fila que contiene el nombre de los campos	9. El sistema muestra una ventana con la solicitud de la especificación de la fila que contiene el nombre de los campos (Recomendando sea en la fila 1).
10. El Auditor especifica la fila que contiene el nombre de los campos.	11. El sistema muestra un mensaje diciendo si está seguro que desea especificar esa fila.
12. El Auditor selecciona	



"Aceptar".	
Sección <b>Mostrar base de datos importada.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	13. El sistema muestra en la ventana de trabajo de datos, la tabla con todos sus campos y las modificaciones realizadas que se guardará en el fichero de la aplicación.
<b>Flujo alternativo.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Acción 3 F1</b> Si se selecciona una extensión cuya definición de registro no coincida con la definición de registros de la aplicación.	<b>3 F1.1</b> El sistema muestra el mensaje "Debe seleccionar el tipo de fichero correcto".
<b>Acción 10 F2</b> Si se teclea un valor no numérico.	<b>10 F2.</b> El sistema le da valor 1.
<b>Requerimientos no Funcionales</b>	
<b>Prioridad</b>	

<b>CU-2</b>	
<b>Cambiar configuración de la columna de datos.</b>	
<b>Actores</b>	Auditor
<b>Propósito</b>	Cambiar los atributos: color, tamaño, tipo de letra y alineación de una columna de datos.
<b>Resumen:</b> El caso de uso se inicia cuando el Auditor solicita cambiar la configuración de la columna de datos, definiendo los nuevos valores de los atributos: color, tamaño, tipo de letra y alineación. Este caso de uso termina mostrada la columna de datos con la nueva configuración.	
<b>Referencias:</b> RF2	
<b>Precondiciones:</b> La Base de Datos Activa debe contener al menos una columna de datos.	
<b>Poscondiciones:</b>	
<b>Prototipo</b>	

**Pantalla 2**



**Descripción.**

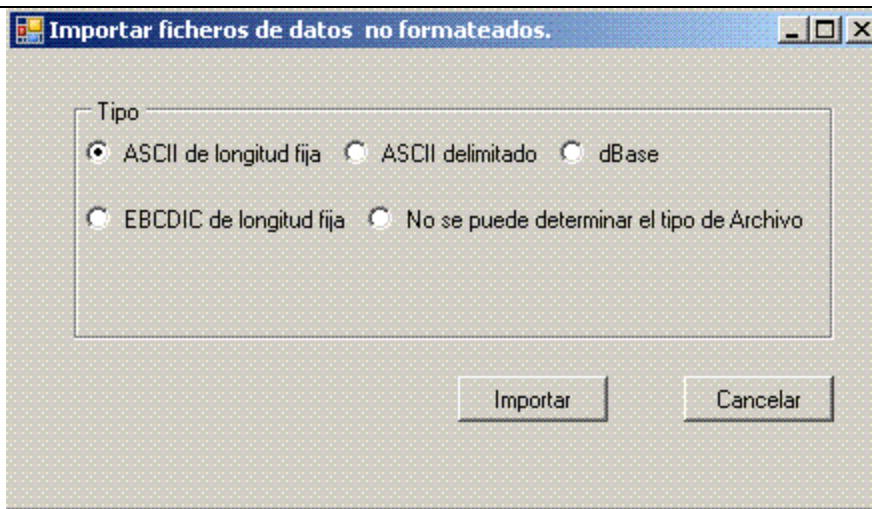
**Flujo Normal de Eventos.**

Acción del actor.	Respuesta del Sistema.
<p>1. El Auditor solicita cambiar la configuración de alguno de los atributos de la columna de datos.</p>	<p>2. El sistema muestra una ventana donde se solicita se especifique la columna a editar y el valor de los atributos: color de la columna de datos, alineación, tamaño y tipo de letra.</p>
<p>3. El Auditor puede seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Cambiar color de la columna de datos.</li> <li>• Cambiar alineación de la columna de datos.</li> <li>• Cambiar tamaño de letra de la columna de datos.</li> <li>• Cambiar tipo de letra de la</li> </ul>	<p>4. Si el Auditor seleccionó:</p> <ul style="list-style-type: none"> <li>• “Cambiar color de la columna de datos.” Ver sección: “Cambiar color de la columna de datos.”</li> <li>• “Cambiar alineación de la columna de datos.” Ver sección: “Cambiar alineación de la columna de datos.”</li> <li>• “Cambiar tamaño de letra de la columna de datos.” Ver sección: “Cambiar tamaño</li> </ul>

columna de datos.	de letra de la columna de datos.”  <ul style="list-style-type: none"> <li>• “Cambiar tipo de letra de la columna de datos.” Ver sección: “Cambiar tipo de letra de la columna de datos.”</li> </ul>
<b>Sección Cambiar color de la columna de datos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
5. El Auditor selecciona la opción Cambiar color de la columna de datos	6. El sistema muestra una ventana donde se solicita se especifique la columna a editar y el valor del atributo: color de la columna de datos.
7. El Auditor especifica el valor del atributo color de la columna de datos.	8. El sistema muestra en la Ventana de Base de Datos la columna editada con el nuevo valor del atributo.
<b>Sección Cambiar alineación de la columna de datos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
9. El Auditor selecciona la opción Cambiar alineación de la columna de datos	10. El sistema muestra una ventana donde se solicita se especifique la columna a editar y el valor del atributo: alineación de la columna de datos.
11. El Auditor especifica el valor del atributo alineación de la columna de datos.	12. El sistema muestra en la Ventana de Base de Datos la columna editada con el nuevo valor del atributo.
<b>Sección Cambiar tamaño de letra de la columna de datos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
13. El Auditor selecciona la opción Cambiar tamaño de letra de la columna de datos	14. El sistema muestra una ventana donde se solicita se especifique la columna a editar y el valor del atributo: tamaño de letra de la columna de datos.
15. El Auditor especifica el valor del atributo tamaño de letra de la columna de datos.	16. El sistema muestra en la Ventana de Base de Datos la columna editada con el nuevo valor del atributo.
<b>Sección Cambiar tipo de letra de la columna de datos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
17. El Auditor selecciona la opción Cambiar tipo de letra de la columna de datos	18. El sistema muestra una ventana donde se solicita se especifique la columna a editar y el valor del atributo: tipo de letra de la columna de datos.
19. El Auditor especifica el valor del atributo tipo de letra de la columna de datos.	20. El sistema muestra en la Ventana de Base de Datos la columna editada con el nuevo valor del atributo.

<b>Flujo alternativo.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Acción 2 F1</b> Si no se especifican los atributos de la columna de datos.	<b>2 F1.1</b> El sistema muestra el mensaje “Debe especificar los atributos que tendrá la columna de datos”.
<b>Requerimientos no Funcionales</b>	
<b>Poscondiciones</b>	
<b>Prioridad</b>	

<b>CU-3</b>	
<b>Importar ficheros de datos no formateados.</b>	
<b>Actores</b>	Auditor
<b>Propósito</b>	Realizar la importación.
<b>Resumen :</b> El caso de uso se inicia cuando el Auditor solicita realizar la importación de un fichero, seleccionando su ubicación en el explorador de archivos. Selecciona la opción indicada (ASCII de longitud fija, ASCII delimitado, dBase, EBCDIC de longitud fija o No se puede determinar el tipo de Archivo.) Este caso de uso termina cuando el fichero es importado satisfactoriamente, mostrando al Auditor la base de datos importada y cuando el Auditor guarda la definición de registro para usarla posteriormente si lo desea.	
<b>Referencias:</b> RF3	
<b>Precondiciones:</b> Debe existir al menos un fichero origen cuya extensión predefinida a importar.	
<b>Poscondiciones:</b> Se crea un fichero dentro de la aplicación con la base de datos importada.	
<b>Prototipo</b>	
<b>Pantalla 3</b>	



**Descripción.**

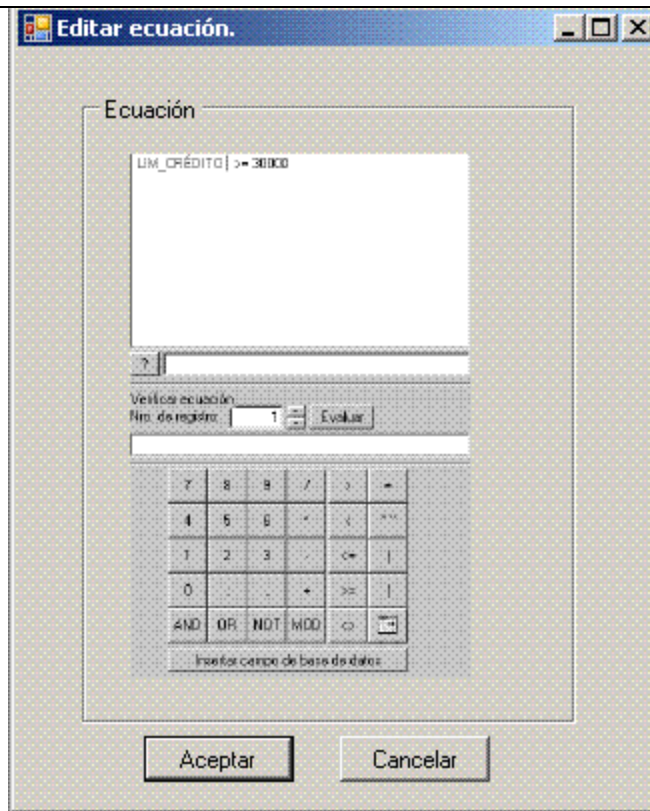
**Flujo Normal de Eventos.**

Acción del actor.	Respuesta del Sistema.
1. El Auditor selecciona la opción Importar Fichero de Datos no Formateados.	2. El sistema muestra una ventana con las opciones de tipo datos no formateados a importar (ASCII de longitud fija, ASCII delimitado, dBase, EBCDIC de longitud fija y No se puede Determinar el tipo de Archivo).
3. El Auditor especifica la opción de datos no formateados a importar.	4. El sistema verifica que si el Auditor modificó la selección del software es posible.
5. El Auditor puede seleccionar estas opciones: <ul style="list-style-type: none"> <li>• “Editar Ecuaciones. “</li> <li>• “Gestionar delimitadores de campos. ”</li> <li>• “Especificar longitud de registro”</li> <li>• “Guardar definición de registro.”</li> </ul>	6. Si el Auditor seleccionó: <ul style="list-style-type: none"> <li>• “Editar Ecuaciones.”Ver descripción del caso de uso “Editar Ecuaciones.”</li> <li>• “Gestionar delimitadores de campos. ” Ver descripción del caso de uso “Gestionar delimitadores de campos.”</li> <li>• “Especificar longitud de registro”. Ver sección “Especificar longitud de registro.”</li> <li>• “Guardar definición de registro.”Ver sección “Guardar definición de registro.”</li> </ul>

Sección **Especificar longitud de registro.**

<b>Acción del actor.</b>	<b>Respuesta del Sistema.</b>
7. El Auditor selecciona la opción Especificar la longitud de registro.	8. El sistema determina la longitud de registro correcta para este tipo de archivo.
	9. El sistema muestra una ventana con la longitud de registro calculada y permite la posibilidad de cambiarla.
10. El Auditor especifica la longitud de registro (se le recomienda la calculada).	11. El sistema muestra un mensaje diciendo si está seguro que esa es la longitud que desea especificar.
12. El Auditor selecciona la opción Aceptar.	
<b>Sección Mostrar la base de datos importada.</b>	
<b>Acción del actor.</b>	<b>Respuesta del Sistema.</b>
	13. El sistema muestra en la ventana de trabajo de datos, la tabla con todos sus campos y las modificaciones realizadas que se guardará en el fichero de la aplicación.
<b>Sección Guardar definición de registro.</b>	
<b>Acción del actor.</b>	<b>Respuesta del Sistema.</b>
14. El Auditor selecciona la opción Guardar Definición de Registro creada en durante el proceso de importación.	15. El sistema guarda la definición de registro creada para que pueda ser utilizada posteriormente. (Longitud de registro, delimitadores y campos creados).
<b>Flujo alternativo.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Acción 3 F1</b> Si selecciona una opción de datos no formateados no apropiada.	<b>3 F1.1</b> El sistema muestra un mensaje especificando que no puede realizar la importación con esa opción de datos no formateados.
<b>Requerimientos no Funcionales</b>	
<b>Poscondiciones</b>	
<b>Prioridad</b>	

<b>CU-4</b>	
<b>Editar ecuación. Extendido de los casos de uso Importar ficheros de formatos predefinidos e Importar ficheros de datos no formateados.</b>	
<b>Actores</b>	
<b>Propósito</b>	Realizar la edición de ecuaciones.
<b>Resumen :</b> El caso de uso se inicia cuando es invocado por alguno de los casos de uso: Importar ficheros de formatos predefinidos o Importar ficheros de datos no formateados, el Auditor solicita un criterio específico para realizar la importación. El criterio lo conforman una serie de condiciones matemáticas que se pueden aplicar sobre el total de los datos posibles a importar para finalmente quedarse con los datos deseados. Todas estas posibles restricciones tienen un orden y formato que queda plasmado en una ecuación. El caso de uso termina cuando queda editada y validada la ecuación para su puesta en práctica posteriormente sobre el volumen de datos a importar.	
<b>Referencias:</b> RF4, RF1, RF3	
<b>Precondiciones:</b> Debe existir al menos un fichero a importar y que contenga al menos un campo.	
<b>Poscondiciones:</b> Se guarda la base de datos importada de acuerdo a la ecuación descrita.	
<b>Prototipo</b>	
<b>Pantalla 4</b>	



**Descripción.**

**Flujo Normal de Eventos.**

Acción del Actor	Respuesta del Sistema.
	1. El sistema muestra una ventana con el prototipo del editor de ecuaciones.
2. El Auditor especifica la fórmula de la ecuación deseada y la verifica o no.	3. El sistema ejecuta la ecuación.
	4. El sistema verifica la ecuación si el Auditor lo pide y ejecuta la ecuación

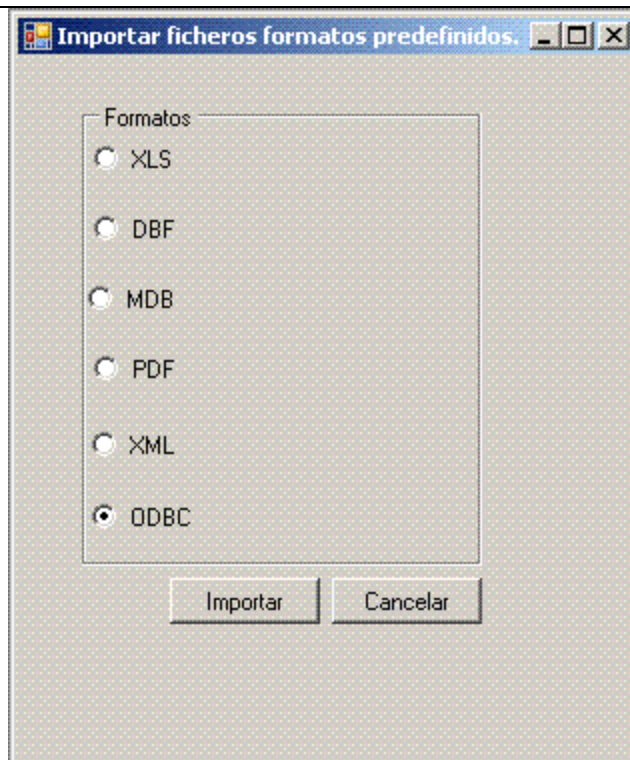
**Flujo alternativo.**

Acción del actor.	Respuesta del sistema.
<b>Acción 2 F1</b> Si se tecldea una ecuación no válida.	<b>2 F1.1</b> El sistema muestra el mensaje "Debe teclear una ecuación válida".



<b>Requerimientos no Funcionales</b>	
<b>Prioridad</b>	

<b>CU-5</b>	
<b>Importar datos mediante ODBC. Extendido del caso de uso Importar ficheros de formatos predefinidos.</b>	
<b>Actores</b>	
<b>Propósito</b>	Realizar la importación.
<b>Resumen :</b> El caso de uso se inicia cuando es invocado por el caso de uso Importar fichero de formato predefinido, entonces se selecciona su ubicación en el explorador de archivos. Luego, selecciona la opción (ODBC) lo que significa que lo hará mediante el ODBC de la Base de Datos instalada. Este caso de uso termina cuando la tabla solicitada es importada satisfactoriamente, mostrando al Auditor la base de datos importada.	
<b>Referencias:</b> RF5, RF1	
<b>Precondiciones:</b> Debe existir al menos una tabla en la Base de Datos seleccionada. Debe tener instalado un Gestor de Base de Datos y los drivers correspondientes.	
<b>Poscondiciones:</b> Se crea un fichero dentro de la aplicación con la base de datos importada.	
<b>Prototipo</b>	
<b>Pantalla 5</b>	



**Descripción.**

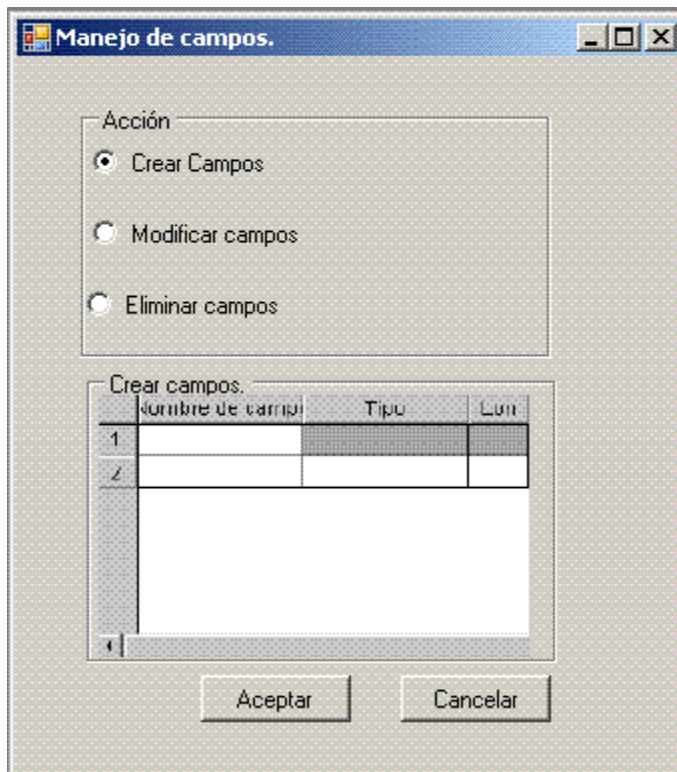
**Flujo Normal de Eventos.**

Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una ventana con las extensiones de ficheros definidas como conocidas (*.xls, *.dbf, *.mdb, *.pdf, *.xml, ODBC).
2. El Auditor especifica la opción de importar mediante el ODBC.	3. El sistema mediante un driver ODBC se conecta indirectamente a la Base de Datos y entra en ella.
	4. Luego la conexión a la Base de Datos es abierta con un script de conexión que contiene 3 parámetros: El proveedor BD (Provider), la fuente de datos (data source), la base de datos (database).
	5. El sistema muestra una ventana con las tablas contenidas en esa Base de Datos.

6. El Auditor selecciona la tabla que desea importar.	7. El sistema muestra una ventana con todos los campos y se le solicita al usuario seleccionar uno o varios campos que identifiquen de forma exclusiva a cada registro (hasta tres).
8. El Auditor selecciona el (los) campos que identifiquen de forma exclusiva cada registro.	9. Se muestra la tabla en la Ventana de Base de Datos.
<b>Flujo alternativo.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Acción 2 F1</b> Si se selecciona una extensión de fichero (*.xls, *.dbf, *.mdb, *.pdf, *.xml) y no la opción ODBC.	<b>2 F1.1</b> El sistema muestra el mensaje "Debe seleccionar el tipo de fichero correcto".
<b>Requerimientos no Funcionales</b>	
<b>Prioridad</b>	

<b>CU-6</b>	
<b>Gestionar campos.</b>	
<b>Actores</b>	Auditor
<b>Propósito</b>	Permitirá crear, modificar, eliminar campos.
<b>Resumen :</b> Este caso de uso se inicia cuando el Auditor solicita alguna de las opciones definidas en el manejo de los campos: crear, modificar y eliminar los campos. Este caso de uso termina el Auditor termina de crear, modificar o eliminar campos o cancela el caso de uso.	
<b>Referencias:</b> RF6	
<b>Precondiciones:</b>	
<b>Poscondiciones:</b> Se crean, eliminan o modifican los campos deseados y se guardan sus atributos.	
<b>Prototipo</b>	

**Pantalla 6**



**Descripción.**

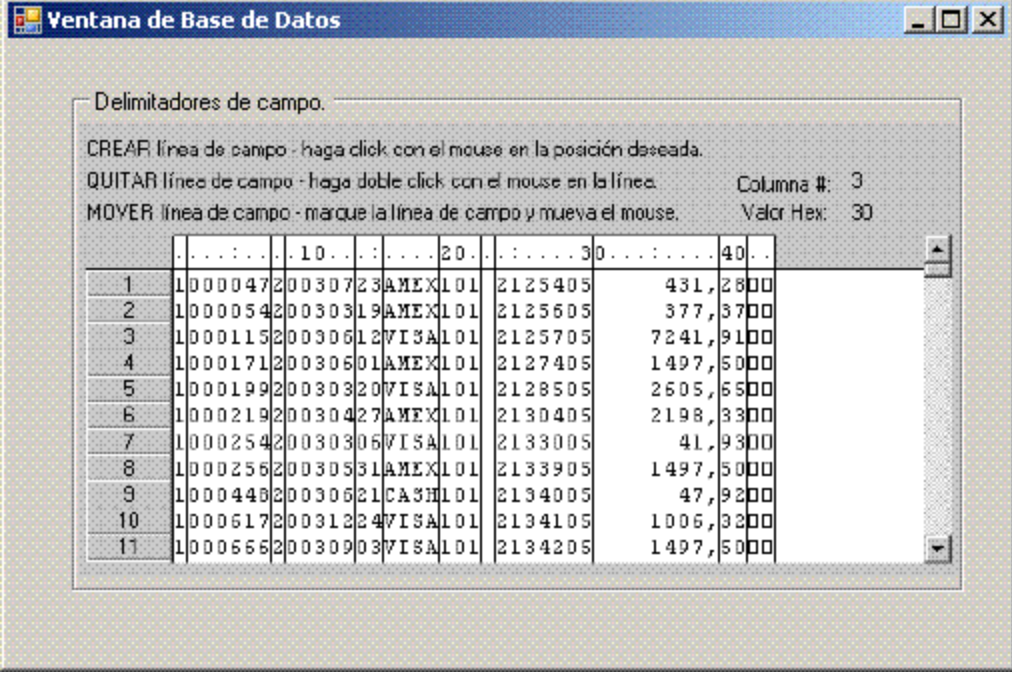
**Flujo Normal de Eventos.**

Acción del Actor.	Respuesta del Sistema.
1.El Auditor selecciona la opción Manejo de campos que incluye: <ul style="list-style-type: none"> <li>• Crear Campos</li> <li>• Modificar Campos</li> <li>• Eliminar Campos</li> </ul>	2.El sistema realiza la operación seleccionada <ul style="list-style-type: none"> <li>• Si seleccionó Crear Campos, ver sección "Crear Campos".</li> <li>• Si seleccionó Modificar Campos, ver sección "Modificar Campos".</li> <li>• Si seleccionó Eliminar Campos, ver sección "Eliminar Campos".</li> </ul>

**Sección Crear campos.**

Acción del actor.	Respuesta del sistema.
-------------------	------------------------

3. El Auditor solicita la opción Crear campos.	4. El sistema muestra una ventana solicitando se especifiquen los valores de los atributos: nombre del campo, tipo y longitud.
5. El Auditor especifica los valores para cada atributo.	6. El sistema guarda los valores introducidos por el Auditor.
	7. El sistema muestra en la Ventana de Base de Datos los campos creados.
<b>Sección Modificar campos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
8. El Auditor solicita la opción Modificar campos.	9. El sistema solicita se especifique los campos a modificar.
10. El Auditor especifica el(los) campos a modificar.	11. El sistema muestra el valor anterior de los atributos nombre del campo, tipo y longitud y solicita ingresar los nuevos valores.
12. El Auditor especifica los nuevos valores de los atributos nombre del campo, tipo y longitud.	13. El sistema guarda los nuevos valores de los atributos especificados por el Auditor.
	14. El sistema muestra los nuevos atributos del (los) campos modificados en la Ventana de Base de Datos
<b>Sección Eliminar campos.</b>	
<b>Acción del actor.</b>	<b>Respuesta del sistema.</b>
15. El Auditor solicita la opción Eliminar campos.	16. El sistema solicita se especifique el (los) campos a eliminar.
17. El Auditor especifica el(los) campos a eliminar.	18. El sistema elimina los campos seleccionados.
	19. El sistema muestra la Ventana de Base de Datos sin los campos eliminados.
<b>Flujo alternativo.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Requerimientos no Funcionales</b>	
<b>Prioridad</b>	

<b>CU-7</b>																																																													
<b>Gestionar delimitadores de campos. Extendido del caso de uso Importar ficheros de datos no formateados.</b>																																																													
<b>Actores</b>																																																													
<b>Propósito</b>	Permitirá crear, mover y eliminar delimitadores de campos.																																																												
<b>Resumen :</b>																																																													
Este caso de uso se inicia cuando es invocado por el caso de uso Importar ficheros de datos no formateados, donde el Auditor del proceso de importación, puede: crear, mover y eliminar los delimitadores de los campos. Este caso de uso termina cuando los campos han sido gestionados, así como los delimitadores demandados por el usuario.																																																													
<b>Referencias:</b> RF7, RF3																																																													
<b>Precondiciones:</b>																																																													
<b>Poscondiciones:</b> Se crean, eliminan o modifican los delimitadores de campo deseados y se guardan sus atributos.																																																													
<b>Prototipo</b>																																																													
<b>Pantalla 7</b>																																																													
 <p>Delimitadores de campo.</p> <p>CREAR línea de campo - haga click con el mouse en la posición deseada.</p> <p>QUITAR línea de campo - haga doble click con el mouse en la línea.      Columna #: 3</p> <p>MOVER línea de campo - marque la línea de campo y mueva el mouse.      Valor Hex: 30</p> <table border="1"> <thead> <tr> <th></th> <th>.....10.....</th> <th>.....20.....</th> <th>.....30.....</th> <th>.....40.....</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>100004720030723</td> <td>AMEX101</td> <td>2125405</td> <td>431,2800</td> </tr> <tr> <td>2</td> <td>100005420030319</td> <td>AMEX101</td> <td>2125605</td> <td>377,3700</td> </tr> <tr> <td>3</td> <td>100011520030612</td> <td>VISA101</td> <td>2125705</td> <td>7241,9100</td> </tr> <tr> <td>4</td> <td>100017120030601</td> <td>AMEX101</td> <td>2127405</td> <td>1497,5000</td> </tr> <tr> <td>5</td> <td>100019920030320</td> <td>VISA101</td> <td>2128505</td> <td>2605,6500</td> </tr> <tr> <td>6</td> <td>100021920030427</td> <td>AMEX101</td> <td>2130405</td> <td>2198,3300</td> </tr> <tr> <td>7</td> <td>100025420030306</td> <td>VISA101</td> <td>2133005</td> <td>41,9300</td> </tr> <tr> <td>8</td> <td>100025620030531</td> <td>AMEX101</td> <td>2133905</td> <td>1497,5000</td> </tr> <tr> <td>9</td> <td>100044020030621</td> <td>CASH101</td> <td>2134005</td> <td>47,9200</td> </tr> <tr> <td>10</td> <td>100061720031224</td> <td>VISA101</td> <td>2134105</td> <td>1006,3200</td> </tr> <tr> <td>11</td> <td>100066620030903</td> <td>VISA101</td> <td>2134205</td> <td>1497,5000</td> </tr> </tbody> </table>			.....10.....	.....20.....	.....30.....	.....40.....	1	100004720030723	AMEX101	2125405	431,2800	2	100005420030319	AMEX101	2125605	377,3700	3	100011520030612	VISA101	2125705	7241,9100	4	100017120030601	AMEX101	2127405	1497,5000	5	100019920030320	VISA101	2128505	2605,6500	6	100021920030427	AMEX101	2130405	2198,3300	7	100025420030306	VISA101	2133005	41,9300	8	100025620030531	AMEX101	2133905	1497,5000	9	100044020030621	CASH101	2134005	47,9200	10	100061720031224	VISA101	2134105	1006,3200	11	100066620030903	VISA101	2134205	1497,5000
	.....10.....	.....20.....	.....30.....	.....40.....																																																									
1	100004720030723	AMEX101	2125405	431,2800																																																									
2	100005420030319	AMEX101	2125605	377,3700																																																									
3	100011520030612	VISA101	2125705	7241,9100																																																									
4	100017120030601	AMEX101	2127405	1497,5000																																																									
5	100019920030320	VISA101	2128505	2605,6500																																																									
6	100021920030427	AMEX101	2130405	2198,3300																																																									
7	100025420030306	VISA101	2133005	41,9300																																																									
8	100025620030531	AMEX101	2133905	1497,5000																																																									
9	100044020030621	CASH101	2134005	47,9200																																																									
10	100061720031224	VISA101	2134105	1006,3200																																																									
11	100066620030903	VISA101	2134205	1497,5000																																																									
<b>Descripción.</b>																																																													

**Flujo Normal de Eventos.**

<b>Acción del Actor.</b>	<b>Respuesta del Sistema.</b>
	1. El sistema muestra en la Ventana de Base de Datos el fichero de origen.
2. El Auditor solicita: <ul style="list-style-type: none"> <li>• Crear línea de campo.</li> <li>• Mover línea de campo</li> <li>• Eliminar línea de campo.</li> </ul>	3. El sistema realiza la operación seleccionada: <ul style="list-style-type: none"> <li>• Si seleccionó Crear línea de campo, ver sección "Crear línea de campo".</li> <li>• Si seleccionó Mover línea de campo, ver sección "Mover línea de campo".</li> <li>• Si seleccionó Eliminar línea de campo, ver sección "Eliminar línea de campo".</li> </ul>
<b>Sección Crear línea de campo.</b>	
<b>Acción del actor.</b>	<b>Respuesta del sistema.</b>
4. El Auditor solicita crear una línea campo.	5. El sistema permite que el usuario marque con el Mouse la ubicación deseada para la línea de campo.
	6. El sistema guarda la posición de la ubicación del bit donde se definió la línea de campo.
	7. El sistema muestra la línea de campo creada.
<b>Sección Mover línea de campo.</b>	
<b>Acción del actor.</b>	<b>Respuesta del sistema.</b>
8. El Auditor solicita mover una línea campo.	9. El sistema permite que el usuario arrastre con el Mouse la línea de campo seleccionada para mover, y le de una nueva ubicación.
	10. El sistema guarda la nueva posición de la ubicación del bit a donde se movió la línea de campo.

	11. El sistema muestra la nueva posición de la línea de campo movida.
<b>Sección Eliminar línea de campo.</b>	
<b>Acción del actor.</b>	<b>Respuesta del sistema.</b>
12. El Auditor especifica con un doble clic la línea de campo que desea eliminar.	13. El sistema elimina la línea de campo seleccionada.
	14. El sistema muestra en la Ventana de Base de Datos los datos sin el delimitador eliminado.
<b>Flujo alternativo.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Requerimientos no Funcionales</b>	
<b>Prioridad</b>	

<b>CU-8</b>	
<b>Mostrar estadísticas de campos.</b>	
<b>Actores</b>	Auditor.
<b>Propósito</b>	Mostrar estadísticas de campo para los campos numéricos seleccionados (número de registros, número de elementos cero, cantidad de registros positivos, cantidad de registros negativos, cantidad de errores de datos, número de registros mínimos, número de registros máximos, desviación estándar poblacional, curtosis poblacional, valor neto, valor absoluto, valor positivo, valor negativo, valor medio, valor mínimo, valor máximo, desviación estándar muestral, varianza muestral, asimetría poblacional.).
<b>Resumen :</b> Este caso de uso se inicia cuando se termina de importar la Base de Datos. Este caso de uso termina cuando se muestra al usuario las estadísticas de campo predefinidas (número de registros, número de elementos cero, cantidad de registros positivos, cantidad de registros negativos, cantidad de errores de datos, número de registros mínimos, número de registros máximos, desviación estándar poblacional, curtosis poblacional, valor neto, valor absoluto, valor positivo, valor negativo, valor medio, valor mínimo, valor máximo, desviación estándar muestral, varianza muestral, asimetría poblacional.)	
<b>Referencias:</b> RF8	

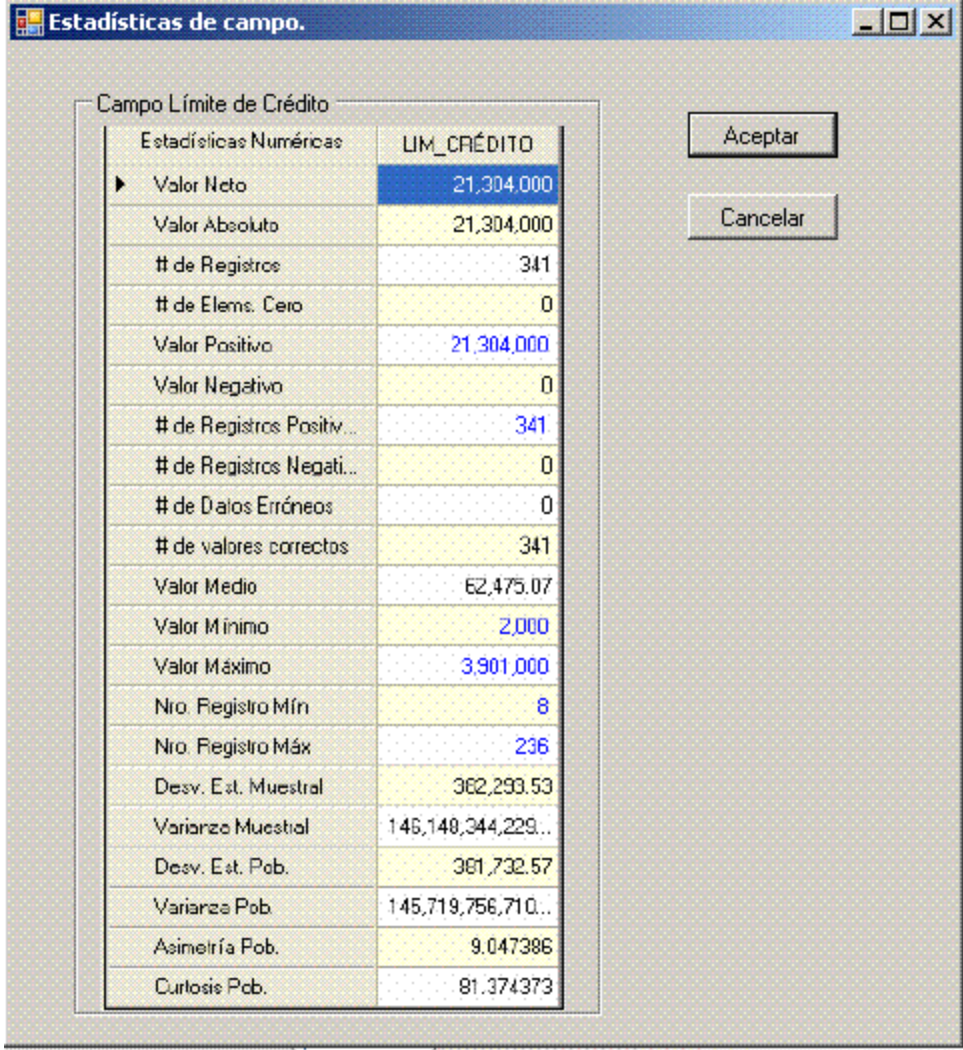


**Precondiciones:** Debe ser un fichero que contenga campos numéricos.

**Poscondiciones:** Se crea un fichero dentro de la aplicación con la base de datos importada.

**Prototipo**

**Pantalla 8**

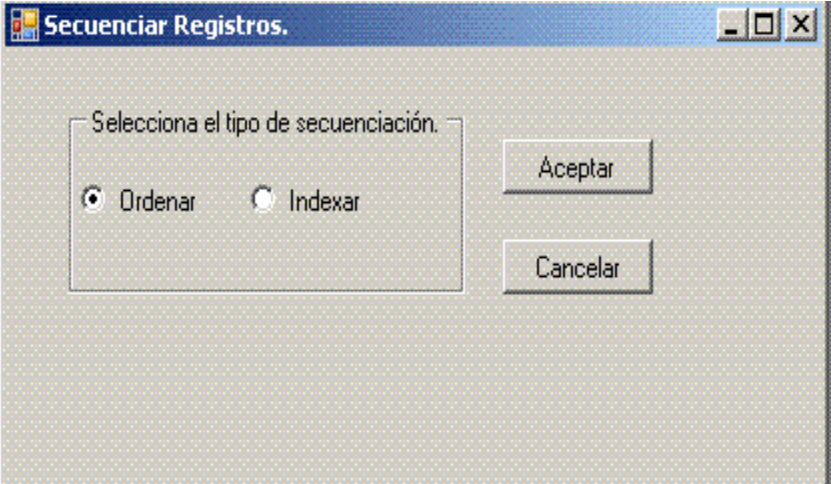


**Descripción.**

**Flujo Normal de Eventos.**

Acción del Actor	Respuesta del Sistema
1. El Auditor selecciona la opción Mostrar estadísticas de campo.	2. El sistema solicita seleccionar el campo deseado para mostrar las estadísticas.

<p>3. El Auditor selecciona el campo deseado para ver sus estadísticas.</p>	<p>4. El sistema cuenta los registros de la Base de Datos Activa.</p> <p>5. El sistema cuenta número de elementos cero.</p> <p>6. El sistema calcula el valor máximo.</p> <p>7. El sistema calcula el valor medio.</p> <p>8. El sistema calcula el valor mínimo.</p> <p>9. El sistema calcula el valor absoluto.</p> <p>10. El sistema calcula el valor positivo.</p> <p>11. El sistema calcula el valor negativo.</p> <p>12. El sistema calcula el número de registros positivos.</p> <p>13. El sistema calcula el número de registros negativos.</p> <p>14. El sistema calcula el número de datos erróneos.</p> <p>15. El sistema calcula el número de valores correctos.</p> <p>16. El sistema calcula el número de registros máximos.</p> <p>17. El sistema calcula el número de registros mínimos.</p> <p>18. El sistema calcula la desviación estándar muestral.</p> <p>19. El sistema calcula la varianza muestral.</p>
	<p>20. El sistema muestra todas las estadísticas de campo calculadas, en una ventana.</p>
<p><b>Flujo alternativo.</b></p>	
<p><b>Acción del actor.</b></p>	<p><b>Respuesta del sistema.</b></p>
<p><b>Acción 3 F1</b> Si se selecciona un campo no numérico.</p>	<p><b>3 F1.1</b> El sistema muestra el mensaje "Debe seleccionar un campo numérico".</p>
<p><b>Requerimientos no Funcionales</b></p>	
<p><b>Prioridad</b></p>	

<b>CU-9</b>	
<b>Secuenciar registros.</b>	
<b>Actores</b>	Auditor.
<b>Propósito</b>	Realizar la secuenciación.
<b>Resumen :</b> El caso de uso se inicia cuando el Auditor solicita secuenciar los registros de una columna de datos. La secuenciación puede realizarse de dos maneras: ordenando e indexando los registros. El caso de uso termina cuando los registros de la columna de datos seleccionada son ordenados o indexados según la demanda del usuario.	
<b>Referencias:</b> RF9	
<b>Precondiciones:</b> Debe existir al menos una columna de datos que secuenciar.	
<b>Poscondiciones:</b> Se debe secuenciar la columna de datos según la selección del usuario.	
<b>Prototipo</b>	
<b>Pantalla 9</b>	
 <p>The screenshot shows a Windows-style dialog box titled "Secuenciar Registros." with a standard title bar (minimize, maximize, close buttons). The main content area contains the text "Selecciona el tipo de secuenciación." followed by two radio button options: "Ordenar" (which is selected) and "Indexar". To the right of these options are two buttons: "Aceptar" and "Cancelar".</p>	
<b>Pantalla 10</b>	

Campo LIM CREDITO ordenado

CLIENTE	NOMBRE APELLIDOS	LIM CREDITO
1001	María Suárez	35.06
1002	Angel Díaz	31.25
1003	Carmen Cruz	30.98
1004	Apolinar Rodríguez	25.14
1005	Raúl Paz	22.65
1006	Ofelia Carmenate	11.84
1007	Enrique Pérez	10.47
1008	Omar Concepción	8.06

### Descripción

#### Flujo Normal de Eventos.

Acción del Actor	Respuesta del Sistema
1. El Auditor selecciona la columna opción Secuenciar Registros.	2. El sistema muestra una ventana con los dos tipos de secuenciación: Ordenar e Indexar.
3. El Auditor selecciona la opción deseada.	4. Si seleccionó la opción <ul style="list-style-type: none"> <li>• Ordenar: "Ver Sección Ordenar."</li> <li>• Indexar "Ver Sección Indexar "</li> </ul>
<b>Sección Ordenar.</b>	
Acción del actor	Respuesta del sistema
5. El Auditor solicita la opción de Ordenar Registros.	6. El sistema muestra una ventana con los tipos de ordenamientos: Mayor a menor y Menor a mayor.
7. El Auditor selecciona el tipo de ordenamiento deseado.	8. El sistema muestra una ventana con los campos por los que se puede realizar el ordenamiento.

9. El Auditor selecciona el campo por el que se va a realizar el ordenamiento.	10. El sistema muestra la Ventana de Base de Datos ya ordenada de los campos seleccionado.
<b>Sección Indexar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
11. El Auditor solicita la opción Indexar registros.	12. El sistema muestra una ventana con los tipos de indexación: Mayor a menor y Menor a mayor.
13. El Auditor selecciona el tipo de indexación deseado.	14. El sistema muestra la Ventana de Base de Datos ya indexada por el o los campos seleccionados
15. El Auditor selecciona el campo por el que se va a realizar la indexación.	16. El sistema muestra la columna de datos ya indexada en la Ventana de Base de Datos..
	17. El sistema crea un nuevo fichero dentro de la aplicación y guarda la Base de Datos el fichero de índices por el campo y criterio seleccionado por el usuario.
<b>Flujo alternativo.</b>	
<b>Acción del actor.</b>	<b>Respuesta del sistema.</b>
<b>Acción 10 y Acción 16 F1</b> Si se selecciona un campo no numérico.	<b>10 y 16 F1.1</b> El sistema muestra el mensaje “Debe seleccionar un campo numérico”.
<b>Requerimientos no Funcionales</b>	
<b>Prioridad</b>	

## **2.7 CONLCLUSIONES.**

En este capítulo se expuso la propuesta del sistema. Se trabajó en la fase de requerimientos, definiendo el modelo del dominio, así como los requerimientos funcionales y no funcionales que regirán el proceso de desarrollo del sistema. Se elaboró el diagrama de casos de uso del sistema y se describieron los mismos. Quedando listo el proceso para entrar en la fase de análisis y diseño, donde se refinarán los requisitos y se ejecutarán otras actividades propias de ese flujo de trabajo.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

### 3.1 INTRODUCCIÓN.

En este capítulo se analizan los requisitos descritos en la captura de los mismos, refinándolos y estructurándolos como parte de su análisis. El análisis entra en el proceso de desarrollo de software como etapa decisiva, y uno de sus propósitos primarios es definir las clases y demás artefactos que intervienen en esta fase del RUP. Las primeras pautas para la implementación del sistema se definen en esta fase, de esta forma se definen las clases del análisis.

El diseño es el centro de atención final de la fase de elaboración y es decisivo, en el comienzo de las iteraciones de la fase de construcción, contribuye a una arquitectura estable y sólida. En el presente capítulo se procede a representar la realización de los casos de uso, los diagramas de secuencia y de clases del diseño. Además se realizará el modelo de datos, como un paso importante en la definición de los flujos de información (datos) dentro del proyecto.

“Durante el análisis y el diseño, transformamos el modelo de casos de uso mediante un modelo de análisis en un modelo de diseño, es decir, en una estructura de clasificadores y realizaciones de casos de uso.”[13]

### 3.2 ANÁLISIS.

El modelo de análisis permite obtener especificaciones más precisas de los requisitos que tenemos como resultado de su captura, además al trabajar en este flujo, se usa un lenguaje mucho más ligado a los desarrolladores, lo que permite adentrarse más en el funcionamiento interno del sistema.

Dicho modelo facilita el entendimiento, preparación y modificación de los requisitos, y es a su vez una primera aproximación a lo que será el modelo de diseño.

“Durante el análisis, analizamos los requisitos que se describen en la captura de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de

los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero, incluyendo su arquitectura”. [14]

### 3.2.1 Diagrama de clases del análisis.

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema.

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP clasifica las clases en:

- Entidad: Modelan información que posee larga vida y que es a menudo persistente.
- Interfaz: Modelan la interacción entre el sistema y sus actores.
- Control: Coordinan la realización de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

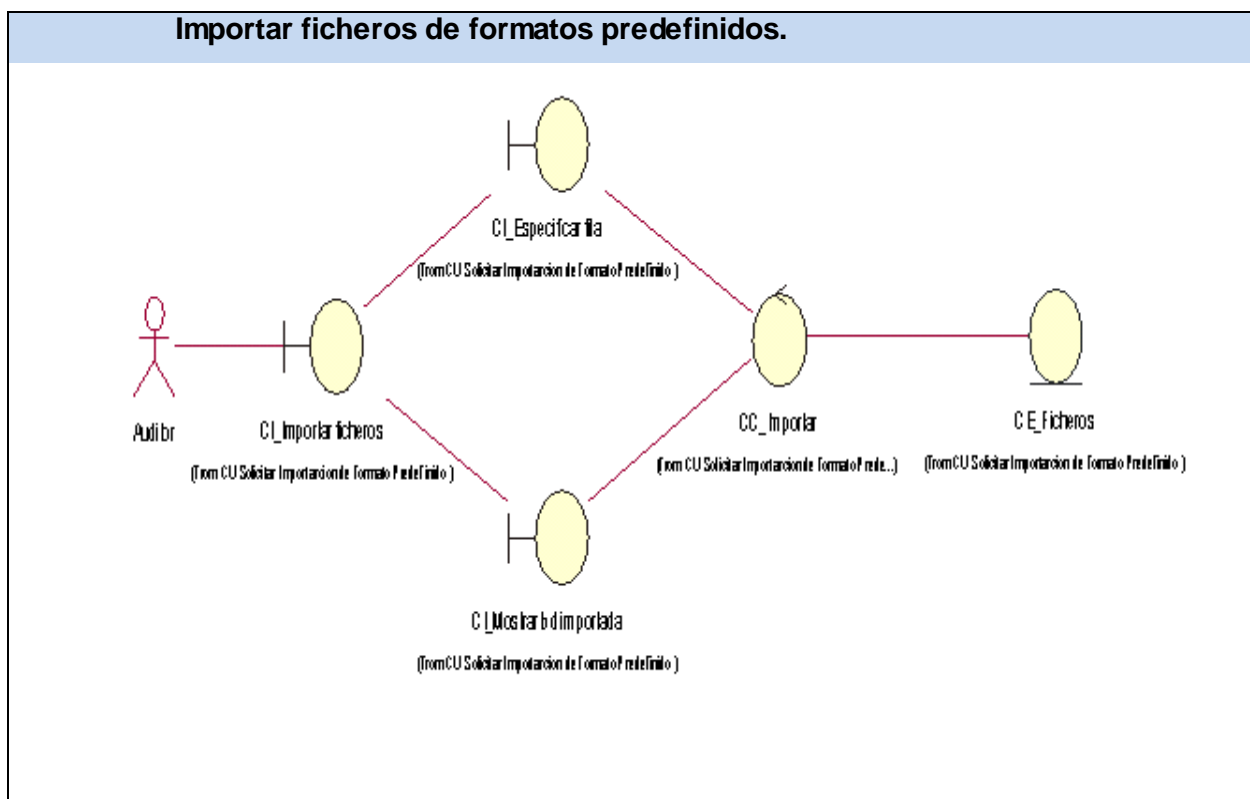
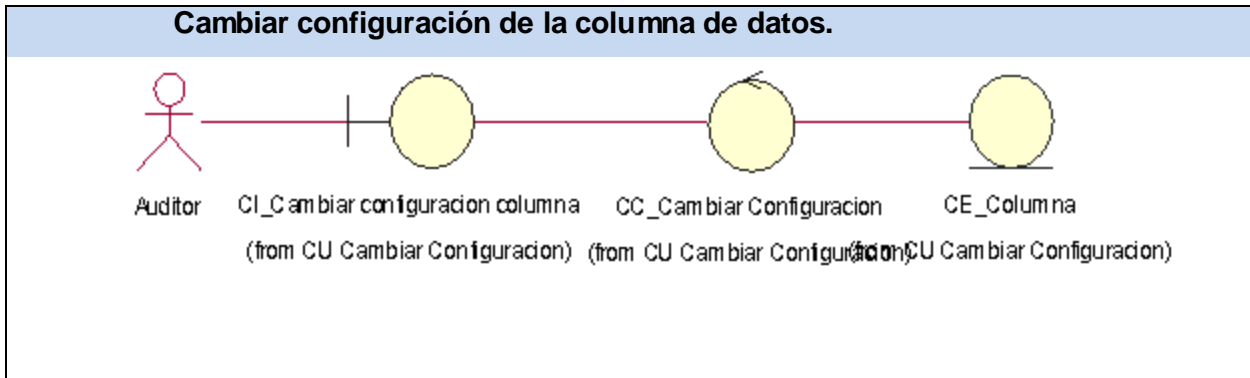
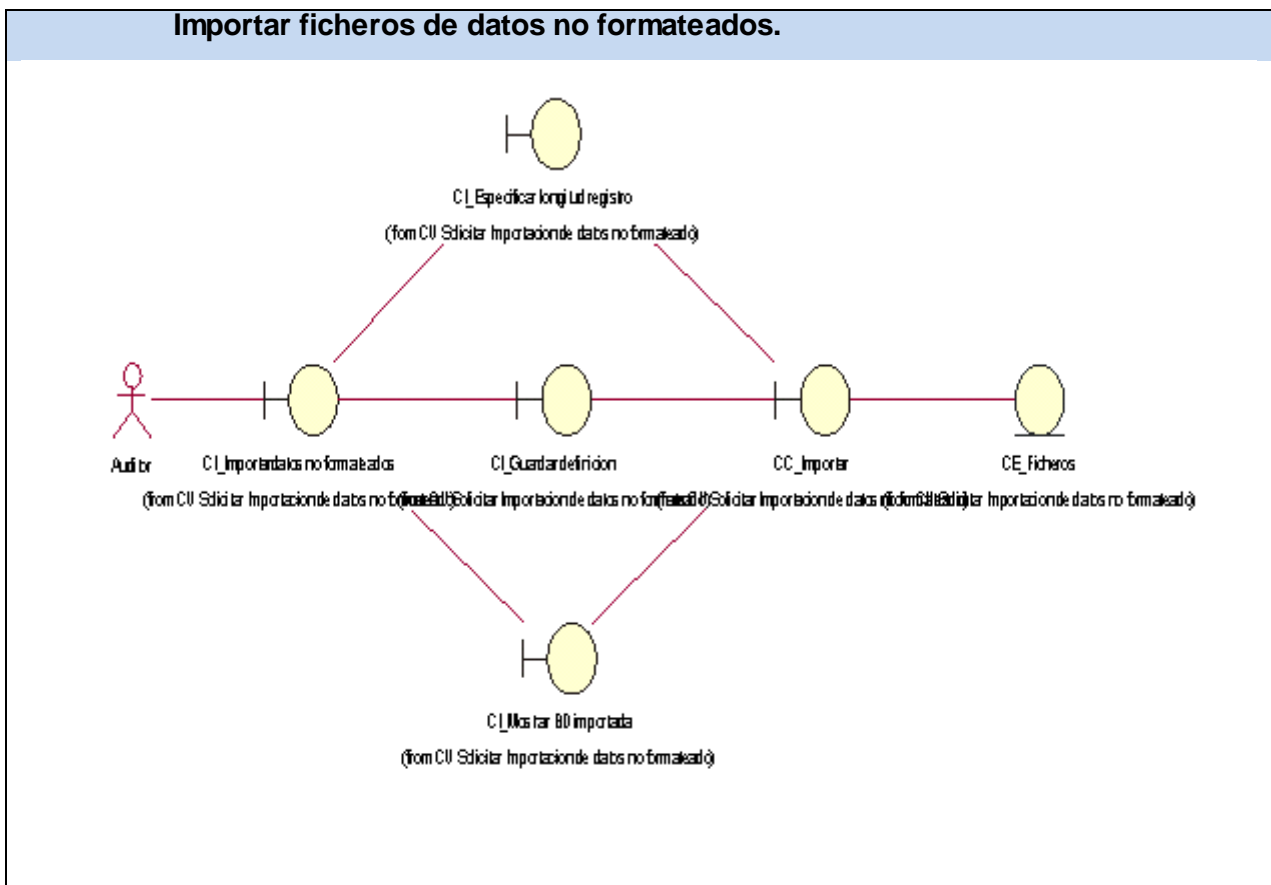


Fig.3.1 Caso de Uso Importar ficheros de formatos predefinidos.

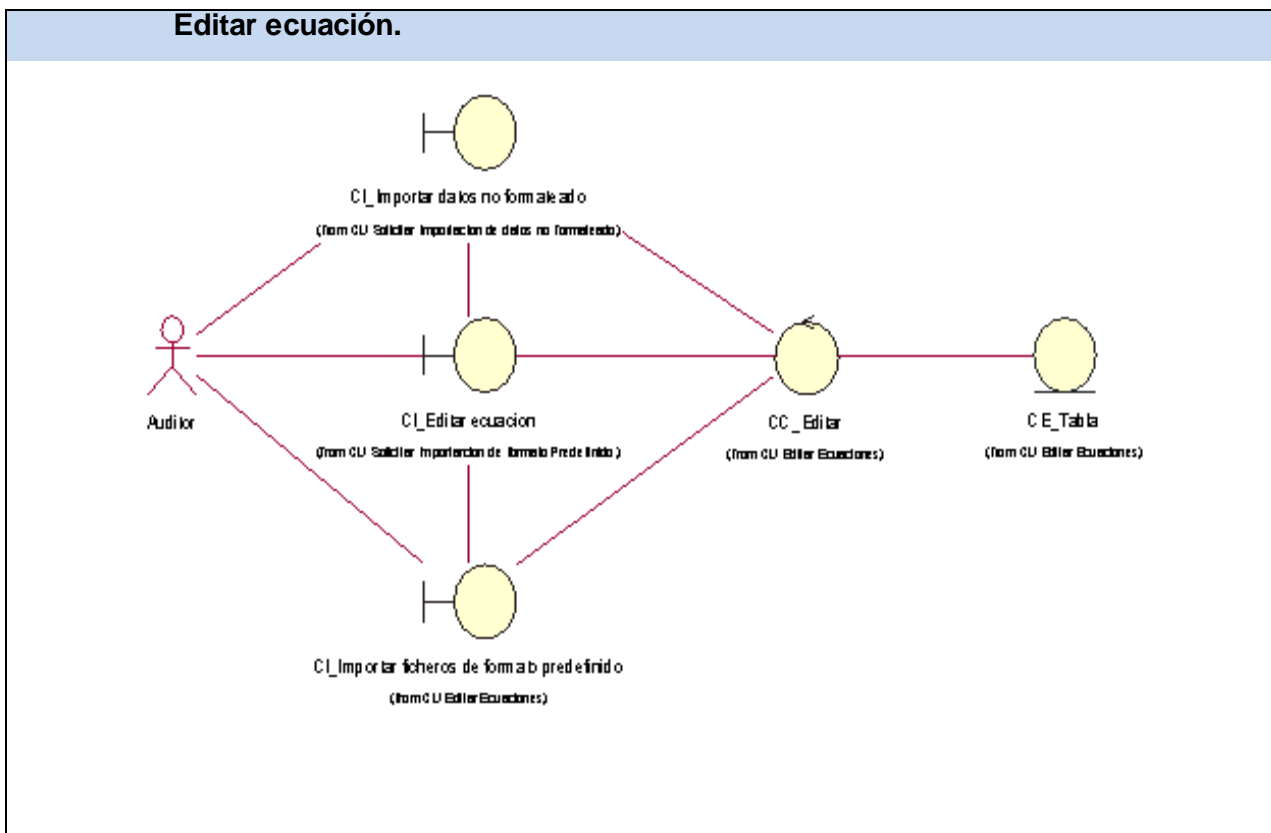




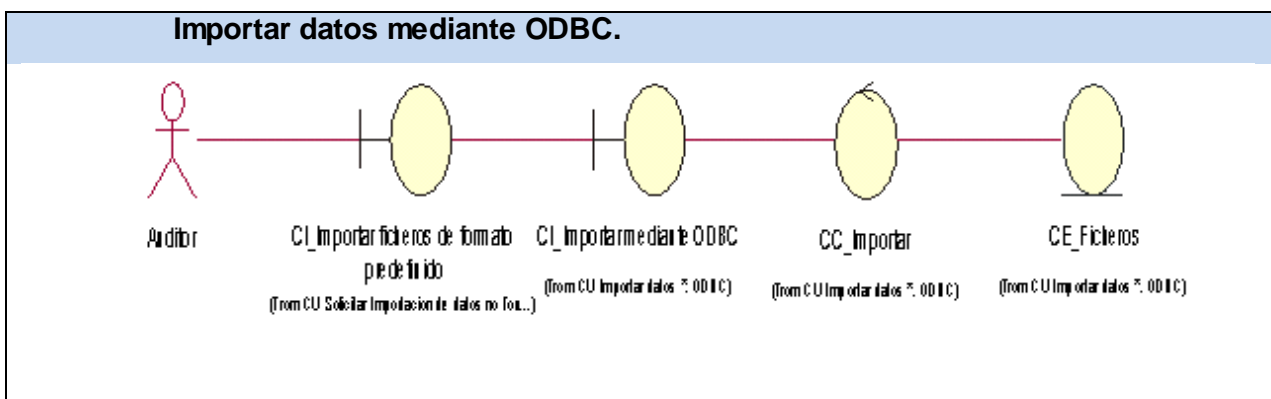
**Fig.3.2** Caso de Uso Cambiar configuración de la columna de datos.



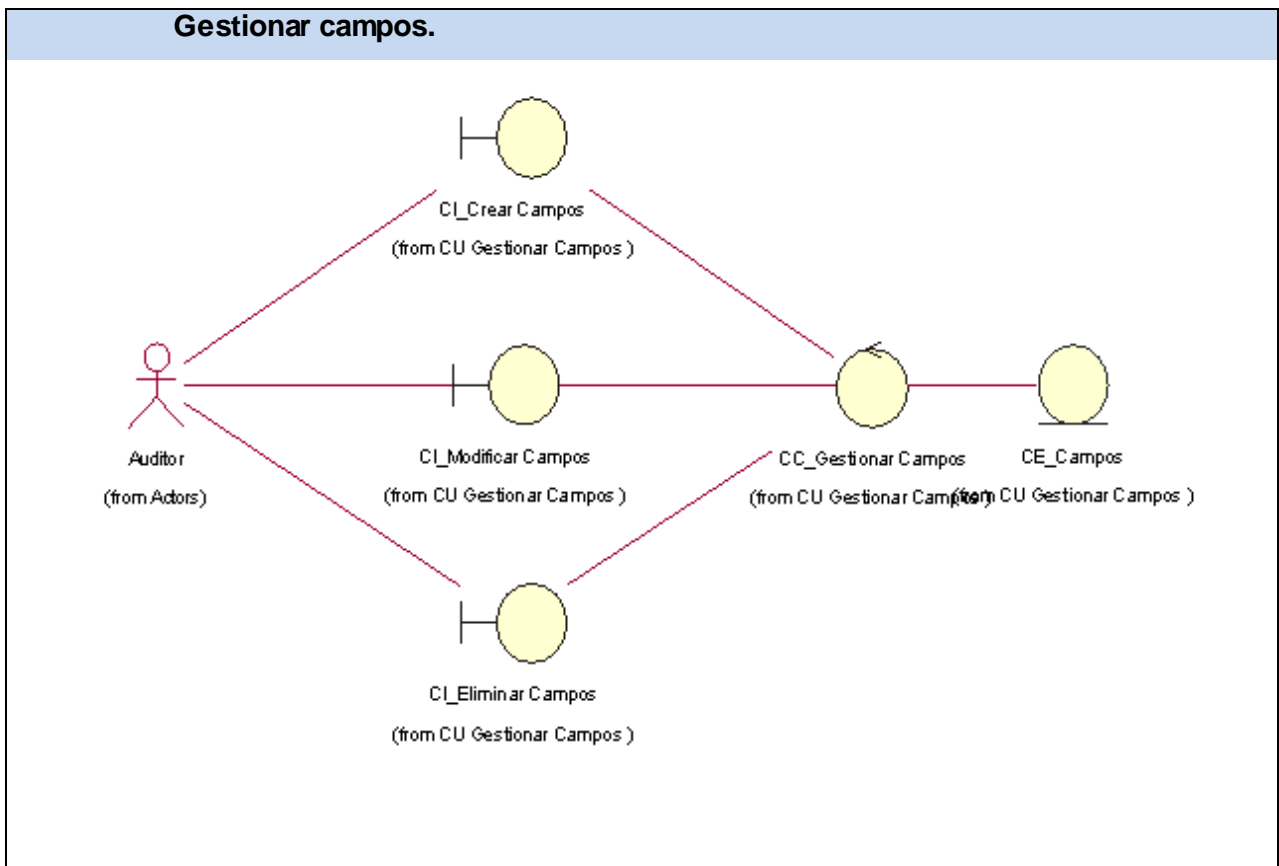
**Fig.3.3** Caso de Uso Importar ficheros de datos no formateados



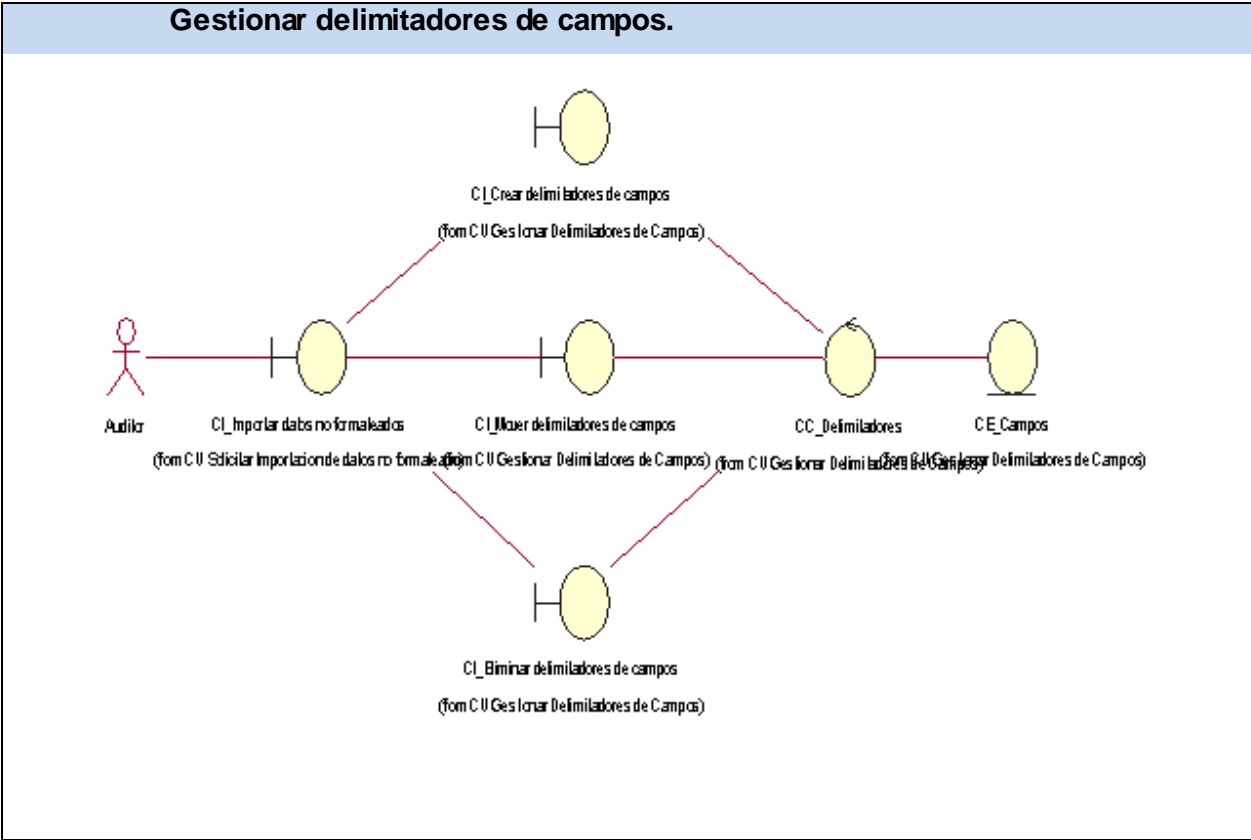
*Fig.3.4 Caso de Uso Editar ecuación.*



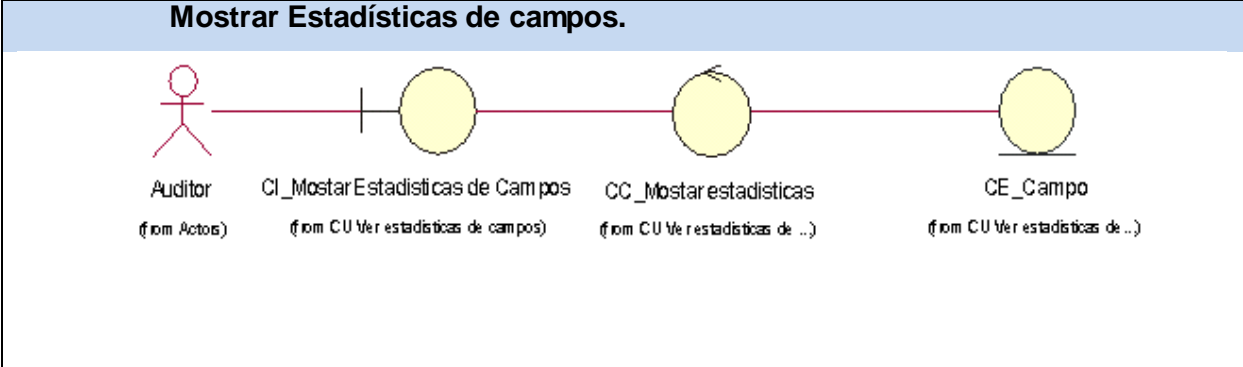
*Fig.3.5 Caso de Uso Importar ficheros mediante ODBC.*



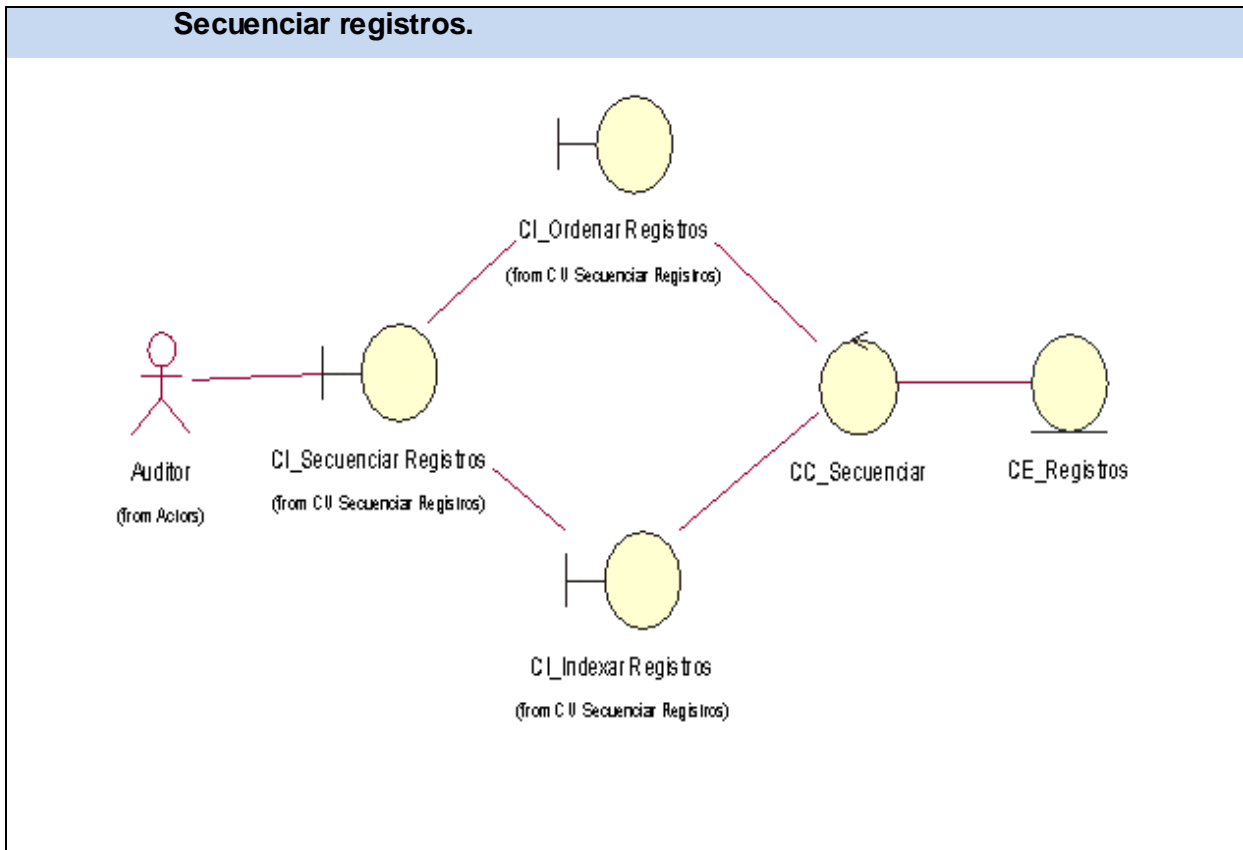
**Fig.3.6** Caso de Uso Gestionar campos.



**Fig.3.7** Caso de Uso Gestionar delimitadores de campos.



**Fig.3.8** Caso de Uso Mostrar estadísticas de campos.



*Fig.3.9 Caso de Uso Secuenciar registros.*

### 3.3 DIAGRAMAS DE COLABORACIÓN DEL ANÁLISIS.

Un Diagrama de Secuencias contribuye a la descripción de la dinámica del sistema en términos de la interacción entre sus objetos. Los diagramas de interacción son usados para identificar las clases y los métodos que ellos entregan. En el caso del análisis este diagrama permite apreciar el intercambio de información entre las diferentes clases del análisis. **(VER ANEXO 1).**

### 3.4 DISEÑO.

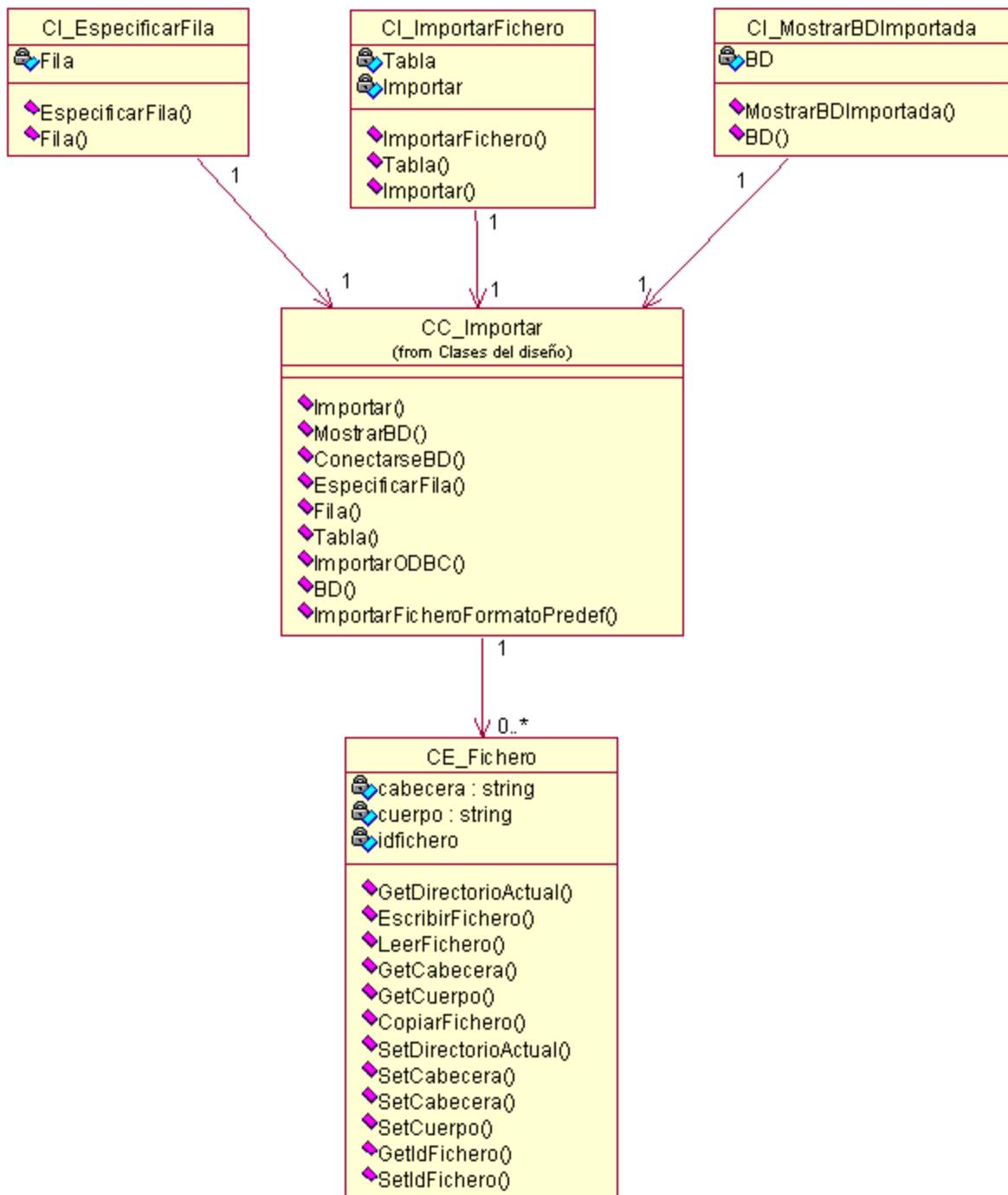
En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales. La esencia de esta fase es la elaboración de diagramas de

interacción, que muestran gráficamente cómo los objetos se comunican entre ellos a fin de cumplir con los requerimientos. Estos diagramas permiten la realización de los diagramas de clases del diseño, los cuales resumen la definición de las clases que se pueden implementar en el software.

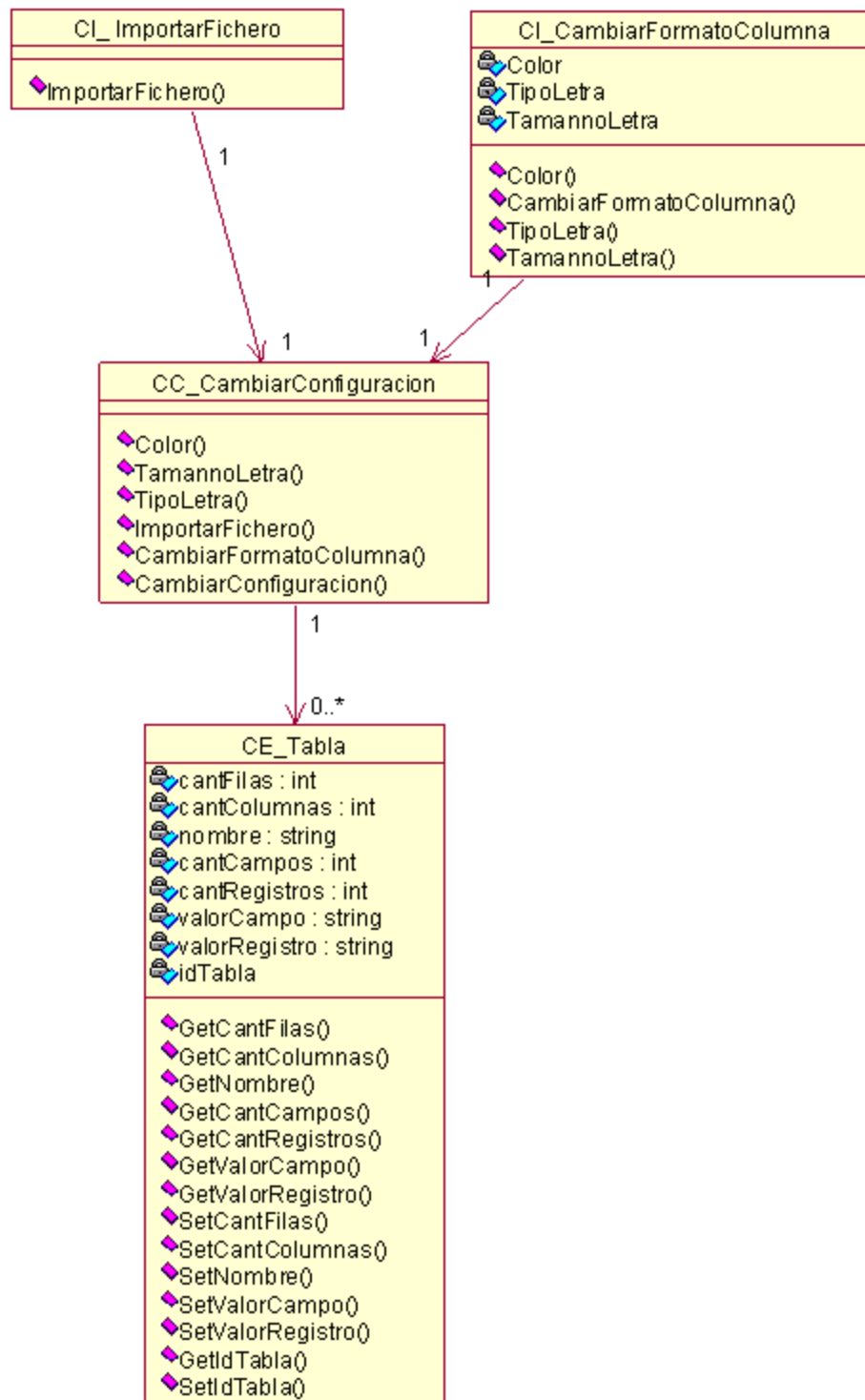
“El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación” [15].

### 3.4.1 Diagramas de clases del diseño.

Un Diagrama de Secuencias contribuye a la descripción de la dinámica del sistema en términos de la interacción entre sus objetos. Los diagramas de interacción son usados para identificar las clases y los métodos que ellos entregan.

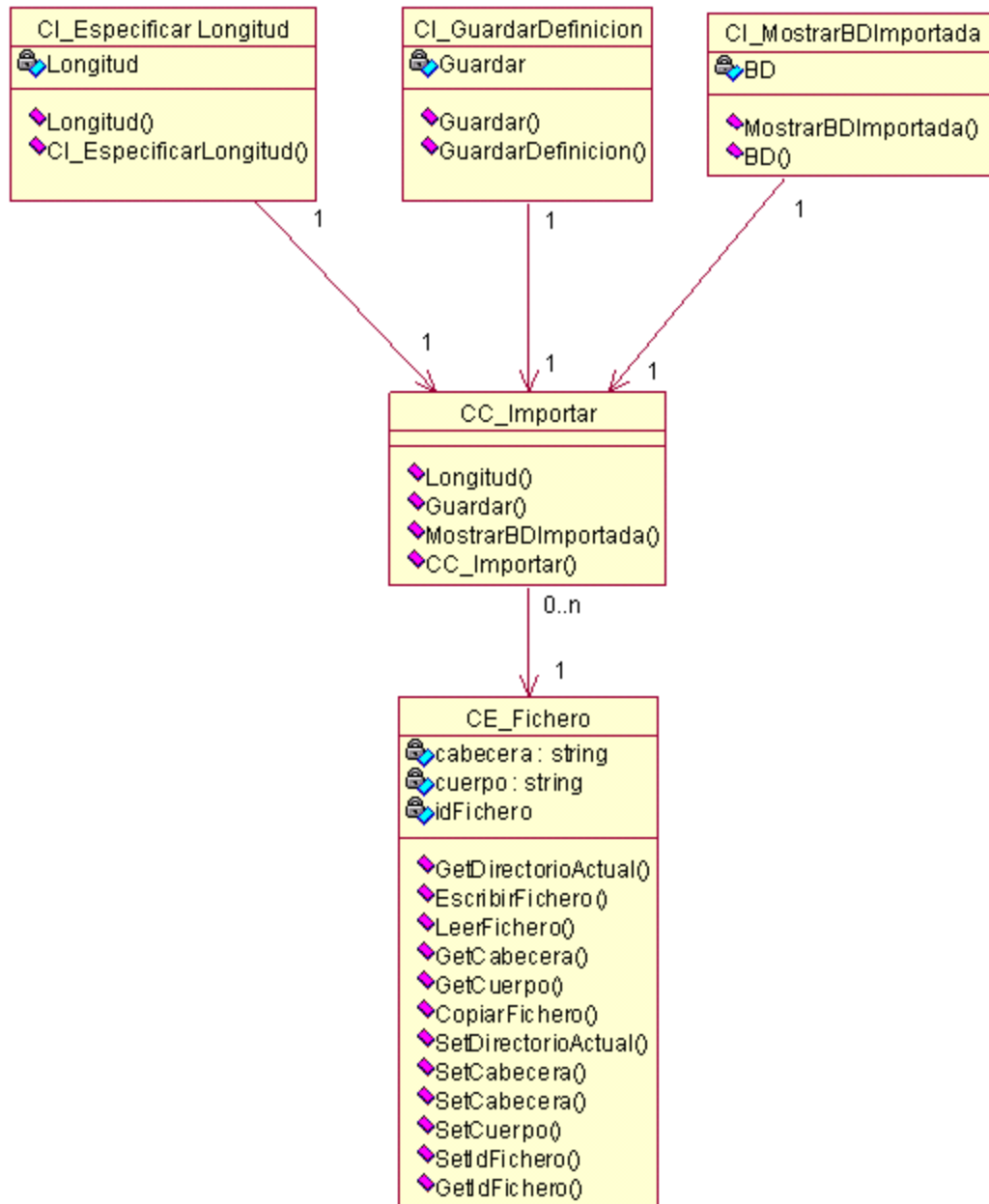


**Fig. 3.10** Diagrama de Clases de Diseño: Importar ficheros de formato predefinido.

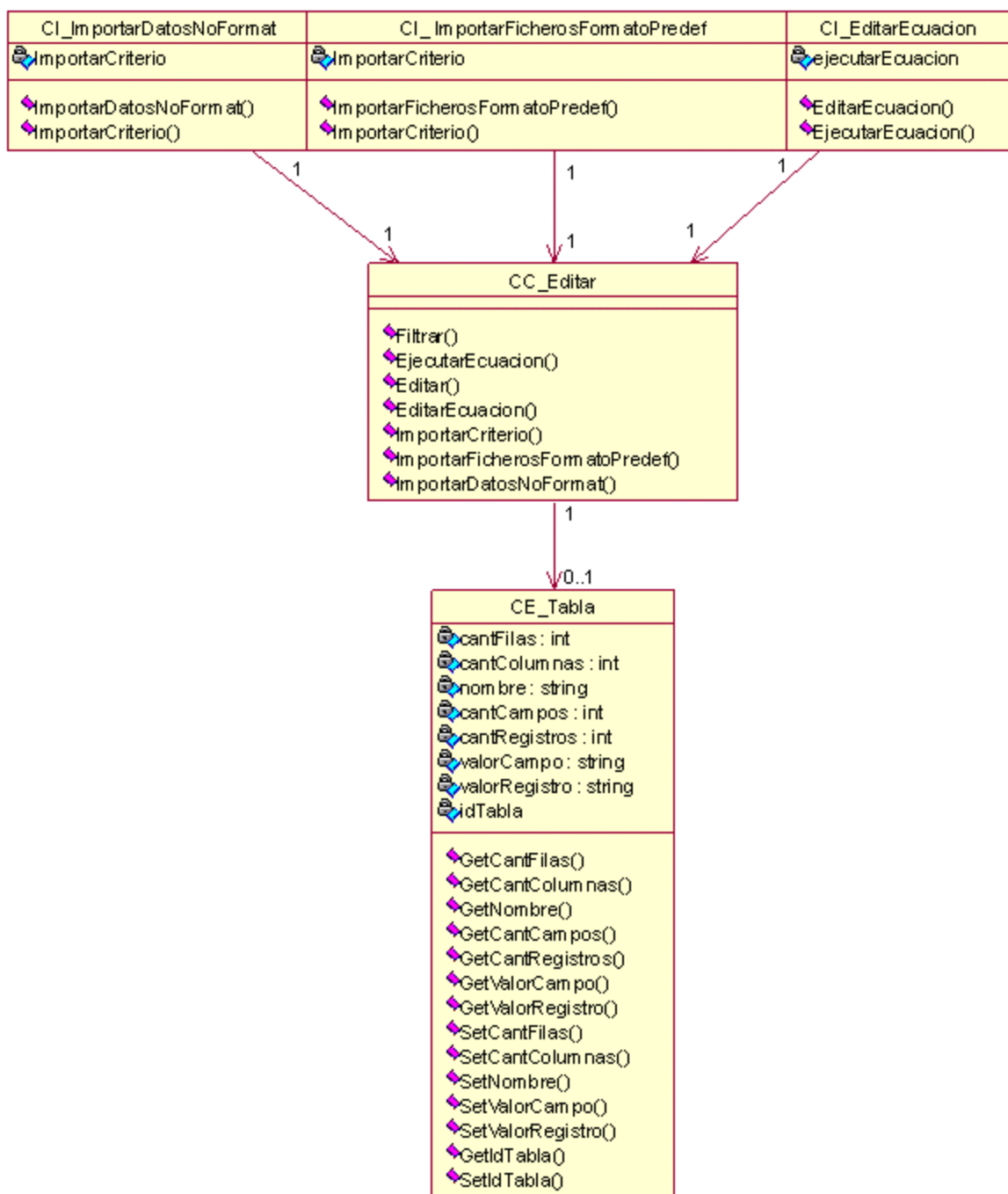


**Fig. 3.11** Diagrama de Clases del Diseño: Cambiar configuración de la columna de datos.

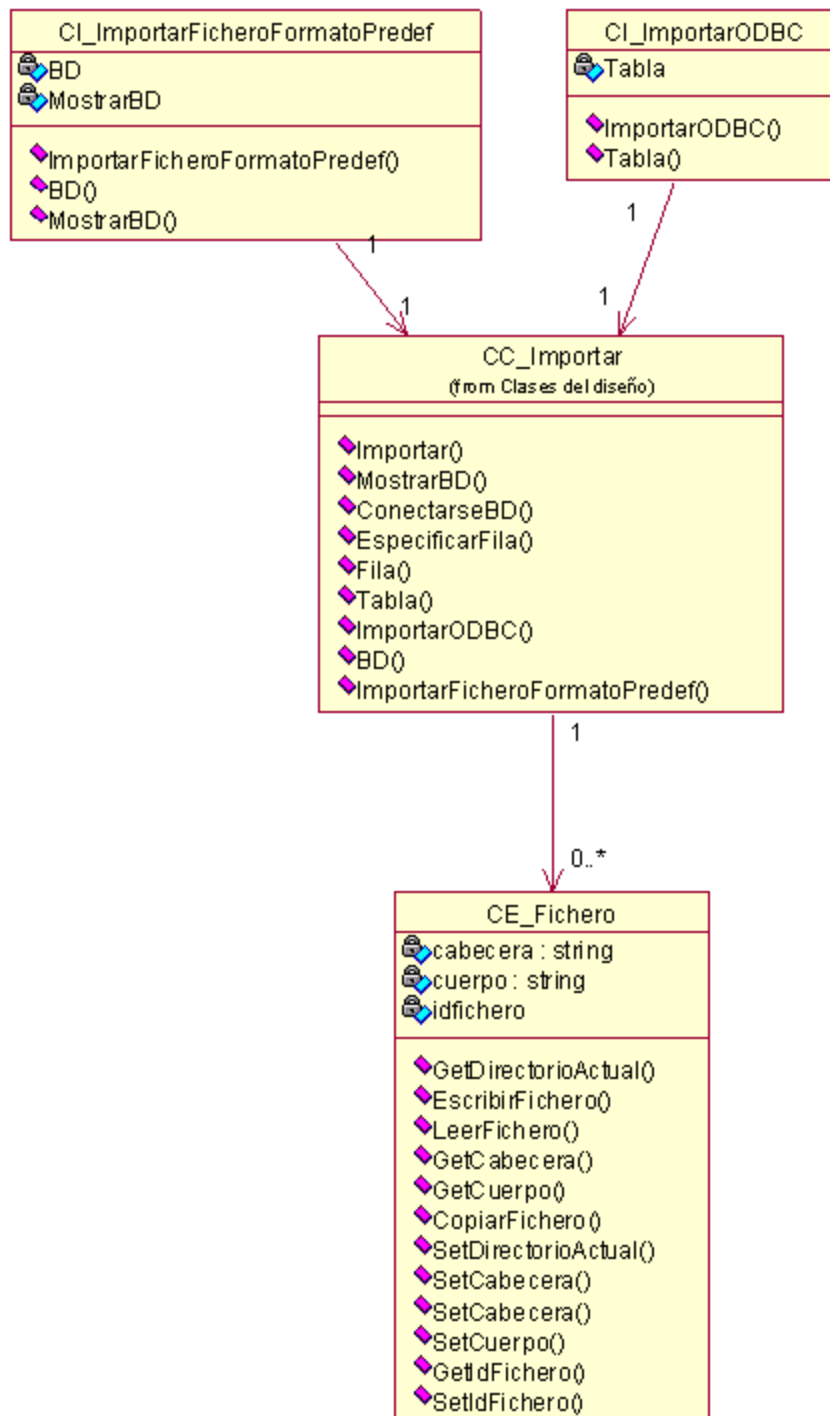




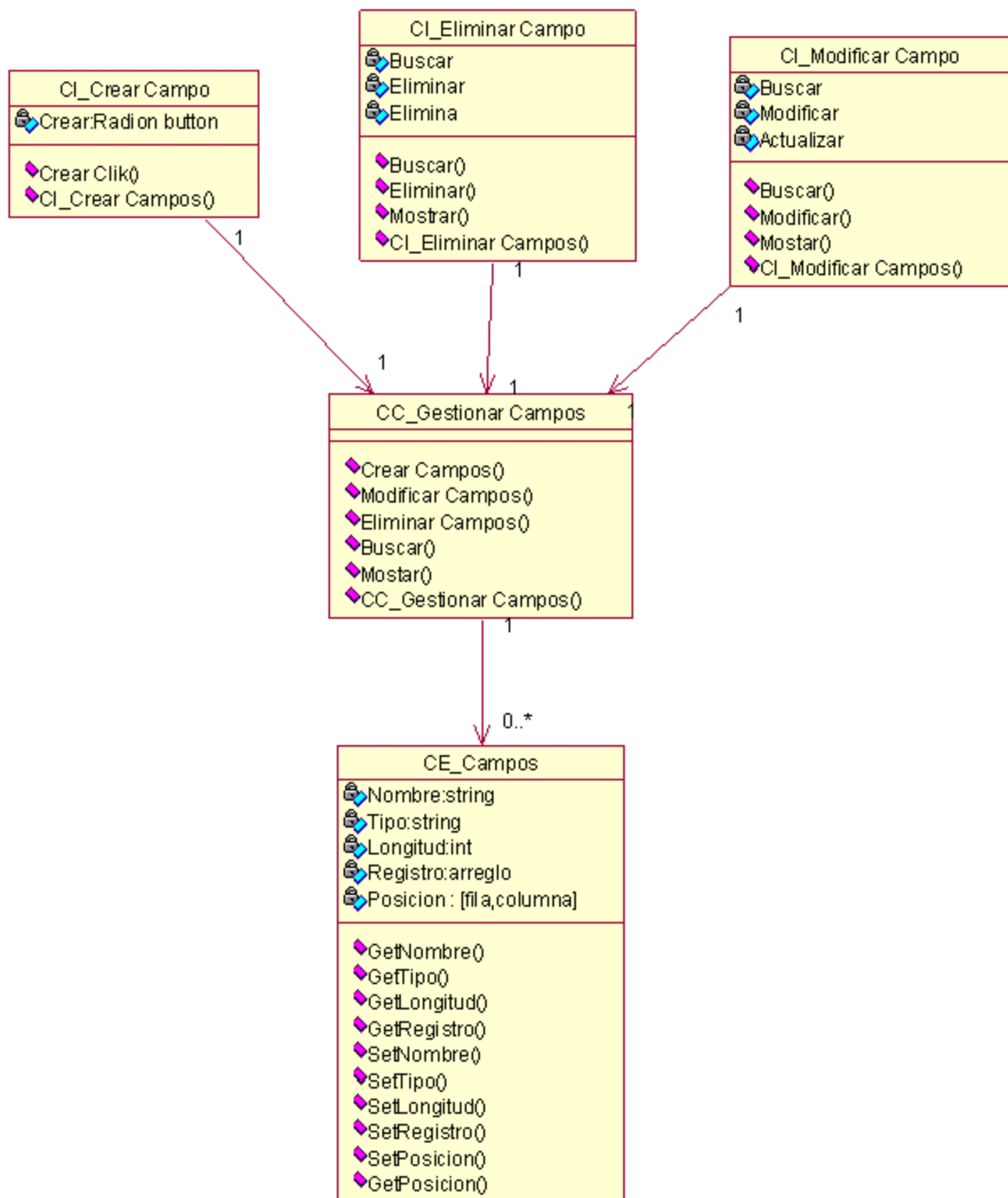
**Fig. 3.12** Diagrama de Clases del Diseño: Importar ficheros de datos no formateados.



**Fig. 3.13** Diagrama de Clases del Diseño: Editar ecuación.



**Fig. 3.14** Diagrama de Clases de Diseño: Importar ficheros mediante ODBC.



**Fig. 3.15** Diagrama de Clases de Diseño: Gestionar campos.

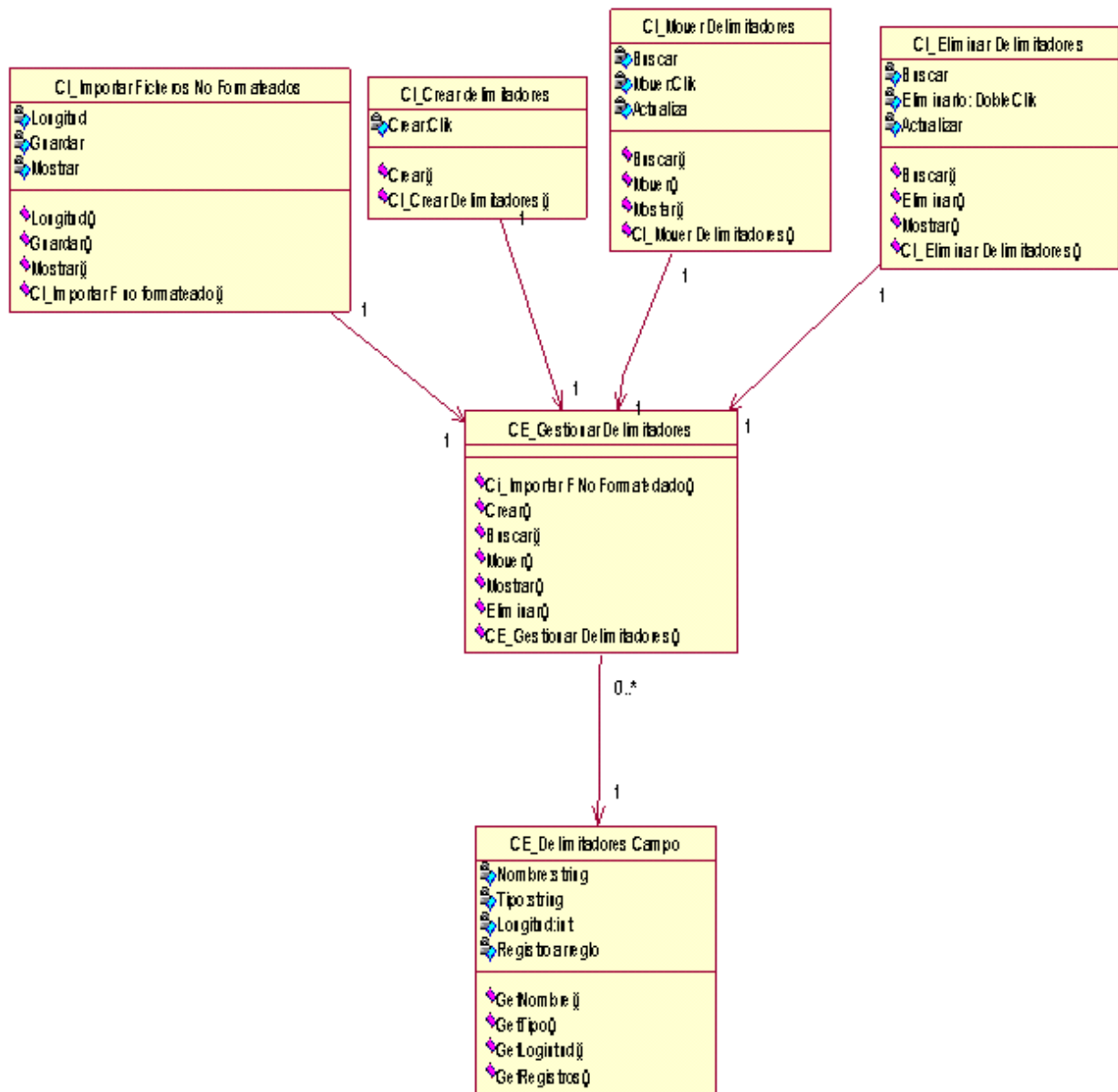
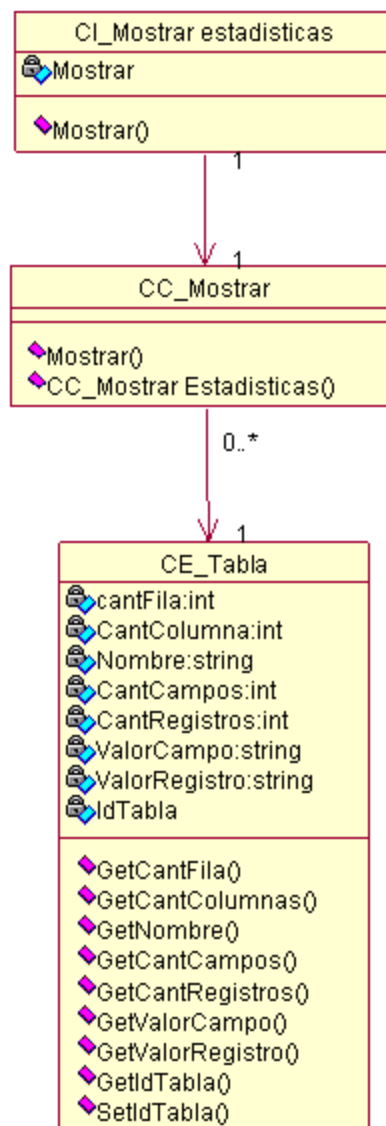
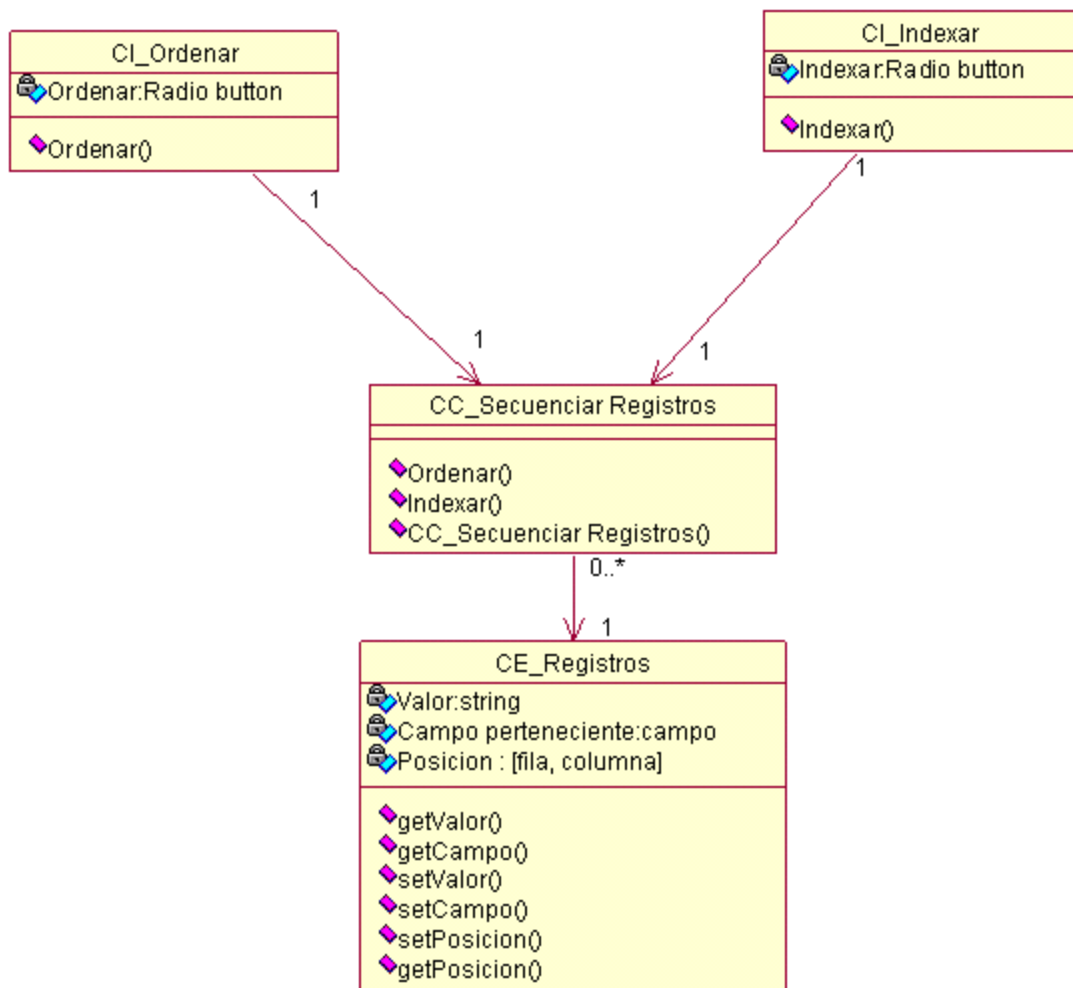


Fig. 3.16 Diagrama de Clases de Diseño: Gestionar delimitadores de campos.



**Fig. 3.17** Diagrama de Clases de Diseño: Mostrar estadísticas de campos.



*Fig. 3.18 Diagrama de Clases de Diseño: Secuenciar registros.*

### 3.5 DIAGRAMAS DE SECUENCIA DEL DISEÑO.

Como ya explicamos, un Diagrama de Secuencias contribuye a la descripción de la dinámica del sistema en términos de la interacción entre sus objetos. Los diagramas de interacción son usados para identificar las clases y los métodos que ellos entregan. En el caso del diseño, los diagramas de colaboración permiten la realización de los diagramas de clases del diseño, los cuales resumen la definición de las clases que se pueden implementar en el software (**VER ANEXO 2**).

### 3.6 DIAGRAMAS DE CLASES PERSISTENTES.

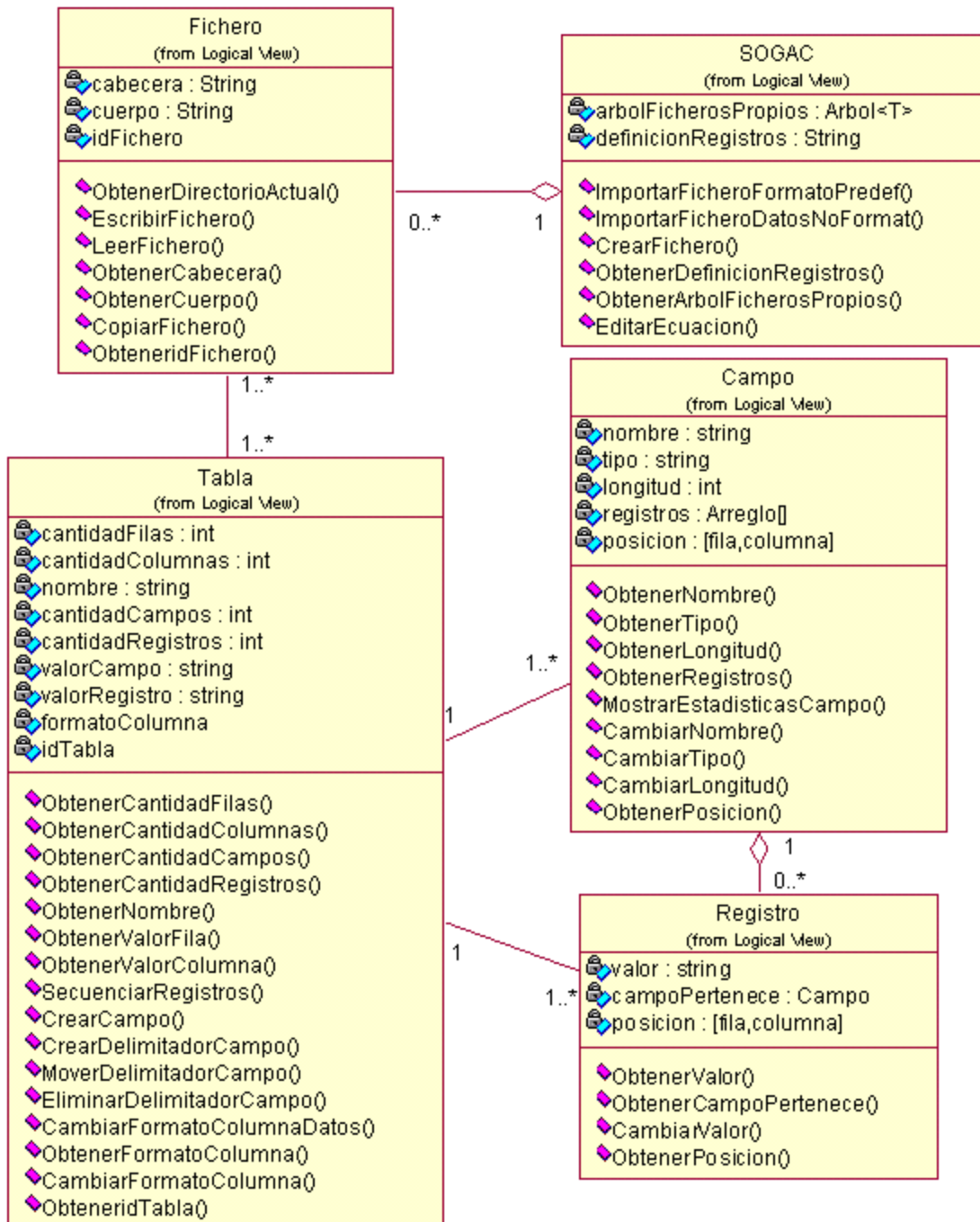


Fig. 3.19 Diagrama de clases persistentes.



### 3.7 MODELO DE DATOS.

Un modelo de datos, es una representación abstracta que describe cómo deben ser representados y relacionados los datos. Sirve para describir la estructura de la base de datos, así como los datos, sus relaciones y las restricciones que deben cumplirse entre ellos.

El modelo físico de los datos contiene un conjunto de tablas que conforman la base de datos. Este modelo constituye entonces la representación lógica y física del modelo de clases persistentes visto anteriormente.

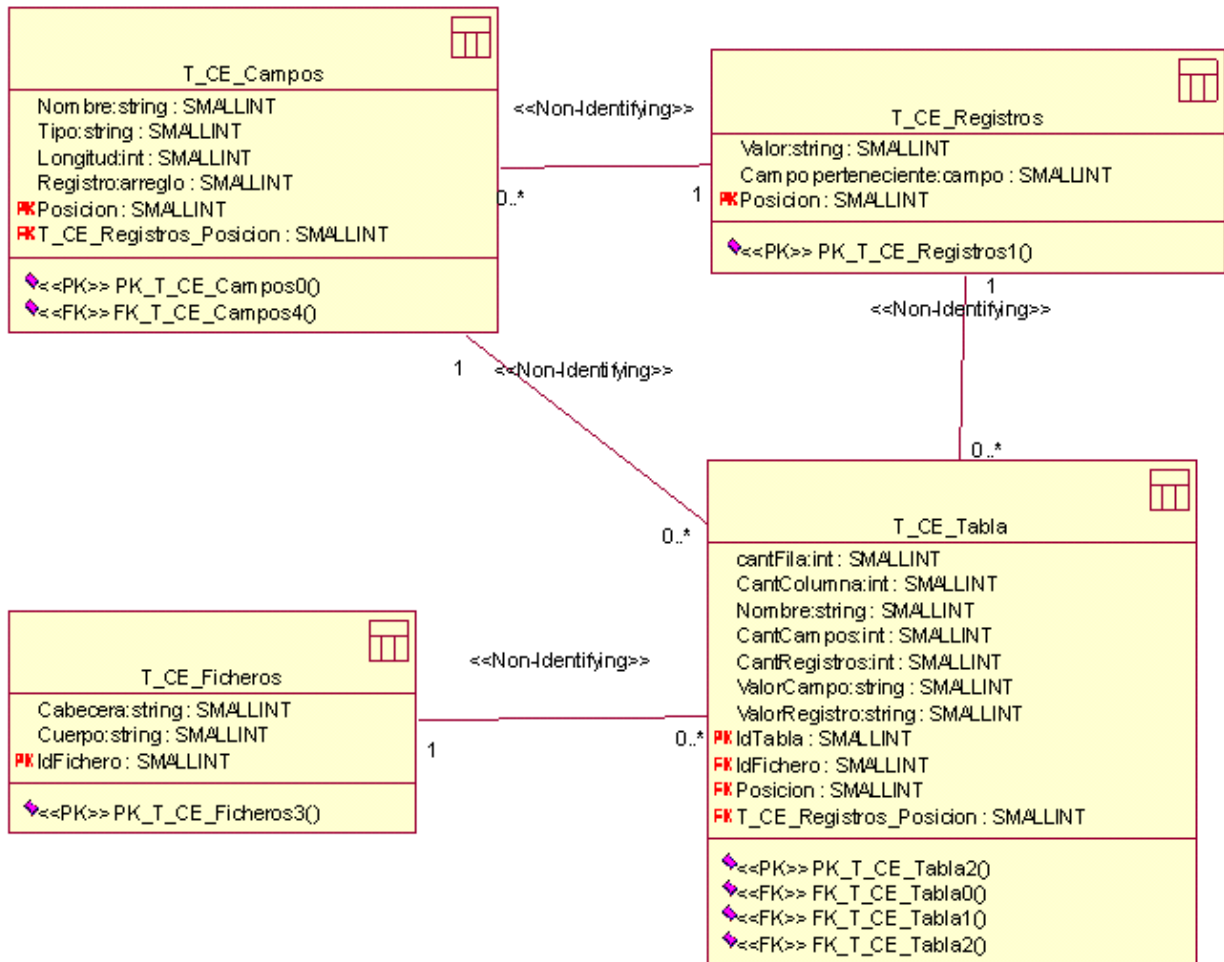


Fig. 3.20 Diagrama de Entidad Relación.

### **3.8 CONCLUSIONES.**

En este capítulo se exponen los resultados del flujo de trabajo de análisis y diseño. Se elaboraron los diagramas de clases del análisis para cada caso de uso así como los diagramas de interacción. También se elaboraron los diagramas de clases del diseño por cada caso de uso. Se realizó el modelo de datos a partir del diagrama de clases persistentes. De esta forma se han realizado las actividades principales comprendidas en la fase del análisis y el diseño.

## CONCLUSIONES

Luego de estudiar el proceso a informatizar y ejecutado las actividades propuestas según **RUP** en las fases de Negocio, Requerimientos y Análisis y Diseño, se obtuvieron todos los artefactos pertenecientes a cada una de ellas.

Se logró definir el entorno del dominio, y realizar la captura de los requisitos. Se definieron además el actor (Auditor) y los casos de uso del sistema (nueve en total), refinándolos y describiéndolos detalladamente.

En la etapa de análisis, se definieron cada una de las clases (interfaz, control y entidad), y se construyeron los diagramas de clases por cada caso de uso, así como los de colaboración, por cada escenario. En la etapa del diseño se definieron las clases del diseño y se realizaron los diagramas de secuencia, para lograr una mayor comprensión en los desarrolladores que estén responsabilizados de implementar la solución propuesta. Además se obtuvo el diagrama de clases persistentes y el modelo de datos.

Al concluir las tareas planteadas se le dieron cumplimiento a los objetivos generales y específicos propuestos al inicio de la investigación y se obtuvo el análisis y diseño del módulo de importación de ficheros y manejo de campos del Software General de Auditoría Cubano (SOGAC).

## RECOMENDACIONES

Luego de concluido el trabajo de análisis y diseño, recomendamos:

- Realizar la fase de implementación de este módulo de importación de ficheros y manejo de campos del Software General de Auditoría Cubano (**SOGAC**).
- Integrar a este módulo los procesos de exportación de datos.
- Integrar este módulo al Software General de Auditoría Cubano (**SOGAC**) como una herramienta acoplada, que permita realizar la importación de datos.
- Estudiar la posibilidad de acoplar esta herramienta a otros softwares que deban importar datos, como un ERP.
- Estudiar la posibilidad de ampliar los tipos de ficheros de formatos predefinidos que permite importar, incluyendo las bases de datos certificadas en Cuba.

## REFERENCIAS BIBLIOGRÁFICAS.

[1] Gary W. Hansen, James V. Hansen. Diseño y Administración de Bases de Datos (Segunda edición). México, Prentice Hall, Pág. 1.

[2] Raga Charlis. Base de Datos. 2008. Disponible en:  
[\[http://www.monografias.com/trabajos7/bada/bada.shtml#base\]](http://www.monografias.com/trabajos7/bada/bada.shtml#base)

[3] Manual de introducción al lenguaje SQL. 2008. Disponible en:  
[\[http://www.wikilearning.com/curso\\_gratis/manual\\_de\\_introduccion\\_al\\_lenguaje\\_sql-concepto\\_de\\_base\\_de\\_datos/4843-1\]](http://www.wikilearning.com/curso_gratis/manual_de_introduccion_al_lenguaje_sql-concepto_de_base_de_datos/4843-1)

[4] Ferrer Pineda Ch. Importación de datos. 2008. Disponible en:  
[\[http://www.google.com/search?q=cache:oAFeOmYK1sUJ:cheli.aradaen.com/wp-content/importacion.odp+manipulacion+datos+importar+ficheros+campos+registros&hl=es&ct=clnk&cd=3&gl=cu.\]](http://www.google.com/search?q=cache:oAFeOmYK1sUJ:cheli.aradaen.com/wp-content/importacion.odp+manipulacion+datos+importar+ficheros+campos+registros&hl=es&ct=clnk&cd=3&gl=cu.)

[5] Ivar Jacobson, Grady Booch y James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Massachusetts, USA, Addison Wesley, 2000, Pág. 4.

[6] Ivar Jacobson, Grady Booch y James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Massachusetts, USA, Addison Wesley, 2000, Pág.7.

[7] Ivar Jacobson, Grady Booch y James Rumbaugh. El Lenguaje Unificado de Modelado. Manual de Referencia. Massachusetts, USA: Addison Wesley, 2000, Pág. 3.

[8] Larman, C. (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. México, Prentice Hall, Pág.87.

[9] Craig Larman, C. (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. México, Prentice Hall, Pág. 41.

[10] Larman, C. (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. México, Prentice Hall, Pág. 6.

[11] Larman, C. (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. México, Prentice Hall, Pág. 52.

[12] Larman, C. (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. México, Prentice Hall, Pág. 54.

[13] Ivar Jacobson, Grady Booch y James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Massachusetts, USA, Addison Wesley, 2000, Pág. 40.

[14] Ivar Jacobson, Grady Booch y James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Massachusetts, USA, Addison Wesley, 2000, Pág. 38.

[15] Ivar Jacobson, Grady Booch y James Rumbaugh. El Proceso Unificado de Desarrollo de Software. Massachusetts, USA, Addison Wesley, 2000, Pág. 46.

## **BIBLIOGRAFÍA CONSULTADA.**

1. ACL Audit Analytics and Continuous Monitoring Software Solutions. 14/03/2008 [disponible en: <http://www.acl.com/products/desktop.aspx>]

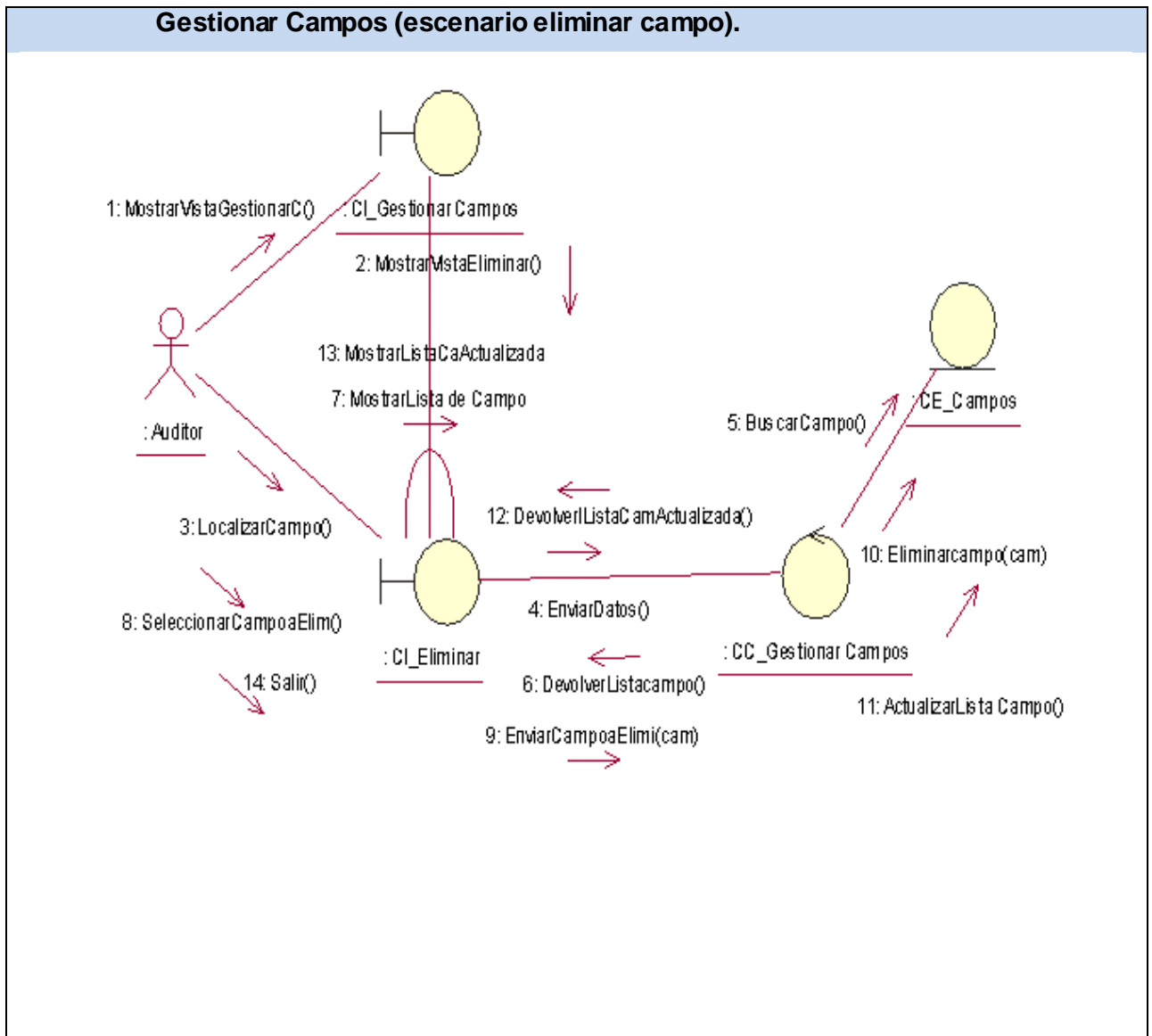
2. Agata Report, agata-report-tu-base-de-datos-al-descubierto. 26/11/2007. [disponible en: <http://www.agata.org.br/> ]

3. agata-report-tu-base-de-datos-al-descubierto. 23/01/2008 [disponible en: <http://www.aplicacionesempresariales.com/general/agata-report-tu-base-de-datos-al-descubierto.html>]

4. Crystal Reports Training, Consulting and Users Group .11/05/2008 [disponible en: <http://crystalreport.net/>]

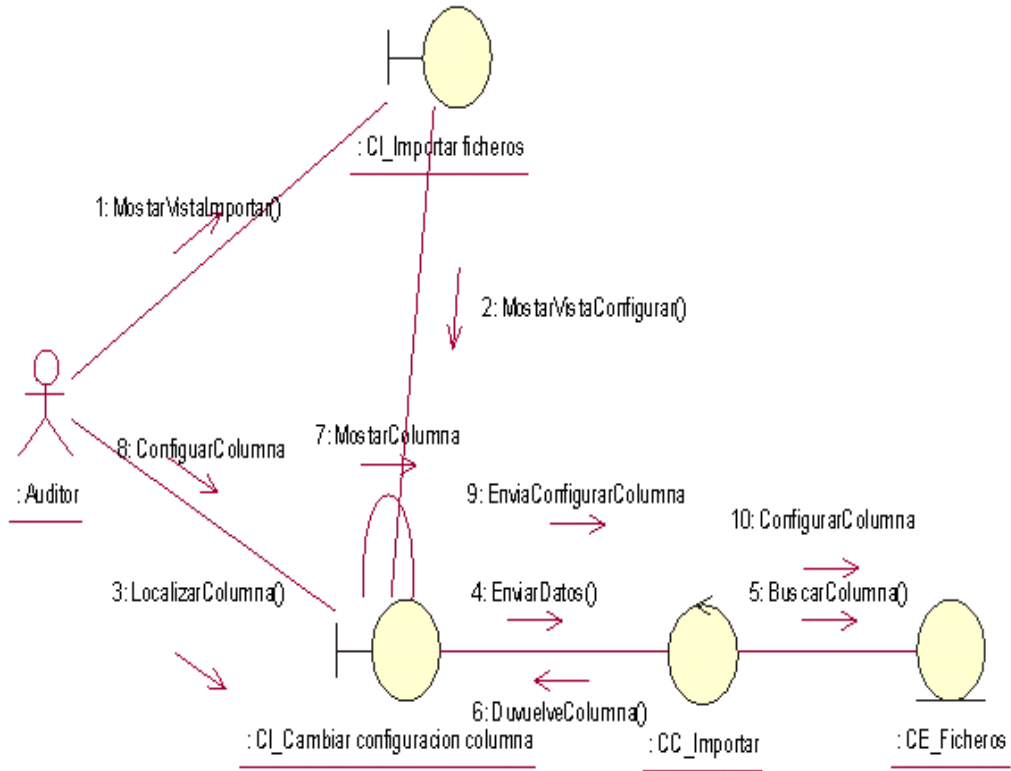
5. Félix Murillo, A. ¿QUE ES LA AUDITORÍA INFORMÁTICA? 01/012008 [disponible en: <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5105/Libro.pdf>]
6. Harwryszkiewicz, I T. Análisis y diseño de base de datos. México, Editorial Megabyte, 1994.
7. Jasper\_Reports- El Directorio.02/02/2008. [disponible en: [http://el-directorio.org/Jasper\\_Reports](http://el-directorio.org/Jasper_Reports)]
8. Joseph Schemuller. Aprendiendo UML en 24 horas. Mexico, Pentrice Hall. 2000.
9. Maria del Carmen Martínez, Dra. Orígenes y clasificación de la auditoría de la información. 15/06/2008. [disponible en: [http://bvs.sld.cu/revistas/aci/vol14\\_5\\_06/aci17506.htm](http://bvs.sld.cu/revistas/aci/vol14_5_06/aci17506.htm)]
10. Monografias.com. Auditoría Informática - Monografía. 08/12/2007 [disponible en: <http://www.monografias.com/trabajos/auditoinfo/auditoinfo.shtml>]
11. Pressman, Roges S. 2005. Ingeniería de Software un Enfoque Práctico. 2005.
12. SAFE Consulting Group. 09/04/ 2008 [disponible en: <http://www.safecg.com/Tools/IDEASAFE.pdf>]
13. Rolando Alfredo Hernandez Leon, Sayda Coello González. El paradigma cuantitativo de la investigación científica. Editorial Universitaria. Ciudad de la Habana, 2002.

**ANEXO1: DIAGRAMAS DE INTERACCIÓN (COLABORACIÓN DEL ANÁLISIS)**

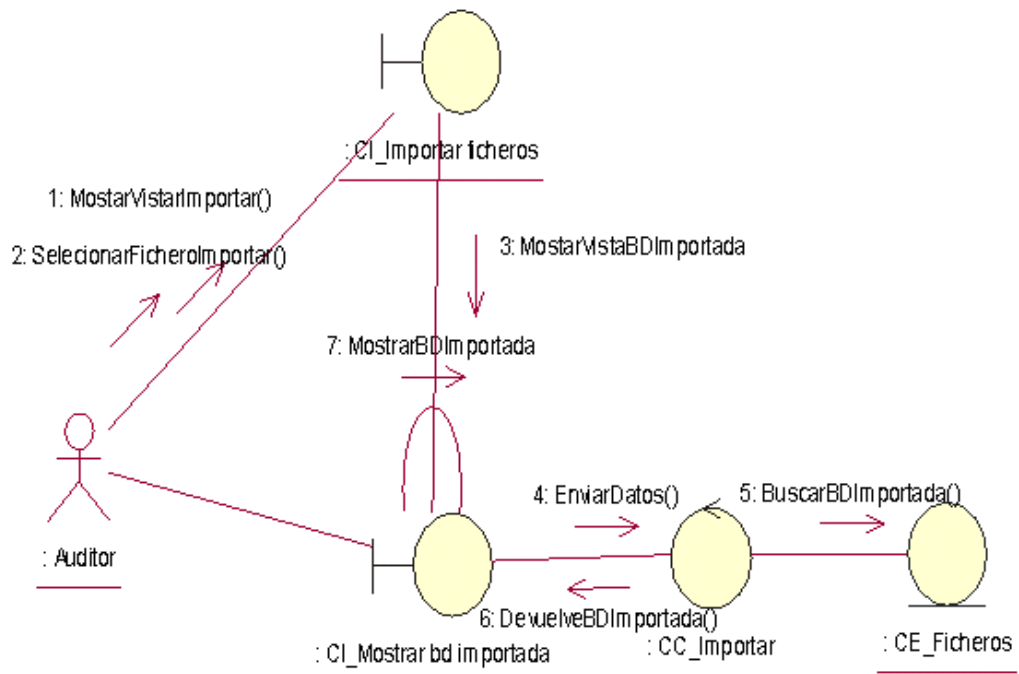




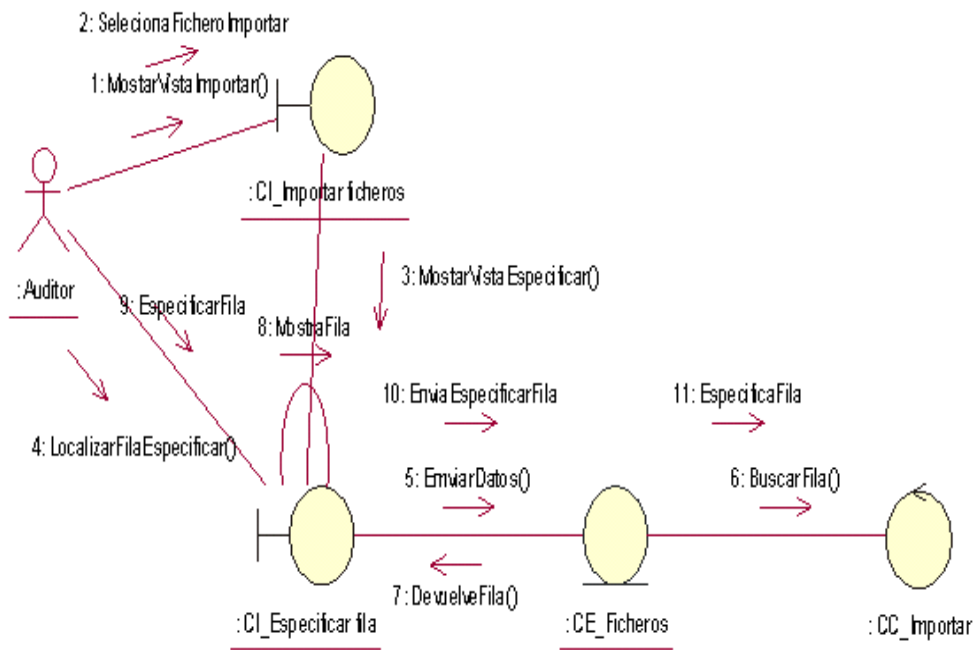
## Cambiar configuración de la columna de datos.



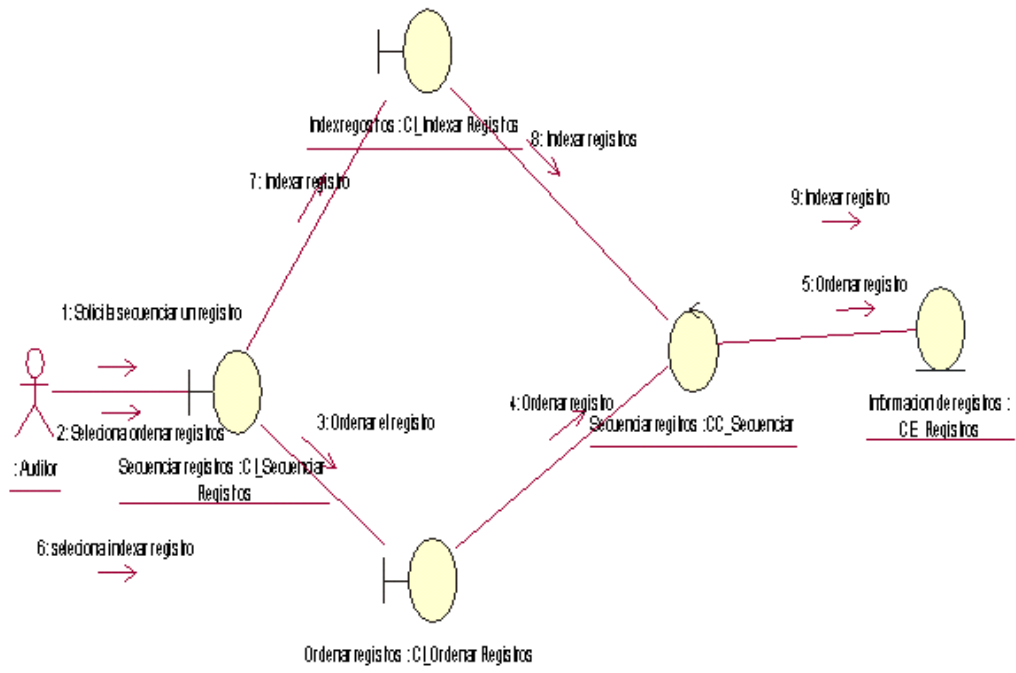
## Importar fichero de formatos predefinidos (escenario Mostar BD Importada).



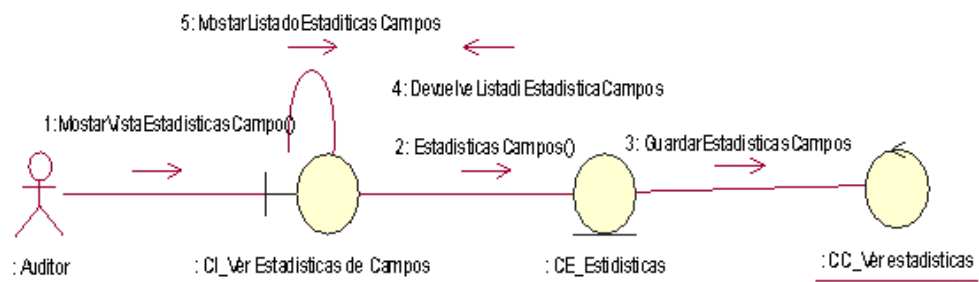
### Importar ficheros de formatos predefinidos (escenario Especificar Fila).



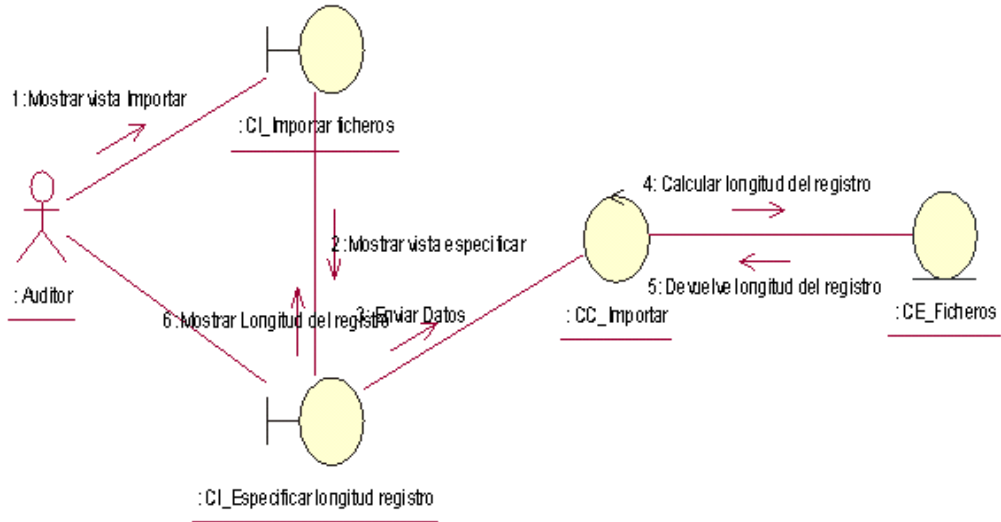
## Secuenciar registros.



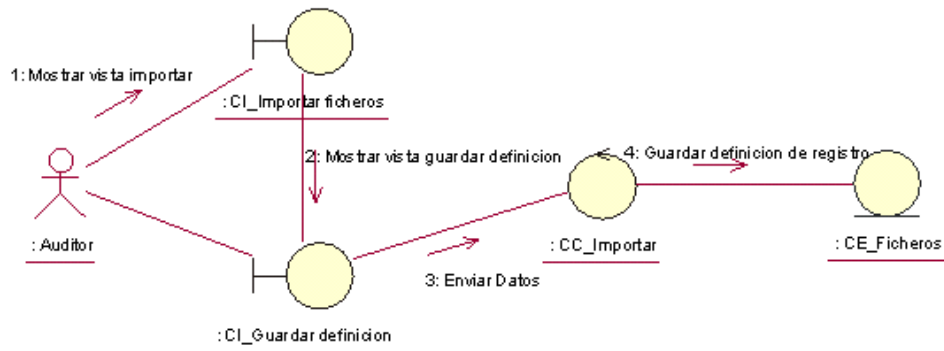
## Mostrar estadísticas de campos.



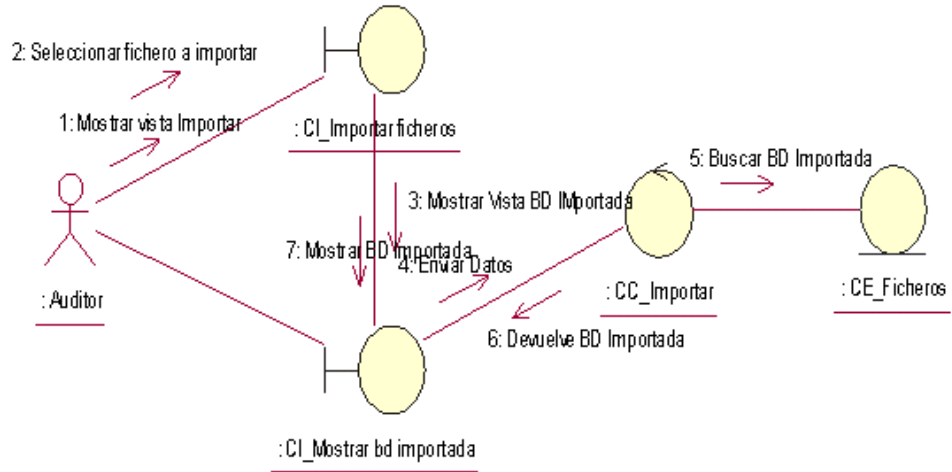
**Importar ficheros de datos no formateados (escenario Especificar la longitud del registro).**



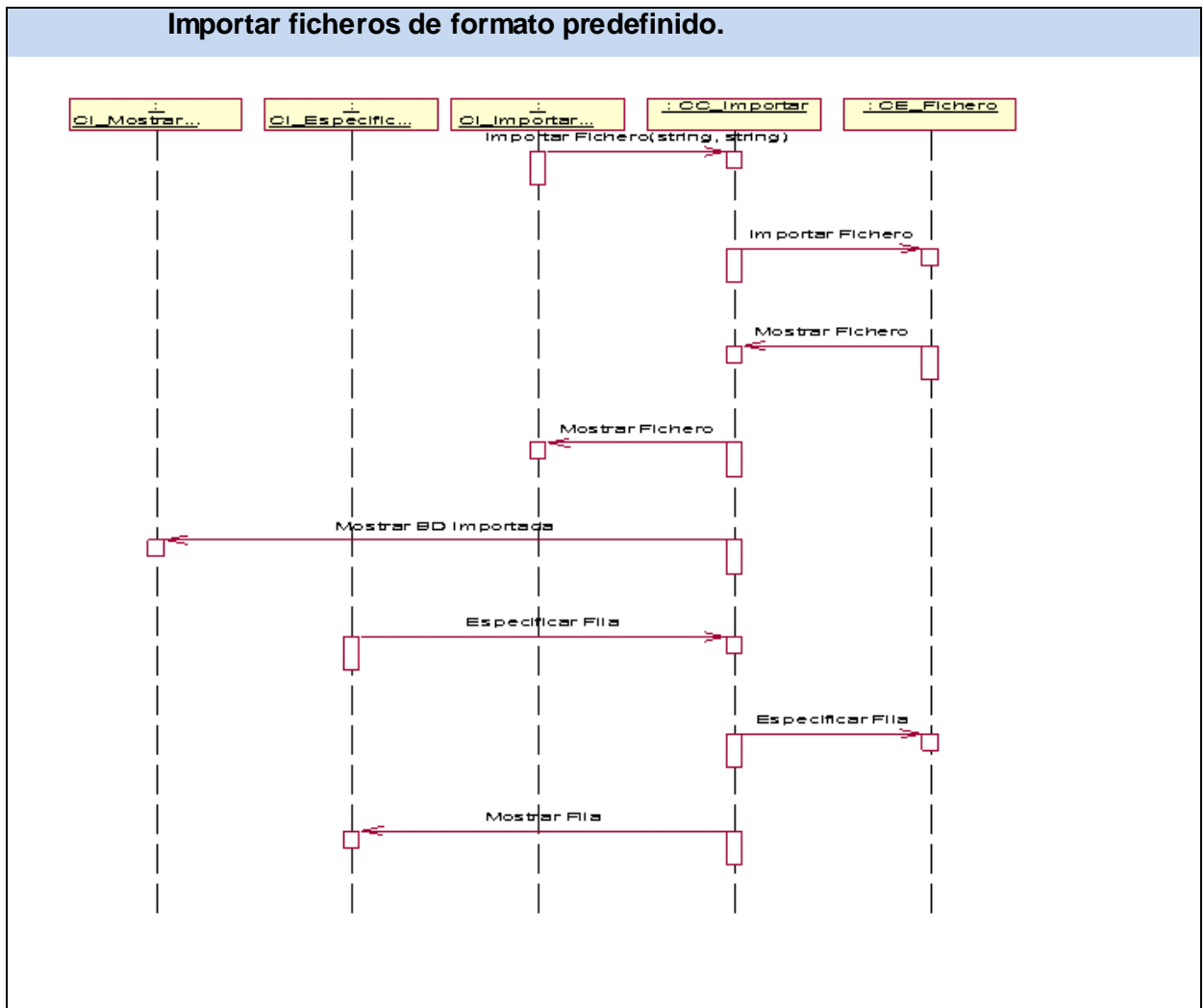
**Importar ficheros de datos no formateados (escenario Guardar definición de registro).**



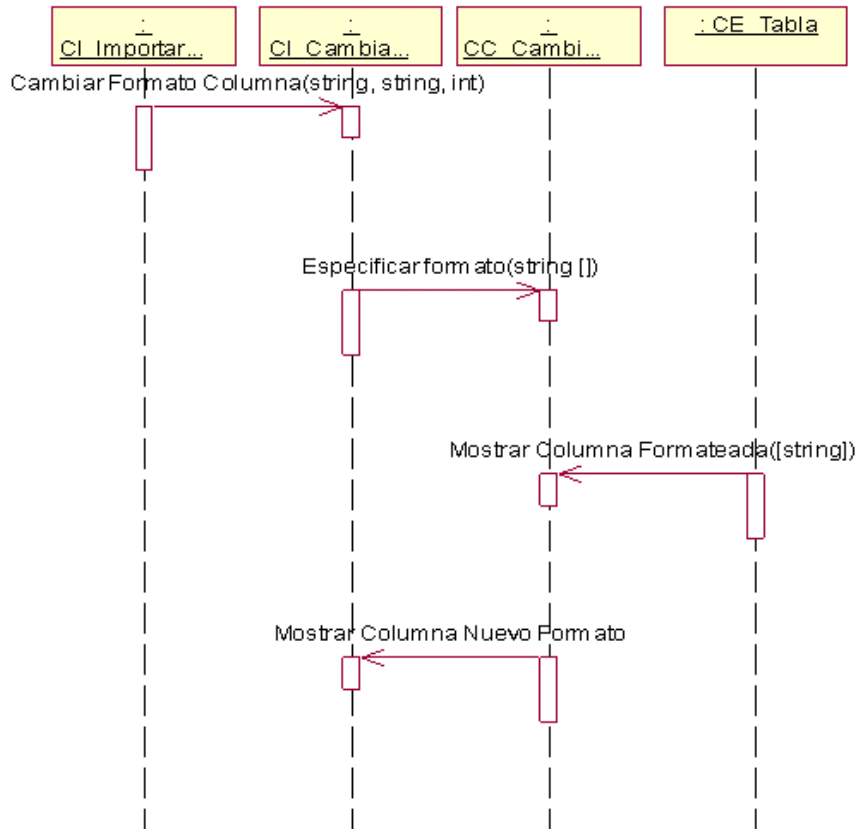
**Importar ficheros de datos no formateados (escenario Mostrar la base de datos importada).**



## ANEXO 2: DIAGRAMAS DE SECUENCIA DEL DISEÑO.

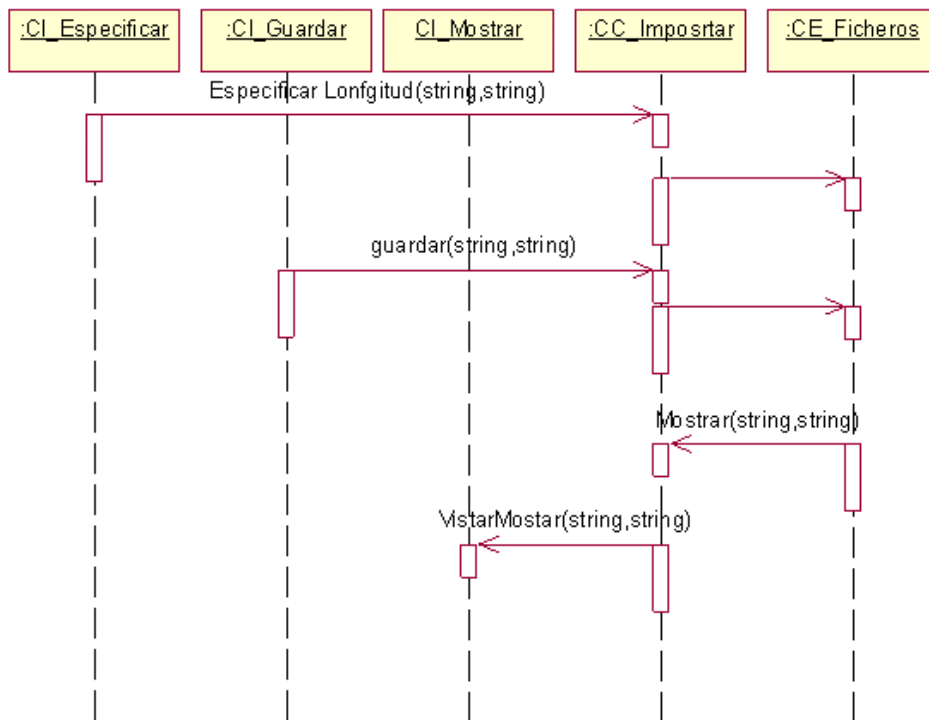


### Cambiar configuración de la columna de datos.

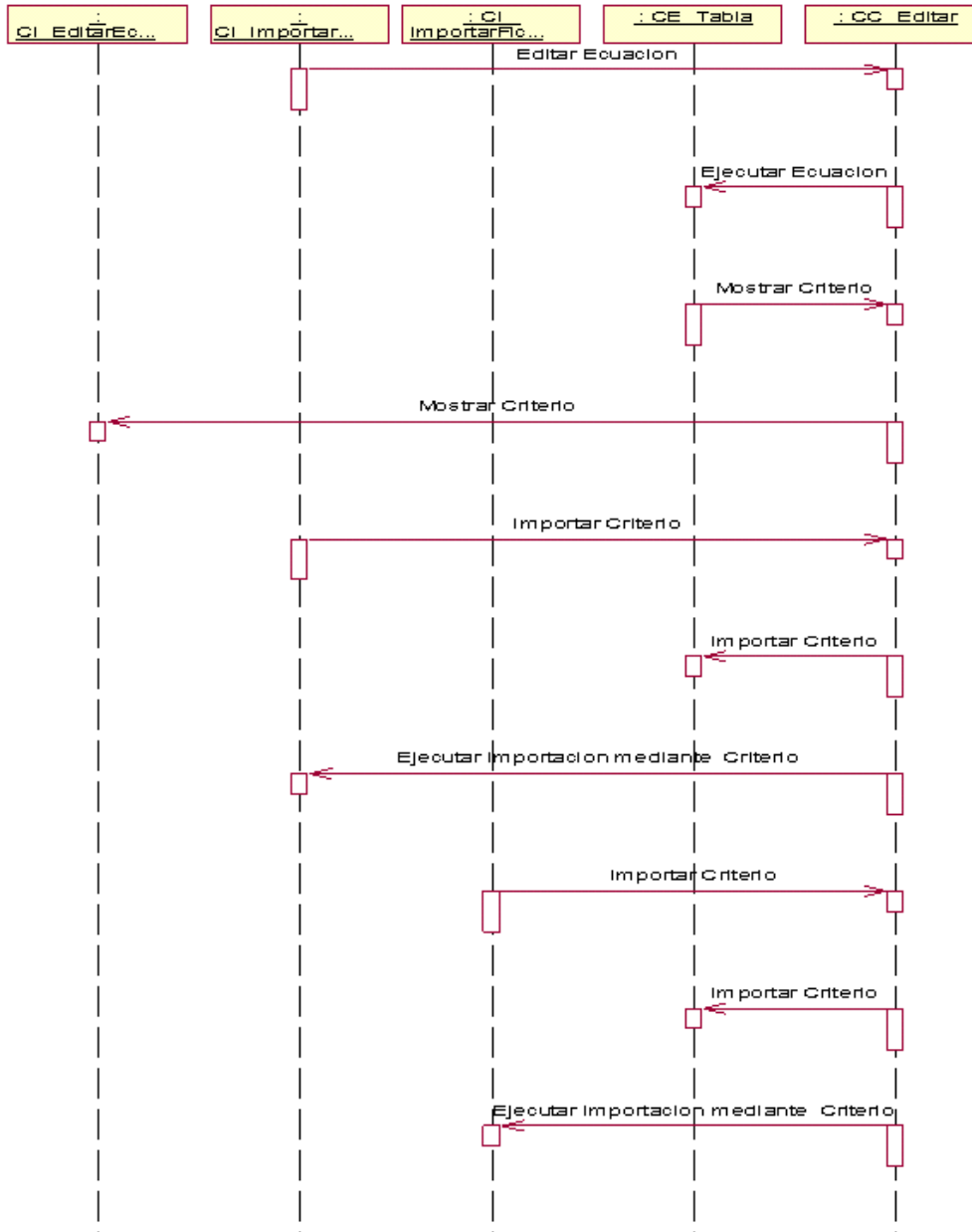




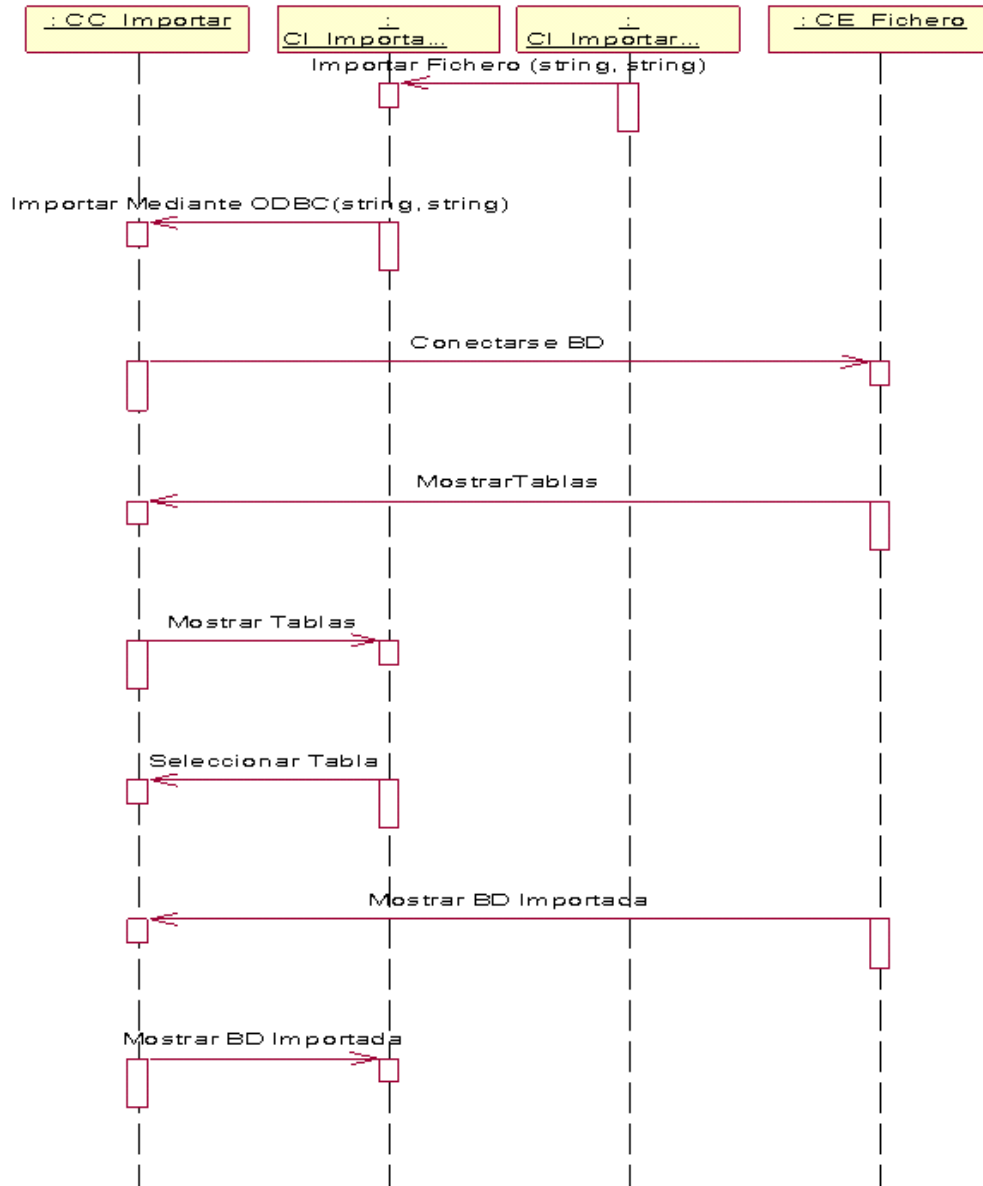
### Importar ficheros de datos no formateados.



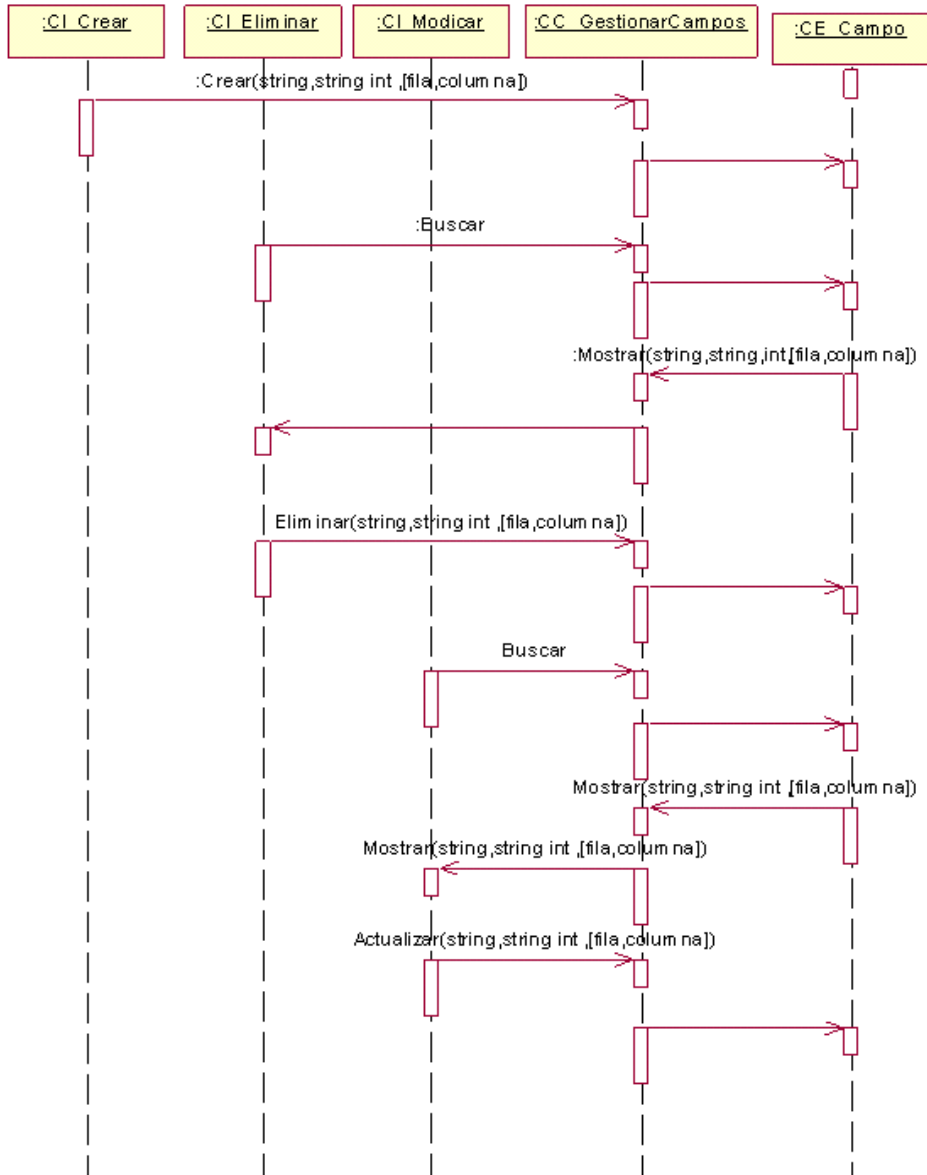
## Editar ecuación.



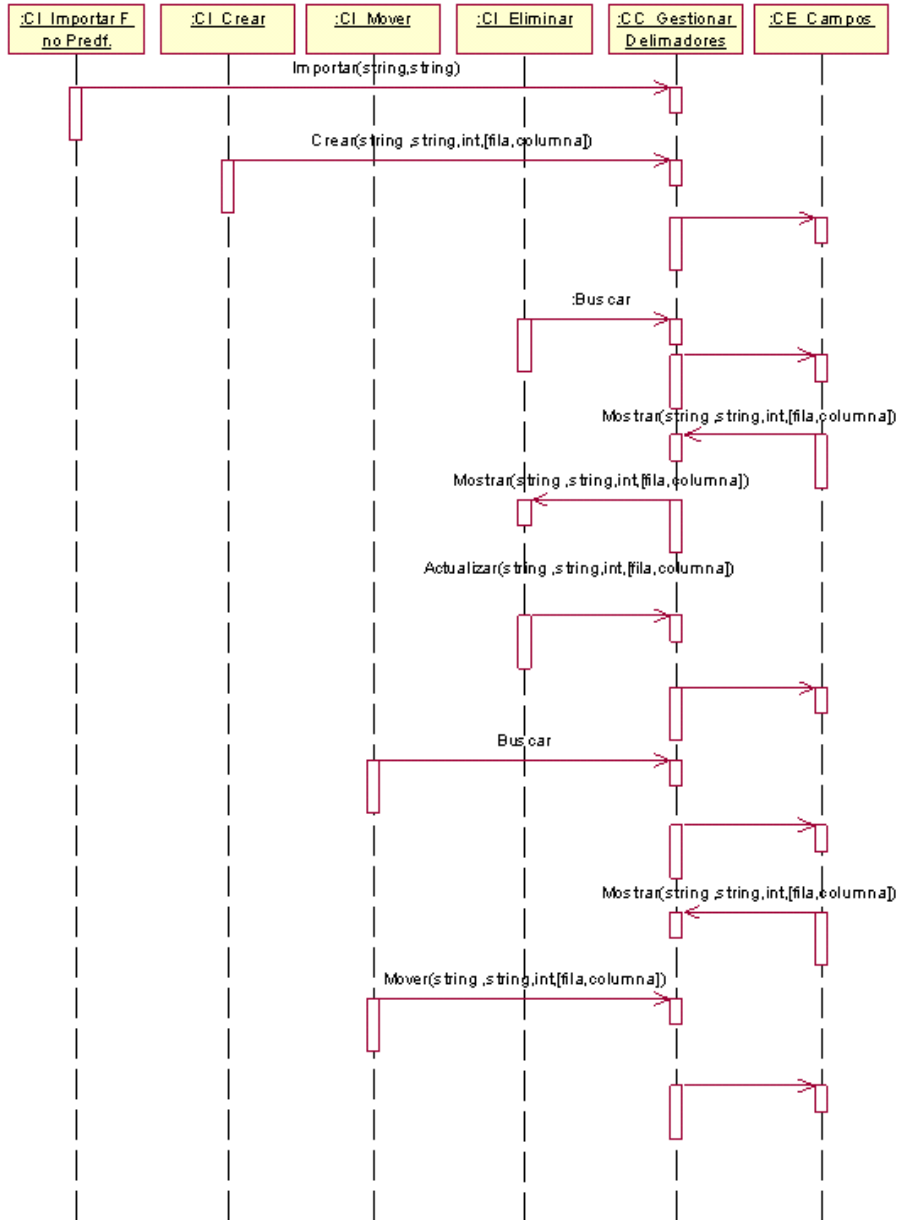
## Importar datos mediante ODBC.



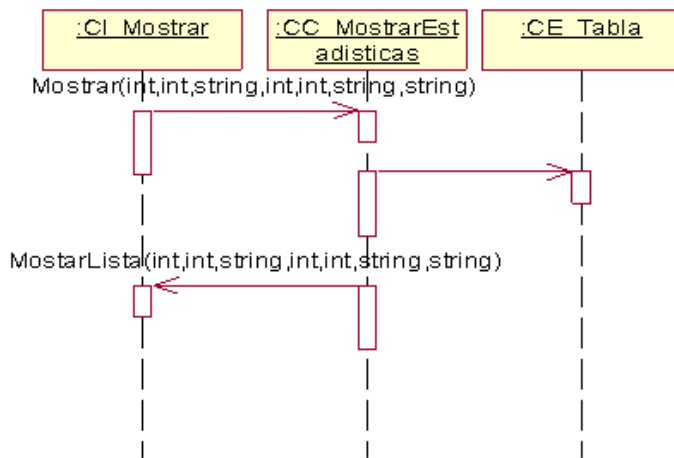
## Gestionar campos.



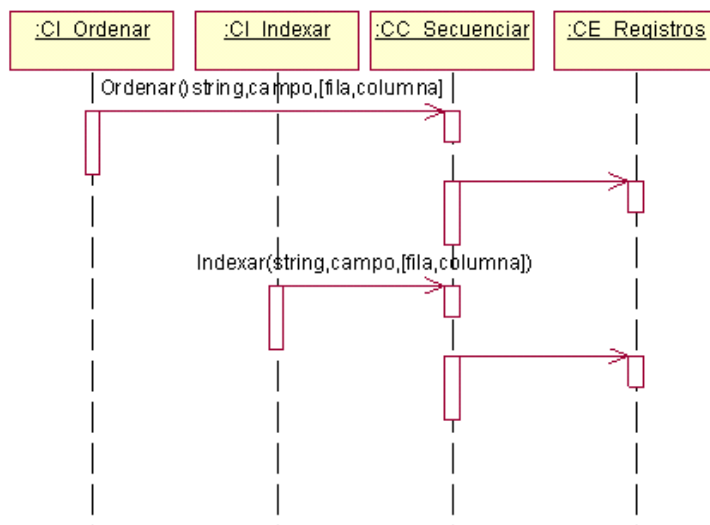
## Gestionar delimitadores de campos.



### Mostrar estadísticas de campos.



### Secuenciar registros.



### A

- 1- **ACL**: software para Auditoria financiera y análisis de datos.
- 2- **Adobe PDF** (Portable Document Format): es una aplicación que permite visualizar e imprimir archivos en formato PDF.
- 3- **AS/400**: (Antes llamado OS/400) es un sistema operativo de IBM. Muy usado en informática de gestión.
- 4- **ASCII** (American Standard Code for Information Interchange): es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales. Norma de codificación de caracteres en 7 bits, sin recomendación de paridad, empleada para el intercambio de datos entre sistemas de información y sistemas de comunicación de datos.

### B

- 5- **booleanos**: que puede tomar dos valores (verdadero o falso) (1 ó 2).

### C

- 6- **CASE** (Computer Aided Software Engineering): son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.
- 7-**COBOL** (Common Business Oriented Language): es un lenguaje común orientado a los negocios es decir, a la llamada informática de gestión.

### D

- 8- **DBase**: Sistema de gestión de base de datos usado ampliamente para microcomputadoras.
- 9- **DB2**: es un motor de base de datos relacional que integra XML de manera nativa.

### E

- 10- **EBCDIC** (Extended Binary Coded Decimal Interchange Code): es un código binario que representa caracteres alfanuméricos, controles y signos de puntuación. Cada carácter está compuesto por 8 bits = 1 byte, por eso EBCDIC define un total de 256 caracteres.

## H

**11- HTML** (Lenguaje de Marcado de Hipertexto): es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

## I

**12- IDEA:** software Auditoria y análisis de datos.

## L

**13- Licencia GNU/GPL** (General Public License ): es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre.

## M

**14- Microsoft Access:** es un programa Sistema de gestión de base de datos relacional creado y modificado por Microsoft para uso personal de pequeñas organizaciones, y actualmente integrado en el paquete Microsoft Office.

**15- Microsoft 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 actualmente integrado en el paquete Microsoft Office.

**16- Microsoft Word:** es un procesador de texto creado por Microsoft, y actualmente integrado en el paquete Microsoft Office.

**17- MySQL:** es uno de los Sistemas Gestores de Bases de Datos más populares.

## O

**18-ODBC** (Open DataBase Connectivity): es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de *ODBC* es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos.

**19- Oracle:** es una herramienta cliente/servidor para la gestión de Bases de Datos.

## P



**20- PHP (Hypertext Preprocessor):** lenguaje de script diseñado para la creación de páginas Web activas, muy popular en Linux, destaca por su capacidad de ser embebido en el código HTML.

S

**21- Sistema Gestor de Bases de Datos:** son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

**22- SOGAC (Software General de Auditoría Cubano):** es un software diseñado para el análisis de datos. Actualmente en fase de análisis y diseño.

T

**TAAC (Técnicas de auditoría asistida por computadora)** Cualquier técnica de auditoría automatizada, tal como el software general de auditoría, generadores de datos de prueba, programas de auditoría computadorizados, y sistemas expertos en auditoría.

X

**23- XML (Extensible Markup Language):** es un metalenguaje extensible de etiquetas, permite definir la gramática de lenguajes específicos.