

Universidad de las Ciencias Informáticas



Facultad 5

Título: Herramienta de importación de datos contenidos en hojas de cálculos xls hacia una base de datos postgres.

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

Autora:

Noridelys González Sobrecueva

Tutor:

Ing. Heidy Alina Nuevo León

Ing. Dayrien Corrales Díaz

Ciudad de la Habana, mayo del 2011



“La Revolución pudo resistir porque sembró ideas...son las ideas las que van a decidir esta lucha universal.”

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco 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 _____.

Noridelys González Sobrecueva

Firma del Autor

DATOS DE CONTACTO

Ing. Dayrien Corrales Díaz

dcorrales@uci.cu

Ingeniero en Ciencias Informáticas. Instructor con tres años de experiencia en el área productiva.

Líder de la línea “Gestión Industrial” del departamento “Construcción de Componentes” del “Centro de Desarrollo de Informática Industrial”.

Ing. Heidy Alina Nuevo León

hanuevo@uci.cu

Ingeniera en Ciencias Informáticas. Adiestrada con dos años de experiencia en el área productiva.

Líder de proyecto “Sistema de Confiabilidad Integral de Activos”.

Agradecimiento

Agradezco a mi padre, por su apoyo, dedicación y entrega para conmigo, por ser el mejor papá que la vida me podía dar. A mi mamá por sus consejos y por ser la buena amiga que siempre necesito tener. A mis hermanitas, Normy, Lorena y Nailan. Especialmente les agradezco a mis abuelos por poner todo su empeño en educarme y guiarme para hoy ser lo que soy. A mis tíos, primos y el resto de la familia ya que todos han estado pendiente de mí siempre.

Agradezco infinitamente a mi tutora Heidy por saber exigirme y guiarme en mi trabajo, por soportar mis nervios antes de cada corte. A mi tutor Dayrien por ayudarme tanto, por responder las tantas preguntas que día a día le he hecho. A Cecilia, por darme el impulso que necesitaba para empezar mi tesis. A la profesora Oneida porque gracias a sus consejos tuve la fuerza para seguir adelante y hoy estoy aquí escribiendo estas líneas. A Niriam, esa personita tan especial que siempre se preocupa por mí.

A Deborat, cuantas cosas por agradecerte, siempre te estaré agradecida por preocuparte y ocuparte de mí en estos años de amistad. A mis amigos que durante los cinco años de la carrera han convivido conmigo. A los buenos profesores que he tenido para llegar a esta primera meta. Por sobradas razones agradezco a todos los que de una forma u otra han puesto un granito de arena en mi formación.

A todos...gracias.

Dedicatoria

A ti papito querido.

Resumen

El presente trabajo de diploma está centrado en el desarrollo de una herramienta que permita automatizar el proceso de importación de datos contenidos en hojas de cálculo (HC) xls a la Base de Datos (BD) postgres de un proyecto.

Incluye además un estudio de las funcionalidades implementadas en herramientas similares a la desarrollada y de las tecnologías más convenientes para concretar estas funcionalidades y dar cumplimiento a los objetivos propuestos.

Como resultado, se obtuvo una herramienta que permite llevar a cabo la importación de datos contenidos en hojas de cálculo xls a una BD postgres; lo que constituye un aumento del nivel de usabilidad, confiabilidad, seguridad y otros aspectos importantes en el funcionamiento de la BD de un proyecto.

Palabras Clave:

Automatización, Base de Datos, Herramientas, Hoja de Cálculo, Importación.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1. Introducción	5
1.2. Conceptos Fundamentales	5
1.2.1. Hoja de Cálculos.....	5
1.2.2. Importar	5
1.2.3. Archivos XML	6
1.2.4. Base de Datos	6
1.2.5. Sistemas Gestores de Base de Datos	6
1.3. Herramientas existentes para la importación de datos.....	7
1.3.1. Plug-in para Microsoft Excel	7
1.3.2. Navicat	7
1.3.3. EMS Data Import para PostgreSQL	8
1.3.4. Sypex.....	9
1.4. Herramientas y Tecnologías.....	10
1.4.1. Sistema Operativo: GNU/Linux.....	10
1.4.2. Lenguaje de Modelado: UML.....	11
1.4.3. Metodología: RUP	11
1.4.4. Lenguaje de programación: Python	12
1.4.5. Biblioteca Gráfica: QT	13
1.4.6. Biblioteca: XLRD	14
1.4.7. ORM para python: SQLAlchemy	14
1.4.8. Gestor de Base de Datos: PostgreSQL	14
1.4.9. Herramienta Case: Visual Paradigm	15
1.4.10. IDE: Eclipse.....	15
1.4.11. Herramienta de diseño: Framework Qt.....	15
1.5. Conclusiones	16
Capítulo 2. Descripción de la solución propuesta	16
2.1. Introducción	16
2.2. Funcionalidades del sistema	17
2.3. Requisitos del sistema	17
2.3.1. Requisitos Funcionales.....	18
2.3.2. Requisitos no funcionales	18

2.4.	Descripción del sistema propuesto.....	21
2.4.1.	Descripción de actores.....	21
2.4.2.	Diagrama de casos de usos del sistema.....	21
2.4.3.	Descripción de los casos de usos del sistema	21
2.4.4.	Diagramas de secuencia	30
2.5.	Estilos arquitectónicos	34
2.6.	Patrones de diseño	36
2.7.	Modelo de diseño	37
2.8.	Conclusiones	38
Capítulo 3.	Implementación y Prueba	39
3.1.	Introducción	39
3.2.	Diagrama de clases por paquetes	39
3.3.	Descripción de las clases por paquete	42
3.4.	Diagrama de componente	44
3.5.	Estándar de Codificación.....	45
3.6.	Estándar de Documentación.....	46
3.7.	Diagrama de Despliegue	46
3.8.	Visión de la Aplicación	47
3.9.	Pruebas	49
3.10.	Conclusiones	52
Conclusiones Generales		53
Recomendaciones		54
Bibliografía.....		55
Glosario de Términos.....		57
Anexos		58

INDICE DE FIGURAS

Ilustración 1.	Menú en el que se puede encontrar la funcionalidad del plug-in para Excel.	7
Ilustración 2.	Funcionalidades del menú que proporciona el plug-in para Excel.....	7
Ilustración 3.	Página principal del Navicat.....	8
Ilustración 4.	Página principal del EMS Data Import para PostgreSQL.	9

Ilustración 5. Función Exportar de la herramienta de respaldo "Sypex".	10
Ilustración 6. UML.	11
Ilustración 7. Metodología RUP.	12
Ilustración 8. Diagrama de CUS.	21
Ilustración 9. Diagrama de secuencia del CU "Cargar xls".	31
Ilustración 10. Diagrama de secuencia del CU "Conectar BD".	31
Ilustración 11. Diagrama de secuencia del CU "Importar hacia la BD".	32
Ilustración 12. Diagrama de secuencia del CU "Administrar Configuración", sesión: "Abrir Configuración"...	32
Ilustración 13. Diagrama de secuencia del CU "Administrar Configuración", sesión: "Aplicar Configuración".	33
Ilustración 14. Diagrama de secuencia del CU "Administrar Configuración", sesión: "Guardar Configuración".	33
Ilustración 15. Diagrama de secuencia del CU "Administrar Configuración", sesión: "Eliminar Configuración".	33
Ilustración 16. Arquitectura en tres capas.	35
Ilustración 17. Modelo-Vista-Controlador.	35
Ilustración 18. Diagrama de la arquitectura.	36
Ilustración 19. Diagrama de paquetes.	38
Ilustración 20. Diagrama de clases del paquete "presentation".	39
Ilustración 21. Diagrama de clases del paquete "business".	40
Ilustración 22. Diagrama de clases del paquete "domain".	40
Ilustración 23. Diagrama de clases del paquete "dao".	41
Ilustración 24. Diagrama de clases del paquete "util".	41
Ilustración 25. Descripción de las clases del paquete "presentation".	42
Ilustración 26. Descripción del paquete "business".	43
Ilustración 27. Descripción del paquete "domain".	43
Ilustración 28. Descripción del paquete "dao".	44
Ilustración 29. Descripción del paquete "util".	44
Ilustración 30. Diagrama de componentes.	45

Ilustración 31. Diagrama de Despliegue.....	46
Ilustración 32. Splash inicial de la aplicación.....	47
Ilustración 33. Ventana principal de la aplicación.....	48
Ilustración 34. Espacio reservado para mostrar la estructura de la BD y el contenido del xls.	48
Ilustración 35. Ventana de autenticación de la BD.	49
Ilustración 36. Por ciento que representan de las pruebas para “Importar”.....	51
Ilustración 37. Por ciento que representan las pruebas para “Administrar Configuración”.	51

INDICE DE TABLAS

Tabla 1. Descripción de los actores.	21
Tabla 2. Descripción del CU “Cargar xls”.	23
Tabla 3. Descripción del CU “Conectar BD”.	26
Tabla 4. Descripción del CUS “Importar hacia la BD”.	28
Tabla 5. Descripción del CUS “Administrar Configuración”.	30
Tabla 6. Resultados de las pruebas.	50

Introducción

A lo largo de los años el desarrollo de la informática ha beneficiado el desarrollo industrial dotando a las grandes industrias de aplicaciones informáticas para el control de procesos industriales. Además se han empleado las HC para el procesamiento de datos que se deben registrar en la BD rectora de cada proceso que se controle.

La Universidad de la Ciencias Informáticas (UCI) ha originado un esquema estudio-trabajo donde los estudiantes se vinculan directamente a la producción. Este se basa en integrar los procesos de formación, investigación y producción en torno a una temática para convertirla en una rama productiva. Dicha integración es denominada “Centro de Desarrollo” y en la facultad 5 específicamente se encuentra el “Centro de Desarrollo de Informática Industrial” (CEDIN) el cual está estructurado por departamentos, uno de ellos es el de componentes que a su vez cuenta con líneas de desarrollo de componentes entre ellas: “Gestión Industrial”, encargada de agrupar proyectos relacionados con procesos industriales.

En el proyecto “Sistema de Confiabilidad Integral de Activos (SCIA)” asociado a la línea “Gestión Industrial” se ha detectado un problema al procesar gran cantidad de información contenida en HC xls generadas por la metodología, implementada en el sistema, llamada “Política de Cuidado Integral de Activos (PCIA)” que deben ser persistida en la BD del proyecto, dicha migración la lleva a cabo el administrador de la BD de forma manual lo que dificulta el procesamiento de la información. Este problema trae consigo una serie de inconvenientes como:

- Pérdida de tiempo, pudiendo emplearlo en otras tareas de mayor importancia.
- Ocurrencia de errores al copiar los datos de un lugar a otro de forma manual.
- Pérdida de privacidad en la BD, ya que se debe mostrar la información que en ella está almacenada.

El problema identificado puede existir no solo en el proyecto en el que se identificó, sino que puede tener lugar en cualquier proyecto asociado a la línea que necesite acceder desde su BD a datos históricos que estén contenidos en HC xls.

Partiendo de la **Situación Problemática** antes planteada surge el **Problema Científico**: ¿Cómo garantizar la integridad de la información durante el proceso de importación de datos contenidos en hojas de cálculos xls?

El **Objeto de Estudio**: herramientas para la importación de datos, y el **Campo de Acción**: herramientas para la importación de datos a partir de hojas de cálculos. Conforme a lo planteado en el problema científico se define la siguiente **Idea a Defender**: el desarrollo de la herramienta propuesta debe permitir automatizar el proceso de importación de datos contenidos en HC xls hacia una BD postgres.

El **Objetivo** de este trabajo de diploma es desarrollar una herramienta que garantice la importación de datos contenidos en HC xls hacia una BD postgres evitando la pérdida o modificación de la información.

Para darle cumplimiento al objetivo planteado se definieron las **Tareas Investigativas** que se listan a continuación:

- Selección y caracterización de los conceptos fundamentales para entender los términos a tratar.
- Caracterización de las funcionalidades básicas de herramientas similares a la que se va a desarrollar.
- Levantamiento de requisitos del software para el desarrollo de la herramienta.
- Selección de las herramientas libres más convenientes y eficientes para cumplir con los requisitos antes definidos.
- Descripción y diseño de los Casos de Uso del sistema (CUS) para documentar el funcionamiento de la herramienta.
- Diseño e implementación de la herramienta de importación de datos.
- Desarrollo de las pruebas de la herramienta para probar sus funcionalidades básicas.

Con el presente trabajo se pretende proveer a la comunidad de desarrolladores de los proyectos asociados a la línea “Gestión Industrial” de una herramienta que agilizará todo lo referente al procesamiento de datos dentro de un proyecto.

Para desarrollar las tareas investigativas propuestas se combinaron varios métodos de procesamiento de la información que a continuación se explican:

Métodos Teóricos:

Analítico-Sintético: Mediante este método se realizó el procesamiento de la información, analizándose documentos y conceptos, haciéndose énfasis en la profundización de los mismos con el objetivo de caracterizar el objeto de estudio. Una vez entendido este, la situación problemática y la teoría, se pasó entonces a la unión de todos estos conocimientos para dar solución al problema.

Histórico-Lógico: Este método brindó la posibilidad de conocer la evolución de los sistemas de importación de datos que han surgido a nivel mundial.

Métodos Empíricos:

Observación: este método permitió hacer una observación del problema existente para tener una mejor idea de cómo se espera que funcione la herramienta a desarrollar.

Entrevistas: Constituyó un medio para el aumento de los conocimiento acerca de lo que se ha venido haciendo para importar los datos de una HC hacia una BD usando software libre.

Estructura del documento

El siguiente trabajo consta de 3 capítulos.

En el **capítulo 1: Fundamentación teórica**, se analiza el Estado del Arte. Se hace un bosquejo de los conceptos necesarios para un mejor entendimiento del tema. Se describen las características con las que debe cumplir la solución propuesta. Y se describen las herramientas libres más convenientes para el desarrollo de la herramienta.

En el **capítulo 2: Descripción de la solución propuesta**, se expone una visión práctica del sistema, describiendo su funcionamiento, así como los requisitos funcionales y no funcionales. Se describe el desarrollo de la solución propuesta a partir de la modelación de diagramas de casos de uso y diagramas de secuencia. Se especifican los patrones de diseño y los estilos arquitectónicos para enriquecer la arquitectura del sistema.

En el **capítulo 3: Implementación y pruebas**, se diseña un sistema de clases, en correspondencia con las técnicas de programación usadas, que luego son implementadas obteniendo una herramienta capaz de importar datos contenidos en un documento Excel a una BD bajo los principios de software libre. Y como punto final se realizan las pruebas necesarias para validar la utilidad de la herramienta en el ámbito que se necesita.

Finalmente como parte de los anexos se muestra el Glosario de Términos y Abreviaturas (GTA) para un mejor entendimiento del lenguaje técnico y las abreviaturas utilizadas.

Capítulo 1: Fundamentación Teórica

1.1. Introducción

En el presente capítulo se describen detalladamente los conceptos que sustentan la herramienta a desarrollar, como fundamentación teórica base para obtener una amplia comprensión del problema a resolver y de su entorno. Se darán a conocer algunas de las soluciones que hasta el momento se han implementado para llevar a cabo la importación de datos desde HC. Y finalmente se abordará todo lo relacionado con las herramientas y tecnologías a utilizar en el desarrollo de la herramienta para importar datos contenidos en HC xls hacia una BD Postgres.

1.2. Conceptos fundamentales

1.2.1. Hoja de cálculos

Una **hoja de cálculos** es un programa que permite trabajar y operar en la máquina con datos alfanuméricos distribuidos en filas y columnas, de manera que pueden ejecutarse diversas operaciones aritméticas de todo tipo de complejidad, y también gráficos. (Lanzillotta, 2005)

Otra de las funcionalidades que actualmente se le da a las HC es la creación de bases de datos, y también para la preparación de informes, cuadros estadísticos, planillas, entre otras aplicaciones.

1.2.2. Importar

La palabra **importar** es muy usada en varios campos como son la economía y la informática por lo que se definen algunos conceptos para un mejor entendimiento de este término. En la economía importar quiere decir: compra de bienes y servicios procedentes del extranjero. Y en la informática traer de otra parte algo. La función Importar se utiliza para incorporar a una herramienta archivos ya sea imagen, video, sonido, etc. Necesarios para su funcionamiento.

1.2.3. Archivos XML

XML (Extensible Markup Language, lenguaje de marcado extensible): es una forma de reutilizar datos en un archivo o de automatizar el proceso de sustitución de datos en un archivo con datos de otro archivo. Utiliza etiquetas para describir partes de un archivo, por ejemplo, un encabezado o un artículo. Estas etiquetas marcan los datos de manera que se pueden almacenar en un archivo de este tipo y se puedan gestionar adecuadamente cuando se exporten a otros archivos. Piense en XML como un mecanismo de conversión de datos. Dicho lenguaje etiqueta el texto de la etiqueta y otro contenido en un archivo de forma que las aplicaciones puedan reconocer y presentar los datos. (2007)

1.2.4. Base de datos

Se define **BD**: al conjunto de datos organizados y relacionados entre sí, los cuales son explotados y recolectados por los sistemas de información de una empresa en particular. (Pérez Valdéz, 2007)

Entre las principales características de las BD se encuentran, que propician una independencia lógica y física de los datos, logran una redundancia mínima, permiten el acceso concurrente por parte de múltiples usuarios, garantizan la integridad de los datos, optimizan consultas de alta complejidad, admiten la seguridad de acceso y auditoría, el respaldo y la recuperación, además de que a ellas se puede acceder a través de lenguajes de programación estándar.

1.2.5. Sistemas Gestores de Base de Datos

Un **Sistema Gestor de Bases de Datos (SGBD)** es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (2010)

1.3. Herramientas existentes para la importación de datos

1.3.1. Plug-in para Microsoft Excel

Plug-in para Microsoft Excel facilita la importación de datos hacia una base de datos, este está disponible para ser instalado a cualquier versión de Microsoft Office, incluyendo un menú extra que muestra varias opciones como: configuración de la conexión con el servidor, importar una tabla de MySQL a Excel, exportar datos de celdas a tabla MySQL, exportar datos de celdas a una tabla existente, etc. (2008)

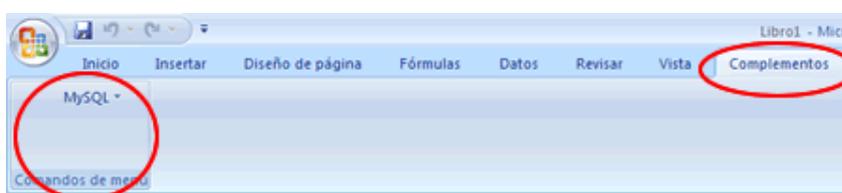


Ilustración 1. Menú en el que se puede encontrar la funcionalidad del plug-in para Excel.

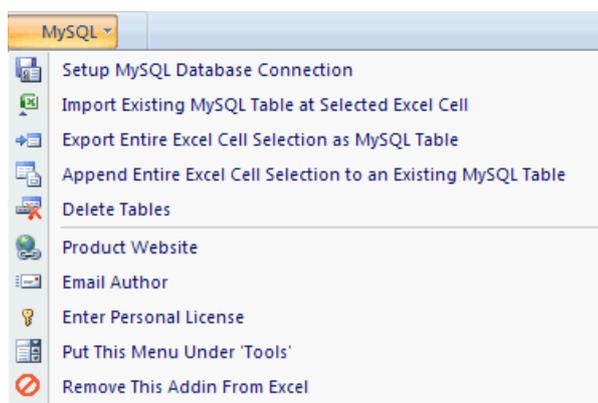


Ilustración 2. Funcionalidades del menú que proporciona el plug-in para Excel.

1.3.2. Navicat

Navicat es una Interfaz gráfica de usuario que incluye todo lo necesario para una interfaz gráfica poderosa en la administración, desarrollo y mantenimiento de bases de datos. Interfaz de instalación fácil e intuitiva que la convierten en una herramienta impecable para PostgreSQL en la red o en su PC de escritorio local. Navicat le permite convertir Access a

PostgreSQL, MS SQL a PostgreSQL, Excel a PostgreSQL, sincroniza y respalda bases de datos en un instante con la muy útil ayuda de un asistente. (2009)

Sus mayores características incluyen un Constructor Visual de Consultas, Opciones de Exportación/Importación, Constructor de Reportes, Opciones de Copia de Seguridad / Restauración, Túneles Seguros SSH y Transferencias de Datos. Navicat también soporta la importación de datos desde ODBC (Conectividad abierta de base de datos), programación de trabajos en lote (creación de programación para Importación/Exportación, Transferencias de datos y guardar consultas), impresión de estructuras de tablas. Incluye también un administrador gráfico para la configuración de los privilegios de acceso y de los usuarios. La versión 8 de Navicat admite Conclusión de Códigos, Visualización de Formularios, Servicio de Notificaciones vía Email, etc. (2009)

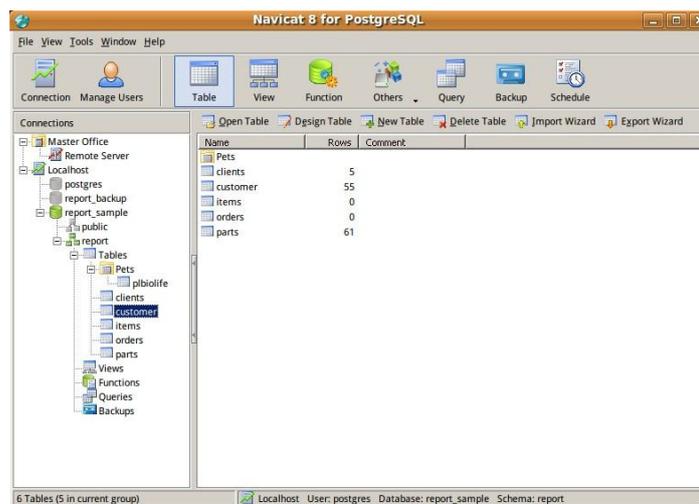


Ilustración 3. Página principal del Navicat.

1.3.3. EMS Data Import para PostgreSQL

EMS Data Import para PostgreSQL es una poderosa herramienta para importar datos, de manera rápida, desde archivos MS Excel, MS Access, DBF, TXT, CSV, XML hacia tablas de PostgreSQL. Provee parámetros configurables de importación, incluyendo formatos de datos de origen para todos los campos y formatos de datos de destino para los campos seleccionados, opciones de confirmación de ejecución (commit), número de registros a

saltar, etc. Data Import para PostgreSQL incluye un asistente que permite configurar todas las opciones de importación para diferentes archivos de manera visual y un utilitario de líneas de comandos para ejecutar la importación en un sólo "clic". (2007)

Entre sus características principales se pueden citar su facilidad de importar en los 10 formatos más comunes (MS Excel 97-2007, MS Access, XML, DBF, TXT, CSV, MS Word 2007, formato OpenDocument y HTML). Importa datos hacia una o varias tablas o vistas desde diferentes bases de datos. Cuenta con varios modos de importación: Insert All (insertar todos), Insert New (insertar nuevo), Insert (insertar) o Update (actualizar), entre otros. Soporte UNICODE. Parámetros configurables de importación para cada archivo de origen. Opción para salvar todos los parámetros de importación configurados en la sesión actual del asistente. Utilitario de líneas de comandos para importar datos utilizando el archivo de configuración. Poderoso módulo visual de opciones. Interfaz tipo asistente amigable para el usuario. (2007)

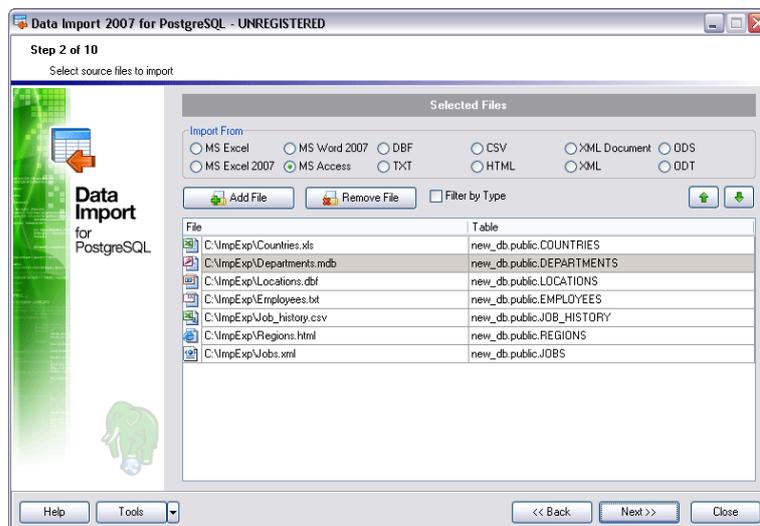


Ilustración 4. Página principal del EMS Data Import para PostgreSQL.

1.3.4. Sypex

Sypex es una aplicación open source para hacer respaldos de bases de datos MySQL. Se trata de una completa herramienta que además permite importar y exportar base de datos, crear bases de datos nuevas, verificarlas y hacerles mantenimiento entre otras opciones.

Además de las completas funciones que Sypex ofrece, se destaca por lo fácil que es de instalarla; sólo hay que subir los archivos al servidor y conectarse usando el usuario y contraseña del servidor de bases de datos. (2010)



Ilustración 5. Función Exportar de la herramienta de respaldo "Sypex".

1.4. Herramientas y tecnologías

1.4.1. Sistema operativo: GNU/Linux

Linux: es un sistema operativo (SO), de libre distribución UNIX para computadoras, servidores y estaciones de trabajo. Es muy eficiente, multitarea, multiusuario, multiplataforma y multiprocesador en las plataformas Intel corre en modo protegido. Protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria.

Además tiene en cuenta las cuatro libertades del software libre donde la cuestión es la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y modificar el software: (2010)

- La libertad de usar el programa, con cualquier propósito.

- La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades. El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puede ayudar a otros.
- La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie. De igual forma que la libertad 1 el acceso al código fuente es un requisito previo.

1.4.2. Lenguaje de modelado: UML

UML es una técnica para la especificación de sistemas en todas sus fases. Es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Un modelo es expresado en un lenguaje de modelado. Un lenguaje de modelado consiste de vistas, diagramas, elementos de modelo los símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas. Es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. (Román Zamitiz, 2011)

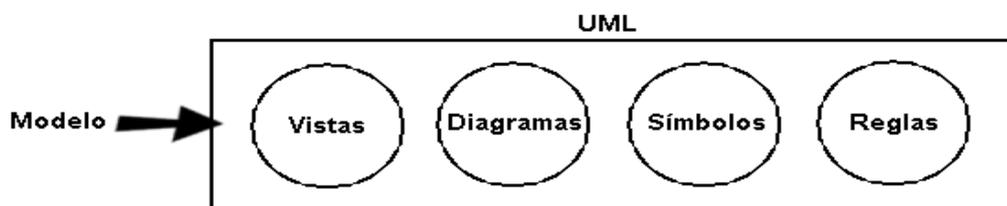


Ilustración 6. UML.

1.4.3. Metodología: RUP

La metodología **RUP** (Proceso Unificado de Rational, Rational Unified Process en inglés), cuyo fin es entregar un producto de software, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Forma disciplinada de asignar tareas y responsabilidades. RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Utiliza el lenguaje unificado de modelado (UML) para especificar, visualizar y documentar los diferentes aspectos relativos a un sistema de software bajo desarrollo.

RUP se divide en cuatro fases:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura.
- Construcción: En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- Transición: El objetivo es llegar a obtener el resultado (release) del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Además RUP define nueve disciplinas a realizar en cada fase del proyecto:

- Modelado del negocio.
- Análisis de requisitos.
- Análisis y diseño.
- Implementación.
- Prueba.
- Distribución.
- Gestión de configuración y cambios.
- Gestión del proyecto.
- Gestión del entorno.

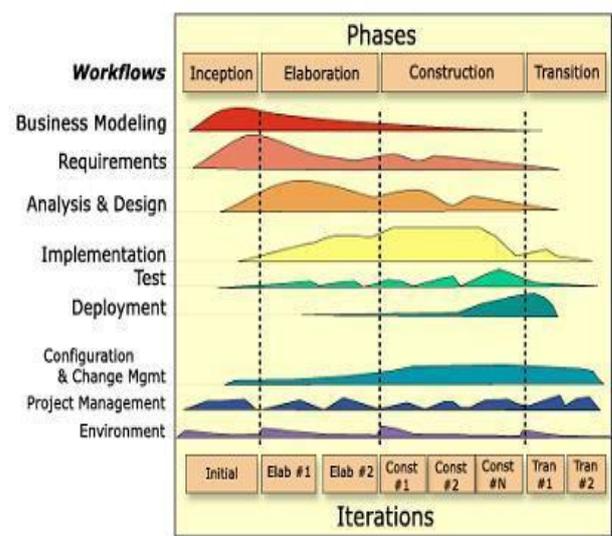


Ilustración 7. Metodología RUP.

1.4.4. Lenguaje de programación: Python

Python es un lenguaje de programación con una sintaxis muy limpia lo que favorece un código legible. Se trata de un lenguaje interpretado, pues se ejecuta utilizando un programa intermedio llamado intérprete y semi-compilado porque se generan bytecodes a partir del código fuente interpretado en lugar de compilar el código fuente a lenguaje máquina que

pueda comprender y ejecutar directamente una computadora. La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo los lenguajes interpretados son más flexibles y más portables. También posee tipado dinámico, ya que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.

Python también se puede ver ligado a QT llamándosele PyQt el cual es un binding (ligadura) de la biblioteca gráfica QT para el lenguaje de programación Python. La biblioteca está desarrollada por la firma británica Riverbank Computing y está disponible para Windows, GNU/Linux y Mac OS X bajo diferentes licencias. Ya que se escogió usar python por su robustez a la hora de desarrollar aplicaciones de software libres, el cual esta soportado bajo el paradigma de la programación orientada a objetos, ideal para la programación de herramientas sencillas como esta, pues tiene un código fácil de entender y sin complicaciones en la compilación.

1.4.5. Biblioteca gráfica: QT

QT es una biblioteca multiplataforma, creada para desarrollar interfaces gráficas de usuarios. Es desarrollada con el lenguaje de programación C++ de forma nativa pero existen módulos para otros lenguajes de programación como son C, Python (PyQT), Java (QT Jambi), Perl (PerlQT), Gambas (gb.qt), Ruby (QTRuby), PHP (PHP-QT), Mono (Qyoto), entre los más reconocidos.

Es una biblioteca totalmente orientada a objetos y es por ello que las Aplicaciones de Interfaces Programadas (API, Application Programming Interface) cuentan con diferentes métodos, además de soportar el uso de diferentes motores de Bases de Datos y el uso de archivos XML, además de otras estructuras de datos tradicionales.

Qt dispone de una amplia gama de herramientas que facilitan la creación de formularios, botones y ventanas de diálogo con el uso del ratón. Las aplicaciones creadas con Qt se ven

muy elegantes y se operan mejor que las aplicaciones nativas. Y una de las características más importantes de QT es que constituye una tecnología multiplataforma.

1.4.6. Biblioteca: XLRD

Para acceder a los datos almacenados en un documento XLS (Excel), python cuenta con una biblioteca llamada XLRD. Mediante la misma se pueden identificar las páginas del libro xls, así como la composición de las mismas. Para un correcto manejo de la información contenida en archivos de este tipo los datos son leídos celda a celda, respetándose el tipo de dato que define cada valor de forma tal que su posterior trabajo no afecte la integridad de este último.

1.4.7. ORM para python: SQLAlchemy

Un ORM simplifica el acceso a una base de datos por parte del programador convirtiendo toda sentencia sql a operaciones con objetos. Así por ejemplo añadir una fila a una tabla es tan fácil como crear un objeto.

Hay una gran cantidad de ORM para todos los gustos: El ORM de Django, Storm, Autum, SqlRelay, SQLAlchemy, Elixir. Son algunos de los más populares. El ORM SQLAlchemy con el cual se puede hacer el mapeo de tablas a objetos sin que el programador intervenga de manera profunda. Está diseñado para bases de datos robustas y ofrece buen desempeño en bases de datos de alta disponibilidad. SQLAlchemy no ve a las bases de datos sólo como una colección de tablas; las ve como motores de álgebra relacional. Su ORM permite mapear clases contra una base de datos de varias formas distintas.

1.4.8. Gestor de Base de Datos: PostgreSQL

PostgreSQL: es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su

desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (2009)

1.4.9. Herramienta Case: Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y de proyectos UML. Posee versiones multiplataforma, buena integración con IDEs, incluye localización en castellano, muy personalizable y soporta muchos más lenguajes de programación.

1.4.10. IDE: Eclipse

Eclipse: es una potente plataforma para el desarrollo de software, incluye la plataforma Eclipse, Java Development Tools, y entorno de desarrollo de complementos (plug-in), incluyendo la fuente y la documentación de usuario y programador. Eclipse comenzó como un IDE para Java, pero desde entonces ha crecido para mucho más y ahora Eclipse cubre proyectos tanto para lenguajes estáticos como dinámicos. Además es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama “Aplicaciones de Cliente Enriquecido”, opuesto a las aplicaciones “Cliente -Liviano” basada en navegadores.

1.4.11. Herramienta de diseño: Framework Qt

QtDesigner se suele utilizar en conjunción con KDevelop, aunque es multiplataforma. Sirve para generar ficheros .ui, que contienen la interfaz gráfica de un programa que utilice el toolkit Qt. Gracias a las utilidades de la biblioteca Qt, se puede generar código automáticamente a partir de los ficheros creados con QtDesigner.

1.5. Conclusiones

A partir del estudio realizado se puede afirmar que los sistemas existentes a pesar de tener funcionalidades que cumplen con muchas de las expectativas de los usuarios.

- No brindan la posibilidad de que el usuario configure como desea que sea su importación.
- Brindan acceso a la información contenida en la BD, acción que debería ser restringida por cuestiones de seguridad.
- No generan un fichero XML que guarde la configuración para ser usada como punto de partida de futuras importaciones.
- Y por último obtener estas herramientas constituye un gasto adicional, debido a la característica que tienen algunas de ellas de no ser desarrolladas bajo el paradigma del software libre.

En este capítulo también se presentaron las herramientas y tecnologías a utilizar para el desarrollo de una herramienta capaz de reunir todas las características con las que no cuentan las evaluadas anteriormente.

Capítulo 2. Descripción de la solución propuesta

2.1. Introducción

En el presente capítulo se dará un enfoque más específico de la herramienta a desarrollar. Se conocerán los requisitos funcionales y los no funcionales con los que debe cumplir la herramienta. Se describirá el sistema mediante los Casos de Usos. Se explicarán los estilos arquitectónicos y los patrones de diseño utilizados para enriquecer la arquitectura del sistema.

2.2. Funcionalidades del sistema

La herramienta DEBI (Importador de Datos para Inteligencia de Negocios) tiene la responsabilidad de resolver problemas de seguridad y rapidez en el trabajo con datos empresariales. Y para ello el objetivo del sistema es llevar a cabo la importación de datos contenidos en hojas de cálculos xls hacia una BD postgres, mediante la funcionalidad “Importar” que soluciona el conflicto de realizar estas tareas de forma manual, dando lugar a errores en el momento de realizar la copia.

Se soluciona la problemática de acceso pleno a la BD, pues la herramienta solo muestra la estructura de la BD a la que está conectada, no dando permiso para leer su información, ni realizar cambios en dicha información, evitando que se transformen los datos almacenados.

Además con esta herramienta se facilita el trabajo de los usuarios de otras sucursales que necesitan importar datos a la misma BD, con solo usar el fichero de configuración XML que se genera cuando el administrador que usa la herramienta “Guarda la Configuración” de la importación.

2.3. Requisitos del sistema

De las muchas definiciones que existen para requisitos, a continuación se presenta la definición más acertada y la más usada para nosotros.

Requisito es:

- Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.
- Una representación documentada de una condición o capacidad como en las definiciones anteriores.

Los requisitos pueden dividirse en funcionales y no funcionales.

2.3.1. Requisitos funcionales

Requisitos Funcionales: son capacidades o condiciones que el sistema debe cumplir. A continuación se presentan los mismos, así como su respectiva descripción.

- RF 1 Cargar XLS: La herramienta debe cumplir con la funcionalidad de cargar un documento xls del cual se tomará la información a procesar.
- RF 2 Visualizar Contenido del XLS: Se debe mostrar el contenido del documento xls que se cargue en la herramienta.
- RF 3 Conectar BD: La herramienta debe permitir conectarse a la BD a la que se quiera importar los datos del documento xls.
- RF 4 Visualizar Estructura de la BD: El sistema debe mostrar la estructura de la BD a la que se conecto el administrador.
- RF 5 Configurar Importación: El sistema debe permitir que un usuario marque los datos del xls que desea importar y el campo de la BD donde los guardará.
- RF 6 Administrar Configuración, el sistema se compromete a permitirle al usuario:
 - RF 6.1 Eliminar Configuración.
 - RF 6.2 Cargar Configuración.
 - RF 6.3 Guardar Configuración.
 - RF 6.4 Aplicar Configuración.
- RF 7 Importar Datos: La herramienta debe ser capaz de importar datos de un documento xls a una BD postgres.

2.3.2. Requisitos no funcionales

Requisitos No Funcionales: son propiedades o cualidades que el producto debe cumplir. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

Usabilidad

La interfaz de usuario debe:

- Ser de fácil operación y basada en ventanas.
- Tener capacidad de escalabilidad a la resolución de la pantalla que se disponga.
- Poseer ayuda en Línea, incluyendo documentación, manuales de ingeniería de confiabilidad.
- Combinaciones de acceso para el trabajo con la herramienta.
- Debe poseer una ventana donde se muestre la explicación paso a paso de las operaciones que debe realizar el usuario correspondiente a las diferentes funcionalidades.

Confiabilidad

- **Integridad de datos:** El sistema debe ser capaz de mantener la calidad del dato de manera que garantice su integridad durante su aplicación, procesamiento y almacenamiento en la BD.
- **Tiempo de recuperación ante fallas:** la herramienta debe restablecer sus funciones 30 segundos después de haber ocurrido la falla. No se desea perder la operatividad por períodos prolongados. El tiempo máximo que esperarán los usuarios para recuperación es de 5 minutos.

Desempeño

- **Garantizar procesos y transacción masiva de datos:** los procesos y transacción masiva de datos deben estar definidos hacia y desde todas las estaciones de trabajo.
- **Acceso a la BD:** la BD debe permitir ser accedida, modificada, actualizada independientemente del número de datos que maneje. La frecuencia máxima de acceso a los datos se estima en períodos semanales. La frecuencia de uso mínima: cada hora todos los días.
- **Tiempos requeridos para realizar consultas a la BD:** Los tiempos de consulta a la BD deben responder a las funciones requeridas por los usuarios en lapsos de espera que permitan cumplir con la funcionalidad solicitada. El tiempo máximo de respuesta esperado por los usuarios es de 30 segundos.

- **Concurrencia de usuarios:** El sistema debe soportar mínimo diez (10) usuarios concurrentes usando todas las funcionalidades del sistema.

Soportabilidad

- **Soporte tecnológico y metodológico:** El sistema debe poseer soporte metodológico y tecnológico para lograr su correcto funcionamiento.
- **Instalación,** configuración y arranque del sistema de fácil instalación, configuración y puesta en marcha.
- **Arquitectura abierta y distribuida modular,** de capacidad escalable y tecnología actualizable de acuerdo a las necesidades operacionales y tendencias tecnológicas de las aplicaciones y componentes que se ejecutan o interactúan con el sistema.
- **Programación orientada a objetos,** la programación del sistema debe estar orientada a objetos.
- **Integración con otras aplicaciones,** el sistema debe proveer todas las facilidades de integración hacia otras aplicaciones.

Restricciones de diseño (implementación)

- **Empleo del mismo conjunto de herramientas en el desarrollo del sistema:** Se debe emplear el mismo conjunto de tecnología para el desarrollo de la aplicación, así como para futuros desarrollos. De manera tal que garantice la escalabilidad y compatibilidad entre los diferentes desarrollos.
- **Rendimiento y alta disponibilidad:** la herramienta debe poseer características de rendimiento y alta disponibilidad de forma que se garantice su operatividad de forma segura, efectiva y confiable.
- **La herramienta debe ser multiplataforma:** la herramienta debe ejecutarse en diversas plataformas de hardware y software, es decir, debe ser portable a varios sistemas operativos como Windows XP y Linux.
- **Licencia bajo código abierto:** la herramienta debe cumplir con los lineamientos necesarios para la producción de software libre y la comunidad de desarrollo y soporte.

2.4. Descripción del sistema propuesto

2.4.1. Descripción de actores

Actor	Objetivo
Usuario	El usuario es la persona con autorización para hacer uso de la herramienta para llevar a cabo la importación de datos de una entidad.

Tabla 1. Descripción de los actores.

2.4.2. Diagrama de casos de usos del sistema

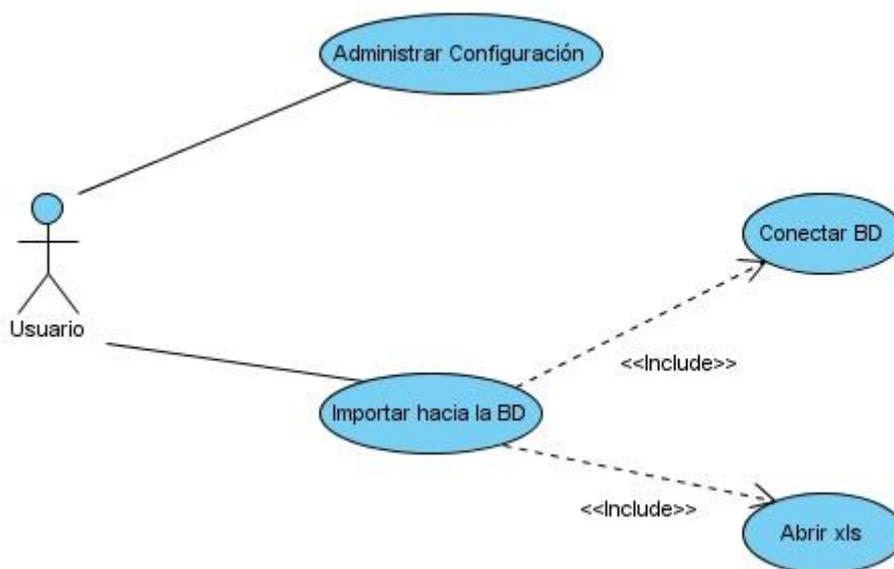


Ilustración 8. Diagrama de CUS.

2.4.3. Descripción de los casos de usos del sistema

A continuación se describirán los CU identificados para el desarrollo de la herramienta: “Abrir xls”, “Conectar BD”, “Importar hacia la BD” y “Administrar Configuración”.

CU 1. Cargar xls.

Objetivo	El objetivo de este CU es cargar en la herramienta el documento xls del que se va a tomar información.
Actores	Usuario.

Resumen	Este CU carga el documento que contiene los datos a procesar.	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	No procede.	
Postcondiciones	No procede.	
Flujo de eventos		
Flujo básico: Cargar XLS		
	Actor	Sistema
1.	Presiona clic en la opción de menú o el botón "Cargar XLS".	
2.		Muestra una ventana para que el actor especifique la dirección donde se encuentra el documento xls.
3.	Especifica la dirección donde se encuentra almacenado el documento xls.	
4.	Presiona clic en el botón "Abrir".	
5.		Cierra la ventana y visualiza el contenido del documento.
Flujos alternos		
El usuario cancela la carga del xls		
	Actor	Sistema
2.	En el paso 4 del flujo básico presiona clic en el botón "Cancelar" de la ventana donde debe especificar la dirección del documento.	
3.		Cierra la ventana y cancela la carga del xls.
Flujos alternos		
En la dirección especificada no hay un xls		

	Actor	Sistema
1.	En el paso 3 del flujo básico especifica una ruta donde no hay ficheros xls.	
2.		Solo visualiza las carpetas y los archivos xls.
3.	Específica una dirección donde hay una archivo xls.	
3.	Presiona clic en el botón "Abrir".	
4.		Cierra la ventana y visualiza el contenido del xls.
Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.
Requisitos no funcionales	No procede.	
Asuntos pendientes	No procede.	

Tabla 2. Descripción del CU "Cargar xls".

CU 2. Conectar BD

Objetivo	El objetivo de este CU es que el usuario que trabaja en la herramienta se conecte a la BD en la que desea almacenar la información.
Actores	Usuario.
Resumen	Este CU permite al usuario conectarse a la BD en la que desea almacenar los datos.
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	No procede.
Postcondiciones	No procede.

Flujo de eventos		
Flujo básico: Conectar a BD		
	Actor	Sistema
1.	Presiona clic en la opción de menú o el botón "Conectar BD".	
2.		Muestra una ventana de autenticación solicitando los siguientes datos: <ul style="list-style-type: none"> - Tipo de Servidor. - Nombre de la BD. - Dirección del Servidor. - Puerto. - Usuario. - Contraseña.
3.	Introduce los datos que el sistema solicita.	
4.	Presiona clic en el botón "Conectar".	
5.		Cierra la ventana y muestra la estructura de la BD a la que se conectó.
Flujos alternos		
Cancelar la Conexión		
	Actor	Sistema
1.	En el paso 4 del flujo básico presiona clic en el botón "Cancelar" de la ventana de autenticación.	
2.		Cierra la ventana y cancela la conexión.
Flujos alternos		
Entrada de datos incorrectos		
	Actor	Sistema

1.	En el paso 3 del flujo básico inserta datos de autenticación incorrectos.	
2.		Muestra un mensaje de error "Datos de autenticación incorrectos".
3.	Presiona clic en el botón "Aceptar".	
4.		Cierra la ventana y muestra la ventana de autenticación nuevamente.
Escenario "Ocultar/Visualizar BD"		
	Actor	Sistema
1.	Presiona clic en la opción de menú o el botón "Mostrar BD conectada".	
2.		Muestra la estructura de la BD a la que está conectado.
Flujos alternos		
Ocultar BD conectada		
	Actor	Sistema
1.	En el paso 1 del flujo básico presiona clic en la opción de menú o el botón "Ocultar BD conectada".	
2.		Oculto la estructura de la BD a la que está conectado.
Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.
Requisitos no funcionales	No procede.	
Asuntos pendientes	No procede.	

Tabla 3. Descripción del CU “Conectar BD”.

CU 3. Importar hacia la BD

Objetivo	El objetivo de este CU es llevar a cabo la importación que el actor necesita para almacenar los datos que tiene contenidos en una hoja de cálculos xls a las tablas de la BD en que trabaja.	
Actores	Usuario.	
Resumen	Este CU importa los datos del documento xls a la BD.	
Complejidad	Alta.	
Prioridad	Critico.	
Precondiciones	Se debe haber iniciado los CU “Conectar BD” y “Cargar xls”.	
Postcondiciones	No procede.	
Flujo de eventos		
Flujo básico Importar		
	Actor	Sistema
1.	Presiona clic en la columna o la celda que desea importar a la BD.	
2.		Marca de azul lo seleccionado por el actor.
3.	Presiona clic en la columna que desea que se guarden los datos en la BD.	
4.		Marca de azul el campo marcado por el actor.
5.	Presiona clic en el botón o la opción de menú “Marcar para importar”.	
6.		Guarda temporalmente la configuración creada por el actor.
7.	Repite los pasos anteriores cuantas veces desee, cuantas celdas disponibles hallan en el documento xls.	

8.	Presiona clic en el botón o la opción de menú "Importar".	
9.		Muestra un mensaje "Se importó exitosamente".
10.		Importa los datos del documento xls a la BD, según lo especificado por el actor.
11.		Se ejecuta la sección 1 Guardar Configuración.
Sección 1: "Guardar Configuración"		
	Actor	Sistema
1.	Presiona clic en el botón o la opción de menú "Guardar Configuración".	
2.		Muestra una ventana para indicar la ruta donde se guardará la configuración.
3.	Especifica la ruta donde desea guardar la configuración.	
4.	Presiona clic el botón "Aceptar".	
5.		Cierra la ventana y guarda la configuración de la importación generando un fichero XML.
Flujos alternos		
1.	En el paso 4 de la sección 1, presiona el botón "Cancelar".	
2.		Cierra la ventana y mantiene la interfaz.
Relaciones	CU Incluidos	Conectar BD. Abrir xls.
	CU Extendidos	No procede.

Requisitos no funcionales	No procede.
Asuntos pendientes	No procede.

Tabla 4. Descripción del CUS “Importar hacia la BD”.

CU 4. Administrar Configuración

Objetivo	El objetivo de este CU es que el usuario pueda Aplicar, Abrir, Guardar y Eliminar una Configuración.	
Actores	Usuario.	
Resumen	Este CU agrupa las funcionalidades: “Aplicar Configuración”, “Cargar Configuración”, “Eliminar Configuración” y “Guardar Configuración”.	
Complejidad	Media.	
Prioridad	Opcional.	
Sección 1: “Cargar Configuración”		
Precondiciones	Debe existir al menos una configuración.	
Postcondiciones	No procede.	
Flujo de eventos		
Flujo básico Cargar Configuración		
	Actor	Sistema
1.	Presiona clic en el botón o la opción de menú “Abrir Configuración”.	
2.		Muestra una ventana para que el actor especifique la dirección donde está la configuración guardada.
3.	Especifica la dirección donde está almacenada la configuración y presiona clic en el botón “Abrir”.	
4.		Cierra la ventana y visualiza el contenido de la configuración.

Flujos alternos		
Cancela la carga		
	Actor	Sistema
1.	En el paso 3 del flujo básico presiona clic en el botón "Cancelar"	
2.		Cierra la ventana.
Sección 2: "Aplicar Configuración"		
Flujo básico El actor aplica la configuración		
	Actor	Sistema
1.	Presiona clic en el botón o la opción de menú "Aplicar Configuración".	
2.		Aplica la configuración.
3.		Marca de azul los datos que se importaran.
4.		Muestra un mensaje "los datos sombreados son los que se importaran a la BD".
5.	Presiona clic en el botón "Aceptar".	
6.		Cierra la ventana.
Sección 3: "Guardar Configuración"		
	Actor	Sistema
1.	Presiona clic en el botón o la opción de menú "Guardar Configuración".	
2.		Muestra una ventana para indicar la ruta donde se guardará la configuración.
3.	Especifica la ruta donde desea guardar la configuración.	

4.	Presiona clic en el botón “Aceptar”.	
5.		Cierra la ventana y guarda la configuración de la importación generando un fichero XML.
Flujos alternos		
1.	En el paso 4 del flujo básico de eventos, presiona clic en el botón “Cancelar”.	
2.		Cierra la ventana y mantiene la interfaz.
Sección 4: “Eliminar Configuración”		
Flujo básico Nombre del flujo básico		
	Actor	Sistema
1.	Presiona clic en el botón o la opción de menú “Eliminar Configuración”.	
2.		Limpia la configuración.
Relaciones	CU Incluidos	No procede.
	CU Extendidos	No procede.
Requisitos no funcionales	No procede.	
Asuntos pendientes	No procede.	

Tabla 5. Descripción del CUS “Administrar Configuración”.

2.4.4. Diagramas de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes en un momento dado y ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos. A continuación se muestran los diagramas de secuencia para cada CU identificados anteriormente:

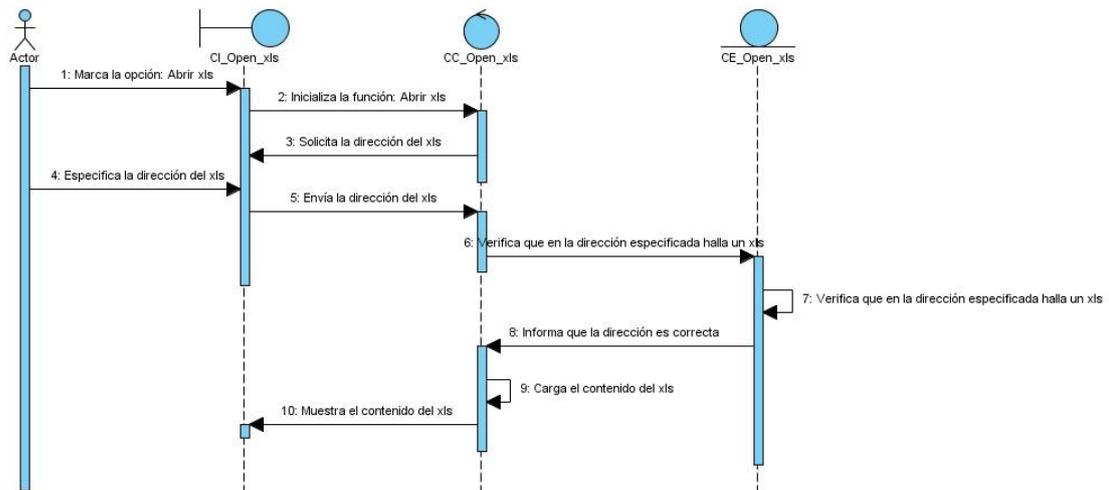


Ilustración 9. Diagrama de secuencia del CU “Cargar xls”.

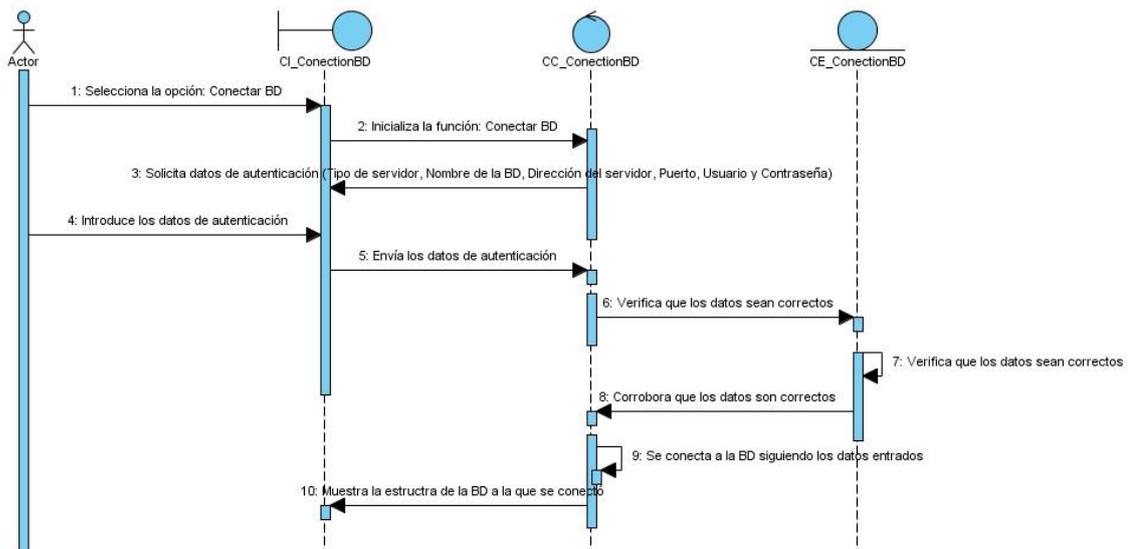


Ilustración 10. Diagrama de secuencia del CU “Conectar BD”.

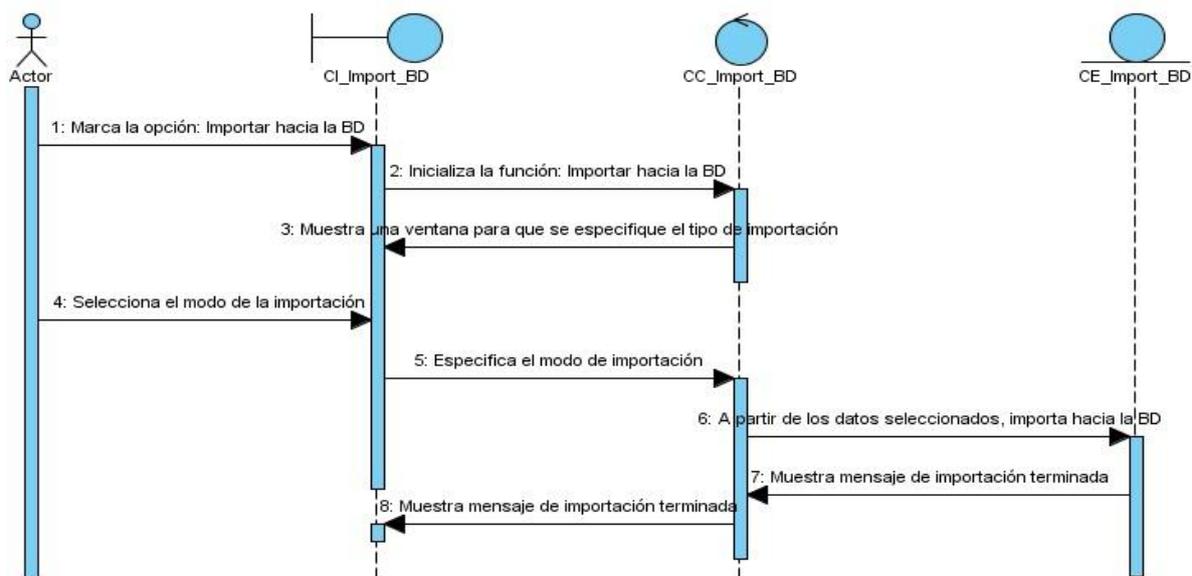


Ilustración 11. Diagrama de secuencia del CU “Importar hacia la BD”.

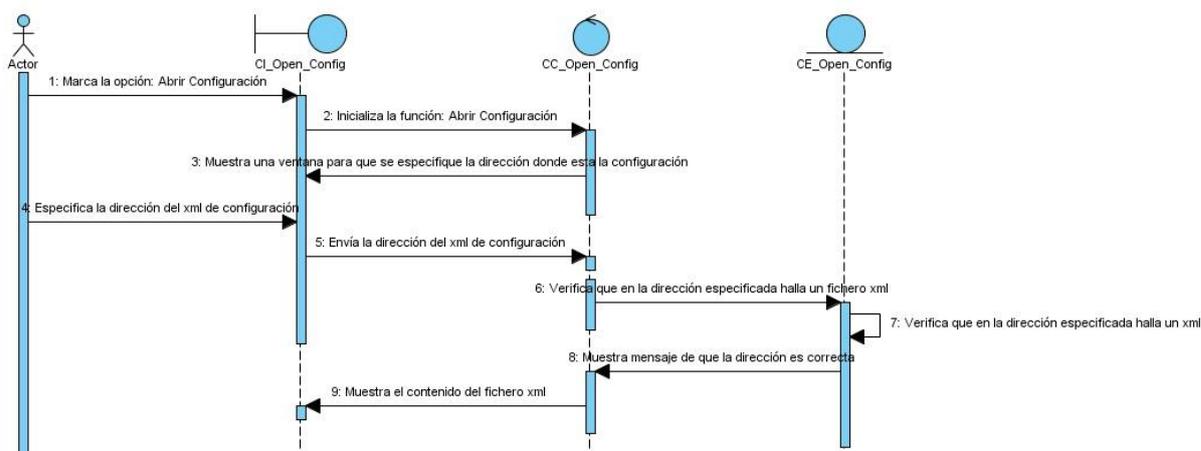


Ilustración 12. Diagrama de secuencia del CU “Administrar Configuración”, sesión: “Abrir Configuración”.

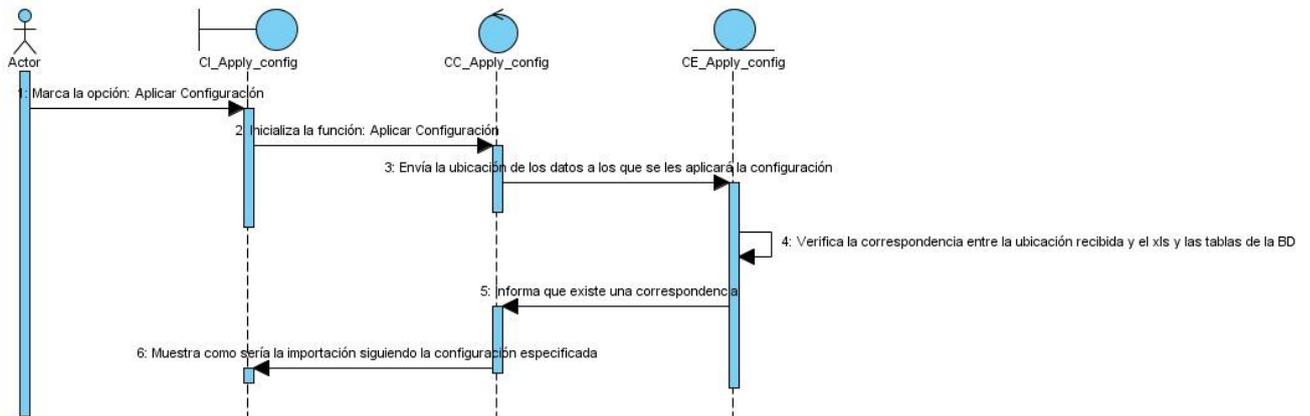


Ilustración 13. Diagrama de secuencia del CU “Administrar Configuración”, sesión: “Aplicar Configuración”.

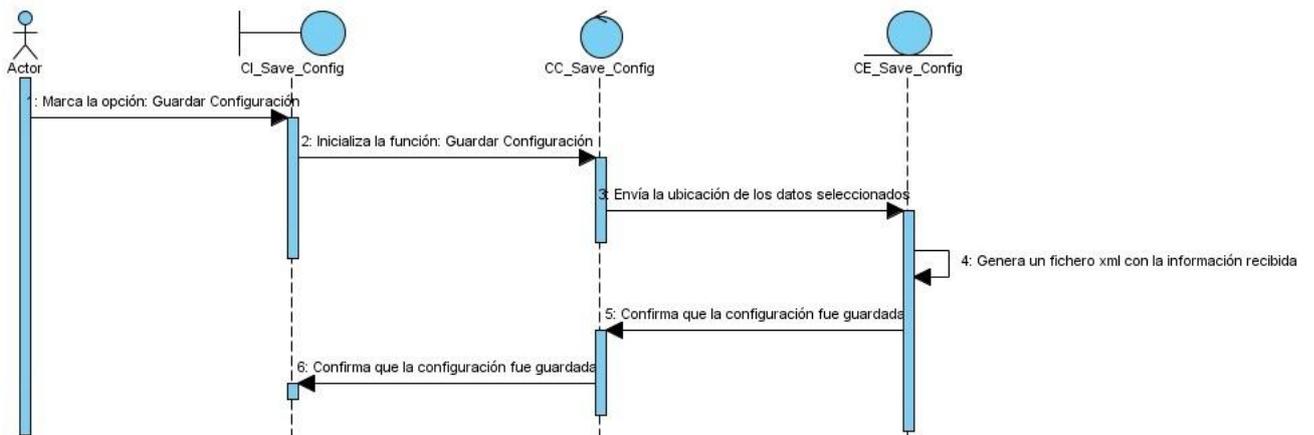


Ilustración 14. Diagrama de secuencia del CU “Administrar Configuración”, sesión: “Guardar Configuración”.

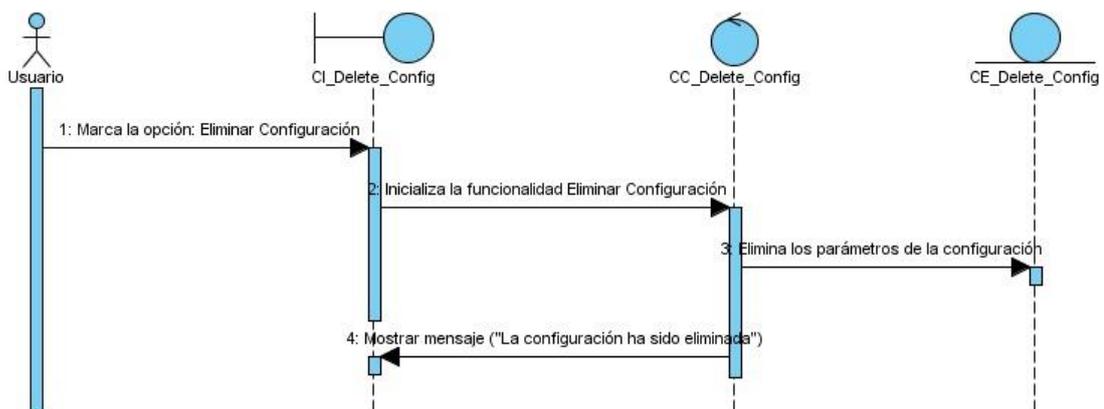


Ilustración 15. Diagrama de secuencia del CU “Administrar Configuración”, sesión: “Eliminar Configuración”.

2.5. Estilos arquitectónicos

Los estilos arquitectónicos forman parte del grupo de herramientas más importantes con las que cuenta el arquitecto para darle forma a la arquitectura que usará en una herramienta que desarrolle.

Un estilo de arquitectura viene definido por un conjunto de componentes, un conjunto de conexiones entre dichos componentes y un conjunto de restricciones sobre cómo se comunican dos componentes cualesquiera conectados. Los estilos de arquitectura se organizan en torno al aspecto de la aplicación sobre el que se centran. Los principales aspectos son: comunicaciones, despliegue, dominio, interacción y estructura.

Lo más usual en una arquitectura es que se base en varios estilos para tomar de ellos solo sus ventajas según el aspecto del sistema en el que se centre cada estilo. Los estilos utilizados en el desarrollo de la herramienta que se desarrolla son: la arquitectura en tres capas y el modelo-vista-controlador.

Arquitectura en Capas

Para el desarrollo de la herramienta se empleó la arquitectura en capas especialmente la de tres niveles creando tres capas independientes (presentación, lógica de negocio y acceso a datos) donde los cambios en una de ellas no afectan el funcionamiento del resto de las capas.

La arquitectura de tres niveles se divide en tres capas distintas:

- Capa de presentación (interfaz de usuario) en esta capa existe una interacción con el usuario, presenta los datos y recibe las entradas.
- Capa de aplicación/negocio (lógica de aplicación) responsable de las tareas propias de la aplicación concreta implementa la lógica de la aplicación y aplica las reglas de negocio sobre los datos y las entradas de usuario.
- Capa de datos (almacenamiento y acceso a datos) responsable de la gestión y almacenamiento permanente de los datos.

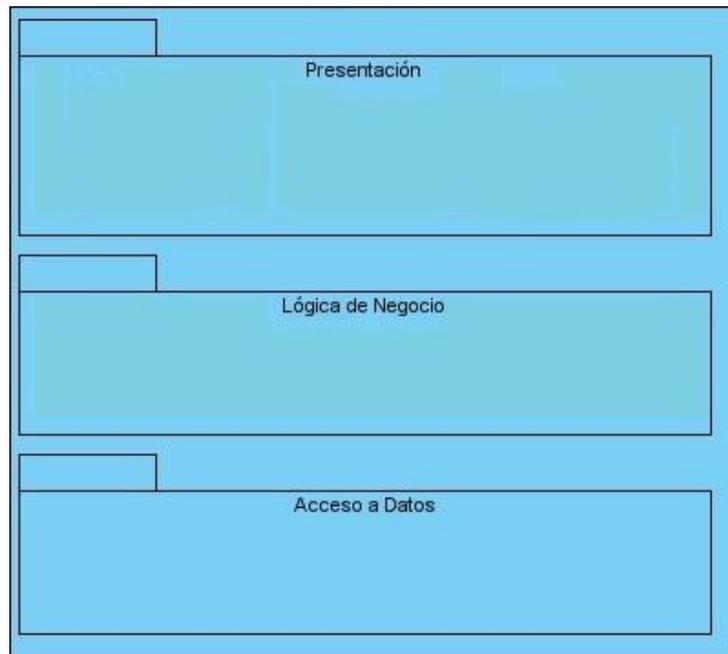


Ilustración 16. Arquitectura en tres capas.

Modelo-Vista-Controlador

El patrón modelo-vista-controlador es un patrón de diseño de arquitectura muy empleado en la actualidad, se utilizó pues al aplicarlo se logró que la aplicación facilitara la reusabilidad y la comprensión durante y después del desarrollo del sistema ya que es posible modificar cualquiera de estos tres componentes sin que esto afecte el funcionamiento del resto.

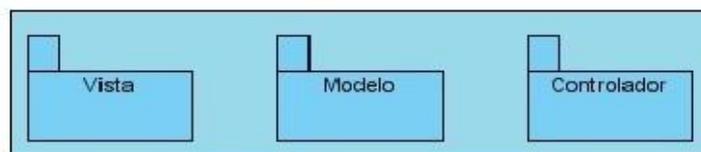


Ilustración 17. Modelo-Vista-Controlador.

- El modelo es la representación de la información que maneja la aplicación, administra los datos dándoles una estructura.
- La vista es la representación del modelo en forma gráfica disponible para la interacción con el usuario, muestra la estructura que le dió el modelo a los datos.

- El controlador maneja y responde a las solicitudes del usuario, como su nombre indica controla las acciones que se pueden realizar sobre los elementos que forman parte del modelo.

De la combinación de los estilos antes descritos resultó una arquitectura que toma lo mejor de cada estilo la cual permitirá guiar el desarrollo de la herramienta.

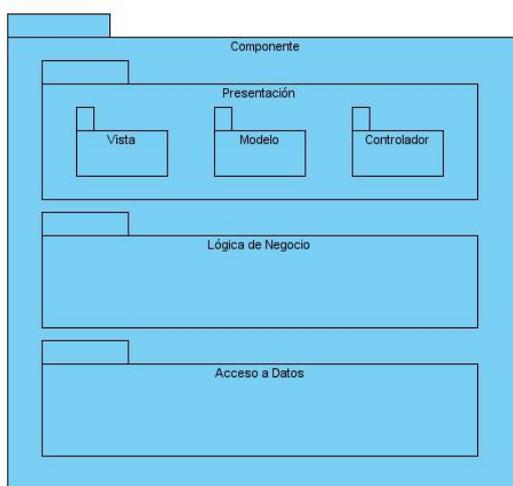


Ilustración 18. Diagrama de la arquitectura.

2.6. Patrones de diseño

Un patrón de diseño es un mecanismo de la ingeniería de software que brinda robustez y flexibilidad a un conjunto de clases. Constituye una solución predeterminada a un problema de diseño común para las aplicaciones de software.

Para el desarrollo de la herramienta de importación se hizo uso del patrón Singleton para la conexión a la BD ya que se garantizó con él, mantener una única instancia de la conexión donde los parámetros siempre son los mismos, lo único que cambia es el contenido de las consultas.

El Singleton es uno de los patrones más sencillos de los patrones de GOF, además de ser uno de los más conocidos y utilizados. En general su objetivo es garantizar que una clase sólo tenga una instancia y proporciona un punto de acceso global a ésta instancia; esto resuelve el

problema de que cuando varios clientes distintos precisen referenciar a un mismo elemento y quieran asegurarse de que no haya más de una instancia de ese elemento y para eso es que la solución que da Singleton es crear una única instancia de esa clase.

Aplicabilidad

Usar cuando:

- Deba haber exactamente una instancia de una clase y ésta deba ser accesible a los clientes desde un punto de acceso conocido.
- La única instancia debería ser extensible mediante herencia y los clientes deberían ser capaces de utilizar una instancia extendida sin modificar su código.

Consecuencias

- Acceso controlado a la única instancia. Puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia.
- Espacio de nombres reducido. El patrón Singleton es una mejora sobre las variables globales.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- Permite un número variable de instancias. El patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase Singleton.
- Más flexible que las operaciones de clase (static en C#, Shared en VB .NET).

2.7. Modelo de diseño

En el diseño de un software se modela el sistema para lograr la forma en que pueda cumplir con los requisitos que se especifican en los primeros pasos de su construcción. Un modelo de diseño es un modelo de objetos que describen la ejecución de los CU concentrándose en el impacto que causan los requisitos funcionales y los no funcionales en el sistema. En el modelo de los CU son desarrollados por clases de diseño y sus objetos.

Uno de los objetivos principales del diseño es: crear una entrada idónea y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales, interfaces y clases.

Para organizar las clases de diseño se usó la estructura de paquetes donde las clases se clasificaron según el papel que juegan cada una de ellas en la herramienta, por lo que fueron creados cinco paquetes (Presentation, Business, domain, dao y útil).

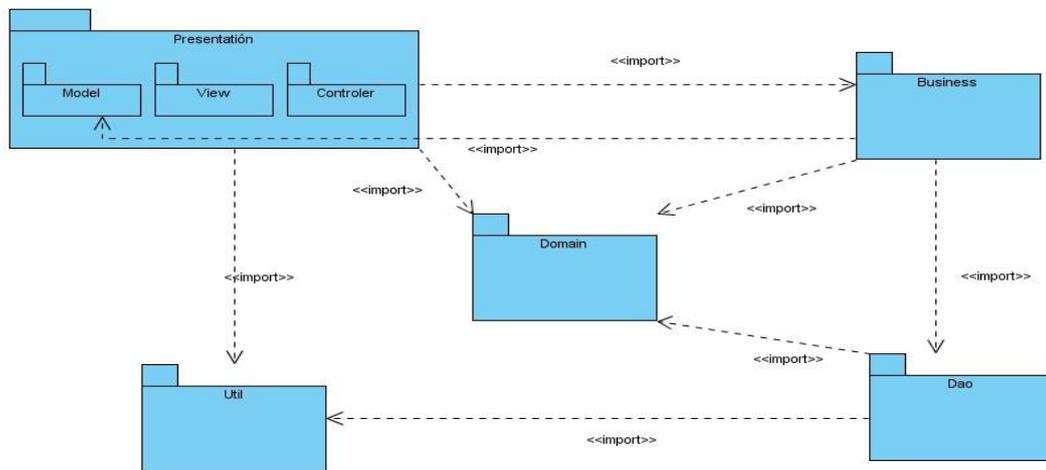


Ilustración 19. Diagrama de paquetes.

2.8. Conclusiones

Con lo mostrado en este capítulo se conoció como funciona la herramienta, describiendo su funcionamiento. Se presentaron los requisitos funcionales y los no funcionales y se describieron los CU identificados. Se mostraron los estilos arquitectónicos empleados para enriquecer la arquitectura del sistema. Se explicaron los patrones de diseño empleados; y finalmente se mostró el modelo de diseño a seguir. Por lo que se puede apreciar lo importante que es todo el estudio realizado para la construcción de un buen software.

Capítulo 3. Implementación y Prueba

3.1. Introducción

En el presente capítulo se hablará un poco más de la implementación de la herramienta en proceso por lo que se mostrará el diagrama de clases por cada paquete en los que está dividido el código así como la descripción de las principales clases de cada paquete. Se explicará el estándar de codificación y documentación que garantizarán la legibilidad del código para futuros cambios. Mediante un diagrama de despliegue se dará a conocer la distribución física de hardware utilizado. Se muestra una visión más amplia de la herramienta funcionando para que el usuario tenga una idea de cómo es su funcionamiento. Y finalmente se hará un análisis de los resultados obtenidos de los casos de prueba que ayudarán a corregir los errores que se encuentren en la herramienta.

3.2. Diagrama de clases por paquetes

El diseño de clases encapsula los principales conceptos del sistema en clases que constituyen la base para la futura implementación. A continuación se muestran los diagramas de clases por paquetes:

Paquete Presentation

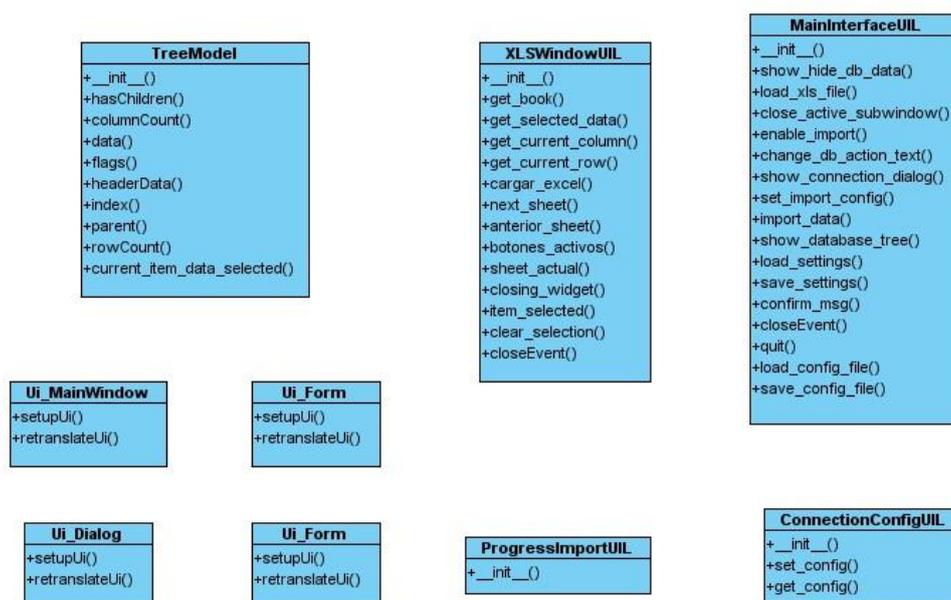


Ilustración 20. Diagrama de clases del paquete "presentation".

Paquete Business

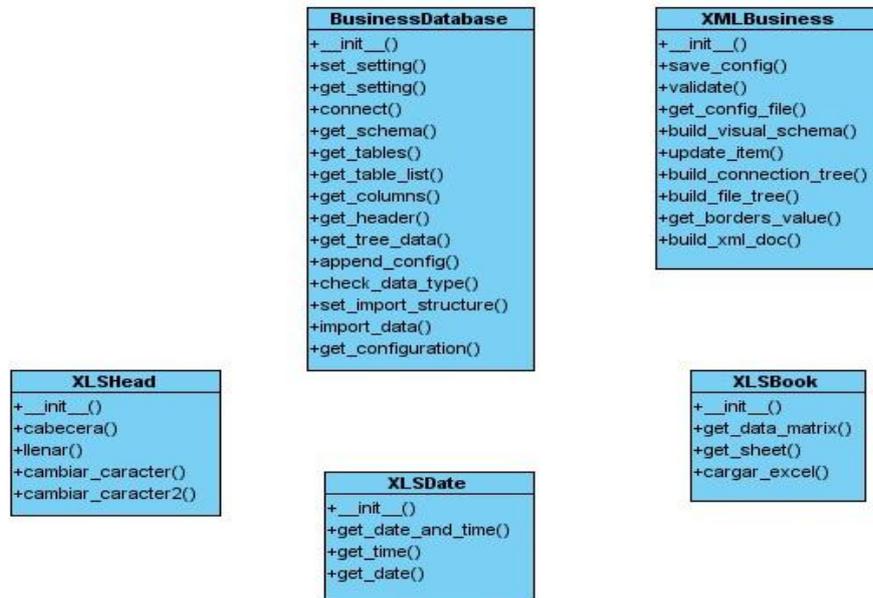


Ilustración 21. Diagrama de clases del paquete “dusiness”.

Paquete Domain

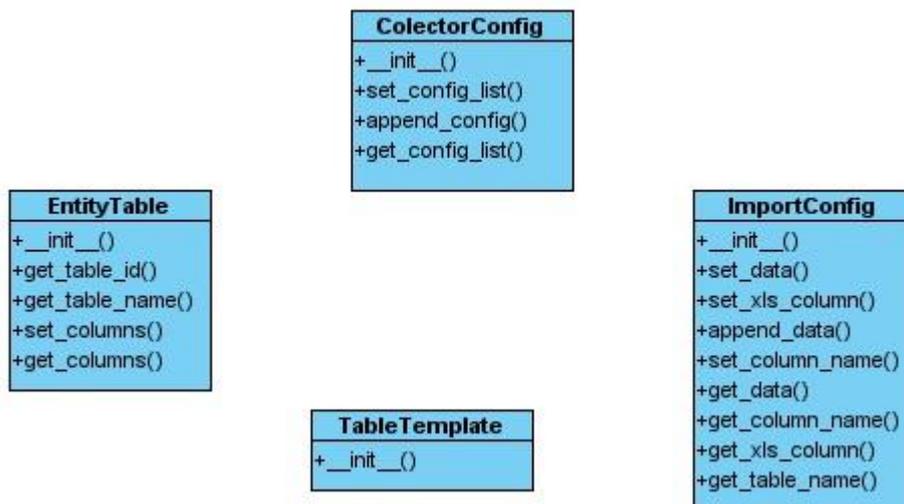


Ilustración 22. Diagrama de clases del paquete “domain”.

Paquete Dao

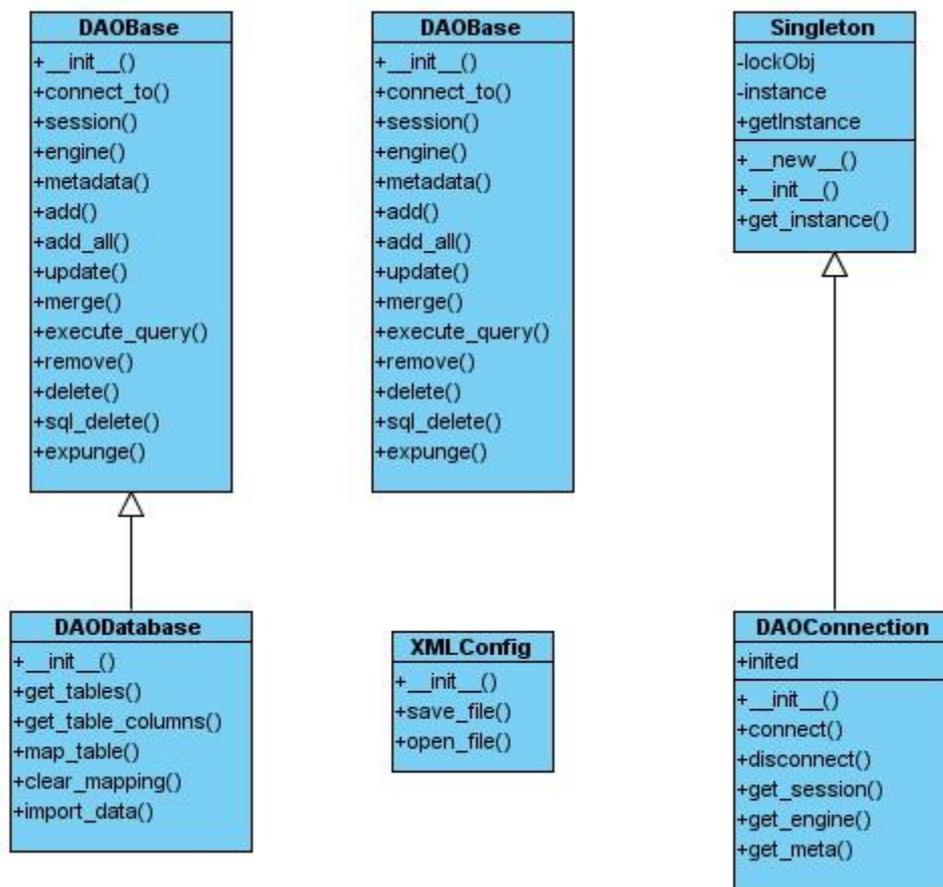


Ilustración 23. Diagrama de clases del paquete “dao”.

Paquete Util

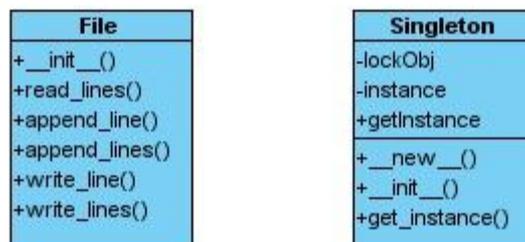


Ilustración 24. Diagrama de clases del paquete “util”.

3.3. Descripción de las clases por paquete

Descripción de las clases del paquete “Presentation”

Nombre del paquete: Presentation	
Para cada paquete:	
Nombre:	Treeltem
Descripción:	Estructura de datos para la representación de datos en forma de árbol.
Nombre:	ConfigTreeModel
Descripción:	Clase modelo para la visualización de estructuras de datos en forma de árbol.
Nombre:	GenericTreeModel
Descripción:	Esta clase genera una instancia de la clase TreeModel.
Nombre:	TreeModel
Descripción:	Esta clase define el modelo de datos usado para la visualización en forma de árbol de los resultados de la búsqueda de objeto técnicos.
Nombre:	ConnectionConfigUIL
Descripción:	Lógica de interfaz para establecer la configuración de la conexión a la BD.

Ilustración 25. Descripción de las clases del paquete “presentation”.

Descripción de las clases del paquete “Business”

Nombre del paquete: Business	
Para cada paquete:	
Nombre:	BusinessDatabase
Descripción:	Maneja todo lo referente a la BD, es decir, se realizan todas las consultas a la BD.

Nombre:	XLSBook
Descripción:	Esta clase es para ofrecer más funcionalidades después de cargado un libro como obtener la fecha de la forma dd/mm/aaaa, obtener en una matriz una hoja de cálculo, todo a partir de un libro de Excel.
Nombre:	XLSDate
Descripción:	Da el formato dd/mm/aaaa a la fecha y hh:mm:ss a la hora.
Nombre:	XMLBusiness
Descripción:	Guarda los parámetros de configuración que se necesitan para generar el fichero xml.

Ilustración 26. Descripción del paquete "business".

Descripción de las clases del paquete "Domain"

Nombre del paquete: Domain	
Para cada paquete:	
Nombre:	ColectorConfig
Descripción:	Recoge temporalmente los datos de importación hasta que se decida si se guarda la configuración o se importan esos datos.
Nombre:	ImportConfig
Descripción:	Con esta clase se toma el dato, la columna y la fila de las celdas seleccionadas en el archivo xls.
Nombre:	TableTemplate
Descripción:	Es para crear objetos de las tablas de la BD.

Ilustración 27. Descripción del paquete "domain".

Descripción de las clases del paquete "Dao"

Nombre del paquete: Dao	
Para cada paquete:	

Nombre:	DAOBase
Descripción:	Clase base de acceso a datos.
Nombre:	DAOConnection
Descripción:	Clase que maneja la conexión a la BD en la aplicación.
Nombre:	DAODatabase
Descripción:	Clase donde se realizan las consultas de la BD.
Nombre:	XMLConfig
Descripción:	Salva, carga y visualiza el contenido de un fichero xml.

Ilustración 28. Descripción del paquete “dao”.

Descripción de las clases del paquete “Util”

Nombre del paquete: Util	
Para cada paquete:	
Nombre:	Singleton
Descripción:	Implementación del Patrón de diseño: SINGLETON, dando soporte a multihilos.

Ilustración 29. Descripción del paquete “util”.

3.4. Diagrama de componente

Un componente es una pieza de software que describe un conjunto de servicios que son usados sólo a través de interfaces bien definidas. El diagrama describe las interacciones existentes entre los principales componentes que integran la solución, así como sus dependencias más significativas.

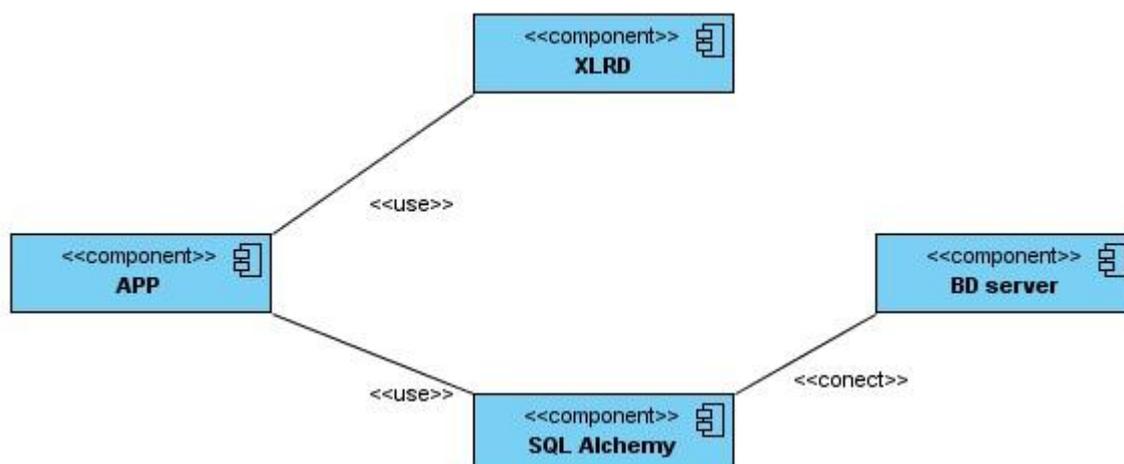


Ilustración 30. Diagrama de componentes.

3.5. Estándar de Codificación

Un estándar de codificación es una tecnología, formato o método desarrollado, adoptado a través de proceso abierto de consenso, con la ventaja de facilitar que el código fuente de cualquier aplicación sea legible, se le pueda dar mantenimiento, sea interoperable y pueda ser distribuido. Los estándares abiertos son capaces de soportar y administrar en forma más sencilla y menos costosa la creciente complejidad de los sistemas, constituyendo una especie de "garantía universal" de compatibilidad, ajena a cualquier proveedor de la industria por sí mismo.

La estandarización del código fuente repercute directamente en la comprensión del programador con respecto a un sistema de software, la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Por lo que, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y posteriormente se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener.

Para la correcta implementación de la herramienta de importación se hizo uso del estándar de codificación pep-8 propuesta por Guido Van Rossun creador del lenguaje python.

3.6. Estándar de Documentación

Como elemento de apoyo a las técnicas de codificación en la facilitación de la inteligibilidad del código se encuentra la documentación interna de un software. Documentar el código de un programa es añadir suficiente información como para explicar lo que hace, de forma que no sólo los ordenadores sepan qué hacer, sino que además los humanos entiendan qué están haciendo y por qué. En el mundo del software todo programa que tenga éxito será modificado en el futuro, bien por el programador original o por otro programador que le sustituya, por lo que es importante que el programa se entienda para poder repararlo y modificarlo en el menor tiempo posible.

Con el objetivo de apoyar la documentación previa a la implementación de la herramienta en proceso se utilizó Doxygen para la generación automática de la documentación interna.

3.7. Diagrama de Despliegue

El diagrama de despliegue es un diagrama utilizado para representar el hardware empleado en la implementación del sistema y sus relaciones. Muestra la forma en que serán desplegados los componentes del sistema y las capacidades de la red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema desarrollado. Para el despliegue de la herramienta se necesita una PC cliente para instalar la aplicación, un servidor de datos para almacenar los datos a procesar y una impresora para el caso en que el usuario desee imprimir alguna información.

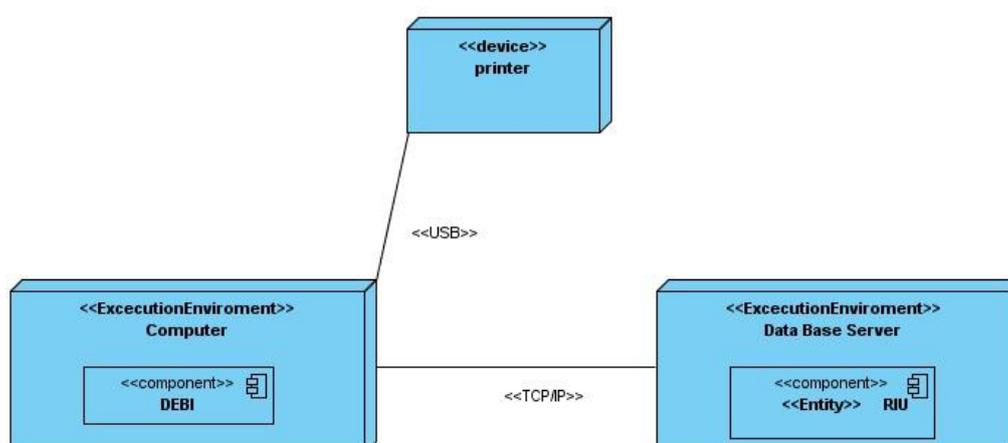


Ilustración 31. Diagrama de Despliegue.

3.8. Visión de la Aplicación

Al ejecutarse la aplicación se muestra el splash inicial de la aplicación (ilustración 32) precedido por la ventana principal de la misma (ilustración 33), dicha ventana cuenta con una barra de título, un menú donde aparecen una serie de opciones de menú y una barra de tareas. Además en la zona lateral derecha reserva un espacio para mostrar la estructura de la BD a la que el usuario decida conectarse así como en el lateral izquierdo se reserva el espacio para visualizar los datos del archivo xls que cargue el usuario (ilustración 34).



Ilustración 32. Splash inicial de la aplicación.

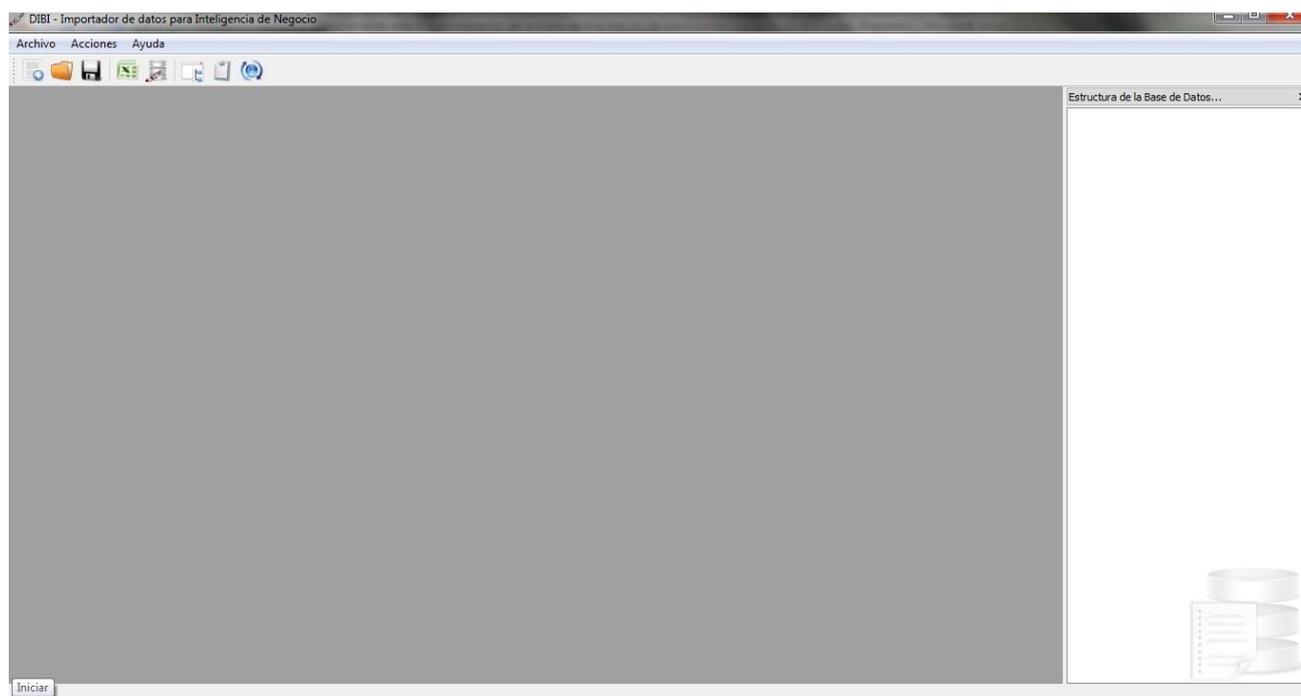


Ilustración 33. Ventana principal de la aplicación.

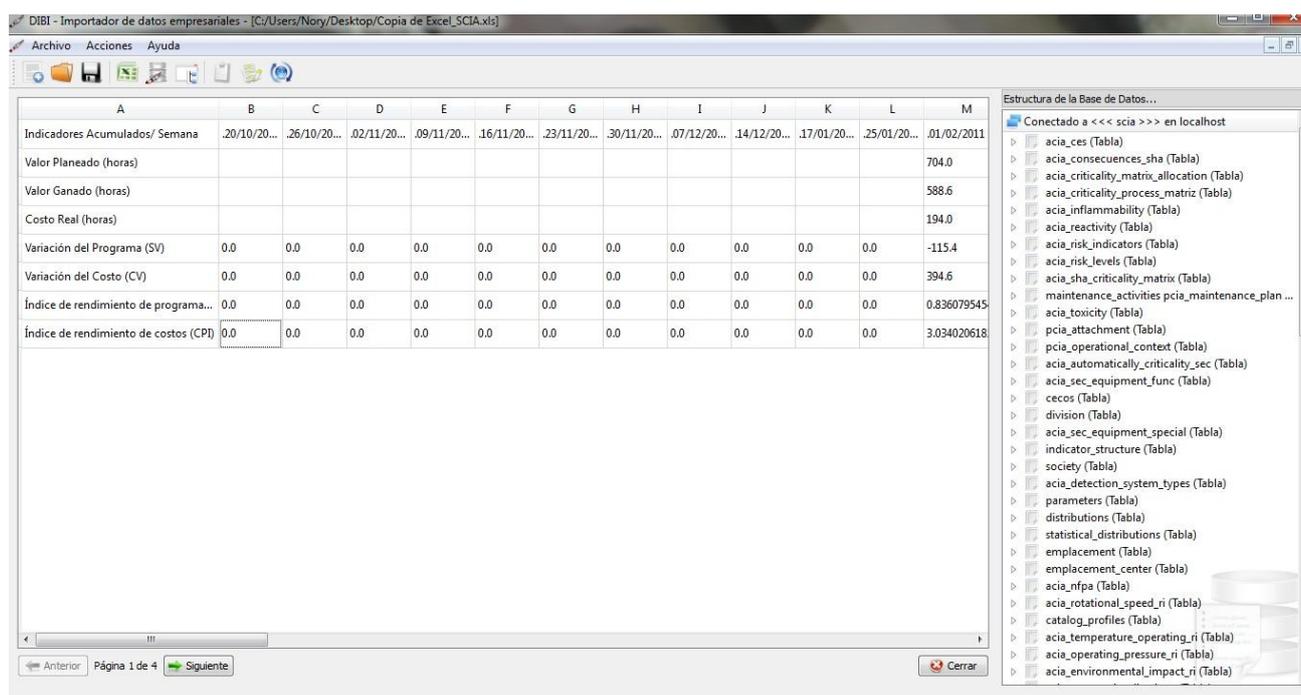


Ilustración 34. Espacio reservado para mostrar la estructura de la BD y el contenido del xls.

Para la conexión de la BD se muestra una ventana (ilustración 35) en la que se solicitan los datos de autenticación (Tipo de Servidor, Nombre de la BD, Dirección del Servidor, Puerto, Usuario y Contraseña).

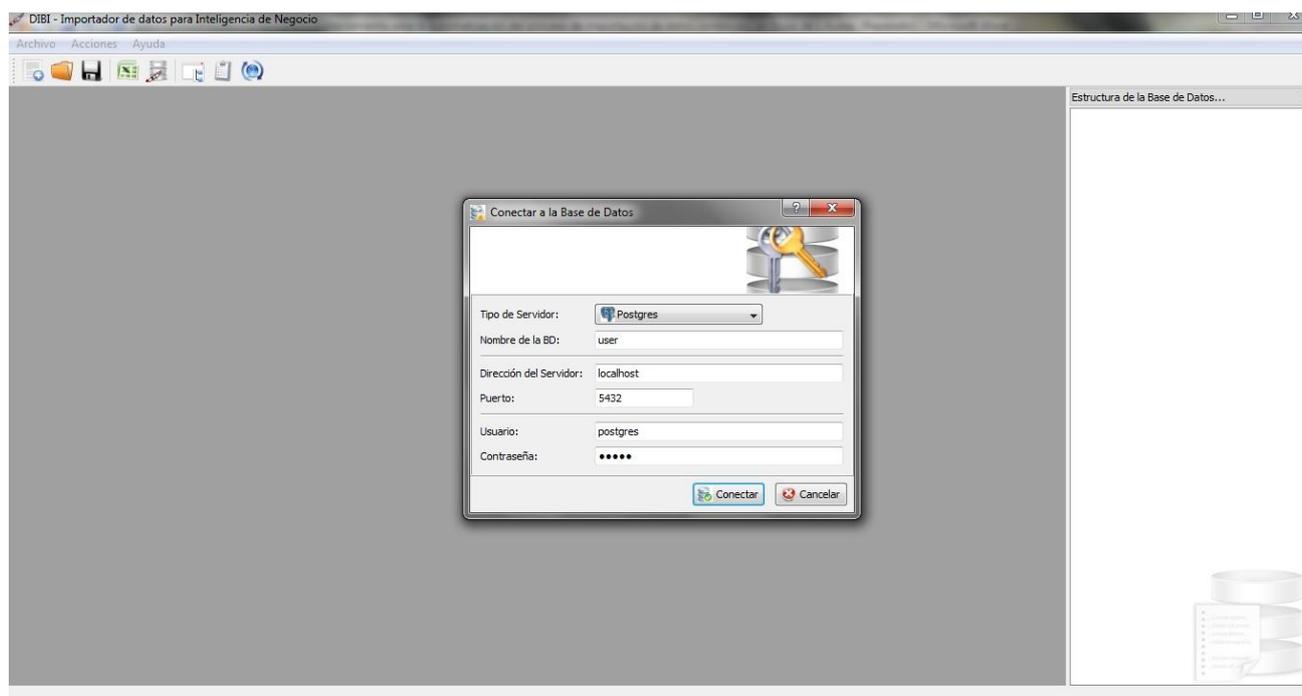


Ilustración 35. Ventana de autenticación de la BD.

3.9. Pruebas

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

Los diferentes niveles de pruebas son:

- Nivel de unidad: Está enfocada al código fuente de los componentes y verifica todos los flujos de control. Este tipo de prueba pasa primero por la revisión del programador.
- Nivel de integración: Prueba los componentes combinados para ejecutar un caso de uso. Tiene como objetivo descubrir errores en las especificaciones de las interfaces de las clases.
- Nivel de sistema: Prueba el software funcionando como un todo. Aceptable cuando el software se encuentra en fase de construcción.
- Nivel de aceptación: Es la prueba final antes del despliegue del sistema. Generalmente las realizan los usuarios finales y es llamada prueba piloto.

Objetivo de las pruebas

Cuando se le aplican pruebas de calidad a un software, los principales objetivos que se persiguen son los siguientes:

- Brindar un mayor nivel de confiabilidad en los productos que se van generando.
- Detectar fallas o errores.
- Aumentar la calidad del producto final.

El objetivo principal es diseñar pruebas que sistemáticamente reflejen diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. Si las pruebas se llevan a cabo con éxito se descubrirán errores en el software, dándole mayor fiabilidad.

Como uno de los pasos finales en la construcción de un software se encuentra las pruebas de las que anteriormente se hizo mención. En la línea Gestión Industrial se tiene problemas para el procesamiento de la información del proyecto SCIA el cual está integrado a la línea. Para validar la herramienta, las pruebas se hicieron usando los planes de mantenimiento generados por metodología PCIA del SCIA, tomando de los resultados de las pruebas, los resultados positivos, los resultados negativos y la cantidad de intentos, para ser analizados y dar una valoración de cuan eficaz es la herramienta desarrollada para resolver el problema que nos ocupa mediante la tabla que se muestra a continuación:

No.	Caso de Prueba	Resultados Positivos	Resultados Negativos	%	Cantidad de Intentos
1	Abrir xls	100	0	100	100
2	Conectar BD	95	5	95	100
3	Importar hacia la BD	45	5	90	50
4	Administrar Configuración	28	2	93.3	30

Tabla 6. Resultados de las pruebas.

Luego de hacer un análisis de los resultados de las pruebas se muestra mediante gráficas el por ciento que representan los intentos fallidos y los exitosos con respecto a la cantidad de pruebas realizadas para las funcionalidades básicas de la herramienta “Importar” y “Administrar Configuración” las cuales incluyen el resto de la funcionalidades.



Ilustración 36. Por ciento que representan de las pruebas para "Importar".



Ilustración 37. Por ciento que representan las pruebas para "Administrar Configuración".

3.10. Conclusiones

En el transcurso de este capítulo se mostró una breve descripción de los estándares de codificación y documentación que se utilizó para favorecer la legibilidad, el mantenimiento e interoperabilidad del software para futuros cambios. Utilizando un diagrama de despliegue se visualizó la distribución del hardware necesario para el despliegue de la herramienta. Como parte de los pasos finales se analizaron los resultados arrojados por las pruebas de sistema realizadas para mitigar los errores encontrados en la herramienta.

Conclusiones Generales

Con la investigación realizada se logró validar una nueva herramienta que agilizará el trabajo en proyectos relacionados con procesos industriales como SCIA, la misma es capaz de importar datos desde una hoja de cálculo xls hacia una base de datos postgres generando un fichero XML con la configuración de la importación si el operador que la trabaja lo desea.

Lográndose con la herramienta:

- Reducir los tiempos de trabajos dedicados a tareas de réplica de la información de las hojas de cálculos.
- Mayor seguridad en la información almacenada en la base de datos del proyecto que la utiliza.
- Evitar la ocurrencia de errores en la réplica de los datos de forma manual.

Recomendaciones

Para una versión superior de la herramienta desarrollada se recomienda:

- Investigar e implementar funcionalidades que faciliten la importación para tablas que incluyen relaciones entre ellas.
- Extender las importaciones a otros tipos de SGBD además de postgres, así como a otros formatos de HC.

Bibliografía

2011. [En línea] 03 de ene de 2011. [Citado el: 5 de mar de 2011.] <http://www.stack.nl/~dimitri/doxygen/>.

2007. [En línea] 02 de sept de 2007. [Citado el: 20 de mar de 2011.] http://chaghi.com.ar/blog/2007/09/02/sqlalchemy_orm_que_sabe_algebra_relacional/.

2010. Ant 2e6 + 1. [En línea] 28 de nov de 2010. [Citado el: 20 de mar de 2011.] <http://ant2e6.site11.com/Articulos/Programacion/Patrones-de-diseno.htm>.

Automatización. [En línea] [Citado el: 18 de mar de 2011.] http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/Automatizacion/Automatizacion.htm.

Di Mare, Adolfo. 2007. [En línea] 2007. [Citado el: 5 de mar de 2011.] <http://www.di-mare.com/adolfo/p/Doxygen.htm>.

2010. effiejay's blog. [En línea] 31 de may de 2010. [Citado el: 20 de mar de 2011.] <http://effiejayx.wordpress.com/2010/05/31/programando-en-linux-1-sqlalchemy-y-elixir/>.

2009. Free Download Manager. [En línea] 27 de abr de 2009. [Citado el: 23 de nov de 2010.] http://www.freedownloadmanager.org/es/downloads/Gratis_Navicat_PostgreSQL_Lite_%28Mac_OS_X%29_-_%23_1_PostgreSQL_herramienta_GUI_para_Mac_OS_X_%28compatible_para_Leopard!%29._59074_p/.

2007. Free Download Manager. [En línea] 23 de ener de 2007. [Citado el: 23 de nov de 2010.] http://www.freedownloadmanager.org/es/downloads/Importaci%C3%B3n_de_Datos_de_SME_2005_para_PostgreSQL_35682_p/.

García, Joaquín. 2005. Ingeniería de Software. [En línea] 27 de may de 2005. [Citado el: 2 de feb de 2011.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

2010. GNU Operating System. [En línea] 9 de dic de 2010. [Citado el: 4 de feb de 2011.] <http://www.gnu.org/philosophy/free-sw.es.html>.

Guido van Rossum, Barry Warsaw. 2007. Guía de estilo del código Python. [En línea] 10 de agos de 2007. [Citado el: 4 de mar de 2011.] <http://mundogeek.net/traducciones/guia-estilo-python.htm>.

Introducción a python. [En línea] [Citado el: 19 de mar de 2011.] <http://users.servicios.retecal.es/tjavier/python/intpython/x92.html>.

Maldonado, Daniel M. y Maldonado, Daniel M. 2010. EL Código K. [En línea] 1 de ene de 2010. [Citado el: 04 de mar de 2011.] <http://www.elcodigok.com.ar/2010/01/pep8-para-que-un-programa-en-python-sea-leido-como-un-poema/>.

2011. msdn. [En línea] 2011. [Citado el: 5 de mar de 2011.] <http://msdn.microsoft.com/es-es/library/bb972272.aspx>.

2011. msdn. *Archivos de Configuración*. [En línea] 2011. [Citado el: 19 de mar de 2011.] <http://msdn.microsoft.com/es-es/library/1xtk877y%28v=vs.80%29.aspx>.

Nicolás Tedeschi. 2011. msdn. [En línea] 2011. [Citado el: 2 de feb de 2011.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

Pérez Verde-Ramo, Ariel. 2008. Automatización Industrial. [En línea] 02 de nov de 2008. [Citado el: 18 de mar de 2011.] <http://automatizacion.bligoo.com/content/view/311549/En-que-consiste-automatizar-un-proceso-industrial.html#content-top>.

2008. Riboso Matic. [En línea] 1 de feb de 2008. [Citado el: 2010 de nov de 2010.] <http://www.ribosomatic.com/articulos/exportar-documentos-de-excel-a-mysql/>.

Román Zamitiz, Ing. Carlos Alberto. 2011. EL LENGUAJE UNIFICADO DE MODELADO (UML). [En línea] 2 de feb de 2011. [Citado el: 5 de feb de 2011.] <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>.

Share Ware. [En línea] [Citado el: marz de 02 de 2011.] <http://softbb.org/es/691742.aspx> .

2009. Sobre Postgres. [En línea] 2009. [Citado el: 22 de nov de 2010.] http://www.postgresql-es.org/sobre_postgresql.

2010. Tus Maya Blog. [En línea] 23 de ene de 2010. [Citado el: 19 de mar de 2011.] <http://tuxmaya.wordpress.com/2010/01/23/programacion-en-c-y-qt-introduccion/>.

2009. Ubicuos.com. [En línea] 28 de oct de 2009. [Citado el: 20 de mar de 2011.] <http://www.ubicuos.com/2009/10/28/python-pygtk-sqlalchemy-en-sql/>.

2010. Zona Linux. [En línea] 17 de feb de 2010. [Citado el: 17 de ene de 2011.] <http://zonalinux.com.ar/sypex-aplicacion-open-source-para-hacer-respaldos-de-bases-de-datos-mysql/>.

2010. Zona WebMaster. [En línea] 27 de agos de 2010. [Citado el: 08 de mar de 2011.] <http://www.webmix.ws/webmaster/las-pruebas-en-aplicaciones-web-conceptos-y-clasificacion-fuente-las-pruebas-en-aplicaciones-web-conceptos-y-clasificacion/>.

Glosario de Términos

Archivo de configuración: Los archivos de configuración son archivos XML que se pueden cambiar según sea necesario. Los programadores pueden usar archivos de configuración para cambiar valores sin volver a compilar las aplicaciones. Los administradores pueden usar los archivos de configuración para establecer directivas que afectan al modo en que se ejecutan las aplicaciones en sus equipos.

Automatización: La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.

Binding (ligadura): en el campo de la programación, un **binding** es una adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquél en el que ha sido escrita.

Lenguaje compilado: es un lenguaje de programación que suele necesitar de un compilador para ser ejecutado.

Lenguajes dinámicos: Un lenguaje dinámico, en pocas palabras es un lenguaje que en tiempo de ejecución realiza cosas que tradicionalmente se harían en tiempo de compilación. Entre estas cosas tenemos:

- Parseo (Análisis) de expresiones.
- Verificación de tipos de datos.
- Redefinición y alteración a los tipos.
- Emisión de bloques ejecutables.

Lenguajes estáticos: son los que proporcionan dinamismo a las aplicaciones web, son una arquitectura común entre las aplicaciones web.

Lenguaje interpretado: un lenguaje es interpretado si sus instrucciones se ejecutan secuencialmente a partir de código fuente.

Anexos

Casos de Prueba

CP 1: Abrir xls

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
Abrir xls	EC 1.1: Abrir xls correctamente.	Esta funcionalidad carga el documento que contiene los datos a procesar.	<p>El usuario presiona clic en la opción de menú o el botón "Cargar XLS".</p> <p>El sistema muestra una ventana para que el actor especifique la dirección dónde se encuentra el documento xls.</p> <p>El usuario especifica la dirección donde se encuentra almacenado el documento xls.</p> <p>El usuario presiona clic en el botón "Abrir".</p> <p>El sistema visualiza el contenido del documento que el actor abrió.</p>
	EC 1.2: Abrir xls. El usuario cancela la carga.	En este escenario el usuario ya no desea abrir el xls.	<p>El usuario presiona clic en el botón "Cancelar" de la ventana donde debe especificar la dirección del documento.</p> <p>El sistema cierra la ventana y cancela la carga del xls.</p>
	EC 1.3: Abrir xls. El archivo especificado no es válido.	Esta funcionalidad carga el documento que contiene los datos a procesar.	<p>El usuario especifica una dirección donde no hay ficheros xls.</p> <p>El sistema solo visualiza los</p>

			<p>archivos con formato xls.</p> <p>El usuario especifica una dirección correcta.</p>
--	--	--	---

CP 2: Conectar BD

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
Conectar BD	EC 1.1: Conectar BD exitosamente.	Este CUS permite al administrador conectarse a la BS en la que desea almacenar los datos.	<p>El usuario presiona clic en la opción de menú o el botón "Conectar BD".</p> <p>El sistema muestra una ventana de autenticación solicitando los siguientes datos:</p> <ul style="list-style-type: none"> - Tipo de Servidor. - Nombre de la BD. - Dirección del Servidor. - Puerto. - Usuario. - Contraseña. <p>Introduce los datos que el sistema solicita.</p> <p>El usuario presiona clic en el botón "Conectar".</p> <p>El sistema muestra la estructura de la BD a la que se conectó.</p>
	EC 1.2: Conectar BD. Se cancela la conexión.	En este escenario el usuario ya no desea conectarse a la BD.	<p>El usuario presiona clic en el botón "Cancelar" de la ventana de autenticación.</p> <p>El sistema cancela la conexión.</p>
	EC 1.3: Conectar BD. Alguno de los datos de autenticación es incorrecto.		El usuario inserta datos de autenticación incorrectos.

			<p>El sistema muestra un mensaje de error “Datos de autenticación incorrectos”.</p> <p>El usuario presiona clic en el botón “Aceptar”.</p> <p>El sistema permite al actor que introduzca los datos correctos y reanude la conexión.</p>
--	--	--	---

CP 3: Importar hacia la BD

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
Importar hacia la BD.	EC 1.1: Importar hacia la BD exitosamente.	Este CUS permite al administrador llevar a cabo la importación de los datos desde la HC xls hacia la BD a la que está conectado.	<p>El usuario presiona clic en la columna o la celda que desea importar a la BD.</p> <p>El sistema marca de azul el campo marcado por el actor.</p> <p>El usuario presiona clic en la columna de la tabla de la BD a la que desea importar los datos.</p> <p>El sistema marca de azul el campo marcado por el actor.</p> <p>El usuario presiona clic en el botón o la opción de menú “Marcar para importar”.</p> <p>El sistema guarda temporalmente la configuración creada por el actor.</p> <p>El sistema repite los pasos anteriores cuantas veces</p>

			<p>desea, cuantas celdas disponibles hallan en el doc.</p> <p>El usuario presiona clic en el botón o la opción de menú "Importar".</p> <p>El sistema importa los datos del doc xls a la BD, según lo especificado por el actor.</p> <p>El sistema manda un mensaje "Los datos han sido importados exitosamente".</p> <p>El usuario presiona clic en el botón "Aceptar".</p> <p>Inicializa la funcionalidad "Guardar Configuración".</p>
	<p>EC 1.2: Importar hacia la BD. Guardar la configuración.</p>	<p>Esta funcionalidad permite al usuario guardar la configuración de la importación para ser usada como punto de partida de futuras importaciones.</p>	<p>El usuario presiona clic en el botón o la opción de menú "Guardar Configuración".</p> <p>El sistema muestra una ventana para que el usuario especifique donde desea guardar el fichero de configuración.</p> <p>El sistema guarda un fichero XML con la configuración creado por el actor.</p>

CP 4: Administrar Configuración

Nombre de la	Escenarios de la	Descripción de la funcionalidad	Flujo central
--------------	------------------	---------------------------------	---------------

sección	sección		
Administrar Configuración	EC 1.1: Administrar Configuración. Abrir Configuración.	Este CUS agrupa las funcionalidades: Guardar, Cargar, Aplicar y Eliminar Configuración.	<p>El usuario presiona clic en el botón o la opción de menú "Abrir Configuración".</p> <p>El sistema muestra una ventana para que el actor especifique la dirección donde está la configuración guardada.</p> <p>El usuario especifica la dirección donde está almacenada la configuración.</p> <p>El usuario presiona clic en el botón "Abrir".</p> <p>El sistema visualiza el contenido del fichero de configuración.</p>
	EC 1.2: Administrar Configuración. Abrir Configuración. El actor especifica una dirección incorrecta.	Este CUS agrupa las funcionalidades: Guardar, Cargar, Aplicar y Eliminar Configuración.	<p>El sistema solo visualiza los archivos con formato xml.</p> <p>El usuario especifica una dirección correcta.</p>
Administrar Configuración	EC 1.3: Administrar Configuración. Aplicar Configuración.	Este CUS agrupa las funcionalidades: Guardar, Cargar, Aplicar y Eliminar Configuración.	<p>El usuario presiona clic en el botón "aplicar".</p> <p>El sistema muestra en el xls los datos que se van a importar sombreados.</p>
Administrar Configuración	EC 1.4: Administrar Configuración. Eliminar Configuración.	Este CUS agrupa las funcionalidades: Guardar, Cargar, Aplicar y Eliminar Configuración.	<p>El usuario presiona clic en el botón o la opción de menú "Eliminar Configuración".</p> <p>El sistema elimina los</p>

			parámetros de la configuración.
	EC 1.5: Administrar Configuración. Guardar Configuración.	Este CUS agrupa las funcionalidades: Guardar, Cargar, Aplicar y Eliminar Configuración.	<p>El usuario presiona clic en el botón o la opción de menú "Guardar Configuración".</p> <p>El sistema muestra una ventana para que el usuario especifique donde desea guardar el fichero de configuración.</p> <p>El sistema guarda un fichero XML con la configuración creado por el actor.</p>