

**UNIVERSIDAD CIENTIFICA DEL SUR**

**CARRERA DE INGENIERÍA DE SISTEMAS DE INFORMACIÓN Y GESTIÓN**

**CPE-2017-0**

**Asignatura :** Lenguajes de Programación

**Profesor :** Erick Gustavo Coronel Castillo

**Tema :** Manejo de archivos Excel: Importar y Exportar

**Alumnos :** Jesús Peñaranda H.

Josselyn Flores P.

Miraflores, 20 de febrero 2017

# INDICE

[RESUMEN 3](#_Toc468788068)

[ANTECEDENTES 4](#_Toc468788069)

[FUNDAMENTO TEORICO 5](#_Toc468788070)

[CASO DEMOSTRATIVO: CREAR UNA HOJA DE CALCULO CON POI 12](#_Toc468788077)

[CONCLUSIONES 14](#_Toc468788078)

[REFERENCIAS BIBLIOGRÁFICAS 15](#_Toc468788080)

# RESUMEN

La posición de la aplicación Microsoft Excel como origen y destino de datos va ganando cada vez más relevancia, las funcionalidades y el poder de manejar distintos orígenes de datos lo han convertido en una de las aplicaciones más utilizadas en todo tipo de ámbito e industria, este trabajo de investigación pretende mostrar las características más importantes, concentrando la atención en sus capacidades de importación / exportación de datos.

En escenarios donde necesitamos que nuestros programas guarden los datos de forma persistente para poderlos recuperar cuando volvamos a arrancar el programa de nuevo, nos encontramos con varias opciones para dar solución a esto: utilizar una base de datos, guardar la configuración o información en archivos de texto plano; sin embargo, en ocasiones, una base de datos es demasiado elaborada para los datos que debemos guardar, y un texto plano acaba siendo un auténtico desorden de información que no sabemos cómo encontrar. Para esto existe una solución a medio camino que permite mantener la información ordenada sin necesidad de construir una compleja base de datos: las hojas de cálculo o de Excel con ayuda de apache POI.

Por otro lado, usuarios más exigentes requieren aprovechar la información contenida en grandes hojas de Excel, o contenidas en bases de datos para obtener reportes o hacer cálculos estadísticos tomando información de diferentes fuentes, para esto Excel se integra con herramientas poderosas como Power BI, SQL Azure Database, entre otros.

# ANTECEDENTES

Microsoft Excel es una aplicación de hojas de cálculo que forma parte de la suite de oficina Microsoft Office, es utilizada en tareas financieras y contables, con fórmulas, gráficos y un lenguaje de programación.

Microsoft comercializó originalmente un programa para hojas de cálculo llamado Multiplan en 1982, que fue muy popular en los sistemas CP/M, pero en los sistemas MS-DOS perdió popularidad frente al Lotus 1-2-3. Microsoft publicó la primera versión de Excel para Mac en 1985, y la primera versión de Windows (numeradas 2-05 en línea con el Mac y con un paquete de tiempo de ejecución de entorno de Windows) en noviembre de 1987. Lotus fue lenta al llevar 1-2-3 para Windows y esto ayudó a Microsoft a alcanzar la posición de los principales desarrolladores de software para hoja de cálculo de PC, superando al también muy popular1 Quattro Pro de Borland. Por esta razón Microsoft se solidificó como un competidor válido y mostró su futuro de desarrollo como desarrollador de software GUI (interfaz gráfica de usuario). Microsoft impulsó su ventaja competitiva lanzando al mercado nuevas versiones de Excel, por lo general cada dos años. La versión actual tanto para la plataforma Windows y Mac OS X es el reciente Excel 2016.

Excel fue la primera hoja de cálculo que permitió al usuario definir la apariencia (las fuentes, atributos de carácter y celdas). También introdujo re computación inteligente de celdas, donde celdas dependientes de otra celda que han sido modificadas, se actualizan al instante (programas de hoja de cálculo anterior recalculaban la totalidad de los datos todo el tiempo o esperaban para un comando específico del usuario). Excel tiene una amplia capacidad gráfica, y permite a los usuarios realizar, entre otras muchas aplicaciones, listados usados en combinación de correspondencia.

Desde 1993, Excel ha incluido Visual Basic para Aplicaciones (VBA), un lenguaje de programación basado en Visual Basic, que añade la capacidad para automatizar tareas en Excel y para proporcionar funciones definidas por el usuario para su uso en las hojas de trabajo. VBA es una poderosa anexión a la aplicación que, en versiones posteriores, incluye un completo entorno de desarrollo integrado (IDE) conocido también como Editor de VBA. La grabación de macros puede producir código (VBA) para repetir las acciones del usuario, lo que permite la automatización de simples tareas. VBA permite la creación de formularios y controles en la hoja de trabajo para comunicarse con el usuario. Admite el uso del lenguaje (pero no la creación) de las DLL de ActiveX (COM); versiones posteriores añadieron soporte para los módulos de clase permitiendo el uso de técnicas de programación básicas orientadas a objetos

# FUNDAMENTO TEORICO

## Excel como hoja de cálculo

Una hoja de cálculo es un programa o aplicación informática que permite la manipulación sobre datos números dispuestos en tablas para la operación sobre cálculos complejos de contabilidad, finanzas y negocios

La hoja de cálculo es una aplicación de los paquetes de informática tradicionales que está programada para el manejo de datos numéricos y alfanuméricos con el propósito de obtener conclusiones informes de contabilidad. Las posibilidades de este tipo de aplicaciones son inmensas, ya que permite operar con cálculos complejos, fórmulas, funciones y elaborar gráficos de todo tipo.

Actualmente y en su forma más tradicional, las hojas de cálculo se emplean para hacer bases de datos numéricos, operaciones de cálculos entre celdas, informes y representaciones en gráfico de torta, barras y otros. Estas funciones no sólo son muy útiles para la administración y decisión a nivel ejecutivo, sino que también son fundamentales a la hora de presentar resultados y conclusiones laborales y de negocios a públicos y clientes.

Por su versatilidad y facilidad de uso a partir de un entrenamiento en el software, este tipo de programas permiten ahorrar mucho tiempo (de elaboración y resolución de cálculos extensos y complicados) y dinero (invertido en contadores y especialistas de cálculos y economía).

## Microsoft Excel: Integración en el Desarrollo de Software

Diversos fabricantes de Software desarrollan APIs que permiten a los programadores crear, modificar y visualizar los archivos de MS Office utilizando distintos lenguajes de programación. Usualmente son bibliotecas de código abierto, y su uso más frecuente es diseñar o modificar los archivos de Microsoft Office utilizando líneas de código. Contiene clases y métodos para decodificar los datos de entrada de usuario o un archivo en documentos de MS Office.

**APIs existentes para Microsoft Excel**

Los Cells Aspose para Java

Aspose para Java se desarrolla puramente con licencia de Java API de Excel y distribuidos por el vendedor Aspose. La última versión de esta API es 8.1.2, lanzado en julio de 2014. Se trata de una API para el diseño del componente de Excel que se puede leer, escribir y manipular hojas de cálculo. Los usos comunes de esta API son las siguientes:

• La presentación de informes de Excel, crear informes dinámicos de Excel

• De alta fidelidad de representación en Excel e impresión

• Importación y exportación de datos de hojas de cálculo Excel

• Generar, editar y convertir hojas.

JXL

JXL es un marco de terceros compatible con la automatización basada en los datos de los navegadores web (actualización automática de los datos en los navegadores web). Sin embargo, también se utiliza como una biblioteca común de soporte para la API JExcel porque tiene características básicas para crear, leer y escribir en hojas de cálculo. Las características básicas son las siguientes:

• Generar archivos de Excel

• Importar datos de libros y hojas de cálculo

• Obtener el número total de filas y columnas

JExcel

JExcel es una API puramente licencia proporcionada por Dev Team. Los programadores pueden leer fácilmente, escribir, mostrar y modificar los libros de Excel en ambos formatos .xls y .xlsx. Esta API puede ser fácilmente integrado con Java Swing y AWT. La última versión de esta API es JExcel-2.6.12, lanzado en 2009. Las principales características son las siguientes.

• Automatizar Excel de aplicaciones, libros de trabajo, hojas de cálculo, etc.

• Libros Embed de una aplicación Java Swing como componente Swing ordinaria

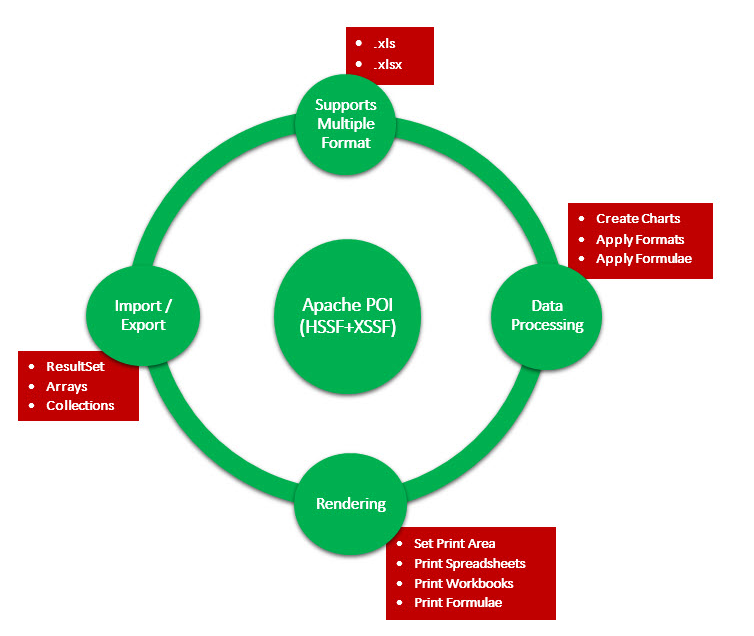
• Añadir detectores de eventos a los libros y hojas de cálculo

• Agregar controladores para manejar el comportamiento de eventos de libro

• Añadir compañeros nativos para desarrollar una funcionalidad personalizada

Apache POI

Apache POI es una biblioteca de código abierto 100% proporcionado por la Apache Software Foundation. La mayor parte de los pequeños y medianos desarrolladores de aplicaciones escala dependen en gran medida de Apache POI (HSSF + XSSF). Es compatible con todas las características básicas de las bibliotecas de Excel; Sin embargo, la representación y la extracción de texto son sus características principales.



## 

## Apache POI

Apache POI es una solución de Java Excel (para Excel 97-2008) y proporciona un API completo para virar otros formatos hacia el lado de babor OOXML y OLE2.

Los archivos OLE2 incluyen la mayoría de los archivos de Microsoft Office tales como XLS, doc., y PPT tan bien como formatos de archivo basados API de la serialización del MFC. El proyecto proporciona los APIs para el sistema de ficheros OLE2 (POIFS) y las características del documento OLE2 (HPSF).

El formato de OpenXML de la oficina es el formato de archivo basado los nuevos estándares de XML encontrado en Microsoft Office 2007 y 2008. Esto incluye XLSX, DOCX y PPTX. El proyecto proporciona un API bajo para apoyar a las convenciones de empaquetado abiertas que usan openxml4j.

Para cada uso de MS Office existe un módulo componente que intente proporcionar una API Java de alto nivel común para los formatos de documento de OLE2 y de OOXML. Esto se desarrolla más para los libros de trabajo de Excel (SS=HSSF+XSSF). El trabajo está progresando para los documentos de la palabra (HWPF+XWPF) y las presentaciones del PowerPoint (HSLF+XSLF).

El proyecto ha agregado recientemente la ayuda para la perspectiva (HSMF). Microsoft abrió las especificaciones en este formato en octubre de 2007. Hay también proyectos para Visio (HDGF) y el editor (HPBF).

Las aplicaciones del Apache POI API incluyen usos de la extracción del texto tales como Web spider, constructores del índice, y sistemas de gestión del contenido.

## POI – CLASES Y MÉTODOS

Algunas clases y métodos en la API de Apache POI que son críticos para trabajar en archivos Excel utilizando programas Java:

LIBRO DE TRABAJO:

Este es el súper-interfaz de todas las clases que crean o mantienen los libros de Excel. Pertenece al paquete org.apache.poi.ss.usermodel. Las dos clases que implementan esta interfaz son los siguientes:

• HSSFWorkbook: Esta clase tiene métodos para leer y escribir archivos de Microsoft Excel en formato .xls. Es compatible con las versiones de MS-Office 97-2003.

• XSSFWorkbook: Esta clase tiene métodos para leer y escribir archivos XML de Open Office y Microsoft Excel en formato .xls o .xlsx. Es compatible con las Versiones de MS-Office 2007 o posterior.

HSSFWorkbook

Es una clase de alto nivel bajo el paquete org.apache.poi.hssf.usermodel. Implementa la interfaz de libro y se utiliza para archivos de Excel en formato .xls. A continuación, se enumeran algunos de los métodos y constructores en virtud de esta clase.

Constructores de clase:

* HSSFWorkbook - Crea un nuevo objeto HSSFWorkbook desde cero.
* HSSFWorkbook (directorio DirectoryNode, preserveNodes booleanas). Crea un nuevo HSSFWworkbook objectinside, un directorio específico.
* HSSFWorkbook (directorio DirectoryNode,fs POIFSFileSystem, preserveNodes booleanas). Dado un objeto POIFSFileSystem y un directorio específico dentro de ella, se crea un objeto SSFWorkbook para leer un libro especificado.
* HSSFWorkbook (java.io.InputStream), crea un nuevo objeto HSSFWorkbook utilizando un flujo de entrada.
* HSSFWorkbook (booleanos, preserveNodes java.io.InputStream s), construye un sistema de archivos POI alrededor de su flujo de entrada.
* HSSFWorkbook (POIFSFileSystem), construye un nuevo objeto HSSFWorkbook utilizando un objeto POIFSFileSystem.
* HSSFWorkbook ( POIFSFileSystem, preserveNodes booleanas), dado un objeto POIFSFileSystem, se crea un nuevo objeto HSSFWorkbook para leer un libro especificado.

XSSFWorkbook

Es una clase que se utiliza para representar los formatos de archivo de alta y baja el nivel de Excel. Pertenece al paquete org.apache.xssf.usemodel e implementa la interfaz libro. A continuación, se enumeran los métodos y constructores en virtud de esta clase.

Constructores de clase:

* XSSFWorkbook. Crea un nuevo objeto XSSFworkbook desde cero.
* XSSFWorkbook (archivo java.io.File). Construye un objeto XSSFWorkbook de un archivo determinado.
* XSSFWorkbook (java.io.InputStream es). Construye un objeto XSSFWorkbook, amortiguando todo el flujo de entrada en la memoria y luego abrir un objeto OPCPackage por ello.
* XSSFWorkbook (camino java.lang.String). Construye un objeto XSSFWorkbook dada la ruta completa de un archivo.

Métodos de clase

* CreateSheet. Crea un XSSFSheet para este libro, lo añade a las hojas, y devuelve una representación de alto nivel.
* CreateSheet (java.lang.String sheetname). Crea una nueva hoja de este libro de trabajo y devuelve una representación de alto nivel.
* CreateFont. Crea un nuevo tipo de letra y lo añade a la tabla de fuente del libro.
* CreateCellStyle. Crea un nuevo XSSFCellStyle y lo agrega a la tabla del estilo del libro.
* CreateFont. Crea un nuevo tipo de letra y lo añade a la tabla de fuente del libro.
* SetPrintArea (int sheetIndex, int StartColumn, int EndColumn, int startRow, int Filafinal). Establece el área de impresión de una hoja determinada de acuerdo con los parámetros especificados.

Hoja

Hoja es una interfaz bajo el paquete org.apache.poi.ss.usermodel y es un súper-interfaz de todas las clases que crean hojas de cálculo de alto o bajo nivel con nombres específicos. El tipo más común de hoja de cálculo es hoja de cálculo, que se representa como una cuadrícula de células.

HSSFSheet

Esta es una clase en el paquete org.apache.poi.hssf.usermodel. Se puede crear hojas de cálculo Excel y permite dar formato a los datos de estilo de hoja y hoja.

Constructores de clase:

* HSSFSheet (HSSFWorkbook libro). Crea un nuevo HSSFSheet llamado por HSSFWorkbook para crear una hoja de cero.
* HSSFSheet (HSSFWorkbook libro, hoja InternalSheet). Crea un HSSFSheet que representa el objeto de hoja determinada.

XSSFSheet

Esta es una clase que representa una representación de alto nivel de la hoja de cálculo Excel. Es bajo org.apache.poi.hssf.usermodel paquete.

Constructores de clase:

* XSSFSheet Crea un nuevo XSSFSheet - llamado por XSSFWorkbook para crear una hoja de cero.
* XSSFSheet (PackagePart parte, PackageRelationship rel). Crea un XSSFSheet que representa la parte paquete determinado y la relación.

Métodos de clase:

* AddMergedRegion (región DireccionAreaCeldas). Añade una región fusionado de las células (por lo tanto, estas células forman una).
* AutoSizeColumn (int columna) Ajusta el ancho de columna para adaptarse a los contenidos.
* Iterador Este método es un alias para RowIterator ()
* AddHyperlink (XSSFHyperlink hipervínculo). Registra un hipervínculo en la colección de hipervínculos en esta hoja.

Fila

Esta es una interfaz bajo el paquete org.apache.poi.ss.usermodel. Se utiliza para la representación de alto nivel de una fila de una hoja de cálculo. Es un súper-interfaz de todas las clases que representan las filas de la biblioteca PDI.

XSSFRow

Esta es una clase en el paquete org.apache.poi.xssf.usermodel. Implementa la interfaz de la fila, por lo que puede crear filas en una hoja de cálculo. A continuación, se enumeran los métodos y constructores en virtud de esta clase.

Métodos de clase:

* createCell (int columnIndex). Crea nuevas células dentro de la fila y lo devuelve.
* setHeight (estatura). Establece la altura en unidades cortas.

Celda

Esta es una interfaz bajo el paquete org.apache.poi.ss.usermodel. Es un súper-interfaz de todas las clases que representan las células en las filas de una hoja de cálculo.

Las células pueden tener varios atributos tales como blanco, numérico, fecha, errores, etc. células deben tener sus propios números (0) basados antes de ser añadido a una fila.

XSSFCell

Esta es una clase en el paquete org.apache.poi.xssf.usermodel. Implementa la interfaz de la célula. Es una representación de alto nivel de las células en las filas de una hoja de cálculo.

Campo Resumen

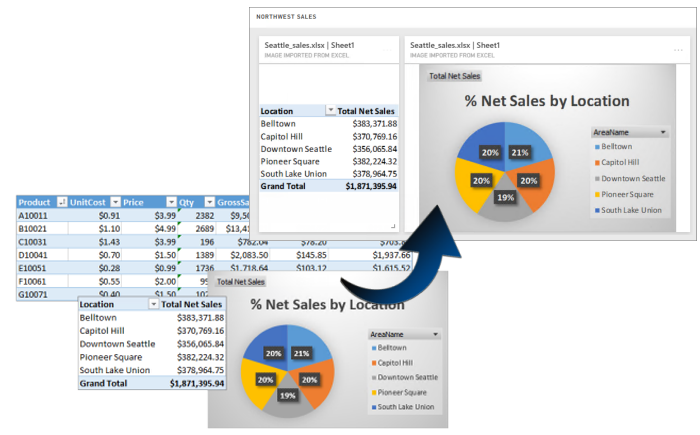
A continuación, se enumeran algunos de los campos de la clase XSSFCell junto con su descripción.

|  |  |
| --- | --- |
| Tipo de célula | Descripción |
| CELL\_TYPE\_BLANK | Representa celda en blanco |
| CELL\_TYPE\_BOOLEAN | Representa celular booleano (verdadero o falso) |
| CELL\_TYPE\_ERROR | Representa el valor de error en una celda |
| CELL\_TYPE\_FORMULA | Representa resultado de la fórmula en una celda |
| CELL\_TYPE\_NUMERIC | Representa los datos numéricos en una célula |
| CELL\_TYPE\_STRING | Representa cadena (texto) en una célula |



## Microsoft Excel: Importación y Exportación de Datos

**Power BI Publisher para Excel**



¿Qué puede anclar? Prácticamente cualquier dato de una hoja de cálculo de Excel. Puede seleccionar un rango de celdas de una hoja o tabla, una tabla dinámica o un gráfico dinámico, ilustraciones e imágenes, y texto.

Al anclar un elemento desde Excel, se agrega un nuevo icono a un panel nuevo o existente en Power BI. El nuevo icono es una instantánea, por lo que no es dinámico, aunque se puede actualizar. Por ejemplo, si realiza un cambio en una tabla dinámica o un gráfico ya anclados, el icono del panel de Power BI no se actualiza automáticamente, pero todavía puede actualizar los elementos anclados mediante el Administrador de anclaje. Puede obtener más información sobre el Administrador de anclaje en las secciones siguientes.

**Power Pivot para Excel**

Usando Power Pivot para Excel, puede transformar grandes cantidades de datos con increíble velocidad información significativa para obtener las respuestas que necesita en segundos.

Aprovechar las características y herramientas familiares de Excel

Proceso de grandes cantidades de datos en segundos

Cargar incluso los más grandes conjuntos de datos desde prácticamente cualquier origen, como bases de datos, las fuentes de datos, Reporting Services y archivos de texto

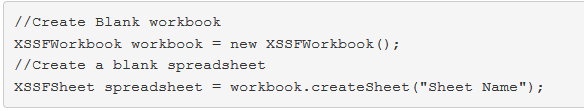
Usar eficaces capacidades analíticas nuevas, como expresiones de análisis de datos (DAX)

## CASO DEMOSTRATIVO: CREAR UNA HOJA DE CÁLCULO CON POI

**Crear una hoja de cálculo**

En primer lugar, vamos a crear una hoja de cálculo utilizando las clases referenciadas discutidos en los capítulos anteriores. Siguiendo el capítulo anterior, crear un libro primero y luego podemos seguir adelante y crear una hoja.

El código que se utiliza para crear una hoja de cálculo:



**Las filas en la hoja de cálculo**

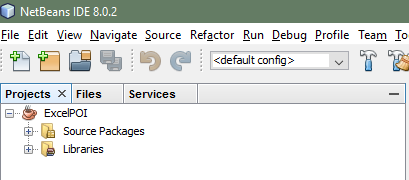
Las hojas de cálculo tienen un diseño de cuadrícula. Las filas y columnas se identifican con nombres específicos. Las columnas se identifican con letras y las filas con números.

El código que se utiliza para crear una fila:

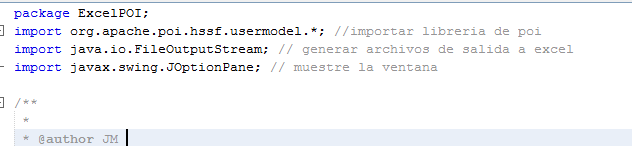


**Ejemplo:**

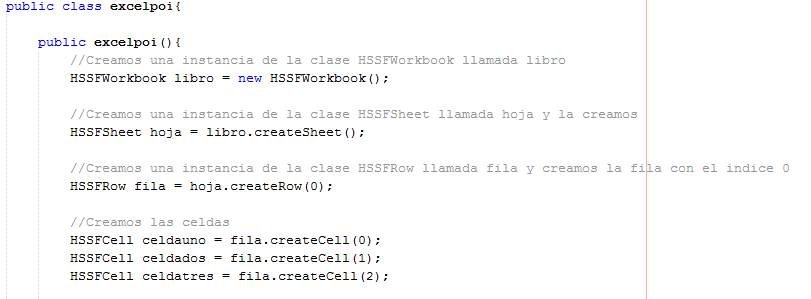
Crear proyecto **ExcelPOI**



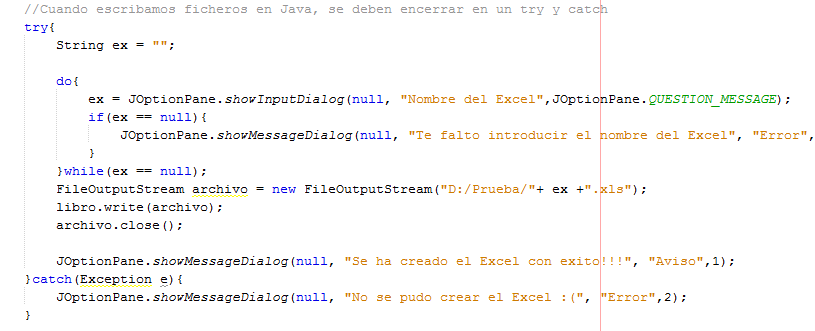
Los tres paquetes que se tienen que importar:



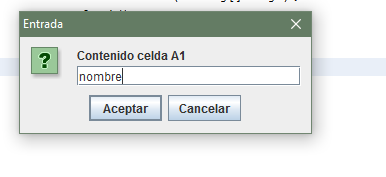
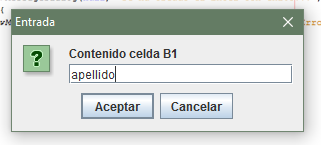
Creación del Libro, Hoja y las filas (Excel)

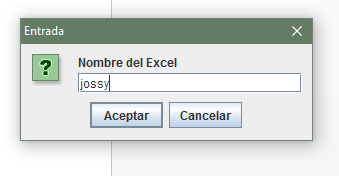
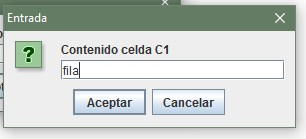


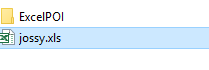
Cuando escribimos ficheros en Java, se deben encerrar en un try Catch



**RESULTADO**





Importando EXCEL:  


# CONCLUSIONES

La posición de Microsoft Excel dentro de las organizaciones ha permitido que su utilización se expanda a distintas aplicaciones y fabricantes de software que aprovechan sus características para entregar al usuario final, resultados que en otros casos sería muy difícil de alcanzar.

Microsoft Excel es capaz de importar datos desde diferentes orígenes para realizar cálculos o análisis financieros, de la misma forma permite generar reporte de gran impacto y relevancia.

Por otro lado, Excel permite ser utilizado como origen de datos, de tal manera que aplicaciones como Power BI consuman la información contenida en sus hojas y libros para entregar reportes con data cruzada para tareas de inteligencia de negocio.

# REFERENCIAS BIBLIOGRÁFICAS

ABC, D. (s.f.). *Hoja de Calculo*. Obtenido de http://www.definicionabc.com/negocios/hoja-de-calculo.php

Apache. (s.f.). *Apache POI - The Java API for MS Documents*. Obtenido de https://poi.apache.org/

CodeJava. (s.f.). *How to Read Excel Files in Java with Apache POI*. Obtenido de http://www.codejava.net/coding/how-to-read-excel-files-in-java-using-apache-poi

Microsoft. (s.f.). *Power BI Publisher for Excel*. Obtenido de https://powerbi.microsoft.com/es-es/documentation/powerbi-publisher-for-excel/

Microsoft. (s.f.). *Power Pivot: Análisis de Datos con Excel*. Obtenido de https://support.office.com/es-es/article/Power-Pivot-an%C3%A1lisis-de-datos-eficaz-y-modelado-de-datos-en-Excel-d7b119ed-1b3b-4f23-b634-445ab141b59b

Point, T. (s.f.). *Apache POI Tutorial*. Obtenido de http://www.tutorialspoint.com/apache\_poi/

Wikipedia. (s.f.). *Microsoft Excel*. Obtenido de https://es.wikipedia.org/wiki/Microsoft\_Excel