

Universidad de San Carlos de Guatemala
Facultad de Ingeniería

Curso: (0772) **Estructuras de Datos**



**Proyecto 1:
Manual Técnico**

Erik Vladimir Girón Márquez
Carnet # 200313492
Sección B

Guatemala, 23 de Marzo de 2006

Introducción.

ElectricSheet es una aplicación de hojas electrónicas, implementando expresiones aritméticas, funciones matemáticas entre celdas, múltiples hojas por documento y apertura y generación de archivos.

El programa implementa un diseño puramente orientado a objetos, que junto con las características de programación por eventos de Java, simplificaron en gran parte el desarrollo visual e interactivo de la aplicación.

Se utilizó entonces para el desarrollo del proyecto, la plataforma Java. Permitiéndose compilar la aplicación bajo *Java JDK v 1.5* y utilizando *netBeans* para la administración del proyecto, que gracias al sencillo editor de GUI con que cuenta este *IDE*, permitió generar de una manera rápida y segura el código necesario para la manipulación de eventos sobre el UI.

En este documento, el programador se podrá guiar por medio de diagramas UML a través del diseño del proyecto y de cada una de las clases que la componen, contando con descripciones de cada clase. Se recomienda que para ampliar la información dada aquí, se consulte el manual *JavaDoc* generado desde la documentación del código de la aplicación, y localizado en la distribución oficial, además de revisar el código fuente si se tiene alguna duda sobre la implantación del código.

Diseño General y descripción general de paquetes y clases.

A continuación se presenta el diseño general de las clases en diagramas *UML* junto con un resumen de los paquetes y clases que componen el proyecto.

Se recomienda complementar esta información con el conjunto de documentos *JavaDoc* en formato *HTML*, generado para el proyecto, en donde se describen detalladamente cada clase y su funcionamiento, con sus métodos y campos; que por causas de extensión no fue impreso en papel. Sin embargo El documento *JavaDoc* se encuentra incluido en la distribución del proyecto

Jerarquía de clases:

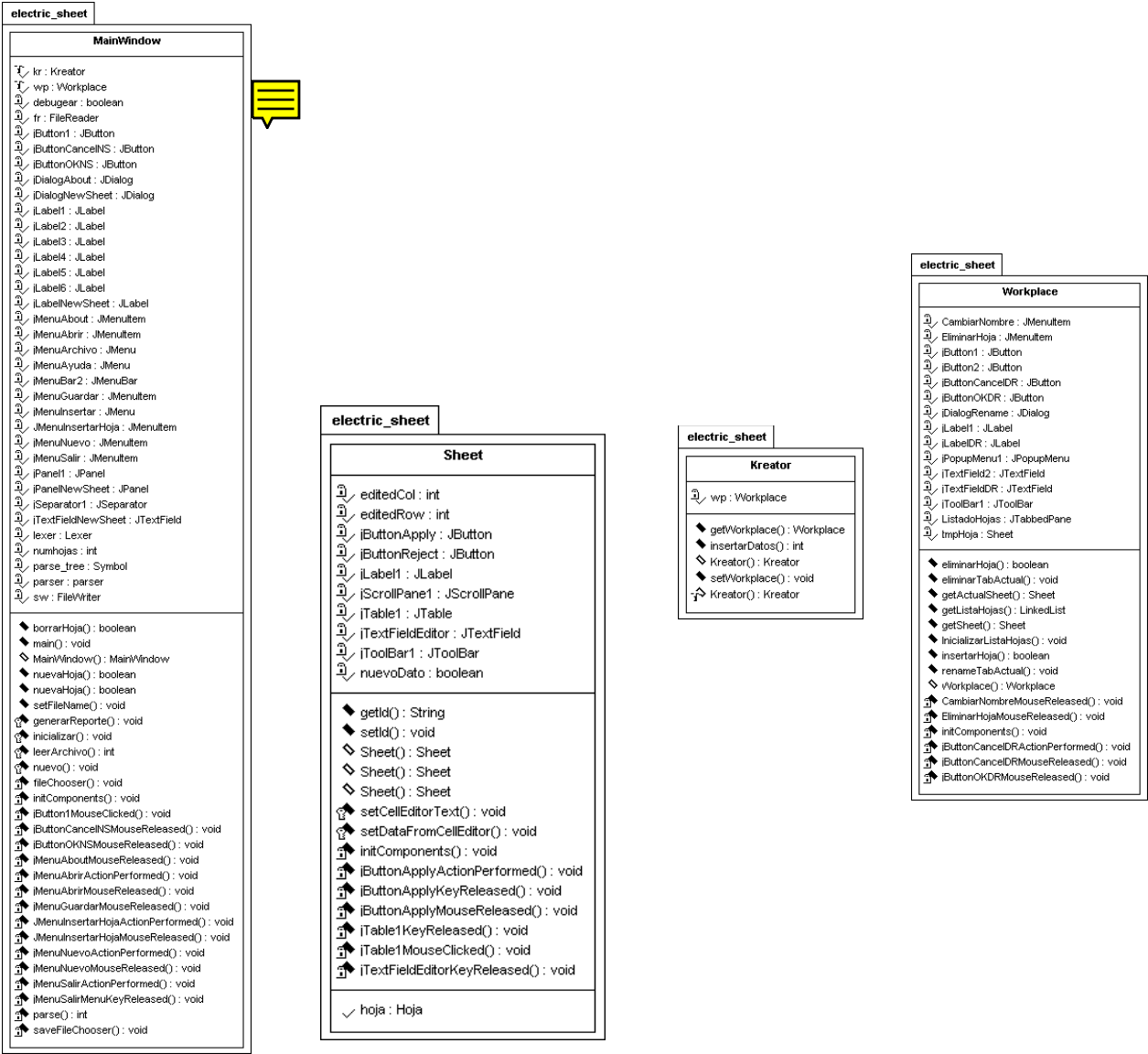
java.lang.Object

- javax.swing.table.AbstractTableModel (implements java.io.Serializable, javax.swing.table.TableModel)
 - tda.[Hoja](#)
- java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
 - java.awt.Container
 - javax.swing.JComponent (implements java.io.Serializable)
 - javax.swing.JPanel (implements javax.accessibility.Accessible)
 - electric_sheet.[Sheet](#)
 - electric_sheet.[Workplace](#)
 - java.awt.Window (implements javax.accessibility.Accessible)
 - java.awt.Frame (implements java.awt.MenuContainer)
 - javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
 - electric_sheet.[MainWindow](#)
 - tda.[Dato](#)
 - electric_sheet.[Kreator](#)
 - lex.[Lexer](#) (implements java_cup.runtime.Scanner)
 - java_cup.runtime.lr_parser
 - lex.[parser](#)
 - tda.[Matriz](#)
 - tda.[Operacion](#)
 - lex.[sym](#)

electric_sheet
Kreator
MainWindow
Sheet
Workplace

Paquete electric_sheet:
 Contiene toda la interfaz gráfica interactiva del proyecto.

Class Summary	
Kreator	Clase generadora de hojas que se van leyendo en el archivo de entrada
MainWindow	Clase principal que contiene la ventana principal de la aplicacion, contiene metodos para leer y escribir archivos, pasando los streams a otros objetos para que lo analizen.
Sheet	Representa a la interfaz de grafica de usuario, de tabla de datos y su respectivo editor de funciones.
Workplace	Espacio de trabajo que anida el listado de hojas



lex
Lexer parser sym

Paquete Lex:

Contiene clases que hacen posible el análisis léxico y sintáctico del proyecto, éstas fueron generadas con JavaLex para el scanner y CUP para el parser.

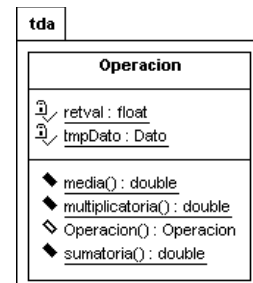
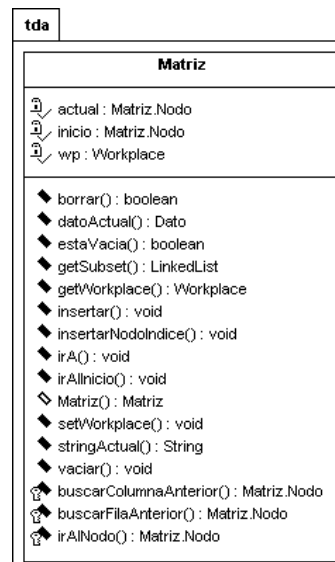
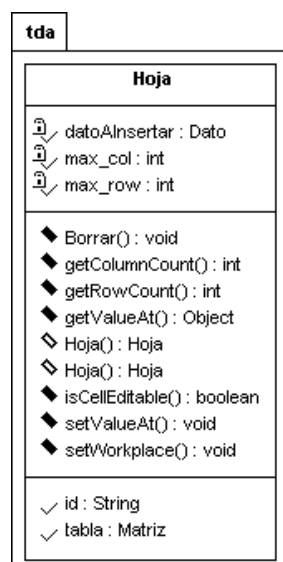
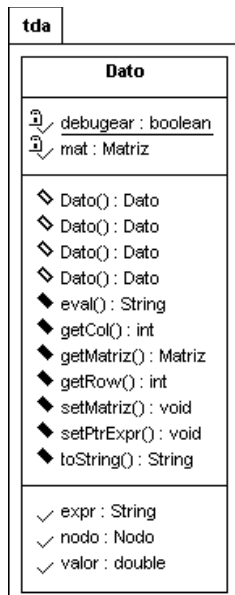
tda
Dato Hoja Matriz Operacion

Paquete TDA:

Incluye la estructura de datos utilizada para representar una matriz dinámica, así como el dato que puede contener la matriz en cada una de sus celdas. Incluye también operaciones que se realizan dentro de las celdas para referenciar a otras celdas.

Class Summary

<u>Dato</u>	Clase que almacena los datos guardados en las celdas de la hoja electronica.
<u>Hoja</u>	Modelo de tabla que implementa una matriz ortogonal como DDE para representar las series de datos
<u>Matriz</u>	Clase de la estructura de datos(matriz Ortogonal) a utilizar
<u>Operacion</u>	Clase que agrupa operaciones a realizar con un conjunto de datos dados en forma de una lista enlazada



Limitaciones

Volcado y recuperación de datos hacia dispositivo de memoria secundario.

Por cuestiones de tiempo, no fue posible implementar un método para generar algún archivo (ya sea *XML*, *ODF* o serializado) para almacenamiento de los datos contenidos en las hojas, mucho menos para recuperar el archivo guardado.