

HOJA ELECTRONICA ESTADISTICA INFERENCIAL

Alex Moreno Salazar¹, Luis Rodriguez Ojeda²

¹Ingeniería en Estadística Informática 2000

²Director de Tesis, Master en Computación, Profesor del Instituto de Ciencias Matemáticas de la ESPOL desde 1979.

RESUMEN

La Hoja Electrónica Estadística Inferencial, la cual hemos llamado HOELEST, es un sistema orientado al análisis de datos y cálculo estadístico inferencial. Discutiremos entonces sobre la creación de una hoja electrónica con funciones incorporadas exclusivamente relacionadas con la Estadística Inferencial, (inferencia o predicción de un suceso).

El objetivo de la Estadística es hacer inferencias con respecto a parámetros poblacionales desconocidos, basadas en la información obtenida mediante datos muestrales. De esta manera, con respecto a Estadística Inferencial, HOELEST proporciona facilidades para: Pruebas de Hipótesis, Modelos Lineales, Diseño de Experimentos, Análisis de Varianza, Covarianzas y Coeficientes de Correlación Lineal Simple, Pruebas de Kruskal-Wallis y Pruebas de Wilcoxon.

INTRODUCCION

La creación de un programa que contenga características tanto de una *Hoja Electrónica* como las de un *Paquete de Análisis Estadístico*, es un desafío muy atractivo para quienes gustan de la Estadística y de la Informática.

Por un lado, un programa que sólo maneja hojas electrónicas, no contiene funciones predefinidas especializadas en análisis estadístico de datos. En caso de que contara con algunas de ellas, el propósito principal de estos programas no es el análisis estadístico, sino prestar sus características para trabajos contables o la capacidad de automatizar cálculos sencillos.

En contraste, los programas estadísticos no proporcionan al usuario bondades como la capacidad de editar fórmulas directamente en las celdas o para presentar sus datos en forma atractiva (formatear), sea por medio visual o impreso; usualmente se trabaja en forma alterna con un procesador de palabras o alguna aplicación de hojas electrónicas, para obtener la apariencia deseada al momento de presentar reportes.

La hoja electrónica estadística pretende reunir las características de estas dos clases de aplicaciones; siendo orientada al análisis estadístico de datos al contar con funciones especializadas para ello y, con la capacidad de las hojas electrónicas para formatear los datos y la edición de fórmulas directamente, sin necesidad de cuadros de diálogos especiales.

CONTENIDO

La Hoja Electrónica Estadística HOELEST, es un sistema integrado por dos componentes ejecutables de código. Uno de ellos se llama Hoeslest.exe, el cual proporciona la interfaz visual de la hoja electrónica, con un conjunto de funciones para la manipulación y presentación de datos. Debido a este componente se justifica la parte “Hoja Electrónica”.

El otro componente se llama EstadisticaInferencialDLL. Proporciona todas las funciones para el análisis estadístico inferencial; de aquí se justifica la parte “Estadística” de HOELEST.

HOELEST es una aplicación controlada por eventos, la ejecución no sigue una ruta predeterminada, sino que ejecuta distintas secciones de código como respuesta a eventos. Los eventos pueden desencadenarse por acciones del usuario, mensajes del sistema o de otras aplicaciones, o incluso desde la propia aplicación. La secuencia de estos eventos determina la secuencia en la que se ejecuta el código. Por tanto, la ruta a través del código de la aplicación difiere cada vez que se ejecuta el programa.

HOELEST fue diseñada para trabajar bajo Microsoft Windows 95 o posterior, es decir, no contempla su funcionamiento en otras plataformas como Macintosh o Linux. Además, no sólo puede ser utilizada como una aplicación de escritorio por un usuario común, ya que gracias a su modelo de objetos, usuarios experimentados (programadores) pueden utilizarlo como servidor de funciones estadísticas para crear aplicaciones personalizadas e incluso expandir su utilización en otras áreas como Finanzas por ejemplo.

1. El Entorno de desarrollo

HOELEST fue desarrollado en Visual Basic 5.0, un lenguaje de programación basado en objetos en el entorno Windows. Para su implementación se utilizaron controles estándar y ActiveX. Básicamente, el control MSFlexGrid de Microsoft proporcionó la interfaz gráfica de la hoja, así como características de scroll (movimiento a través de la hoja). Otros controles tales como el TextBox se utilizaron para edición de texto, números y fórmulas.

Los objetos tienen propiedades, métodos y eventos. Las propiedades son los datos que describen un objeto, los métodos son lo que se puede hacer con el objeto y los eventos son tareas que realiza el objeto. Por ejemplo, un auto es un objeto, entre sus propiedades podemos hablar de color, modelo, número de puertas; entre sus métodos están: mover, detener. Por último, un par de eventos pueden ser encenderse o apagarse.

Los objetos de Visual Basic se crean a partir de clases; así, un objeto se dice que es una instancia de una clase. La clase define las interfaces de un objeto, si el objeto es público y en qué circunstancias se puede crear. Las

descripciones de las clases se almacenan en bibliotecas de tipos y se pueden ver con examinadores de objetos.

Para utilizar un objeto se tiene que mantener una referencia a él en una variable de objeto. El tipo de enlace determina la velocidad con la que se tiene acceso a los métodos de un objeto mediante la variable de objeto. Una variable de objeto puede ser un enlace en tiempo de compilación (el más lento) o un enlace en tiempo de diseño. Al conjunto de propiedades y métodos se le llama interfaz.

2. El Modelo de Objetos Componentes

El Modelo de objetos componentes (COM) es una arquitectura orientada a objetos para generar aplicaciones y la base sobre la que se generan los componentes ActiveX. Los componentes ActiveX permiten a los programadores combinar fragmentos de código reutilizables para crear aplicaciones y servicios.

COM tiene dos aspectos diferentes. El primero es que la especificación COM proporciona una definición (o modelo) de lo que es un objeto. El segundo es que COM proporciona servicios para la creación de objetos y para las comunicaciones entre el cliente y el servidor.

Un componente es una unidad de código ejecutable que proporciona funcionalidad. Los componentes los proporcionan los servidores que son archivos .exe, .dll u .ocx. Los servidores pueden contener más de un componente y los componentes proporcionan las plantillas a partir de las cuales se crean los objetos. COM especifica cómo se crean los componentes y cómo las aplicaciones cliente se conectan con dichos componentes. COM también procesa las solicitudes de las aplicaciones cliente para crear objetos. Un objeto es una combinación de código y datos que se puede tratar como una unidad. Los objetos tienen una duración; es decir, se crean y se destruyen.

Automatización es la parte de la especificación COM que define un método estándar para la creación de componentes y el uso de los objetos. Los objetos de Automatización también se conocen como objetos programables. Para un programador de aplicaciones cliente, un objeto recibe una entrada y proporciona una salida, pero el programador no puede tener acceso a su funcionamiento interno. Los programadores sólo necesitan saber cómo pueden utilizar la funcionalidad ofrecida por los objetos. Dicha funcionalidad se ofrece a través de una o varias interfaces publicadas. Estas interfaces son el medio por el que las aplicaciones cliente se comunican con el componente.

2.1. Ventajas del uso de COM

La tecnología de componentes facilita la creación de aplicaciones más eficientes al proporcionar ventajas como la compatibilidad y reutilización del código y el control de versiones. HOELEST tiene estas características.

3. Ventajas de HOELEST

Al ser un programa que contiene algunas características de las hojas electrónicas y de los paquetes estadísticos, resulta más amigable que un paquete estadístico puro. Recordemos que el propósito principal de HOELEST es el Análisis Estadístico. En sus ventajas podemos detallar:

- ☒ Es posible editar fórmulas sin necesidad de ingresar a un menú o a un cuadro de diálogo especial. (esta característica no suele presentarse en paquetes estadísticos)
- ☒ Se presentan los datos con mayor vistosidad.
- ☒ Es un programa flexible.
- ☒ Esta a disposición de los estudiantes del ICM.
- ☒ Proporciona un modelo de objetos a programadores experimentados.
- ☒ Reutilización del código
- ☒ Obtiene las ventajas del Modelo de Componentes COM

Al ser desarrollado en Visual Basic 5.0, recibe beneficios como:

- ☒ Desarrollo más rápido
- ☒ Calidad más alta
- ☒ Mantenimiento más fácil

4. Desventajas de HOELEST

- ☒ HOELEST no cuenta con algunas características de los programas de hojas electrónicas, como por ejemplo la revisión ortográfica.
- ☒ Incompatibilidad con otras aplicaciones (Microsoft Excel por ejemplo).
- ☒ No existe una ventana de comandos propia. Para utilizarlo mediante código, se necesita que el usuario tenga Visual Basic 5.0 o superior instalado en su equipo.

5. Requisitos del Sistema para HOELEST

Se requiere el siguiente hardware y software:

- ☒ Microsoft Windows NT 3.51 o posterior, o Microsoft Windows 95 o posterior.
- ☒ Microprocesador 80486 o superior.
- ☒ Pantalla VGA o de resolución superior compatible con Microsoft Windows.
- ☒ 8 MB de RAM para aplicaciones
- ☒ 16 MB de RAM para el entorno de desarrollo de Visual Basic (en caso de que el usuario desee modificar la estructura de Hoelest o se desee manipularlo mediante código externo).

6. Componentes de código.

Un componente de código ActiveX es una aplicación servidora que expone funcionalidad que otras aplicaciones pueden utilizar y reutilizar mediante

Automatización. Los componentes de código son bibliotecas de objetos. Las aplicaciones cliente utilizan los componentes de código mediante la creación de objetos a partir de las clases proporcionadas por el componente. Los clientes llaman a las propiedades, métodos y eventos proporcionados por el objeto.

Cualquier aplicación que admita Automatización estándar puede utilizar los componentes de código ActiveX creados en Visual Basic. Esto incluye Visual Basic, Microsoft Excel, Microsoft Access, Microsoft Project, Microsoft Visual FoxPro, Microsoft Visual C++ e incluso HOELEST.

7. Componente EstadísticaInferencialDLL

El propósito de este componente, es brindar a sus clientes un conjunto de funciones y procedimientos para el análisis estadístico de datos; por ejemplo, se proporciona un método ANOVA para realizar un análisis de varianza.

EstadísticaInferencialDLL mantiene a su vez, referencias a otros dos componentes DLL: Matrices y DistribucionesProy. A continuación detallaremos los servicios que prestan cada uno de ellos.

7.1. Matrices

Matrices es un componente de código DLL Active X. El propósito de este componente, es brindar a sus clientes un conjunto de funciones y procedimientos que manipulan matrices; por ejemplo, se proporciona un método para obtener la inversa de una matriz cuadrada.

7.1.1. Objetos del Componente Matrices.

El componente Matrices proporciona un sólo objeto, cuyo único propósito es permitir el acceso a los métodos proporcionados por el componente. El nombre del objeto es Matrices. No tiene interfaz visual.

7.2. Componente DistribucionesProy

DistribucionesProy es un componente de código DLL Active X. El propósito de este componente, es brindar a sus clientes un conjunto de funciones que devuelven valores y probabilidades de las principales funciones de probabilidad, como por ejemplo proporciona un método para calcular la probabilidad de una variable aleatoria Normal dado un punto en la recta real.

7.2.1. Objetos del Componente DistribucionesProy.

El componente DistribucionesProy proporciona un sólo objeto, cuyo único propósito es permitir el acceso a los métodos proporcionados por el componente. El nombre del objeto es Distribuciones. No tiene interfaz visual.

7.3. Diseño del modelo de objetos de Estadística Inferencial DLL

El componente Estadística Inferencial DLL proporciona un sólo objeto, cuyo único propósito es permitir el acceso a los métodos proporcionados por el componente. EL nombre del objeto es Estadística Inferencial. No tiene interfaz visual.

8. Componente Hoelest.exe

El nombre del componente EXE ActiveX es HOELEST. Su función principal es la de proporcionar la interfaz visual de la aplicación de la hoja electrónica. HOELEST.EXE utiliza el componente de código DLL ActiveX Estadística Inferencial DLL para el análisis estadístico de datos y, se encarga de la presentación visual e impresión de datos y resultados.

8.1. Diseño de la Interfaz Visual

8.1.1. Estilos de interfaz

Hay dos estilos principales de interfaz de usuario: la interfaz de documento único (SDI) y la interfaz de documentos múltiples (MDI). Un ejemplo de la interfaz SDI es la aplicación WordPad incluida en Microsoft Windows. En WordPad, sólo se puede abrir un documento y se debe cerrar un documento para abrir otro.

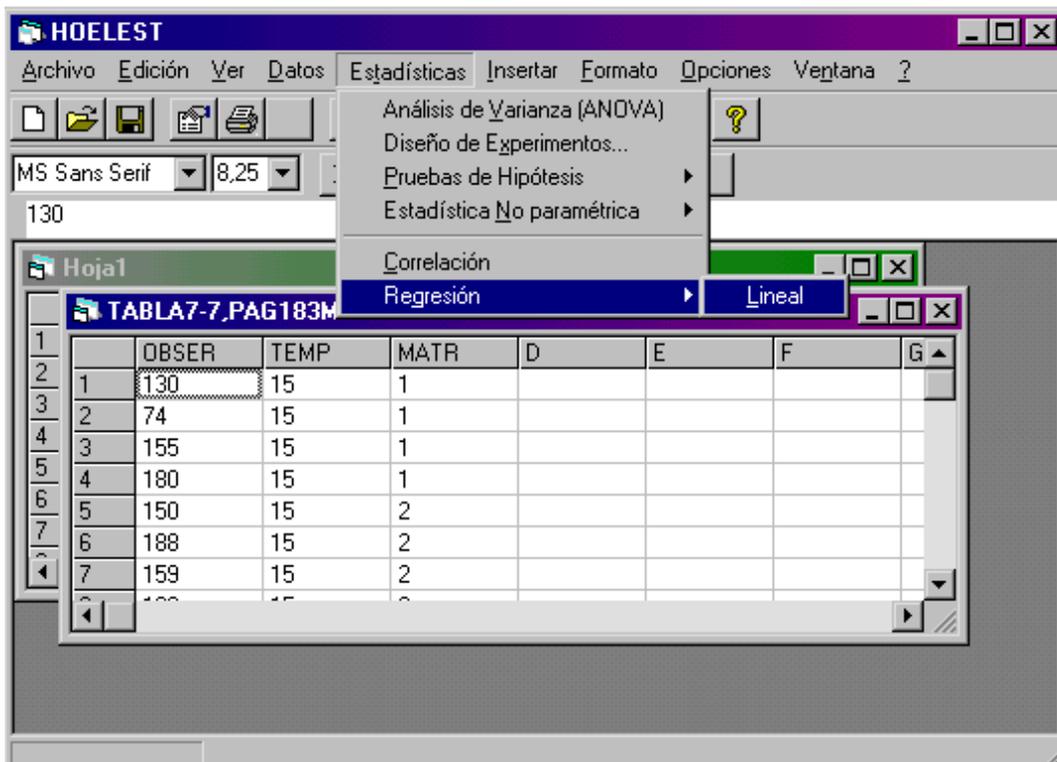


Figura 1. Ventana Principal de Hoelest

Aplicaciones como Microsoft Excel para Windows son interfaces MDI; le permiten presentar varios documentos al mismo tiempo y cada documento se presenta en su propia ventana. Los documentos o ventanas secundarias están contenidos en una ventana primaria, que proporciona un espacio de trabajo para todas las ventanas secundarias de la aplicación.

8.1.2. La Ventana Principal.

Para determinar cuál es el mejor estilo de interfaz, nos basamos en el propósito de la aplicación. Decidimos que HOELEST debería tener el estilo de interfaz MDI, ya que es posible que el usuario necesite trabajar con más de una hoja electrónica a la vez, por ejemplo comparar Tablas de Análisis de Varianza de dos experimentos similares. La figura 1 corresponde al formulario MDI de HOELEST, en el que podemos apreciar la barra de menús, las barras de herramientas y de estado.

8.2. Diseño del Modelo de Objetos

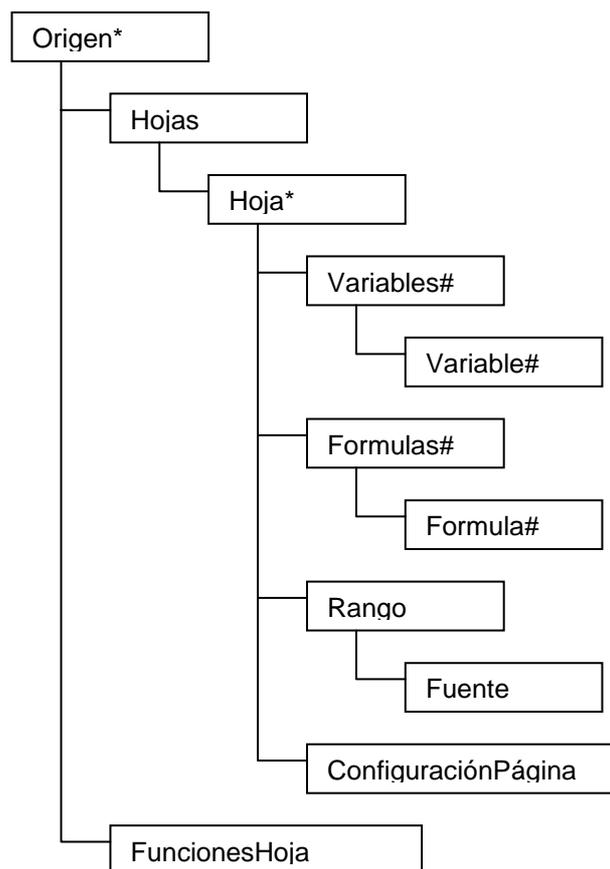


Figura 2. Modelo de Objetos del componente Hoelest.Exe

El modelo de objetos de Hoelest define la relación entre los objetos del componente. Se trata de un pequeño modelo, que permite a los programadores manipular los datos contenidos en la hoja. Permite además

que la aplicación sea utilizada para otros servicios, por ejemplo la creación de un servidor de funciones de cálculo actuarial. Los programadores de aplicaciones cliente pueden ver el modelo con ayuda de examinadores de objetos.

En la figura 2 podemos apreciar 11 objetos. Aquellos con el signo (*) a la derecha del nombre, son objetos de creación externa, es decir, mediante código externo es posible crear instancias de ellos. Aquellos con el signo (#) son objetos privados, es decir, no son visibles para los programadores debido a que no hay razón alguna para permitir que éstos puedan manipularlos.

El objeto *Origen* representa a la aplicación completa; contiene valores y opciones de toda la aplicación. El objeto *Hojas* es un objeto de conjunto, es decir, contiene a varios objetos *Hoja*. Cada objeto *Hoja* representa a una hoja electrónica. Los objetos *Formulas* y *Variables* también contienen a varios objetos *Formula* y *Variable* respectivamente. Como se puede apreciar, pueden existir varios objetos *Formula* y *Variables* dentro de una hoja electrónica.

El objeto *Rango* representa una celda, una fila, una columna, o una selección de celdas que contienen uno o más bloques contiguos de celdas. Por medio de él es posible manipular los datos contenidos en las celdas. El objeto *Fuente* sirve para establecer propiedades de formato de texto. El objeto *ConfiguraciónPágina* permite manipular fácilmente las opciones de configuración de la impresión de datos. Por último, el objeto *FuncionesHoja* proporciona funciones que los usuarios utilizan frecuentemente (ej: sumas).

Se recomienda que esta aplicación (su estructura interna) se ponga a disposición de los alumnos de la Escuela Superior Politécnica del Litoral que siguen carreras afines a la computación, para que sea utilizada como material de referencia en cursos de programación orientada a objetos y tecnología de componentes ActiveX. Para los estudiantes de Ingeniería en Estadística Informática, reviste principal interés, debido al componente estadístico que forma parte del sistema, pudiendo ser de utilidad en cursos de estadística computacional por ejemplo.

CONCLUSIONES

- ∅ El componente Hoesest.exe es un servidor fuera de proceso, que permite interactuar con el usuario común. Puede ser ejecutado como una aplicación de escritorio (como Microsoft Word) o como un componente servidor. Los objetos que proporciona permiten manipularlo mediante código, así como Microsoft Excel puede ser manipulado por medio de macros.
- ∅ Cada componente de la Hoja Electrónica Estadística tiene su propio modelo de objetos. Esta característica, propia de los componentes ActiveX, permiten que puedan ser utilizados como servidores en cualquier plataforma

y por cualquier programa que permita Automatización. Sin embargo, Hoelest se desarrolló exclusivamente para su uso en el sistema operativo Windows, siendo una opción expandir su nivel de operación a otras plataformas, como Macintosh por ejemplo.

- ∅ Quedan fuera del alcance de Hoelest varias técnicas estadísticas como por ejemplo: análisis de correlación canónica, análisis de series de tiempo, herramientas para el control de calidad, etc.
- ∅ En tiempo de ejecución, es decir cuando un usuario está utilizando Hoelest, se generan errores. Hoelest intercepta estos errores gracias a su código de tratamiento de errores, por lo que se convierte en una aplicación más robusta. Además permite a una aplicación externa interceptar los errores que se producen en él.
- ∅ El componente Hoelest.exe tiene una interfaz de múltiples documentos, que le permite trabajar con varios archivos abiertos a la vez. Esta es una característica que no suele presentarse en las aplicaciones existentes de cálculo estadístico.
- ∅ Las descripciones de las clases de EstadísticaInferencialDLL y Hoelest.exe se almacenan en bibliotecas de tipos y se pueden ver con examinadores de objetos.
- ∅ El tipo de enlace determina la velocidad con la que se tiene acceso a los métodos de un objeto mediante la variable de objeto. El componente Hoelest.exe se comunica internamente entre sus objetos por medio de enlaces en tiempo de diseño, lo que mejora su rendimiento sustancialmente.
- ∅ Las interfaces múltiples son una característica del Modelo de Objetos Componentes (COM); permiten que tanto EstadísticaInferencialDLL como Hoelest.exe evolucionen con el tiempo, agregando nueva funcionalidad sin afectar al código existente.
- ∅ Los componentes ActiveX EstadísticaInferencialDLL y Hoelest.exe permiten a los programadores combinar fragmentos de código reutilizables para crear aplicaciones y servicios. De esta manera, un programador puede convertir a Microsoft Excel en un cliente de EstadísticaInferencialDLL y utilizar sus métodos. También se puede convertir Hoelest.exe en un cliente, digamos de un servidor de funciones financieras, para utilizar sus métodos y presentar los resultados como lo hace con EstadísticaInferencialDLL.
- ∅ Un programador de aplicaciones cliente no puede tener acceso al funcionamiento interno de los objetos de EstadísticaInferencialDLL o Hoelest.exe.

- ∅ Debido a la compatibilidad binaria y desarrollo multiplataforma, las aplicaciones cliente y los servidores compatibles con COM son independientes del lenguaje de desarrollo y del sistema operativo, en otras palabras, un programador de Visual C++ puede utilizar la funcionalidad de análisis de datos que proporciona EstadísticaInferencialDLL, que es un componente desarrollado en Visual Basic.
- ∅ Actualización más sencilla debido al control de versiones. Nunca se pierde funcionalidad cuando se actualizan los componentes; siempre se mejora o se amplía. Esto significa que es posible crear una nueva versión de EstadísticaInferencialDLL y aquellas aplicaciones cliente, como Hoelest.exe, que utilizaban la versión anterior no generarán problemas.
- ∅ La comunicación entre el cliente Hoelest.exe y el servidor EstadísticaInferencialDLL proporciona el intercambio de información más eficiente. Como el cliente y el componente servidor comparten el mismo espacio de direcciones, las llamadas entre ellos se hacen directamente.
- ∅ Interfaces claramente definidas. A la funcionalidad implementada en un componente de código ActiveX se tiene acceso mediante propiedades, métodos y eventos. Estas interfaces se documentan a sí mismas en cuanto a que el componente de código las expone automáticamente.
- ∅ El modelo de objetos de los componentes EstadísticaInferencialDLL y Hoelest.exe define una relación entre los objetos de cada componente. Al definir las relaciones entre los objetos utilizados en el programa, un modelo de objetos organiza los objetos de forma que sea más fácil programar.

REFERENCIAS

- a) A. Moreno. "Hoja Electrónica Estadística" (Tesis, Instituto de Ciencias Matemáticas, Escuela Superior Politécnica del Litoral, 2000)
- b) MICROSOFT CORPORATION. Mastering Visual Basic 5 (versión 1.0. Año 1997)
- c) MICROSOFT CORPORATION. Libros en Pantalla de Visual Basic (versión 1.0. Año 1996)
- d) Joyanes Aguilar Luis, Programación Orientada a objetos (Editorial McGraw-Hill. 1996)
- e) Manzano Vicente, Inferencia Estadística. Aplicaciones con SPSS/PC+. (Editorial RA-MA. Madrid 1995)
- f) Mendehall William, Wackerly Dennis, Scheaffer Richard, Estadística Matemática con aplicaciones, (Segunda Edición. Grupo Editorial Iberoamérica. México 1994)