

UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES.
INSTITUTO DE ESTADÍSTICA APLICADA Y COMPUTACIÓN

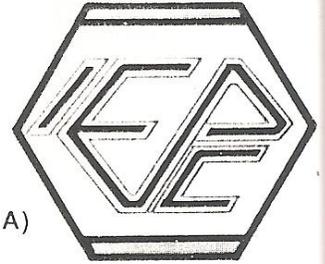
BASE DE DATOS DE LA ECONOMÍA VENEZOLANA.

POR: ING. ANGEL A. ZAMBRANO M.
TUTOR: PROF. GIANPAOLO ORLANDONI

Proyecto de grado presentado ante la ilustre Universidad de Los Andes
como requisito final para optar al título de
Magister Scientae en Estadística Aplicada

Mérida-Venezuela. Octubre 2000.

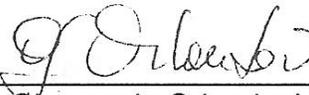
[Escribir texto]



ACTA

Quienes suscribimos, Miembros del Jurado designado por el Consejo Técnico según oficio No: 99-030, de fecha 1ro. de Marzo de 1999, para calificar el Trabajo de Grado titulado: "**BASE DE DATOS DE LA ECONOMIA VENEZOLANA**", presentado por el alumno: ANGEL ALBERTO ZAMBRANO MORALES, como credencial para optar al grado de: "**MAGISTER SCIENTIAE EN ESTADISTICA APLICADA**", hacemos constar por medio de la presente Acta, que con fecha 31 de Octubre de 2000, nos reunimos para oír la exposición de dicho Trabajo de Grado por parte de su autor. El alumno expuso ante el Jurado el contenido del Trabajo y respondió a las preguntas que se le formularon, después de lo cual, el Jurado decidió por unanimidad, **APROBAR** el Trabajo en cuestión, por cuanto fueron satisfechos todos los requisitos.

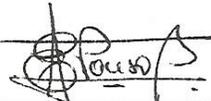
Prof.


Giampaolo Orlandoni Merli
Tutor

Prof.


Elizabeth Torres Rivas
Miembro del Jurado

Prof.


Ernesto Ponsot Balaguer
Miembro del Jurado

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a las personas que de una u otra manera contribuyeron a la finalización de éste trabajo:

- A mi tutor, profesor Giampaolo Orlandoni, por su rol de orientador y motivador.
- A los profesores de la maestría, por los conocimientos aportados en las respectivas materias.
- Al profesor Asdrubal Baptista, por haberme facilitado y autorizado el uso de los datos.
- Al profesor Ernesto Ponsot, por su tiempo dedicado en la revisión inicial del modelo.
- Al profesor Rafael Borges, por permitir el uso del S-PLUS.

DEDICATORIA

Esta conquista quiero dedicarla
a Oricia y a mis hijos Oriana y
Angel

RESUMEN

En éste trabajo se construyó una versión digitalizada de los datos (series) contenidos en el libro “Bases Cuantitativas de la Economía Venezolana. 1830-1989” del profesor Asdrubal Baptista, mediante el diseño y construcción de una base de datos en MS Access y el desarrollo de un programa en Visual Basic que manipula la misma. Además de la manipulación de las series (consulta) se implementó la aplicación de modelos ARIMA usando Intercambio Dinámico de Datos con el S-PLUS.

Las series pueden ser consultadas por Internet en la siguiente dirección:

<http://hotelling.faces.ula.ve/bases>

Palabras clave: Base de datos, ARIMA, Economía Venezolana, MS Access, Visual Basic

INDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA.....	iii
RESUMEN	iv
INDICE GENERAL.....	v
INDICE DE TABLAS.....	vi
INDICE DE FIGURAS	vi
Introducción	1
1. Las Bases Cuantitativas de la Economía Venezolana	3
2. Conceptos básicos.....	10
2.1. Datos y Base de Datos.....	10
2.2. Diseño de una base de datos.....	11
2.3. Lenguaje Estructurado de Consultas (SQL).....	18
2.4. Dynamic Data Exchange.....	20
2.5. Internet y el WWW.....	24
2.6. Modelos ARIMA	33
3. Análisis y diseño de la base de datos de la Economía Venezolana.....	40
3.1. Modelo Entidad-Relación.....	40
3.2. Resumen descriptivo del modelo de datos.....	42
3.3. Modelo Relacional de la Base de datos	46
3.4. Implementación de la Base de Datos.....	48
3.5. Instalación de Bases Cuantitativas.....	52
3.6. ¿Cómo ejecutar Bases Cuantitativas?	53
3.7. Bases cuantitativas en la Web.....	65
CONCLUSIONES.....	69
BIBLIOGRAFIA	70
ANEXOS.....	1

INDICE DE TABLAS

Tabla 1: Tópicos soportados por S-PLUS para las conversaciones DDE.....	21
---	----

INDICE DE FIGURAS

Fig. 1 Partes de un documento HTML.....	29
Figura 2. Diagrama Entidad-Relación de la base de datos.....	41
Figura 3. Modelo relacional de la base de datos.....	48
Figura 4. Ventana principal de la aplicación que permite ampliar las Bases Cuantitativas.....	49
Figura 5. Formulario para dar entrada a los datos de una serie.....	50
Figura 6 Cómo ejecutar Bases Cuantitativas.....	52
Figura 7. Pantalla de presentación y bienvenida.....	53
Figura 8. Ventana principal de Bases Cuantitativas.....	54
Figura 9. Ventana de la opción Ver Cuadros.....	55
Figura 10. Ventana Salir del programa.....	56
Figura 11. Cuadrícula con las diferentes series de una tabla.....	57
Figura 12. Cuadro de diálogo para Almacenar Cuadro.....	58
Figura 13. Mensaje de fin de Almacenar.....	58
Figura 14. Ventana Ver Bases Cuantitativas por Series.....	59
Figura 15. Caja de diálogo para introducir la serie a buscar.....	60
Figura 16. Caja de texto para solicitar el nuevo criterio de recorrido de las series.....	61
Figura 17. Cuadro de diálogo para introducir el nombre de la serie a almacenar.....	61
Figura 18. Ejemplo de gráfico de dispersión de una serie.....	62
Figura 19. Ventana Análisis ARIMA.....	63
Figura 20. Resultado de la corrida del modelo ARIMA a los datos de una serie.....	64
Figura 21. Mostrar Cuadros en el navegador WEB.....	66
Figura 22. Parte superior de la pantalla Mostrar Series en el navegador WEB.....	67
Figura 23. Parte final de la pantalla Mostrar Series en el navegador WEB.....	67
Figura 24. Consulta por Series en el navegador WEB.....	68

Introducción

Cualquier profesional dedicado a la investigación, la docencia o al ejercicio propio de su profesión requiere el uso de datos para cumplir con sus actividades. Estudiantes de todos los niveles y carreras deben manejar datos a lo largo de sus estudios. En general, la población de un país, región, ciudad o comunidad, al momento de proceder a la interpretación de la realidad circundante y establecer conclusiones que puedan de alguna forma influir en el funcionamiento de la misma para resolver los problemas que le aquejan, de una u otra forma, necesita el conocimiento y uso de datos.

Asdrúbal Baptista en su libro “Bases Cuantitativas de la Economía Venezolana: 1830-1989” cuantifica toda una serie de datos económicos de Venezuela, algunos de ellos desde 1830 hasta 1989, acerca de tópicos como la población, la actividad económica, el petróleo, la renta nacional, etc. Estos datos son útiles para cualquier estudioso de la economía venezolana, tal es el caso de los investigadores económicos, profesores y estudiantes de ciencias económicas y sociales, profesionales de la economía y la estadística. También son de gran utilidad para aquellas personas responsables de llevar las políticas económicas del país.

Con el desarrollo tecnológico del manejo de la información, los datos se pueden tener disponibles en forma rápida mediante el uso de técnicas de diseño de Bases de Datos. Una base de datos es una colección de datos relacionados entre si y organizados para un uso y fin determinado, la cual permite almacenar y recuperar datos en forma rápida y eficiente.

El objetivo principal del presente trabajo es la implementación de una versión digital (computarizada) de las “Bases Cuantitativas de la Economía Venezolana” de Baptista, mediante el diseño de una base de datos para el almacenamiento seguro de los datos y la recuperación (consulta) rápida y efectiva de los mismos en forma de cuadros y series económicas. Como objetivos secundarios del

trabajo se tiene la consulta de las tablas y series en una página web (INTERNET) y la aplicación de los modelos ARIMA a las series que se consultan.

En el diseño de una base de datos lo primero que hay que realizar, y entender, es el estudio de la organización subyacente, para de esta forma conocer los objetos que intervienen en la misma. A continuación, es menester identificar las relaciones entre los diferentes objetos (modelo Entidad-Relación) y por ultimo definir las tablas que intervienen en el modelo (modelo relacional de la base de datos).

La implementación de la base de datos para el almacenamiento de las tablas se realizó mediante el MS ACCESS 7.0, y la manipulación de las tablas y series del libro se programó en Visual Basic 5.0 que es un lenguaje de uso general, orientado a objetos y controlado por eventos.

La aplicación de los modelos ARIMA se realizó mediante el uso del Intercambio Dinámico de datos (Dynamic Data Enchanged – DDE) entre el Visual Basic y el S-plus.

En la primera parte del trabajo se hace una descripción del libro y su organización, necesaria para extraer los objetos implicados en el diseño. En la segunda parte se presentan algunos conceptos básicos relacionados con Bases de Datos y con las técnicas a utilizar en el desarrollo del trabajo (DDE y ARIMA). Después se plantea el modelo lógico y el respectivo modelo relacional de la Base de Datos. Por último se describe el programa y la forma de uso del mismo.

1. Las Bases Cuantitativas de la Economía Venezolana.

Asdrúbal Baptista con el libro “Bases Cuantitativas de la Economía Venezolana” intenta, según sus propias palabras, satisfacer, en lo fundamental, ciertas exigencias cuantitativas que impone el conocimiento de la Economía Política de Venezuela. Dichas exigencias provienen de un doble origen, a saber: de peculiaridades de la estructura económica venezolana, cuyo carácter y significación, como es del todo natural sólo pueden apreciarse a través de un desenvolvimiento que abarca prolongadas dimensiones temporales; y de la necesidad de disponer de evidencias que presten sustento cuantitativo a las concepciones teóricas presentes en ese conocimiento. [1]

El conocimiento de la Economía Política, dice el autor, evoca varias reacciones:

- Es un conocimiento al que sólo cabe ver como un antecedente de lo que los contemporáneos entienden como el verdadero conocimiento económico.
- En el mejor de los casos se lo ve como un precursor de las disciplinas económicas que hoy prevalecen.
- Otros lo contemplan como un afán, menos que precientífico que el recto método económico ha colocado en su lugar con el paso de los años.
- Y hay otros para quienes la denominación misma del conocimiento es útil en la delimitación de un ámbito de estudio que tiene por exclusivo objeto la actividad económica del Estado.

La Economía es Política sólo en lo que concierne a esa actividad circunstancial del Estado, y fuera de ella tal calificación es innecesaria o equivocada. Es decir, cuando se habla de Economía Política es porque está de por medio la presencia económica del Estado, mientras que las relaciones económicas entre los hombres son materia simplemente del término Economía.

A la Economía Política le concierne un sustrato esencial de la sociedad, que a su vez es el espacio natural de la política, el cual es el ámbito de asociación de los hombres con ocasión de la subsistencia. Por ser ese su objeto, y por ser ella una ciencia social, pareciera que el método científico como camino metodológico de estudio pudiera resultar en un enorme equívoco. De hecho, la alusión al método científico refiere de suyo al estudio del ámbito de la naturaleza física, y positivamente cabe decir que el traslado al campo de lo económico del método de indagación propio de otras áreas del conocimiento, con objetos distintos de lo social, puede resultar infértil.

Y bien, al momento de definir lo social para los fines de su indagación científica, ¿a cuál rasgo dominante, fuera del tiempo histórico, debe prestarse especial atención? Si no se prestara esa atención debida, como es el caso de mucho de lo que contemporáneamente pasa por el análisis de la realidad económica, la consecuencia fatal sería la inanidad teórica. Pero si se obedeciera, como es de esperar, la imposición que sobre el investigador fuerza la propia realidad de las cosas, entonces surge de inmediato otra pregunta, que es no menos fundamental: ¿es acaso la naturaleza del tiempo histórico acorde con lo propio de la naturaleza de la matemática?

En efecto, la consustanciación de la matemática con lo puramente espacial se acompaña, al unísono, con su esencial atemporalidad y, por lo tanto, con su radical incapacidad para atender lo específico de lo histórico. No basta con fechar los acontecimientos económicos, o con introducir retardos o adelantos temporales explícitos en las relaciones causales a través del uso de ecuaciones diferenciales o en diferencias finitas. La cuestión que yace de por medio es de un carácter esencialmente distinto, a saber, que el cálculo numérico no puede captar lo fundamental de la diferencia entre lo pasado y lo porvenir dentro del ámbito de la existencia humana.

El método de la Economía Política, concluye el autor, tiene por misión establecer las pautas críticas relativas a la naturaleza de las teorías propias de la disciplina, a los medios e instrumentos de indagación, así como a las evidencias que habrán de prestar el sustento empírico propio de lo científico en el terreno social. Una de esas evidencias, de gran significación empírica, es la cuantitativa. Y es que no hay duda de que algunos hechos económicos fundamentales ya se presentan a la observación como cantidades hechas número por la misma realidad social. De modo que la tarea de la investigación científica, en el ámbito de la Economía Política, requiere de la evidencia que brindan los números, dentro de los términos sugeridos, en cuanto ellos son aptos para indicar la ocurrencia o no del hecho considerado, y más, en cuanto brindan sustento al juicio sobre las tendencias implícitas en el movimiento del que es expresión la ocurrencia -o no ocurrencia- en cuestión. [1]

En el libro el autor presenta una serie de estas evidencias o datos cuantitativos de la historia económica y social venezolana, algunas desde 1830, pero la gran mayoría desde 1920, época del advenimiento del petróleo en la vida económica del país. Estas evidencias las presenta en forma de cuadros, los cuales están divididos en seis capítulos atendiendo el ámbito propio de cada conjunto de evidencias. Cada capítulo lo ordena de la siguiente manera:

- En primer lugar los cuadros estadísticos fundamentales.
- Luego la descripción de los métodos de trabajo utilizados en cada circunstancia para definir y acopiar la información ofrecida.
- En tercer lugar los cuadros estadísticos relativos a la información originaria de la cual se desprenden los cuadros señalados antes (apéndices). Y por último, las fuentes documentales o bibliográficas de cada cuadro.

A continuación se describirá el contenido de cada uno de los seis capítulos que comprende el libro:

- El primero enumerado **I**, se titula “La población” y contiene cuadros relacionados con la Población total (Cuadro I-1), Población urbana y rural (Cuadro I-2), así como también el movimiento demográfico venezolano desde el año 1830 hasta el año 1989.
- El segundo, **II**, titulado “La actividad Económica valorada a los precios de 1968”, contiene cuadros relacionados con la economía venezolana desde el año 1920 (algunos desde el año 1830) hasta 1989. En estos cuadros se presenta el Índice general de la actividad económica tomando como base el año 68 (Cuadro II-1), la actividad económica por sectores (Cuadro II-2), Composición sectorial del Producto Interno Bruto (Cuadro II-3), Gasto dedicado al PIB (Cuadros II-4 y 5), Producto Interno Neto (Cuadros II-6 y 7). En los cuadros secundarios se presentan datos de producción agrícola (Cuadro B-1, 10 y 11), Contribución de los diferentes sectores a la actividad económica (Cuadros B-2 al 4), Exportaciones (Cuadro B-5), Importaciones (Cuadro B-6), Gasto público (Cuadro B-7), El comercio exterior (Cuadros B-8 y 9), Petróleo (Cuadro B-12), Índices de producción (Cuadros B-13 al 18), La construcción (Cuadro B-19), El transporte y las comunicaciones (Cuadro B-20), La electricidad (Cuadro B-21), PIB del servicio de inmuebles (Cuadro B-2), PIB de servicios de la administración pública (Cuadro B-23), Consumo (Cuadros B-24 al 34) y las tasas de cambio (Cuadros B-35 y 36).
- El tercero, **III**, denominado “La actividad Económica valorada a los precios de 1984. La renta petrolera y el PIB no rentístico”. Se presenta el sistema de cuentas nacionales tomando como año base a 1984, que fue el que tomó el sistema de estadísticas de Venezuela con el fin de presentar ciertas magnitudes a precios constantes. Los cuadros de este capítulo son los siguientes: índice general de la actividad económica con año base 1984 (Cuadro III-1), Sectores del PIB no rentístico (Cuadro III-2 y 7), Gasto dedicado

al PIB no rentístico (Cuadro III-3, 4 y 8), Producto Interno Neto no rentístico (Cuadro III-5 y 6), y en los cuadros secundarios cuentas nacionales de EEUU (Cuadro C-1), PIB de Venezuela (Cuadro C-2), Gastos dedicados al PIB de Venezuela (Cuadro C-3), Rentabilidad del capital petrolero (Cuadro C-4), Información para el cálculo del PIB petrolero no rentístico (Cuadro C-5), índices del PIB (Cuadro C-6), Índices del consumo privado (Cuadro C-7) y La renta petrolera (Cuadro C-8).

- El cuarto, **IV**, llamado “Acumulación de capital”. Presenta los diversos agentes de inversión que se han hecho en el país, tanto públicos como privados. Contiene los siguientes cuadros: Inversión bruta total (Cuadros IV-1 al 3), Inversión Bruta por agentes económicos (Cuadros IV-4 al 6), Inversión bruta del sector privado (Cuadros IV-7 al 9), Inversión bruta del sector público (Cuadros IV-10 al 12), Inversión bruta del sector petrolero (Cuadros IV-13 al 15), Acervo bruto total de capital (Cuadros IV-16 al 18), Acervo neto total de capital (Cuadros IV-19 al 21), Acervo bruto por agentes económicos (Cuadros IV-22 al 24), Acervo neto por agentes económicos (Cuadros IV-25 al 27), Acervo bruto del sector público (Cuadros 28-16 al 30), Acervo neto del sector público (Cuadros IV-31 al 33), Acervo bruto del sector privado (Cuadros IV-34 al 36), Acervo neto del sector público (Cuadros IV-37 al 39), Acervo bruto del sector petrolero (Cuadros IV-40 al 42), Acervo neto del sector petrolero (Cuadros IV-43 al 45), Consumo de capital fijo (Cuadros IV-46 al 48), Depreciación (Cuadros IV-49 al 51) y los siguientes cuadros secundarios: Inversión pública en estructuras (Cuadros D-1 y 2), Inversión en maquinarias y equipos de transporte (Cuadro D-3), Producción nacional de maquinarias (Cuadro D-4), Importación de maquinarias (Cuadro D-5), Producción de vehículos nacionales (Cuadro D-6), Importación de equipos de transporte (Cuadro D-7), Composición del parque automotor venezolano (Cuadro D-8), Información para la estimación de la inversión en estructuras residenciales (Cuadro D-9) e Importaciones petroleras (Cuadro D-10).

- El quinto, **V**, titulado “Empleo, Ingreso Nacional y Distribución del Ingreso”. Presenta cuadros acerca de la fuerza laboral del país y del ingreso nacional. Contiene los siguientes cuadros: Población económicamente activa, empleada, desempleada y empleada asalariada (Cuadros V-1 al 3), Empleo del gobierno, petróleo y manufactura (Cuadro V-4), Ingreso nacional (Cuadro V-5), Distribución del ingreso nacional (Cuadro V-6), Remuneración mensual por trabajador (Cuadro V-7), Remuneración mensual por obrero en la construcción (Cuadro V-8), distribución del ingreso familiar en percentiles (Cuadro V-9 y 10), Medidas de la desigualdad en la distribución del ingreso (Cuadro V-11 y 12) y los cuadros anexos Distribución del ingreso personal en 1944, 1957, 1962, 1970, 1975 a 1989 (Cuadros E-1 a 34).
- El último, **VI**, denominado “Precios y Dinero”. Se refiere a cuadros con el índice general de precios y liquidez monetaria. Los cuadros principales de este capítulo son los siguientes: Índice de precios en Venezuela (Cuadros VI-1 al 6), deflatores de los sectores del PIB (Cuadros VI-7 al 14) y Liquidez monetaria (Cuadros VI-15 y 16) y los secundarios son: Índice de precios de la exportaciones (Cuadros F-1, 12 y 13), Índice de precios de las importaciones (Cuadros F-2 y 3), Índice de precios al consumidor (Cuadro F-4 al 6), Índice de precios de la inversión (Cuadro F-7 al 11 y 14).

Cada uno de estos cuadros o tablas contiene una o más series temporales referidas a un tópico particular del tema a que se refiere el cuadro. Por ejemplo el cuadro I-2 del libro denominado “Población urbana y rural: 1920-1989” contiene dos series: “Urbana” y “Rural”, con periodo anual que comienza en 1920 y termina en 1989. (ver anexo 1).

Cada cuadro esta organizada de la siguiente forma:

Título de la tabla. (Año comienzo - Año final)

Unidad de los datos

Período - Año	Título serie1	Título serie2	Título serie3
Valor periodo	Valor serie	valor serie	valor serie

Una serie se refiere a un conjunto de valores o datos que representan una variable desde el período inicial hasta el período final, con una unidad de medida y por un período de tiempo.

En resumen el libro está formado por capítulos, cada capítulo tiene varios cuadros (principales y anexos) y cada cuadro esta formado por una o más series.

2. Conceptos básicos.

2.1. Datos y Base de Datos.

Por lo general los términos datos e información se manejan como sinónimos, sin embargo se emplea “datos” cuando se refiere a los valores almacenados en realidad e “información” cuando se refiere al significado de esos valores desde el punto de vista de algún usuario.

Un sistema de Bases de Datos es básicamente un sistema para archivar en el computador, es decir, un sistema computarizado cuyo propósito general es mantener información y hacer que esté disponible cuando se solicite [6]. La información puede ser cualquier cosa que se considere importante para el individuo o la organización a la cual debe servir el sistema.

Conviene llamar persistentes a los datos de una base de datos; esto tiene por objeto sugerir que la información de una base de datos difiere de otros tipos de datos más efímeros, como son los datos de entrada y salida, los resultados intermedios y, en general cualquier información cuya naturaleza sea hasta cierto punto transitoria. Por supuesto, la distinción entre datos persistentes y datos transitorios no es rígida y nítida, sino que depende hasta cierto punto del contexto.

Una base de datos está constituida por cierto conjunto de datos persistentes utilizado por los sistemas de aplicaciones de una empresa determinada. El término empresa se refiere a cualquier organización comercial, científica, técnica o de otro tipo con un grado razonable de autosuficiencia.

Los sistemas de base de datos tienen las siguientes ventajas sobre los métodos tradicionales de mantener registros en papel:

- Son compacto.
- Son rápido.

- Son menos laborioso.
- Son actuales.
- Ofrece un control centralizado de la información.

2.2. Diseño de una base de datos.

En el diseño de una Base de Datos es importante en primer lugar indicar cómo funciona la misma a nivel lógico, para lo cual hay que crear el modelo de datos, es decir, la estructura lógica de la base de datos. Esto se realiza mediante una colección de herramientas conceptuales para describir datos. Los diferentes modelos de datos que se han propuesto se clasifican en tres grupos diferentes: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos [3].

2.2.1. Modelos lógicos basados en objetos

Los modelos lógicos basados en objetos se usan para describir datos en los niveles lógico y de vistas. Se caracterizan por el hecho de que proporcionan capacidades estructurales muy flexibles y permiten que las ligaduras de datos sean especificadas explícitamente. Hay modelos muy diferentes, y probablemente aparezcan más. Varios de los más ampliamente conocidos son:

- El modelo de entidad-relación.
- El modelo orientado a objetos.
- El modelo de datos semántico.
- El modelo de datos funcional.

El modelo entidad-relación ha conseguido gran aceptación en el diseño de bases de datos y se usa ampliamente en la práctica. El modelo orientado a objetos incluye muchos de los conceptos del modelo entidad-relación, pero representa

tanto código ejecutable como datos, está ganando aceptación rápidamente en la práctica. A continuación se dará una breve descripción de ambos modelos.

2.2.1.1. Modelo Entidad-Relación

El modelo Entidad – Relación es una de estas herramientas, está basado en una percepción del mundo real y describe de forma sencilla los datos como un conjunto de entidades y relaciones existentes entre ellas.

El modelo Entidad-Relación es una técnica para definir las necesidades de información en la organización. Proporciona una buena base para sistemas de alta calidad dirigidos a satisfacer las necesidades de la organización.

El modelo Entidad-Relación en su forma más simple implica identificar los objetos o asuntos de importancia dentro de la organización (entidades), las propiedades de estos objetos (atributos) y cómo se relacionan entre sí (relación). Los objetivos primordiales de este modelo son:

- Proporcionar un modelo preciso de las necesidades de información de la organización.
- Proporcionar un modelo independiente de cualquier almacenamiento de datos y método de acceso.

Las entidades son los objetos o cosas, tangibles o intangibles, que por sus características propias y únicas pueden distinguirse de otros objetos. Es esencial establecer las entidades que fluyen en el sistema para describirlas, es decir, para desglosarlas en término de los atributos o variables que las identifican. Por ejemplo, una de las series del libro puede considerarse una entidad, o un cuadro.

Los atributos son las características o detalles que sirven para calificar, identificar, clasificar o expresar el estado de una entidad, es decir, cualquier descripción del objeto, y a su vez poseen un rango o dominio que limita su espacio de acción.

Cada serie de datos posee características propias como por ejemplo: el título, el año de comienzo, la unidad de medida de los datos, etc.

Una entidad débil es una entidad que no es del todo independiente, es decir, tiene uno de sus atributos relacionado con una entidad en el modelo.

Una relación es una asociación significativa nombrada entre dos o más entidades. Es una entidad especial que relaciona dos o más entidades. En nuestro sistema hay una relación de pertenencia entre las series y los cuadros de tal manera que uno dice que una serie pertenece a un cuadro o, que los cuadros están formados por series; también hay relación entre los capítulos y los cuadros donde podemos decir que los capítulos están formados por cuadros.

El conjunto de todas las entidades del mismo tipo se denomina “conjunto de entidades” y el de relaciones se llama “conjunto de relaciones”. De esta forma podemos decir por ejemplo el conjunto de entidades series.

Además de las entidades (con sus atributos) y las relaciones, el modelo E-R debe también expresar la correspondencia de cardinalidades, que muestra el número de entidades con las que una entidad se puede asociar a través de un conjunto de relaciones, puede ser: uno a uno, uno a varios, varios a uno y varios a varios. La relación existente entre las series y los cuadros es muchos a uno, porque muchas series se relacionan con o pertenecen a un cuadro mientras que un único cuadro se relaciona con cada serie; entre los capítulos y los cuadros la cardinalidad es de uno a muchos porque un capítulo se relaciona con muchos cuadros.

La estructura lógica de la base de datos se expresa mediante un diagrama Entidad-Relación, que consta de:

- Rectángulos, que representan un conjunto de entidades.
- Elipses, que representan los atributos de las entidades y relaciones.
- Rombos, que representan relaciones entre conjunto de entidades.

- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con los conjuntos de relaciones.

Cada componente (rectángulo, elipse o rombo) se marca con la respectiva entidad, atributo o relación que representa.

Conceptualmente, las entidades y relaciones individuales son distintas, la diferencia entre ellas se debe expresar en términos de sus atributos. Para hacer tales distinciones se utiliza el concepto de clave.

Una superclave es un conjunto de atributos que permite identificar de forma única a una entidad dentro del conjunto de entidades. Una superclave podría contener atributos innecesarios. Por lo general interesan aquellas superclaves tales que los subconjuntos de atributos de ellas no sean superclaves, a estas se les denominan claves candidatas.

Se usa el término de clave primaria para denotar aquella clave candidata que se usa como elemento principal para identificar las entidades dentro de un conjunto de entidades.

2.2.1.2. Modelo orientado a objetos

Como el modelo E-R, el modelo orientado a objetos está basado en una colección de objetos. Un objeto contiene valores almacenados en variables de ejemplares (instance variables) dentro de ese objeto. Un objeto también contiene fragmentos de código que operan en el objeto. Estos fragmentos de código se llaman métodos.

Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan juntos en clases. Una clase se puede ver como una definición de tipo para los objetos. Esta combinación de datos y métodos constituyendo una definición de tipo es similar a un tipo abstracto de datos en un lenguaje de programación.

La única manera de que un objeto pueda acceder a los datos de otro objeto es mediante la invocación de un método de ese otro objeto. Esta acción se llama paso de mensaje al otro objeto. Así, la interfaz de llamada de los métodos de un objeto define la parte visible externamente del objeto. La parte interna del objeto —las variables de ejemplares y el código de los métodos— no es visible externamente. El resultado es obtener dos niveles de abstracción de datos.

Al contrario que las entidades en el modelo E-R, cada objeto tiene su propia identidad única, independientemente de los valores que contenga. Así, dos objetos que contienen los mismos valores son completamente diferentes. La distinción entre objetos individuales se mantiene en el nivel físico a través de la asignación de diferentes identificadores a los objetos.

2.2.2. Modelos lógicos basados en registros

Los modelos lógicos basados en registros se usan para describir datos en los niveles lógico y de vistas. En contraste con los modelos de datos basados en objetos, se usan tanto para especificar la estructura lógica completa de la base de datos como para proporcionar una descripción de alto nivel de la implementación.

Los modelos basados en registros se llaman así debido a que la base de datos se estructura en registros de formato fijo de diferentes tipos. En cada tipo de registro se define un número fijo de campos o atributos, y cada campo tiene normalmente una longitud fija. El uso de registros de longitud fija simplifica la implementación en el nivel físico de la base de datos. Esta simplicidad contrasta con muchos de los modelos basados en objetos, cuya estructura más rica a menudo conduce a registros de longitud variable en el nivel físico.

Los tres modelos basados en registros más ampliamente aceptados son el modelo relacional, el modelo de red y el modelo jerárquico. El modelo relacional, que ha prevalecido sobre los otros dos en los últimos años, se examinará con detalle. Los modelos de red y jerárquico, todavía usados en gran número de bases de datos más antiguas, sólo se describirán.

2.2.2.1. Modelo relacional

En el modelo relacional se usa una colección de tablas para representar tanto los datos como las relaciones entre esos datos. Cada tabla tiene varias columnas, y cada columna tiene un nombre único.

Este modelo está basado en tablas que muestran los registros con sus respectivos valores para cada campo, permite representar tanto los datos como las relaciones existentes entre estos.

Su principal ventaja es que las tablas representan un concepto simple e intuitivo para visualizar la estructura lógica de la base de datos, corrigiendo omisiones que mejoren el diseño original. Las tablas tienen una estructura parecida al modelo Entidad-Relación, sólo que aquí se especifica más a fondo el contenido de cada uno de los atributos dejando de lado las relaciones direccionales entre tablas.

Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional.

La siguiente notación se usará para representar las entidades, entidades débiles y relaciones en el modelo relacional:

Nombre de la entidad		
Campo1	...	CampoN

Nombre de la entidad débil		
Campo1	...	CampoN

Nombre de la relación		
Campo1	...	CampoN

2.2.2.2. Modelo de red

Los datos en el modelo de red se representan mediante colecciones de registros (en el sentido de Pascal) y las relaciones entre los datos se representan mediante enlaces, que se pueden ver como punteros. Los registros en la base de datos se organizan como colecciones de grafos dirigidos.

2.2.2.3. Modelo jerárquico

El modelo jerárquico es similar al modelo de redes, en el sentido en que los datos y las relaciones entre los datos se representan mediante registros y enlaces, respectivamente. Éste se diferencia del modelo de redes en que los registros se organizan como colecciones de árboles en lugar de grafos dirigidos.

2.2.2.4. Diferencias entre los modelos

El modelo relacional se diferencia de los modelos de redes y jerárquico en que no usa punteros o enlaces. En su lugar, el modelo relacional relaciona registros mediante los valores que ellos contienen. Esta liberación del uso de punteros permite que se defina mediante un fundamento matemático formal.

2.3. Lenguaje Estructurado de Consultas (SQL)

Un lenguaje de consulta es un lenguaje en el que un usuario solicita información de la base de datos. Estos lenguajes suelen ser de un nivel superior que el de los lenguajes de programación habituales. Los lenguajes de consulta pueden clasificarse como procedimentales o no procedimentales. En los lenguajes procedimentales, el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos para calcular el resultado deseado. En los lenguajes no procedimentales, el usuario describe la información deseada sin dar un procedimiento concreto para obtener esa información [3].

La mayor parte de los sistemas comerciales de bases de datos relacionales ofrecen un lenguaje de consulta que incluye elementos de los enfoques procedimental y no procedimental.

Existen en el mercado varios lenguajes de consulta, a saber: SQL (Structured Query Language, Lenguaje estructurado de consultas), QBE (Query By Example, Consulta mediante ejemplos), Quel (lenguaje de consultas para el sistema de base de datos Ingres) y DATALOG (lenguaje de consultas no procedimental basado en el PROLOG).

Aunque el lenguaje SQL se considere un «lenguaje de consultas», contiene muchas otras capacidades además de la consulta en bases de datos. Incluye características para definir la estructura de los datos, para la modificación de los datos en la base de datos y para la especificación de ligaduras de seguridad.

SQL se ha establecido claramente como el lenguaje estándar de bases de datos relacionales. Hay numerosas versiones de SQL. La versión original se desarrolló en el Laboratorio de investigación de San José (San José Research Center). Este lenguaje, originalmente denominado Sequel, se implementó como parte del proyecto System R, a principios de 1970. El lenguaje Sequel ha evolucionado desde entonces y su nombre ha pasado a ser SQL. Actualmente, numerosos productos son compatibles con el lenguaje SQL.

2.3.1. Estructura básica

Una base de datos relacional consiste en un conjunto de relaciones, a cada una de las cuales se le asigna un nombre único. SQL permite el uso de valores nulos para indicar que el valor o bien es desconocido o no existe. Se fijan criterios que permiten al usuario especificar a qué atributos no se puede asignar valor nulo.

La estructura básica de una expresión SQL consiste en tres cláusulas: **select**, **from** y **where**.

- La cláusula **select** corresponde a la operación proyección del álgebra relacional. Se usa para listar los atributos deseados del resultado de una consulta.
- La cláusula **from** corresponde a la operación producto cartesiano del álgebra relacional. Lista las relaciones que deben ser analizadas en la evaluación de la expresión.
- La cláusula **where** corresponde al predicado selección del álgebra relacional. Es un predicado que engloba a los atributos de las relaciones que aparecen en la cláusula **from**.

Una consulta típica en SQL tiene la forma:

Select A1, A2, ..., An **from** r1, r2, ..., rm **where** P

Cada A_i representa un atributo, y cada r_i una relación. P es un predicado.

Si se omite la cláusula **where**, el predicado P es cierto.

SQL forma el producto cartesiano de las relaciones incluidas en la cláusula **from**, lleva a cabo la selección del álgebra relacional usando el predicado de la cláusula **where** y entonces proyecta el resultado sobre los atributos de la cláusula **select**. En la práctica, SQL puede convertir la expresión en una forma equivalente que puede ser procesada más eficientemente [3].

2.4. Dynamic Data Exchange

Las comunicaciones entre los programas pueden tener lugar de varias maneras. Es claramente posible pasar datos entre los programas teniendo un programa que escriba un archivo ASCII al disco y teniendo otro programa que lee ese archivo. Es también posible para los programas intercambiar datos usando el Portapapeles de Windows (Clipboard). En cada uno de estos métodos, el proceso es secuencial y normalmente requiere intervención humana para coordinar la acción.

DDE (Dynamic Data Exchange), por otro lado es la técnica de enviar y recibir datos entre las aplicaciones Windows mediante conversaciones.

El Intercambio de Datos Dinámicos (DDE) es un mecanismo apoyado por Microsoft Windows que permite que dos programas diferentes que corren bajo Windows se comuniquen. Esta comunicación puede tomar la forma de paso de datos de un programa hacia otro, o puede tomar la forma de un programa que pide al otro programa tomar una acción específica o ejecutar una tarea. Esta comunicación puede tener lugar bajo el control del programa (sin la intervención humana) y tan a menudo como se requiera [12].

En una conversación DDE, un programa inicializa la conversación y el otro programa le responde. El programa que comienza la conversación se llama el programa destino o el *cliente* y el que responde se llama el programa fuente o el *servidor*.

Hay ocasiones que los usuarios de S-PLUS pueden necesitar usar otros programas para manipular datos. Por ejemplo, la sección de contabilidad puede tener una hoja de cálculo que resume datos financieros para la compañía, y un analista puede pedir resumir esos datos. DDE habilita el llamado de funciones S-PLUS dentro de una hoja de cálculo u otro programa.

S-PLUS funciona como un servidor (aplicación fuente) de DDE, que suministra datos a otras aplicaciones. Nuestra aplicación (Bases Cuantitativas) le enviara

datos y ordenes al S-PLUS para que este ejecute el módulo arima.mle y una vez ejecutado le enviará los resultados que nuestra aplicación le solicite para mostrarlos.

Tabla 1: Tópicos soportados por S-PLUS para las conversaciones DDE.

Tópico	Descripción
System [System	El tópico System se usa para pedir información especial de S-PLUS. Esta información incluye los nombres de objetos de datos (como vectores y tablas de datos) que pueden usarse en conversaciones, el nombre del tópico de la conversación aceptada, y otra información.
[DataSheet]	El tópico [DataSheet] es un identificador usado para especificar que la conversación es acerca de un bloque de datos de un objeto de datos, y es opcional. Usted puede simplemente especificar el nombre del objeto de datos como el tópico de la conversación
Cualquier nombre de objeto de datos (tal como el nombre de una tabla de datos o vector)	Igual como el tópico [DataSheet] visto arriba
[Execute] SCommand	[EXECUTE] o SCommand identifica que la conversación contenga cadena de comandos de S-PLUS válidos a ser ejecutados por S-PLUS

DDE usa la metáfora de una conversación. Debido a que una aplicación puede tener múltiples conversaciones DDE abiertas al mismo tiempo, es necesario tener un identificador para cada conversación. Llamaremos este el número de canal. El número de canal se establece cuando una conversación DDE se inicializa.

Una aplicación cliente DDE abre una conversación con un servidor particular. Cada servidor tiene un nombre al que responderá. Nosotros llamaremos este el nombre del servidor. El nombre del servidor para la S-PLUS es "S-PLUS".

S-PLUS soporta conversaciones DDE en el formato de Microsoft CF_TEXT, el cual es simplemente una serie de caracteres de texto terminada por un carácter del nulo (null) y donde cada línea finaliza con el par de caracteres “retorno de carro-próxima línea” (CR-LF). Las transferencias binarias no son soportadas.

Cualquier aplicación que soporte las funciones de cliente DDE puede inicializar una conversación con el S-PLUS, como por ejemplo: Visual Basic, VBA, Visual C++, Excel, Access, Lotus, Power Building y otras.

Una aplicación creada con Visual Basic puede ser origen o destino de una conversación DDE, en nuestro caso la usaremos como aplicación destino ya que solicitará un servicio al S-Plus que se comportará como la aplicación fuente o suministradora de datos.

El intercambio dinámico de datos lo realiza el VB a través de los controles PictureBox, TextBox o Label (Caja de gráfico, caja de texto o etiqueta de texto).

Estos controles tienen 4 propiedades que se ajustan debidamente para enlazar nuestras aplicaciones con otras:

- LinkTopic, se utiliza para fijar el tema de la conversación, el formato es el siguiente: “Aplicación|tema”, donde aplicación es en nuestro caso S-PLUS y tema cualquiera de los vistos en la tabla 1.
- LinkItem, es el elemento de la conversación, por lo general se refiere a un control o a una celda de un objeto.
- LinkMode, se refiere al modo de realizar la comunicación: ninguna, manual, automática o notificada.

- LinkTimeOut, es el tiempo máximo que una aplicación va a esperar por la respuesta de la otra aplicación.

Una vez fijadas las propiedades de la comunicación se utiliza uno de los siguientes métodos para realizar la comunicación:

- LinkExecute, permite enviar una serie de comandos desde la aplicación; la aplicación de destino espera que la aplicación de origen realice la operación descrita en la cadena.
- LinkPoke, transfiere el contenido de un control Label, PictureBox o TextBox a la aplicación de origen de una conversación DDE, es decir, envía datos a la aplicación fuente.
- LinkRequest, pide a la aplicación de origen de una conversación DDE que actualice el contenido de un control Label, PictureBox o TextBox con el respectivo elemento del tema de conversación.

2.5. Internet y el WWW

2.5.1. Redes de Computadores

Una red de computadores consiste de dos o más computadoras conectadas por un canal de comunicación de manera tal que puedan compartir datos, programas y equipo periférico. Hay dos tipos básicos de redes: redes locales, que conectan equipos que están físicamente cerca (LAN, Local Area Network), y redes amplias, que cubren regiones geográficas mayores (WAN, Wide Area Network). [LCOM98]

2.5.2. Internet [13]

Internet nació en EE.UU. hace unos 30 años. Un proyecto militar llamado ARPANET pretendía poner en contacto una importante cantidad de ordenadores de las instalaciones del ejército de EE.UU. Este proyecto gastó mucho dinero y recursos en construir la red de ordenadores más grande en aquella época.

Al cabo del tiempo, a esta red se fueron añadiendo otras empresas. Así se logró que creciera por todo el territorio de EE.UU. Hará unos 10 años se conectaron las instituciones públicas como las Universidades y también algunas personas desde sus casas. Fue entonces cuando se empezó a extender Internet por los demás países del Mundo, abriendo un canal de comunicaciones entre Europa y EE.UU.

Internet crece a un ritmo vertiginoso. Constantemente se mejoran los canales de comunicación con el fin de aumentar la rapidez de envío y recepción de datos; se conectan nuevas máquinas a la red. Cada día que pasa se publican en la Red miles de documentos nuevos, y se conectan por primera vez miles de personas. Con relativa frecuencia aparecen nuevas posibilidades de uso de Internet, y constantemente se están creando nuevos términos para poder entenderse en este nuevo mundo que no para de crecer.

La red Internet es hoy en día la red de computadores más extensa del planeta, para ser más precisos, Internet es una red que enlaza centenares de miles de redes locales heterogéneas.

El nexo común de todos los sistemas que integran la red Internet es el uso de los protocolos de comunicaciones TCP/IP. Estos protocolos son la base de la infraestructura que permite el intercambio de información (datos, documentos, sonidos, vídeos, fotos, etc) de forma virtualmente independiente de los sistemas en que ésta se encuentra almacenada. Sobre la base de esta infraestructura de transporte de datos se han construido otros protocolos que permiten enviar correo electrónico (SMTP), establecer conexiones con máquinas remotas (Telnet), acceder a foros de discusión (NNTP), transmitir archivos (FTP), World Wide Web (WWW), Grupos de Noticias, las conversaciones en línea (Internet Relay Chat, IRC) y Servicios de Telefonía.

2.5.3. El World Wide Web [13]

La World Wide Web, o WWW como se suele abreviar, se inventó a finales de los 80 en el CERN, el Laboratorio de Física de Partículas más importante del Mundo. Se trata de un sistema de distribución de información tipo revista. En la Red quedan almacenadas lo que se llaman Páginas Web, que no son más que páginas de texto con gráficos o fotos. Aquellos que se conecten a Internet pueden pedir acceder a dichas páginas y acto seguido éstas aparecen en la pantalla de su ordenador. Este sistema de visualización de la información revolucionó el desarrollo de Internet. A partir de la invención de la WWW, muchas personas empezaron a conectarse a la Red desde sus domicilios, como mero entretenimiento. Internet recibió un gran impulso, hasta el punto de que hoy en día casi siempre que hablamos de Internet, nos referimos a la WWW.

Todas estas herramientas han permitido desde hace bastante tiempo el intercambio de información entre los usuarios de Internet.

2.5.3.1. Arquitectura del Web.

Internet funciona con la estrategia "Cliente/Servidor", lo que significa que en la Red hay computadores Servidores que dan una información concreta en el momento que se solicite, y por otro lado están los computadores que piden dicha información, los llamados Clientes.

El cliente Web es un programa con el que interacciona el usuario para solicitar a un servidor Web el envío de las páginas de información. Este programa se encarga de acceder al servidor y transferir las páginas siguiendo el protocolo HTTP (HyperText Transfer Protocol, protocolo de transferencia de hipertexto).

Las páginas que se reciben son documentos de texto escritos en lenguaje HTML. El cliente web debe interpretar dichos documentos para mostrárselos al usuario con el formato adecuado. Además cuando lo que se recibe no es un documento de texto, sino un objeto multimedia (vídeo, sonido, etc) no reconocido por él, deberá encargarse de activar una aplicación externa capaz de gestionarlo.

Algunos de los clientes Web disponibles en la actualidad son: NCSA Mosaic, IBM Web Explorer, Cello, Netscape Navigator y Microsoft Internet Explorer.

El servidor Web es un programa que está escuchando permanentemente las solicitudes de conexión mediante el protocolo HTTP. Su principio de funcionamiento es muy simple: si encuentra en su sistema de archivos el documento HTML solicitado por el cliente, lo envía y cierra la conexión; en caso contrario, se envía un código de error y libera la conexión.

El servidor se encarga también de gestionar los aspectos de seguridad, comprobando si el usuario tiene acceso a los documentos.

En el modelo cliente-servidor en que se basa el WWW lo que se transmite por la red en la mayor parte de los casos son documentos codificados en lenguaje HTML

(Hyper Text Mark-up Language, Lenguaje para realizar Hipertexto) e interpretados por los clientes Web. Estas páginas de información contienen enlaces hipertexto que proporcionan al usuario otros documentos u objetos multimedia.

Los enlaces pueden ser de dos tipos:

- Internos, conducen a otro punto del documento actual
- Externos, apuntan a otros objetos (documentos o archivos multimedia) situados en el mismo servidor de donde proviene la página actual o en otros servidores accesibles por la red.

De esta forma, se crea una estructura no jerárquica y compleja de relaciones entre los documentos de distintos servidores y es éste el motivo por el cual se ha bautizado al sistema de información con el nombre de Worl Wide Web (Telaraña a nivel mundial).

2.5.4. Hipertexto.

Hipertexto es texto que no está restringido a una forma lineal o plana. Cualquier palabra en hipertexto se puede especificar como un enlace o apuntador a otro documento bajado, posiblemente también, en hipertexto, donde se puede hallar más información sobre esa palabra. Usando diferentes métodos, dependiendo de la interfase utilizada, el lector puede acceder al segundo documento seleccionando la palabra. El segundo documento a su vez pudiera contener enlaces a otros documentos. Cada documento será presentado al lector según su solicitud mediante un “click” del ratón o presionando la tecla Enter cuando el cursor está sobre el enlace. [LCOM98]

2.5.5. El lenguaje HTML básico.

2.5.5.1. Generalidades

Los documentos HTML tienen un formato de texto plano (ASCII) y se pueden crear usando cualquier editor o procesador de textos. Para especificar la apariencia de

la información que se desea mostrar, se parte de la base de un documento ASCII con los contenidos de la información a mostrar, y posteriormente se da formato a los mismos usando códigos o marcas especiales que regulan su apariencia.

Estos códigos o marcas tienen un aspecto singular: están encerrados por los símbolos < y >, como en <TITLE> .

En la especificación HTML existe una notable cantidad de códigos, que normalmente se usan por parejas (códigos pareados), para delimitar la parte del documento HTML que se ve afectado por su acción. En el caso de los códigos pareados, el código que indica el final es el mismo que el código inicial; la diferencia radica en que en el código final se antepone una barra al cuerpo del código, como es el caso de </TITLE>.

Por ejemplo: Este texto aparece en- negrita

Sin embargo, no todos los códigos HTML son pareados; existen operaciones de formato que sólo tienen efecto en el lugar en el que se incluyen (o desde el lugar en que aparecen hasta el final del documento), por lo que sólo es preciso usar un único código. Es el caso del código de fin de párrafo <P> . En este caso, existe el código </P>, pero no tiene efecto, y sólo se ha incluido a fin de lograr una cierta uniformidad.

En cualquier caso es indiferente poner los códigos en mayúsculas o minúsculas, pero es una buena práctica ponerlos siempre en mayúsculas para una mayor legibilidad y claridad de los documentos.

Como se puede apreciar, una buena parte del conocimiento del lenguaje HTML se refiere al conocimiento de los diferentes códigos disponibles y su correspondiente utilidad. Existe una complicación adicional: sólo un pequeño conjunto de códigos son estándar y están universalmente reconocidos por todos los clientes Web. Es por ello que únicamente algunos códigos tienen un efecto visible en todos los

clientes Web. En cualquier caso, si un determinado cliente Web no reconoce un código, simplemente lo ignora.

2.5.5.2. Partes de un documento HTML

A fin de simplificar las cosas a los clientes Web, es muy conveniente indicar de forma explícita que nuestro documento es de tipo HTML. Para ello se deberán incluir los códigos `<HTML>` y `</HTML>` al principio y al final del archivo, respectivamente.

Si el cliente Web no reconoce el tipo de información que está recibiendo, considera que se trata de un archivo binario e indica al usuario si desea almacenarlo en disco.

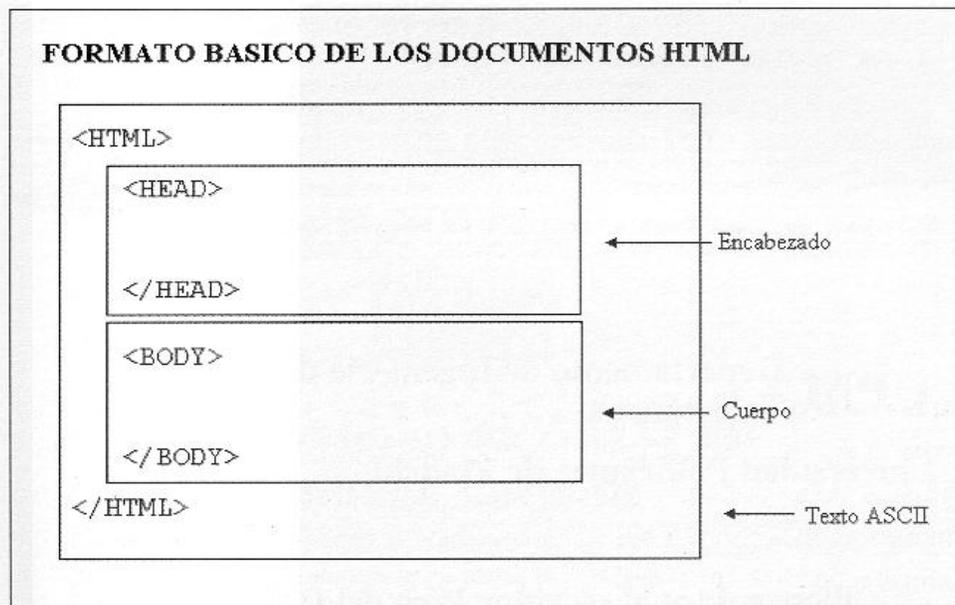


Fig. 1 Partes de un documento HTML

Todo documento HTML consta de dos partes fundamentales: la cabecera y el cuerpo. En la cabecera se incluye el título del documento, y en el cuerpo aparece el contenido, adecuadamente formateado.

La cabecera es todo lo que está delimitado por los códigos `<HEAD>` y `</HEAD>`.

En la cabecera se incluye información relativa al propio documento, como puede ser el título y en ocasiones, un índice. El título indicado sólo aparece en la barra de título de la ventana activa del cliente Web.

El título se indica usando los códigos pareados `<TITLE>` y `</TITLE>`, y se especifica usando un texto plano, sin códigos HTML de formato (no se puede poner en negrita o cursiva; los clientes Web ignoran estas especificaciones de formato).

Es muy aconsejable poner título a todos los documentos, y además, se deberá procurar que éste sea lo más descriptivo posible, puesto que si algún usuario decide incluir nuestra página en su lista de bookmarks, será el título lo que quede almacenado en dicha lista.

El cuerpo de un documento HTML se especifica usando los códigos pareados `<BODY>` y `</BODY>`. Es dentro de este cuerpo donde se incluye toda la información que se desea mostrar en la página de web.

Dicha información podrá visualizarse con apariencias muy diversas, según el formato aplicado a las distintas secciones que la componen. Por ejemplo, las operaciones de formato más comunes son los encabezados o títulos de secciones.

De esta manera, se pueden codificar grupos de palabras o frases completas usando estilos especiales. Existen dos tipos de estilos: lógico y físico. Con el estilo lógico se da formato al texto según su semántica, mientras que con el estilo físico se especifica exactamente cuál debe ser la apariencia final del texto.

Aunque pudiera parecer que ambos estilos son funcionalmente idénticos, lo cierto es que el estilo lógico permite una mayor flexibilidad, ya que permite a los usuarios indicar al cliente Web cómo desean ver determinados elementos de la página, personalizando de este modo la apariencia final de los documentos.

2.5.6. Páginas Active Server (ASP)

Páginas Active Server es un conjunto de secuencias de comandos de servidor que ejecuta secuencias de comandos ActiveX y componentes ActiveX en un servidor. Los programadores pueden combinar secuencias de comandos y componentes para crear aplicaciones basadas en Web. Estas páginas son archivos que contienen etiquetas HTML, texto y secuencias de comandos. Las páginas ASP pueden llamar a componentes ActiveX para que realicen tareas, como la conexión con bases de datos o cálculos comerciales y además pueden agregar contenido interactivo a las páginas Web o generar aplicaciones Web que utilicen páginas HTML como interfaz con el usuario. [7]

Los componentes ActiveX se han diseñado para que se ejecuten en el servidor Web como parte de las aplicaciones Web. Estos componentes proporcionan importante funcionalidad en las aplicaciones Web, como el acceso a bases de datos, de modo que no tenga que volver a crear código para realizar estas tareas. [7]

2.5.6. VBScript

VBScript, es un subconjunto del lenguaje Microsoft Visual Basic, se implementa como un intérprete rápido, portátil y ligero para su uso en exploradores de World Wide Web y otras aplicaciones que usan controles ActiveX, servidores de Automatización OLE y subprogramas Java. Suministra una variedad de comandos que pueden ser utilizados para la creación de páginas activas, incluidas las secuencias de comandos del cliente de Web en MS Internet Explorer y las secuencias de comandos en MS Internet Information Server. [7]

2.5.7. JavaScript

Java es un lenguaje creado por Sun Microsystems para la programación orientada a objetos multiplataforma. Un derivado del lenguaje C++; Java es un lenguaje de programación distribuida de SunSoft que se ofrece como un estándar abierto.

JavaScript es un lenguaje de secuencias de comandos que evolucionó a partir del lenguaje LiveScript de Netscape y que se hizo más compatible con Java. Utiliza una página HTML como interfaz. [7]

2.6. Modelos ARIMA

Hay dos objetivos principales en el análisis de series de tiempo: (a) identificar la naturaleza del fenómeno representado por la secuencia de observaciones, y (b) predecir valores futuros de la variable de la serie de tiempo. Estas metas requieren que el patrón de datos de la serie de tiempo observada sea identificado y más o menos descrito formalmente. Una vez que el patrón se establece, se puede interpretar e integrarlo con otros. Sin tener en cuenta la profundidad de nuestra comprensión y la validez de nuestra interpretación del fenómeno, es posible extrapolar el modelo identificado para predecir eventos futuros.

Como en la mayoría de los análisis, en el análisis de series de tiempo se supone que los datos consisten en un patrón sistemático (normalmente un conjunto de componentes identificables) y ruido aleatorio (error) el cual usualmente hace que el modelo sea difícil de identificar. La mayoría de las técnicas de análisis de series de tiempo involucran alguna forma de filtrado del ruido para hacer el modelo más prominente.

La mayoría de los patrones de series de tiempo pueden describirse en términos de dos clases básicas de componentes: tendencia y estacionalidad. El primero representa a un componente lineal general sistemático o no-lineal que cambia con el tiempo y no se repite o por lo menos no se repite dentro del rango de tiempo capturado por nuestros datos. El último puede tener una naturaleza formalmente similar, sin embargo, se repite con el tiempo en intervalos sistemáticos.

2.6.1. Análisis de la tendencia

No hay técnicas "automáticas" probadas para identificar componentes de la tendencia en los datos de serie de tiempo; sin embargo, con tal de que la tendencia sea monótona (aumentando de forma consistente o disminuyendo) esa parte del análisis de los datos no es típicamente muy difícil. Si los datos de serie de tiempo contienen error considerable, entonces el primer paso en el proceso de identificación de la tendencia es el suavizamiento.

El suavizamiento siempre involucra alguna forma de promedio local de los datos tal que el componente no sistemático de las observaciones individuales cancelan a las que están por fuera. La técnica más común es el suavizamiento de los promedios móviles el cual reemplaza cada elemento de la serie por los promedios simples o pesados de n elementos circundantes, donde n es el ancho de la "ventana" de suavizamiento. Las Medianas pueden usarse en lugar de las medias. La ventaja principal de la mediana como comparador al suavizamiento de los promedios móviles es que sus resultados son menos influenciados por los valores extremos. La desventaja principal del suavizamiento de la mediana es que en la ausencia de valores extremos claros puede producir curvas más "puntiagudas" que el promedio móvil.

2.6.2. Análisis de estacionalidad

La Dependencia estacional (estacionalidad) es otro componente general del patrón de series de tiempo. Se define formalmente como la dependencia correlacional de orden k entre cada i -ésimo elemento de la serie y el $(i-k)$ -ésimo y medido por autocorrelación, k es normalmente llamado el retraso. Si el error de la medida no es demasiado grande, la estacionalidad puede ser visualmente identificada en las series como un patrón que repite cada k elementos.

Los patrones estacionales de las series de tiempo pueden ser examinados mediante correlogramas. El correlograma (autocorrelograma) muestra gráfica y numéricamente la función de autocorrelación (ACF), esto es, los coeficientes de correlación serial (y sus errores estándar) para los retrasos consecutivos en un rango especificado de retrasos. Los rangos de dos errores estándar para cada retraso son normalmente marcados en correlogramas pero típicamente el tamaño de auto-correlación es de más interés que su fiabilidad porque usualmente estamos interesados solo en las autocorrelaciones muy fuertes (y así muy significativas).

Mientras examinamos los correlogramas uno debe tener presente que las autocorrelaciones para retrasos consecutivos son formalmente dependientes. Considere el siguiente ejemplo, si el primer elemento se relaciona estrechamente al segundo, y el segundo al tercero, entonces el primer elemento también debe relacionarse un poco al tercero, etc. Esto implica que el patrón de dependencias seriales puede cambiar considerablemente después de quitar la auto correlación de primer orden (es decir, después de diferenciar la serie con un retraso de 1).

Otro método útil para examinar dependencias seriales es examinar la función de autocorrelación parcial (PACF), una extensión de autocorrelación, donde la dependencia en los elementos intermedios es removida. En otras palabras la autocorrelación parcial es similar a la autocorrelación, sólo que al calcularla, las (auto) correlaciones con todos los elementos dentro del retraso están parcialmente fuera (BOXJ76). Si un retraso de 1 es especificado entonces la autocorrelación parcial es equivalente a la auto correlación. En cierto sentido, la autocorrelación parcial provee una descripción "limpiadora" de dependencias seriales para los retrasos individuales (no confundidos por otras dependencias seriales).

La dependencia serial para un retraso particular de k puede ser removida diferenciando la serie, esto es convirtiendo cada i -ésimo elemento de la serie en su diferencia del $(i-k)$ -ésimo elemento. Hay dos grandes razones para tales transformaciones. Primero, uno puede identificar la naturaleza oculta de las dependencias estacionales en la serie. Recuerde que, como se mencionó en el párrafo anterior, las autocorrelaciones para retrasos consecutivos son interdependientes. Por consiguiente, quitando alguna de las autocorrelaciones cambiarán las otras auto correlaciones, es decir, puede eliminarlos o puede hacer algunas otras estacionalidades más claras. La otra razón para quitar las dependencias estacionales es hacer la serie estacionaria. Lo cual se hace con ARIMA y otras técnicas.

2.6.3. Metodología ARIMA

Los procedimientos de modelaje y predicción involucran conocimiento sobre el modelo matemático del proceso. Sin embargo, en la investigación y práctica de la vida real, los patrones de los datos son poco evidentes, las observaciones individuales involucran considerable error, y a pesar de eso no sólo necesitamos descubrir los patrones ocultos en los datos sino también generar predicciones. La metodología de ARIMA desarrollada por Box y Jenkins (1976) nos permite hacer justo eso; ha ganado enorme popularidad en muchas áreas, y la práctica de la investigación confirma su poder y flexibilidad, sin embargo, es una técnica compleja; no es fácil su uso, requiere mucha experiencia, y aunque produce a menudo resultados satisfactorios, esos resultados dependen del nivel de especialización del investigador.

2.6.3.1. Procesos autorregresivos.

La mayoría de las series de tiempo consisten en elementos que son consecutivamente dependientes en el sentido que uno puede estimar un coeficiente o un juego de coeficientes que describen elementos consecutivos de la serie a partir de elementos específicos anteriores. Esto puede resumirse en la ecuación:

$$x_t = \xi + \phi_1 * x_{t-1} + \phi_2 * x_{t-2} + \phi_3 * x_{t-3} + \dots + \varepsilon$$

Donde:

ξ es una constante (intercepto), y

$\phi_1, \phi_2, \phi_3, \dots$ los parámetros autoregresivos del modelo.

Puesto en palabras, cada observación se compone de un componente de error aleatorio (ϵ_t) y una combinación lineal de observaciones anteriores.

2.6.3.2. Modelo de los promedios móviles autoregresivos.

El modelo general introducido por Box y Jenkins trata la autoregresión así como también los parámetros de los promedios móviles, y explícitamente incluye diferenciación en la formulación del modelo. Específicamente, los tres tipos de parámetros en el modelo son: los parámetros autoregresivos (p), el número de pasos diferenciados (d), y los parámetros de los promedios móviles (q). En la notación introducida por Box y Jenkins, los modelos son resumidos como ARIMA (p, d, q); así, por ejemplo, un modelo descrito como (0, 1, 2) significa que contiene cero parámetros autoregresivos (p) y dos parámetros de promedios móviles (q) los cuales se computaron para la serie después de que fue diferenciada una vez.

2.6.3.3. Identificación.

Como se mencionó antes, la serie de entrada para el ARIMA necesita ser estacionaria, esto es, debe tener una media constante, varianza y autocorrelación a través del tiempo. Por consiguiente, normalmente la serie necesita ser diferenciada primero hasta que sea estacionaria (esto requiere también a menudo una transformación logarítmica a los datos para estabilizar la varianza). El número de veces que la serie necesita ser diferenciada para lograr estacionariedad se refleja en el parámetro d (vea el párrafo anterior). Para determinar el nivel necesario a diferenciar, uno debe examinar el gráfico de los datos y el autocorrelograma. Cambios significativos en el nivel (cambios ascendentes o descendentes fuertes) normalmente requieren diferenciación de primer orden no estacional; los cambios fuertes de pendiente normalmente requieren diferenciación de segundo orden no estacional. Los modelos estacionales requieren la respectiva diferenciación estacional. Si los coeficientes de autocorrelación estimados disminuyen lentamente a retrasos más largos, necesita diferenciación de primer

orden. Sin embargo, uno debe tener presente que algunas series de tiempo pueden requerir poca o ninguna diferenciación, y que series sobre diferenciadas producen estimaciones de coeficientes menos estables.

En esta fase (que normalmente se llama fase de identificación) también se necesita decidir cuántos parámetros autoregresivos (p) y de promedio móviles (q) son necesarios para producir un modelo efectivo, pero aún así parsimonioso del proceso (parsimonioso significa que tiene menos parámetros y el mayor número de grados de libertad entre todos los modelos que estiman los datos).

2.6.3.4. Estimación y Predicción.

En el próximo paso (estimación), los parámetros son estimados para que la suma de residuos cuadrados se minimice. Se usan las estimaciones de los parámetros en la última fase (predicción) para calcular nuevos valores de la serie (más allá de aquellos incluidos en el conjunto de datos de entrada) e intervalos de confianza para esos valores predichos. El proceso de estimación es realizado sobre los datos transformados (diferenciados). Antes de que las predicciones se generen, la serie necesita ser integrada (la integración es el inverso de la diferenciación) para que las previsiones se expresen en valores compatibles con los datos de entrada. Este rasgo de la integración automática es representado por la letra I en el nombre de la metodología (ARIMA = Auto Regresive Integrates Moving Average - Promedios Móviles Integrados Auto Regresivos).

La constante en los modelos ARIMA: además de los parámetros estándar autoregresivos y de promedios móviles, los modelos ARIMA pueden incluir también una constante, como se describió anteriormente. La interpretación de una constante (estadísticamente significativa) depende del modelo que se está ajustando. Específicamente, (1) si no hay ningún parámetro autoregresivo en el modelo, entonces el valor esperado de la constante es μ , la media de la serie; (2) si hay parámetros autoregresivos en la serie, entonces la constante representa el intercepto. Si la serie es diferenciada, entonces la constante representa la media o

intercepto de la serie diferenciada; Por ejemplo, si la serie se diferencia una vez, y no hay ningún parámetro autoregresivo en el modelo, entonces la constante representa la media de la serie diferenciada, y por consiguiente la pendiente de la tendencia lineal de la serie no diferenciada.

3. Análisis y diseño de la base de datos de la Economía Venezolana.

3.1. Modelo Entidad-Relación

Como se dijo en el capítulo 1, el libro de las “Bases Cuantitativas de la Economía Venezolana” está formado por Capítulos, cada Capitulo por Cuadros y los Cuadros tienen Series de datos (temporales).

Visto desde el punto de vista del modelo Entidad-Relación el libro está formado por:

- La entidad Capítulos, que se refiere a los capítulos del libro. Cada capítulo contiene varios cuadros.
- El objeto o entidad Cuadros, formado por los diferentes cuadros de datos del tema del capítulo, tanto principales y secundarios. Cada cuadro contiene una o más series de datos (temporales).
- Entidad Series, que se refiere a las diferentes series de datos del cuadro. Una Serie está formada por varios datos o valores.

Además de las entidades nombradas anteriormente el modelo contiene las siguientes relaciones:

- Relación CAPÍTULOS-CUADROS, que relaciona a un capítulo con varios Cuadros.
- Relación CUADROS-SERIES, que relaciona a un Cuadro con una o más Series.
- Relación UNIDADES-SERIES, que relaciona una serie con su unidad de medida.
- Relación PERIODOS-SERIES, que relaciona una serie con su período.

En la figura 2 se muestra el modelo Entidad-Relación de las “Bases Cuantitativas”.

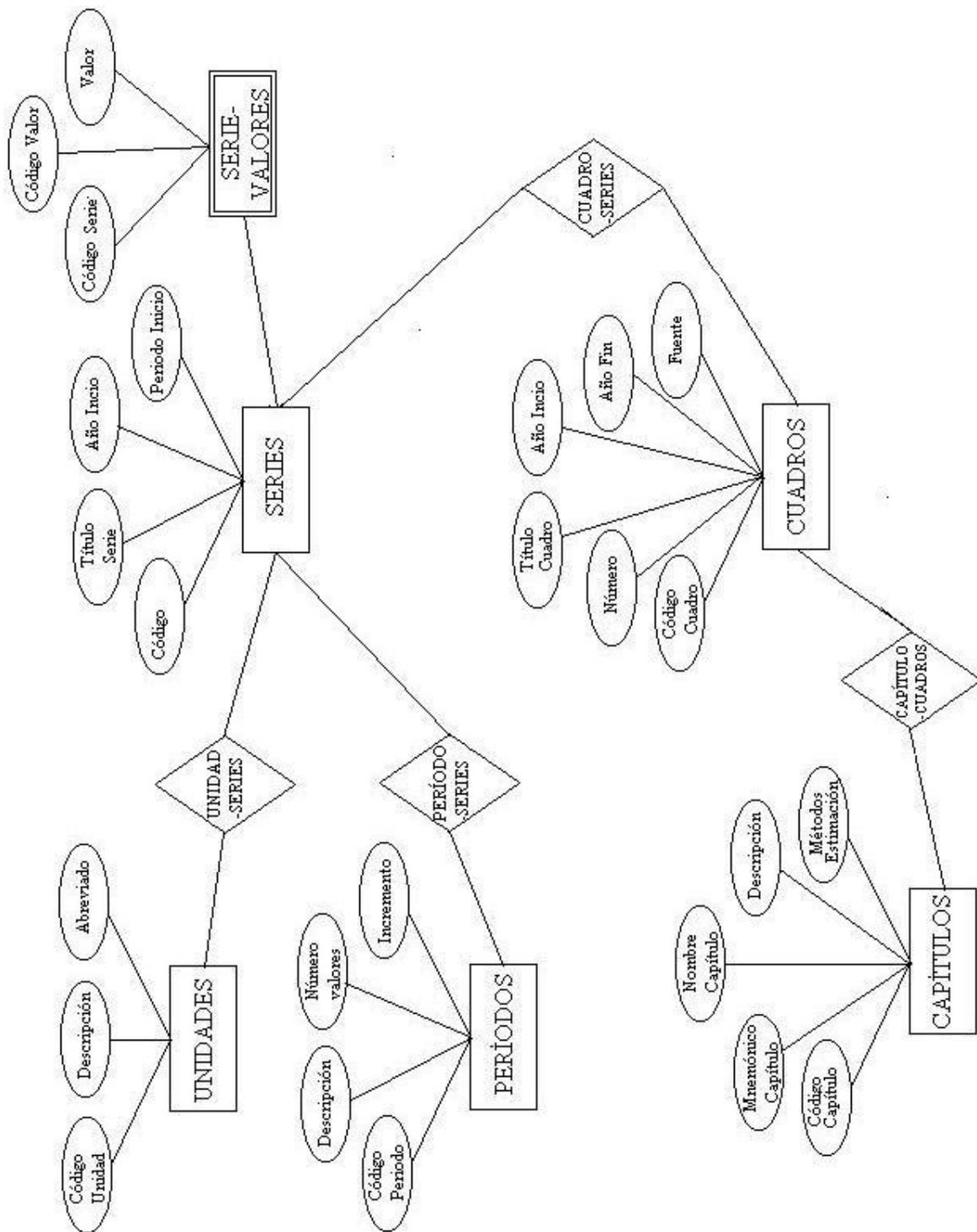


Figura 2. Diagrama Entidad-Relación de la base de datos.

3.2. Resumen descriptivo del modelo de datos.

Esta descripción consiste en desglosar las entidades y relaciones en los atributos que las identifican. Como se dijo anteriormente cada atributo posee un rango o dominio que limita su espacio de acción, los mismos se listan a continuación:

- An : permite introducir una cadena de caracteres (alfanumérica) de longitud n, longitud máxima 255 caracteres. Ej: el título de la serie es una cadena de longitud 50, es decir, A50.
- N : admite cualquier número o dígito, perteneciente a un subconjunto de los números reales.
- M : para almacenar textos mayores a 255 caracteres hasta 64K, se le conoce como campos MEMO.

1. Entidad SERIES: Se refiere a la Entidad principal del sistema que se está diseñando, pues contiene la información de las diferentes SERIES de datos del libro. Cada serie se identificará de entre las demás por los siguientes atributos:

1.1. Código de serie (N): es un número secuencial desde 1 en adelante que identifica la serie (Clave principal).

1.2. Título de la serie (A50): especifica el nombre o título de la serie.

1.3. Año de inicio (N): especifica el año de comienzo de la serie.

1.4. Período de inicio (N): identifica el período de inicio dentro del período principal. Si el período de la serie es anual, el período de inicio siempre es 1, pero si el período es trimestral el período de inicio puede ser 1, 2, 3 o 4.

2. Entidad CUADROS: se refiere al cuadro dentro del libro al cual pertenece la serie, se caracteriza por los siguientes atributos:

- 2.1. Código de Cuadro (N): un número secuencial desde 1 en adelante que identifica de manera única cada cuadro (Clave).
 - 2.2. Número (N): un entero, que se refiere al número del cuadro dentro del capítulo del libro. Junto con el mnemónico del capítulo representa la nomenclatura del autor del libro para los cuadros en el libro.
 - 2.3. Título del cuadro (A50): especifica el título o nombre del cuadro.
 - 2.4. Año Inicio (N): Representa el menor año de inicio de las diferentes series que pertenecen al cuadro.
 - 2.5. Año Fin (N): Representa el mayor año final de las diferentes series que pertenecen a la tabla.
 - 2.6. Fuente (M): Mediante este atributo se describe las fuentes que usó el autor para conseguir los datos de las diferentes series del cuadro.
3. Entidad Capítulos: Mediante esta entidad se describe el capítulo del libro a cual pertenece un determinado cuadro. Se identifica por los siguientes atributos.
 - 3.1. Código del Capítulo (N): Número secuencial que identifica de manera única a cada capítulo (CLAVE).
 - 3.2. Mnemónico del Capítulo (A5): Identificador usado en el libro para cada capítulo (A, B, I ,II,...), junto con el número de cuadro representa la nomenclatura del autor para los cuadros en el libro.
 - 3.3. Nombre del capítulo (A15): Representa el nombre del capítulo en el libro (CAPITULO I, CAPITULO II, APENDICE A, APENDICE B, ...).
 - 3.4. Descripción (A50): Representa el título del capítulo, y se refiere a una descripción general de lo que contiene el capítulo.

- 3.5. Métodos de estimación (M): contiene una breve descripción de cómo el autor consiguió los diferentes datos que se presentan en el capítulo.
4. Entidad UNIDADES: Con esta entidad se describe la unidad de medida en que se representan los datos, por ejemplo: bolívares, millones de bolívares, dólares, 1968=100, etc. Contiene los siguientes atributos:
- 4.1. Código de unidad (N): número entero que identifica cada una de las unidades de medidas (CLAVE).
- 4.2. Descripción (A30): breve comentario acerca de la unidad.
- 4.3. Abreviado (A12): la descripción en forma abreviada utilizada en algunas consultas e impresión.
5. Entidad PERÍODOS: esta entidad describe los diferentes períodos en que se pueden representar las series, aunque por lo general estas series están presentadas en forma anual. Se describe con los siguientes atributos:
- 5.1. Código de período (N): número entero consecutivo desde 1 que representa la clave de la entidad.
- 5.2. Descripción (A30): Descripción o nombre del período.
- 5.3. Número de valores (N): número de subunidades en que se divide la unidad principal, en el caso semestral son 2, en caso de trimestres son 4, si es anual, bianual o quinquenal es 1.
- 5.4. Incremento (N): número real que representa el incremento en el período de un valor a otro, en el caso diario, semanal, mensual hasta anual es 1; si es bianual es 2, quinquenal 5, etc.
6. Entidad débil SERIE-VALORES: entidad dependiente de la entidad serie que contiene los diferentes valores de la serie almacenada.
- 6.1. Código de serie (N): código de la serie

- 6.2. Código de valor (N): número secuencia desde 1 hasta el número de valores de la serie, junto con el código de serie representa la clave de esta entidad.
- 6.3. Valor (N): número real (doble precisión) para guardar el valor en cada período.
- 7. Relación CUADROS-SERIES: asocia cada serie con el respectivo cuadro en el libro mediante los códigos de cada una de ellas.
 - 7.1. Código de serie.
 - 7.2. Código de cuadro.
- 8. Relación CAPÍTULOS-CUADROS: asocia a cada cuadro del libro con el respectivo capítulo al que pertenece.
 - 8.1. Código de cuadro.
 - 8.2. Código de capítulo.
- 9. Relación UNIDADES-SERIES: es la forma de asociarle a cada serie su respectiva unidad de medida.
 - 9.1. Código de serie.
 - 9.2. Código de unidad.
- 10. Relación PERÍODOS-SERIES: es la forma de asociarle a cada serie un período.
 - 10.1. Código de serie.
 - 10.2. Código de período.

3.3. Modelo Relacional de la Base de datos

A continuación se presenta el modelo relacional de la base de datos en forma de tablas:

SERIES			
Código de serie (N)	Título de la serie (A50)	Año de inicio (N)	Período de inicio (N)

CUADROS					
Código de Cuadro (N)	Número (N)	Título del cuadro (A50)	Año Inicio (N)	Año Fin (N)	Fuente (M)

CAPITULOS				
Código del Capítulo (N)	Mnemónico del Capítulo (5)	Nombre del capítulo (A15)	Descripción (A50)	Métodos de estimación (M)

UNIDADES		
Código de unidad (N)	Descripción (30)	Abreviado (A12)

PERÍODOS			
Código de período (N)	Descripción (A30)	Número de valores (N)	Incremento (N)

SERIE-VALORES

Código de serie (N)	Código de valor (N)	Valor (N)
------------------------	------------------------	-----------

CUADROS-SERIES

Código de Cuadro (N)	Código de Serie (N)
----------------------	---------------------

CAPÍTULOS-CUADROS

Código de Capítulo (N)	Código de Cuadro (N)
------------------------	----------------------

UNIDADES-SERIES

Código de Serie (N)	Código de Unidad (N)
---------------------	----------------------

PERÍODOS-SERIES

Código de Serie (N)	Código de Período (N)
---------------------	-----------------------

3.4. Implementación de la Base de Datos.

La base de datos se implementó en el Microsoft Access 7.0, el cual es un Sistema Manejador de Base de Datos Relacional. La figura 3 muestra el modelo.

Los datos estaban disponibles en forma de hojas de cálculo en EXCEL, tal y como están en el libro. Para llenar la base de datos se manipularon mediante las funciones de cortar y pegar y fueron llevadas a las diferentes tablas en el modelo implementado en ACCESS. Los valores faltantes o no disponibles fueron marcados o etiquetados con el valor -1000, se utilizó este valor porque no figura como un valor posible de los datos que se almacenan.

Con respecto a los campos FUENTE de la tabla CUADROS y METODOS DE ESTIMACIÓN de la tabla CAPITULO se tomaron directamente del libro digitalizando las páginas del mismo.

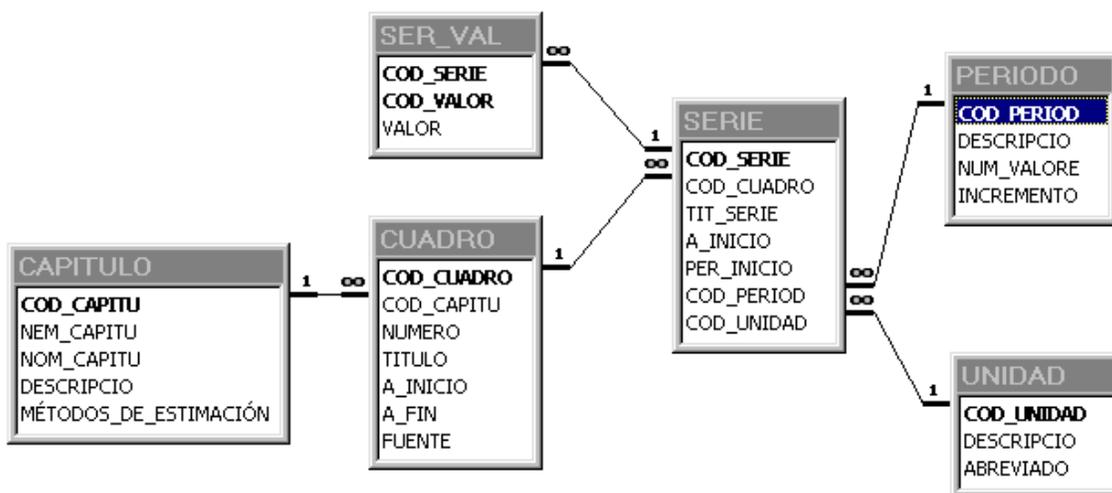


Figura 3. Modelo relacional de la base de datos

A pesar de que la base de datos una vez implementada es estática, en el sentido de que lo que se quiere tener (y representa) es una versión digital de los datos presentados en el libro, se desarrolló una aplicación en ACCESS para aquellos usuarios que quieren que estos datos crezcan y puedan dar entrada a nuevas

tablas o hacer que las mismas se amplíen en el tiempo (valores para los años posteriores a 1989).

La aplicación se ejecuta en el momento que se carga el archivo BASESCUA.MBD presentando la ventana que se muestra en la figura 4, en la cual se pueden ver las tablas que componen la base de datos y/o anexar nuevos datos a las mismas.

Si lo que desea es agregar nuevos valores a las series seleccione la opción 3 (Anexar/Ver Serie) y allí busque la serie que desea y agregue en el cuadro de lista de la parte derecha los valores continuando la numeración para el campo COD_VALOR (ver figura 5).

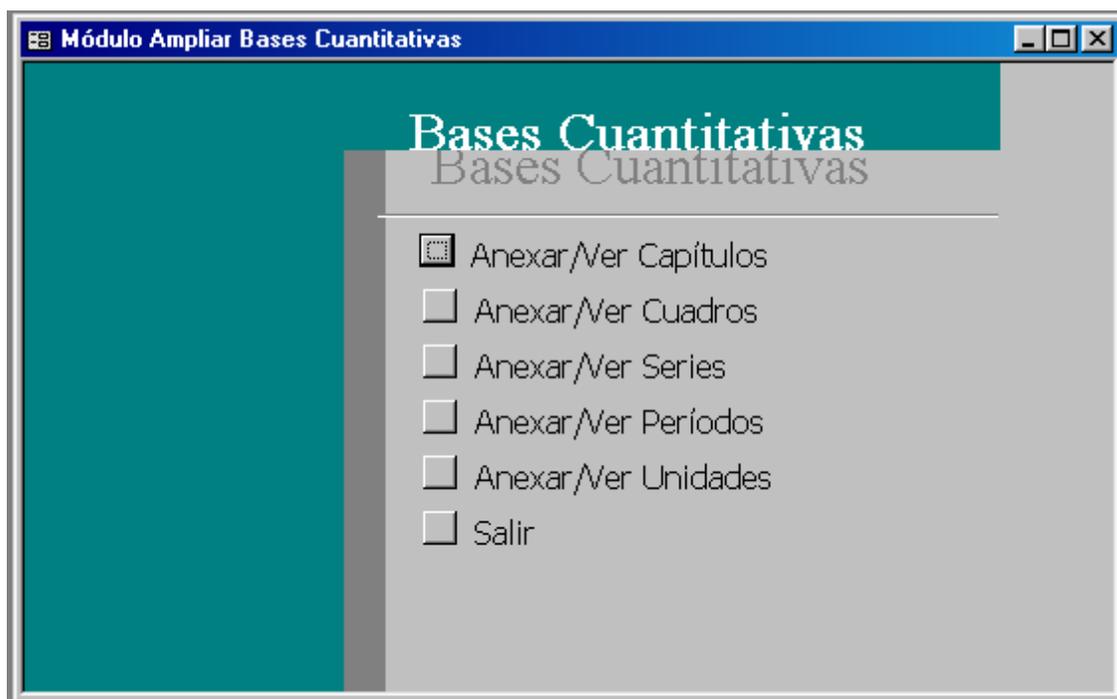
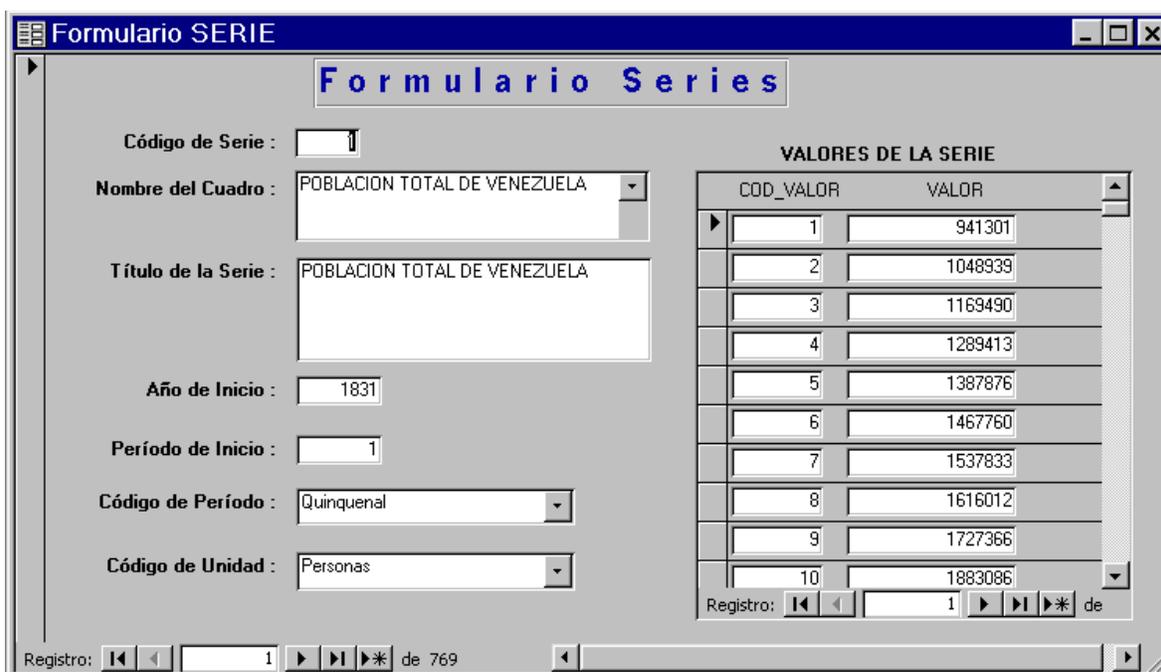


Figura 4. Ventana principal de la aplicación que permite ampliar las Bases Cuantitativas.

Cuando el usuario desea insertar una serie nueva con descripción y valores debe agregar un nuevo registro a la tabla SERIE utilizando el formulario anterior presionando nuevo registro en la barra de desplazamiento de la tabla situada en la parte inferior derecha, el control de registro se situará en un registro con todos

los campos vacíos, llene los datos y de esta forma estará creando una nueva serie.

Presionando la opción 1 (Anexar/Ver Capítulo) del Panel principal de la figura 4 puede crear un nuevo capítulo; de la misma forma puede crear nuevos cuadros (opción 2) y, de ser necesarias nuevas unidades y períodos, por ejemplo hoy en día se habla de Millardos en vez de millones de millones.



COD_VALOR	VALOR
1	941301
2	1048939
3	1169490
4	1289413
5	1387876
6	1467760
7	1537833
8	1616012
9	1727366
10	1883086

Figura 5. Formulario para dar entrada a los datos de una serie.

Como se dijo anteriormente, una base de datos es un conjunto de datos interrelacionados entre sí junto con un conjunto de programas que los manipulan.

Una vez creada la base de datos se diseñó e implementó un programa en Visual Basic para la manipulación de los datos ya almacenados en la base de datos y poderlos visualizar en la misma manera en que se presentan en el libro en forma de cuadros, como si se tratara de una versión del libro pero en el computador.

Esta versión digital va mucho mas allá de lo citado en el párrafo anterior, una vez localizado el cuadro deseado y las series asociadas al mismo, podemos imprimirlo o almacenarlo en forma de un archivo texto para editarlo después.

También podemos recorrer una a una todas las series del libro y luego imprimirlas, almacenarlas, ver un gráfico de dispersión o inclusive correr algún modelo ARIMA a la serie.

En los siguientes apartados (Instalación y Cómo funciona las Bases Cuantitativas) se describirá detalladamente la aplicación.

3.5. Instalación de Bases Cuantitativas

Bases cuantitativas es un programa diseñado en Visual Basic que permite ver o recorrer las series y tablas (cuadros) del libro “Bases Cuantitativas de la economía venezolana. 1830-1989” de Asdrúbal Baptista.

Bases cuantitativas se distribuye como una aplicación para Windows 95EOM (W96) o Windows 98 en 4 discos de 3 ½ “. Para su instalación introduzca el disco marcado con el número uno y siga cualquiera de los siguientes procedimientos:

- Seleccione Inicio | Ejecutar... y en el cuadro de diálogo Ejecutar escriba a:setup
- Desde Mi PC o el Explorador de Windows active la unidad A: y luego seleccione el icono Setup y le da doble click.

Luego siga las instrucciones de instalación y al finalizar la misma, el icono de Bases Cuantitativas se verá en el menú Inicio | Programas (ver fig. 6).

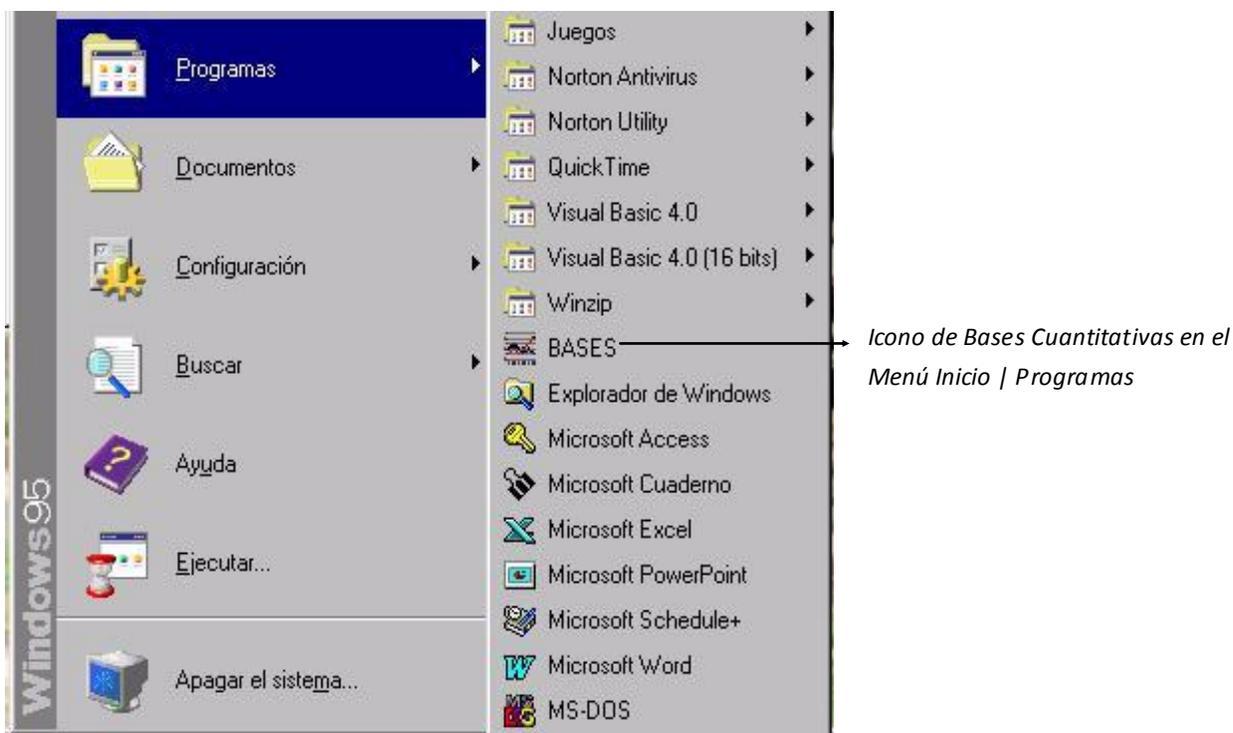


Figura 6 Cómo ejecutar Bases Cuantitativas.

3.6. ¿Cómo ejecutar Bases Cuantitativas?

Para ejecutar el programa Bases Cuantitativas, una vez instalado, se busca en el menú Inicio del Windows en: Inicio | Programas | Bases, como se muestra en la figura 6.

Al ejecutarse Bases Cuantitativas se mostrará la pantalla de presentación y bienvenida (ver figura 7), presione la tecla Enter o presione el botón Aceptar para continuar con la carga del programa.

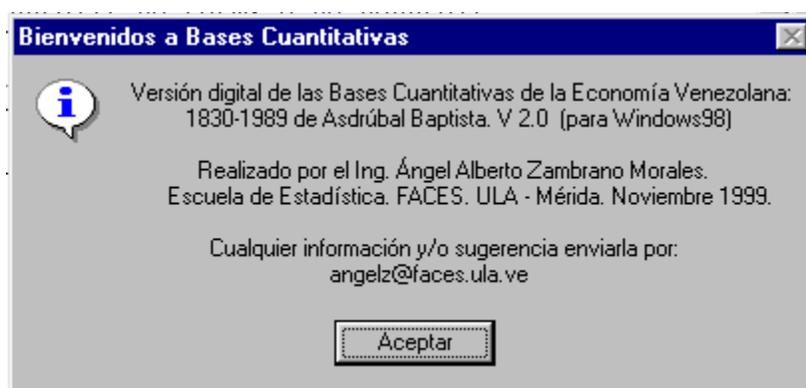


Figura 7. Pantalla de presentación y bienvenida.

Una vez cargado el programa, aparecerá centrada en la pantalla una ventana titulada “Bases Cuantitativas” (Figura 8), que corresponde al ambiente de interacción con el usuario, desde esta ventana se invocan las dos actividades generales del programa. Las mismas pueden invocarse de dos formas, bien sea a través de los botones ubicados hacia la parte izquierda (Ver Cuadros, Ver Series o Salir) o seleccionándolos en la barra de menú (Ver Cuadros, Ver Series, Salir o ?).

- Ver Cuadros : mediante esta opción se recorren las Bases Cuantitativas secuencialmente en forma de cuadros, mostrando para cada cuadro las series asociadas a él.
- Ver Series : con esta opción recorreremos las Bases Cuantitativas secuencialmente serie por serie, presentando para cada una de ellas los aspectos más resaltantes.

- Salir : para terminar una sesión de trabajo con las Bases Cuantitativas.
- ? : con esta opción, sólo disponible en la barra de menú, se muestra la pantalla Acerca de... o de Bienvenida.

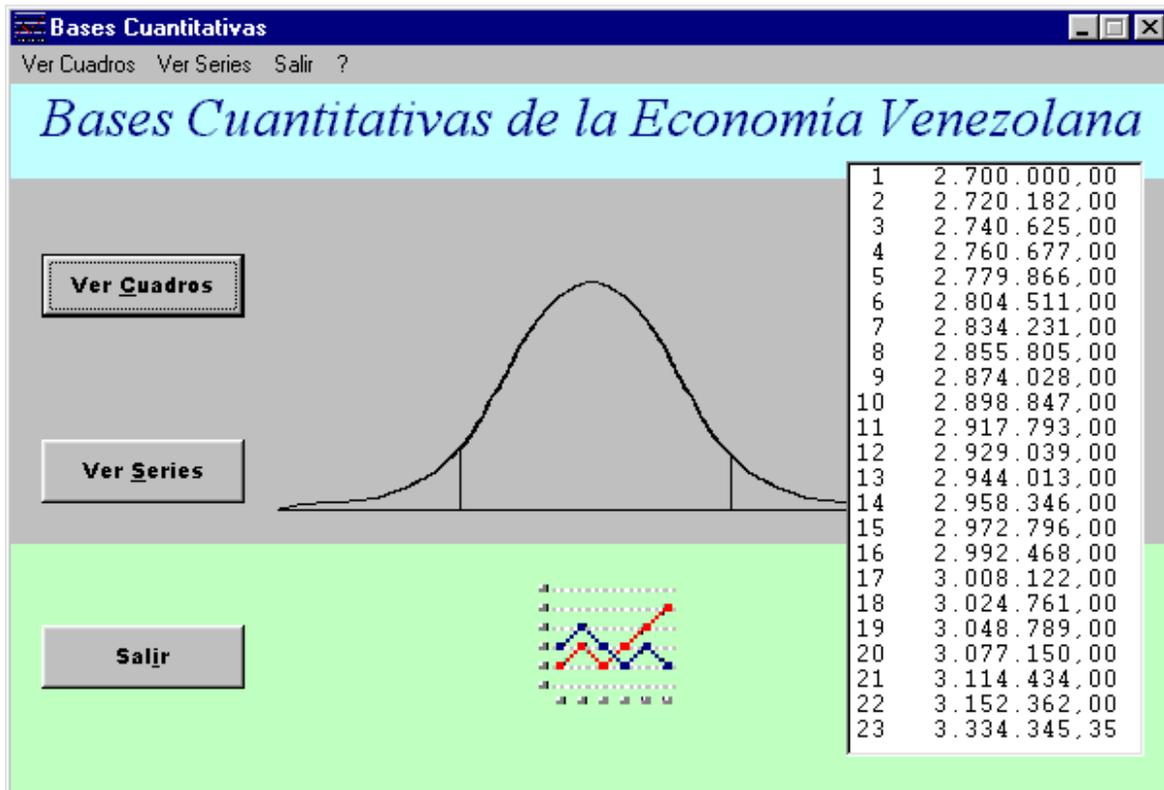


Figura 8. Ventana principal de Bases Cuantitativas.

Ver Cuadros.

Como se dijo antes, esta opción permite visualizar las *Bases Cuantitativas* por cuadros, comenzando desde el primero que se refiere a la Población total de Venezuela, se recorre cuadro a cuadro, hasta llegar al último que se denomina Índice de precios en la manufactura.

Cuando se ejecuta Ver Cuadros se presenta la ventana titulada: Ver Bases Cuantitativas por Cuadros... (figura 9). Esta ventana se divide en cuatro zonas para mostrar cada cuadro, estas son:

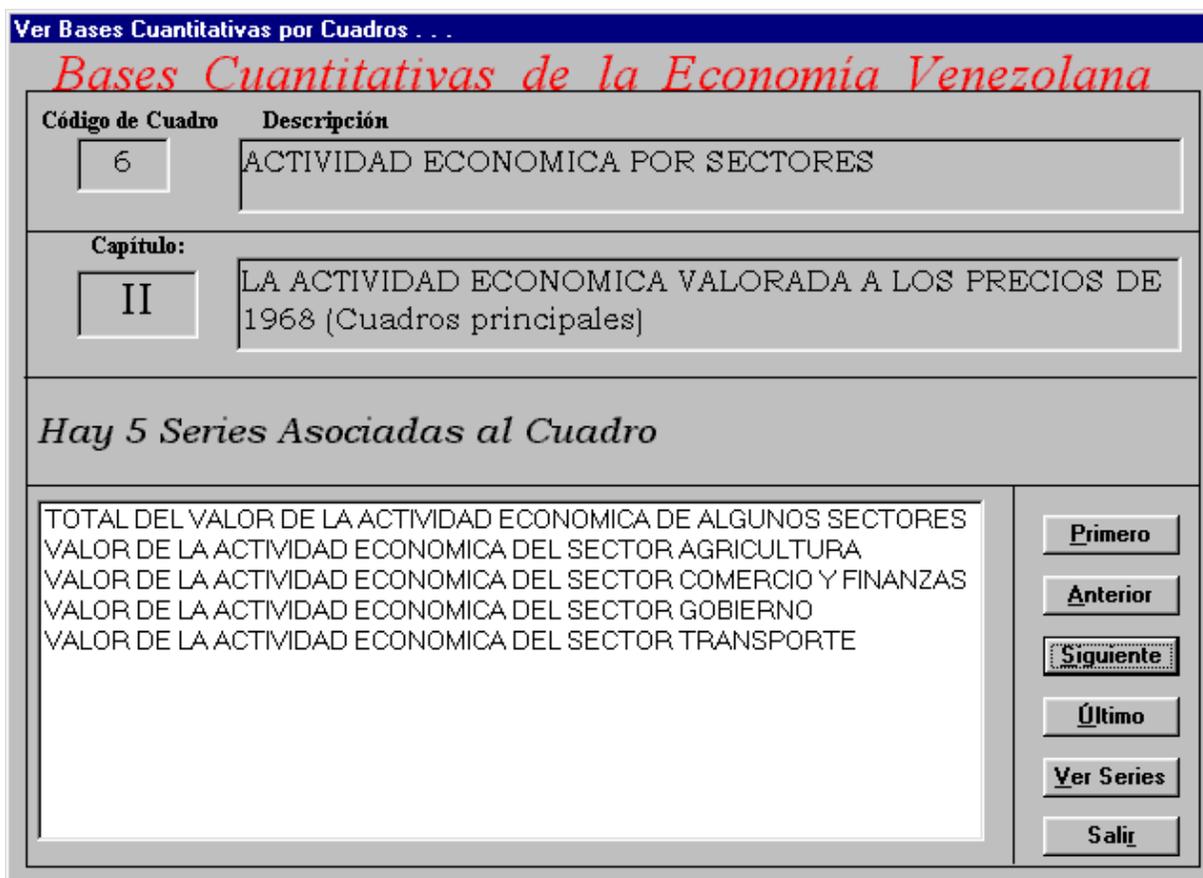


Figura 9. Ventana de la opción Ver Cuadros.

- 1) La parte superior de la ventana, en forma de cajas de texto muestra los siguientes datos: Código del cuadro, Descripción o título del cuadro, así como también el Mnemónico y la Descripción del capítulo que corresponde en el libro.
- 2) En la parte central se muestra un mensaje indicando el número de series asociadas o contenidas en el cuadro.
- 3) En la parte inferior se muestra, en forma de un cuadro de lista, el título de cada una de las series contenidas en el cuadro.
- 4) A la derecha de la ventana y en la parte inferior se encuentran seis botones, los cuales se describen a continuación:

- Primero, para ir al primer cuadro.
- Anterior, para retroceder al cuadro precedente.
- Siguiente, para avanzar al próximo cuadro.
- Ultimo, para movernos hasta el final de los cuadros.
- Salir, permite regresar a la ventana principal de las Bases Cuantitativas, presentando previamente una caja de mensajes (ver figura 10) donde se pide verificación si desea salir y regresar a la ventana principal.

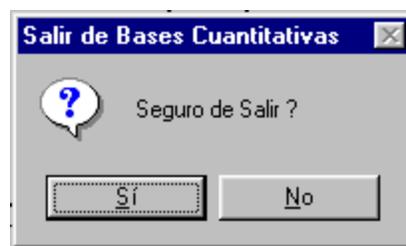


Figura 10. Ventana Salir del programa.

- Ver Series, presenta en una cuadrícula, formada por filas y columnas (ver figura 11), cada columna contiene los datos de las diferentes series contenidas en la tabla, enumeradas con las palabras Col1, Col2, Col3, ...

La primera columna contiene los años o períodos para los cuales está definida. La cuadrícula permite movernos a través de las columnas y filas de la misma, mediante las barras de desplazamientos (horizontal y vertical).

En la parte inferior de la ventana el cuadro de lista donde estaban los nombres de las series se hace más pequeño, y en él se describe el título de cada columna, en la figura 10 se puede observar que la primera columna (Col1) corresponde a la serie "TOTAL DEL VALOR DE LA ACTIVIDAD ECONOMICA DE ALGUNOS SECTORES".

Al mismo tiempo tres de los botones que estaban en la parte derecha de la ventana desaparecen y los otros tres cambian de la siguiente manera:

- Imprimir, permite generar una tabla en la impresora predeterminada del Windows de todas las series que pertenecen al cuadro que se está observando. Si son más de 4 series, se generaran dos (o más) listados paralelos, uno con las cuatro primeras series, el siguiente con las cuatro siguientes, y así sucesivamente (ver anexo 3).

Ver Bases Cuantitativas por Cuadros . . .

Bases Cuantitativas de la Economía Venezolana

	Col1	Col2	Col3	Col4	Col5
1878	167,81	73,04	31,28	59,30	
1879	167,85	80,54	36,24	46,88	
1880	209,50	122,10	37,57	44,61	
1881	246,44	136,08	45,21	59,01	
1882	291,75	170,87	64,65	48,97	
1883	251,29	122,02	52,91	70,10	6,26
1884	261,77	138,75	48,66	66,85	7,51
1885	280,53	155,71	54,66	59,73	10,43
1886	264,71	132,34	62,41	54,94	15,02
1887	293,14	132,21	64,71	80,65	15,57
1888	303,97	146,54	66,90	71,07	19,47
1889	295,50	126,70	67,00	86,22	15,57
1890	314,74	146,31	56,33	92,35	19,75
1891	285,70	125,97	58,15	84,20	17,38
1892	248,24	117,26	46,67	62,89	21,41
1893	347,40	161,13	64,41	97,95	23,92
1894	299,74	136,78	58,24	82,90	21,83
1895	346,77	142,76	59,13	124,72	20,16

Col1 -> TOTAL DEL VALOR DE LA ACTIVIDAD ECONOMICA DE ALGUNOS SECT
 Col2 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR AGRICULTURA
 Col3 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR COMERCIO Y FINA
 Col4 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR GOBIERNO
 Col5 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR TRANSPORTE

Imprimir
 Almacenar
 Cerrar

Figura 11. Cuadrícula con las diferentes series de una tabla.

- Almacenar, permite guardar la tabla, aproximadamente como se ve en el libro, en un archivo tipo texto en el directorio actual con el nombre sugerido de “Tablai.txt” donde la i corresponde al código de la tabla (ver figura 11). En el archivo que se genera se muestra: Número de la tabla, Título, año de comienzo, año final, una primera columna para los períodos de la tabla y tantas columnas como series tiene la tabla; al final de las series se presenta la leyenda para cada una de las series.

- Cerrar, se cierra la cuadrícula que muestra las series del cuadro y se restauran los botones a su aspecto original (figura 9).

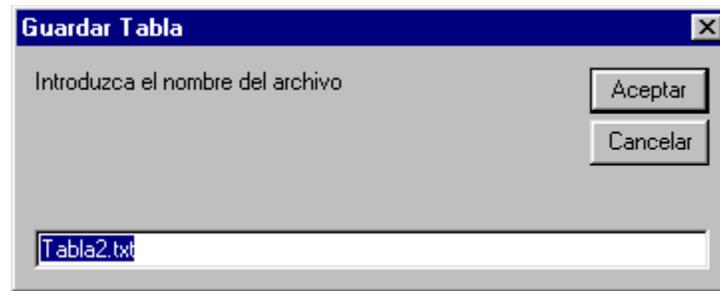


Figura 12. Cuadro de diálogo para Almacenar Cuadro.

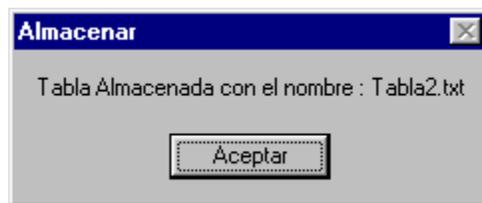


Figura 13. Mensaje de fin de Almacenar.

Ver Series.

Esta opción de la ventana principal permite recorrer, una tras otra, las series de las Bases Cuantitativas, puede buscar una en especial por su código secuencial, imprimirla, obtener un gráfico de tendencia, almacenarla o aplicarle la metodología ARIMA a los valores de la serie.

Cuando se activa o ejecuta esta opción aparece la ventana que se muestra en la figura 14. Esta ventana se divide en cuatro zonas:

- 1) En la parte superior derecha se presenta en forma de cajas de texto, información acerca del cuadro a que pertenece la serie (código de la tabla y la descripción o título), debajo de esta se ve información referente a la serie (título, código, año de inicio, período y la unidad de medida).
- 2) Hacia la parte izquierda en forma de un cuadro de lista se muestra los valores de la serie con su respectivo año o período. Puede desplazarse a través de la

lista moviendo la barra de desplazamiento o los botones de subir y bajar de un cuadro de lista.

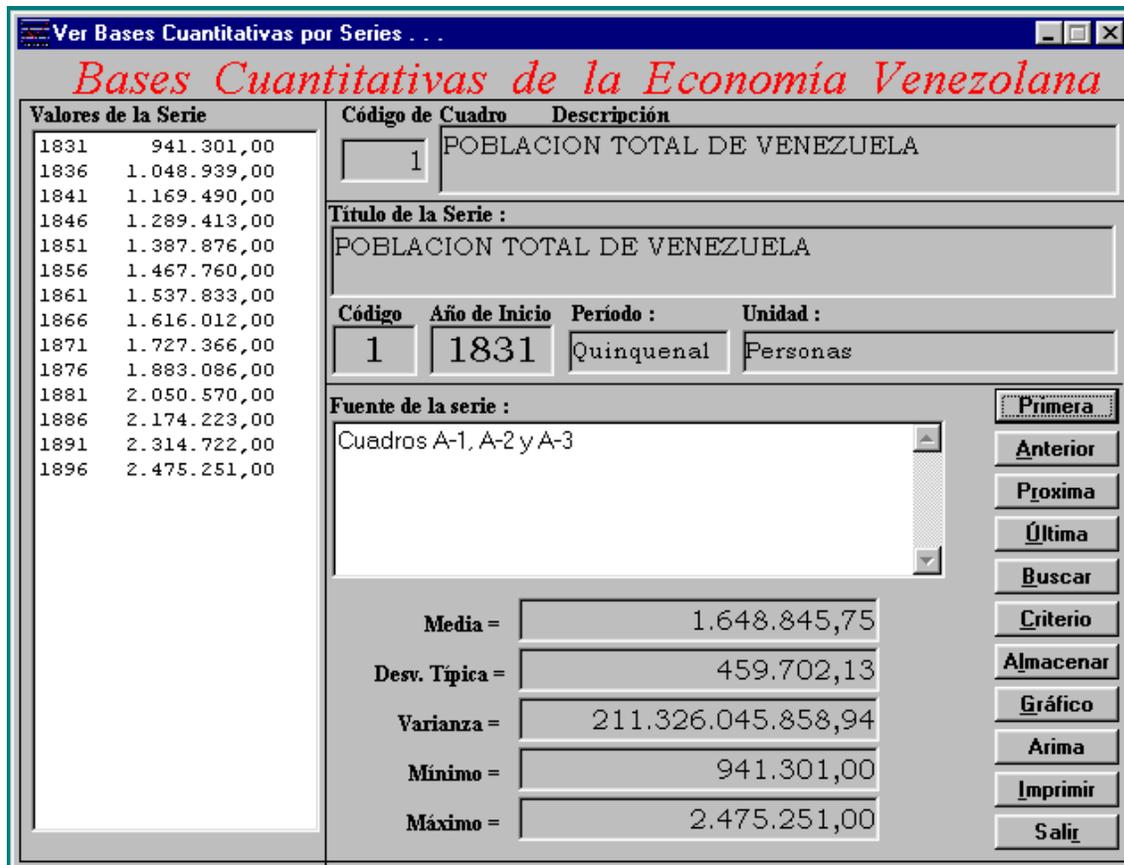


Figura 14. Ventana Ver Bases Cuantitativas por Series.

- 3) En la parte central se observa mediante un cuadro de texto multilínea la fuente de los datos de la serie.
- 4) Hacia la parte inferior central se observa en cuadros de textos los valores de los estadísticos básicos: la media, la desviación estándar, la varianza, el mínimo y el máximo de la serie.
- 5) Hacia la parte inferior derecha se muestran once botones que permiten realizar las diferentes actividades con la serie que se está observando:
 - Primera, para ir la primera serie.
 - Anterior, para movernos a la serie anterior.

- Próxima, para ir a la siguiente serie.
- Ultima, para movernos a la última serie.
- Buscar, permite buscar una serie en particular dado su código, para lo cual muestra una caja de diálogo donde se pide ingresar el código de la misma (ver figura 14). El programa hará una búsqueda y la presentará, en caso de no conseguirla o cancelar la acción se deja la misma serie que estamos observando.

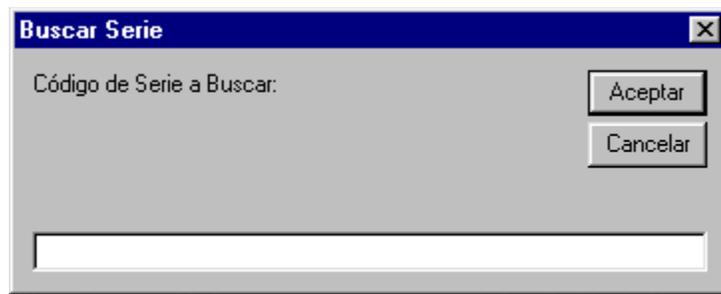


Figura 15. Caja de diálogo para introducir la serie a buscar.

- Criterio, permite cambiar el criterio de recorrido de las diferentes series. Inicialmente no hay definido ningún criterio de recorrido de las series, en este caso el recorrido se realiza secuencialmente desde la serie marcada con el código 1, luego la 2, después la 3 y así sucesivamente hasta la última.

Al ejecutarse esta opción el sistema solicita que el usuario suministre, mediante un cuadro de texto (fig 15), una palabra para así buscar en la base de datos aquellas series que contenga en su título la secuencia que se le suministró; de no encontrarse alguna serie que cumpla con este requisito se mostrará un error y vuelve al criterio original en el cual se presentan todas las series. Si desea volver al recorrido secuencial inicial presione la tecla "Enter" cuando se le solicite el nuevo criterio.

Si el criterio de búsqueda es aceptado entonces el recorrido se hace ahora secuencial pero dentro de aquellas series que contengan la palabra

especificada. Como ejemplo supóngase que se desea buscar las series que contengan la palabra “PETROLEO” en su título, la primera serie que contiene esta palabra en su título es la 18 que se titula “PIB DEL SECTOR PETROLEO”, si le damos al botón “Próxima” se muestra la 98, luego la 100, 101, 102, etc hasta la última que es la 622.

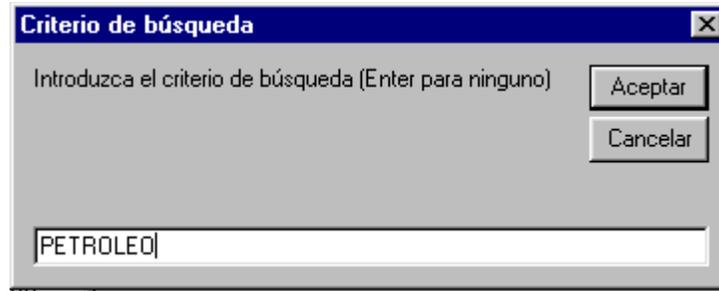


Figura 16. Caja de texto para solicitar el nuevo criterio de recorrido de las series.

- Almacenar, permite guardar la serie, en un archivo tipo texto en el directorio actual con el nombre sugerido de “Seriei.txt” donde la i corresponde al código de la serie (ver figura 16).

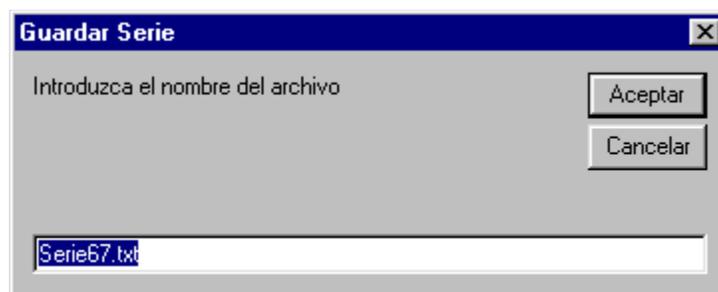


Figura 17. Cuadro de diálogo para introducir el nombre de la serie a almacenar.

El archivo que se crea contiene la siguiente información: Título de la serie, Tipo de período (anual, quinquenal, etc.), Unidad de los datos (personas, bolívares, etc.), el Título del cuadro al cual pertenece la serie y luego dos columnas, una para el período y otra para el valor de la serie en el respectivo período

- Gráfico, muestra un gráfico de dispersión XY, en el cual el eje X representa los años o períodos de la serie y el eje Y el valor de la serie. Un ejemplo de un gráfico es presentado en la figura 17.

Estando el gráfico presente, si se presiona el botón Gráfico este desaparece como un botón de activar y desactivar, continuando en la misma serie.

Si presiona un botón que mueva la serie actual (Primera, Anterior, Próxima, Última o Buscar) el gráfico desaparece y aparece la nueva serie con sus datos.

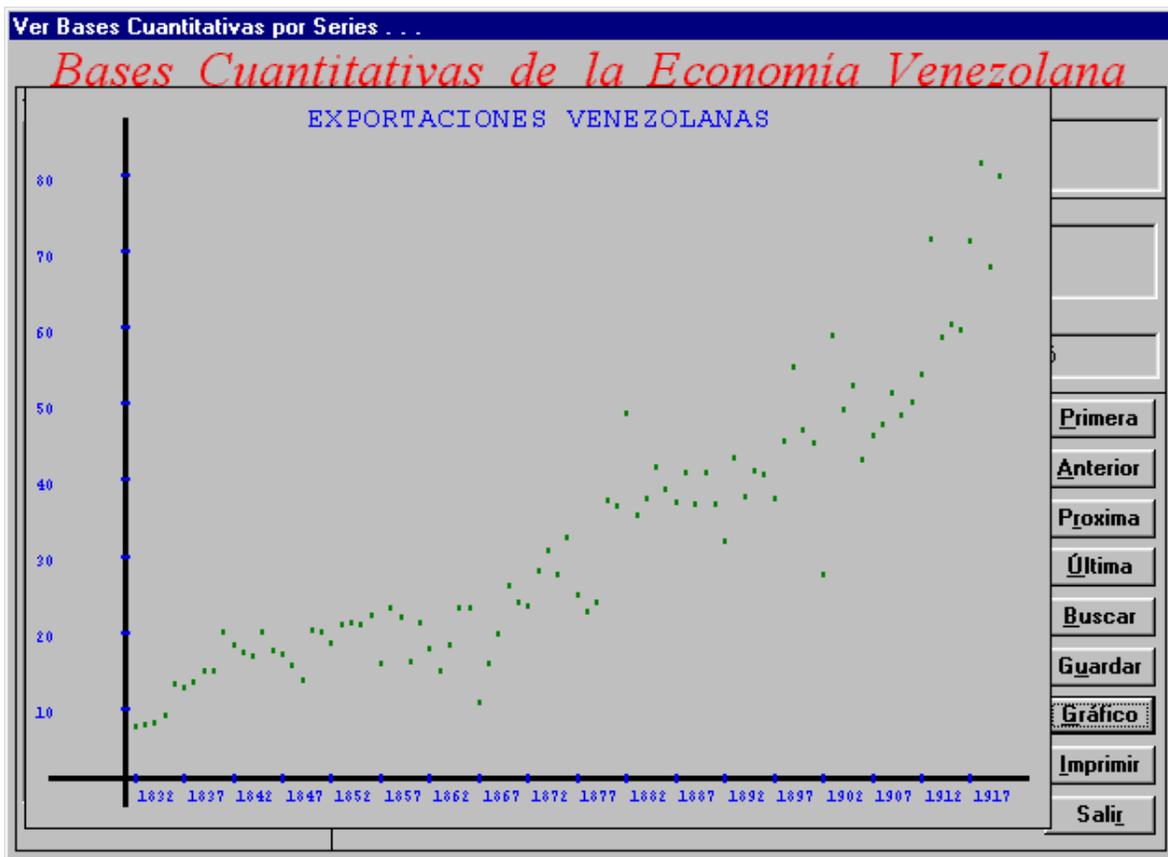


Figura 18. Ejemplo de gráfico de dispersión de una serie.

- Imprimir, permite generar un reporte en forma de texto la serie que se muestra actualmente (activa) en la impresora Como encabezado imprime el

título de la serie, el año de comienzo, período, la unidad en que se encuentran los datos y el título del cuadro al que pertenece la serie con su respectivo año de comienzo y final (ver Anexo 4).

Como cuerpo propiamente de la serie se imprime el período y su respectivo valor desde el comienzo hasta el final de la misma distribuidos en tres grupos. Si es necesaria otra página se vuelve a imprimir el encabezado con los respectivos valores.

- La opción presentada en el botón Arima permite realizar o correr un modelo ARIMA dados sus parámetros en los correspondientes cuadros de texto que se muestran en la ventana Análisis ARIMA, estos parámetros son: el autoregresivo (p), la diferenciación (d) y la media móvil (q), ver figura 18.

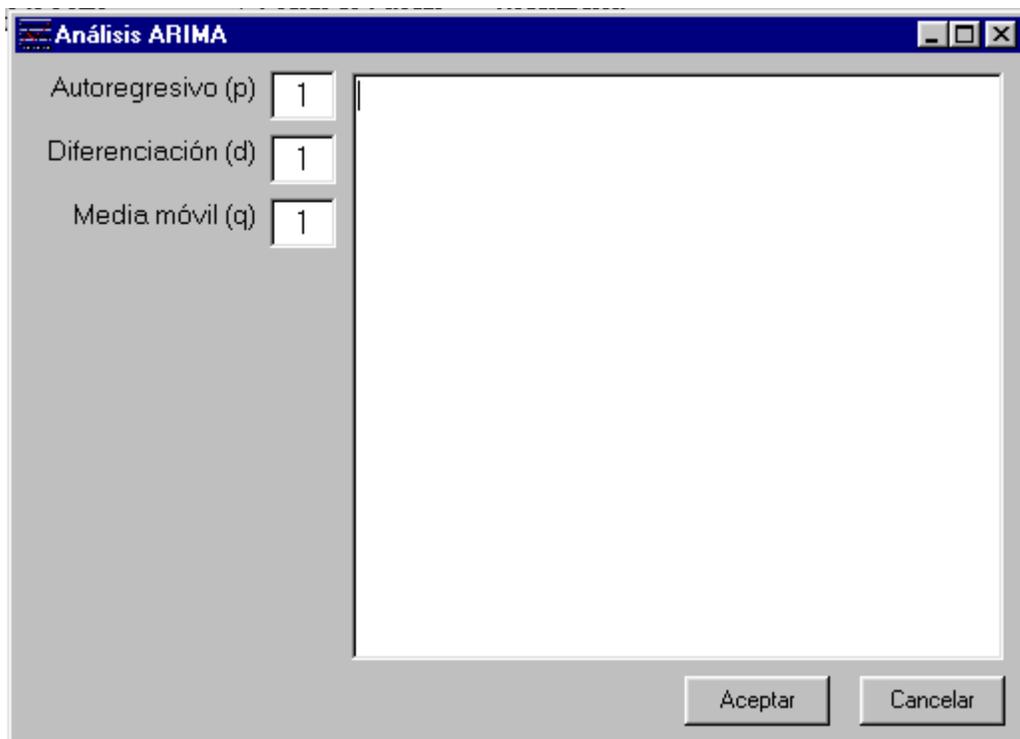


Figura 19. Ventana Análisis ARIMA.

Una vez llenados los parámetros del modelo ARIMA que se desea correr se presiona el botón Aceptar para que el programa mediante Intercambio Dinámico de Datos invoque al S-PLUS le envíe los datos que necesite y le

devuelva los resultados del modelo. Si no desea realizar análisis alguno, o en caso de que el programa no consiga el S-Plus en el sistema y le dé un mensaje de error, presione el botón Cancelar para regresar a la ventana Ver Series.

Una vez ejecutado el modelo ARIMA solicitado al S-PLUS y este haya regresado los resultados, los mismos se muestran en el cuadro de texto que aparece en la parte derecha, como se muestra en la figura 19. Junto con los resultados se activa el botón Imprimir para enviar estos resultados a la impresora.

- La opción Salir, regresar al menú o ventana principal de Bases Cuantitativas.

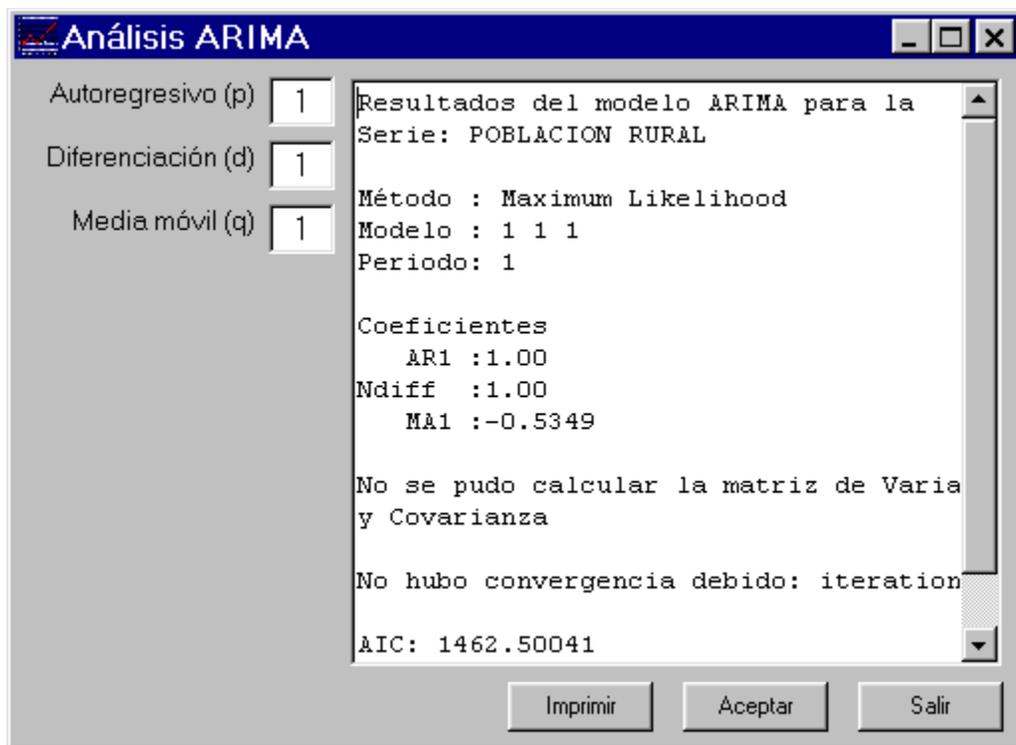


Figura 20. Resultado de la corrida del modelo ARIMA a los datos de una serie.

3.7. Bases cuantitativas en la Web.

El sitio Web en el cual se encuentra las “Bases Cuantitativas” tiene la siguiente dirección URL: <http://hotelling.faces.ula.ve/bases>.

Este fue construido con el FrontPage 98, el cual es un programa de Microsoft que permite ir construyendo en forma “Visual” las páginas y automáticamente genera el código HTML necesario. Al código HTML generado se le agrega las instrucciones necesarias para la consulta de la base de datos.

Las bases de datos se accesan mediante conexiones de tipo ODBC (Open Database Connectivity, Conectividad abierta de bases de datos), una interfaz de programación de aplicaciones que les permite tener acceso a datos desde diversas especificaciones estándares de orígenes de datos, y usando el lenguaje estructurado de consultas (SQL). [7].

Una vez inicializado un navegador, preferiblemente el Microsoft Internet Explorer, e introducida la dirección URL (<http://hotelling.faces.ula.ve/bases>), se presentará la pantalla de bienvenida, en la misma existen dos vínculos activos: uno para hacer el recorrido del libro por Cuadros y el otro para hacerlo por Series.

Si activa o selecciona el primer vínculo (recorrido por Cuadros) el navegador presentará una pantalla aproximadamente igual a la mostrada en la figura 20; en la misma pueden observarse las diferentes características de cada uno de los cuadros del libro comenzando por el primero. Las características que se presentan son: Código, Mnemónico del cuadro en el libro, Título, Año de inicio, año final, a qué capítulo pertenece y por último, en forma de un cuadro de lista, las series asociadas al mismo.

En la parte inferior puede observarse 4 botones de navegación a través de los cuadros, a saber, Ir al primer cuadro , Ir al cuadro anterior, Ir al siguiente cuadro e Ir al último cuadro; luego hay un botón (Mostrar Series) para ver los valores de las series del cuadro, y por último un vínculo para regresar a la página de bienvenida.

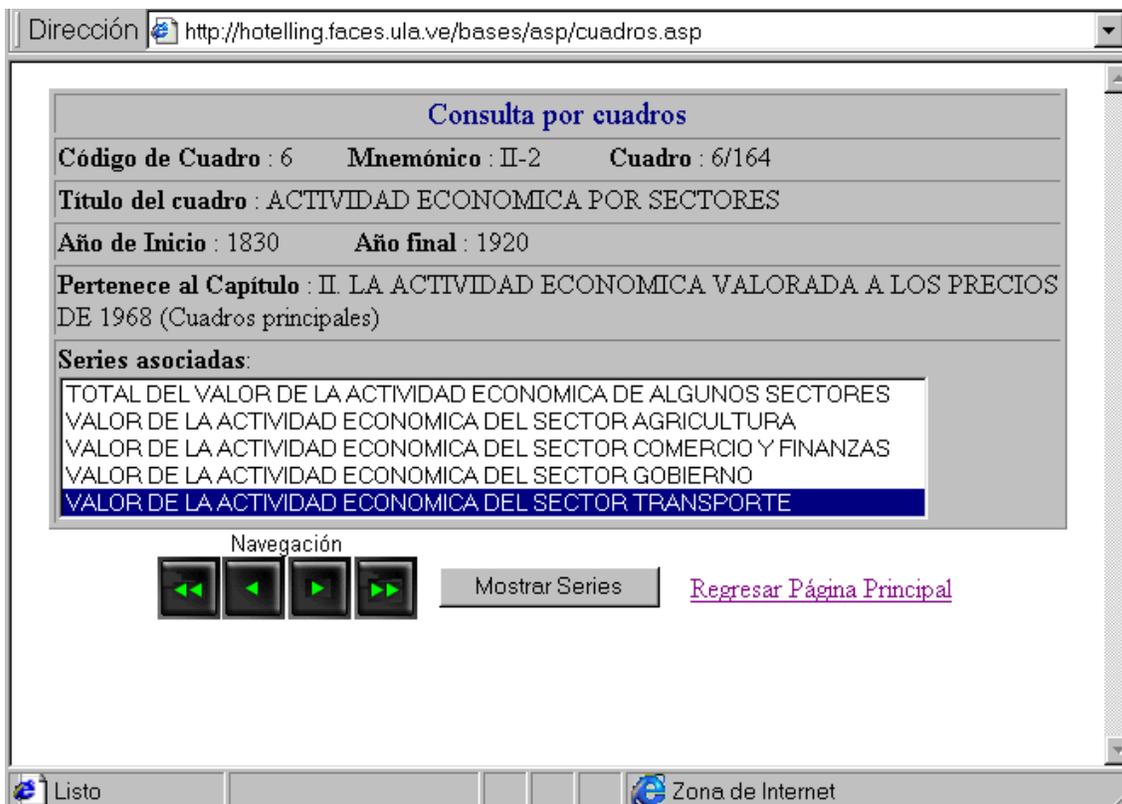


Figura 21. Mostrar Cuadros en el navegador WEB.

Al presionar (activar) el botón **Mostrar Series**, se presenta la página que se muestra en las figuras 22 y 23, en la cual aparece un encabezado con el nombre del cuadro y a continuación en forma de una tabla donde la primera fila contiene la palabra **Serie***i* (*i* es un número consecutivo desde 1 hasta el número de series que contiene el cuadro).

Al final de esta página, después de la tabla de datos, se presenta la leyenda de cada una de las series.

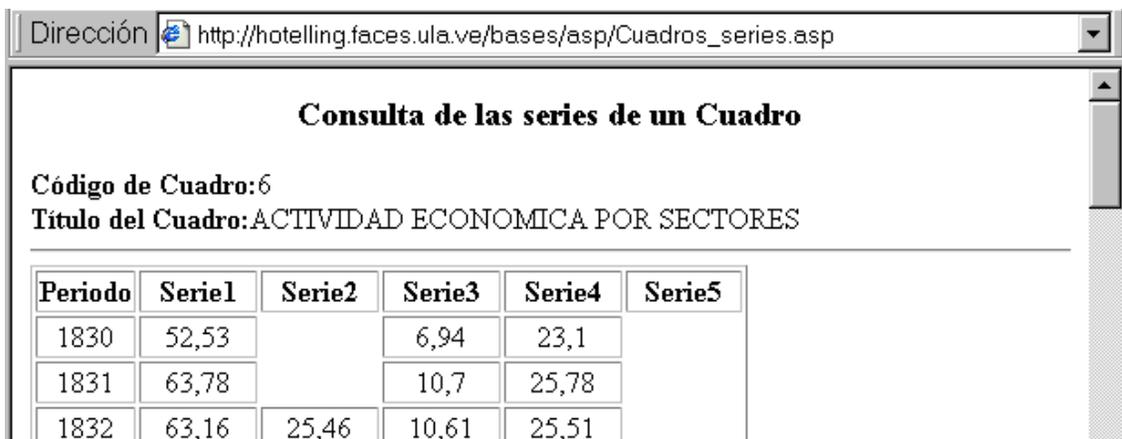


Figura 22. Parte superior de la pantalla Mostrar Series en el navegador WEB.

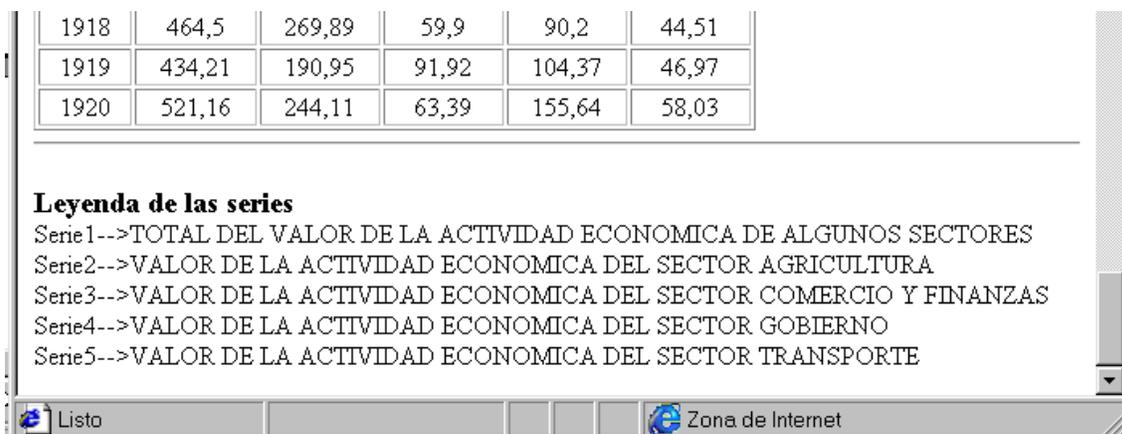


Figura 23. Parte final de la pantalla Mostrar Series en el navegador WEB.

Si en la página de bienvenida se selecciona el vínculo para recorrer el libro por series, se mostrará en el navegador la pantalla que se presenta en la figura 23. En la misma puede observarse las características de las series comenzando desde la primera, así como también una lista con los valores de cada una.

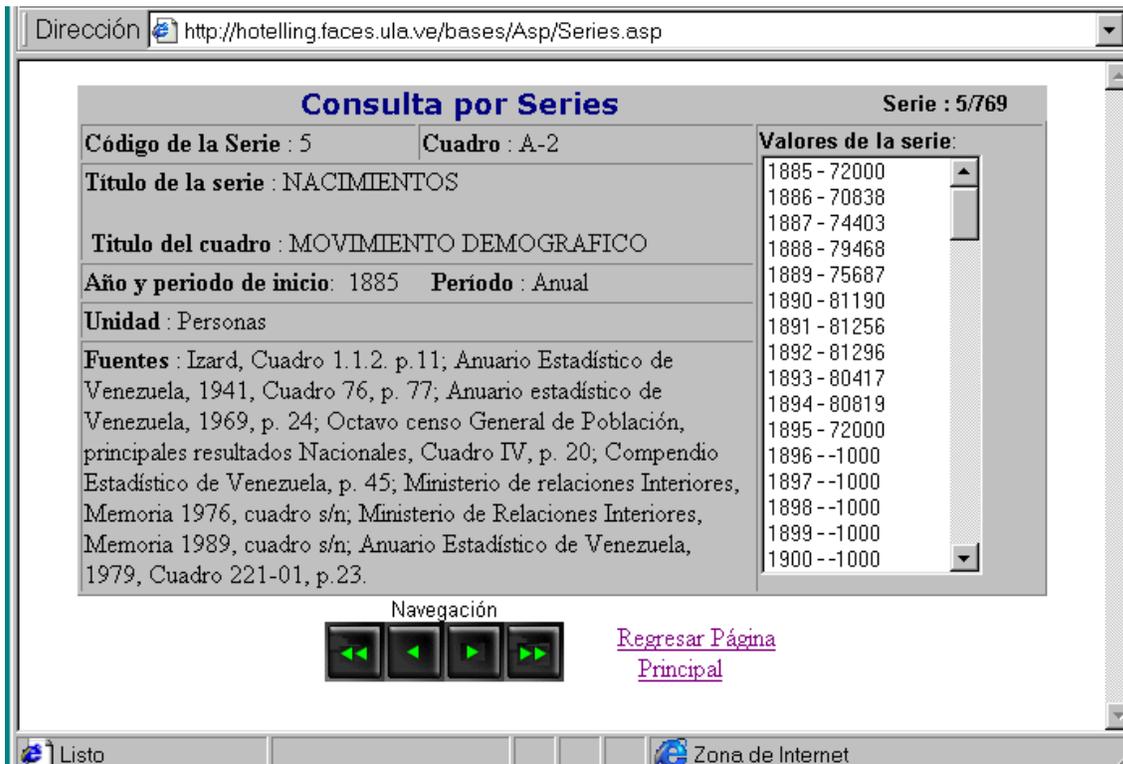


Figura 24. Consulta por Series en el navegador WEB.

CONCLUSIONES

- 1) Las bases de datos como medio de almacenamiento permiten la recuperación rápida y eficiente de información tal y como se muestra en el presente trabajo con las “Bases Cuantitativas de la Economía Venezolana”.
- 2) La base de datos de la Economía Venezolana queda implementada hasta 1989, pudiendo ser actualizada hasta el presente. Para algunas de las series se actualizó hasta el año 1999. También se le pueden agregar mas series o conjunto de series (tablas), así como también conjunto de tablas (capítulos), como muestra de esto, la base de datos contiene un capítulo que se denomina “Índice de Precios al Consumidor para el Área Metropolitana de Caracas”, que contiene series de datos mensuales desde enero de 1999 hasta enero de 2000.
- 3) El uso de la técnica de Intercambio Dinámico de Datos entre diversas aplicaciones permite realizar análisis y procesos que ya están implementados en los paquetes estadísticos.
- 4) En el trabajo sólo se utilizó los procedimientos que tienen que ver con la técnica ARIMA, pero el S-PLUS permite DDE con cualquier procedimiento inclusive gráfico, lo que se podría aprovechar en el futuro para realizar otros análisis.

BIBLIOGRAFIA

- [1] Asdrubal Baptista. “*Bases Cuantitativas de la Economía Venezolana. 1830-1989*”
- [2] Alfons González. (1996). *Programación de bases de datos con Visual Basic*. RA-MA Editorial. 1996.
- A. Silberschatz – H.F. Korth – S. Sudarshan. *Fundamentos de Bases de Datos*. Tercera. edición Mc. Graw Hill. 1998.
- [4] Box y Jenkins. *Análisis de Series de Tiempo*, Holden Day. 1976.
- [5] C. Batini –S. Ceri –S. Navathe.. *Diseño conceptual de bases de datos. Un enfoque de entidades-interrelaciones*. Adison-Wesley/Diaz de Santos. 1994.
- [6] C.J. Date. *Introducción a los Sistemas de Bases de Datos. V 1*. Quinta Edición. Adison-Wesley. 1990.
- [7] *Documentación del Personal Web Server de Windows 98*.
- [8] F.M. Morcillo – R.I. Aparicio. *Diccionario de términos financieros y de inversión*. Mc Graw Hill. 1995.
- [9] J.C. Poindexter. *Macroeconomía*. 2da Edición. Editorial Interamericana. 1984.
- [10] *Literacia de computadores* <http://www.cayey1.upr.clu.edu/sici3006/curso.htm>
- [11] Richar Barker. *El modelo entidad-relación CASE*METHOD*. Adison-Wesley/Diaz de Santos. 1990.
- [12] S-Plus Programmer’s Guide. Versión 4.5. MathSoft. 1998.
- [13] Sergio Rios Aguilar. *Lenguajes HTML, JAVA y CGI. El diseño de páginas web para Internet a su alcance*. Abeto Editorial. 1996.

ANEXOS

Anexo 1. Ejemplos de cuadro del libro (Cuadro I-2 y Cuadro II-2)

CUADRO I-2

Población urbana y rural: 1920-1989

	Urbana	Rural		Urbana	Rural
1920	490765	2501703	1955	2892056	3261255
1921	501552	2506570	1956	3081783	3338598
1922	513325	2511436	1957	3269575	3416672
1923	532486	2516303	1958	3465355	3479244
1924	555980	2521710	1959	3669962	3540179
1925	588398	2526036	1960	3869262	3604499
1926	621459	2530903	1961	4126402	3659262
1927	649637	2535770	1962	4382612	3703388
1928	680122	2540636	1963	4648310	3742141
1929	713753	2545503	1964	4926316	3777422
1930	749844	2550370	1965	5225866	3799810
1931	780496	2555237	1966	5536043	3815381
1932	808337	2560103	1967	5865822	3829752
1933	835616	2564970	1968	6207481	3836986
1934	860962	2569836	1969	6551846	3831428
1935	890290	2574703	1970	6880986	3837061
1936	930048	2574570	1971	7206883	3846618
1937	980379	2584635	1972	7534770	3864277
1938	1036264	2587038	1973	7859676	3888719
1939	1098616	2600427	1974	8191605	3926154
1940	1169188	2614592	1975	8540496	3982225
1941	1234453	2623213	1976	8911739	4022570
1942	1306230	2628198	1977	9285594	4055777
1943	1378838	2641099	1978	9700554	4078642
1944	1456358	2657648	1979	10137280	4100473
1945	1541400	2681614	1980	10571684	4131632
1946	1634417	2712437	1981	11009316	4175947
1947	1735999	2749786	1982	11451533	4235499
1948	1857606	2798049	1983	11875259	4303568
1949	1985792	2857603	1984	12293863	4364488
1950	2114632	2920206	1985	12708802	4442078
1951	2253842	2975490	1986	13109339	4510740
1952	2393088	3033415	1987	13497270	4571364
1953	2546937	3100373	1988	13913564	4613152
1954	2710515	3181909	1989	14373664	4639119

CUADRO II-2

Actividad económica por sectores: 1930-1875
(Millones de bolívares de 1936)

	Total	Agricultura	Comercio y Finanzas	Gobierno	Transporte
1830	52.5		6.9	23.1	
1831	63.8		10.7	25.8	
1832	63.2	25.5	10.6	25.5	
1833	63.4	24.1	10.8	26.9	
1834	64.5	24.8	13.9	24.2	
1835	68.0	29.7	12.6	24.0	
1836	94.2	42.5	17.9	31.4	
1837	91.9	49.5	13.1	27.0	
1838	89.4	44.6	17.0	25.6	
1839	106.5	48.7	22.3	32.8	
1840	104.8	52.0	26.6	23.5	
1841	123.1	70.0	25.9	24.1	
1842	133.8	64.2	22.9	43.4	
1843	110.0	58.9	21.3	27.1	
1844	121.9	54.6	22.9	41.4	
1845	128.6	69.6	24.6	31.2	
1846	136.3	64.4	21.9	46.6	
1847	106.9	57.2	17.0	30.0	
1848	114.6	63.9	14.6	33.2	
1849	113.3	52.5	19.5	38.4	
1850	143.2	76.3	25.6	37.8	
1851	144.8	75.7	24.7	40.7	
1852	117.6	63.9	19.5	31.2	
1853	146.2	68.8	21.6	52.1	
1854	153.4	76.7	21.4	51.4	
1855	170.1	74.3	20.0	71.6	
1856	150.7	65.6	24.8	56.7	
1857	132.2	55.2	22.2	51.5	
1858	164.3	86.8	28.5	44.9	
1859	125.5	73.2	24.5	24.6	
1860	140.3	51.7	17.0	68.1	
1861	170.5	69.5	17.8	78.9	
1862	146.5	71.1	14.1	57.6	
1863	147.5	48.5	11.1	26.8	
1864	148.5	62.0	25.9	57.0	
1865	170.9	80.1	25.3	61.2	
1866	148.2	74.4	21.2	48.9	
1867	134.3	36.3	13.0	33.8	
1868	120.5	52.8	19.7	45.0	
1869	134.6	69.8	13.6	47.9	
1870	139.0	94.5	19.7	22.0	
1871	123.8	73.0	26.5	21.1	
1872	148.0	84.0	27.5	32.8	
1873	159.4	98.4	21.0	36.0	
1874	200.4	109.0	33.9	52.5	
1875	191.5	96.0	44.8	46.0	

Anexo 2. Glosario de términos económicos [8] y [9]

- Acervo: conjunto de valores morales, culturales, sociales, etc de una colectividad. Patrimonio de una colectividad.
- Año base: es el año de referencia al que se asigna un valor de 100 cuando se construye el índice.
- Deflación: situación caracterizada por un descenso general de precios. Medidas económicas encaminadas a combatir la inflación. La principal medida tomada en épocas de deflación es la reducción de la circulación fiduciaria cuando ha adquirido excesivo volumen por efecto de la inflación.
- Deflactor del PIB: es la relación entre el PIB nominal y el PIB real expresada en forma de índice. Es la medida de inflación mas utilizada después del Índice de Precios al Consumidor. La diferencia entre ambos índices se encuentra en los bienes sobre los que se aplica, ya que, mientras el IPC sólo incluye los precios de los bienes adquiridos por el consumidor típico, el deflactor del PIB tiene en cuenta los precios de todos los bienes producidos en la economía.
- Formación Bruta de Capital: inversión real en construcción, bienes de equipo y material de transporte, más variación de existencia.
- Índice: es un cálculo estadístico basado en las variaciones de un grupo determinado de precios de acuerdo con un determinado criterio de ponderación. Generalmente, el número índice se expresa como un porcentaje del valor que tenía en el año base.
- Índice de Precios al Consumidor (IPC): promedio ponderado de los precios de los bienes y servicios consumidos por las familias.
- Inversión: utilización de una parte de la producción corriente para aumentar el stock de capital.
- Producto Interno Bruto:

Anexo 3. Ejemplo de Imprimir las series de un cuadro.

Cuadro :II-2

Título :ACTIVIDAD ECONOMICA POR SECTORES
(1830 , 1920)

Capítulo :LA ACTIVIDAD ECONOMICA VALORADA A LOS PRECIOS DE 1968 (Cuadros principa

Leyenda de las columnas

Col1 -> TOTAL DEL VALOR DE LA ACTIVIDAD ECONOMICA DE ALGUNOS SECTORES
Col2 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR AGRICULTURA
Col3 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR COMERCIO Y FINANZAS
Col4 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR GOBIERNO
Col5 -> VALOR DE LA ACTIVIDAD ECONOMICA DEL SECTOR TRANSPORTE

PERIODO	Col1 (Mill.Bs.1936)	Col2 (Mill.Bs.1936)	Col3 (Mill.Bs.1936)	Col4 (Mill.Bs.1936)	Col5 (Mill.Bs.1936)
1830	52,53			6,94	23,10
1831	63,78			10,70	25,78
1832	63,16	25,46		10,61	25,51
1833	63,42	24,14		10,79	26,91
1834	64,51	24,80		13,93	24,17
1835	68,01	29,68		12,63	24,00
1836	94,17	42,52		17,92	31,38
1837	91,87	49,47		13,09	27,02
1838	89,38	44,60		16,96	25,59
1839	106,48	48,71		22,28	32,83
1840	104,75	52,04		26,64	23,47
1841	123,12	70,02		25,89	24,14
1842	133,83	64,16		22,89	43,45
1843	110,04	58,93		21,28	27,10
1844	121,91	54,64		22,86	41,37
1845	128,59	69,56		24,64	31,18
1846	136,33	64,38		21,92	46,64
1847	106,86	57,19		16,97	30,03
1848	114,58	63,89		14,61	33,22
1849	113,27	52,52		19,53	38,40
1850	143,24	76,26		25,64	37,77
1851	144,82	75,75		24,74	40,73
1852	117,58	63,94		19,50	31,22
1853	146,17	68,78		21,62	52,13
1854	153,43	76,72		21,45	51,44
1855	170,12	74,27		19,99	71,62
1856	150,73	65,58		24,75	56,65
1857	132,23	55,24		22,18	51,51
1858	164,30	86,79		28,52	44,90
1859	125,45	73,20		24,51	24,62
1860	140,25	51,66		17,05	68,06
1861	170,51	69,53		17,82	78,91
1862	146,47	71,09		14,08	57,65
1863	147,51	48,48		11,11	26,81
1864	148,55	62,01		25,87	56,97
1865	170,94	80,15		25,30	61,23
1866	148,19	74,40		21,23	48,86
1867	134,32	36,29		13,00	33,76
1868	120,45	52,76		19,68	45,01
1869	134,62	69,81		13,57	47,88
1870	139,75	94,48		19,74	22,05
1871	123,77	73,03		26,54	21,13
1872	148,00	84,04		27,52	32,75
1873	159,41	98,43		20,99	36,02
1874	200,39	108,97		33,93	52,50
1875	191,48	95,96		44,75	46,01

1876	214,95	112,26	47,30	50,03	
1877	182,13	91,54	42,42	43,63	
1878	167,81	73,04	31,28	59,30	
1879	167,85	80,54	36,24	46,88	
1880	209,50	122,10	37,57	44,61	
1881	246,44	136,08	45,21	59,01	
1882	291,75	170,87	64,65	48,97	
1883	251,29	122,02	52,91	70,10	6,26
1884	261,77	138,75	48,66	66,85	7,51
1885	280,53	155,71	54,66	59,73	10,43
1886	264,71	132,34	62,41	54,94	15,02
1887	293,14	132,21	64,71	80,65	15,57
1888	303,97	146,54	66,90	71,07	19,47
1889	295,50	126,70	67,00	86,22	15,57
1890	314,74	146,31	56,33	92,35	19,75
1891	285,70	125,97	58,15	84,20	17,38
1892	248,24	117,26	46,67	62,89	21,41
1893	347,40	161,13	64,41	97,95	23,92
1894	299,74	136,78	58,24	82,90	21,83
1895	346,77	142,76	59,13	124,72	20,16
1896	310,30	138,44	65,72	89,87	16,27
1897	285,31	134,37	45,72	86,31	18,91
1898	322,19	168,08	67,41	67,64	19,05
1899	320,58	203,15	55,91	46,09	15,43
1900	310,22	165,34	51,10	72,23	21,55
1901	305,08	164,12	57,05	60,96	22,94
1902	331,87	97,90	27,59	43,76	21,14
1903	358,67	204,67	58,59	70,53	24,89
1904	350,23	183,02	47,87	93,34	26,00
1905	346,10	183,95	46,24	90,47	25,45
1906	323,04	145,94	45,80	108,64	22,67
1907	322,84	159,70	47,34	86,74	29,06
1908	345,21	184,76	45,81	82,80	31,84
1909	357,35	179,36	49,45	89,88	38,66
1910	368,93	164,08	60,83	104,66	39,35
1911	394,75	173,11	76,67	107,56	37,40
1912	408,10	193,36	73,23	103,55	37,96
1913	471,34	250,32	72,00	106,47	42,55
1914	411,97	236,94	51,52	72,75	50,75
1915	424,63	222,24	63,00	92,95	46,44
1916	400,80	187,67	70,85	92,08	50,20
1917	467,63	285,46	48,22	83,05	50,89
1918	464,50	269,89	59,90	90,20	44,51
1919	434,21	190,95	91,92	104,37	46,97
1920	521,16	244,11	63,39	155,64	58,03

Anexo 4. Ejemplo de Imprimir una serie.

Serie :INDICE DEL NIVEL GENERAL DE LA ACTIVIDAD ECONOMICA (1920)
 PERIODO :Anual
 UNIDAD :1968=100

Cuadro :INDICE DEL NIVEL GENERAL DE LA ACTIVIDAD ECONOMICA (1830 , 1989)

PERIODO	VALOR	PERIODO	VALOR
Período	Valor	1965	86,71
1920	3,41	1966	88,27
1921	3,55	1967	93,61
1922	3,65	1968	100,00
1923	4,21	1969	103,59
1924	4,87	1970	111,56
1925	6,30	1971	116,01
1926	7,62	1972	120,37
1927	8,55	1973	127,56
1928	9,57	1974	133,11
1929	10,85	1975	141,58
1930	11,05	1976	154,16
1931	8,93	1977	166,52
1932	8,55	1978	172,90
1933	9,36	1979	172,38
1934	9,99	1980	167,27
1935	10,71	1981	168,49
1936	11,76	1982	172,96
1937	13,50	1983	167,05
1938	14,59	1984	160,86
1939	15,47	1985	164,15
1940	14,87	1986	172,54
1941	14,63	1987	182,67
1942	12,79	1988	196,92
1943	13,96	1989	178,41
1944	17,23		
1945	20,94		
1946	25,12		
1947	30,05		
1948	33,45		
1949	35,50		
1950	36,32		
1951	38,85		
1952	42,24		
1953	43,87		
1954	48,41		
1955	52,46		
1956	57,01		
1957	65,50		
1958	66,60		
1959	70,60		
1960	70,83		
1961	68,64		
1962	71,67		
1963	74,95		
1964	81,32		

Anexo 5. Listado de los programas de Visual Basic.

` **Declaraciones Generales**

```
Option Explicit
Global Nv, Nvl As Integer
Global Periodo(500) As Integer
Global Valores(500) As Single
Global Minimo, Maximo, Media, Varianza, Desv As Single
Global SaltoLinea as String
Global Margen as String
Global TamFuente as String
```

` **Declaraciones del formulario Principal**

```
Private Sub BotonCuadros_Click()
    FormPrincipal.Visible = False
    Form1Principal.Enabled = False
    FormCuadros.Enabled = True
    FormCuadros.Visible = True
End Sub
```

```
Private Sub BotonSeries_Click()
    FormPrincipal.Visible = False
    FormPrincipal.Enabled = False
    FormSeries.Enabled = True
    FormSeries.Visible = True
End Sub
```

```
Private Sub MenuCuadro_Click()
    FormPrincipal.Visible = False
    FormPrincipal.Enabled = False
    FormCuadros.Enabled = True
    FormCuadros.Visible = True
End Sub
```

```
Private Sub MenuSerie_Click()
    Form1.Visible = False
    Form1.Enabled = False
    FormSeries.Enabled = True
    FormSeries.Visible = True
End Sub
```

` **Declaraciones del formulario Ver Cuadros**

```
Dim CodSeries(30) As Integer
Dim NomSeries(30) As String
Dim NSeries, VerSerie, GuardaSerie, Nv As Integer
```

```
Private Sub BotónPrimero_Click()
    DataCuadros.Recordset.MoveFirst
End Sub
```

```
Private Sub BotónAnterior_Click()
    DataCuadros.Recordset.MovePrevious
    If DataCuadros.Recordset.BOF Then
        DataCuadros.Recordset.MoveFirst
    End If
End Sub
```

```
Private Sub BotónSiguiente_Click()
    DataCuadros.Recordset.MoveNext
    If DataCuadros.Recordset.EOF Then
        DataCuadros.Recordset.MoveLast
    End If
End Sub
```

```

Private Sub BotónUltimo_Click()
    DataCuadros.Recordset.MoveLast
End Sub

Private Sub BotónVerSeries_Click()
    If GuardaSerie = 1 Then
        NomArchivo$ = ""
        NomArchivo$ = InputBox("Introduzca el nombre del archivo", "Guardar Tabla",
"Tabla" & Label04.Caption & ".txt")
        If NomArchivo$ = "" Then Exit Sub
        p = InStr(1, NomArchivo$, ".")
        If p <> 0 Then
            NomArchivo$ = Mid(NomArchivo$, 1, p - 1)
        End If
        NomArchivo$ = NomArchivo$ & ".txt"
        BotónVerSeries.Enabled = False
        BotónSalir.Enabled = False
        Open NomArchivo$ For Output As #1
        With DataCuadros.Recordset
            Print #1, "Cuadro " & Label4.Caption & "-" & .Fields("NUMERO").Value
            Print #1,
                Print #1, "Titulo : " & .Fields("Titulo").Value & " (" &
.Fields("A_Inicio").Value _
                & " , " & .Fields("A_fin").Value & ")"
        End With
        Print #1,
            cad$ = " PERIODO "
            cad1$ = " "
        Grid1.Row = 0
        For j = 1 To NSeries
            Grid1.Col = j
            cad$ = cad$ + Space(8) + Grid1.Text + Space(5)
            Criterio$ = "COD_SERIE=" & CodSeries(j)
            DataSeries.Recordset.FindFirst Criterio$
            Criterio$ = "COD_UNIDAD=" &
DataSeries.Recordset.Fields("COD_UNIDAD").Value &
            DataUnidad.Recordset.FindFirst Criterio$
            f$ = "(" + DataUnidad.Recordset.Fields("ABREVIADO").Value + ")"
            cad1$ = cad1$ + Space(15 - Len(f$)) + f$ + Space(2)
        Next j
        Print #1, cad$
        Print #1, cad1$
        raya$ = String(Len(cad$) + 5, "-")
        Print #1, raya$
        For i = 1 To Nv
            Grid1.Row = i
            Grid1.Col = 0
            cad$ = " " + Str(Grid1.Text) + Space(4)
            For j = 1 To NSeries
                Grid1.Col = j
                f$ = Format(Grid1.Text, "###,###,###.00")
                cad$ = cad$ + Space(15 - Len(f$)) + f$ + Space(2)
            Next j
            Print #1, cad$
        Next i
        Print #1, raya$
        Print #1, "Leyenda de las columnas"
        Print #1, "-----"
        Print #1,
            For i = 1 To NSeries
                Print #1, List1.List(i - 1)
            Next i
        Close #1
        f$ = MsgBox("Tabla Almacenada con el nombre : " + NomArchivo$, vbOKOnly,
"Almacenar")
    
```

```

BotónVerSeries.Enabled = True
BotónSalir.Enabled = True
Else
BotónPrimero.Enabled = False
BotónAnterior.Enabled = False
BotónSiguiente.Enabled = False
BotónUltimo.Enabled = False
BotónVerSeries.Enabled = False
BotónSalir.Enabled = False
BotónVerSeries.Caption = "&Almacenar"
BotónSalir.Caption = "&Cerrar"
VerSerie = 1
GuardaSerie = 1
Criterio$ = "COD_SERIE=" & CodSeries(1)
DataSeries.Recordset.FindFirst Criterio$
Criterio$ = "COD_PERIOD=" & DataSeries.Recordset.Fields("COD_PERIOD").Value
DataPeriodo.Recordset.FindFirst Criterio$
numval = DataPeriodo.Recordset.Fields("Num_Valore")
With DataCuadros.Recordset
  AFin = .Fields("A_FIN").Value
  AInicio = .Fields("A_INICIO").Value
End With
Nv = (AFin - AInicio + 1) * numval
MSFlexGrid1.Cols = NSeries + 1
MSFlexGrid1.Rows = Nv + 1
For i = 0 To MSFlexGrid1.Cols - 1
  MSFlexGrid1.Col = i
  For j = 0 To MSFlexGrid1.Rows - 1
    MSFlexGrid1.Row = j
    MSFlexGrid1.Text = Space(10)
  Next j
Next i
fila = 1
MSFlexGrid1.Col = 0
MSFlexGrid1.Row = 0
MSFlexGrid1.Text = "Período"
MSFlexGrid1.ColWidth(0) = 800
For i = AInicio To AFin
  For j = 1 To numval
    MSFlexGrid1.Row = fila
    MSFlexGrid1.Text = i + (j - 1) / numval
    fila = fila + 1
  Next j
Next i
MSFlexGrid1.Row = 0
For i = 1 To NSeries
  MSFlexGrid1.ColWidth(i) = 1350
  List1.List(i - 1) = "Col" & i & " -> " & List1.List(i - 1)
  MSFlexGrid1.Col = i
  MSFlexGrid1.Text = "Col" & i
  MSFlexGrid1.FixedAlignment(i) = 4
  MSFlexGrid1.ColAlignment(i) = 1
Next i
If NSeries < 6 Then
  MSFlexGrid1.Width = 1350 * (NSeries + 1)
Else
  MSFlexGrid1.Width = 9200
End If
For i = 1 To NSeries
  DataSerieValor.Recordset.MoveFirst
  Criterio$ = "COD_SERIE=" & CodSeries(i)
  DataSeries.Recordset.FindFirst Criterio$
  Criterio$ = "COD_PERIOD=" &
DataSeries.Recordset.Fields("COD_PERIOD").Value
  DataPeriodo.Recordset.FindFirst Criterio$
  incre = DataPeriodo.Recordset.Fields("incremento")
  Criterio$ = "COD_SERIE=" & CodSeries(i)

```

```

j = 0
MSFlexGrid1.Col = 0
Enc = False
While Not Enc
    j = j + 1
    MSFlexGrid1.Row = j
    Enc = (MSFlexGrid1.Text = DataSeries.Recordset.Fields("A_INICIO").Value)
Wend
MSFlexGrid1.Col = i
DataSerieValor.Recordset.FindFirst Criterio$
While Not DataSerieValor.Recordset.NoMatch
    MSFlexGrid1.Row = j
    ValorX = DataSerieValor.Recordset.Fields("VALOR").Value
    If ValorX <> -1000 Then
        f$ = Format(ValorX, "###,###.00")
    Else
        f$ = "N.D."
    End If
    MSFlexGrid1.Text = f$
    DataSerieValor.Recordset.FindNext Criterio$
    j = j + incre
Wend
Next i
MSFlexGrid1.Visible = True
BotónPrimero.Visible = False
BotónAnterior.Visible = False
BotónSiguiente.Visible = False
BotónUltimo.Visible = False
BotónVerSeries.Enabled = True
BotónImprimir.Visible = True
BotónSalir.Enabled = True
End If
End Sub

Private Sub DataCuadros_Reposition()
    If Label04.Caption <> nul Then
        DataSeries.Recordset.MoveFirst
        Criterio$ = "COD_CUADRO=" & Label04.Caption
        List1.Clear
        DataSeries.Recordset.FindFirst Criterio$
        NSeries = 0
        While Not DataSeries.Recordset.NoMatch
            f$ = DataSeries.Recordset.Fields("TIT_SERIE").Value
            List1.AddItem f$
            NSeries = NSeries + 1
            CodSeries(NSeries) = DataSeries.Recordset.Fields("COD_SERIE").Value
            NomSeries(NSeries) = DataSeries.Recordset.Fields("TIT_SERIE").Value
            DataSeries.Recordset.FindNext Criterio$
        Wend
        If NSeries = 1 Then
            Labell1.Caption = "Hay " & NSeries & " Serie Asociada al Cuadro"
        Else
            Labell1.Caption = "Hay " & NSeries & " Series Asociadas al Cuadro"
        End If
        DataCapitulo.Recordset.MoveFirst
        Criterio$ = "COD_CAPITU=" & DataCuadros.Recordset.Fields("COD_CAPITU").Value
        DataCapitulo.Recordset.FindFirst Criterio$
    End If
End Sub

Private Sub Encabezado()
    Printer.Print
    Printer.Print
    Printer.Print
    Printer.Font.Size = TamFuente + 2
    With DataCuadros.Recordset
        Printer.FontBold = True
    End With
End Sub

```

```

Printer.Print Margen + "Cuadro :";
Printer.FontBold = False
Printer.Print Label4.Caption & "-" & .Fields("NUMERO").Value
Printer.FontBold = True
Printer.Print Margen + "Titulo :";
Printer.FontBold = False
Printer.Print .Fields("Titulo").Value
Printer.Print Margen & Space(8) & "(" & .Fields("A_Inicio").Value _
    & " , " & .Fields("A_fin").Value & ")"
End With
Printer.FontBold = True
Printer.Print Margen + "Capítulo :";
Printer.FontBold = False
Printer.Print DataCapitulo.Recordset.Fields("Descripcion").Value

Printer.Font.Size = TamFuente
Printer.Print
End Sub

Private Sub BotónImprimir_Click()
Dim cad, cad1 As String

BotónSalir.Enabled = False
BotónVerSeries.Enabled = False
BotónImprimir.Enabled = False

Printer.Font.Name = "Courier New"
Printer.EndDoc
i = 0
Do
    i = i + 1
    Printer.Font.Size = i
Loop Until Printer.TextHeight("AZ") >= 250
TamFuente = i
Printer.Font.Size = TamFuente
NPuntos = Printer.TextHeight("AZ")
NroLineas% = (Printer.Width) / NPuntos + 1

NroBloques = Int((Grid1.Cols - 1) / 4) + 1

For Bloque = 1 To NroBloques
    Call Encabezado
    Printer.Print Margen + "Leyenda de las columnas"
    Printer.Print Margen + "-----"
    cad = " PERIODO "
    cad1 = " "
    Grid1.Row = 0
    j = (Bloque - 1) * 4 + 1
    contador = 0
    While (j <= NSeries) And (contador < 4)
        Printer.Print Margen + List1.List(j - 1)

        Grid1.Col = j
        cad = cad + Space(5) + Grid1.Text + Space(13 - Len(Grid1.Text))
        Criterio$ = "COD_SERIE=" & CodSeries(j)
        DataSeries.Recordset.FindFirst Criterio$
        Criterio$ = "COD_UNIDAD=" & DataSeries.Recordset.Fields("COD_UNIDAD").Value
        DataUnidad.Recordset.FindFirst Criterio$
        f$ = "(" + DataUnidad.Recordset.Fields("ABREVIADO").Value + ")"
        cad1 = cad1 + Space(15 - Len(f$)) + f$ + Space(3)
        contador = contador + 1
        j = j + 1
    Wend
    For j = contador To 4
        Printer.Print
    Next j
    Printer.Print Margen + cad

```

```

Printer.Print Margen + cad1
raya$ = String(Len(cad) + 2, "-")
Printer.Print Margen + raya$

pagina = 1
fila = 1
While fila < Grid1.Rows - 1
    Printer.Font.Name = "Courier New"
    Printer.Font.Size = 1
    Printer.Font.Size = TamFuente
    If pagina = 1 Then
        linea = 12
    Else
        linea = 0
        Printer.Print
        Printer.Print
        Printer.Print
        Printer.Print
    End If
    While (linea < NroLineas%) And (fila <= Grid1.Rows - 1)
        Grid1.Row = fila
        Grid1.Col = 0

        j = (Bloque - 1) * 4 + 1
        cad = Space(3) + Grid1.Text + Space(1)
        contador = 0
        While (j <= NSeries) And (contador < 4)
            Grid1.Col = j
            f$ = Format(Grid1.Text, "###,###,###.00")
            cad = cad + Space(15 - Len(f$)) + f$ + Space(3)
            contador = contador + 1
            j = j + 1
        Wend
        Printer.Print Margen + cad
        fila = fila + 1
        linea = linea + 1
    Wend
    Printer.Print Margen + raya$
    Printer.Print Space(70) + "Pag: "; Bloque; "-"; pagina
    Printer.NewPage
    pagina = pagina + 1
Wend
Next Bloque

Printer.EndDoc
BotónSalir.Enabled = True
BotónVerSeries.Enabled = True
BotónImprimir.Enabled = True
End Sub

Private Sub BotónSalir_Click()
    Dim Respuesta As Integer
    If VerSerie = 0 Then
        Respuesta = MsgBox("Seguro de Salir ?", 36, "Salir de Ver Lista de Series")
        If Respuesta = 6 Then
            DataSeries.Recordset.MoveFirst
            DataCuadros.Recordset.MoveFirst
            FormCuadros.Visible = False
            FormCuadros.Enabled = False
            Form1.Visible = True
            Form1.Enabled = True
        End If
    Else
        BotónPrimero.Visible = True
        BotónAnterior.Visible = True
        BotónSiguiente.Visible = True
    End If
End Sub

```

```

        BotónUltimo.Visible = True
        BotónPrimero.Enabled = True
        BotónAnterior.Enabled = True
        BotónSiguiente.Enabled = True
        BotónUltimo.Enabled = True
        BotónImprimir.Visible = False
        BotónVerSeries.Caption = "&Ver Series"
        BotónSalir.Caption = "Sali&r"
        MSFlexGrid1.Visible = False
        VerSerie = 0
        GuardaSerie = 0
        For i = 1 To NSeries
            List1.List(i - 1) = NomSeries(i)
        Next i
    End If
End Sub

Private Sub Form_Load()
    Call CentrarForm(Me)
    VerSerie = 0
    GuardaSerie = 0
    DataSeries.DatabaseName = CurDir + "\BasesCua.mdb"
    DataCuadros.DatabaseName = CurDir + "\BasesCua.mdb"
    DataPeriodo.DatabaseName = CurDir + "\BasesCua.mdb"
    DataUnidad.DatabaseName = CurDir + "\BasesCua.mdb"
    DataSerieValor.DatabaseName = CurDir + "\BasesCua.mdb"
    DataCapitulo.DatabaseName = CurDir + "\BasesCua.mdb"
End Sub

` Declaraciones del formulario Ver Series

Private Sub Salir(c As Integer)
    Dim Respuesta As Integer
    Respuesta = MsgBox("Seguro de Salir ?", 36, "Salir de Ver Series por Código")
    If Respuesta = 6 Then
        DataSeries.Recordset.MoveFirst
        FormSeries.Visible = False
        FormSeries.Enabled = False
        FormaPrincipal.Visible = True
        FormaPrincipal.Enabled = True
    Else
        c = 1
    End If
End Sub

End Sub

Private Sub Encabezado()
    Printer.Print
    Printer.Print
    Printer.Print
    Printer.Font.Size = TamFuente + 1
    With DataSeries.Recordset
        Printer.FontBold = True
        Printer.Print Margen & "Serie :";
        Printer.FontBold = False
        Printer.Print .Fields("Tit_serie").Value & _
            " (" & .Fields("A_Inicio").Value & ")"
    End With

    Printer.FontBold = True
    Printer.Print Margen & "PERIODO :";
    Printer.FontBold = False
    Printer.Print DataPeriodo.Recordset.Fields("descripcion").Value

    Printer.FontBold = True
    Printer.Print Margen & "UNIDAD :";
    Printer.FontBold = False

```

```

Printer.Print DataUnidad.Recordset.Fields("descripcio").Value
Printer.Print

With DataCuadros.Recordset
    Printer.FontBold = True
    Printer.Print Margen & "Cuadro :";
    Printer.FontBold = False
    Printer.Print .Fields("Titulo").Value & " (" & .Fields("A_Inicio").Value _
        & " , " & .Fields("A_fin").Value & ")"
End With

Printer.Print
Printer.Font.Size = TamFuente
End Sub

Private Sub BotónArima_Click()
    FormSeries.Enabled = False
    FormArima.Visible = True
    FormArima.Enabled = True
End Sub

Private Sub BotónCriterio_Click()
Dim criterio
    criterio = InputBox("Introduzca el criterio de búsqueda (Enter para
ninguno)", _
        "Criterio de búsqueda", "")
    criterio = "SELECT * from SERIE where TIT_SERIE like '*' & criterio & '*'"
    DataSeries.RecordSource = criterio
    DataSeries.Refresh
    On Error GoTo CError
    DataSeries.Recordset.MoveFirst
    Exit Sub
CError:
    MsgBox "No se han encontrado coincidencias con su criterio de búsqueda", _
        vbOKOnly, "Error de búsqueda"
    DataSeries.RecordSource = "SERIE"
    DataSeries.Refresh
End Sub

Private Sub BotónPrimera_Click()
    DataSeries.Recordset.MoveFirst
End Sub

Private Sub BotónAnterior_Click()
    DataSeries.Recordset.MovePrevious
    If DataSeries.Recordset.BOF Then
        DataSeries.Recordset.MoveFirst
    End If
End Sub

Private Sub BotónProxima_Click()
    DataSeries.Recordset.MoveNext
    If DataSeries.Recordset.EOF Then
        DataSeries.Recordset.MoveLast
    End If
End Sub

Private Sub BotónUltima_Click()
    DataSeries.Recordset.MoveLast
End Sub

Private Sub BotónBuscar_Click()
Dim R As Integer
Dim cc As String, bb As String, marca As String, criterio As String
    cc = Chr(34)
    bb = InputBox("Código de Serie a Buscar:", "Buscar Serie")
    If bb = "" Then Exit Sub

```

```

marca = DataSet.Recordset.Bookmark
criterio = "COD_SERIE=" & Val(bb)
DataSet.Recordset.FindFirst criterio
If DataSet.Recordset.NoMatch Then
    R = MsgBox("No se encuentra el registro", 0, "Error")
    DataSet.Recordset.Bookmark = marca
End If
End Sub

Private Sub BotónAlmacenar_Click()
Dim NomArchivo As String, F As String
Dim p As Integer, I As Integer
    NomArchivo = ""
    NomArchivo = InputBox("Introduzca el nombre del archivo", "Guardar Serie",
"Serie" & Label8.Caption & ".txt")
    If NomArchivo = "" Then Exit Sub
    p = InStr(1, NomArchivo, ".")
    If p <> 0 Then
        NomArchivo = Mid(NomArchivo, 1, p - 1)
    End If
    NomArchivo = NomArchivo & ".txt"
    BotónPrimera.Enabled = False
    BotónProxima.Enabled = False
    BotónAnterior.Enabled = False
    BotónUltima.Enabled = False
    BotónAlmacenar.Enabled = False
    BotónGráfico.Enabled = False
    BotónSalir.Enabled = False
    BotónBuscar.Enabled = False
    BotónImprimir.Enabled = False
    Open NomArchivo For Output As #1
    With DataSet.Recordset
        Print #1, "Titulo de la Serie : " & .Fields("Tit_serie").Value & _
            " (" & .Fields("A_Inicio").Value & ")"
    End With
    Print #1, "PERIODO : " & DataPeriodo.Recordset.Fields("descripcio").Value
    Print #1, "UNIDAD : " & DataUnidad.Recordset.Fields("descripcio").Value
    Print #1,
    Print #1,
    With DataCuadros.Recordset
        Print #1, "Titulo del Cuadro : " & .Fields("Titulo").Value & " (" &
.Fields("A_Inicio").Value _
            & " , " & .Fields("A_fin").Value & ")"
    End With
    Print #1,
    Print #1, "PERIODO VALOR"
    Print #1, "-----"
    For I = 1 To Nv
        Print #1, ListaSerie.List(I - 1)
    Next I
    Print #1, "-----"
    Close #1
    F = MsgBox("Serie Almacenada con el nombre : " + NomArchivo, vbOKOnly,
"Almacenar")
    BotónPrimera.Enabled = True
    BotónProxima.Enabled = True
    BotónAnterior.Enabled = True
    BotónUltima.Enabled = True
    BotónBuscar.Enabled = True
    BotónAlmacenar.Enabled = True
    BotónGráfico.Enabled = True
    BotónSalir.Enabled = True
    BotónImprimir.Enabled = True
End Sub

Private Sub BotónGráfico_Click()
    Dim CX As Single, CY As Single, DX As Single, DY As Single

```

```
Dim Mitad As Single, Pot As Single, Temp As Single
Dim Sx, Sy, Sxy, Sxx, Pendiente, Corte, X1, X2, Y1, Y2 As Single
Dim I As Integer, p As Integer, N As Integer, Cuantos As Integer
Dim Msg1 As String, Msg2 As String
```

```
Picture1.Width = 8300
Picture1.Height = 6400
Picture1.Visible = Not Picture1.Visible
Picture1.Cls
```

```
CX = 800
CY = Picture1.ScaleHeight - 400
```

```
Picture1.BackColor = QBColor(7)
Picture1.ForeColor = QBColor(0)
Picture1.DrawWidth = 3
Picture1.Line (CX, 250)-(CX, Picture1.ScaleHeight - 200)
Picture1.Line (200, CY)-(Picture1.ScaleWidth - 200, CY)
```

```
DX = (Picture1.ScaleWidth - 1200) / Nv1
DY = (Picture1.ScaleHeight - 1000) / Maximo
```

```
Picture1.ForeColor = RGB(0, 0, 255)
```

```
Msg1 = Label10.Caption
Msg2 = ""
If Len(Msg1) > 40 Then
    I = 40
    p = 0
    While (p = 0) And (I <= Len(Msg1))
        If Mid(Msg1, I, 1) = " " Then
            p = I
        Else
            I = I + 1
        End If
    Wend
    Msg2 = Right(Msg1, Len(Msg1) - p)
    Msg1 = Left(Msg1, p)
End If
```

```
Picture1.Font.Size = 12
Mitad = Picture1.TextWidth(Msg1) / 2
Picture1.CurrentY = 100
Picture1.CurrentX = Picture1.ScaleWidth / 2 - Mitad
Picture1.Print Msg1
Mitad = Picture1.TextWidth(Msg2) / 2
Picture1.CurrentY = 320
Picture1.CurrentX = Picture1.ScaleWidth / 2 - Mitad
Picture1.Print Msg2
```

```
Picture1.DrawWidth = 2
Picture1.Font.Size = 7
Cuantos = 5 / (Periodo(2) - Periodo(1))
For I = 1 To Nv1 Step Cuantos
    Picture1.CurrentX = CX + DX * I - 50
    Picture1.CurrentY = CY + 50
    Picture1.Print Periodo(I)
    Picture1.Line (CX + DX * I, CY + 25)-(CX + DX * I, CY - 25)
Next I
```

```
Pot = 0.1
Temp = Maximo
While Temp > 1
    Temp = Temp / 10
    Pot = Pot * 10
Wend
```

```
N = Int(Maximo / Pot)
```

```

While N < 8
    Pot = Pot / 2
    N = Int(Maximo / Pot)
Wend

Temp = Pot
While Temp < Maximo
    Picture1.CurrentX = 0
    Picture1.CurrentY = CY - Temp * DY
    Picture1.Print Temp
    Picture1.Line (CX - 25, CY - Temp * DY + 50)-(CX + 25, CY - Temp * DY + 50)
    Temp = Temp + Pot
Wend

Sx = 0
Sy = 0
Sxy = 0
Sxx = 0
For I = 1 To Nv1
    If Valores(I) <> -1000 Then
        Picture1.PSet (CX + DX * I, CY - Valores(I) * DY), QBColor(12)
        Sx = Sx + I
        Sy = Sy + Valores(I)
        Sxy = Sxy + I * Valores(I)
        Sxx = Sxx + I ^ 2
    End If
Next I
Pendiente = (Nv * Sxy - Sx * Sy) / (Nv * Sxx - (Sx ^ 2))
Corte = (Sy - Pendiente * Sx) / Nv
X1 = 1
X2 = Nv1
Y1 = Pendiente * X1 + Corte
Y2 = Pendiente * X2 + Corte
Picture1.Line (CX + DX * X1, CY - Y1 * DY)-(CX + DX * X2, CY - Y2 * DY),
QBColor(1)
End Sub

Private Sub BotónImprimir_Click()
Dim cad, cad1 As String
Dim I As Integer, NPuntos As Integer, Pagina As Integer
Dim NroLineas As Integer, NumPaginas As Integer, Quedan As Integer
Dim NroColumnas As Integer, NroFilas As Integer, Col As Integer
Dim Fila As Integer, pos As Integer, Nv As Integer

BotónPrimera.Enabled = False
BotónProxima.Enabled = False
BotónAnterior.Enabled = False
BotónUltima.Enabled = False
BotónAlmacenar.Enabled = False
BotónGráfico.Enabled = False
BotónSalir.Enabled = False
BotónBuscar.Enabled = False
BotónImprimir.Enabled = False

Printer.Font.Name = "Courier New"
Printer.EndDoc
I = 0
Do
    I = I + 1
    Printer.Font.Size = I
Loop Until Printer.TextHeight("AZ") >= 220
TamFuente = I
Printer.Font.Size = TamFuente
NPuntos = Printer.TextHeight("AZ")
NroLineas% = (Printer.Width - 500) / NPuntos
NroLineas% = NroLineas% - 4
Nv = ListaSerie.ListCount

```

```

NumPaginas = Int(Nv1 / (NroLineas% * 3)) + 1

For Pagina = 1 To NumPaginas
    Call Encabezado
    Quedan = Nv - (Pagina - 1) * (NroLineas% * 3)
    If Quedan > (NroLineas% * 3) Then
        NroColumnas = 3
    Else
        NroColumnas = Int(Quedan / NroLineas%) + 1
    End If
    If Quedan > NroLineas% Then
        NroFilas = NroLineas%
    Else
        NroFilas = Quedan
    End If
    cad = ""
    For Col = 1 To NroColumnas
        cad = cad + "PERIODO      VALOR      "
        cad1 = cad1 + "-----"
    Next Col
    Printer.Print Margen & cad
    Printer.Print Margen & cad1
    For Fila = 1 To NroFilas
        cad = " "
        For Col = 1 To NroColumnas
            pos = (Pagina - 1) * (3 * NroLineas%) + (Col - 1) * NroLineas% + Fila
            If pos <= Nv Then cad = cad + ListaSerie.List(pos) + "      "
        Next Col
        Printer.Print Margen & cad
    Next Fila
    Printer.Print Margen & cad1
    Printer.CurrentY = 13800
    Printer.Print Space(68); "Pag: "; Pagina
    Printer.NewPage
Next Pagina
Printer.EndDoc

BotónPrimera.Enabled = True
BotónProxima.Enabled = True
BotónAnterior.Enabled = True
BotónUltima.Enabled = True
BotónAlmacenar.Enabled = True
BotónGráfico.Enabled = True
BotónSalir.Enabled = True
BotónBuscar.Enabled = True
BotónImprimir.Enabled = True
End Sub

Private Sub BotónSalir_Click()
Dim c As Integer
    Call Salir(c)
End Sub

Private Sub DataSeries_Reposition()
Dim cc As String, criterio As String, F As String, Elemento As String
Dim s As Single, s1 As Single, Va As Single
Dim Cva As Integer, Aini As Integer, Incre As Integer
Dim A As Single, N As Integer, I As Integer
Dim NumValPer As Integer, Increl As Single
Picture1.Visible = False
BotónAlmacenar.Enabled = False
BotónGráfico.Enabled = False
BotónImprimir.Enabled = False
cc = Chr(34)

```

```

If Label9.Caption <> "" Then
    DataCuadros.Recordset.MoveFirst
    criterio = "COD_CUADRO=" & Label9.Caption
    DataCuadros.Recordset.FindFirst criterio
    DataPeríodo.Recordset.MoveFirst
    criterio = "COD_PERIOD=" & DataSeries.Recordset.Fields("COD_PERIOD").Value
    DataPeríodo.Recordset.FindFirst criterio
    DataUnidad.Recordset.MoveFirst
    criterio = "COD_UNIDAD=" & DataSeries.Recordset.Fields("COD_UNIDAD").Value
    DataUnidad.Recordset.FindFirst criterio
    ListaSerie.Clear
    criterio = "COD_SERIE=" & Label8.Caption
    DataSerieValor.Recordset.FindFirst criterio
    s = 0
    s1 = 0
    Minimo = DataSerieValor.Recordset.Fields("VALOR").Value
    Maximo = DataSerieValor.Recordset.Fields("VALOR").Value
    Nv = 0
    Nv1 = 0
    Aini = DataSeries.Recordset.Fields("A_INICIO").Value
    Incre = DataPeríodo.Recordset.Fields("INCREMENTO").Value
    NumValPer = DataPeríodo.Recordset.Fields("NUM_VALORE").Value
    Incre1 = Incre / NumValPer
    ListaSerie.AddItem "Período      Valor"
    While Not DataSerieValor.Recordset.NoMatch
        Cva = DataSerieValor.Recordset.Fields("COD_VALOR").Value
        Va = DataSerieValor.Recordset.Fields("VALOR").Value
        A = Aini + Incre1 * (Cva - 1)
        Nv1 = Nv1 + 1
        Periodo(Nv1) = A
        Valores(Nv1) = Va
        If Va = -1000 Then
            F = "          N.D."
        Else
            Nv = Nv + 1
            s = s + Va
            s1 = s1 + Va ^ 2
            If Minimo > Va Then Minimo = Va
            If Maximo < Va Then Maximo = Va
            F = Format(Va, "###,##0.00")
            N = Len(F)
            For I = 1 To (14 - N)
                F = " " & F
            Next I
        End If
        Elemento = ""
        If NumValPer <> 1 Then
            Elemento = "-" & (Int(NumValPer * (A - Int(A) + Incre1) + 0.5))
        End If
        Elemento = Int(A) & Elemento
        Elemento = Elemento & Space(7 - Len(Elemento)) & F

        ListaSerie.AddItem Elemento
        DataSerieValor.Recordset.FindNext criterio
    Wend
    Media = s / Nv
    Varianza = s1 / Nv - Media ^ 2
    Desv = Sqr(Varianza)
    Label17.Caption = Format(Media, "###,##0.00")
    Label19.Caption = Format(Varianza, "###,##0.00")
    Label24.Caption = Format(Desv, "###,##0.00")
    Label21.Caption = Format(Minimo, "###,##0.00")
    Label23.Caption = Format(Maximo, "###,##0.00")
End If
BotónAlmacenar.Enabled = True
BotónGráfico.Enabled = True
BotónImprimir.Enabled = True

```

```

End Sub

Private Sub Form_Load()
    Call CentrarForm(Me)
    Picture1.Visible = False
    DataSeries.DatabaseName = CurDir + "\BasesCua.mdb"
    DataSeries.RecordSource = "SELECT * FROM SERIE WHERE TIT_SERIE LIKE '*'"
    DataCuadros.DatabaseName = CurDir + "\BasesCua.mdb"
    DataPeriodo.DatabaseName = CurDir + "\BasesCua.mdb"
    DataUnidad.DatabaseName = CurDir + "\BasesCua.mdb"
    DataSerieValor.DatabaseName = CurDir + "\BasesCua.mdb"
    DataCuadros.Refresh
    DataPeriodo.Refresh
    DataUnidad.Refresh
    DataSerieValor.Refresh
    DataSeries.Refresh
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Call Salir(Cancel)
End Sub

` Declaraciones del Formulario ARIMA

Option Explicit
Dim p As Integer, d As Integer, q As Integer
Dim ErrorS As Double
Dim Cmd, Linea

Private Sub Salir(c As Integer)
    Dim Respuesta As Integer
    c = 0
    Respuesta = MsgBox("Seguro de Salir ?", 36, "Salir de Análisis ARIMA")
    If Respuesta = 6 Then
        FormArima.Visible = False
        FormArima.Enabled = False
        FormSeries.Enabled = True
    Else
        c = 1
    End If
End Sub

Private Sub BAlmacenar_Click()
Dim NomArchivo As String, F As String
Dim p As Integer, I As Integer
    NomArchivo = ""
    NomArchivo = InputBox("Introduzca el nombre del archivo", "Guardar Análisis",
"Serie" & FormSeries.Label8.Caption & ".txt")
    If NomArchivo = "" Then Exit Sub
    p = InStr(1, NomArchivo, ".")
    If p <> 0 Then
        NomArchivo = Mid(NomArchivo, 1, p - 1)
    End If
    NomArchivo = NomArchivo & ".txt"

    Command1.Enabled = False
    Command2.Enabled = False
    BImprimir.Enabled = False
    BAlmacenar.Enabled = False

    Open NomArchivo For Output As #1
    Print #1, RichTextBox1.Text
    Close #1

    Command1.Enabled = True
    Command2.Enabled = True
    BImprimir.Enabled = True

```

```

BALmacenar.Enabled = True

End Sub

Private Sub Command1_Click()
    Dim QQ, LF, CR          ' Declara variables.
    Dim I As Integer, J As Integer
    Dim Media As Double, Li As Double, Ls As Double

    QQ = Chr(34)           ' Las comillas.
    LF = Chr(10)           ' Line Feed
    CR = Chr(13)           ' Carry Return

    p = Val(Text1.Text)    ' Parámetro autoregresivo
    d = Val(Text2.Text)    ' Parámetro diferenciación
    q = Val(Text3.Text)    ' Parámetro promedio móvil

    On Error Resume Next

    RichTextBox1.Text = "Procesando....."
    RichTextBox1.Text = RichTextBox1.Text & LF & "Cargando el S-PLUS....."

    ErrorS = Shell("d:\Archivos de programa\sp2000\cmd\SPLUS.EXE", 4)    ' Inicia
S-PLUS
    If ErrorS = 0 Then
        ErrorS = Shell("c:\Archivos de programa\sp2000\cmd\SPLUS.EXE", 4)    '
Inicia S-PLUS
    End If

    If ErrorS = 0 Then
        MsgBox "No conseguí el S-PLUS en su Sistema. Verifique.....", vbOKOnly,
"Error de Sistema"
        Exit Sub
    End If

    Command2.Caption = "Salir"
    Command1.Visible = False
    Command2.Visible = False
    BImprimir.Visible = False
    BALmacenar.Visible = False

    CajaCom.LinkTimeout = 300
    CajaCom.LinkTopic = "S-PLUS|SCommand"    ' Establece el tema del vínculo.
    Cmd = "Nuevo<-data.frame()"
    CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.
    CajaCom.LinkExecute Cmd                ' Crea una matriz de datos

    RichTextBox1.Text = RichTextBox1.Text & LF & "..... enviando datos"

    Cmd = ""
    For I = 1 To Nv1
        If Valores(I) = -1000 Then
            Cmd = Cmd & ""
        Else
            Cmd = Cmd & Valores(I)
        End If
        If I <> Nv1 Then
            Cmd = Cmd + LF + CR
        End If
    Next I
    Linea = ""
    For I = 1 To Len(Cmd)
        If Mid(Cmd, I, 1) = "," Then
            Linea = Linea & "."
        Else
            Linea = Linea & Mid(Cmd, I, 1)
        End If
    Next I

```

```

Next I
CajaCom.Text = Linea
CajaCom.LinkTimeout = 1
CajaCom.LinkTopic = "S-PLUS|Nuevo$V1"
CajaCom.LinkItem = "r1c1"
CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.
CajaCom.LinkPoke          ' Se envia los datos al data frame
Nuevo

CajaCom.LinkTimeout = 300
CajaCom.LinkTopic = "S-PLUS|SCommand"    ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.

Cmd = "modelo <- list(list(order=c(" & p & "," & d & "," & q & "),period=1))"
Cmd = Cmd & LF & "Ari<-arima.mle(Nuevo$V1,model=modelo)"

CajaCom.LinkExecute Cmd                  ' Ejecuta el comando ARIMA

Cmd = "res<-data.frame(Ari$method)"
CajaCom.LinkTimeout = 200
CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.
CajaCom.LinkExecute Cmd                  '

CajaCom.LinkTimeout = 200
CajaCom.LinkTopic = "S-PLUS|res$Ari.method"      ' Establece el tema del
vínculo.
CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.
CajaCom.LinkItem = "r1c1"
CajaCom.LinkRequest

RichTextBox1.Text = "Resultados del modelo ARIMA" & LF & LF
                                & "Método : " & CajaCom.Text & LF & "Modelo : " &
Text1.Text _
                                & " " & Text2.Text & " " & Text3.Text & LF & "Periodo: 1"
-
                                & LF & LF & "Coeficientes" & LF

Cmd = "res<-data.frame(Ari$model$ar)"
CajaCom.LinkTimeout = 200
CajaCom.LinkTopic = "S-PLUS|SCommand"    ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.
CajaCom.LinkExecute Cmd                  ' Ejecuta el comando

For I = 1 To p
    CajaCom.LinkTimeout = 200
    CajaCom.LinkTopic = "S-PLUS|res$Ari.model.ar"      ' Establece el tema del
vínculo.
    CajaCom.LinkMode = vbLinkManual          ' Establece el modo del
vínculo.
    CajaCom.LinkItem = "r" & I & "c1"
    CajaCom.LinkRequest
    RichTextBox1.Text = RichTextBox1.Text & " AR" & I & " :" & CajaCom.Text
Next I

Cmd = "res<-data.frame(Ari$model$ndiff)"
CajaCom.LinkTimeout = 200
CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.
CajaCom.LinkTopic = "S-PLUS|SCommand"    ' Establece el tema del vínculo.
CajaCom.LinkExecute Cmd                  ' Ejecuta el comando

CajaCom.LinkTopic = "S-PLUS|res$Ari.model.ndiff"
CajaCom.LinkMode = vbLinkManual          ' Establece el modo del vínculo.
CajaCom.LinkItem = "r1c1"
CajaCom.LinkRequest

RichTextBox1.Text = RichTextBox1.Text & " Ndiff :" & CajaCom.Text

```

```

Cmd = "res<-data.frame(Ari$model$ma)"
CajaCom.LinkTopic = "S-PLUS|SCommand"      ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual             ' Establece el modo del vínculo.
CajaCom.LinkExecute Cmd                     ' Ejecuta el comando

For I = 1 To q
  CajaCom.LinkTopic = "S-PLUS|res$Ari.model.ma"  ' Establece el tema del
vínculo.
  CajaCom.LinkItem = "r" & I & "c1"
  CajaCom.LinkMode = vbLinkManual             ' Establece el modo del
vínculo.
  CajaCom.LinkRequest
  RichTextBox1.Text = RichTextBox1.Text & "  MA" & I & " :" & CajaCom.Text
Next I

Cmd = "res<-data.frame(Ari$aic)"
CajaCom.LinkTopic = "S-PLUS|SCommand"      ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual             ' Establece el modo del vínculo.
CajaCom.LinkExecute Cmd                     ' Ejecuta el comando

CajaCom.LinkTopic = "S-PLUS|res$Ari.aic"
CajaCom.LinkMode = vbLinkManual             ' Establece el modo del vínculo.
CajaCom.LinkItem = "r1c1"
CajaCom.LinkRequest

RichTextBox1.Text = RichTextBox1.Text & LF & "AIC : " & CajaCom.Text & LF

Cmd = "res<-data.frame(Ari$var.coef)"
CajaCom.LinkTopic = "S-PLUS|SCommand"      ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual             ' Establece el modo del vínculo.
CajaCom.LinkExecute Cmd                     ' Ejecuta el comando

RichTextBox1.Text = RichTextBox1.Text & "Matriz de varianza y covarianza para
los" _
          & " coeficientes p y q" & LF
RichTextBox1.Text = RichTextBox1.Text & "  "
For I = 1 To p
  RichTextBox1.Text = RichTextBox1.Text & "  ar(" & I & ")  "
Next I
For I = 1 To q
  RichTextBox1.Text = RichTextBox1.Text & "  ma(" & I & ")  "
Next I
RichTextBox1.Text = RichTextBox1.Text & LF

For I = 1 To p + q
  For J = 1 To p + q
    CajaCom.LinkTopic = "S-PLUS|res$var.coef"      ' Establece el tema del
vínculo.
    CajaCom.LinkItem = "r"&I & "c"&J
    CajaCom.LinkMode = vbLinkManual             ' Establece el modo del
vínculo.
    CajaCom.LinkRequest
    RichTextBox1.Text = RichTextBox1.Text & "  " & "  " &
Format(Val(CajaCom.Text), "##0.00000")
  Next J
  RichTextBox1.Text = RichTextBox1.Text & LF
Next I
RichTextBox1.Text = RichTextBox1.Text & LF

Cmd = "res<-data.frame(Ari$converged)"
CajaCom.LinkTopic = "S-PLUS|SCommand"      ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual             ' Establece el modo del vínculo.
CajaCom.LinkExecute Cmd                     ' Ejecuta el comando

CajaCom.LinkTopic = "S-PLUS|res$converged"    ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual             ' Establece el modo del vínculo.
CajaCom.LinkItem = "r1c1"

```

```

CajaCom.LinkRequest

If CajaCom.Text Then
    RichTextBox1.Text = RichTextBox1.Text & "Convergencia de tipo: "
    Cmd = "res<-data.frame(Ari$conv.type)"
    CajaCom.LinkTopic = "S-PLUS|SCommand"      ' Establece el tema del vínculo.
    CajaCom.LinkMode = vbLinkManual            ' Establece el modo del
vínculo.
    CajaCom.LinkExecute Cmd                    ' Ejecuta el comando

    CajaCom.LinkTopic = "S-PLUS|res$conv.type"  ' Establece el tema del
vínculo.
    CajaCom.LinkMode = vbLinkManual            ' Establece el modo del
vínculo.
    CajaCom.LinkItem = "r1c1"
    CajaCom.LinkRequest
    RichTextBox1.Text = RichTextBox1.Text & CajaCom.Text
Else
    RichTextBox1.Text = RichTextBox1.Text & "No hubo Convergencia"
End If

RichTextBox1.Text = RichTextBox1.Text & LF & "Predicciones del 95%" & LF

CajaCom.LinkTimeout = 300
CajaCom.LinkTopic = "S-PLUS|SCommand"      ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual            ' Establece el modo del vínculo.

Cmd = Cmd & LF & "Pred<-arima.forecast(Nuevo$V1,model=modelo,5)"

CajaCom.LinkExecute Cmd                      ' Ejecuta el comando ARIMA
(predicciones)

Cmd = "Pre<-data.frame(Pred)"
CajaCom.LinkTopic = "S-PLUS|SCommand"      ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual            ' Establece el modo del vínculo.
CajaCom.LinkExecute Cmd                      ' Ejecuta el comando

CajaCom.LinkTopic = "S-PLUS|Pre"          ' Establece el tema del vínculo.
CajaCom.LinkMode = vbLinkManual            ' Establece el modo del vínculo.
CajaCom.LinkItem = "r1c1"
CajaCom.LinkRequest

Media = Val(CajaCom.Text)
For I = 1 To 5
    CajaCom.LinkTopic = "S-PLUS|Pre"      ' Establece el tema del vínculo.
    CajaCom.LinkMode = vbLinkManual        ' Establece el modo del
vínculo.
    CajaCom.LinkItem = "r" & I & "c2"
    CajaCom.LinkRequest
    Li = Media - 1.96 * Val(CajaCom.Text)
    Ls = Media + 1.96 * Val(CajaCom.Text)
    RichTextBox1.Text = RichTextBox1.Text & "(" & Format(Li, "###000.00000") &
" ; " & Format(Ls, "###000.00000") & ")" & LF
Next I

If ErrorS <> 0 Then
    CajaCom.LinkTimeout = 200
    CajaCom.LinkTopic = "S-PLUS|SCommand"  ' Establece el tema del vínculo.
    Cmd = "q()"
    CajaCom.LinkMode = vbLinkAutomatic     ' Establece el modo del vínculo.
    CajaCom.LinkExecute Cmd                ' Crea un vector de datos de datos
End If

Command1.Visible = True
Command2.Visible = True
BImprimir.Visible = True
BAlmacenar.Visible = True

```

```

End Sub

Private Sub Command2_Click()
Dim c As Integer
  RichTextBox1.Text = ""
  BImprimir.Visible = False
  BAlmacenar.Visible = False
  Call Salir(c)
End Sub

Private Sub BImprimir_Click()
  Dim I As Integer, NPuntos As Integer, NroLineas As Integer

  Command2.Enabled = False
  Command1.Enabled = False
  BImprimir.Enabled = False
  BAlmacenar.Enabled = False

  Printer.Font.Name = "Courier New"
  Printer.EndDoc
  I = 0
  Do
    I = I + 1
    Printer.Font.Size = I
  Loop Until Printer.TextHeight("AZ") >= 240
  TamFuente = I
  Printer.Font.Size = TamFuente
  NPuntos = Printer.TextHeight("AZ")
  NroLineas = (Printer.Width) / NPuntos + 1

  Printer.ScaleLeft = -1000
  Printer.ScaleTop = 1000

  Printer.Print RichTextBox1.Text

  Printer.EndDoc
  Command2.Enabled = True
  Command1.Enabled = True
  BImprimir.Enabled = True
  BAlmacenar.Enabled = True

End Sub

Private Sub Form_Load()
  Call CentrarForm(Me)
  BImprimir.Visible = False
  BAlmacenar.Visible = False
  Text1.Text = 1
  Text2.Text = 1
  Text3.Text = 1
End Sub

Private Sub Form_Unload(Cancel As Integer)
  Call Salir(Cancel)
End Sub

```

Anexo 6. Listado del código de páginas Web

` **Página Principal (Index.htm)**

```
<html>

<head>
  <title>Bases</title>
  <meta name="GENERATOR" content="Microsoft FrontPage 3.0">
</head>

<frameset framespacing="0" border="false" frameborder="0" rows="40,*">
  <frame name="titular" scrolling="no" noresize target="contenido"
src="Titulo.htm"
  marginwidth="0" marginheight="0" style="background-color: rgb(192,192,192)">
  <frameset cols="116,*">
    <frame name="contenido" target="principal" src="Menu.htm" scrolling="auto">
    <frame name="principal" src="Vacía.htm" scrolling="auto" style="color:
rgb(192,192,192)">
  </frameset>
</frameset>
<noframes>
<body>
  <p>Esta página usa marcos, pero su explorador no los admite.</p>
</body>
</noframes>
</frameset>
</html>
```

` **Archivo titulo.htm**

```
<html>

<head>
<title>Titulo Bases</title>
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<base target="contenido">
<meta name="Microsoft Border" content="none">
</head>

<body>

<h1 align="center"><font color="#0080C0"><marquee bgcolor="#EA0F3A" border="1"
style="color: rgb(0,81,40)">Bases Cuantitativas de la Economía
Venezolana</marquee></font></h1>
</body>
</html>
```

` **Archivo menu.htm**

```
<html>

<head>
<title>Menu Bases</title>
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<base target="principal">

<meta name="Microsoft Border" content="none"></head>

<body>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>
<applet code="fphover.class" codebase="_fpclass/" width="85" height="50">
```

```

<param name="text" value="Ver Cuadros">
<param name="color" value="#008080">
<param name="hovercolor" value="#0000FF">
<param name="textcolor" value="FFFFFF">
<param name="effect" value="glow">
<param name="url" value="Asp/Cuadros.asp" valuetype="ref">
<param name="bgcolor" value="FF0000">
<param name="font" value="Dialog">
<param name="fontstyle" value="bold">
<param name="fontsize" value="15">
<param name="target" value="principal">
</applet>
</p>

<p align="center">
<applet code="fphover.class" codebase="fpclass/" width="85" height="50">
  <param name="text" value="Ver series">
  <param name="color" value="#008080">
  <param name="hovercolor" value="#0000FF">
  <param name="textcolor" value="FFFFFF">
  <param name="effect" value="glow">
  <param name="url" value="Asp/Series.asp" valuetype="ref">
  <param name="font" value="Dialog">
  <param name="fontstyle" value="bold">
  <param name="fontsize" value="15">
  <param name="target" value="principal">
  <param name="image" value valuetype="ref">
  <param name="hoverimage" value valuetype="ref">
  <param name="sound" value valuetype="ref">
  <param name="hoversound" value valuetype="ref">
</applet>
<br>
</p>
</body>
</html>

```

` Archivo vacia.htm

```

<html>

<head>
<title>Datos</title>
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<meta name="Microsoft Border" content="none">
</head>

<body>

<p align="center"><font size="5"><br>
Esta página contiene la versión digital del libro:</font><br>
<font color="#FF0000" size="6">&quot;Bases Cuantitativas de la Economía
Venezolana.
1830-1989&quot;</font><br>
<font size="6">de <em>Asdrúbal Baptista</em></font></p>

<p align="justify"><font size="5">Los datos del libro pueden ser consultados por:
</font></p>

<ul>
<li><p align="justify"><font size="5">Cuadros, tal y como aparecen en el libro
</font></p>
</li>
<li><p align="justify"><font size="5">Series individuales </font></p>
</li>
</ul>

```

```

<p align="justify"><font size="5">Para lo cual pulse el respectivo boton en la
parte
izquierda.<br>
</font><br>
<br>
<font size="3">Cualquier información y/o sugerencia hacerla llegar a: <a
HREF="mailto:angelz@faces.ula.ve">angelz@faces.ula.ve</a></font> </p>
</body>
</html>

```

` Archivo adofunc.inc

```

<%
function Abrir_conn(oConn)
on error resume next
oConn.Open "Bases"
If err.number<>0 then
    Response.Write "Usted es un Usuario del Dominio,                pero no puede
conectarse a esta Base de                Datos.<BR>"
    Response.Write "<b>Error:</b> " & Hex(Err.Number)                & "<BR>"
    Response.Write "<b>Descripcion:</b><br>" &                Err.Description & "<br>"
    Abrir_conn=False
else
    Abrir_conn=True
end if
end function

function ejecutar (oConn,oRS,oSQL)
on error resume next
oRS.Open oSQL, oConn, adOpenKeyset,adLockOptimistic

If err.number<>0 then
    Response.Write "Usted es un Usuario Al del dominio,                pero no puede
Conectarse a esta Base de Datos.<BR>"
    Response.Write "<b>Error:</b> " & Hex(Err.Number)                & "<BR>"
    Response.Write "<b>Descripcion:</b><br>" &                Err.Description & "<br>"
    ejecutar=False
else
    ejecutar=True
end if
end function

Function OCadena(texto)
if texto="" then
    OCadena="NULL"
else
    OCadena="'"&texto&""
end if
End Function

Function ONum(texto)
if texto="" then
    ONum="NULL"
else
    ONum=texto
end if
End Function

Function ODBCDateTime(TheDateTime)
If IsNull(TheDateTime) OR NOT (TheDateTime <> "") Then
    ODBCDateTime = "NULL"
Else
    TheDateTime=CDate(TheDateTime)
    TheMonth = CStr(Month(TheDateTime))
    If Month(TheDateTime) < 10 Then
        TheMonth = "0" + TheMonth
    end if

```

```

TheDay = CStr(Day(TheDateTime))
If Day(TheDateTime) < 10 Then
    TheDay = "0" + TheDay
end if
TheHour = CStr(Hour(TheDateTime))
If Hour(TheDateTime) < 10 Then
    TheHour = "0" + TheHour
end if
TheMinute = CStr(Minute(TheDateTime))
If Minute(TheDateTime) < 10 Then
    TheMinute = "0" + TheMinute
end if
TheSecond = CStr(Second(TheDateTime))
If Second(TheDateTime) < 10 Then
    TheSecond = "0" + TheSecond
End If
ODBCDateTime = "" + CStr(Year(TheDateTime)) + "/" + TheMonth + "/" + TheDay +
" " + TheHour + ":" + TheMinute + ":" + TheSecond + ""
end if
End Function
%>

```

\ Archivo cuadros.asp

```

<% Response.Expires = 0 %>

<!-- #include file= "adovbs.inc" -->
<!-- #include file= "adofunc.inc" -->
<html>

<head>
<title>Consulta de Cuadros</title>
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<script language="Javascript">

<!--

var insertando = false

var x_opc = "n"

function navega(fc)
{
    var ir_a = parseInt(fc.pagina_actual.value)
    var ultima_pag = parseInt(fc.ultimo.value)
    if (x_opc == "pr")
    {
        if (ir_a != 1) { ir_a = 1 }
    }
    else
    {
        if (x_opc == "an")
        {
            if (ir_a != 1) { ir_a = ir_a - 1 }
        }
        else
        {
            if (x_opc == "si")
            {
                if (ir_a != ultima_pag) { ir_a = ir_a + 1 }
            }
            else
            {
                if (x_opc == "ul")
                {
                    if (ir_a != ultima_pag) { ir_a = ultima_pag }
                }
            }
        }
    }
}

```

```

    }
  }
}
if (parseInt(fc.pagina_actual.value) != ir_a)
{
  fc.pagina_actual.value = ir_a
  return true
}
return false
}

```

```
//-->
```

```
//-->
```

```
</script>
```

```
<meta name="Microsoft Border" content="none"></head>
```

```
<body>
```

```
<%
```

```
elsql="SELECT * FROM CUADRO"
```

```
pagina_actual=Request("pagina_actual")
```

```
if pagina_actual="" then
```

```
  pagina_actual=1
```

```
end if
```

```
Set Conn = Server.CreateObject("ADODB.Connection")
```

```
Set RS = Server.CreateObject("ADODB.RecordSet")
```

```
If Abrir_conn(Conn) then
```

```
  r=ejecutar(Conn,RS,elsql)
```

```
  RS.PageSize=1
```

```
  RS.AbsolutePage = pagina_actual
```

```
  Set RSCap = Server.CreateObject("ADODB.RecordSet")
```

```
  CapSQL="SELECT * FROM CAPITULO WHERE CAPITULO.COD_CAPITU="&RS(1)
```

```
  r=ejecutar(Conn,RSCap,Capsql)
```

```
  Set RSSer = Server.CreateObject("ADODB.RecordSet")
```

```
  SerSQL="SELECT * FROM SERIE WHERE SERIE.COD_CUADRO="&RS(0)
```

```
  r=ejecutar(Conn,RSSer,Sersql)
```

```
  ultima_pagina = RS.PageCount
```

```
  lapos = pagina_actual & "/" & ultima_pagina
```

```
  if Not RS.EOF then
```

```
%>
```

```
<form method="post" align="center" name="ForCuadro" onsubmit="return navega(ForCuadro)">
```

```
  <input type="hidden" name="pagina_actual" value="<%=pagina_actual%>"><input type="hidden" name="ultimo" value="<%=ultima_pagina%>"><div
```



```

        las series</strong></td>
    </tr>
    <tr align="center">
        <td
            valign="middle"
            align="center"
            width="40"><input
src="../images/Bprimero.gif" name="I13" alt="IR AL PRIMER CUADRO" type="image"
onclick="x_opc='pr'" WIDTH="37" HEIGHT="37"></td>
        <td
            valign="middle"
            align="center"
            width="39"><input
src="../images/Banterior.gif" name="I11" alt="IR AL ANTERIOR CUADRO" type="image"
onclick="x_opc='an'" WIDTH="37" HEIGHT="37"></td>
        <td
            valign="middle"
            align="center"
            width="41"><input
src="../images/Bproximo.gif" name="I3" alt="IR AL PROXIMO CUADRO" type="image"
onclick="x_opc='si'" WIDTH="37" HEIGHT="37"></td>
        <td width="41"><input src="../images/Bultimo.gif" name="I8" alt="IR AL
ULTIMO CUADRO" type="image" onclick="x_opc='ul'" WIDTH="37" HEIGHT="37"></td>
        <td valign="middle" align="center" width="24"></td>
        <td
            valign="middle"
            align="center"
            width="170"><input
src="../images/Globe.gif" name="I14" alt="Ver Series" type="image"
onclick="x_opc='S'" width="37" height="30"></td>
    </tr>
</table>
</center></div>
</form>
<% end if

Conn.close

set rs = nothing

set Conn = nothing

end if

%>
</body>
</html>

```

' Archivo series.asp

```

<% Response.Expires = 0 %>

<!-- #include file= "adovbs.inc" -->
<!-- #include file= "adofunc.inc" -->
<html>

<head>
<title>Consulta de Series</title>
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<script language="Javascript">

<!--

var insertando = false

var x_opc = "n"

function navega(fc)
{
    var ir_a = parseInt(fc.pagina_actual.value)
    var ultima_pag = parseInt(fc.ultimo.value)
    if (x_opc == "pr")
    {
        if (ir_a != 1) { ir_a = 1 }
    }
    else
    {
        if (x_opc == "an")
        {

```

```

        if (ir_a != 1) { ir_a = ir_a - 1 }
    }
else
    {
    if (x_opc == "si")
        {
        if (ir_a != ultima_pag) { ir_a = ir_a + 1 }
        }
    else
        {
        if (ir_a != ultima_pag) { ir_a = ultima_pag }
        }
    }
}
if (parseInt(fc.pagina_actual.value) != ir_a)
{
fc.pagina_actual.value = ir_a
return true
}
return false
}

```

//-->

//-->

</script>

<meta name="Microsoft Border" content="none"></head>

<body>

<%

elsql="SELECT * FROM SERIE"

pagina_actual=Request("pagina_actual")

if pagina_actual="" then

 pagina_actual=1

end if

Set Conn = Server.CreateObject("ADODB.Connection")

Set RS = Server.CreateObject("ADODB.RecordSet")

If Abrir_conn(Conn) then

 r=ejecutar(Conn,RS,elsql)

 RS.PageSize=1

 RS.AbsolutePage = pagina_actual

 Set RSCua = Server.CreateObject("ADODB.RecordSet")

 CuaSQL="SELECT * FROM CUADRO WHERE CUADRO.COD_CUADRO=" &RS(1)

 r=ejecutar(Conn,RSCua,Cuasql)

 Set RSCap = Server.CreateObject("ADODB.RecordSet")

 CapSQL="SELECT * FROM CAPITULO WHERE CAPITULO.COD_CAPITU=" &RSCua(1)

 r=ejecutar(Conn,RSCap,Capsql)

```

Set RSPer = Server.CreateObject("ADODB.RecordSet")

PerSQL="SELECT * FROM PERIODO WHERE PERIODO.COD_PERIOD=" &RS (5)

r=ejecutar (Conn,RSPer,Persql)

Set RSUni = Server.CreateObject("ADODB.RecordSet")

UniSQL="SELECT * FROM UNIDAD WHERE UNIDAD.COD_UNIDAD=" &RS (6)

r=ejecutar (Conn,RSUni,Unisql)

Set RSSV = Server.CreateObject("ADODB.RecordSet")

RSSQL="SELECT * FROM SER_VAL WHERE SER_VAL.COD_SERIE=" &RS (0)

r=ejecutar (Conn,RSSV,RSsql)

ultima_pagina = RS.PageCount

lapos = pagina_actual & "/" & ultima_pagina

if Not RS.EOF then
%>

<form method="post" align="center" name="ForSerie" onsubmit="return
navega (ForSerie)" >
  <input type="hidden" name="pagina_actual" value="<%=pagina_actual%>"><input
type="hidden" name="ultimo" value="<%=ultima_pagina%>"><div
align="center"><center><table border="0" cellspacing="0" cellpadding="0"
width="627">
  <tr>
    <td width="627" align="center"><table border="0" width="100%"
cellpadding="0" cellspacing="0">
      <tr>
        <td width="79%"><div align="center"><center><p><font
face="Verdana"><strong>Consulta por
Series</strong></font></td>
        <td width="21%" align="center"><div align="center"><center><p><font
face="Arial" color="#0000FF"><small><strong>Serie
:&nbsp;<%=lapos%></strong></small></font></td>
      </tr>
    </table>
    <div align="center"><center><table border="1" width="100%" cellspacing="1"
bgcolor="#C0C0C0">
      <tr>
        <td width="35%"><strong>Código de la Serie:&nbsp;</strong><input
type="text" name="codcua" size="5" value="<%=RS(0)%>"><strong> </strong></td>
        <td width="35%"><strong>Cuadro:</strong> <input type="text"
name="Capitulo" size="5" value="<%=RSCap(1)&"-"&RSCua(2)%>" style="text-align:
center; vertical-align: center"></td>
        <td width="30%" rowspan="4" checked="false" valign="top"><font
face="Arial"><strong><small>Valores
de la serie:<br>
</small></strong></font><select name="SerieValor" size="11">
%>
        <option selected name="Xvalor" value=" <%=RS(0)%>">
<%= (RS(3)+RSSV(1)-1) &" - "&RSSV(2) &" "%> </option>
%>
        <option name="RSCua" value="<%=RSCua(2)%>">
<%= RSCua(2) &" "&RSCua(2) &" "%> </option>
%>
        <option name="RSCap" value="<%=RSCap(1)%>">
<%= RSCap(1) &" "&RSCap(1) &" "%> </option>
%>
      </select></td>
      <td width="70%" colspan="2"><div align="left"><p><strong>Título de la
serie:</strong><textarea rows="2" name="TituloSerie"
cols="49"><%=RS(2)%></textarea><br>
<strong>&nbsp;<%=RS(2)%></strong></div></td>
    </tr>
  </table>
</center></div>

```



```

<body>
<%
    dim T ( )

    CCuadro=Request.form("pagina_actual")
    if CCuadro="" then
        CCuadro=1
    end if
    elsql="SELECT * FROM CUADRO where CUADRO.COD_CUADRO=" & CCuadro
    Set Conn = Server.CreateObject("ADODB.Connection")
    Set RSCua = Server.CreateObject("ADODB.RecordSet")
    If Abrir_conn(Conn) then
        r=ejecutar(Conn,RSCua,elsql)
        Set RSCap = Server.CreateObject("ADODB.RecordSet")

        CapsQL="SELECT * FROM CAPITULO WHERE CAPITULO.COD_CAPITU=" & RSCua(1)

        r=ejecutar(Conn,RSCap,Capsql)

        Set RSSer = Server.CreateObject("ADODB.RecordSet")

        SerSQL="SELECT * FROM SERIE WHERE SERIE.COD_CUADRO=" & RSCua(0)

        r=ejecutar(Conn,RSSer,Sersql)

        Set RSPer = Server.CreateObject("ADODB.RecordSet")

        PerSQL="SELECT * FROM PERIODO WHERE PERIODO.COD_PERIOD=" & RSSer(5)

        r=ejecutar(Conn,RSPer,Persql)

        if Not RSCua.EOF then
            NFil=(RSCua(5)-RSCua(4)+1)*RSPer(2)
            NCol=RSSer.RecordCount
            ReDim T(NFil,NCol)

            a=0
            p=0
            For i=1 to NFil
                p=p+1
                T(i,0)=(RSCua(4)+a)
                if RSPer(2)<>1 then
                    T(i,0)=T(i,0) & "-" & p
                end if
                if p=RSPer(2) then
                    a=a+1
                    p=0
                end if
            next

            j=1
            while not RSSer.eof
                Set RSSV = Server.CreateObject("ADODB.RecordSet")
                SVSQL="SELECT * FROM SER_VAL WHERE SER_VAL.COD_SERIE=" & RSSer(0)
                r=ejecutar(Conn,RSSV,SVsql)
                while not RSSV.eof
                    Fila=(RSSer(3)-RSCua(4))
                    T(Fila+RSSV(1),j)=RSSV(2)
                    RSSV.moveNext
                Wend
                Set RSSV = nothing
                RSSer.MoveNext
                j=j+1
            Wend

            Response.write("<p align=" & OCadena("center") & "><big><strong>Consulta de las
series de un Cuadro</strong></big><br>")

```

```

Response.write("<p><strong>Código de Cuadro:</strong>"&RSCua(0)&"<br>")
Response.write("<strong>Título del Cuadro:</strong>"&RSCua(3)&"<br><hr>")

response.write("<table border="&OCadena(1)&" width="&OCadena((12*ncol+9)&"%")
&" >")
response.write("<tr>")
response.write("<td width="&OCadena(9&"%")&"
align="&OCadena("center")&"><strong>Periodo</strong></td>")
for j=1 to Ncol
response.write("<td width="&OCadena(12&"%")&"
align="&OCadena("center")&"><strong>Serie"&j&"</strong></td>")
next
response.write("</tr>")
For i=1 to Nfil
response.write("<tr>")
response.write("<td width="&OCadena(9&"%")&"
align="&OCadena("center")&">"&T(i,0)&"</td>")
for j=1 to Ncol
if T(i,j)=-1000 then
Valor="N.D."
else
if T(i,j)=0 then
valor=" "
else
Valor=T(i,j)
end if
end if
response.write("<td width="&OCadena(12&"%")&"
align="&OCadena("center")&">"&Valor&"</td>")
next
response.write("</tr>")
next

response.write("</table><hr>")
response.write("<p><big><strong>Leyenda de las series</strong></big><br>")
RSSer.movefirst
i=1
While not RSSer.EOF
response.write("Serie"&i&"-->"&Rsser(2)&"<br>")
RSSer.movenext
i=i+1
Wend
end if

Conn.close

set RSCua = nothing

set Conn = nothing

end if

%>
</body>
</html>

```