

# Desarrollo Computacional del Catálogo de Bienes

Ponsot B., Ernesto

## Ponsot B., Ernesto

Ingeniero de Sistemas  
M.Sc. en Estadística. Profesor  
Agregado de la Facultad de Ciencias  
Económicas y Sociales de la ULA  
[ernesto@ula.ve](mailto:ernesto@ula.ve)

Recibido: 10-03-05  
Revisado: 17-10-05  
Aceptado: 07-12-05

Se presenta un esquema de bases de datos apropiado y su tratamiento práctico en la interfaz usuario-computadora, para la implantación del catálogo de bienes de un inventario estándar. Se ilustra cómo pueden incorporarse los datos del catálogo de manera técnicamente conveniente, en sistemas de información administrativos o contables. Se desprende del análisis la aplicación de las técnicas descritas en la resolución de problemas isomorfos como la informatización del plan de cuentas contable y presupuestario, entre otros. Metodológicamente, se emplean los enfoques de bases de datos Entidad - Asociación y Relacional, aplicados al caso de las asociaciones entre entidades fuertes y débiles, que derivan en un esquema relacional de tablas anidadas jerárquicamente. Para ejemplificar el modelo se utiliza el Sistema Manejador de Bases de Datos Microsoft Access XP®, incluyendo sus interfaces y su versión del lenguaje de consulta estructurado para la recuperación de información.

**Palabras clave:** Inventario, catálogo, bases de datos relacionales.

**RESUMEN**

This work proposes an appropriate database plan and its practical implementation on the user - computer interface, for one standard inventory catalog. We show how the catalog data can be incorporated in a convenient technical manner, into administrative and accounting information systems. From the analysis we derive techniques that can be applied to other equally structured problems like automation of the accounting and budget plan, among other situations. Methodologically, we use the Entity - Relationship and Relational approaches of the database theory, applied to the particular case of associations between strong and weak tables arriving at a relational structure of nested hierarchical tables. To test the model we use the Microsoft Access XP® Database Management System, including its interfaces and its version of the structured query language for information retrieval.

**Key words:** Inventory, catalog, relational databases.

**ABSTRACT**

## 1. Introducción

En el trabajo de Base de Datos (BD), el diseño conceptual para la disposición de las tablas que contendrán tales datos, representa la etapa clave de todo el proceso de desarrollo. Una decisión de diseño equivocada o inconveniente, difícilmente produce sistemas de información de calidad.

Para el diseño del esquema de la base de datos que se presenta en este trabajo, se han empleado los modelos de datos Entidad– Asociación (Chen, 1976) y Relacional (Codd, 1970). Ambos modelos están hoy día más vigentes que nunca y conforman la base de los desarrollos de sistemas informáticos en todo el mundo. Aunque la comunidad de BD ha procurado avanzar en direcciones diferentes como sería, por ejemplo, la orientación a objetos (OO), hasta el presente ningún modelo teórico ha logrado imponerse a los establecidos por estos dos grandes autores. Las bases de datos orientadas a objetos (sus más cercanas rivales), no han logrado superar el paradigma de la independencia entre datos y programas que subyace en las ideas de Codd y Chen y, a lo sumo, la OO ha impactado con solidez y éxito en lenguajes de programación que acceden a bases de datos, pero no hasta el punto de superar el esquema teórico-metodológico que subyace en su implantación (Darwen y Date, 1995).

Por otra parte, el uso de asociaciones entre entidades débiles y fuertes (Korth y Silverschatz, 1987) no es común entre diseñadores de BD, debido a que implica ciertos esfuerzos de profundización en los conceptos teóricos, más allá de los que la mayoría está dispuesto a hacer.

El objetivo general del trabajo que se presenta en estas líneas es entonces describir e ilustrar con precisión los distintos aspectos técnicos que deben ser tomados en consideración para el

correcto desarrollo computacional del catálogo de bienes. Se explica en detalle un planteamiento de diseño conveniente al problema de asociar entidades débiles y fuertes, aplicado a este tema común y eminentemente administrativo.

Supóngase una situación en la que el diseño de base de datos exige un anidamiento o encadenamiento de varias tablas, a partir de sus claves primarias (Date, 1993). Por ejemplo, estúdiese la configuración de una cuenta cualquiera de la contabilidad (1-04-008-07). Aquí, el primer nivel del código (1) es independiente del resto de los niveles, el segundo nivel (04) depende, está inmerso y sólo tiene sentido formando parte de las subcuentas del primer nivel, el tercer nivel (008) depende de los dos anteriores y, por último, el cuarto nivel (07) depende de los tres anteriores. En términos de bases de datos, podría decirse que una de las tablas, independiente del resto, representa una entidad fuerte, una segunda es entidad débil de la primera, una tercera, es entidad débil de la segunda (y en cierta medida lo es también de la primera), y así sucesivamente. Este grupo de tablas “anidadas” (en el ejemplo, las cuatro que definen los distintos niveles del código de la cuenta contable), se emplea luego como fuente en otra tabla (en el ejemplo podría ser el libro diario de la contabilidad) que requiere los datos del último eslabón de la cadena. El problema consiste en implantar de forma apropiada, tanto para el esquema de las tablas, como para la interfaz usuario-computadora, el soporte informático a la situación. Como se verá, la solución emplea elementos avanzados del SMBD<sup>1</sup>, más allá de los asistentes que pueda proporcionar el programa, e incluye generalmente algo de código fuente nativo<sup>2</sup>.

---

<sup>1</sup> Se utiliza en este trabajo, Microsoft Access XP, marca registrada de Microsoft Corporation.

<sup>2</sup> El código fuente nativo del Access es *Visual Basic for Applications*.

Son variadas las situaciones en que este esquema de BD es útil. Como muestra el ejemplo, es de beneficio en el catálogo de la contabilidad, así como en el de un inventario, en la definición de un presupuesto por programas, partidas, niveles, actividades y hasta en la división político territorial de un país (en Venezuela por ejemplo, Entidad–Municipio–Parroquia (INE, 2005)).

Las líneas que siguen, aún cuando dedicadas al caso particular del catálogo de bienes, tienen entonces aplicación práctica en cualquier situación de informatización o sistematización, que requiera incluir en la BD un subconjunto de tablas conceptualmente jerarquizadas entre sí.

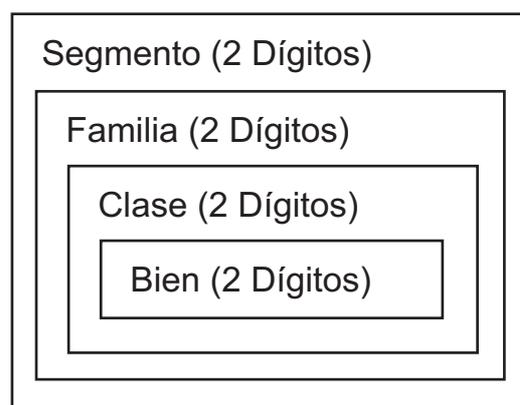
## 2. El catálogo de bienes

Todo inventario requiere la definición de un catálogo de aquellos bienes que serán almacenados. Existen diferentes formas de catalogar los bienes, dependiendo del punto de vista que se adopte. Por ejemplo, desde el punto de vista de la contabilidad, la catalogación se realiza atendiendo al plan de cuentas de la organización. Desde el punto de vista de la producción, posiblemente se deba aplicar una catalogación más sencilla que la contable. Para los efectos de este trabajo, se ha seleccionado un estándar de catalogación sugerido por la *United Nations Standard Products and Services Code* (UNSPSC, 2004), en español, Codificación Estándar de Productos y Servicios de las Naciones Unidas. Este estándar de catalogación es auspiciado por el Programa de Naciones Unidas para el Desarrollo y resulta sencillo para los efectos de exponer cómo puede informatizarse, utilizando técnicas de bases de datos relacionales, el catálogo de un inventario.

La Figura 1 muestra la catalogación UNSPSC, la cual propone una estructura jerárquica de códigos, en cuatro niveles, con dos dígitos cada

uno, en la que se pueden catalogar un máximo de  $100^4$  elementos distintos del inventario (considérese que con dos dígitos pueden representarse hasta 100 elementos, desde el “00” hasta el “99”).

**Figura 1.** Esquema de niveles del catálogo de bienes UNSPSC



Desde el punto de vista de BD, el diseño de esta sección es interesante, por cuanto se supone que el usuario no deba poder incluir un bien al catálogo, que no tenga relación de pertenencia a una clase, ni una clase que no tenga relación de pertenencia a una familia, ni una familia que no la tenga a un segmento. Además, en el dato final (la venta -en este trabajo-, pero lo mismo aplica a la compra o la descarga por obsolescencia, entre otras operaciones), deben aparecer representados estos cuatro niveles, formando el código o identificador único del bien.

Es interesante el diseño ya que entre las formas de pensarlo están, por ejemplo, cuatro tablas independientes una de la otra, o un simple campo de texto que contenga los cuatro elementos. Sin embargo estas formas de plantear la situación no son adecuadas. Hacer tablas independientes, no garantiza que no se incluirá por error, un bien inexistente dentro de una clase,

familia y segmento. Hacer un solo campo de texto, representa múltiples problemas para la estandarización de la información y su posterior recuperación en consultas a la BD, que deban discriminar por estos factores. Entonces, la mejor oportunidad es utilizar tablas anidadas y representar con ellas las jerarquías de los códigos.

### 2.1. Esquema de la base de datos

La estrategia para diseñar la BD del catálogo de bienes, incluye fundamentalmente seis tablas, a saber:

- **tb\_Segmentos**: que almacenará las definiciones de todos los segmentos posibles.
- **tb\_Familias**: que almacenará las definiciones de todas las familias posibles, dentro de cada segmento.
- **tb\_Clases**: que almacenará las definiciones de todas las clases posibles, dentro de cada familia y segmento.
- **tb\_Bienes**: que almacenará las definiciones de todos los bienes posibles, dentro de cada clase, familia y segmento.
- **Unidades**: que almacenará las descripciones

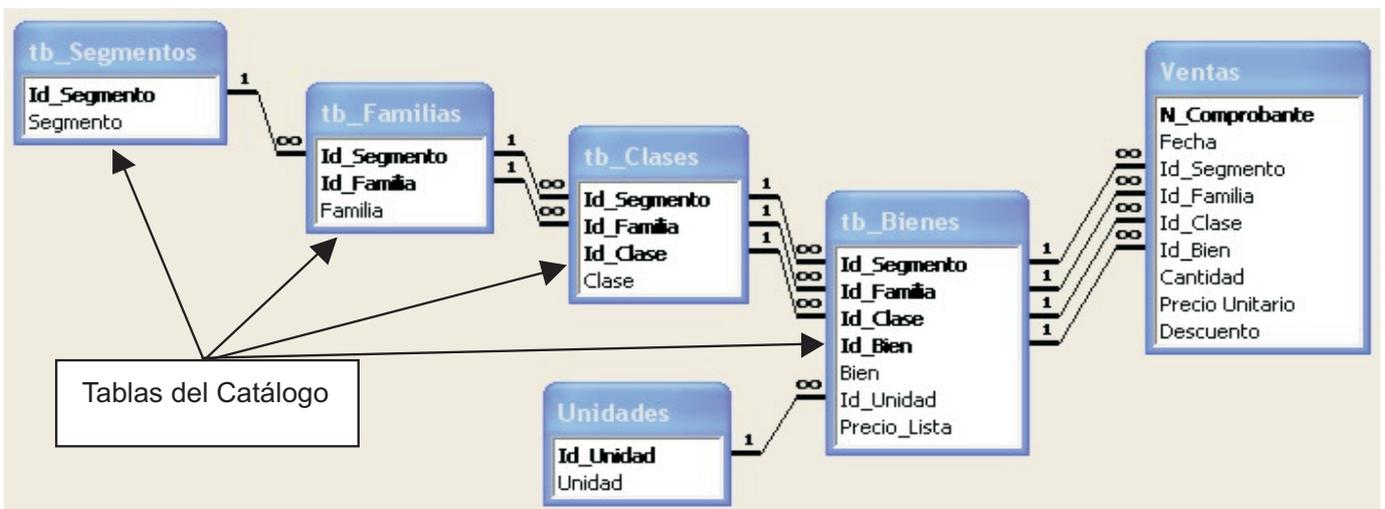
de todas las unidades en que pueden expresarse las cantidades de un bien cualquiera.

- **Ventas**: que almacenará la información de cada transacción de venta de un bien presente en el catálogo del inventario.

La Figura 2 muestra las tablas mencionadas, con su conjunto de campos simplificado al mínimo para efectos de demostración. Los identificadores de las tablas del catálogo (señalados con negritas en la figura) son de tipo texto, aunque se esperan dígitos. En el diseño de BD, siempre que un dígito no necesite ser objeto de operaciones aritméticas y este es el caso, debe tratarse como texto, pues es un formato de almacenamiento compacto y eficiente.

Otro aspecto a resaltar del diseño es la inclusión de un campo para el precio de lista de cada bien en la tabla **tb\_Bienes** y otro para el precio de realización de la transacción en la tabla **Ventas**. La idea es que la organización pueda establecer los precios a los que los bienes deberán ofertarse en primera instancia pero, llegado el

Figura 2. Esquema de la BD del catálogo de bienes simplificado



momento de la venta, éstos puedan ser objeto de rebaja, si se quiere, de acuerdo con un cierto descuento porcentual aplicado de forma discriminada a los clientes. Este elemento de diseño también supone que los precios puedan variar en el tiempo. De ser así, en la tabla *tb\_Bienes* se espera encontrar siempre el precio vigente del bien, pero en la tabla de ventas se encuentra el precio del bien para el momento de la transacción.

La tabla *unidades*, está presente para describir de cada bien, las unidades en que se miden sus cantidades, por ejemplo, kilogramos, litros, toneladas, entre otras.

En *Access*, accediendo a la facilidad 'Relaciones', se pueden asociar unas tablas con otras, de acuerdo con sus claves primarias. Estas asociaciones son la base del modelo relacional, pues establecen validaciones de integridad referencial. Las reglas de integridad referencial para este caso son simples: no puede haber un bien definido en la BD que no se asocie a una clase, una clase que no se asocie a una familia y una familia que no se asocie a un segmento. No puede establecerse una unidad de medida para un bien, que no esté definida previamente en la tabla *Unidades* y, por último, no puede un bien ser vendido, si no se encuentra registrado en la tabla *tb\_Bienes*.

En la Figura 2 puede notarse como las claves de una tabla del catálogo van siendo incluidas en la siguiente. Realmente, lo que se necesita para las transacciones es el bien, es decir, la combinación de las cuatro claves que forma el identificador único de cada bien del inventario. El bien es el objeto susceptible de venta, ni la clase, ni la familia, ni el segmento, lo son. Aquellos se usan para efectos administrativos y de catalogación, muy importantes, pero no terminales.

Otro logro colateral de la automatización de la integridad referencial que permite *Access* (y

también muchos otros SMD profesionales) es la determinación automática de la "cardinalidad de mapeo" (Korth y Silverschatz, 1987, p. 25). La cardinalidad de mapeo se indica en el diagrama de la Figura 2, con los símbolos 1 e  $\infty$ . Esta cardinalidad señala el tipo de relación entre dos tablas, en términos del número general de registros involucrados de cada una. Hay cuatro posibilidades: "Uno a Uno" (1-1), "Uno a Muchos" (1- $\infty$ ), "Muchos a Uno" ( $\infty$ -1) y "Muchos a Muchos" ( $\infty$ - $\infty$ ). En este caso, las cardinalidades son las siguientes:

- De *tb\_Segmentos* a *tb\_Familias*, (1- $\infty$ ): Esto es, en la tabla de familias puede aparecer en muchos registros un mismo segmento, pero en la tabla de segmentos cada uno aparece una sola vez.
- De *tb\_Familias* a *tb\_Clases*, (1- $\infty$ ): Nuevamente, en la tabla de familias cada combinación *Id\_Segmento + Id\_Familia*, esto es, cada identificador de una familia, puede aparecer una sola vez, pero en la tabla *tb\_Clases*, esta combinación puede aparecer en muchos registros. En otras palabras, cada familia es única pero dentro de una familia puede haber muchas clases que pertenezcan a ella.
- De *tb\_Clases* a *tb\_Bienes*, (1- $\infty$ ): Esto es, cada clase es única, pero dentro de una clase puede haber muchos bienes asociados.
- De *tb\_Bienes* a *Unidades*, ( $\infty$ -1): Cada unidad está definida una sola vez en la tabla *Unidades*, pero puede aparecer muchas veces en la tabla *tb\_Bienes*.
- De *tb\_Bienes* a *Ventas*, (1- $\infty$ ): Cada bien es único, pero un mismo bien puede venderse muchas veces.

## 2.2. Los datos

La Figura 3 recrea, para un segmento del

Figura 3. Datos vinculados para un segmento del catálogo

Id_Segmento	Segmento	
- 14	Materiales y Productos de Papel	
+ 10	Materiales de papel	
+ 11	Productos de papel	
- 12	Papel para uso industrial	
+ 15	Cartón y papel para embalaje	
+ 16	Papel de seda	
+ 17	Papeles laminados	
+ 18	Papeles bañados	
+ 19	Papel prensa y de offset	
+ 21	Papeles de base sin bañar	
- 22	Papeles especializados de uso industrial	
+ 01	Papeles para germinar semillas	
	Id_Unidad	
	Precio_Lista	
	Plg	Bs 175,00
*		Bs 0,00

catálogo, cómo quedarían almacenados y vinculados los datos, siguiendo el diseño propuesto.

La visualización jerárquica mostrada en la figura, es una de las características más útiles y atractivas con que cuenta Access en su versión XP. Sólo es posible aprovecharla si se han realizado correctamente las consideraciones de diseño señaladas en la sección anterior. Nótese que se despliegan las tablas que forman el catálogo, esta vez con su contenido y no con su estructura, a partir de la "raíz" del código (los segmentos) hasta las "hojas" del código (los bienes) ocultando, por innecesarios, los códigos de niveles superiores que contienen y deben arrastrarse de una tabla a otra.

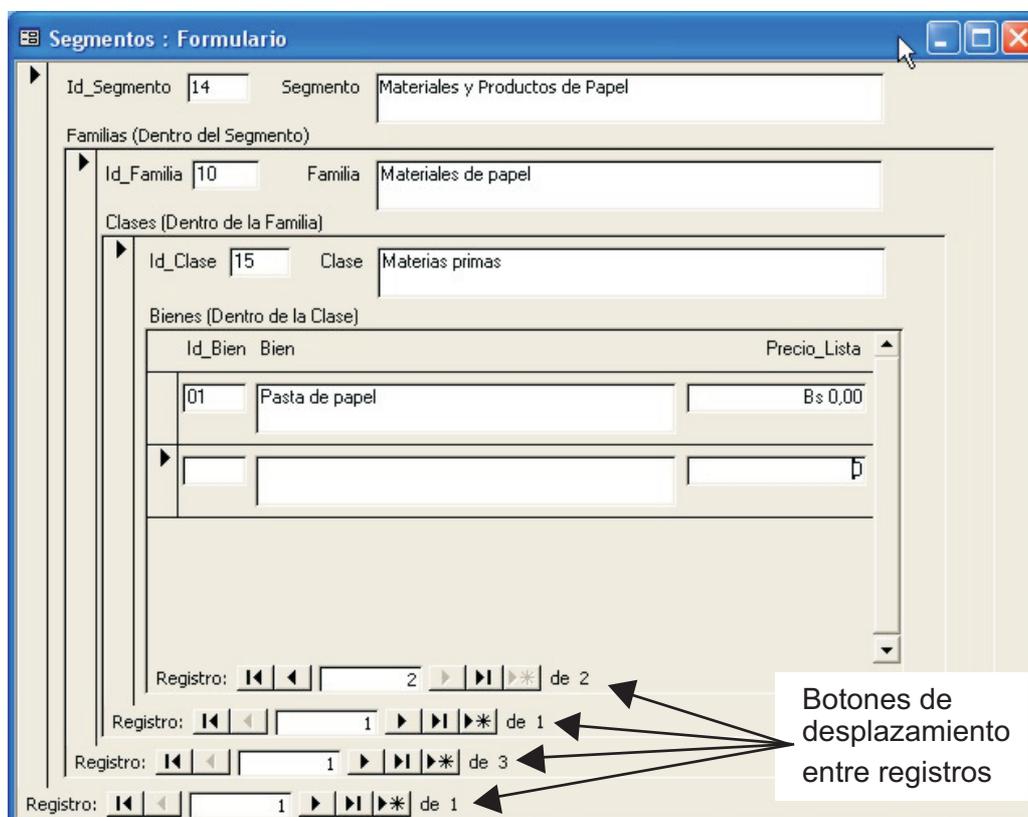
Esto significa que el SMDB gestiona

automáticamente el arrastre de códigos, si han sido satisfechas las condiciones de diseño apropiadas y las reglas de integridad referencial de la BD.

### 2.3. Interfaz de edición de datos

Es habitual que existan múltiples maneras de cargar los datos del catálogo de bienes. Pueden importarse directamente de las hojas de cálculo que les contienen (presentes, por ejemplo, en el sitio Web del estándar reseñado antes); pueden digitalizarse desde una impresión del catálogo y luego transformarse para ser insertados en la BD, o pueden, en la forma más común, transcribirse desde el teclado. Se propone a continuación una forma de implantar esta última alternativa, la más frecuente, basada en ventanas de formulario y subformularios vinculados, de acuerdo con las reglas establecidas en el diseño del esquema de la BD.

Figura 4. Interfaz para la edición de datos del catálogo



La Figura 4 despliega la versión final de un formulario donde pueden editarse los segmentos, dentro de éstos, las familias, dentro, las clases y finalmente, los bienes.

Esta representación es también jerárquica, como la mostrada en la Figura 3. En el diseño, los datos del catálogo de bienes se insertan, eliminan o actualizan, en una sola ventana o formulario que contiene dentro subformularios vinculados de acuerdo con las claves primarias de las entidades precedentes, según sea el caso. Así, para incorporar un nuevo segmento se insertan su código y descripción, respectivamente en los campos Id\_Segmento, Segmento. Para insertar una nueva familia dentro del segmento que esté cargado para el momento, se rellenan los campos

Id\_Familia y Familia, obviando la inclusión del identificador del segmento al que corresponde la familia. De este campo Id\_Segmento, dentro de la tabla tb\_Familias, se encarga automáticamente el SMDB.

En el ejercicio de la Figura 4, por ejemplo, el dato que se incluye dentro de la tabla Familias, es en realidad Id\_Segmento='14', Id\_Familia='10' y Familia='Materiales de papel'. El hecho que el transcriptor no necesite teclear un valor para Id\_Segmento, ahorra tiempo, esfuerzo y minimiza la posibilidad de equivocación en la adjudicación de una familia a un segmento. La navegación entre jerarquías se realiza con los botones de desplazamiento entre registros, señalados en la figura.

### 3. Utilización del catálogo: La venta de bienes

Una vez que los bienes del catálogo se encuentren transcritos a la BD, incluyendo la jerarquía sugerida Segmento / Familia / Clase / Bien, el paso siguiente es la implementación en la interfaz usuario-computadora del registro que debe producir, por ejemplo, la venta de un bien.

La Figura 5 muestra una forma con tres secciones, en la que se implanta la entrada de datos de ventas. La primera sección simplemente Incluye controles de texto para el número del comprobante y la fecha de la venta. Las restantes resultan más interesantes, como se verá a continuación.

#### 3.1. La segunda sección: "Datos del Bien"

En ella se implantan las jerarquías en cuanto a

la conformación del código del bien involucrado en la venta. De la figura 5 se desprende la inclusión de cuatro etiquetas, cuatro cuadros combinados activados para la edición e igual número de cuadros de texto desactivados, vinculados a los identificadores del segmento, familia, clase y bien. En esta sección del formulario el usuario selecciona el bien objeto de la transacción. Cada cuadro combinado permite introducir desde el teclado una cadena de texto de dos dígitos o bien seleccionar un elemento de entre los que se despliegan en la lista del control, como se muestra en la parte superior de la Figura 6.

En cada componente del código del bien, se han establecido las propiedades de las respectivas listas desplegables, de forma similar a la mostrada en la parte inferior de la Figura 6. Estas

Figura 5. Interfaz para la venta de bienes

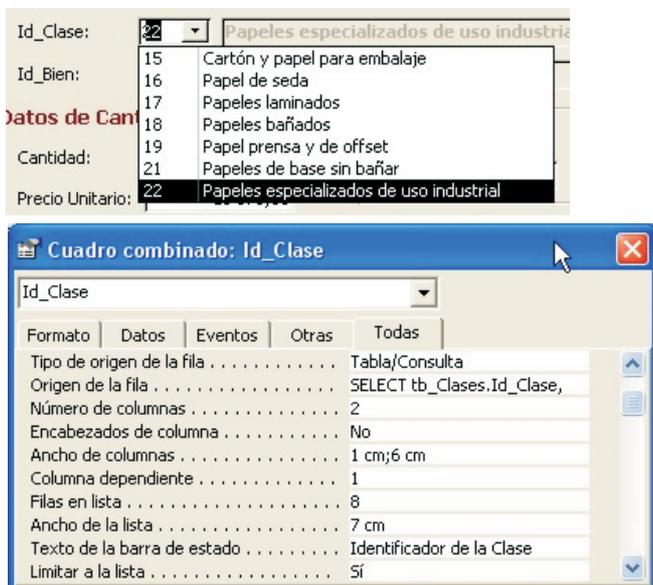
The screenshot shows a software window titled "Ventas : Formulario". It contains three main sections:

- Datos del Comprobante de la Venta:** Includes a text field for "N\_Comprobante:" with the value "1" and a date field for "Fecha:" with the value "19/05/2005".
- Datos del Bien:** A hierarchical selection system with four dropdown menus and corresponding text labels:
  - Id\_Segmento: 14 (Materiales y Productos de Papel)
  - Id\_Familia: 12 (Papel para uso industrial)
  - Id\_Clase: 22 (Papeles especializados de uso industrial)
  - Id\_Bien: 01 (Papeles para germinar semillas)
- Datos de Cantidad y Precio:** Includes text fields for "Cantidad:" (2 Plg), "Precio Unitario:" (Bs 175,00), and "Descuento:" (5,00%).

At the bottom right, there is a button labeled "Esta Venta..." with a sub-field showing "Bs 332,50". Below this are buttons for "Guardar" (represented by a floppy disk icon), "Imprimir" (represented by a printer icon), and "Cancelar".

At the bottom left, there is a record navigation bar: "Registro: [Navigation icons] 1 [Navigation icons] de 1".

**Figura 6.** Propiedades y comportamiento del control Id\_Clase

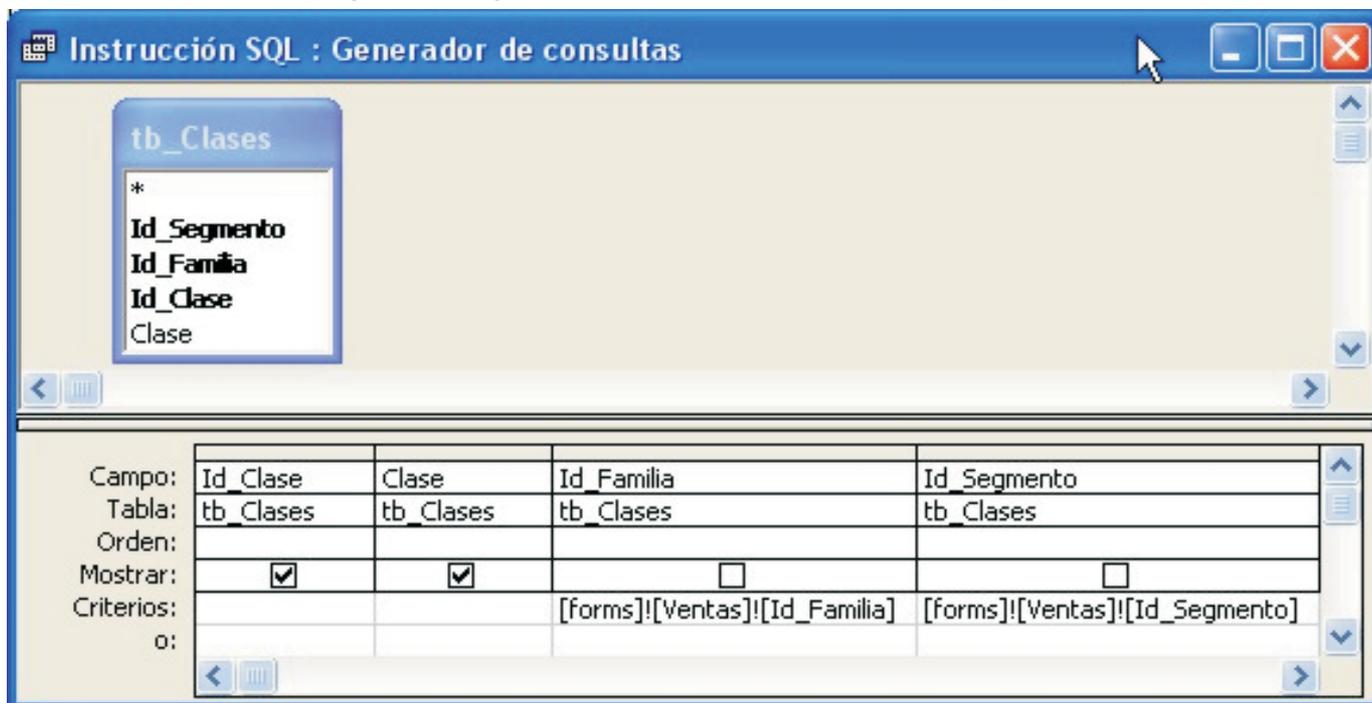


propiedades en general tienen que ver con el formato de presentación y visualización de la lista, excepto la propiedad “Origen de la fila”, que amerita un comentario adicional.

La Figura 6 se refiere al control para la clase del bien cuya venta está registrándose; sin embargo, el tratamiento que se comenta a continuación es válido para el resto de los controles de identificación del bien.

Del esquema de la BD se deduce que la clase que podría editar un usuario en este acto, debe ser tomada de entre aquellas, jerárquicamente subalternas a la familia y al segmento seleccionados (y no de entre todas las clases presentes en la tabla tb\_Clases). Para lograr que el sistema discrimine aquellas clases que pertenecen a la familia y segmento, se debe

**Figura 7.** Origen de la fila para la lista del control Id\_Clase



<sup>3</sup> SQL: *Structured Query Language*, en español, Lenguaje de Consulta Estructurado. Lenguaje de programación procedimental, estándar para la recuperación de información en BD relacionales.

construir una consulta SQL<sup>3</sup>. La Figura 7 muestra la “vista diseño” (Balter, 1999) en la forma que proporciona el SMDB.

La consulta de la Figura 7 se despliega sobre la tabla `tb_Clases` y en ella se seleccionan sólo los campos `Id_Clase` y `Clase`. Los campos que permiten el anidamiento con las familias y segmentos, es decir, `Id_Segmento` e `Id_Familia`, no son devueltos por la consulta, pero participan en ella estableciendo los criterios de filtrado de datos de acuerdo con los valores que, para el momento, estén presentes en el formulario Ventas (específicamente en los controles homónimos). Nótese que tal filtrado se logra incorporando parámetros a la consulta como criterios intersectados, lo que en lógica computacional significa, empleando el operador AND. El resultado se almacena en la BD, como se dijo antes, dentro de la propiedad “Origen de la fila”, en forma de sintaxis SQL, tal como muestra resaltado el cuadro 1.

El cuadro 1 muestra las sintaxis SQL necesarias para los controles de identificación del bien. Resalta que sólo de los segmentos, por representar éstos el primer nivel en la jerarquía de los códigos, se presentan todos los registros de la tabla `tb_Segmentos`, sin discriminación. Los

demás casos efectivamente necesitan condiciones.

### 3.2. La tercera sección: “Datos de Cantidad y Precio”

Esta sección del formulario Ventas (Figura 5), se destina a la captura de la cantidad de bienes vendidos, el precio unitario, el descuento y a partir de ellos, el cómputo del valor efectivo de la venta. Se muestran en ella las unidades en que se expresa la cantidad del bien y se disponen botones apropiados para que el usuario almacene la información recabada, imprima un informe sobre ella o cancele la operación. El Cuadro 2 (página siguiente) contiene el listado VBA necesario para dar el comportamiento apropiado a los controles numéricos. La funcionalidad de los botones se ha dejado fuera del análisis en este trabajo.

La primera de las subrutinas mostradas en el cuadro 2, “Form\_Current” se ejecuta cada vez que el formulario de ventas presenta un nuevo registro. Su función es simplemente recalcular el valor del control “PrecioTotal”, de acuerdo con el resultado de aplicar la función “CalculaPrecioTotal”. De igual forma, las tres subrutinas siguientes actualizan dicho precio total, en los casos en que cambie la cantidad, el precio unitario o el descuento. La función “CalculaPrecioTotal” aplica el descuento al

**Cuadro 1.** SQL para el filtrado de registros de controles en la tabla Ventas

Cuadro de lista	Sintaxis SQL
Id_Segmento	SELECT Id_Segmento, Segmento FROM tb_Segmentos;
Id_Familia	SELECT Id_Familia, Familia FROM tb_Familias WHERE (Id_Segmento=forms!Ventas!Id_Segmento);
Id_Clase	<b>SELECT Id_Clase, Clase FROM tb_Clases WHERE (Id_Familia=forms!Ventas!Id_Familia) And (Id_Segmento=forms!Ventas!Id_Segmento);</b>
Id_Bien	SELECT Id_Bien, Bien FROM tb_Bienes WHERE (Id_Clase=forms!Ventas!Id_Clase) And (Id_Familia=forms!Ventas!Id_Familia) And (Id_Segmento=forms!Ventas!Id_Segmento);

**Cuadro 2.** Código fuente VBA necesario para la sección “Datos de Cantidad y Precio”  
en el módulo Ventas.

```
Private Sub Form_Current()
    [PrecioTotal] = CalculaPrecioTotal
End Sub
Private Sub Cantidad_AfterUpdate()
    [PrecioTotal] = CalculaPrecioTotal
End Sub
Private Sub Precio_Unitario_AfterUpdate()
    [PrecioTotal] = CalculaPrecioTotal
End Sub
Private Sub Descuento_AfterUpdate()
    [PrecioTotal] = CalculaPrecioTotal
End Sub
Private Function CalculaPrecioTotal() As Currency
    CalculaPrecioTotal = (1-Descuento)*Cantidad*[Precio Unitario]
End Function
Private Sub Id_Bien_AfterUpdate()
    [Precio Unitario] = BuscarPrecioLista(Id_Segmento, _
        [Id_Familia], [Id_Clase], [Id_Bien])
    [PrecioTotal] = CalculaPrecioTotal
End Sub
Public Function BuscarPrecioLista(Id_Seg As String, _
    Id_Fam As String, Id_Cla As String, _
    Id_Bie As String) As Currency
    Dim strSQL As String, MiCon As ADODB.Connection
    Dim MiReg As ADODB.Recordset, Precio As Currency
    strSQL = "SELECT Precio_Lista " & _
        "FROM tb_Bienes " & _
        "WHERE (Id_Segmento = '" & Id_Seg & "') and " & _
        "(Id_Familia = '" & Id_Fam & "') and " & _
        "(Id_Clase = '" & Id_Cla & "') and " & _
        "(Id_Bien = '" & Id_Bie & "'"
    Set MiCon = CurrentProject.Connection
    Set MiReg = New ADODB.Recordset
    With MiReg
        .Open strSQL, MiCon, adOpenKeyset
        If .RecordCount <> 1 Then
            Precio = 0
        Else
            Precio = !Precio_Lista
        End If
        .Close
    End With
    MiCon.Close
    BuscarPrecioLista = Precio
End Function
```

producto de la cantidad por el precio unitario.

El aspecto más interesante de las funciones dispuestas para esta sección, es que automáticamente localizan el precio base del bien cuya venta se está registrando y lo cargan en el control "Precio Unitario". El usuario es libre de alterar luego este valor, pero la puesta por defecto seguramente le resultará muy útil en la mayoría de los casos. Esto ocurre cada vez que se actualiza el identificador del bien y además se recalcula nuevamente el precio total de la venta para el nuevo bien registrado.

La función "BuscarPrecioLista", que implementa esta característica, es quizás la de mayor complejidad mostrada en este trabajo. Pasándole como parámetros los identificadores del segmento, familia, clase y bien, ella encuentra en la tabla de bienes el precio de lista que coincide con dichos parámetros. Para hacerlo se construye una cadena de caracteres que incluye los criterios deseados, dentro de una sintaxis SQL válida. Sobre una conexión ADO<sup>4</sup> a la base de datos, la sintaxis SQL construida dinámicamente se ejecuta y el resultado es devuelto en una variable *Recordset* o de conjunto de registros. Si este *Recordset* contuviera un número de registros distinto de 1, el precio es inválido (o bien no se encontró el registro coincidente con los parámetros, o bien se encontró una ambigüedad no esperada), en caso contrario, el precio es válido y devuelto por la función.

#### 4. Conclusiones

Se han descrito e ilustrado con precisión distintos aspectos técnicos que deben ser tomados en consideración para el correcto desarrollo computacional del catálogo de bienes,

objetivo del trabajo propuesto. Un buen diseño pasa necesariamente por un esquema apropiado de bases de datos, sin lo cual, puede llegar a complicarse la implantación computacional de jerarquías o anidamientos, necesarios para el tratamiento de códigos de este tipo.

Las características profesionales que habría que dar a un sistema de información administrativo con soporte de inventarios, están cubiertas teóricamente desde la perspectiva de las asociaciones entre entidades fuertes y débiles que se conforman en niveles, por lo tanto, lo aquí expuesto sirve perfectamente como solución ante cualquier situación de este tipo. Resaltan los problemas computacionales de la codificación de cuentas contables o presupuestarias, estructuralmente isomorfos al problema de la codificación de inventarios.

Aunque no puede obviarse del todo la escritura de código fuente nativo para la solución profesional del problema, un diseño jerárquico apropiado lleva al mínimo el esfuerzo de programación requerido.

Por razones de espacio, se ha mostrado una ínfima parte de la funcionalidad que puede darse sobre un catálogo de inventarios. Por supuesto, así como se manejan las ventas, puede hacerse para las compras, devoluciones, etc. Muchos otros datos de interés podrán ser añadidos al diseño mostrado, pero en esencia, la base de la computarización del catálogo es la presentada.

Se espera que estas ideas sean de utilidad a todo aquel analista o investigador interesado en el desarrollo de sistemas de información administrativos o contables, en los cuales el tratamiento de inventarios es siempre materia de primera importancia.

---

<sup>4</sup>ADO: *ActiveX Data Object*. Tecnología estándar de Microsoft para la conexión a fuentes de datos.

## Referencias bibliográficas

---

- Balter, Alison. (1999). **Mastering Microsoft Access 2000 Development. The Authoritative Solution.** 2ª Edición. Sams Publishing, Indianapolis, Estados Unidos.
- Chen, Peter. (1976). **The Entity - Relationship Model Toward a Unified View of Data.** ACM Transactions on Database Systems, Vol. 1, N° 1, Marzo, EEUU, pp 9-36.
- Codd, E.F. (1970). **A Relational Model of Data for Large Shared Data Banks.** Communications of the ACM, Vol. 13, N° 6, Junio, EEUU, pp 377-387.
- Darwen, H. y Date, C. (1995). **Introducing ... The Third Manifesto.** Database Programming & Design 1(8). Enero, EEUU, pp 25-35.
- Date, C. (1993). **Introducción a los Sistemas de Bases de Datos.** Vol. 1 de 2. 5ª Edición. Addison-Wesley Iberoamericana S. A., Wilmington - Estados Unidos.
- Korth, H. Silverschatz, A. (1987). **Fundamentos de Bases de Datos.** 1ª Edición. McGraw-Hill Book Company, Inc., México - México.

## Consultas electrónicas

---

- Sitio Web del Instituto Nacional de Estadística de la República Bolivariana de Venezuela, <http://www.ine.gov.ve>, consultado el 1º de mayo de 2005.
- United Nations Standard Products and Services Code. Development Program, <http://www.unspsc.com/>, consultado el 16 de abril de 2005.