

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS (PLAN 2001)

PFC

APLICACIÓN DE GESTIÓN EN VISUAL BASIC 6.0 INMOBIFICH 2.45



REALIZADO POR:
Miguel Monzó Pastor

DIRIGIDO POR:
Maria Ángeles Pastor Cubillo

Valencia, 29 de Julio de 2011

INDICE

1. INTRODUCCIÓN	3
1.1 Programas secuenciales, interactivos y orientados a objetos.....	3
1.2 Programas para el entorno Windows.....	4
1.3 El entorno de programación Visual Basic 6.0.....	8
2. LENGUAJE BASIC	10
2.1 Introducción.....	10
2.2 Comentarios y otras utilidades en la programación con Visual Basic.....	10
2.3 Proyectos y módulos.....	11
2.4 Variables.....	15
2.5 Operadores.....	18
2.6 Sentencias de control.....	20
2.7 Funciones y procedimientos.....	24
2.8 Arrays.....	28
2.9 Estructuras: Sentencias Type.....	30
3. BASES DE DATOS EN VISUAL BASIC 6 (CONTROL DATA)	33
Definiciones.	
4. PROYECTO EN VISUAL BASIC 6: INMOBIFICH 2.45	36
4.1 Introducción teórica.....	36
4.2 Descripción del proyecto.....	37
4.3 Planificación del proyecto.....	38
4.3.1 Análisis de requerimientos.....	39
4.3.2 Diseño de B.D (Bases de datos).....	39
4.3.3 Codificación.....	39
4.3.4 Pruebas.....	40
4.3.5 Instalación.....	40
4.3.6 Mantenimiento.....	40
4.4 Problema que resuelve.....	41
4.5 Ventajas que aporta la aplicación.....	41
4.6 Componente software de la aplicación.....	41
4.7 Tecnologías empleadas.....	42
4.8 Esquema de la base de datos.....	42
Tabla Descripción.....	42
Tabla Inmueble.....	45
5. DOCUMENTACIÓN DEL PROYECTO	48
5.1 Descripción de los formularios.....	48
5.2 Pantallazos de los formularios.....	49
5.3 Código significativo.....	55
FormInmobiliaria.....	55
FormCalc.....	102

1. INTRODUCCIÓN

Visual Basic 6.0 es un lenguaje de programación visual, también llamado lenguaje de 4ª Generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

Visual Basic 6.0 es también un programa *basado en objetos*, aunque no *orientado a objetos* como **C++** o **Java**. La diferencia está en que **Visual Basic 6.0** utiliza **objetos** con **propiedades** y **métodos**, pero carece de los mecanismos de **herencia** y **polimorfismo** propios de los verdaderos lenguajes orientados a objetos como **Java** y **C++**.

1.1 PROGRAMAS SECUENCIALES, INTERACTIVOS Y ORIENTADOS A EVENTOS

Existen distintos tipos de programas. En los primeros tiempos de los ordenadores los programas eran de tipo **secuencial** (también llamados tipo **batch**) Un programa secuencial es un programa que se arranca, lee los datos que necesita, realiza los cálculos e imprime o guarda en el disco los resultados. De ordinario, mientras un programa secuencial está ejecutándose no necesita ninguna intervención del usuario. A este tipo de programas se les llama también *programas basados u orientados a procedimientos o a algoritmos (procedural languages)*. Este tipo de programas siguen utilizándose ampliamente en la actualidad, pero la difusión de los PCs ha puesto de actualidad otros tipos de programación.

Los programas **interactivos** exigen la intervención del usuario en tiempo de ejecución, bien para suministrar datos, bien para indicar al programa lo que debe hacer por medio de menús. Los programas interactivos limitan y orientan la acción del usuario. Un ejemplo de programa interactivo podría ser **Matlab**.

Por su parte los programas **orientados a eventos** son los programas típicos de **Windows**, tales como **Netscape**, **Word**, **Excel** y **PowerPoint**. Cuando uno de estos programas ha arrancado, lo único que hace es quedarse a la espera de las acciones del usuario, que en este caso son llamadas **eventos**. El usuario dice si quiere abrir y modificar un fichero existente, o bien comenzar a crear un fichero desde el principio. Estos programas pasan la mayor parte de su tiempo esperando las acciones del

usuario (eventos) y respondiendo a ellas. Las acciones que el usuario puede realizar en un momento determinado son variadísimas, y exigen un tipo especial de programación: **la programación orientada a eventos**. Este tipo de programación es sensiblemente más complicada que la secuencial y la interactiva.

1.2 PROGRAMAS PARA EL ENTORNO WINDOWS

Visual Basic 6.0 está orientado a la realización de programas para **Windows**, pudiendo incorporar todos los elementos de este entorno informático: ventanas, botones, cajas de diálogo y de texto, botones de opción y de selección, barras de desplazamiento, gráficos, menús, etc.

Prácticamente todos los elementos de interacción con el usuario de los que dispone **Windows 95/98/NT** pueden ser programados en **Visual Basic 6.0** de un modo muy sencillo. En ocasiones bastan unas pocas operaciones con el ratón y la introducción a través del teclado de algunas sentencias para disponer de aplicaciones con todas las características de **Windows 95/98/NT**.

1.2.1 Modo de Diseño y Modo de Ejecución

La aplicación **Visual Basic** de **Microsoft** puede trabajar de dos modos distintos: en modo de diseño y en modo de ejecución. En **modo de diseño** el usuario construye interactivamente la aplicación, colocando **controles** en el **formulario**, definiendo sus **propiedades**, y desarrollando **funciones** para gestionar los **eventos**.

La aplicación se prueba en **modo de ejecución**. En ese caso el usuario actúa sobre el programa (introduce **eventos**) y prueba cómo responde el programa. Hay algunas **propiedades** de los **controles** que deben establecerse en modo de diseño, pero muchas otras pueden cambiarse en tiempo de ejecución desde el programa escrito en **Visual Basic 6.0**, en la forma en que más adelante se verá. También hay **propiedades** que sólo pueden establecerse en modo de ejecución y que no son visibles en modo de diseño.

1.2.2 Formularios y Controles

Cada uno de los elementos gráficos que pueden formar parte de una aplicación típica de **Windows 95/98/NT** es un tipo de **control**: los botones, las cajas de diálogo y de texto, las cajas de selección desplegables, los botones de opción y de selección, las barras de desplazamiento horizontales y verticales, los gráficos, los menús, y muchos otros tipos de elementos son controles para **Visual Basic 6.0**. Cada control debe tener un **nombre** a través del cual se puede hacer referencia a él en el programa. **Visual Basic 6.0** proporciona nombres *por defecto* que el usuario puede modificar. En la terminología de **Visual Basic 6.0** se llama **formulario** (*form*) a una ventana. Un formulario puede ser considerado como una especie de contenedor para los controles. Una aplicación puede tener varios formularios, pero un único formulario puede ser suficiente para las aplicaciones más sencillas. Los formularios deben también tener un nombre, que puede crearse siguiendo las mismas reglas que para los controles.

1.2.3 Objetos y Propiedades

Los formularios y los distintos tipos de controles son entidades genéricas de las que puede haber varios ejemplares concretos en cada programa. En *programación orientada a objetos* (más bien *basada en objetos*, habría que decir) se llama **clase** a estas entidades genéricas, mientras que se llama **objeto** a cada ejemplar de una clase determinada. Por ejemplo, en un programa puede haber varios botones, cada uno de los cuales es un **objeto** del tipo de control **command button**, que sería la **clase**.

Cada formulario y cada tipo de control tienen un conjunto de **propiedades** que definen su aspecto gráfico (tamaño, color, posición en la ventana, tipo y tamaño de letra, etc.) y su forma de responder a las acciones del usuario (si está activo o no, por ejemplo). Cada propiedad tiene un **nombre** que viene ya definido por el lenguaje.

Por lo general, las propiedades de un **objeto** son datos que tienen valores lógicos (*True, False*) o numéricos concretos, propios de ese objeto y distintos de las de otros objetos de su clase. Así pues, cada clase, tipo de objeto o control tiene su conjunto de propiedades, y cada objeto o control concreto tiene unos valores determinados para las propiedades de su clase.

Casi todas las propiedades de los objetos pueden establecerse en tiempo de diseño y también -casi siempre- en tiempo de ejecución. En este segundo caso se accede a sus valores por medio de las sentencias del programa, en forma análoga a como se accede a cualquier variable en un lenguaje

de programación. Para ciertas propiedades ésta es la única forma de acceder a ellas.

Se puede **acceder a una propiedad** de un objeto por medio del **nombre del objeto** a que pertenece, seguido de un **punto** y el **nombre de la propiedad**, como por ejemplo **optColor.objName**. En el siguiente apartado se estudiarán las reglas para dar nombres a los objetos.

1.2.4 Nombres de objetos

En principio cada objeto de **Visual Basic 6.0** debe tener un nombre, por medio del cual se hace referencia a dicho objeto. El nombre puede ser el que el usuario desee, e incluso **Visual Basic 6.0** proporciona **nombres por defecto** para los diversos controles. Estos nombres por defecto hacen referencia al tipo de control y van seguidos de un número que se incrementa a medida que se van introduciendo más controles de ese tipo en el formulario (por ejemplo **VScroll1**, para una barra de desplazamiento *-scroll bar-* vertical, **HScroll1**, para una barra horizontal, etc.).

Los **nombres por defecto no son adecuados** porque hacen referencia al tipo de control, pero no al uso que de dicho control está haciendo el programador. Por ejemplo, si se utiliza una barra de desplazamiento para introducir una temperatura, conviene que su nombre haga referencia a la palabra **temperatura**, y así cuando haya que utilizar ese nombre se sabrá exactamente a qué control corresponde. Un nombre adecuado sería por ejemplo **hsbTemp**, donde las tres primeras letras indican que se trata de una *horizontal scroll bar*, y las restantes (empezando por una mayúscula) que servirá para definir una *temperatura*.

Existe una convención ampliamente aceptada que es la siguiente: *se utilizan siempre tres letras minúsculas que indican el tipo de control, seguidas por otras letras (la primera mayúscula, a modo de separación) libremente escogidas por el usuario, que tienen que hacer referencia al uso que se va a dar a ese control*. La Tabla 1.1 muestra las abreviaturas de los controles más usuales, junto con la nomenclatura inglesa de la que derivan. En este mismo capítulo se verán unos cuantos ejemplos de aplicación de estas reglas para construir nombres.

Abreviatura	Control	Abreviatura	Control
chk	check box	cbo	combo y drop-list box
cmd	command button	dir	dir list box
drv	drive list box	fil	file list box
frm	form	fra	frame
hsb	horizontal scroll bar	img	image
lbl	label	lin	line
lst	list	mnu	menu
opt	option button	pct	pictureBox
shp	shape	txt	text edit box
tmr	timer	vsb	vertical scroll bar

1.2.5 Eventos

Ya se ha dicho que las acciones del usuario sobre el programa se llaman **eventos**. Son eventos típicos el clicar sobre un botón, el hacer doble clic sobre el nombre de un fichero para abrirlo, el arrastrar un icono, el pulsar una tecla o combinación de teclas, el elegir una opción de un menú, el escribir en una caja de texto, o simplemente mover el ratón. Más adelante se verán los distintos tipos de eventos reconocidos por **Windows 95/98/NT** y por **Visual Basic 6.0**.

Cada vez que se produce un evento sobre un determinado tipo de control, **Visual Basic 6.0** arranca una determinada **función** o **procedimiento** que realiza la acción programada por el usuario para ese evento concreto. Estos procedimientos se llaman con un nombre que se forma a partir del nombre del objeto y el nombre del evento, separados por el carácter (**_**), como por ejemplo **txtBox_click**, que es el nombre del procedimiento que se ocupará de responder al evento **click** en el objeto **txtBox**.

1.2.6 Métodos

Los **métodos** son funciones que también son llamadas desde programa, pero a diferencia de los procedimientos no son programadas por el usuario, sino que vienen ya pre-programadas con el lenguaje. Los métodos realizan tareas típicas, previsibles y comunes para todas las aplicaciones. De ahí que vengan con el lenguaje y que se libere al usuario de la tarea de programarlos. Cada tipo de objeto o de control tiene sus propios métodos.

Por ejemplo, los controles gráficos tienen un método llamado **Line** que se encarga de dibujar líneas rectas. De la misma forma existe un método llamado **Circle** que dibuja circunferencias y arcos de circunferencia. Es obvio que el dibujar líneas rectas o circunferencias es una tarea común para todos los programadores y que **Visual Basic 6.0** da ya resuelta.

1.2.7 Proyectos y ficheros

Cada aplicación que se empieza a desarrollar en **Visual Basic 6.0** es un nuevo **proyecto**. Un proyecto comprende otras componentes más sencillas, como por ejemplo los **formularios** (que son las ventanas de la interface de usuario de la nueva aplicación) y los **módulos** (que son conjuntos de funciones y procedimientos sin interface gráfica de usuario).

1.3 EL ENTORNO DE PROGRAMACIÓN VISUAL BASIC 6.0

Cuando se arranca **Visual Basic 6.0** aparece en la pantalla una configuración similar a la mostrada en la Figura 1.1. En ella se pueden distinguir los siguientes elementos:

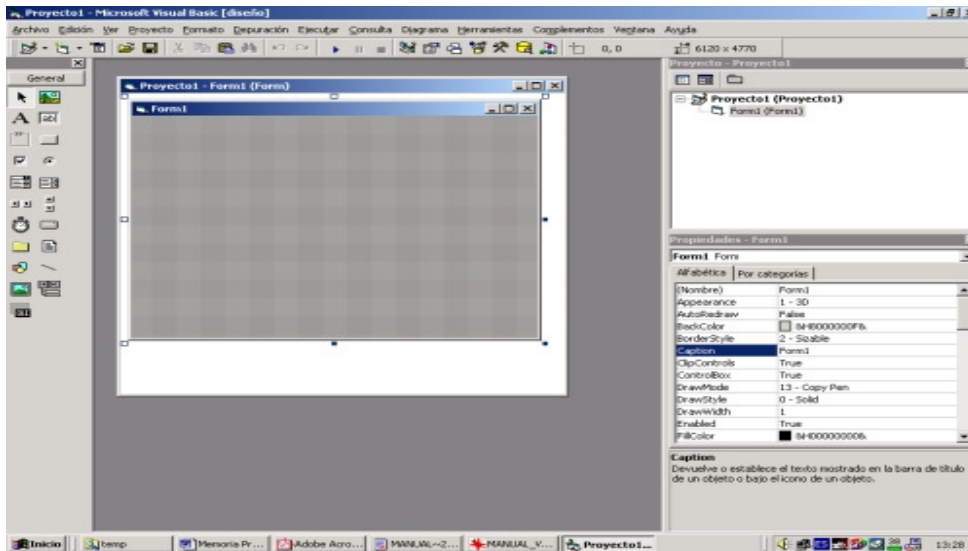
1. La **barra de títulos**, la **barra de menús** y la **barra de herramientas** de **Visual Basic 6.0** en modo **Diseño** (parte superior de la pantalla).
2. **Caja de herramientas (toolbox)** con los controles disponibles (a la izquierda de la ventana).
3. **Formulario (form)** en gris, en que se pueden ir situando los controles (en el centro). Está dotado de una rejilla (**grid**) para facilitar la alineación de los controles.

4. Ventana de *proyecto*, que muestra los formularios y otros módulos de programas que forman parte de la aplicación (arriba a la derecha).

5. Ventana de *Propiedades*, en la que se pueden ver las propiedades del objeto seleccionado o del propio formulario (en el centro a la derecha). Si esta ventana no aparece, se puede hacer visible con la tecla <F4>.

6. Ventana *FormLayout*, que permite determinar la forma en que se abrirá la aplicación cuando comience a ejecutarse (abajo a la derecha).

Existen otras ventanas para edición de código (**Code Editor**) y para ver variables en tiempo de ejecución con el *depurador* o **Debugger** (ventanas **Immediate**, **Locals** y **Watch**). Todo este conjunto de herramientas y de ventanas es lo que se llama un *entorno integrado de desarrollo* o IDE (**Integrated Development Environment**).



2. LENGUAJE BASIC

2.1 INTRODUCCIÓN

Los programas de **Visual Basic** se dice que son *interpretados* y no compilados ya que el código no se convierte a código máquina sino que hay otro programa que durante la ejecución "interpreta" las líneas de código que ha escrito el programador.

2.2 COMENTARIOS Y OTRAS UTILIDADES EN LA PROGRAMACIÓN CON VISUAL BASIC

Visual Basic 6.0 interpreta que todo lo que está a la derecha del **carácter (')** en una línea cualquiera del programa es un **comentario** y no lo tiene en cuenta para nada. El comentario puede empezar al comienzo de la línea o a continuación de una instrucción que debe ser ejecutada, por ejemplo:

```
' Esto es un comentario
```

```
A = B*x+3.4 ' también esto es un comentario
```

Otro aspecto práctico en la programación es la posibilidad **de escribir una sentencia en más de una línea**. En el caso de sentencias bastante largas es conveniente cortar la línea para que entre en la pantalla. En otro caso la lectura del código se hace mucho más pesada. Para ello es necesario dejar un **espacio en blanco** al final de la línea y escribir el **carácter (_)** tal y como se muestra en el siguiente ejemplo:

```
str1 = "Londres" : str2 = "París" 'Se inicializan las variables
```

```
Frase = "Me gustaría mucho viajar a " & _
```

```
str1 & " y a " & str2
```

'El contenido de Frase sería: "Me gustaría mucho viajar a Londres y a París

Una limitación a los comentarios en el código es que no se pueden introducir en una línea en la que se ha introducido el carácter de continuación (_).

La sintaxis de **Visual Basic 6.0** permite también incluir **varias sentencias en una misma**

línea. Para ello las sentencias deben ir separadas por el **carácter dos puntos (:)**. Por ejemplo:

```
m = a : n = b : resto = m Mod n ' Tres sentencias en una línea
```

2.3 PROYECTOS Y MÓDULOS

Un **proyecto** realizado en **Visual Basic 6.0** es el conjunto de todos los ficheros o **módulos** necesarios para que un programa funcione. La información referente a esos ficheros se almacena en un fichero del tipo **ProjectName.vbp**. La extensión ***.vbp** del fichero hace referencia a **Visual Basic Project**.

Si se edita este fichero con cualquier editor de texto se comprueba que la información que almacena es la localización en los discos de los módulos que conforman ese proyecto, los controles utilizados (ficheros con extensión **.ocx**), etc. En el caso más simple un proyecto está formado por un único formulario y constará de dos ficheros: el que define el proyecto (***.vbp**) y el que define el formulario (***.frm**).

Los módulos que forman parte de un proyecto pueden ser de varios tipos: aquellos que están asociados a un formulario (***.frm**), los que contienen únicamente líneas de código **Basic** (***.bas**) llamados **módulos estándar** y los que definen agrupaciones de código y datos denominadas clases (***.cls**), llamados **módulos de clase**.

Un módulo ***.frm** está constituido por un **formulario** y toda la información referente a los **controles** (y a sus propiedades) en él contenidos, además de todo el código programado en los **eventos** de esos controles y, en el caso de que existan, las **funciones** y **procedimientos** propios de ese formulario. En general se llama **función** a una porción de código independiente que realiza una determinada actividad. En **Visual Basic** existen dos tipos de funciones: las llamadas **function**, que se caracterizan por tener valor de retorno, y los **procedimientos** o *procedures*, que no lo tienen. En otros lenguajes, como C/C++/Java, las **function** realizan los dos papeles.

Un módulo de código estándar ***.bas** contendrá una o varias funciones y/o procedimientos, además de las variables que se desee, a los que se podrá acceder desde cualquiera de los módulos que forman el proyecto.

2.3.1 Ámbito de las variables y los procedimientos

Se entiende por **ámbito** de una variable (ver Apartado 3.3.1, en la página 25) la parte de la aplicación donde la variable es *visible* (accesible) y por lo tanto puede ser utilizada en cualquier

expresión.

2.3.1.1 Variables y funciones de ámbito local

Un módulo puede contener variables y procedimientos o funciones **públicos** y **privados**. Los **públicos** son aquellos a los que se puede acceder libremente desde cualquier punto del proyecto. Para definir una variable, un procedimiento o una función como **público** es necesario preceder a la definición de la palabra **Public**, como por ejemplo:

```
Public Variable1 As Integer
Public Sub Procedimiento1 (Parametro1 As Integer, ...)
Public Function Funcion1 (Parametro1 As Integer, ...) As Integer
```

Para utilizar una variable **Public** o llamar a una función **Public** definidas en un formulario desde otro módulo se debe preceder el nombre de la variable o procedimiento con el nombre del formulario al que pertenece, como por ejemplo:

```
Modulo1.Variable1
Call Modulo1.Procedimiento1(Parametro1, ...)
Retorno = Modulo1.Funcion1(Parametro1, ...)
```

Sin embargo si el módulo al que pertenecen la variable o el procedimiento **Public** es un módulo estándar (*.bas) no es necesario poner el nombre del módulo más que si hay coincidencia de nombres con los de otro módulo también estándar. Una variable **Private**, por el contrario, no es accesible desde ningún otro módulo distinto de aquél en el que se haya declarado.

Se llama variable **local** a una variable definida dentro de un procedimiento o función. Las variables locales no son accesibles más que en el procedimiento o función en que están definidas. Una variable **local** es reinicializada (a cero, por defecto) cada vez que se entra en el *procedimiento*. Es decir, una variable **local** no conserva su valor entre una llamada al *procedimiento* y la siguiente. Para hacer que el valor de la variable se conserve hay que declarar la variable como **static** (como por ejemplo: *Static n As Integer*). **Visual Basic** inicializa una variable estática

solamente la primera vez que se llama al *procedimiento*. Para declarar una variable estática, se utiliza la palabra **Static** en lugar de **Dim**. Un poco más adelante se verá que **Dim** es una palabra utilizada para crear variables. Si un procedimiento se declara **Static** todas sus variables locales tienen carácter **Static**.

2.3.1.2 Variables y funciones de ámbito global

Se puede acceder a una variable o función global desde cualquier parte de la aplicación. Para hacer que una variable sea global, hay que declararla en la **parte general** de un módulo ***.bas** o de un formulario de la aplicación. Para declarar una variable global se utiliza la palabra **Public**. Por ejemplo:

```
Public var1_global As Double, var2_global As String
```

De esta forma se podrá acceder a las variables **var1_global**, **var2_global** desde todos los formularios

La diferencia entre las variables y/o procedimientos **Public** de los formularios y de los módulos estándar está en que las de los procedimientos deben ser cualificadas (precedidas) por el nombre del formulario cuando se llaman desde otro módulo distinto, mientras que las de un módulo estándar (***.bas**) sólo necesitan ser cualificadas si hay colisión o coincidencia de nombres.

Tipo de variable	Lugar de declaración	Accesibilidad
Global o Public	Declaraciones de *.bas	Desde todos los formularios
Dim o Private	Declaraciones de *.bas	Desde todas las funciones de ese módulo
Public	Declaraciones de *.frm	Desde cualquier procedimiento del propio formulario y desde otros precedida del nombre

		del modulo en el que se ha declarado
Dim o Private	Declaraciones de *.frm	Desde cualquier procedimiento del propio formulario
Dim	Cualquier procedimiento de un módulo	Desde el propio procedimiento

2.4 VARIABLES

2.4.1 Identificadores

Visual Basic 6.0, como todos los demás lenguajes de programación, tiene sus propias reglas para elegir los **identificadores**. Los usuarios pueden elegir con gran libertad los nombres de sus variables y funciones, teniendo siempre cuidado de respetar las reglas del lenguaje y de no utilizar un conjunto de **palabras reservadas (keywords)**, que son utilizadas por el propio lenguaje.

2.4.2 Variables y constantes

Para declarar un dato como constante únicamente es necesario utilizar la palabra **Const** en la declaración de la variable. Si durante la ejecución se intenta variar su valor se producirá un error.

Ejemplos:

Const MyVar = 459 ' Las constantes son privadas por defecto.

Public **Const** MyString = "HELP" ' Declaración de una constante pública.

Private **Const** MyInt As Integer = 5 ' Declaración de un entero constante.

Const Str = "Hi", PI As Double = 3.14 ' Múltiples constantes en una línea.

Visual Basic 6.0 tiene sus propias constantes. Algunas ya se han visto al hablar de los colores. En general estas constantes empiezan por ciertos caracteres como **vb** (u otros similares que indican a que grupo pertenecen) y van seguidas de una o más palabras que indican su significado.

2.4.3 Nombres de variables

El nombre de una variable (o de una constante) tiene que comenzar siempre por una letra y puede tener una longitud hasta 255 caracteres. No se admiten espacios o caracteres en blanco, ni puntos (.), ni otros caracteres especiales.

Los caracteres pueden ser letras, dígitos, el carácter de subrayado () y los caracteres de declaración del tipo de la variable (% , & , # , ! , @ , y \$). El nombre de una variable no puede ser una

palabra reservada del lenguaje (*For, If, Loop, Next, Val, Hide, Caption, And, ...*).

Visual Basic 6.0 no distingue entre minúsculas y mayúsculas. Por tanto, las variables **LongitudTotal** y **longitudtotal** son consideradas como idénticas (la misma variable). En **Visual Basic 6.0** es habitual utilizar las letras mayúsculas para separar las distintas palabras que están unidas en el nombre de una variable, como se ha hecho anteriormente en la variable **LongitudTotal**. La declaración de una variable o la primera vez que se utiliza determinan cómo se escribe en el resto del programa.

También es habitual entre los programadores, aunque no obligado, el utilizar nombres con todo mayúsculas para los nombres de las constantes simbólicas, como por ejemplo **PI**.

2.4.4 Tipos de datos

Tipo	Descripción	Carácter de declaración	Rango
Boolean	Binario		True o False
Byte	Entero corto		0 a 255
Integer	Entero (2 bytes)	%	-32768 a 32767
Long	Entero largo (4 bytes)	&	-2147483648 a 2147483647
Single	Real simple precisión (4 bytes)	!	-3.40E+38 a 3.40E+38
Double	Real doble precisión (8 bytes)	#	-1.79D+308 a 1.79D+308
Currency	Número con punto decimal fijo (8 bytes)	@	-9.22E+14 a 9.22E+14
String	Cadena de caracteres (4 bytes + 1	\$	0 a 65500 caracteres.

	byte/car hasta 64 K)		
Date	Fecha (8 bytes)		1 de enero de 100 a 31 de diciembre de 9999. Indica también la hora, desde 0:00:00 a 23:59:59.
Variant	Fecha/hora; números enteros, reales, o caracteres (16 bytes + 1 byte/car. en cadenas de caracteres)	ninguno	F/h: como Date números: mismo rango que el tipo de valor almacenado
User-defined	Cualquier tipo de dato o estructura de datos. Se crean utilizando la sentencia Type	ninguno	

En el lenguaje **Visual Basic 6.0** existen dos formas de agrupar varios valores bajo un mismo nombre. La primera de ellas son los **arrays** (vectores y matrices), que agrupan datos de tipo homogéneo. La segunda son las **estructuras**, que agrupan información heterogénea o de distinto tipo. En **Visual Basic 6.0** las estructuras son verdaderos *tipos de datos definibles por el usuario*. Para declarar las variables se utiliza la sentencia siguiente:

Dim NombreVariable As TipoVariable

cuyo empleo se muestra en los ejemplos siguientes:

Dim Radio As Double, Superficie as Single

Dim Nombre As String

Dim Etiqueta As String * 10

Dim Francos As Currency

Dim Longitud As Long, X As Currency

Es importante evitar declaraciones del tipo:

Dim i, j As Integer

pues contra lo que podría parecer a simple vista no se crean dos variables *Integer*, sino una *Integer* (j) y otra *Variant* (i).

En *Visual Basic 6.0* no es estrictamente necesario declarar todas las variables que se van a utilizar (a no ser que se elija la opción *Option Explicit* que hace obligatorio el declararlas).

2.5 OPERADORES

Tipo	Operación	Operador en Visual Basic
Aritmético	Exponenciación	^
	Cambio de signo (operador unario)	-
	Multiplicación, División	*, /
	División entera	\
	Resto de una división entera	Mod
	Suma y resta	+, -
Concatenación	Concatenar o enlazar	& +
Relacional	Igual a	=
	Distinto	<>
	Menor que / menor o igual que	< <=
	Mayor que / mayor o igual que	> >=

Otros	Comparar dos expresiones de caracteres	Like
	Comparar dos referencias a objetos	Is
Lógico	Negación	Not
	And	And
	Or inclusivo	Or
	Or exclusivo	Xor
	Equivalencia (opuesto a Xor)	Eqv
	Implicación (<i>False</i> si el primer operando es <i>True</i> y el segundo operando es <i>False</i>)	Imp

2.6 SENTENCIAS DE CONTROL

Visual Basic 6.0 dispone de las siguientes estructuras de control:

If ... Then ... Else

Select Case

For ... Next

Do ... Loop

While ... Wend

For Each ... Next

3.6.1 Sentencia IF ... THEN ... ELSE ...

Esta estructura permite ejecutar condicionalmente una o más sentencias y puede escribirse de dos formas. La primera ocupa sólo una línea y tiene la forma siguiente:

If condición **Then** sentencia1 [**Else** sentencia2]

La segunda es más general y se muestra a continuación:

If condicion **Then**

sentencia(s)

[Else

sentencia(s)]

End If

Si **condición** es **True (verdadera)**, se ejecutan las sentencias que están a continuación de **Then**, y si **condición** es **False (falsa)**, se ejecutan las sentencias que están a continuación de **Else**, si esta cláusula ha sido especificada (pues es opcional). Para indicar que se quiere ejecutar uno de varios bloques de sentencias dependientes cada uno de ellos de una condición, la estructura adecuada es la siguiente:

If condicion1 **Then**

sentencias1

Elseif condicion2 **Then**

sentencias2

Else

sentencia-n

End If

Si se cumple la **condicion1** se ejecutan las **sentencias1**, y si no se cumple, se examinan secuencialmente las condiciones siguientes hasta **Else**, ejecutándose las sentencias correspondientes al primer **Elseif** cuya condición se cumpla. Si todas las condiciones son falsas, se ejecutan las sentencias-n correspondientes a **Else**, que es la opción por defecto.

2.6.2 Sentencia SELECT CASE

Esta sentencia permite ejecutar una de entre varias acciones en función del valor de una expresión. Es una alternativa a **If ... Then ... Elseif** cuando se compara la misma expresión con diferentes valores. Su forma general es la siguiente:

Select Case expresion

Case etiq1

[sentencias1]

Case etiq2

[sentencias2]

Case Else

sentenciasn

End Select

donde **expresión** es una expresión numérica o alfanumérica, y **etiq1**, **etiq2**, ... pueden adoptar las formas siguientes:

1. *expresión*
2. *expresión To expresión*
3. *Is operador-de-relación expresion*

4. combinación de las anteriores separadas por comas

Cuando se utiliza la forma *expresión To expresión*, el valor más pequeño debe aparecer en primer lugar.

Cuando se ejecuta una sentencia **Select Case**, **Visual Basic** evalúa la **expresión** y el control del programa se transfiere a la sentencia cuya etiqueta tenga el mismo valor que la expresión evaluada, ejecutando a continuación el correspondiente bloque de sentencias. Si no existe un valor igual a la **expresión** entonces se ejecutan las sentencias a continuación de **Case Else**.

2.6.3 Sentencia FOR ... NEXT

La sentencia **For** da lugar a un lazo o bucle, y permite ejecutar un conjunto de sentencias cierto número de veces. Su forma general es:

For variable = **expresion1 To expresion2** [**Step expresion3**]

[sentencias]

Exit For

[sentencias]

Next [variable]

Cuando se ejecuta una sentencia **For**, primero se asigna el valor de la **expresion1** a la variable y se comprueba si su valor es mayor o menor que la **expresion2**. En caso de ser menor se ejecutan las sentencias, y en caso de ser mayor el control del programa salta a las líneas a continuación de **Next**. Todo esto sucede en caso de ser la **expresion3** positiva. En caso contrario se ejecutarán las sentencias cuando la variable sea mayor que **expresion2**. Una vez ejecutadas las sentencias, la variable se incrementa en el valor de la **expresion3**, o en 1 si **Step** no se especifica, volviéndose a efectuar la comparación entre la variable y la **expresion2**, y así sucesivamente.

La sentencia **Exit For** es opcional y permite salir de un bucle **For ... Next** antes de que éste finalice.

2.6.4 Sentencia DO ... LOOP

Un **Loop (bucle)** repite la ejecución de un conjunto de sentencias mientras una condición dada sea cierta, o hasta que una condición dada sea cierta. La condición puede ser verificada antes o después de ejecutarse el conjunto de sentencias. Sus posibles formas son las siguientes:

' Formato 1:

Do [{While/Until} condición]

[sentencias]

[**Exit Do**]

[sentencias]

Loop

' Formato 2:

Do

[sentencias]

[**Exit Do**]

[sentencias]

Loop [{While/Until} condición]

La sentencia opcional **Exit Do** permite salir de una bucle **Do ... Loop** antes de que finalice éste.

2.6.5 Sentencia WHILE ... WEND

Esta sentencia es otra forma de generar bucles que se recorren mientras se cumpla la condición inicial. Su estructura es la siguiente:

While condición

[sentencias]

Wend

2.6.6 Sentencia FOR EACH ... NEXT

Esta construcción es similar al bucle **For**, con la diferencia de que la variable que controla la repetición del bucle no toma valores entre un mínimo y un máximo, sino a partir de los elementos de un array (o de una colección de objetos). La forma general es la siguiente:

For Each variable **In** grupo

[sentencias]

Next variable

Con arrays *variable* tiene que ser de tipo **Variant**. Con colecciones *variable* puede ser **Variant** o una variable de tipo **Object**. Esta construcción es muy útil cuando no se sabe el número de elementos que tiene el array o la colección de objetos.

ESIISS: Aprenda Visual Basic 6.0 como si estuviera en Primero página 36

2.7 FUNCIONES Y PROCEDIMIENTOS

2.7.1 Funciones (function)

La sintaxis correspondiente a una función es la siguiente:

[**Static**] [**Private**] **Function** nombre ([parámetros]) [**As** tipo]

[sentencias]

[nombre = expresión]

[**Exit Function**]

[sentencias]

[nombre = expresión]

End Function

donde **nombre** es el nombre de la función. Será de un tipo u otro dependiendo del dato que devuelva. Para especificar el tipo se utiliza la cláusula **As Tipo** (**Integer**, **Long**, **Single**, **Double**, **Currency**, **String** o **Variant**). **parámetros** son los argumentos que son pasados cuando se llama a la función. **Visual Basic** asigna el valor de cada argumento en la llamada al parámetro que ocupa su misma posición. Si no se indica un tipo determinado los argumentos son **Variant** por defecto. Como se verá en un apartado posterior, los argumentos pueden ser pasados **por referencia** o **por valor**.

El **nombre de la función**, que es el valor de retorno, actúa como una variable dentro del cuerpo de la función. El valor de la variable **expresión** es almacenado en el propio nombre de la función. Si no se efectúa esta asignación, el resultado devuelto será 0 si la función es numérica, nulo ("") si la función es de caracteres, o *Empty* si la función es **Variant**.

Exit Function permite salir de una función antes de que ésta finalice y devolver así el control del programa a la sentencia inmediatamente a continuación de la que efectuó la llamada a la función.

La sentencia **End Function** marca el final del código de la función y, al igual que la **Exit Function**, devuelve el control del programa a la sentencia siguiente a la que efectuó la llamada, pero lógicamente una vez finalizada la función.

La **llamada a una función** se hace de diversas formas. Por ejemplo, una de las más usuales es la siguiente:

```
variable = nombre ([argumentos])
```

donde **argumentos** son un lista de constantes, variables o expresiones separadas por comas que son pasadas a la función. En principio, el número de argumentos debe ser igual al número de parámetros de la función. Los **tipos** de los argumentos deben coincidir con los tipos de sus correspondientes parámetros, de lo contrario puede haber fallos importantes en la ejecución del programa. Esta regla no rige si los argumentos se pasan **por valor** (concepto que se verá más adelante).

En cada llamada a una función hay que incluir los paréntesis, aunque ésta no tenga argumentos.

A diferencia de C y C++ en **Visual Basic 6.0** no es necesario devolver explícitamente el valor de retorno, pues el nombre de la función ya contiene el valor que se desea devolver. Tampoco es necesario declarar las funciones antes de llamarlas.

2.7.2 Procedimientos Sub

La sintaxis que define un **procedimiento Sub** es la siguiente:

```
[Static] [Private] Sub nombre [( parámetros)]  
[ sentencias]
```

[Exit Sub]

[sentencias]

End Sub

La explicación es análoga a la dada para funciones.

La llamada a un **procedimiento Sub** puede ser de alguna de las dos formas siguientes:

Call nombre[(argumentos)]

o bien, sin pasar los argumentos entre paréntesis, sino poniéndolos a continuación del nombre simplemente separados por comas:

nombre [argumentos]

A diferencia de una *función*, un **procedimiento Sub** no puede ser utilizado en una expresión pues no devuelve ningún valor. Por supuesto una función puede ser llamada al modo de un **procedimiento Sub**, pero en esta caso no se hace nada con el valor devuelto por la función.

2.7.3 Argumentos por referencia y por valor

En las *funciones* (**Function**) y en los **procedimientos Sub** de **Visual Basic**, por defecto los argumentos se pasan por **referencia**²; de este modo, cualquier cambio de valor que sufra un parámetro dentro de la *función* o del **procedimiento Sub** también se produce en el argumento correspondiente de la llamada a la *función* o al **procedimiento Sub**.

Cuando se llama a una *función* o a un **procedimiento Sub**, se podrá especificar que el valor de una argumento **no sea cambiado** por la función o por el procedimiento, poniendo dicho argumento entre paréntesis en la llamada. Un argumento entre paréntesis en la llamada es un **argumento pasado por valor**. Por ejemplo,

Raíz ((Num)) ' En el caso de la función

Raíz (Num), F ' En el caso del procedimiento

El argumento **Num** es pasado **por valor**. Significa que lo que se pasa es una copia de **Num**. Si el procedimiento cambia ese valor, el cambio afecta sólo a la copia y no a la propia variable **Num**.

Otra forma de especificar que un argumento será siempre pasado **por valor** es anteponiendo la palabra **ByVal** a la declaración del parámetro en la cabecera del procedimiento (**Sub** o **Function**).

Por ejemplo,

Function Raíz (**ByVal** N As Double)

Sub Raíz (**ByVal** N As Double, F As Double)

Pasar argumentos **por valor** evita modificaciones accidentales, pero tiene un coste en tiempo y memoria que puede ser significativo cuando se pasan grandes volúmenes de información, como sucede con vectores, matrices y estructuras.

2.7.4 Procedimientos con argumentos opcionales

Puede haber procedimientos en los que algunos de los argumentos incluidos en su definición sean opcionales, de forma que el programador pueda o no incluirlos en la llamada de dichos procedimientos. La forma de incluir un argumento opcional es incluir la palabra **Optional** antes de dicho argumento en la definición del procedimiento. Si un argumento es opcional, todos los argumentos que vienen a continuación deben también ser opcionales.

Cuando un argumento es opcional y en la llamada es omitido, el valor que se le pasa es un **Variant** con valor *Empty*. A los argumentos opcionales se les puede dar en la definición del procedimiento un valor por defecto para el caso en que sean omitidos en la llamada, como por ejemplo:

```
Private Sub miProc(x as Double, Optional n=3 As Integer)
sentencias
End Sub
```

2.7.5 Número indeterminado de argumentos

Este caso es similar pero diferente del anterior. En este caso no es que haya argumentos opcionales que puedan omitirse en la llamada, sino que realmente no se sabe con cuántos argumentos va a llamarse la función; unas veces se llamará con 2, otras con 3 y otras con 8. En este caso los argumentos se pasan al procedimiento por medio de un array, especificándolo con la palabra **ParamArray** en la definición del procedimiento, como por ejemplo:

```
Public Function maximo(ParamArray numeros())
```

```
For Each x in numeros
sentencias
maximo = x
Next x
End Function
```

2.7.6 Utilización de argumentos con nombre

Visual Basic 6.0 ofrece también la posibilidad de llamar a las **funciones** y **procedimientos Sub** de una forma más libre y menos formal, pasando los argumentos en la llamada al procedimiento con un orden arbitrario. Esto se consigue incluyendo el **nombre** de los argumentos en la llamada y asignándoles un valor por medio de una construcción del tipo **miArgumento:=unValor**. Unos argumentos se separan de otros por medio de comas (,). Considérese el siguiente ejemplo:

```
Public Sub EnviarCarta(dirección As String, destinatario As String)
```

```
sentencias
```

```
End Sub
```

que se puede llamar en la forma:

```
EnviarCarta destinatario:="Mike Tyson", dirección:="Las Vegas"
```

No todas las funciones que se pueden llamar en **Visual Basic 6.0** admiten argumentos con nombre. Con **AutoQuickInfo** puede obtenerse más información al respecto.

2.8 ARRAYS

Un **array** permite referirse a una serie de elementos del mismo tipo con un mismo nombre, y hace referencia un único elemento de la serie utilizando uno o más índices, como un vector o una matriz en Álgebra.

Visual Basic 6.0 permite definir arrays de variables de una o más dimensiones (hasta 60) y de cualquier tipo de datos (tipos fundamentales y definidos por el usuario). Pero además **Visual Basic** introduce una nueva clase de arrays, *los arrays de controles* (esto es, arrays de botones, de etiquetas, de paneles, etc.) que permiten una programación más breve y clara. En este apartado sólo se tratarán los arrays de variables.

Todos los elementos de un array deben ser del mismo tipo y están almacenados de forma contigua en la memoria. Por supuesto, si el array es de tipo **Variant** cada elemento puede contener un dato de tipo diferente, e incluso puede contener otro array.

Entre los arrays de variables cabe distinguir dos tipos fundamentales, dependiendo de que número de elementos sea constante o pueda variar durante la ejecución del programa.

1. **Arrays estáticos**, cuya dimensión es siempre la misma.
2. **Arrays dinámicos**, cuya dimensión se puede modificar durante la ejecución de programa.

2.8.1 Arrays estáticos

La declaración de un array estático dependerá de su ámbito.

La declaración de un array público se hace en la sección de declaraciones de un módulo utilizando la sentencia **Public**.

La declaración de un array a nivel del módulo o del formulario se hace en la sección de declaraciones del módulo o del formulario utilizando la sentencia **Dim** o **Private**.

Para declarar un array local a un procedimiento, se utiliza la sentencia **Dim**, **Private** o **Static** dentro del propio procedimiento.

A continuación se presentan algunos ejemplos:

Dim vector (19) As Double

Este ejemplo declara un array de una dimensión, llamado **vector**, con veinte elementos, *vector (0)*, *vector (1)*, ... , *vector (19)*, cada uno de los cuales permite almacenar un **Double**. Salvo que se indique otra cosa, los índices se empiezan a contar en cero.

Dim matriz (3, 1 To 6) As Integer

Este ejemplo declara un array de dos dimensiones, llamado **matriz**, con 4x6 elementos, *matriz (0,1)*, ... *matriz (3,6)*, de tipo entero.

Public cadena (1 To 12) As String

El ejemplo anterior declara un array de una dimensión, **cadena**, con doce elementos, *caract (1)*, ... , *caract (12)*, cada uno de los cuáles permite almacenar una cadena de caracteres.

La declaración de los arrays estáticos es bastante cómoda. Se declaran una vez. Sin embargo tienen el inconveniente que en la mayoría de los casos están sobredimensionados y utilizan más memoria de la que realmente necesitan. Esto implica que se está malgastando memoria. Para solucionar este problema se utilizan los arrays dinámicos.

2.8.2 Arrays dinámicos

El espacio necesario para un array estático se asigna al iniciarse el programa y permanece fijo durante su ejecución. El espacio para un array dinámico se asigna durante la ejecución del programa. Un array dinámico, puede ser redimensionado en cualquier momento de la ejecución. La forma mejor de redimensionar los arrays es mediante variables que contienen los valores adecuados. Para crear un array dinámico primero hay que declararlo como si fuera un array estático, pero sin darle dimensión. Es decir, se deja la lista -entre paréntesis- vacía sin ponerle ningún número. Esto se hace con la sentencia **Public** si se quiere que sea global, con **Dim** o **Private** si se quiere a nivel de módulo o con **Static**, **Dim** o **Private** si se quiere que sea local. Para asignar el número actual de elementos del array se utiliza la sentencia **ReDim**. La sentencia **ReDim** puede aparecer solamente en un procedimiento y permite cambiar el número de elementos del array y sus límites inferior y superior, pero no el número de dimensiones. Esto quiere decir que, por ejemplo, no se puede transformar un vector en una matriz.

2.9 ESTRUCTURAS: SENTENCIA TYPE

Una **estructura** (según la nomenclatura típica del lenguaje C) es un nuevo tipo de datos, un tipo definido por el usuario, que puede ser manipulado de la misma forma que los tipos predefinidos (**Int**, **Double**, **String**, ...). Una estructura puede definirse como una colección o agrupación de datos de diferentes tipos evidentemente relacionados entre sí.

Para crear una estructura con **Visual Basic 6.0** hay que utilizar la sentencia **Type ... End Type**. Esta sentencia solamente puede aparecer en la sección **General** o de declaraciones de un módulo. Pueden crearse como **Public** o como **Private** en un módulo estándar o de clase y sólo como **Private** en un formulario. **Dim** equivale a **Public**. Véase el siguiente ejemplo,

Public Type Alumno

Nombre As **String**

Direccion As **String** *40

Telefono As **Long**

DNI As **Long**

End Type

Este ejemplo declara un tipo de datos denominado **Alumno** que consta de cuatro *miembros* o campos, denominados **Nombre**, **Dirección**, **Teléfono** y **DNI**.

Una vez definido un nuevo tipo de datos, en un módulo estándar o de clase se pueden declarar variables **Public** o **Private** de ese tipo (en un formulario sólo **Private**), como por ejemplo:

```
Public Mikel as Alumno
```

Para referirse a un determinado miembro de una estructura se utiliza la notación *variable.miembro*. Por ejemplo,

```
Mikel.DNI = 34103315
```

A su vez, un miembro de una estructura puede ser otra estructura, es decir un tipo definido por el usuario. Por ejemplo,

```
Type Fecha
```

```
Dia As Integer
```

```
Mes As Integer
```

```
Anio As Integer
```

```
End Type
```

```
Type Alumno
```

```
Alta As Fecha
```

```
Nombre As String
```

```
Direccion As String * 40
```

```
Telefono AS Long
```

```
DNI As Long
```

End Type

Dentro de una estructura puede haber arrays tanto estáticos como dinámicos.

En **Visual Basic 6.0** se pueden definir **arrays de estructuras**. La declaración de un array de estructuras se hará con la palabra **Public**, **Private** o **Static**, dependiendo de su ámbito. La forma de hacerlo es la siguiente:

```
Public grupoA (1 To 100) As Alumno
```

```
Private grupoB (1 To 100) As Alumno
```

```
Static grupoC (1 To 100) As Alumno
```

En **Visual Basic 6.0**, a la hora de declarar arrays de estructuras, sucede lo mismo que al

declarar arrays de cualquier tipo de variables. Con **GrupoA (1 To 100)**, se crea un vector de estructuras de 100 elementos de tipo **Alumno** (*grupoA (1), grupoA (2), ..., grupoA (100)*). Con

grupoB (100), se crearía un vector de estructuras de 101 elementos (*grupoB (0), grupoB (1), ... grupoB (100)*).

Es posible asignar una estructura a otra del mismo tipo. En este caso se realiza una copia miembro a miembro. Véase el siguiente ejemplo:

GrupoA (1) = delegado

Las **estructuras** pueden ser también **argumentos** en las llamadas a funciones y procedimientos **Sub**. Siempre son pasados **por referencia**, lo cual implica que pueden ser modificados dentro del procedimiento y esas modificaciones permanecen en el entorno de llamada al procedimiento. En el caso de las funciones, las estructuras pueden ser también **valores de retorno**.

3. BASES DE DATOS EN VISUAL BASIC 6 (CONTROL DATA)

Definiciones:

Base de Datos relacional

Es un depósito de información. Almacena datos en tablas, que están formadas por filas y columnas. Permite recuperar o consultar subconjuntos de datos de las tablas. Permite conectar varias tablas con el fin de recuperar datos relacionados que están almacenados en tablas diferentes.

Motor de Bases de datos

Proporciona las funciones básicas de una base de datos. Es un sistema de software que administra la manera en que se almacenan y se recuperan los datos.

Los objetos de acceso a datos – Data Access Objects (**DAO**) y el control **Data** usan el motor de base de datos Microsoft Jet para acceder a bases de datos. El motor de base de datos Jet puede acceder a los siguientes 3 tipos de bases de datos:

Bases de datos Jet

Estas bases de datos son creadas y manipuladas directamente por el motor Jet. Microsoft Access y Visual Basic emplean el mismo motor de base de datos Jet.

Bases de datos de Método de Acceso Secuencial Indexado (ISAM)

Los formatos de estas bases de datos incluyen Btrieve, dBase, Microsoft Visual FoxPro, y Paradox.

Bases de datos compatibles con ODBC (Open DataBase Connectivity - Conectividad

Abierta de Base de Datos)

Estas bases de datos incluyen las bases de datos cliente/servidor que conforman el estándar ODBC, tal como Microsoft SQL Server. La mayoría de bases de datos que soportan ODBC pueden ser accedidas empleando Visual Basic.

Tablas y campos

Las bases de datos están formadas por tablas que, a su vez, están formadas por registros, y éstos, a su vez, por campos. El código de Visual Basic se utiliza para hacer referencia y manipular bases de datos, tablas, registros y campos.

Una tabla es un medio de almacenar datos que organiza la información dentro de una base de datos. Las tablas organizan la información en filas y columnas. Dentro de una tabla, una fila de datos se denomina registro, mientras que las columnas se denominan campos.

Un registro representa un elemento determinado de datos.

Un campo representa una subdivisión de los datos de un registro.

La Clave Principal

Cada tabla debe de tener una clave principal, que es un campo (o una combinación de campos) que es único para cada registro en la tabla. Una tabla puede también tener claves foráneas, que son campos que hacen referencia a una clave principal de otra tabla.

Índices

Los índices de una tabla de una base de datos son listas ordenadas que son más rápidas para las búsquedas que las tablas en sí. Para habilitar un acceso más rápido a una base de datos, la mayoría de bases de datos emplean uno o más índices.

RecordSet

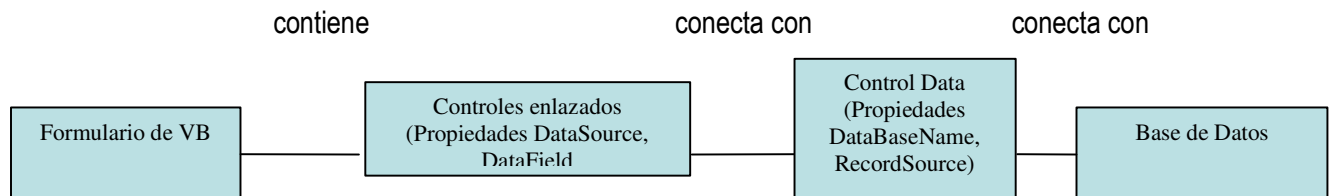
Un objeto Recordset es un conjunto de datos proporcionado por el motor de bases de datos Jet.

Tipos de objetos Recordset: Table, Dynaset, Snapshot, Forward-Only, Dynamic.

Trabajando con el Control Data

El control Data gestiona la conexión entre un formulario de Visual Basic y una Base de datos. También ofrece funciones básicas que permiten a la aplicación desplazarse por un RecordSet y agregar y actualizar registros.

Componentes implicados en la conexión de una aplicación de Visual Basic con una base de datos mediante control Data:



Uso de Controles Enlazados a Datos

Cuando un control que ha colocado en un formulario se enlaza a datos, se muestran automáticamente los datos de la base de datos en el control enlazado. Si un usuario cambia los datos de un control enlazado, dichos cambios se actualizarán automáticamente en la base de datos en cuanto el usuario se desplace a otro registro. Muchos controles intrínsecos o estándar de Visual Basic permiten ser enlazados a datos, como es el caso de los controles **CheckBox**, **Image**, **Label**, **PictureBox**, **TextBox**, **ListBox**, **ComboBox** y los **contenedores OLE**.

4. PROYECTO EN VISUAL BASIC 6: INMOBIFICH 2.45

4.1 Introducción teórica

Un proyecto es un esfuerzo temporal acometido para crear un único servicio o producto. Temporal quiere decir que todo proyecto tiene un comienzo claro y un final claro. Único significa que el producto o servicio es diferente de alguna forma clara de todos los productos o servicios similares.

Características de un proyecto:

- Existe un objetivo claro.
- Tiene principio y fin en el tiempo.
- Se requiere un nivel de calidad.
- Se puede identificar un conjunto de tareas.
- Es necesario la intervención de especialistas.
- Existen limitaciones en los recursos.
- Se requiere una planificación.

Fases de un proyecto

Se distinguen dos grandes fases:

- *Planeación:* Se distinguen dos subfases:
 - Definición del problema: ¿Cuál es el problema?, ¿Dónde está la oportunidad?
 - Planificación del proyecto: ¿Cuándo se lleva a cabo el plan previsto?, ¿Cuáles son los costes?, ¿con que recursos?
- *Ejecución:* Se distinguen tres subfases:
 - Puesta en marcha: ¿Quién hace qué?
 - Producción: Se llevan a cabo las tareas.
 - Conclusión del proyecto: Entrega del producto o servicio

4.2 Descripción del proyecto

Almenar inmobiliaria es una pyme que trabaja en el sector inmobiliario .Los comerciales realizan visitas programadas a los inmuebles para anotar toda la información de interés tanto de los inmuebles como de los clientes que quieren vender o alquilar dicho inmueble . Dicha información es escrita in situ en unas fichas de cartulina. Las fichas tienen 2 caras y proporcionan una descripción detallada útil para la inmobiliaria. Paralelamente a la escritura en las fichas los comerciales realizan fotos del inmueble y opcionalmente graban imágenes en video.

El problema surge ante la gran cantidad de fichas que tenían que manejar y el consecuente inconveniente de las búsquedas en esas fichas.

Solicitaron una aplicación visual y fácil de utilizar que resolviera el problema de almacenamiento y manipulación de las fichas.

Después de un estudio previo llegué a la conclusión que la forma más sencilla, dado el pequeño número de PC's que poseía la empresa, de afrontar el problema era programar la aplicación en un lenguaje visual altamente compatible e integrado en el sistema Windows (Visual Basic 6.0). Del mismo modo la base de datos se especificó para un fácil mantenimiento por tanto la solución más adecuada era Access 2000. Se especificó la necesidad de una solución a la búsqueda de fichas en la base de datos, esa búsqueda tendría que ser no muy compleja. La impresión de las fichas tenía que estar resuelta mediante un sencillo interfaz del sistema.

Por tanto, inicialmente la aplicación estaba constituida por dos formularios de introducción de datos a lo que se añadiría un formulario de búsqueda sencilla en la base de datos y una opción de impresión de fichas.

Posteriormente se introdujo, a sugerencia mía, un formulario que resolviera el problema de los cálculos complejos (calculadora hecha íntegramente por mí) y un visor rápido de fotografías.

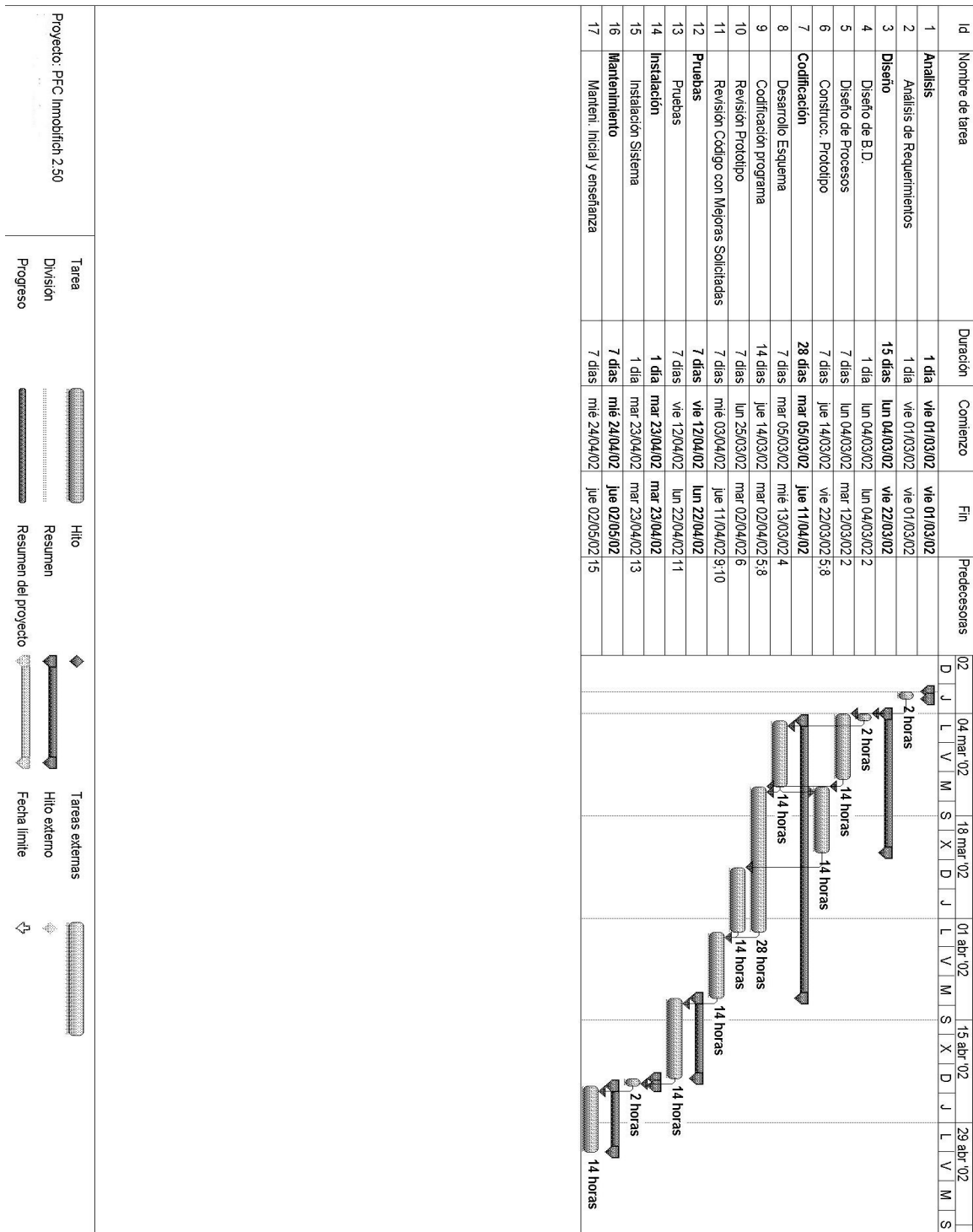
Una vez programado con eficiencia y corrección la aplicación se la dotó de un aspecto gráfico adecuado para una posterior visualización por los trabajadores y por los clientes de la empresa

El funcionamiento por parte de los comerciales es el siguiente: en un primer momento anotan todas las características de los inmuebles en unas fichas (fichas con dos caras) y realizan fotos y videos, y a continuación en las oficinas insertan las fotos y toda la información anotada en las fichas escritas en la base de datos Inmobiliaria.mdb a través de la interfaz gráfica Inmobifich 2.45.

4.3 Planificación del proyecto

Duración sin contar Mantenimiento: 52 días

Total de horas empleadas sin contar Mantenimiento: 118 horas



4.3.1 Análisis de requerimientos

El análisis de requerimientos consistió en concertar una cita en las oficinas para detallar:

- 1- La especificación de la aplicación: plasmar en un programa las fichas de cartulina que utilizaban por las dos caras.
- 2- Como debería de funcionar en el entorno Windows: menús, barra de herramientas y opciones fácilmente seleccionables mediante dispositivo señalizador (ratón).
- 3- Cuáles deberían de ser los requerimientos tanto de software (Base de datos Access sobre Sistema Operativo Windows XP) como de hardware (PC's compatibles IBM Intel Pentium).

4.3.2 Diseño

4.3.2.1 Diseño de B.D. (Bases de Datos)

El diseño de la Base de Datos fue una tarea relativamente sencilla ya que del análisis de requerimientos surgieron dos entidades del modelo entidad-relación que correspondían a cada una de las caras de las fichas de cartulina (cara delantera y cara trasera de la ficha de cartulina)

Esas dos entidades del modelo entidad-relación (cara delantera y cara trasera de la ficha de cartulina) las relacioné mediante una relación uno a uno.

Esa relación uno a uno de las entidades del modelo entidad-relación se concretó en la etapa de implementación mediante las claves primarias de cada una de las dos tablas que surgieron del modelo entidad-relación.

En la etapa de implementación las claves primarias se especificaron como incrementales. Eso significa que tenían un valor único con límite muy grande (entero largo).

4.3.2.2 Diseño de procesos

No existen procesos con carga de CPU elevada. Solo se puede considerar un proceso con carga relativa de CPU y es el proceso de búsqueda de fichas por la Base de Datos.

La búsqueda de fichas está implementada solo mediante sentencias Select en modo iterativo.

4.3.2.3 Construcción de Prototipo

Construí un prototipo completamente funcional en la capa de Presentación.

La interfaz gráfica era rudimentaria pero permitía mostrar con toda claridad todas las funcionalidades que presentaría la aplicación posteriormente.

En esta etapa surgieron algunas modificaciones por parte del cliente en se reflejaron en la versión final del producto (Inmobifich 2.45).

4.3.3 Codificación

La codificación de la aplicación la realicé atendiendo al modelo incremental:

- 1- Secuencia de aplicaciones de ciclo clásico
- 2- Cada iteración produce un incremento del producto (Inmobifich)
- 3- Finaliza cuando se entrega el producto final (Inmobifich 2.45)

En cada iteración realicé una funcionalidad completa (calculadora, visor de fotos, búsqueda, descripción, inmobiliaria y sobre).

4.3.4 Pruebas

Las pruebas se realizaban en cada ciclo clásico.

4.3.5 Instalación

La instalación de la aplicación Inmobifich 2.45 la realicé sin problemas y empezando por el PC que hacía de servidor. A continuación instalé la aplicación en cada uno de los 3 ordenadores cliente.

Comprobé que había comunicación con el servidor:

- 1- Revisé los dispositivos de red (Hub, Switch, Router) y el cableado (cables UTP categoría 5)
- 2- Comprobé que estaba correctamente instalado el protocolo de red TCP/IP
- 3- Comprobé que todos los ordenadores estaban en red.
- 4- Finalmente hice pruebas con la aplicación para comprobar in situ que todo funcionaba correctamente. Para ello di de alta una nueva ficha, modifiqué dicha ficha, guardé los datos, posteriormente recupere dicha ficha y para finalizar la borré definitivamente de la Base de datos.

4.3.6 Mantenimiento

Durante la etapa de mantenimiento me dediqué a enseñar el funcionamiento de la aplicación a los usuarios de la aplicación. No tuve que realizar ninguna modificación en la aplicación, funcionaba correctamente y estaba muy optimizada.

Pueden perfectamente acceder a la aplicación en red hasta 6 usuarios además del servidor (en total 7 usuarios accediendo concurrentemente).

4.4 Problema que resuelve

El software Inmobifich 2.45 resuelve el problema del almacenamiento y la manipulación de gran cantidad de información proporcionada por los comerciales.

La aplicación fue concebida para dar respuesta a una necesidad de la empresa de llevar una gestión computarizada de sus recursos. Cuando hablamos de recursos nos referimos a todos los inmuebles que una inmobiliaria vende o alquila

4.5 Ventajas que aporta la aplicación

La aplicación se programó originalmente como Cliente/Servidor .Un PC, que tiene también la aplicación instalada , actúa de servidor donde estaba alojada la base de datos en Access 2000, y 3 PC's clientes, con la aplicación instalada, hacían peticiones al servidor. Para la realización de este proyecto, la aplicación se ha modificado para que solo la máquina local donde se instala la aplicación acceda a la base de datos en Access 2000, que a su vez está almacenada en el disco duro local.

Algunas de las ventajas son:

- Una interfaz gráfica de usuario consistente y fácil de utilizar reduce el tiempo de aprendizaje de la aplicación.
- Permiten un mejor aprovechamiento de los sistemas existentes (Windows XP, Office 2000), protegiendo la inversión. La base de datos se puede instalar en cualquier PC.
- La existencia de varios ordenadores Cliente proporciona una red más fiable: un fallo en uno de los equipos no significa necesariamente que el sistema deje de funcionar.

4.6 Componentes software de la aplicación

El proyecto consta de una aplicación informática de gestión cuya interfaz de usuario está desarrollada en Visual Basic 6.0 y que su vez enlaza con una base de datos hecha en ACCESS 2000 a través de un Control Data. Tanto la base de datos como el archivo ejecutable se encuentran alojados en el directorio de trabajo (c/archivos de programa/Inmobifich) una vez realizada la instalación automatizada (para instalaciones en S.O's Windows Xp SP3 o superiores se recomienda no sobrescribir las librerías durante la instalación).

Los archivos que forman el proyecto son los siguientes:

- El archivo ejecutable -> Inmobifich.exe
- La base de datos en Access 2000 -> Inmobiliaria.mdb
- Librerías dinámicas dll-> vbrun60 servicepack 6
- Controles-> Controles Ocx, ActiveX

El conjunto de todos estos ficheros se cuentan encapsulados en un archivo de instalación automatizado llamado SETUP.exe.

4.7 Tecnologías empleadas

Entorno de definición de datos: ACCESS 2000

Desarrollo de la interfaz: VISUAL BASIC 6.0

Modelo de objetos de acceso a datos: DAO/JET

Motor de Base de Datos: JET

4.8 Esquema de la base de datos

Hay 2 tablas, **Descripción e Inmueble**.

Las dos tablas están relacionadas mediante el campo **IdInmueble** (Clave primaria, tipo autonómico).

La tabla **Descripción** contiene una descripción detallada del inmueble con todas sus características y permite también almacenar fotos (y cualquier información multimedia que pueda ser abierta por una aplicación del sistema – ejemplo windows media player, Paint etc...) del inmueble. **La entrada de datos es el formulario FORMMASDATOS**

La tabla **Inmueble** contiene información útil para el comercial y para los trabajadores de la empresa. Dicha información no debe conocerla el cliente de la inmobiliaria. Es básicamente información de gestión interna de la propia inmobiliaria. **La entrada de datos es el formulario FORMINMOBILIARIA**

TABLA DESCRIPCIÓN

NOMBRE DEL CAMPO	TIPO DE DATOS	DESCRIPCIÓN
IdInmueble	Autonómico	Clave primaria
S_util	Número	Superficie útil
S_const	Número	Superficie construida
S_parcela	Número	Superficie parcela
Baños	Texto	Baños
Aseos	Texto	Aseos
Cocina	Texto	Cocina
Euros_m2	Número	Euros el m2

Pts_m2	Número	Pesetas el m2
Gtos_euros	Número	Gastos en euros
Gtos	Número	Gastos
Agua	Sí/No	¿Tiene agua?
Luz	Sí/No	¿Tiene luz?
Gas	Sí/No	¿Tiene gas?
Teléfono	Sí/No	¿Tiene Tlf?
Nuevo	Sí/No	¿Inmueble nuevo?
Amueblado	Sí/No	¿Está amueblado?
Modular	Sí/No	¿Es modular?
Rehabilitada	Sí/No	¿Esta rehabilitado?
Aire_Acond	Sí/No	¿Tiene aire acondicionado?
Galería	Sí/No	¿Tiene galería?
Exterior	Sí/No	¿Es exterior?
Diáfano	Sí/No	¿Es diáfano?
Pta_Blindada	Sí/No	¿Tiene puerta blindada?
Internet	Sí/No	¿Tiene conexión de Internet?
Nota_Encargo	Sí/No	¿Tiene nota de encargo?
Ascensor	Texto	Descripción ascensor
Garaje	Texto	Descripción garaje
Trastero	Texto	Descripción trastero
Calefacción	Texto	Descripción calefacción
Agua_Cal	Texto	Descripción agua caliente

Terraza	Texto	Descripción terraza
Armarios	Texto	Descripción armarios
Balcón	Texto	Descripción balcón
Piscina	Texto	Descripción piscina
Escaparate	Texto	Descripción escaparate
N_Puertas_fach	Número	Nº de puertas en fachada
M_Vestibulo	Número	Nº vestíbulo
M_Eles_com	Número	Nº elementos com.
M_Sal_com	Número	Nº salas/comedor
Años_finca	Número	Antigüedad finca
M_Vestidor	Número	M. vestidor
N_Alturas	Número	Nº de alturas
Permuta_por	Texto	Permuta por
Comunicado	Texto	Comunicado
Portero	Sí/No	¿Tiene portero?
Orientación	Texto	Orientación
Sauna	Sí/No	¿Tiene sauna?
Fecha1	Fecha/Hora	Fecha 1
Fecha2	Fecha/Hora	Fecha 2
Fecha3	Fecha/Hora	Fecha 3
Fecha4	Fecha/Hora	Fecha 4
Otros1	Texto	Otras características 1
Otros2	Texto	Otras características 2

Otros3	Texto	Otras características 3
Otros4	Texto	Otras características 4
Otros5	Texto	Otras características 5
Otros6	Texto	Otras características 6
Observaciones	Texto	Observaciones
Foto1	Objeto OLE	Foto inmueble 1
Foto2	Objeto OLE	Foto inmueble 2
Foto3	Objeto OLE	Foto inmueble 3
Foto4	Objeto OLE	Foto inmueble 4
Foto5	Objeto OLE	Foto inmueble 5
Foto6	Objeto OLE	Foto inmueble 6

TABLA INMUEBLE

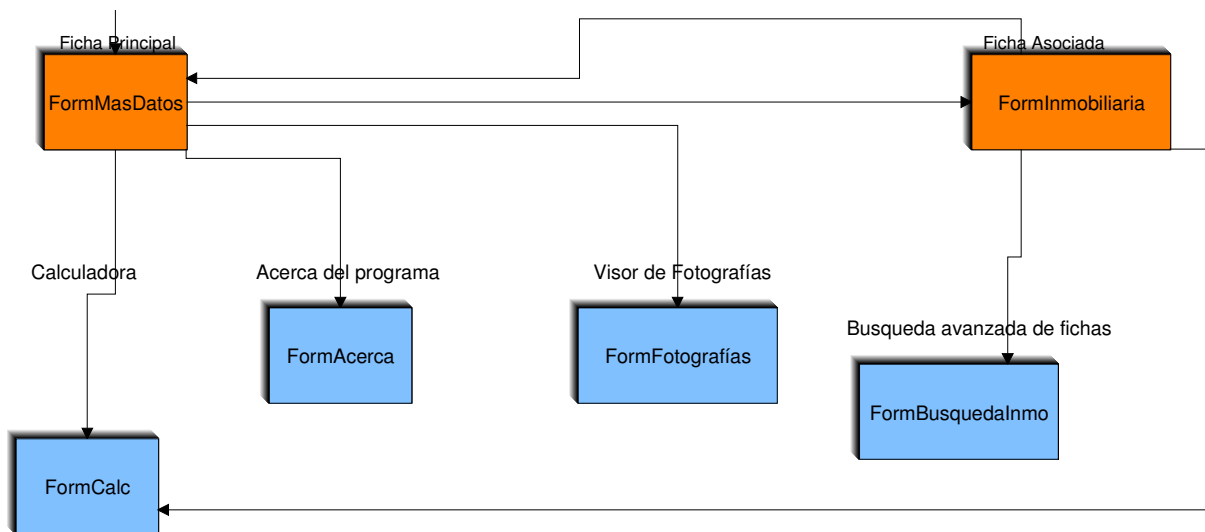
NOMBRE DEL CAMPO	TIPO DE DATOS	DESCRIPCIÓN
IdInmueble	Autonumérico	Clave primaria
Mi_Id	Número	Identificado Inmueble
Modo	Texto	Modo
PrecioPts	Número	Precio en pesetas
PrecioEuros	Número	Precio en euros
PrecioGaraje	Número	Precio de garaje
NBA	Número	Nº de habitaciones
Propietario	Texto	Nombre del Propietario
Tipo	Texto	Tipo

Zona	Texto	Zona donde se encuentra el inmueble
Edificio	Texto	Edificio
Dirección	Texto	Dirección
Número	Número	Número
Piso	Número	Piso
Puerta	Texto	Puerta
Población	Texto	Población
CP	Número	Código postal
Provincia	Texto	Provincia
País	Texto	País
Excl.	Texto	Excl.
Exp	Texto	Expediente
Obs_Gastos	Texto	Observación gastos
Fecha Capt	Fecha/Hora	Fecha captación
Valor_Suelo	Número	Valor del suelo
Valor Urbano	Número	Valor urbano
G_Comunidad	Número	Gastos comunidad
Finca_Registral	Texto	Finca registral
N_Registral	Número	Numero Registral
Llaves	Texto	Llaves
Contacto	Texto	Persona de contacto
Telefonos1	Texto	Nº de teléfono 1
Telefonos2	Texto	Nº de teléfono 2

Tasación	Número	Valor Tasación
Captador	Texto	Captador
Fax	Texto	Nº de fax
N_Finca	Número	Nº de finca
Libro	Texto	Libro
Com_Cliente	Número	Comisión cliente
Com_Propietario	Número	Comisión Propietario
Com_Captador	Número	Comisión Captador
Observaciones	Texto	Observaciones

5. DOCUMENTACIÓN DEL PROYECTO

5.1 Descripción de los formularios



Formulario FORMMASDATOS: Formulario de entrada de datos de información detallada del inmueble en cuestión (incluida las fotos). Desde este formulario se llama a FORMCALC, FORMINMOBILIARIA, FORMCERCA y FORMFOTOGRAFIAS

Formulario FORMINMOBILIARIA: Formulario de entrada de datos de información de gestión y control para la empresa inmobiliaria. Desde este formulario se llama a FORMCALC, FORMMASDATOS, FORMBUSQUEDAINMO y FORMCALC

Formulario FORMBUSQUEDAINMO: Permite realizar búsqueda de fichas mediante la introducción de parámetros de búsqueda.

Formulario FORMCALC: Permite realizar cálculos aritméticos básicos y complejos (sumas, restas, conversión de euros etc...). Esta calculadora ha sido desarrollada por mí completamente con el objetivo de integrarla en este proyecto. Además de las funciones habituales incluye una ventana con el número en memoria (es original e inédito).

Formulario FORMFOTOGRAFIAS: Permite visualizar fotografías en varios formatos (GIF, JPG etc...) almacenadas en el PC.

Formulario FORMACERCA: Contiene información sobre mí, la aplicación y el sistema.

5.2 Pantallazos de los formularios

Formulario FORMMASDATOS

Descripción de los botones de la barra de herramientas:

1. Editar ficha: Inicialmente cuando se muestran los formularios principales (FormMasDatos y Form Inmobiliaria) no se pueden modificar, es necesario pulsar el botón editar ficha para poder modificar los formularios -> ficha de datos e inmobiliaria
2. Nueva ficha: Crea una nueva ficha de datos (FormMasDatos).
3. Calcular: Se muestra el formulario de la calculadora (FormCalc)
4. Eliminar Ficha: Se elimina la ficha actual de datos y su ficha asociada de inmobiliaria de la base de datos
5. Imprimir Ficha: Se imprime la ficha que se muestra en pantalla actualmente.
6. Salir: Finaliza la aplicación en curso.
7. Acerca de: Información del creador de la aplicación o sea yo, Miguel Monzó Pastor
8. Fotos: Se muestra el formulario de visualización de fotos (FormFotografias)
9. Ficha asociada: Se muestra el formulario FormInmobiliaria -> La ficha asociada (ficha de inmobiliaria)

Formulario FORMINMOBILIARIA

Descripción de los botones de la barra de herramientas:

1. Editar ficha: Inicialmente cuando se muestran los formularios principales (FormMasDatos y Form Inmobiliaria) no se pueden modificar, es necesario pulsar el botón editar ficha para poder modificar los formularios -> ficha de datos e inmobiliaria
2. Buscar Fichas: Realiza una búsqueda de fichas por la base de datos.
3. Imprimir Ficha: Se imprime la ficha que se muestra en pantalla actualmente.
4. Calcular: Se muestra el formulario de la calculadora (FormCalc)
5. Salir: Finaliza la aplicación en curso.
6. Ficha principal: Se muestra el formulario FormMasDatos -> La ficha principal (ficha de datos)

Formulario FORMBUSQUEDAINMO

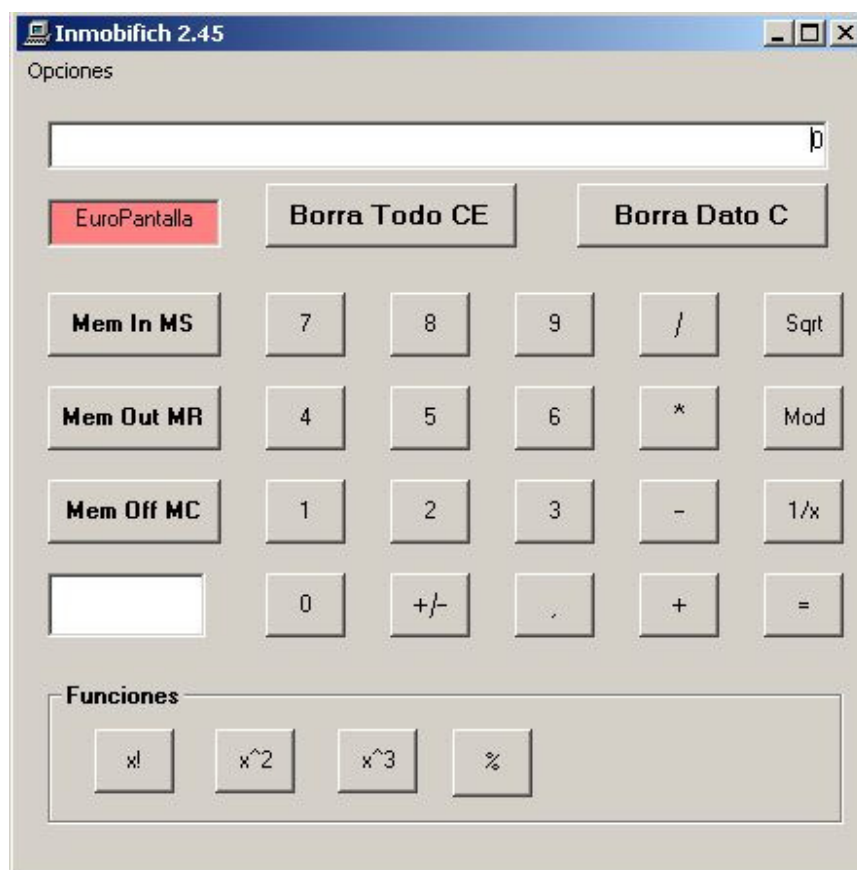
El formulario FormBusquedaInmo realiza una búsqueda de fichas por la base de datos atendiendo a diversos criterios (para habilitar la búsqueda hay que seleccionarla mediante el recuadro Si/No que está a la izquierda del criterio de búsqueda. Ejem.: ID INM):

- Por identificador de inmueble (ID INM)
- Por otros criterios:
 - Lugar:
 - Zona
 - Dirección
 - Características:
 - Nº de habitaciones
 - Precio:
 - Precio Mínimo
 - Precio Máximo

Descripción de los botones:

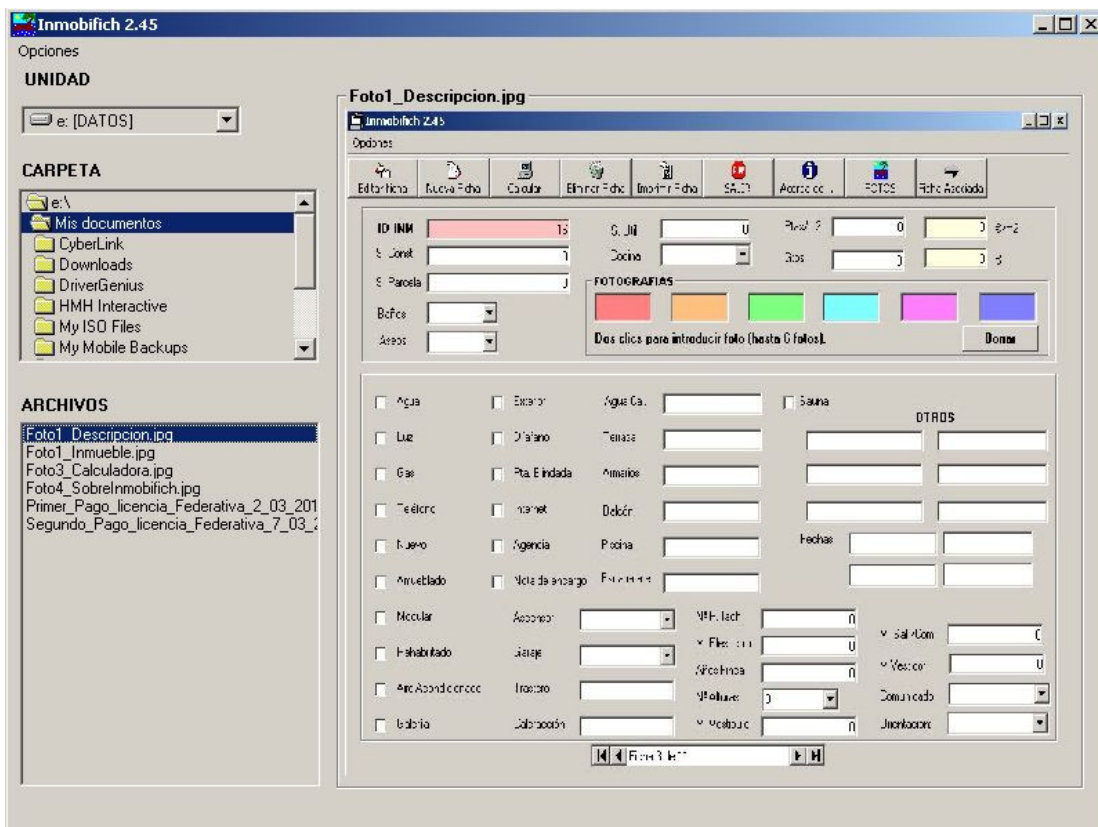
1. Buscar: Realiza una búsqueda de fichas que coincidan con los criterios seleccionados. Hay que comentar que o bien se puede realizar una búsqueda por identificador o bien por el resto de criterios, no es posible realizar una búsqueda con los dos criterios habilitados (identificador y otros criterios - cualquiera de ellos -).
2. Cerrar: Cierra el formulario
3. Borrar todo: Deselecciona todos los criterios y borra las ventanas.

Formulario FORMCALC



Calculadora científica con ventana para mostrar el número almacenado en memoria y conversor de euros a pesetas y viceversa (EuroPantalla). Para habilitar una u otra conversión hay que hacer click sobre la EuroPantalla hasta habilitar la conversión deseada.

Formulario FORMFOTOGRAFIAS



Visor de fotografías básico con adaptación de la foto al marco (redimensionado). Seleccionamos primero la unidad, a continuación la carpeta y finalmente el archivo (imagen) que queremos mostrar en el visor.

Formulario FORMACERCA



Información de mí y del sistema

5.3 Código significativo

FORMINMOBILIARIA

Option Explicit

'Private cerrojo As Boolean

Private contador As Integer 'Numero de registros que se han encontrado

'Private reg_ant As String

'El evento Reposition ocurre cuando cambia el registro activo

Private Sub DataInmobiliaria_Reposition()

 On Error Resume Next

 'Elimino los espacios delante y detrás y paso a mayúsculas

 ComboZona.Text = UCase(Trim(ComboZona.Text))

 TextTipo.Text = UCase(Trim(TextTipo.Text))

 TextEdificio.Text = UCase(Trim(TextEdificio.Text))

 TextDireccion.Text = UCase(Trim(TextDireccion.Text))

 TextPoblacion.Text = UCase(Trim(TextPoblacion.Text))

 TextPaís.Text = UCase(Trim(TextPaís.Text))

 TextPropietario.Text = UCase(Trim(TextPropietario.Text))

 TextPuerta.Text = UCase(Trim(TextPuerta.Text))

 TextObsGastos.Text = UCase(Trim(TextObsGastos.Text))

 TextContacto.Text = UCase(Trim(TextContacto.Text))

 TextLlaves.Text = UCase(Trim(TextLlaves.Text))

 TextCaptador.Text = UCase(Trim(TextCaptador.Text))

 TextLibro.Text = UCase(Trim(TextLibro.Text))

 TextObservaciones.Text = UCase(Trim(TextObservaciones.Text))

 'Deshabilito los frames

 FrameDatosProp1.Enabled = False

 FrameDatosProp2.Enabled = False

```
Frameld.Enabled = False
```

```
DataInmobiliaria.Caption = "Ficha " & DataInmobiliaria.Recordset.AbsolutePosition + 1 & " de " & _  
DataInmobiliaria.Recordset.RecordCount  
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
Private Sub Form_Activate()  
On Error Resume Next
```

```
DataInmobiliaria.Connect = "Access 2000;"  
DataInmobiliaria.DatabaseName = App.Path & "\Inmobiliaria.mdb"  
DataInmobiliaria.DefaultCursorType = 0 'Defaultcursor  
DataInmobiliaria.DefaultType = 2 'Usejet  
DataInmobiliaria.RecordsetType = 1 'Dynaset  
DataInmobiliaria.RecordSource = "Inmueble"  
DataInmobiliaria.Refresh
```

```
'Implementación de la búsqueda. Se puede buscar por Identificador o  
' por el resto de campos de búsqueda
```

```
'cerrojo = True  
If Busqueda_activada Then 'Si se quiere realizar una búsqueda  
If Buscar_por_ID = True Then  
FormBusquedaInmo.TextBuscarID.Text = UCase(Trim(FormBusquedaInmo.TextBuscarID.Text))  
If ValidarNumero(FormBusquedaInmo.TextBuscarID.Text) > 0 Then  
Exit Sub  
End If
```



```

DataInmobiliaria.Recordset.FindFirst "[IdInmueble] LIKE '" & FormBusquedaInmo.TextBuscarID.Text
& ""
If DataInmobiliaria.Recordset.NoMatch = True Then
    MsgBox "No se han encontrado fichas con ID INM: " & FormBusquedaInmo.TextBuscarID.Text
End If
Else
    Select Case FormBusquedaInmo.CheckBuscarZona.Value
    Case 1 ' Esta seleccionada la opcion de zona
        Select Case FormBusquedaInmo.CheckBuscarDireccion.Value
        Case 0: ' No esta seleccionada la opcion Direccion
            FormBusquedaInmo.TextBuscarZona.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarZona.Text))
            Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value
            Case 1 'Esta seleccionado el N° de habitaciones
                FormBusquedaInmo.TextBuscarHabitaciones.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))
                Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

                Case 1 'OptionBuscarPMinimo.Value = 1
                    FormBusquedaInmo.TextBuscarPMinimo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
                    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
                    Case 1: 'OptionBuscarPMaximo.Value = 1
                        FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
                        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "" _
                        & "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
                        & "AND" & "[PrecioEuros] >= " & _
                        Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
<= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

                    Do
                        If Not (DataInmobiliaria.Recordset.NoMatch) Then

```

```

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
                & "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
                Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
"[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
        "Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
            & "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
        & "AND" & "[PrecioEuros] >= " & _
            Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                contador = contador + 1
                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then

```

```

        DataInmobiliaria.Recordset.FindNext "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
        & "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
        Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
    Else
        Exit Do
    End If
Else
    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
            FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
            DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
            & "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
& "AND" & "[PrecioEuros] <= " & _
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

```

```

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
        DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND" & "[NHab] = '" &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] <= '" & _
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If

    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0

    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND" & "[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                contador = contador + 1
                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                    DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _

```

```

      & "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text
      Else
      Exit Do
      End If
      Else
      DataInmobiliaria.Recordset.MoveFirst
      Exit Do
      End If
      Loop While Not (DataInmobiliaria.Recordset.EOF)

      MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
      contador = 0
      End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value'

Case 0 'No esta seleccionado el N° de habitaciones
  Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

    Case 1 'OptionBuscarPMinimo.Value = 1
      FormBusquedaInmo.TextBuscarPMinimo.Text =
      UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
          FormBusquedaInmo.TextBuscarPMaximo.Text =
          UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
          DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
          & "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
          "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

```

```

Do
  If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1

    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then

      DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
        & "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
        "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

      Else

        Exit Do

      End If

      Else

        DataInmobiliaria.Recordset.MoveFirst

        Exit Do

      End If

      Loop While Not (DataInmobiliaria.Recordset.EOF)
      MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
      contador = 0

      Case 0: 'OptionBuscarPMaximo.Value = 0

        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
          & "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

        Do

          If Not (DataInmobiliaria.Recordset.NoMatch) Then

```

```

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' & _
                "AND" & "[PrecioEuros] >= '" &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
            FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
            DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' & _
                & "AND" & "[PrecioEuros] <= '" &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

        Do

```

```

    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1

        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then

            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
                & "AND" & "[PrecioEuros] <= '" &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

            Else

                Exit Do

            End If

        Else

            DataInmobiliaria.Recordset.MoveFirst

            Exit Do

        End If

        Loop While Not (DataInmobiliaria.Recordset.EOF)

        MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador

        contador = 0

        Case 0: 'OptionBuscarPMaximo.Value = 0

            DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "'

    Do

        If Not (DataInmobiliaria.Recordset.NoMatch) Then

            contador = contador + 1

            If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then

                DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "'

```



```

Else
  Exit Do
End If
Else
  DataInmobiliaria.Recordset.MoveFirst
  Exit Do
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value
End Select 'FormBusquedaInmo.CheckBuscarHabitaciones.Value

Case 1: 'Esta seleccionada la opcion Direccion
  FormBusquedaInmo.TextBuscarDireccion.Text =
  UCase(Trim(FormBusquedaInmo.TextBuscarDireccion.Text))
  Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value
    Case 1 'Esta seleccionado el Nº de habitaciones
      FormBusquedaInmo.TextBuscarHabitaciones.Text =
      UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

        Case 1 'OptionBuscarPMinimo.Value = 1
          FormBusquedaInmo.TextBuscarPMinimo.Text =
          UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
          Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
            Case 1: 'OptionBuscarPMaximo.Value = 1
              FormBusquedaInmo.TextBuscarPMaximo.Text =
              UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))

```

```

        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
        & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "' & _
        "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
        Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
<= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
            "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
& "AND" & "[PrecioEuros] >= " & _
            Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
"[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

```

```

Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "'" _
    & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "'" & _
    "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
    Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "'" _
            & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
            "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
            Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

```

```
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
```

```
Case 0 'OptionBuscarPMinimo.Value = 0
```

```
Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
```

```
Case 1: 'OptionBuscarPMaximo.Value = 1
```

```
FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
& "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "' & "AND" & _
"[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" &
"[PrecioEuros] <= '" & _
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
If Not (DataInmobiliaria.Recordset.NoMatch) Then

contador = contador + 1
If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
& "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & "AND" & _
"[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND"
& "[PrecioEuros] <= '" & _
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
Else
Exit Do
End If
Else
DataInmobiliaria.Recordset.MoveFirst
```

```

Exit Do
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
& "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "' & _
"AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text

Do
If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
        DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
& "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
"AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text
    Else
        Exit Do
    End If
Else
    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

```

```

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
  End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value'

Case 0 'No esta seleccionado el N° de habitaciones
  Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

    Case 1 'OptionBuscarPMinimo.Value = 1
      FormBusquedaInmo.TextBuscarPMinimo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
          FormBusquedaInmo.TextBuscarPMaximo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
          DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
    FormBusquedaInmo.TextBuscarZona.Text & "' _
          & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
    & "' _
          "AND" & "[PrecioEuros] >= " &
    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
          "AND" & "[PrecioEuros] <= " &
    Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

        Do
          If Not (DataInmobiliaria.Recordset.NoMatch) Then

            contador = contador + 1

            If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
    vbOKCancel + vbInformation, "Busqueda") = vbOK Then
              DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
    FormBusquedaInmo.TextBuscarZona.Text & "' _

```

```

    & "AND [Direccion] LIKE " &
FormBusquedaInmo.TextBuscarDireccion.Text & "" & _
    "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
    "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
    Else
    Exit Do
    End If
    Else
    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
    End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
    & "AND [Direccion] LIKE " & FormBusquedaInmo.TextBuscarDireccion.Text
& "" & _
    "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

    Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _

```

```

      & "AND [Direccion] LIKE " &
FormBusquedaInmo.TextBuscarDireccion.Text & "" & _
      "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
      Else
      Exit Do
      End If
      Else
      DataInmobiliaria.Recordset.MoveFirst
      Exit Do
      End If
      Loop While Not (DataInmobiliaria.Recordset.EOF)

      MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
      contador = 0
      End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

      Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
      Case 1: 'OptionBuscarPMaximo.Value = 1
      FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
      DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" & _
      & "AND [Direccion] LIKE " & FormBusquedaInmo.TextBuscarDireccion.Text
      & "" & _
      "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

      Do
      If Not (DataInmobiliaria.Recordset.NoMatch) Then

```



```

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' _
            "AND" & "[PrecioEuros] <=" &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
        "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text &
''''

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then

```

```
        DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "'" & _
        "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'"
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value
End Select 'FormBusquedaInmo.CheckBuscarHabitaciones.Value

End Select 'FormBusquedaInmo.CheckBuscarDireccion.Value
```

Case 0 ' No Esta seleccionada la opcion zona

Select Case FormBusquedaInmo.CheckBuscarDireccion.Value

Case 0: ' No esta seleccionada la opcion Direccion

Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value

Case 1 'Esta seleccionado el N° de habitaciones

FormBusquedaInmo.TextBuscarHabitaciones.Text =

UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))

Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

Case 1 'OptionBuscarPMinimo.Value = 1

FormBusquedaInmo.TextBuscarPMinimo.Text =

UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))

Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 1: 'OptionBuscarPMaximo.Value = 1

FormBusquedaInmo.TextBuscarPMaximo.Text =

UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))

DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
 FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _

Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
 <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do

If Not (DataInmobiliaria.Recordset.NoMatch) Then

contador = contador + 1

If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
 vbOKCancel + vbInformation, "Busqueda") = vbOK Then

DataInmobiliaria.Recordset.FindNext "[NHab] = " &
 FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _

```

      Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
"[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
      Else
      Exit Do
      End If
      Else
      DataInmobiliaria.Recordset.MoveFirst
      Exit Do
      End If
      Loop While Not (DataInmobiliaria.Recordset.EOF)
      MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
      contador = 0
      Case 0: 'OptionBuscarPMaximo.Value = 0
      DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
      Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

      Do
      If Not (DataInmobiliaria.Recordset.NoMatch) Then

      contador = contador + 1
      If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
      DataInmobiliaria.Recordset.FindNext "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
      Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
      Else
      Exit Do
      End If
      Else
      DataInmobiliaria.Recordset.MoveFirst
      Exit Do

```

```
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
Case 1: 'OptionBuscarPMaximo.Value = 1
FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] <= " & _
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
If Not (DataInmobiliaria.Recordset.NoMatch) Then

contador = contador + 1
If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
DataInmobiliaria.Recordset.FindNext "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] <= " & _
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
Else
Exit Do
End If
Else
DataInmobiliaria.Recordset.MoveFirst
Exit Do
```

```

    End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                contador = contador + 1
                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                    DataInmobiliaria.Recordset.FindNext "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text
                Else
                    Exit Do
                End If
            Else
                DataInmobiliaria.Recordset.MoveFirst
                Exit Do
            End If
        Loop While Not (DataInmobiliaria.Recordset.EOF)

        MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
        "Numero de fichas encontradas: " & contador
        contador = 0

    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value'

```

Case 0 'No esta seleccionado el N° de habitaciones

```
Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value
```

```
Case 1 'OptionBuscarPMinimo.Value = 1
```

```
FormBusquedaInmo.TextBuscarPMinimo.Text =  
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
```

```
Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
```

```
Case 1: 'OptionBuscarPMaximo.Value = 1
```

```
FormBusquedaInmo.TextBuscarPMaximo.Text =  
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
```

```
DataInmobiliaria.Recordset.FindFirst "[PrecioEuros] >= " &  
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
```

```
"AND" & "[PrecioEuros] <= " &  
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
```

```
Do
```

```
If Not (DataInmobiliaria.Recordset.NoMatch) Then
```

```
contador = contador + 1
```

```
If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",  
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
```

```
DataInmobiliaria.Recordset.FindNext "[PrecioEuros] >= " &  
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
```

```
"AND" & "[PrecioEuros] <= " &  
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
```

```
Else
```

```
Exit Do
```

```
End If
```

```
Else
```

```
DataInmobiliaria.Recordset.MoveFirst
```

```
Exit Do
```

```
End If
```

```
Loop While Not (DataInmobiliaria.Recordset.EOF)
```

```
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
```

```
"Numero de fichas encontradas: " & contador
```

```

    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                contador = contador + 1
                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                    DataInmobiliaria.Recordset.FindNext "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
                Else
                    Exit Do
                End If
            Else
                DataInmobiliaria.Recordset.MoveFirst
                Exit Do
            End If
        Loop While Not (DataInmobiliaria.Recordset.EOF)

        MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
        contador = 0
    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

    Case 0 'OptionBuscarPMinimo.Value = 0

        Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
            Case 1: 'OptionBuscarPMaximo.Value = 1
                FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))

```



```

DataInmobiliaria.Recordset.FindFirst "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
  If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
      DataInmobiliaria.Recordset.FindNext "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
    Else
      Exit Do
    End If
  Else
    DataInmobiliaria.Recordset.MoveFirst
  Exit Do
  End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

Case 0: 'OptionBuscarPMaximo.Value = 0
  MsgBox "NO HAY NINGUN CAMPO DE BUSQUEDA SELECCIONADO"

End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value
End Select 'FormBusquedaInmo.CheckBuscarHabitaciones.Value

```

Case 1: 'Esta seleccionada la opcion Direccion

```

FormBusquedaInmo.TextBuscarDireccion.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarDireccion.Text))
  Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value
    Case 1 'Esta seleccionado el N° de habitaciones
      FormBusquedaInmo.TextBuscarHabitaciones.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

        Case 1 'OptionBuscarPMinimo.Value = 1
          FormBusquedaInmo.TextBuscarPMinimo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
          Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
            Case 1: 'OptionBuscarPMaximo.Value = 1
              FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
              DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
                "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
                  Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
<= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

                Do
                  If Not (DataInmobiliaria.Recordset.NoMatch) Then

                    contador = contador + 1
                    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                      DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
                        "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
& "AND" & "[PrecioEuros] >= " & _
                          Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
"[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
                    Else

```

```

    Exit Do
  End If
Else
  DataInmobiliaria.Recordset.MoveFirst
  Exit Do
End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
  DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
  "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
  Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

Do
  If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
      DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
      "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
      Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
    Else
      Exit Do
    End If
  Else
    DataInmobiliaria.Recordset.MoveFirst
  End If

```

```

    Exit Do
  End If
  Loop While Not (DataInmobiliaria.Recordset.EOF)

  MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
  "Numero de fichas encontradas: " & contador
  contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

  Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
    Case 1: 'OptionBuscarPMaximo.Value = 1
      FormBusquedaInmo.TextBuscarPMaximo.Text =
      UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
      DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
      FormBusquedaInmo.TextBuscarDireccion.Text & "' & "AND" & _
      "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" &
      "[PrecioEuros] <= " & _
      Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

      Do
        If Not (DataInmobiliaria.Recordset.NoMatch) Then

          contador = contador + 1
          If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
          vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
            FormBusquedaInmo.TextBuscarDireccion.Text & "' & "AND" & _
            "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND"
            & "[PrecioEuros] <= " & _
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
          Else

```

```

Exit Do
End If
Else
DataInmobiliaria.Recordset.MoveFirst
Exit Do
End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
"AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text

Do
If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
"AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text
Else
Exit Do
End If
Else
DataInmobiliaria.Recordset.MoveFirst
Exit Do
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

```

```

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
  End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value'

Case 0 'No esta seleccionado el N° de habitaciones
  Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

    Case 1 'OptionBuscarPMinimo.Value = 1
      FormBusquedaInmo.TextBuscarPMinimo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
          FormBusquedaInmo.TextBuscarPMaximo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
          DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
    FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
            "AND" & "[PrecioEuros] >= " &
    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
            "AND" & "[PrecioEuros] <= " &
    Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

          Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

              contador = contador + 1
              If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
    vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
    FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
                    "AND" & "[PrecioEuros] >= " &
    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _

```

```

    "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
    Else
    Exit Do
    End If
    Else
    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
    End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
    "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

    Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI, CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
            "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
        Else
        Exit Do
        End If
    Else

```

```

    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
  End If
  Loop While Not (DataInmobiliaria.Recordset.EOF)

  MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
  "Numero de fichas encontradas: " & contador
  contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

  Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
    Case 1: 'OptionBuscarPMaximo.Value = 1
      FormBusquedaInmo.TextBuscarPMaximo.Text =
      UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
      DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
      FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
      "AND" & "[PrecioEuros] <= " &
      Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

      Do
        If Not (DataInmobiliaria.Recordset.NoMatch) Then

          contador = contador + 1
          If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
          vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
            FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
            "AND" & "[PrecioEuros] <= " &
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
          Else
            Exit Do
          End If
        End If
      Loop
    End Select
  End Select

```



```

    End If
  Else
    DataInmobiliaria.Recordset.MoveFirst
  Exit Do
End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
  DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & ""

Do
  If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
      DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & ""
    Else
      Exit Do
    End If
  Else
    DataInmobiliaria.Recordset.MoveFirst
  Exit Do
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador

```

```

    contador = 0
    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
  End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value
  End Select 'FormBusquedaInmo.CheckBuscar Habitaciones.Value

  End Select 'FormBusquedaInmo.CheckBuscarDireccion.Value

  End Select 'FormBusquedaInmo.checkBuscarZona.Value
  End If 'Buscar_por_ID = True

Else ' sino se quiere realizar una busqueda

  FrameDatosProp1.Enabled = False
  FrameDatosProp2.Enabled = False
  Frameld.Enabled = False

  DataInmobiliaria.Caption = "Ficha " & DataInmobiliaria.Recordset.AbsolutePosition + 1 & " de " & _
  DataInmobiliaria.Recordset.RecordCount

  'Si no hemos borrado el registro del formulario FormMasDatos,no hay que borrar
  ' el registro asociado en FormInmobiliaria.
  'Buscamos el registro asociado y mostramos la ficha
  If borrar_inmobiliaria = False Then
    DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_ficha
    'Si no encontramos ficha asociada creamos un nuevo registro asociado
    If DataInmobiliaria.Recordset.NoMatch = True Then
      MsgBox "No se ha encontrado ninguna ficha asociada,creando ficha asociada...." , ,
      "¡Atencion!"
      DataInmobiliaria.Recordset.AddNew
      DataInmobiliaria.Recordset("IdInmueble") = Id_ficha
      DataInmobiliaria.Recordset("FECHA CAPT") = Date
    End If
  End If
End If

```

```
DataInmobiliaria.Recordset.Update ' Actualiza la base de datos (obligatorio,hasta que no se  
pone no se hace nada)
```

```
DataInmobiliaria.Recordset.MoveLast
```

```
End If
```

```
'Si hemos borrado el registro del formulario FormMasDatos,hay que borrar  
' el registro asociado en FormInmobiliaria.
```

```
Else
```

```
borrar_inmobiliaria = False
```

```
DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_borrar
```

```
DataInmobiliaria.Recordset.Delete
```

```
DataInmobiliaria.Refresh
```

```
DataInmobiliaria.Recordset.Close
```

```
FormInmobiliaria.Hide
```

```
FormMasDatos.Show
```

```
End If
```

```
End If 'Busqueda_activada
```

```
Busqueda_activada = False
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
'Inicializamos el formulario
```

```
Private Sub Form_Initialize()
```

```
TextPrecio = ""
```

```
'cerrojo = True
```

```
ComboProvincia.AddItem "OTRA"
```

```
ComboProvincia.AddItem "VALENCIA"
```

```
ComboProvincia.AddItem "CASTELLÓN"
```

```
ComboProvincia.AddItem "ALICANTE"
```

```
ComboModo.AddItem ""
```

```
ComboModo.AddItem "VENTA"  
ComboModo.AddItem "ALQUILER"  
ComboModo.AddItem "TRASPASO"
```

```
ComboNHabitaciones.AddItem "0"  
ComboNHabitaciones.AddItem "1"  
ComboNHabitaciones.AddItem "2"  
ComboNHabitaciones.AddItem "3"  
ComboNHabitaciones.AddItem "4"  
ComboNHabitaciones.AddItem "5"  
ComboNHabitaciones.AddItem "6"  
ComboNHabitaciones.AddItem "7"  
ComboNHabitaciones.AddItem "8"
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    On Error Resume Next
```

```
    DataInmobiliaria.Connect = "Access 2000;"  
    DataInmobiliaria.DatabaseName = App.Path & "\Inmobiliaria.mdb"  
    DataInmobiliaria.DefaultCursorType = 0 'Defaultcursor  
    DataInmobiliaria.DefaultType = 2 'Usejet  
    DataInmobiliaria.RecordsetType = 1 'Dynaset  
    DataInmobiliaria.RecordSource = "Inmueble"  
    DataInmobiliaria.Refresh
```

```
FrameDatosProp1.Enabled = False
```

FrameDatosProp2.Enabled = False

FrameId.Enabled = False

DataInmobiliaria.Caption = "Ficha " & DataInmobiliaria.Recordset.AbsolutePosition + 1 & " de " & _
DataInmobiliaria.Recordset.RecordCount

If borrar_inmobiliaria = False Then

 DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_ficha

 ,

 If DataInmobiliaria.Recordset.NoMatch = True Then

 MsgBox "No se ha encontrado ninguna ficha asociada,creando ficha asociada....", , "¡Atencion!"

 DataInmobiliaria.Recordset.AddNew

 DataInmobiliaria.Recordset("IdInmueble") = Id_ficha

 DataInmobiliaria.Recordset("FECHA CAPT") = Date

 DataInmobiliaria.Recordset("Pais") = "ESPAÑA"

 DataInmobiliaria.Recordset.Update ' Actualiza la base de datos (obligatorio, hasta que no se
pone no se hace nada)

 DataInmobiliaria.Recordset.MoveLast

 End If

Else

 borrar_inmobiliaria = False

 DataInmobiliaria.Recordset.Delete

 DataInmobiliaria.Refresh

 DataInmobiliaria.Recordset.MoveNext

 If DataInmobiliaria.Recordset.EOF Then

 DataInmobiliaria.Recordset.MoveLast

 End If

DataInmobiliaria.Recordset.Close

Id_ficha = TextIdInm.Text 'Actualizo la variable Id_ficha

cerrar_inmobiliaria = True

Menu2MasDatos_Click

End If

If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation

End Sub

Private Sub Menu2Buscar_Click()

FormBusquedaInmo.Show 1

End Sub

'Imprimo la pantalla

Private Sub Menu2Imprimir_Click()

'Oculto partes del formulario

DataInmobiliaria.Visible = False

ToolbarInmobiliaria.Visible = False

'Imprimo formulario

FormInmobiliaria.PrintForm

'Vuelvo a mostrar las partes del formulario

DataInmobiliaria.Visible = True

ToolbarInmobiliaria.Visible = True

End Sub

Private Sub Menu2Salir_Click()

```
FormMasDatos.MenuSalir_Click
```

```
End Sub
```

```
Private Sub TextComCapt_Click()
```

```
    TextComCapt.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextComCapt.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextComCapt.SelLength = Len(TextComCapt.Text) 'Selecciona todo el texto
```

```
    TextComCapt.SelText = "0"
```

```
End Sub
```

```
Private Sub TextComCliente_Click()
```

```
    TextComCliente.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextComCliente.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextComCliente.SelLength = Len(TextComCliente.Text) 'Selecciona todo el texto
```

```
    TextComCliente.SelText = "0"
```

```
End Sub
```

```
Private Sub TextComProp_Click()
```

```
    TextComProp.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextComProp.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextComProp.SelLength = Len(TextComProp.Text) 'Selecciona todo el texto
```

```
    TextComProp.SelText = "0"
```

```
End Sub
```

```
Private Sub TextCP_Click()
```

```
    TextCP.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextCP.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextCP.SelLength = Len(TextCP.Text) 'Selecciona todo el texto
```

```
    TextCP.SelText = "0"
```

```
End Sub
```

```
Private Sub TextGComunidad_Click()  
    TextGComunidad.SetFocus 'Poner el cursor sobre TextClave  
    TextGComunidad.SelStart = 0 'Posiciona el cursor al comienzo del texto  
    TextGComunidad.SelLength = Len(TextGComunidad.Text) 'Selecciona todo el texto  
    TextGComunidad.SelText = "0"  
End Sub
```

```
Private Sub TextMild_Click()  
    TextMild.SetFocus 'Poner el cursor sobre TextClave  
    TextMild.SelStart = 0 'Posiciona el cursor al comienzo del texto  
    TextMild.SelLength = Len(TextMild.Text) 'Selecciona todo el texto  
    TextMild.SelText = "0"  
End Sub
```

```
Private Sub TextNFinca_Click()  
    TextNFinca.SetFocus 'Poner el cursor sobre TextClave  
    TextNFinca.SelStart = 0 'Posiciona el cursor al comienzo del texto  
    TextNFinca.SelLength = Len(TextNFinca.Text) 'Selecciona todo el texto  
    TextNFinca.SelText = "0"  
End Sub
```

```
Private Sub TextNRegistro_Click()  
    TextNRegistro.SetFocus 'Poner el cursor sobre TextClave  
    TextNRegistro.SelStart = 0 'Posiciona el cursor al comienzo del texto  
    TextNRegistro.SelLength = Len(TextNRegistro.Text) 'Selecciona todo el texto  
    TextNRegistro.SelText = "0"  
End Sub
```



```
Private Sub TextNumero_Click()
```

```
    TextNumero.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextNumero.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextNumero.SelLength = Len(TextNumero.Text) 'Selecciona todo el texto
```

```
    TextNumero.SelText = "0"
```

```
End Sub
```

```
Private Sub TextPGaraje_Click()
```

```
    TextPGaraje.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextPGaraje.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextPGaraje.SelLength = Len(TextPGaraje.Text) 'Selecciona todo el texto
```

```
    TextPGaraje.SelText = "0"
```

```
End Sub
```

```
Private Sub TextPiso_Click()
```

```
    TextPiso.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextPiso.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextPiso.SelLength = Len(TextPiso.Text) 'Selecciona todo el texto
```

```
    TextPiso.SelText = "0"
```

```
End Sub
```

```
Private Sub TextPrecio_Click()
```

```
    TextPrecio.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextPrecio.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextPrecio.SelLength = Len(TextPrecio.Text) 'Selecciona todo el texto
```

```
    TextPrecio.SelText = "0"
```

```
    TextPrecioEuros.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextPrecioEuros.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextPrecioEuros.SelLength = Len(TextPrecioEuros.Text) 'Selecciona todo el texto
```

```
    TextPrecioEuros.SelText = "0"
```

```
TextPrecio.SetFocus
```

```
End Sub
```

```
Public Sub TextPrecio_KeyPress(KeyAscii As Integer)
```

```
On Error Resume Next
```

```
Select Case KeyAscii
```

```
Case 48 To 57: Call Numero(KeyAscii - 48, 0)
```

```
Case 44:
```

```
    If InStr(1, TextPrecio, ",") > 0 Then
```

```
        KeyAscii = 0
```

```
    Else: Coma 0 ' Pulsacion de la coma
```

```
    End If
```

```
Case Else
```

```
    KeyAscii = 0
```

```
End Select
```

```
KeyAscii = 0
```

```
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub TextTasacion_Click()
```

```
TextTasacion.SetFocus 'Poner el cursor sobre TextClave
```

```
TextTasacion.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
TextTasacion.SelLength = Len(TextTasacion.Text) 'Selecciona todo el texto
```

```
TextTasacion.SelText = "0"
```

```
End Sub
```

```
Private Sub TextVsuelo_Click()
```

```
TextVsuelo.SetFocus 'Poner el cursor sobre TextClave
```

```
TextVsuelo.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextVsuelo.SelLength = Len(TextVsuelo.Text) 'Selecciona todo el texto
TextVsuelo.SelText = "0"
End Sub

Private Sub TextVUrbano_Click()
    TextVUrbano.SetFocus 'Poner el cursor sobre TextClave
    TextVUrbano.SelStart = 0 'Posiciona el cursor al comienzo del texto
    TextVUrbano.SelLength = Len(TextVUrbano.Text) 'Selecciona todo el texto
    TextVUrbano.SelText = "0"
End Sub

Private Sub ToolbarInmobiliaria_ButtonClick(ByVal Button As ComctlLib.Button)
    On Error Resume Next
    Select Case Button.Index 'según el botón se ejecutan unas instrucciones
        Case 1: Menu2Editar_Click
        Case 2: Menu2Buscar_Click
        Case 3: Menu2Imprimir_Click
        Case 4: FormMasDatos.MenuCalcular_Click
        Case 5: FormMasDatos.MenuSalir_Click
        Case 6: Menu2MasDatos_Click
    End Select
    If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
End Sub

Private Sub DataInmobiliaria_Validate(Action As Integer, Save As Integer)
    Dim respuesta As Integer
```

Select Case Action

Case vbDataActionDelete, vbDataActionAddNew, vbDataActionFind, vbDataActionClose,
vbDataActionMoveFirst, vbDataActionMovePrevious, vbDataActionMoveNext, vbDataActionMoveLast

 If Save = True Then

 respuesta = MsgBox("¿Desea guardar los cambios?", vbYesNo, "Inmobifich")

 If respuesta = vbNo Then

 Save = False

 DataInmobiliaria.UpdateControls 'Actualiza los cambios

 End If

 End If

End Select

End Sub

Private Sub DataInmobiliaria_Error(DataErr As Integer, Response As Integer)

 'If DataErr = 3022 Then 'Error de clave principal duplicada

 ' MsgBox "Ingrese un Id de inmobiliaria único"

 ' TextIdInm.SetFocus

 ' Response = 0

 'Else

 Response = 1 'Muestra un mensaje de error estándar

 'End If

End Sub

Private Sub Menu2Editar_Click()

 On Error Resume Next

 FrameDatosProp1.Enabled = True

 FrameDatosProp2.Enabled = True

 FrameId.Enabled = True

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
Private Sub Menu2MasDatos_Click()
```

```
    On Error Resume Next
```

```
    FrameDatosProp1.Enabled = False
```

```
    FrameDatosProp2.Enabled = False
```

```
    FrameId.Enabled = False
```

```
FormInmobiliaria.Hide 'Oculto el formulario FormInmobiliaria
```

```
FormMasDatos.Show 'Muestro el formualrio FormMasDatos
```

```
Id_ficha = TextIdInm.Text 'Id_ficha contiene el identificado del registro secundario
```

```
cerrar_inmobiliaria = True 'Indicamos que hay que buscar el registro principal asociado a  
    ' al registro secundario
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

FORMCALC

Option Explicit

Private Memoria As String

Private NumClics As Byte

Private ValorNum1 As Double

Private ValorNum2 As Double

Private Num1 As Boolean

Private Oper As String

Private Sub CommandBorraDato_Click()

On Error Resume Next

TextPantalla.SetFocus

TextPantalla.SelStart = 0 'Las cadenas empiezan de izquierda a derecha

TextPantalla.SelLength = Len(TextPantalla) - 1

TextPantalla = TextPantalla.SelText

Convertidor (NumClics)

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandBorraTodo_Click()

On Error Resume Next

Inicializar

Convertidor (NumClics)

TextPantalla.SetFocus

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandComa_Click()

TextPantalla.SetFocus

```
TextPantalla = TextPantalla & ","
End Sub

Private Sub CommandCoseno_Click()
    On Error Resume Next
    TextPantalla = Cos(CDb1(TextPantalla))
    Convertidor (NumClics)
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
End Sub

Private Sub CommandCuadrado_Click()
    Dim num As Double

    On Error Resume Next
    num = CDb1(TextPantalla)
    TextPantalla = CStr(num * num)
    Convertidor (NumClics)
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandCubo_Click()
    Dim num As Double

    On Error Resume Next
    num = CDb1(TextPantalla)
    TextPantalla = CStr((num * num) * num)
    Convertidor (NumClics)
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

End Sub

Private Sub CommandFact_Click()

Dim i As Integer

Dim num As Long

On Error Resume Next

num = Int(CDbI(TextPantalla))

TextPantalla = CStr(num)

If num > 12 Then

MsgBox "Demasiado grande para calcular x!", vbExclamation, "Aviso"

Exit Sub

End If

For i = num - 1 To 2 Step -1

num = num * i

Next

TextPantalla = CStr(num)

Convertidor (NumClics)

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandIqual_Click()

On Error Resume Next

If (Num1 = True) Then

If (TextPantalla <> "") Then

If CDbI(TextPantalla) <> ValorNum1 Then

ValorNum2 = CDbI(TextPantalla)

End If

End If

Select Case Oper

Case "+": TextPantalla = CStr(ValorNum1 + ValorNum2)

Case "-": TextPantalla = CStr(ValorNum1 - ValorNum2)

Case "*": TextPantalla = CStr(ValorNum1 * ValorNum2)

Case "/": TextPantalla = CStr(ValorNum1 / ValorNum2)

Case "mod": TextPantalla = CStr(ValorNum1 Mod ValorNum2)

Case "%": TextPantalla = CStr((ValorNum1 * ValorNum2) / 100)

End Select

Convertidor (NumClics)

ValorNum1 = CDb(TextPantalla)

End If

TextPantalla.SetFocus

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandInversa_Click()

On Error Resume Next

TextPantalla = CDb(1 / CDb(TextPantalla))

Convertidor (NumClics)

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandMasMenos_Click()

On Error Resume Next

TextPantalla = CDb(0 - CDb(TextPantalla))

Convertidor (NumClics)

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandMemIn_Click()

On Error Resume Next

TextPantalla.SetFocus

Memoria = TextPantalla

TextNumEnMemoria.Locked = False

TextNumEnMemoria = Memoria

TextNumEnMemoria.Locked = True

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandMemOff_Click()

On Error Resume Next

Memoria = ""

TextNumEnMemoria.Locked = False

TextNumEnMemoria = ""

TextNumEnMemoria.Locked = True

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandMemOut_Click()

On Error Resume Next

If Memoria <> "" Then

TextPantalla = Memoria

Else

MsgBox "No hay número memorizado", vbInformation, "Aviso"

End If

Convertidor (NumClics)

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandMod_Click()

On Error Resume Next

Ejecutar "mod"

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandNum_Click(Index As Integer)

Dim num As String

On Error Resume Next

num = ""

TextPantalla.SetFocus

Select Case Index

Case 0: num = num & "0"

Case 1: num = num & "1"

Case 2: num = num & "2"

Case 3: num = num & "3"

Case 4: num = num & "4"

Case 5: num = num & "5"

Case 6: num = num & "6"

Case 7: num = num & "7"

Case 8: num = num & "8"

Case 9: num = num & "9"

End Select

If Num1 = False Then

TextPantalla = ""

Num1 = True

End If

```
TextPantalla = TextPantalla & num
Convertidor (NumClics)
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandDivision_Click()
    On Error Resume Next
    Ejecutar "/"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandMult_Click()
    On Error Resume Next
    Ejecutar "*"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandPorciento_Click()
    On Error Resume Next
    Ejecutar "%"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandResta_Click()
    On Error Resume Next
    Ejecutar "-"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandSeno_Click()  
    On Error Resume Next  
    TextPantalla = Sin(CDb1(TextPantalla))  
    Convertidor (NumClics)  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandSqrt_Click()  
    On Error Resume Next  
    If CDb1(TextPantalla) > 0 Then  
        TextPantalla = CStr(Sqr(CDb1(TextPantalla)))  
        Convertidor (NumClics)  
    Else  
        MsgBox "El número es negativo", vbExclamation, "Aviso"  
    End If  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandSuma_Click()  
    On Error Resume Next  
    Ejecutar "+"  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandTangente_Click()  
    On Error Resume Next  
    TextPantalla = Tan(CDb1(TextPantalla))  
    Convertidor (NumClics)  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

End Sub

Private Sub Form_Load()

On Error Resume Next

Inicializar

Memoria = ""

TextNumEnMemoria = ""

NumClics = 0

TextNumEnMemoria.Locked = True

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub Form_Unload(Cancel As Integer)

On Error Resume Next

If MsgBox("¿Seguro que quiere salir de la EUROCalculadora?", vbOKCancel + vbExclamation, "Confirmación") = vbOK Then

FormCalc.Hide

End If

If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation

End Sub

Private Sub Salir_Click()

On Error Resume Next

If MsgBox("¿Seguro que quiere salir de la EUROCalculadora?", vbOKCancel + vbExclamation, "Confirmación") = vbOK Then

FormCalc.Hide

End If

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub TextEuroPantalla_Click()

On Error Resume Next

TextEuroPantalla.BackColor = RGB(255, 255, 255)

TextEuroPantalla.ForeColor = RGB(0, 0, 0)

TextEuroPantalla.SetFocus

Select Case NumClics

Case 0

NumClics = 1

Convertidor (NumClics)

Case 1

NumClics = 2

Convertidor (NumClics)

Case 2

NumClics = 0

Convertidor (NumClics)

End Select

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub Ejecutar(operacion As String)

Dim Valor As Double

On Error Resume Next

Valor = 0

If Num1 = True Then

```
Oper = operacion
Valor = CDbI(TextPantalla)
End If
If Valor <> ValorNum1 Then
    ValorNum1 = Valor
    Num1 = True
End If
TextPantalla = ""
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub
'Reconoce la tecla pulsada y ejecuta la accion asociada
Private Sub TextPantalla_KeyPress(KeyAscii As Integer)

    On Error Resume Next
    Select Case KeyAscii
        Case 48 To 57: CommandNum_Click (KeyAscii - 48)

        Case 42: CommandMult_Click 'Multiplicacion

        Case 43: CommandSuma_Click 'Suma

        Case 44:
            If InStr(1, TextPantalla, ",") > 0 Then
                KeyAscii = 0
            Else: CommandComa_Click ' Pulsacion de la coma
            End If

        Case 45: CommandResta_Click 'Resta

        Case 47: CommandDivision_Click 'Division

        Case 13: CommandIguar_Click 'Retorno de linea (INTRO)
```


Case 61: CommandIguar_Click "="

Case 127:

 CommandBorraTodo_Click

Case Else

 KeyAscii = 0

End Select

KeyAscii = 0

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

'Funcion para asignar valores iniciales

Private Sub Inicializar()

 On Error Resume Next

 TextPantalla = "0"

 Num1 = False

 ValorNum1 = 0

 ValorNum2 = 0

 Oper = ""

 TextNumEnMemoria.FontSize = 10

 If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

'Funcion para convertir euros a pesetas y viceversa

Private Sub Convertidor(NumClics As Byte)

 On Error Resume Next

 If TextPantalla = "" Then

 TextPantalla = "0"

 End If

 Select Case NumClics

 Case 1: TextEuroPantalla = Format(CDbl(TextPantalla) / 166.386, "#,##0.00 €")

Case 2: TextEuroPantalla = Format(CDbl(TextPantalla) * 166.386, "#,##0 Pts")

Case 0: TextEuroPantalla.BackColor = RGB(255, 128, 128)

TextEuroPantalla = "EuroPantalla"

End Select

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

6. LISTADO DE CÓDIGO COMENTADO

FORMACERCA

Option Explicit

' Opciones de seguridad de clave del Registro...

Const READ_CONTROL = &H20000

Const KEY_QUERY_VALUE = &H1

Const KEY_SET_VALUE = &H2

Const KEY_CREATE_SUB_KEY = &H4

Const KEY_ENUMERATE_SUB_KEYS = &H8

Const KEY_NOTIFY = &H10

Const KEY_CREATE_LINK = &H20

Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
 KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
 KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Tipos ROOT de clave del Registro...

Const HKEY_LOCAL_MACHINE = &H80000002

Const ERROR_SUCCESS = 0

Const REG_SZ = 1 ' Cadena Unicode terminada en valor nulo

Const REG_DWORD = 4 ' Número de 32 bits

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"

Const gREGVALSYSINFOLOC = "MSINFO"

Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"

Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As Long

```
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
```

```
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long
```

```
Private Sub cmdSysInfo_Click()
```

```
    Call StartSysInfo
```

```
End Sub
```

```
Private Sub cmdOK_Click()
```

```
    FormAcerca.Hide
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
    'FormAcerca.LabelDiasNoReg.Caption = CStr(dias_sin_registrar)
```

```
    'FormAcerca.LabelNumEjec.Caption = num_ejec
```

```
    'FormAcerca.LabelDiasEjec.Caption = num_dias_ejec
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Me.Caption = "Acerca de " & "INMOBIFICHA"
```

```
    'FormAcerca.LabelDiasNoReg.Caption = CStr(dias_sin_registrar)
```

```
    'FormAcerca.LabelNumEjec.Caption = num_ejec
```

```
    'FormAcerca.LabelDiasEjec.Caption = num_dias_ejec
```

```
End Sub
```

```
Public Sub StartSysInfo()
```

```
    On Error GoTo SysInfoErr
```

```
    Dim rc As Long
```

Dim SysInfoPath As String

' Intentar obtener ruta de acceso y nombre del programa de Info. del sistema a partir del Registro...

If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath)
Then

' Intentar obtener sólo ruta del programa de Info. del sistema a partir del Registro...

Elseif GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC,
SysInfoPath) Then

' Validar la existencia de versión conocida de 32 bits del archivo

If (Dir(SysInfoPath & "MSINFO32.EXE") <> "") Then

 SysInfoPath = SysInfoPath & "MSINFO32.EXE"

' Error: no se puede encontrar el archivo...

Else

 GoTo SysInfoErr

End If

' Error: no se puede encontrar la entrada del Registro...

Else

 GoTo SysInfoErr

End If

Call Shell(SysInfoPath, vbNormalFocus)

Exit Sub

SysInfoErr:

 MsgBox "La información del sistema no está disponible en este momento", vbOKOnly

End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String, ByRef
KeyVal As String) As Boolean

 Dim i As Long ' Contador de bucle

 Dim rc As Long ' Código de retorno

 Dim hKey As Long ' Controlador de una clave de Registro abierta

```

Dim hDepth As Long          '
Dim KeyValType As Long      ' Tipo de datos de una clave de Registro
Dim tmpVal As String        ' Almacenamiento temporal para un valor de clave de
Registro
Dim KeyValSize As Long      ' Tamaño de variable de clave de Registro
'-----
' Abrir clave de registro bajo KeyRoot {HKEY_LOCAL_MACHINE...}
'-----
rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Abrir clave de Registro

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Error de controlador...

tmpVal = String$(1024, 0)          ' Asignar espacio de variable
KeyValSize = 1024                  ' Marcar tamaño de variable

'-----
' Obtener valor de clave de Registro...
'-----
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
                    KeyValType, tmpVal, KeyValSize) ' Obtener o crear valor de clave

If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Controlar errores

If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then      ' Win95 agregar cadena terminada en valor nulo...
    tmpVal = Left(tmpVal, KeyValSize - 1)          ' Encontrado valor nulo, se va a quitar de la cadena
Else                                                ' En WinNT las cadenas no terminan en valor nulo...
    tmpVal = Left(tmpVal, KeyValSize)              ' No se ha encontrado valor nulo, sólo se va a
    extraer la cadena
End If
'-----
' Determinar tipo de valor de clave para conversión...
'-----

```

```

Select Case KeyValType          ' Buscar tipos de datos...
Case REG_SZ                    ' Tipo de datos String de clave de Registro
    KeyVal = tmpVal            ' Copiar valor de cadena
Case REG_DWORD                 ' Tipo de datos Double Word de clave del Registro
    For i = Len(tmpVal) To 1 Step -1      ' Convertir cada bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Generar valor carácter a carácter
    Next
    KeyVal = Format("&h" + KeyVal)        ' Convertir Double Word a cadena
End Select

```

```

GetKeyValue = True              ' Se ha devuelto correctamente
rc = RegCloseKey(hKey)         ' Cerrar clave de Registro
Exit Function                   ' Salir

```

```

GetKeyError:   ' Borrar después de que se produzca un error...
    KeyVal = ""          ' Establecer valor a cadena vacía
    GetKeyValue = False ' Fallo de retorno
    rc = RegCloseKey(hKey) ' Cerrar clave de Registro
End Function

```

FORMBUSQUEDAINMO

Option Explicit

Dim error As Boolean

Private Sub CommandBorrarBusqueda_Click()

'Asignamos valores iniciales

CheckBuscarID.Value = 0 '0 es desactivado (predeterminado), 1 es activado y 2 es atenuado.

CheckBuscarZona.Value = 0

CheckBuscarDireccion.Value = 0

CheckBuscarHabitaciones.Value = 0

CheckBuscarPMinimo.Value = 0

CheckBuscarPMaximo.Value = 0

```
TextBuscarID.Text = ""  
TextBuscarZona.Text = ""  
TextBuscarDireccion.Text = ""  
TextBuscarHabitaciones.Text = ""  
TextBuscarPMinimo.Text = ""  
TextBuscarPMaximo.Text = ""  
Buscar_por_ID = False  
Busqueda_activada = False
```

End Sub

```
Private Sub CommandBuscar_Click()
```

```
    error = False
```

```
    'Podemos buscar o bien por Identificador de registro o por
```

```
    'Zona,Direccion,numero de habitaciones,Precio minimo y máximo
```

```
    If CheckBuscarID.Value = 1 Then
```

```
        MsgBox "Seleccionado ID INM....BUSQUEDA SOLO POR IDENTIFICADOR", vbInformation +  
vbOKOnly, "BUSQUEDA POR ID INM"
```

```
        Buscar_por_ID = True
```

```
        If TextBuscarID.Text = "" Then
```

```
            error = True
```

```
        End If
```

```
    Else
```

```
        If CheckBuscarZona.Value = 1 Then
```

```
            If TextBuscarZona.Text = "" Then error = True
```

```
        End If
```

```
        If CheckBuscarDireccion.Value = 1 Then
```



```
        If TextBuscarDireccion.Text = "" Then error = True
    End If
    If CheckBuscarHabitaciones.Value = 1 Then
        If TextBuscarHabitaciones.Text = "" Then error = True
    End If
    If CheckBuscarPMinimo.Value = 1 Then
        If TextBuscarPMinimo.Text = "" Then error = True
    End If
    If CheckBuscarPMaximo.Value = 1 Then
        If TextBuscarPMaximo.Text = "" Then error = True
    End If
End If

If error = True Then 'Si ha ocurrido un error
    MsgBox "ERROR: Hay un campo seleccionado sin datos", vbExclamation + vbOKOnly, "ERROR"

Else 'Si no ha ocurrido un error
    Busqueda_activada = True 'Activamos la busqueda
    FormBusquedaInmo.Hide 'Ocultamos el formulario FormBusquedaInmo
    FormInmobiliaria.Show 'Mostramos el formulario FormInmobiliaria
End If

End Sub

Private Sub CommandCerrarBuscarInmo_Click()

    FrameLugar.Enabled = True
    FrameCaracteristicas.Enabled = True
    Buscar_por_ID = False 'Deshabilitamos busqueda por Identificador
    FormBusquedaInmo.Hide 'Ocultamos el formulario FormBusquedaInmo
```

```
FormInmobiliaria.Show 'Mostramos el formulario FormInmobiliaria
End Sub

Private Sub Form_Activate()

    error = False

    Buscar_por_ID = False 'Deshabilitamos búsqueda por Identificador
    Busqueda_activada = False 'Deshabilitamos búsqueda

End Sub

Private Sub Form_Load()
    'Asignamos valores iniciales
    CheckBuscarID.Value = 0 '0 es desactivado (predeterminado), 1 es activado y 2 es atenuado.
    CheckBuscarZona.Value = 0
    CheckBuscarDireccion.Value = 0
    CheckBuscarHabitaciones.Value = 0
    CheckBuscarPMinimo.Value = 0
    CheckBuscarPMaximo.Value = 0

    TextBuscarID.Text = ""
    TextBuscarZona.Text = ""
    TextBuscarDireccion.Text = ""
    TextBuscarHabitaciones.Text = ""
    TextBuscarPMinimo.Text = ""
    TextBuscarPMaximo.Text = ""
    error = False

    Buscar_por_ID = False
```

Busqueda_activada = False

End Sub

FORMCALC

Option Explicit

Private Memoria As String

Private NumClics As Byte

Private ValorNum1 As Double

Private ValorNum2 As Double

Private Num1 As Boolean

Private Oper As String

Private Sub CommandBorraDato_Click()

On Error Resume Next

TextPantalla.SetFocus

TextPantalla.SelStart = 0 'Las cadenas empiezan de izquierda a derecha

TextPantalla.SelLength = Len(TextPantalla) - 1

TextPantalla = TextPantalla.SelText

Convertidor (NumClics)

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandBorraTodo_Click()

On Error Resume Next

Inicializar

Convertidor (NumClics)

TextPantalla.SetFocus

```
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
End Sub
```

```
Private Sub CommandComa_Click()
    TextPantalla.SetFocus
    TextPantalla = TextPantalla & ","
End Sub
```

```
Private Sub CommandCoseno_Click()
    On Error Resume Next
    TextPantalla = Cos(CDbl(TextPantalla))
    Convertidor (NumClics)
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
End Sub
```

```
Private Sub CommandCuadrado_Click()
    Dim num As Double

    On Error Resume Next
    num = CDbl(TextPantalla)
    TextPantalla = CStr(num * num)
    Convertidor (NumClics)
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub
```

```
Private Sub CommandCubo_Click()
    Dim num As Double

    On Error Resume Next
```

```
num = CDbI(TextPantalla)
TextPantalla = CStr((num * num) * num)
Convertidor (NumClics)
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandFact_Click()
```

```
Dim i As Integer
Dim num As Long
```

```
On Error Resume Next
```

```
num = Int(CDbI(TextPantalla))
TextPantalla = CStr(num)
```

```
If num > 12 Then
```

```
MsgBox "Demasiado grande para calcular x!", vbExclamation, "Aviso"
```

```
Exit Sub
```

```
End If
```

```
For i = num - 1 To 2 Step -1
```

```
num = num * i
```

```
Next
```

```
TextPantalla = CStr(num)
```

```
Convertidor (NumClics)
```

```
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandIguar_Click()
```

```
On Error Resume Next
```

```
If (Num1 = True) Then
```

```
If (TextPantalla <> "") Then
    If CDb(TextPantalla) <> ValorNum1 Then
        ValorNum2 = CDb(TextPantalla)
    End If
End If

Select Case Oper
    Case "+": TextPantalla = CStr(ValorNum1 + ValorNum2)
    Case "-": TextPantalla = CStr(ValorNum1 - ValorNum2)
    Case "*": TextPantalla = CStr(ValorNum1 * ValorNum2)
    Case "/": TextPantalla = CStr(ValorNum1 / ValorNum2)
    Case "mod": TextPantalla = CStr(ValorNum1 Mod ValorNum2)
    Case "%": TextPantalla = CStr((ValorNum1 * ValorNum2) / 100)
End Select

Convertidor (NumClics)
ValorNum1 = CDb(TextPantalla)
End If
TextPantalla.SetFocus
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

End Sub

```
Private Sub CommandInversa_Click()
    On Error Resume Next
    TextPantalla = CDb(1 / CDb(TextPantalla))
    Convertidor (NumClics)
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

End Sub

```
Private Sub CommandMasMenos_Click()
    On Error Resume Next
```

```
TextPantalla = CDb(0 - CDb(TextPantalla))  
Convertidor (NumClics)  
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

End Sub

```
Private Sub CommandMemIn_Click()  
    On Error Resume Next  
    TextPantalla.SetFocus  
    Memoria = TextPantalla  
    TextNumEnMemoria.Locked = False  
    TextNumEnMemoria = Memoria  
    TextNumEnMemoria.Locked = True  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

End Sub

```
Private Sub CommandMemOff_Click()  
    On Error Resume Next  
    Memoria = ""  
    TextNumEnMemoria.Locked = False  
    TextNumEnMemoria = ""  
    TextNumEnMemoria.Locked = True  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

End Sub

```
Private Sub CommandMemOut_Click()  
    On Error Resume Next  
    If Memoria <> "" Then  
        TextPantalla = Memoria
```

```
Else
    MsgBox "No hay número memorizado", vbInformation, "Aviso"
End If
Convertidor (NumClics)
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandMod_Click()
```

```
    On Error Resume Next
    Ejecutar "mod"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandNum_Click(Index As Integer)
```

```
    Dim num As String

    On Error Resume Next
    num = ""
    TextPantalla.SetFocus
    Select Case Index
        Case 0: num = num & "0"
        Case 1: num = num & "1"
        Case 2: num = num & "2"
        Case 3: num = num & "3"
        Case 4: num = num & "4"
        Case 5: num = num & "5"
        Case 6: num = num & "6"
        Case 7: num = num & "7"
        Case 8: num = num & "8"
        Case 9: num = num & "9"
```

```
End Select
```



```
If Num1 = False Then
    TextPantalla = ""
    Num1 = True
End If
TextPantalla = TextPantalla & num
Convertidor (NumClics)
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandDivision_Click()
    On Error Resume Next
    Ejecutar "/"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandMult_Click()
    On Error Resume Next
    Ejecutar "*"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandPorciento_Click()
    On Error Resume Next
    Ejecutar "%"
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub CommandResta_Click()
    On Error Resume Next
```

Ejecutar "-"

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandSeno_Click()

On Error Resume Next

TextPantalla = Sin(CDbl(TextPantalla))

Convertidor (NumClics)

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandSqrt_Click()

On Error Resume Next

If CDbl(TextPantalla) > 0 Then

TextPantalla = CStr(Sqr(CDbl(TextPantalla)))

Convertidor (NumClics)

Else

MsgBox "El número es negativo", vbExclamation, "Aviso"

End If

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub CommandSuma_Click()

On Error Resume Next

Ejecutar "+"

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

```
Private Sub CommandTangente_Click()  
    On Error Resume Next  
    TextPantalla = Tan(CDb(TextPantalla))  
    Convertidor (NumClics)  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation  
  
End Sub  
  
Private Sub Form_Load()  
    On Error Resume Next  
    Inicializar  
    Memoria = ""  
    TextNumEnMemoria = ""  
    NumClics = 0  
    TextNumEnMemoria.Locked = True  
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation  
  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    On Error Resume Next  
    If MsgBox("¿Seguro que quiere salir de la EUROCalculadora?", vbOKCancel + vbExclamation,  
"Confirmación") = vbOK Then  
  
        FormCalc.Hide  
    End If  
    If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation  
End Sub  
  
Private Sub Salir_Click()
```

```
On Error Resume Next
If MsgBox("¿Seguro que quiere salir de la EUROCalculadora?", vbOKCancel + vbExclamation,
"Confirmación") = vbOK Then

    FormCalc.Hide
End If

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub TextEuroPantalla_Click()
    On Error Resume Next
    TextEuroPantalla.BackColor = RGB(255, 255, 255)
    TextEuroPantalla.ForeColor = RGB(0, 0, 0)
    TextEuroPantalla.SetFocus
    Select Case NumClics
    Case 0
        NumClics = 1
        Convertidor (NumClics)
    Case 1
        NumClics = 2
        Convertidor (NumClics)

    Case 2
        NumClics = 0
        Convertidor (NumClics)
    End Select
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub Ejecutar(operacion As String)
```

```
Dim Valor As Double

On Error Resume Next
Valor = 0
If Num1 = True Then
    Oper = operacion
    Valor = CDbI(TextPantalla)
End If
If Valor <> ValorNum1 Then
    ValorNum1 = Valor
    Num1 = True
End If
TextPantalla = ""
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub
'Reconoce la tecla pulsada y ejecuta la accion asociada
Private Sub TextPantalla_KeyPress(KeyAscii As Integer)

    On Error Resume Next
    Select Case KeyAscii
        Case 48 To 57: CommandNum_Click (KeyAscii - 48)

        Case 42: CommandMult_Click 'Multiplicacion

        Case 43: CommandSuma_Click 'Suma

        Case 44:
            If InStr(1, TextPantalla, ",") > 0 Then
                KeyAscii = 0
            Else: CommandComa_Click ' Pulsacion de la coma
            End If
```

Case 45: CommandResta_Click 'Resta

Case 47: CommandDivision_Click 'Division

Case 13: CommandIguar_Click 'Retorno de linea (INTRO)

Case 61: CommandIguar_Click "="

Case 127:

 CommandBorraTodo_Click

Case Else

 KeyAscii = 0

End Select

KeyAscii = 0

If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

'Funcion para asignar valores iniciales

Private Sub Inicializar()

 On Error Resume Next

 TextPantalla = "0"

 Num1 = False

 ValorNum1 = 0

 ValorNum2 = 0

 Oper = ""

 TextNumEnMemoria.FontSize = 10

 If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

'Funcion para convertir euros a pesetas y viceversa

Private Sub Convertidor(NumClics As Byte)

 On Error Resume Next

```
If TextPantalla = "" Then
    TextPantalla = "0"
End If
Select Case NumClics
    Case 1: TextEuroPantalla = Format(CDbI(TextPantalla) / 166.386, "#,##0.00 €")
    Case 2: TextEuroPantalla = Format(CDbI(TextPantalla) * 166.386, "#,##0 Pts")
    Case 0: TextEuroPantalla.BackColor = RGB(255, 128, 128)
        TextEuroPantalla = "EuroPantalla"
End Select
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub
```

FORMFOTOGRAFIAS

```
Option Explicit
'Evento que se produce cuando el usuario selecciona una nueva unidad
Private Sub dirDirectorio_Change()
    filLista.Path = dirDirectorio.Path 'Guardamos el directorio
End Sub

'Evento que se produce cuando el usuario selecciona un nuevo directorio
Private Sub drvUnidad_Change()

    dirDirectorio.Path = drvUnidad.Drive 'Guardamos la unidad
End Sub

'Evento que se produce cuando el usuario hace click en un archivo
Private Sub filLista_Click()
    ImageFoto.Picture = LoadPicture(dirDirectorio.Path & "\" & filLista.FileName)
    FrameFotografia.Caption = filLista.FileName
End Sub
```

End Sub

Private Sub Form_Activate()

'El gráfico cambia su tamaño para ajustarse al de el control image
ImageFoto.Stretch = True

End Sub

Private Sub Form_Load()

ImageFoto.Stretch = True

End Sub

Private Sub Menu3Salir_Click()

FormFotografias.Hide 'Ocultar formulario FormFotografias

End Sub

FORMINMOBILIARIA

Option Explicit

'Private cerrojo As Boolean

Private contador As Integer 'Numero de registros que se han encontrado

'Private reg_ant As String

'El evento Reposition ocurre cuando cambia el registro activo

Private Sub DataInmobiliaria_Reposition()

On Error Resume Next

'Elimino los espacios delante y detrás y paso a mayúsculas

ComboZona.Text = UCase(Trim(ComboZona.Text))

TextTipo.Text = UCase(Trim(TextTipo.Text))

TextEdificio.Text = UCase(Trim(TextEdificio.Text))

TextDireccion.Text = UCase(Trim(TextDireccion.Text))


```
TextPoblacion.Text = UCase(Trim(TextPoblacion.Text))
TextPaís.Text = UCase(Trim(TextPaís.Text))
TextPropietario.Text = UCase(Trim(TextPropietario.Text))
TextPuerta.Text = UCase(Trim(TextPuerta.Text))
TextObsGastos.Text = UCase(Trim(TextObsGastos.Text))
TextContacto.Text = UCase(Trim(TextContacto.Text))
TextLlaves.Text = UCase(Trim(TextLlaves.Text))
TextCaptador.Text = UCase(Trim(TextCaptador.Text))
TextLibro.Text = UCase(Trim(TextLibro.Text))
TextObservaciones.Text = UCase(Trim(TextObservaciones.Text))
'Deshabilito los frames
FrameDatosProp1.Enabled = False
FrameDatosProp2.Enabled = False
FrameId.Enabled = False

DataInmobiliaria.Caption = "Ficha " & DataInmobiliaria.Recordset.AbsolutePosition + 1 & " de " & _
DataInmobiliaria.Recordset.RecordCount
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

End Sub

```
Private Sub Form_Activate()
```

```
On Error Resume Next
```

```
DataInmobiliaria.Connect = "Access 2000;"
DataInmobiliaria.DatabaseName = App.Path & "\Inmobiliaria.mdb"
DataInmobiliaria.DefaultCursorType = 0 'Defaultcursor
DataInmobiliaria.DefaultType = 2 'Usejet
DataInmobiliaria.RecordsetType = 1 'Dynaset
DataInmobiliaria.RecordSource = "Inmueble"
```

DataInmobiliaria.Refresh

'Implementación de la búsqueda. Se puede buscar por Identificador o
' por el resto de campos de búsqueda

'cerrojo = True

If Busqueda_activada Then 'Si se quiere realizar una búsqueda

If Buscar_por_ID = True Then

FormBusquedaInmo.TextBuscarID.Text = UCase(Trim(FormBusquedaInmo.TextBuscarID.Text))

If ValidarNumero(FormBusquedaInmo.TextBuscarID.Text) > 0 Then

Exit Sub

End If

DataInmobiliaria.Recordset.FindFirst "[!IdInmueble] LIKE " & FormBusquedaInmo.TextBuscarID.Text
& ""

If DataInmobiliaria.Recordset.NoMatch = True Then

MsgBox "No se han encontrado fichas con ID INM: " & FormBusquedaInmo.TextBuscarID.Text

End If

Else

Select Case FormBusquedaInmo.CheckBuscarZona.Value

Case 1 ' Esta seleccionada la opcion de zona

Select Case FormBusquedaInmo.CheckBuscarDireccion.Value

Case 0: ' No esta seleccionada la opcion Direccion

FormBusquedaInmo.TextBuscarZona.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarZona.Text))

Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value

Case 1 'Esta seleccionado el N° de habitaciones

FormBusquedaInmo.TextBuscarHabitaciones.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))

Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

Case 1 'OptionBuscarPMinimo.Value = 1

```

FormBusquedaInmo.TextBuscarPMinimo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
            FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
            DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
                & "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
                & "AND" & "[PrecioEuros] >= " & _
                    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
                    <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

            Do
                If Not (DataInmobiliaria.Recordset.NoMatch) Then

                    contador = contador + 1
                    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                        DataInmobiliaria.Recordset.FindNext "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
                            & "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
                                Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
                                "[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
                            Else
                                Exit Do
                            End If
                        Else
                            DataInmobiliaria.Recordset.MoveFirst
                            Exit Do
                        End If
                    Loop While Not (DataInmobiliaria.Recordset.EOF)
                    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _

```

```

"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
        & "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
& "AND" & "[PrecioEuros] >= " & _
        Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
                & "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
                Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

```

Case 0 'OptionBuscarPMinimo.Value = 0

Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 1: 'OptionBuscarPMaximo.Value = 1

```

    FormBusquedaInmo.TextBuscarPMaximo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
    FormBusquedaInmo.TextBuscarZona.Text & "' _
    & "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
    & "AND" & "[PrecioEuros] <= " & _
    Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

    Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
            FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND" & "[NHab] = " &
            FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] <= " & _
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If

    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _

```

```

"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
    & "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value'

Case 0 'No esta seleccionado el N° de habitaciones

```

```
Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value
```

```
Case 1 'OptionBuscarPMinimo.Value = 1
```

```
FormBusquedaInmo.TextBuscarPMinimo.Text =  
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
```

```
Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
```

```
Case 1: 'OptionBuscarPMaximo.Value = 1
```

```
FormBusquedaInmo.TextBuscarPMaximo.Text =  
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
```

```
DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &  
FormBusquedaInmo.TextBuscarZona.Text & "' _
```

```
& "AND" & "[PrecioEuros] >= " &  
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
```

```
"AND" & "[PrecioEuros] <= " &  
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
```

```
Do
```

```
If Not (DataInmobiliaria.Recordset.NoMatch) Then
```

```
contador = contador + 1
```

```
If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",  
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
```

```
DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &  
FormBusquedaInmo.TextBuscarZona.Text & "' _
```

```
& "AND" & "[PrecioEuros] >= " &  
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
```

```
"AND" & "[PrecioEuros] <= " &  
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
```

```
Else
```

```
Exit Do
```

```
End If
```

```
Else
```

```
DataInmobiliaria.Recordset.MoveFirst
```

```
Exit Do
```

```

        End If
        Loop While Not (DataInmobiliaria.Recordset.EOF)
        MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
        "Numero de fichas encontradas: " & contador
        contador = 0
        Case 0: 'OptionBuscarPMaximo.Value = 0
            DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                contador = contador + 1
                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                    DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' & _
                    "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
                Else
                    Exit Do
                End If
            Else
                DataInmobiliaria.Recordset.MoveFirst
                Exit Do
            End If
        Loop While Not (DataInmobiliaria.Recordset.EOF)

        MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
        "Numero de fichas encontradas: " & contador
        contador = 0

```


End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 1: 'OptionBuscarPMaximo.Value = 1

FormBusquedaInmo.TextBuscarPMaximo.Text =
 UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))

DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
 FormBusquedaInmo.TextBuscarZona.Text & "' _

& "AND" & "[PrecioEuros] <= '" &
 Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do

If Not (DataInmobiliaria.Recordset.NoMatch) Then

contador = contador + 1

If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
 vbOKCancel + vbInformation, "Busqueda") = vbOK Then

DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
 FormBusquedaInmo.TextBuscarZona.Text & "' _

& "AND" & "[PrecioEuros] <= '" &
 Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Else

Exit Do

End If

Else

DataInmobiliaria.Recordset.MoveFirst

Exit Do

End If

Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _

"Numero de fichas encontradas: " & contador

```

    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & ""

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                contador = contador + 1
                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                    DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & ""
                Else
                    Exit Do
                End If
            Else
                DataInmobiliaria.Recordset.MoveFirst
                Exit Do
            End If
        Loop While Not (DataInmobiliaria.Recordset.EOF)

        MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
        contador = 0
    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value
End Select 'FormBusquedaInmo.CheckBuscar Habitaciones.Value

```

Case 1: 'Esta seleccionada la opcion Direccion

```

FormBusquedaInmo.TextBuscarDireccion.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarDireccion.Text))
    Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value
        Case 1 'Esta seleccionado el N° de habitaciones
            FormBusquedaInmo.TextBuscarHabitaciones.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))
                Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

                    Case 1 'OptionBuscarPMinimo.Value = 1
                        FormBusquedaInmo.TextBuscarPMinimo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
                            Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
                                Case 1: 'OptionBuscarPMaximo.Value = 1
                                    FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
                                        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
                                            & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "' & _
                                                "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
                                                    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
<= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

                                        Do
                                            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                                                contador = contador + 1
                                                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                                                    DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
                                                        & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _

```

```

        "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
& "AND" & "[PrecioEuros] >= " & _
        Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
"[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
        Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "'" & _
        & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "'" & _
        "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
        Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

        Do
            If Not (DataInmobiliaria.Recordset.NoMatch) Then

                contador = contador + 1
                If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                    DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "'" & _
                    & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _

```

```

    "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscar Habitaciones.Text & "AND" & "[PrecioEuros] >= " & _
    Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
    Else
    Exit Do
    End If
    Else
    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
    End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
    Case 1: 'OptionBuscarPMaximo.Value = 1
        FormBusquedaInmo.TextBuscarPMaximo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
        DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
        & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "' _
        & "AND" & _
        "[NHab] = " & FormBusquedaInmo.TextBuscar Habitaciones.Text & "AND" &
"[PrecioEuros] <= " & _
        Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

    Do
        If Not (DataInmobiliaria.Recordset.NoMatch) Then

```

```

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & "AND" & _
            "[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND"
& "[PrecioEuros] <= '" & _
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
    & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "' & _
    "AND" & "[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1

```

```

    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
        DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' _
            & "AND [NHab] = '" &
FormBusquedaInmo.TextBuscarHabitaciones.Text
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0

    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value

Case 0 'No esta seleccionado el N° de habitaciones
    Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

        Case 1 'OptionBuscarPMinimo.Value = 1
            FormBusquedaInmo.TextBuscarPMinimo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
            Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
                Case 1: 'OptionBuscarPMaximo.Value = 1
                    FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))

```

```

    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
    & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
& "' & _
    "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
    "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' _
            & "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
            "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
            "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

```



```

Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
    & "AND [Direccion] LIKE " & FormBusquedaInmo.TextBuscarDireccion.Text
& "" & _
    "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE " &
FormBusquedaInmo.TextBuscarZona.Text & "" _
            & "AND [Direccion] LIKE " &
FormBusquedaInmo.TextBuscarDireccion.Text & "" & _
            "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
  
```

Case 0 'OptionBuscarPMinimo.Value = 0

Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 1: 'OptionBuscarPMaximo.Value = 1

FormBusquedaInmo.TextBuscarPMaximo.Text =
 UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))

DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
 FormBusquedaInmo.TextBuscarZona.Text & "' _
 & "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text
 & "' & _

"AND" & "[PrecioEuros] <= " &
 Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do

If Not (DataInmobiliaria.Recordset.NoMatch) Then

contador = contador + 1

If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
 vbOKCancel + vbInformation, "Busqueda") = vbOK Then

DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
 FormBusquedaInmo.TextBuscarZona.Text & "' _

& "AND [Direccion] LIKE '" &
 FormBusquedaInmo.TextBuscarDireccion.Text & "' & _

"AND" & "[PrecioEuros] <= " &
 Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Else

Exit Do

End If

Else

DataInmobiliaria.Recordset.MoveFirst

Exit Do

End If

Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _

```

"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' & _
    "AND [Direccion] LIKE '" & FormBusquedaInmo.TextBuscarDireccion.Text &
    ""
Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then
        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Zona] LIKE '" &
FormBusquedaInmo.TextBuscarZona.Text & "' & _
            "AND [Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & ""
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value

```

End Select 'FormBusquedaInmo.CheckBuscarHabitaciones.Value

End Select 'FormBusquedaInmo.CheckBuscarDireccion.Value

Case 0 ' No Esta seleccionada la opcion zona

Select Case FormBusquedaInmo.CheckBuscarDireccion.Value

Case 0: ' No esta seleccionada la opcion Direccion

Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value

Case 1 'Esta seleccionado el N° de habitaciones

FormBusquedaInmo.TextBuscarHabitaciones.Text =

UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))

Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

Case 1 'OptionBuscarPMinimo.Value = 1

FormBusquedaInmo.TextBuscarPMinimo.Text =

UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))

Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 1: 'OptionBuscarPMaximo.Value = 1

```

    FormBusquedaInmo.TextBuscarPMaximo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
    DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
    FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
    <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

    Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1

        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
        vbOKCancel + vbInformation, "Busqueda") = vbOK Then

            DataInmobiliaria.Recordset.FindNext "[NHab] = " &
            FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
            Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
            "[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If

    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0

    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
        FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
        Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

    Do

```

```

    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
            Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
            FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
            DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] <= " & _
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

        Do

```

```

If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
        DataInmobiliaria.Recordset.FindNext "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] <= " & _
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
    Else
        Exit Do
    End If
Else
    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text
        Else
            Exit Do
        End If
    End If
End Do

```

```

    End If
  Else
    DataInmobiliaria.Recordset.MoveFirst
  Exit Do
End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value'
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value'

Case 0 'No esta seleccionado el N° de habitaciones
  Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

    Case 1 'OptionBuscarPMinimo.Value = 1
      FormBusquedaInmo.TextBuscarPMinimo.Text =
      UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
          FormBusquedaInmo.TextBuscarPMaximo.Text =
          UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
          DataInmobiliaria.Recordset.FindFirst "[PrecioEuros] >= " &
          Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
          "AND" & "[PrecioEuros] <= " &
          Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

        Do
          If Not (DataInmobiliaria.Recordset.NoMatch) Then

            contador = contador + 1

```



```

    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
        DataInmobiliaria.Recordset.FindNext "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
            "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

    Do
        If Not (DataInmobiliaria.Recordset.NoMatch) Then

            contador = contador + 1

            If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                DataInmobiliaria.Recordset.FindNext "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
            Else
                Exit Do
            End If
        Else
            Exit Do
        End If
    Else

```

```

        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
            FormBusquedaInmo.TextBuscarPMaximo.Text =
            UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
            DataInmobiliaria.Recordset.FindFirst "[PrecioEuros] <= " &
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

            Do
                If Not (DataInmobiliaria.Recordset.NoMatch) Then

                    contador = contador + 1
                    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
                    vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                        DataInmobiliaria.Recordset.FindNext "[PrecioEuros] <= " &
                        Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
                    Else
                        Exit Do
                    End If
                Else
                    Exit Do
                End If
            Else
                DataInmobiliaria.Recordset.MoveFirst
            Exit Do

```

```

    End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
    MsgBox "NO HAY NINGUN CAMPO DE BUSQUEDA SELECCIONADO"

    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
  End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value
End Select 'FormBusquedaInmo.CheckBuscarHabitaciones.Value

Case 1: 'Esta seleccionada la opcion Direccion
  FormBusquedaInmo.TextBuscarDireccion.Text =
  UCase(Trim(FormBusquedaInmo.TextBuscarDireccion.Text))
  Select Case FormBusquedaInmo.CheckBuscarHabitaciones.Value
    Case 1 'Esta seleccionado el Nº de habitaciones
      FormBusquedaInmo.TextBuscarHabitaciones.Text =
      UCase(Trim(FormBusquedaInmo.TextBuscarHabitaciones.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

        Case 1 'OptionBuscarPMinimo.Value = 1
          FormBusquedaInmo.TextBuscarPMinimo.Text =
          UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
          Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
            Case 1: 'OptionBuscarPMaximo.Value = 1
              FormBusquedaInmo.TextBuscarPMaximo.Text =
              UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
              DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
              FormBusquedaInmo.TextBuscarDireccion.Text & "' & _

```

```

"AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" & "[PrecioEuros]
<= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
                "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text
& "AND" & "[PrecioEuros] >= " & _
                    Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & "AND" &
"[PrecioEuros] <= " & Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
        "Numero de fichas encontradas: " & contador
    contador = 0
    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
            "AND" & "[NHab] = " & FormBusquedaInmo.TextBuscarHabitaciones.Text &
"AND" & "[PrecioEuros] >= " & _
                Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

```

```

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
                "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" & "[PrecioEuros] >= " & _
                    Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
        "Numero de fichas encontradas: " & contador
    contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

    Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1
            FormBusquedaInmo.TextBuscarPMaximo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))

```

```

        DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & "AND" & _
        "[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND" &
"[PrecioEuros] <= '" & _
        Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & "AND" & _
            "[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text & "AND"
& "[PrecioEuros] <= '" & _
            Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If

Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

Case 0: 'OptionBuscarPMaximo.Value = 0
    DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
        "AND" & "[NHab] = '" & FormBusquedaInmo.TextBuscarHabitaciones.Text

```

```

Do
  If Not (DataInmobiliaria.Recordset.NoMatch) Then

    contador = contador + 1
    If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
      DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "' & _
      "AND" & "[NHab] = " &
FormBusquedaInmo.TextBuscarHabitaciones.Text
    Else
      Exit Do
    End If
  Else
    DataInmobiliaria.Recordset.MoveFirst
    Exit Do
  End If
Loop While Not (DataInmobiliaria.Recordset.EOF)

MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0

End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value

Case 0 'No esta seleccionado el N° de habitaciones
  Select Case FormBusquedaInmo.CheckBuscarPMinimo.Value

    Case 1 'OptionBuscarPMinimo.Value = 1
      FormBusquedaInmo.TextBuscarPMinimo.Text =
UCase(Trim(FormBusquedaInmo.TextBuscarPMinimo.Text))
      Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
        Case 1: 'OptionBuscarPMaximo.Value = 1

```

```

    FormBusquedaInmo.TextBuscarPMaximo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
        DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
        "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
        "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

Do
    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
            "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text) & _
            "AND" & "[PrecioEuros] <= " &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
        Else
            Exit Do
        End If
    Else
        DataInmobiliaria.Recordset.MoveFirst
        Exit Do
    End If
Loop While Not (DataInmobiliaria.Recordset.EOF)
MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
contador = 0
Case 0: 'OptionBuscarPMaximo.Value = 0

```



```

        DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
        "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)

    Do
        If Not (DataInmobiliaria.Recordset.NoMatch) Then

            contador = contador + 1
            If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
                "AND" & "[PrecioEuros] >= " &
Int(FormBusquedaInmo.TextBuscarPMinimo.Text)
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0
End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value

Case 0 'OptionBuscarPMinimo.Value = 0

Select Case FormBusquedaInmo.CheckBuscarPMaximo.Value
    Case 1: 'OptionBuscarPMaximo.Value = 1

```

```

    FormBusquedaInmo.TextBuscarPMaximo.Text =
    UCase(Trim(FormBusquedaInmo.TextBuscarPMaximo.Text))
        DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
        "AND" & "[PrecioEuros] <= '" &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)

    Do
        If Not (DataInmobiliaria.Recordset.NoMatch) Then

            contador = contador + 1

            If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
                DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'" & _
                "AND" & "[PrecioEuros] <= '" &
Int(FormBusquedaInmo.TextBuscarPMaximo.Text)
            Else
                Exit Do
            End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If

    Loop While Not (DataInmobiliaria.Recordset.EOF)
    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
    "Numero de fichas encontradas: " & contador
    contador = 0

    Case 0: 'OptionBuscarPMaximo.Value = 0
        DataInmobiliaria.Recordset.FindFirst "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'"

    Do

```

```

    If Not (DataInmobiliaria.Recordset.NoMatch) Then

        contador = contador + 1
        If MsgBox("Desea seguir buscando: ACEPTAR = SI,CANCELAR = NO",
vbOKCancel + vbInformation, "Busqueda") = vbOK Then
            DataInmobiliaria.Recordset.FindNext "[Direccion] LIKE '" &
FormBusquedaInmo.TextBuscarDireccion.Text & "'"
        Else
            Exit Do
        End If
        Else
            DataInmobiliaria.Recordset.MoveFirst
            Exit Do
        End If
    Loop While Not (DataInmobiliaria.Recordset.EOF)

    MsgBox "BUSQUEDA FINALIZADA" & Chr(13) & _
"Numero de fichas encontradas: " & contador
    contador = 0

    End Select 'FormBusquedaInmo.CheckBuscarPMaximo.Value
    End Select 'FormBusquedaInmo.CheckBuscarPMinimo.Value
    End Select 'FormBusquedaInmo.CheckBuscar Habitaciones.Value

    End Select 'FormBusquedaInmo.CheckBuscarDireccion.Value

    End Select 'FormBusquedaInmo.checkBuscarZona.Value
    End If 'Buscar_por_ID = True

Else ' sino se quiere realizar una busqueda

    FrameDatosProp1.Enabled = False
    FrameDatosProp2.Enabled = False

```

```
Frameld.Enabled = False
```

```
DataInmobiliaria.Caption = "Ficha " & DataInmobiliaria.Recordset.AbsolutePosition + 1 & " de " & _  
DataInmobiliaria.Recordset.RecordCount
```

```
'Si no hemos borrado el registro del formulario FormMasDatos,no hay que borrar
```

```
' el registro asociado en FormInmobiliaria.
```

```
'Buscamos el registro asociado y mostramos la ficha
```

```
If borrar_inmobiliaria = False Then
```

```
    DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_ficha
```

```
    'Si no encontramos ficha asociada creamos un nuevo registro asociado
```

```
    If DataInmobiliaria.Recordset.NoMatch = True Then
```

```
        MsgBox "No se ha encontrado ninguna ficha asociada,creando ficha asociada...." , ,  
"¡Atencion!"
```

```
        DataInmobiliaria.Recordset.AddNew
```

```
        DataInmobiliaria.Recordset("IdInmueble") = Id_ficha
```

```
        DataInmobiliaria.Recordset("FECHA CAPT") = Date
```

```
        DataInmobiliaria.Recordset.Update ' Actualiza la base de datos (obligatorio,hasta que no se  
pone no se hace nada)
```

```
        DataInmobiliaria.Recordset.MoveLast
```

```
    End If
```

```
'Si hemos borrado el registro del formulario FormMasDatos,hay que borrar
```

```
' el registro asociado en FormInmobiliaria.
```

```
Else
```

```
    borrar_inmobiliaria = False
```

```
    DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_borrar
```

```
    DataInmobiliaria.Recordset.Delete
```

```
    DataInmobiliaria.Refresh
```

```
    DataInmobiliaria.Recordset.Close
```

```
    FormInmobiliaria.Hide
```

```
    FormMasDatos.Show
```

```
End If
```

```
End If 'Busqueda_activada  
Busqueda_activada = False
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation  
End Sub
```

```
'Inicializamos el formulario
```

```
Private Sub Form_Initialize()
```

```
    TextPrecio = ""
```

```
    'cerrojo = True
```

```
    ComboProvincia.AddItem "OTRA"
```

```
    ComboProvincia.AddItem "VALENCIA"
```

```
    ComboProvincia.AddItem "CASTELLÓN"
```

```
    ComboProvincia.AddItem "ALICANTE"
```

```
    ComboModo.AddItem ""
```

```
    ComboModo.AddItem "VENTA"
```

```
    ComboModo.AddItem "ALQUILER"
```

```
    ComboModo.AddItem "TRASPASO"
```

```
    ComboNHabitaciones.AddItem "0"
```

```
    ComboNHabitaciones.AddItem "1"
```

```
    ComboNHabitaciones.AddItem "2"
```

```
    ComboNHabitaciones.AddItem "3"
```

```
    ComboNHabitaciones.AddItem "4"
```

```
    ComboNHabitaciones.AddItem "5"
```

```
    ComboNHabitaciones.AddItem "6"
```

```
    ComboNHabitaciones.AddItem "7"
```

```
    ComboNHabitaciones.AddItem "8"
```

```
End Sub
```

Private Sub Form_Load()

On Error Resume Next

DataInmobiliaria.Connect = "Access 2000;"

DataInmobiliaria.DatabaseName = App.Path & "\Inmobiliaria.mdb"

DataInmobiliaria.DefaultCursorType = 0 'Defaultcursor

DataInmobiliaria.DefaultType = 2 'Usejet

DataInmobiliaria.RecordsetType = 1 'Dynaset

DataInmobiliaria.RecordSource = "Inmueble"

DataInmobiliaria.Refresh

FrameDatosProp1.Enabled = False

FrameDatosProp2.Enabled = False

FrameId.Enabled = False

DataInmobiliaria.Caption = "Ficha " & DataInmobiliaria.Recordset.AbsolutePosition + 1 & " de " & _

DataInmobiliaria.Recordset.RecordCount

If borrar_inmobiliaria = False Then

DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_ficha

,

If DataInmobiliaria.Recordset.NoMatch = True Then

MsgBox "No se ha encontrado ninguna ficha asociada,creando ficha asociada....", , "¡Atencion!"

```
DataInmobiliaria.Recordset.AddNew
DataInmobiliaria.Recordset("IdInmueble") = Id_ficha
DataInmobiliaria.Recordset("FECHA CAPT") = Date
DataInmobiliaria.Recordset("Pais") = "ESPAÑA"
DataInmobiliaria.Recordset.Update ' Actualiza la base de datos (obligatorio,hasta que no se
pone no se hace nada)
DataInmobiliaria.Recordset.MoveLast
End If
Else
borrar_inmobiliaria = False

DataInmobiliaria.Recordset.Delete
DataInmobiliaria.Refresh
DataInmobiliaria.Recordset.MoveNext
If DataInmobiliaria.Recordset.EOF Then
DataInmobiliaria.Recordset.MoveLast
End If
DataInmobiliaria.Recordset.Close

Id_ficha = TextIdInm.Text 'Actualizo la variable Id_ficha
cerrar_inmobiliaria = True
Menu2MasDatos_Click

End If

If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation

End Sub
```

```
Private Sub Menu2Buscar_Click()
```

```
    FormBusquedaInmo.Show 1
```

```
End Sub
```

```
'Imprimo la pantalla
```

```
Private Sub Menu2Imprimir_Click()
```

```
    'Oculto partes del formulario
```

```
    DataInmobiliaria.Visible = False
```

```
    ToolbarInmobiliaria.Visible = False
```

```
    'Imprimo formulario
```

```
    FormInmobiliaria.PrintForm
```

```
    'Vuelvo a mostrar las partes del formulario
```

```
    DataInmobiliaria.Visible = True
```

```
    ToolbarInmobiliaria.Visible = True
```

```
End Sub
```

```
Private Sub Menu2Salir_Click()
```

```
    FormMasDatos.MenuSalir_Click
```

```
End Sub
```

```
Private Sub TextComCapt_Click()
```

```
    TextComCapt.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextComCapt.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextComCapt.SelLength = Len(TextComCapt.Text) 'Selecciona todo el texto
```

```
    TextComCapt.SelText = "0"
```

```
End Sub
```

```
Private Sub TextComCliente_Click()
```

```
    TextComCliente.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextComCliente.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextComCliente.SelLength = Len(TextComCliente.Text) 'Selecciona todo el texto
```

```
    TextComCliente.SelText = "0"
```


End Sub

Private Sub TextComProp_Click()

TextComProp.SetFocus 'Poner el cursor sobre TextClave

TextComProp.SelStart = 0 'Posiciona el cursor al comienzo del texto

TextComProp.SelLength = Len(TextComProp.Text) 'Selecciona todo el texto

TextComProp.SelText = "0"

End Sub

Private Sub TextCP_Click()

TextCP.SetFocus 'Poner el cursor sobre TextClave

TextCP.SelStart = 0 'Posiciona el cursor al comienzo del texto

TextCP.SelLength = Len(TextCP.Text) 'Selecciona todo el texto

TextCP.SelText = "0"

End Sub

Private Sub TextGComunidad_Click()

TextGComunidad.SetFocus 'Poner el cursor sobre TextClave

TextGComunidad.SelStart = 0 'Posiciona el cursor al comienzo del texto

TextGComunidad.SelLength = Len(TextGComunidad.Text) 'Selecciona todo el texto

TextGComunidad.SelText = "0"

End Sub

Private Sub TextMild_Click()

TextMild.SetFocus 'Poner el cursor sobre TextClave

TextMild.SelStart = 0 'Posiciona el cursor al comienzo del texto

TextMild.SelLength = Len(TextMild.Text) 'Selecciona todo el texto

TextMild.SelText = "0"

End Sub

Private Sub TextNFinca_Click()

TextNFinca.SetFocus 'Poner el cursor sobre TextClave
TextNFinca.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextNFinca.SelLength = Len(TextNFinca.Text) 'Selecciona todo el texto
TextNFinca.SelText = "0"

End Sub

Private Sub TextNRegistro_Click()

TextNRegistro.SetFocus 'Poner el cursor sobre TextClave
TextNRegistro.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextNRegistro.SelLength = Len(TextNRegistro.Text) 'Selecciona todo el texto
TextNRegistro.SelText = "0"

End Sub

Private Sub TextNumero_Click()

TextNumero.SetFocus 'Poner el cursor sobre TextClave
TextNumero.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextNumero.SelLength = Len(TextNumero.Text) 'Selecciona todo el texto
TextNumero.SelText = "0"

End Sub

Private Sub TextPGaraje_Click()

TextPGaraje.SetFocus 'Poner el cursor sobre TextClave
TextPGaraje.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextPGaraje.SelLength = Len(TextPGaraje.Text) 'Selecciona todo el texto
TextPGaraje.SelText = "0"

End Sub

```
Private Sub TextPiso_Click()
```

```
    TextPiso.SetFocus 'Poner el cursor sobre TextClave  
    TextPiso.SelStart = 0 'Posiciona el cursor al comienzo del texto  
    TextPiso.SelLength = Len(TextPiso.Text) 'Selecciona todo el texto  
    TextPiso.SelText = "0"
```

```
End Sub
```

```
Private Sub TextPrecio_Click()
```

```
    TextPrecio.SetFocus 'Poner el cursor sobre TextClave  
    TextPrecio.SelStart = 0 'Posiciona el cursor al comienzo del texto  
    TextPrecio.SelLength = Len(TextPrecio.Text) 'Selecciona todo el texto  
    TextPrecio.SelText = "0"  
    TextPrecioEuros.SetFocus 'Poner el cursor sobre TextClave  
    TextPrecioEuros.SelStart = 0 'Posiciona el cursor al comienzo del texto  
    TextPrecioEuros.SelLength = Len(TextPrecioEuros.Text) 'Selecciona todo el texto  
    TextPrecioEuros.SelText = "0"  
    TextPrecio.SetFocus
```

```
End Sub
```

```
Public Sub TextPrecio_KeyPress(KeyAscii As Integer)
```

```
    On Error Resume Next
```

```
    Select Case KeyAscii
```

```
        Case 48 To 57: Call Numero(KeyAscii - 48, 0)
```

```
        Case 44:
```

```
            If InStr(1, TextPrecio, ",") > 0 Then
```

```
                KeyAscii = 0
```

```
            Else: Coma 0 ' Pulsacion de la coma
```

```
            End If
```

```
        Case Else
```

```
            KeyAscii = 0
```

```
End Select
KeyAscii = 0
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

```
Private Sub TextTasacion_Click()
    TextTasacion.SetFocus 'Poner el cursor sobre TextClave
    TextTasacion.SelStart = 0 'Posiciona el cursor al comienzo del texto
    TextTasacion.SelLength = Len(TextTasacion.Text) 'Selecciona todo el texto
    TextTasacion.SelText = "0"
```

```
End Sub
```

```
Private Sub TextVsuelo_Click()
    TextVsuelo.SetFocus 'Poner el cursor sobre TextClave
    TextVsuelo.SelStart = 0 'Posiciona el cursor al comienzo del texto
    TextVsuelo.SelLength = Len(TextVsuelo.Text) 'Selecciona todo el texto
    TextVsuelo.SelText = "0"
```

```
End Sub
```

```
Private Sub TextVUrbano_Click()
    TextVUrbano.SetFocus 'Poner el cursor sobre TextClave
    TextVUrbano.SelStart = 0 'Posiciona el cursor al comienzo del texto
    TextVUrbano.SelLength = Len(TextVUrbano.Text) 'Selecciona todo el texto
    TextVUrbano.SelText = "0"
```

```
End Sub
```

```
Private Sub ToolbarInmobiliaria_ButtonClick(ByVal Button As ComctlLib.Button)
```

On Error Resume Next

Select Case Button.Index 'según el botón se ejecutan unas instrucciones

Case 1: Menu2Editar_Click

Case 2: Menu2Buscar_Click

Case 3: Menu2Imprimir_Click

Case 4: FormMasDatos.MenuCalcular_Click

Case 5: FormMasDatos.MenuSalir_Click

Case 6: Menu2MasDatos_Click

End Select

If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation

End Sub

Private Sub DataInmobiliaria_Validate(Action As Integer, Save As Integer)

Dim respuesta As Integer

Select Case Action

Case vbDataActionDelete, vbDataActionAddNew, vbDataActionFind, vbDataActionClose,
vbDataActionMoveFirst, vbDataActionMovePrevious, vbDataActionMoveNext, vbDataActionMoveLast

If Save = True Then

respuesta = MsgBox("¿Desea guardar los cambios?", vbYesNo, "Inmobifich")

If respuesta = vbNo Then

Save = False

DataInmobiliaria.UpdateControls 'Actualiza los cambios

End If

End If

End Select

End Sub

Private Sub DataInmobiliaria_Error(DataErr As Integer, Response As Integer)

```
'If DataErr = 3022 Then 'Error de clave principal duplicada
' MsgBox "Ingrese un Id de inmobiliaria único"
' TextIdInm.SetFocus
' Response = 0
'Else
  Response = 1 'Muestra un mensaje de error estándar

'End If
```

End Sub

```
Private Sub Menu2Editar_Click()
  On Error Resume Next
  FrameDatosProp1.Enabled = True
  FrameDatosProp2.Enabled = True
  FrameId.Enabled = True
  If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

End Sub

```
Private Sub Menu2MasDatos_Click()
  On Error Resume Next
  FrameDatosProp1.Enabled = False
  FrameDatosProp2.Enabled = False
  FrameId.Enabled = False
```

```
FormInmobiliaria.Hide 'Oculto el formulario FormInmobiliaria
FormMasDatos.Show 'Muestro el formualrio FormMasDatos
```

```
Id_ficha = TextIdInm.Text 'Id_ficha contiene el identificado del registro secundario  
cerrar_inmobiliaria = True 'Indicamos que hay que buscar el registro principal asociado a  
    ' al registro secundario
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

FORMMASDATOS

```
Option Explicit
```

```
' Cuando pulsamos sobre el botón "borrar" llamamos a la función
```

```
' Borrar_imagen
```

```
Private Sub CommandBorrarImagen_Click()
```

```
    Borrar_imagen
```

```
End Sub
```

```
'Evento que se produce cuando un registro se convierte en el
```

```
'registro actual
```

```
Private Sub DataInmobiliaria_Reposition()
```

```
    On Error Resume Next
```

```
'Para cada TextBox y Combobox quitamos los espacios delante y detrás y
```

```
' convertimos a mayúsculas.
```

```
TextAguaCal.Text = UCase(Trim(TextAguaCal.Text))
```

```
TextAñosFinca.Text = UCase(Trim(TextAñosFinca.Text))
```

```
TextArmarios.Text = UCase(Trim(TextArmarios.Text))
```

```
TextBalcon.Text = UCase(Trim(TextBalcon.Text))
```

```
TextCalefaccion.Text = UCase(Trim(TextCalefaccion.Text))
```

```
TextEscaparate.Text = UCase(Trim(TextEscaparate.Text))
```

```
TextOtros1.Text = UCase(Trim(TextOtros1.Text))
TextOtros2.Text = UCase(Trim(TextOtros2.Text))
TextOtros3.Text = UCase(Trim(TextOtros3.Text))
TextOtros4.Text = UCase(Trim(TextOtros4.Text))
TextOtros5.Text = UCase(Trim(TextOtros5.Text))
TextOtros6.Text = UCase(Trim(TextOtros6.Text))
TextPiscina.Text = UCase(Trim(TextPiscina.Text))
TextTerraza.Text = UCase(Trim(TextTerraza.Text))
TextTrastero.Text = UCase(Trim(TextTrastero.Text))
ComboAscensor.Text = UCase(Trim(ComboAscensor.Text))
ComboComunicado.Text = UCase(Trim(ComboComunicado.Text))
ComboGaraje.Text = UCase(Trim(ComboGaraje.Text))
'Deshabilitamos los frames (no se puede modificar)
FrameDatosDescripcion1.Enabled = False
FrameDatosDescripcion2.Enabled = False
'Indicamos el numero de registro en el que estamos situados del total de registros
DataInmobiliaria.Caption = "Ficha " & DataInmobiliaria.Recordset.AbsolutePosition + 1 & " de " & _
DataInmobiliaria.Recordset.RecordCount
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
'Evento que ocurre cuando el formulario FormMasDatos se convierte
```

```
' en la ventana activa
```

```
Private Sub Form_Activate()
```

```
On Error Resume Next
```

```
'Escribimos varias propiedades del control Data DataInmobiliaria
```

```
' en tiempo de ejecución
```

```
DataInmobiliaria.Connect = "Access 2000;"
```



```
DataInmobiliaria.DatabaseName = App.Path & "\Inmobiliaria.mdb"  
DataInmobiliaria.DefaultCursorType = 0 'Defaultcursor  
DataInmobiliaria.DefaultType = 2 'Usejet  
DataInmobiliaria.RecordsetType = 1 'Dynaset. Recordset actualizable  
DataInmobiliaria.RecordSource = "Descripcion"  
DataInmobiliaria.Refresh  
DataInmobiliaria.Recordset.MoveLast  
DataInmobiliaria.Recordset.MoveFirst
```

```
FrameDatosDescripcion1.Enabled = False  
FrameDatosDescripcion2.Enabled = False
```

```
'Si cerrar_inmobiliaria es cierto indica que venimos de el  
'formulario FormInmobiliaria.
```

```
If cerrar_inmobiliaria = True Then
```

```
    cerrar_inmobiliaria = False
```

```
    'Debemos buscar el registro principal asociado
```

```
    DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_ficha
```

```
    If DataInmobiliaria.Recordset.NoMatch = True Then
```

```
        MsgBox "No se ha encontrado ninguna ficha asociada", , "TABLA Inmueble"
```

```
    End If
```

```
End If
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
' Es el primer evento que se ejecuta cuando se visualiza un formulario
```

```
' El orden de eventos es: Initialize, Load, Activate, Paint
```

Private Sub Form_Initialize()

cerrar_inmobiliaria = False

borrar_inmobiliaria = False

Id_ficha = ""

'Inicializamos objetos

ComboBaños.AddItem "0"

ComboBaños.AddItem "1"

ComboBaños.AddItem "2"

ComboBaños.AddItem "3"

ComboBaños.AddItem "4"

ComboAseos.AddItem "0"

ComboAseos.AddItem "1"

ComboAseos.AddItem "2"

ComboAseos.AddItem "3"

ComboAseos.AddItem "4"

ComboCocina.AddItem ""

ComboCocina.AddItem "OFFICE"

ComboCocina.AddItem "AMERICANA"

ComboCocina.AddItem "NORMAL"

ComboCocina.Text = ""

ComboNAlturas.AddItem "0"

ComboNAlturas.AddItem "1"

ComboNAlturas.AddItem "2"

ComboNAlturas.AddItem "3"

ComboNAlturas.AddItem "4"

ComboNAlturas.AddItem "5"

ComboNAlturas.AddItem "6"

```
ComboOrientacion.AddItem "NORTE"  
ComboOrientacion.AddItem "SUR"  
ComboOrientacion.AddItem "ESTE"  
ComboOrientacion.AddItem "OESTE"
```

```
End Sub
```

```
'Evento que se produce al cargar el formulario FormMasDatos
```

```
Private Sub Form_Load()
```

```
    On Error Resume Next
```

```
FrameDatosDescripcion1.Enabled = False
```

```
FrameDatosDescripcion2.Enabled = False
```

```
DataInmobiliaria.Connect = "Access 2000;"
```

```
DataInmobiliaria.DatabaseName = App.Path & "\Inmobiliaria.mdb"
```

```
DataInmobiliaria.DefaultCursorType = 0 'Defaultcursor
```

```
DataInmobiliaria.DefaultType = 2 'Usejet
```

```
DataInmobiliaria.RecordsetType = 1 'Dynaset
```

```
DataInmobiliaria.RecordSource = "Descripcion"
```

```
DataInmobiliaria.Refresh
```

```
'Nos movemos al último registro
```

```
DataInmobiliaria.Recordset.MoveLast
```

```
'Nos movemos al priemr registro
```

```
DataInmobiliaria.Recordset.MoveFirst
```

```
If cerrar_inmobiliaria = True Then
```

```
    cerrar_inmobiliaria = False
```

```
DataInmobiliaria.Recordset.FindFirst "[IdInmueble] = " & Id_ficha
If DataInmobiliaria.Recordset.NoMatch = True Then
    MsgBox "No se ha encontrado ninguna ficha asociada", , "TABLA Descripcion"
End If
End If
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
Private Sub Menu2Salir_Click()
    MenuSalir_Click
End Sub
```

'Si pulsamos la opción Borrar del menu, eliminamos el registro actual

```
Private Sub MenuBorrar_Click()
```

```
    On Error Resume Next
```

'Ponemos en la variable Id_borrar el identifiacdor de la ficha actual

```
Id_borrar = TextIDDescripcion.Text
```

```
If MsgBox("¿DESEA BORRAR LA FICHA ACTUAL?", vbOKCancel + vbExclamation, "Borrar ficha actual") = vbOK Then
```

```
    DataInmobiliaria.Recordset.Delete 'Borramos registro
```

```
    DataInmobiliaria.Refresh 'Actualizamos Recordset
```

```
End If
```

```
borrar_inmobiliaria = True 'Indicamos que hay que borrar el registro asociado
```

```
MenuInmobiliaria_Click 'Cargamos el formulario FormInmobiliaria
```

```
If Err.Number > 0 Then MsgBox Err.Description, vbExclamation
```

```
End Sub
```

'Si pulsamos la opción Fotos del menu,mostramos la ventana para visualizar fotografías

```
Private Sub MenuFotografias_Click()
```

```
FormFotografias.Show 0
```

```
End Sub
```

```
'Si pulsamos la opción Imprimir del menu,imprimimos la pantalla con la ficha
```

```
Private Sub MenuImprimir_Click()
```

```
    'Ocultamos elementos del formulario
```

```
    ToolbarDescripcion.Visible = False
```

```
    DataInmobiliaria.Visible = False
```

```
    FrameFotos.Visible = False
```

```
    'Imprimimos formulario
```

```
    FormMasDatos.PrintForm
```

```
    'Volvemos a mostrar elementos del formulario
```

```
    ToolbarDescripcion.Visible = True
```

```
    DataInmobiliaria.Visible = True
```

```
    FrameFotos.Visible = True
```

```
End Sub
```

```
'Si hacemos doble click sobre el objeto OLE1 se nos muestra
```

```
'una ventana de diálogo para insertar objeto
```

```
Private Sub OLE1_DblClick()
```

```
    OLE1.InsertObjDlg
```

```
End Sub
```

```
Private Sub OLE2_DblClick()
```

```
    OLE2.InsertObjDlg
```

```
End Sub
```

```
Private Sub OLE3_DblClick()
```

```
    OLE3.InsertObjDlg
```

```
End Sub
```

```
Private Sub OLE4_DblClick()
```

```
    OLE4.InsertObjDlg
```

End Sub

Private Sub OLE5_DblClick()

OLE5.InsertObjDlg

End Sub

Private Sub OLE6_DblClick()

OLE6.InsertObjDlg

End Sub

'Cuando hacemos click en el TextBox "TextAñosFinca" borramos lo que hay
' y ponemos un cero

Private Sub TextAñosFinca_Click()

TextAñosFinca.SetFocus 'Poner el cursor sobre TextClave

TextAñosFinca.SelStart = 0 'Posiciona el cursor al comienzo del texto

TextAñosFinca.SelLength = Len(TextAñosFinca.Text) 'Selecciona todo el texto

TextAñosFinca.SelText = "0"

End Sub

'Idem

Private Sub TextGtos_Click()

TextGtos.SetFocus 'Poner el cursor sobre TextClave

TextGtos.SelStart = 0 'Posiciona el cursor al comienzo del texto

TextGtos.SelLength = Len(TextGtos.Text) 'Selecciona todo el texto

TextGtos.SelText = "0"

TextGtosEuros.SetFocus 'Poner el cursor sobre TextClave

TextGtosEuros.SelStart = 0 'Posiciona el cursor al comienzo del texto

TextGtosEuros.SelLength = Len(TextGtosEuros.Text) 'Selecciona todo el texto

TextGtosEuros.SelText = "0"

TextGtos.SetFocus

End Sub

'Idem

```
Private Sub TextMSalCom_Click()
```

```
    TextMSalCom.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextMSalCom.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextMSalCom.SelLength = Len(TextMSalCom.Text) 'Selecciona todo el texto
```

```
    TextMSalCom.SelText = "0"
```

```
End Sub
```

```
Private Sub TextMVestibulo_Click()
```

```
    TextMVestibulo.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextMVestibulo.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextMVestibulo.SelLength = Len(TextMVestibulo.Text) 'Selecciona todo el texto
```

```
    TextMVestibulo.SelText = "0"
```

```
End Sub
```

```
Private Sub TextMVestidor_Click()
```

```
    TextMVestidor.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextMVestidor.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextMVestidor.SelLength = Len(TextMVestidor.Text) 'Selecciona todo el texto
```

```
    TextMVestidor.SelText = "0"
```

```
End Sub
```

```
Private Sub TextNElesCom_Click()
```

```
    TextNElesCom.SetFocus 'Poner el cursor sobre TextClave
```

```
    TextNElesCom.SelStart = 0 'Posiciona el cursor al comienzo del texto
```

```
    TextNElesCom.SelLength = Len(TextNElesCom.Text) 'Selecciona todo el texto
```

```
    TextNElesCom.SelText = "0"
```

```
End Sub
```

```
Private Sub TextNPFach_Click()
```

```
TextNPFach.SetFocus 'Poner el cursor sobre TextClave
TextNPFach.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextNPFach.SelLength = Len(TextNPFach.Text) 'Selecciona todo el texto
TextNPFach.SelText = "0"
```

```
End Sub
```

```
Private Sub TextPtasM2_Click()
```

```
TextPtasM2.SetFocus 'Poner el cursor sobre TextClave
TextPtasM2.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextPtasM2.SelLength = Len(TextPtasM2.Text) 'Selecciona todo el texto
TextPtasM2.SelText = "0"
TextEurosM2.SetFocus 'Poner el cursor sobre TextClave
TextEurosM2.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextEurosM2.SelLength = Len(TextEurosM2.Text) 'Selecciona todo el texto
TextEurosM2.SelText = "0"
TextPtasM2.SetFocus
```

```
End Sub
```

'Cuando pulsamos una tecla en el TextBox "TextPtasM2" comprobamos si

' la tecla pulsada es un número, una coma o cualquier otra tecla.

' Si es un número lo escribimos en el Texbox

```
Private Sub TextPtasM2_KeyPress(KeyAscii As Integer)
```

```
On Error Resume Next
```

```
Select Case KeyAscii
```

```
Case 48 To 57: Call Numero(KeyAscii - 48, 1)
```

```
Case 44:
```

```
  If InStr(1, TextPtasM2, ",") > 0 Then
```

```
    KeyAscii = 0
```

```
  Else: Coma 1 ' Pulsacion de la coma
```

```
  End If
```

```
Case Else
```



```
        KeyAscii = 0
    End Select
    KeyAscii = 0
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

Private Sub TextGtos_KeyPress(KeyAscii As Integer)

    On Error Resume Next
    Select Case KeyAscii
        Case 48 To 57: Call Numero(KeyAscii - 48, 2)
        Case 44:
            If InStr(1, TextGtos, ",") > 0 Then
                KeyAscii = 0
            Else: Coma 2 ' Pulsacion de la coma
            End If
        Case Else
            KeyAscii = 0
    End Select
    KeyAscii = 0
    If Err.Number > 0 Then MsgBox Err.Description, vbExclamation

End Sub

'Evento que se produce cuando el Textbox "TextIDDescripcion" cambia
Private Sub TextIDDescripcion_Change()
    On Error Resume Next
    'Escribimos el identificador del registro actual en la variable global
    'Id_ficha_inmobiliaria para poder pasar posteriormente a su ficha asociada
    'Id_ficha_inmobiliaria = TextIDDescripcion.Text
    If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

End Sub

Private Sub TextSConst_Click()

TextSConst.SetFocus 'Poner el cursor sobre TextClave
TextSConst.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextSConst.SelLength = Len(TextSConst.Text) 'Selecciona todo el texto
TextSConst.SelText = "0"

End Sub

Private Sub TextSParcela_Click()

TextSParcela.SetFocus 'Poner el cursor sobre TextClave
TextSParcela.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextSParcela.SelLength = Len(TextSParcela.Text) 'Selecciona todo el texto
TextSParcela.SelText = "0"

End Sub

Private Sub TextSUtil_Click()

TextSUtil.SetFocus 'Poner el cursor sobre TextClave
TextSUtil.SelStart = 0 'Posiciona el cursor al comienzo del texto
TextSUtil.SelLength = Len(TextSUtil.Text) 'Selecciona todo el texto
TextSUtil.SelText = "0"

End Sub

'Bara de herramientas. Según el botón que se pulse se ejecutará una función
'u otra.

Private Sub ToolbarDescripcion_ButtonClick(ByVal Button As ComctlLib.Button)

On Error Resume Next

Select Case Button.Index 'según el botón se ejecutan unas instrucciones

Case 1: MenuEditar_Click

Case 2: MenuNueva_Click
Case 3: MenuCalcular_Click
Case 4: MenuBorrar_Click
Case 5: MenuImprimir_Click
Case 6: MenuSalir_Click
Case 7: MenuAcerca_Click
Case 8: MenuFotografias_Click
Case 9: MenuInmobiliaria_Click

End Select

If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation

End Sub

'Evento que se produce justo antes de que Visual Basic escriba en la base de datos

'los cambios procedentes de los controles enlazados y de que vuelva a

'colocar el puntero del registro actual en otro registro de la base de datos.

'Se utiliza para que el usuario confirme los cambios realizados

Private Sub DataInmobiliaria_Validate(Action As Integer, Save As Integer)

Dim respuesta As Integer

Select Case Action

Case vbDataActionDelete, vbDataActionAddNew, vbDataActionFind, vbDataActionClose,
vbDataActionMoveFirst, vbDataActionMovePrevious, vbDataActionMoveNext, vbDataActionMoveLast

If Save = True Then

respuesta = MsgBox("¿Desea guardar los cambios?", vbYesNo, "Inmobifich")

If respuesta = vbNo Then

```
        Save = False
        DataInmobiliaria.UpdateControls 'Actualiza los cambios
    End If
End If
End Select
```

```
End Sub
```

```
Private Sub DataInmobiliaria_Error(DataErr As Integer, Response As Integer)
```

```
    'If DataErr = 3022 Then 'Error de clave principal duplicada
```

```
        ' MsgBox "Ingrese un Id de inmobiliaria único"
```

```
        ' TextIdInm.SetFocus
```

```
        ' Response = 0
```

```
    'Else
```

```
        Response = 1 'Muestra un mensaje de error estándar
```

```
    'End If
```

```
End Sub
```

```
'Si pulsamos la opción Acerca del menu, mostramos la ventana de Acerca de..
```

```
Private Sub MenuAcerca_Click()
```

```
    FormAcerca.Show 0
```

```
End Sub
```

```
'Si pulsamos la opción Calcular del menu, mostramos la ventana que contiene
```

```
'la calculadora
```

```
Public Sub MenuCalcular_Click()
```

```
    On Error Resume Next
```

```
    FormCalc.Show 0
```

```
    FormCalc.SetFocus 'Mueve el enfoque al formulario FormCalc
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
'Si pulsamos la opción Editar del menu, habilitamos los frames (se puede modificar)
```

```
Private Sub MenuEditar_Click()
```

```
    On Error Resume Next
```

```
    'Habilito los frames
```

```
    FrameDatosDescripcion1.Enabled = True
```

```
    FrameDatosDescripcion2.Enabled = True
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
'Si pulsamos la opción Ficha asociada del menu, cargamos el formulario
```

```
'FormInmobiliaria
```

```
Private Sub MenuInmobiliaria_Click()
```

```
    On Error Resume Next
```

```
'Almacenamos en la variable global Id_ficha el identificador del registro actual
```

```
Id_ficha = TextIDDescripcion.Text
```

```
FormMasDatos.Hide 'Ocultamos formualio FormMasDatos
```

```
FormInmobiliaria.Show 0 'permite cambiar a otro formulario sin que el primero sea cerrado
```

```
FormInmobiliaria.SetFocus
```

```
If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation
```

```
End Sub
```

```
Private Sub MenuNueva_Click()
```

On Error Resume Next

DataInmobiliaria.Recordset.AddNew

DataInmobiliaria.Recordset.Update ' Actualiza la base de datos (obligatorio,hasta que no se pone no se hace nada)

DataInmobiliaria.Recordset.MoveLast

If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation

End Sub

'Declaro el procedimiento MenuSalir_Click () como público

'Salimos del programa

Public Sub MenuSalir_Click()

On Error Resume Next

If MsgBox("¿Seguro que quiere salir del programa?", vbOKCancel + vbExclamation, "Confirmación") = vbOK Then

End

End If

If Err.Number > 0 Then MsgBox Err.descripcion, vbExclamation

End Sub