

Implementación de SaceTIC

La materialización de lo diseñado es lograda al implementarse en el ordenador. Es necesario por una parte del uso de un gestor de Base de Datos para poder implementar el diseño de esta y de su seguridad y por otra de un Lenguaje de Programación para implementar el diseño de la Interfaz Usuario. Habría que preguntarse ¿Qué gestor de Base de Datos y Lenguaje de Programación elegir? Al dar la respuesta debemos tener en cuenta aspectos como:

- ❖ Que cubran nuestras necesidades y con los requerimientos de hardware y de software donde será empleada la aplicación.
- ❖ Que disponga de herramientas que permitan un rápido desarrollo y utilicen entornos visuales.
- ❖ Que posibiliten emigrar de ser necesario a otras plataformas y gestores
- ❖ Que dispongamos suficiente documentación.
- ❖ De ser posible que tengamos la mayor experiencia de su uso.

Para implementar SaceTIC fue escogido como gestor **MSAccess** y como Lenguaje de Programación el **Visual Basic para Aplicaciones** lenguaje embebido en todas las aplicaciones de Office entre ellas MSAccess. Estas elecciones son justificadas en el capítulo mediante una breve caracterización de cada uno con uso de la bibliografía consultada al respecto.

El capítulo inicia con la implementación con el gestor antes mencionado del Modelo Lógico Extendido de SaceTIC (campos añadidos a tablas del Modelo Lógico y nuevas tablas incorporadas) el que se obtuvo a partir del diseño realizado de la Base de Datos, detallándose las posibilidades ofrecidas por este gestor para realizarse la implementación como son las de poder garantizar las Integridades Referencial y de Llave, las operaciones de borrado y actualizado en cascada de los datos, sin tener que controlar dichos aspectos mediante código, los tipos de



datos que el gestor brinda los que cubren a plenitud las necesidades de nuestro problema y las facilidades con que en el pueden ser creadas las consultas y examinar el código SQL de estas.

En el segundo epígrafe se abordan las características por las cuales se seleccionó Visual Basic para Aplicaciones para implementar la Interfaz de Usuario de SaceTIC, como el hecho de ser un entorno de programación completamente visual, poder con la utilización de controles ActiveX aumentar las potencialidades del entorno para desarrollar de aplicaciones , las potencialidades brindadas por el como la manipulación de errores, el cambio de las propiedades de los objetos en tiempo de ejecución, uso de estructuras repetitivas y selectivas, su potente depurador de errores y editor, etc.

En el último epígrafe se especifica como se implemento el diseño de la seguridad de la Base de Datos de SaceTIC con el gestor de la Microsoft el que permite otorgar los permisos a nivel de los objetos de la Base de Datos, además el de crear las cuentas de grupo y de usuario previstas para SaceTIC .



III.1.- Implementación de la Base de Datos de SaceTIC.

Para facilitar las tareas de administración de los datos y acelerar el desarrollo de la aplicación se hace necesario realizar una selección adecuada del Sistema Gestor de Bases de Datos (SGBD) en este caso de MSAccess.

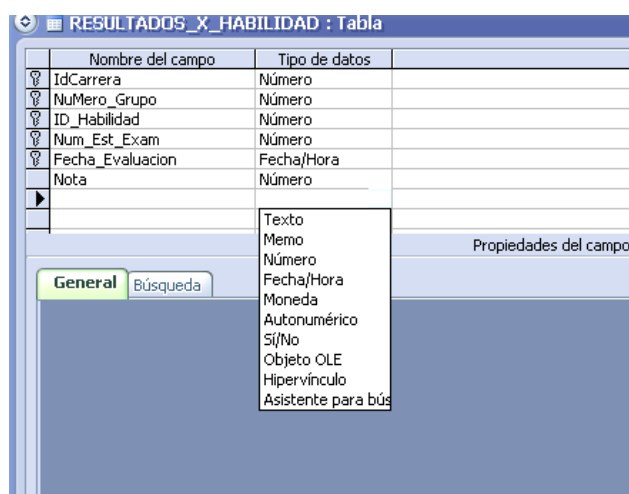
Teniendo en cuenta la literatura referenciada podemos destacar que MSAccess es una buena opción para la pequeña y mediana empresa. Es muy visual, siguiendo los pasos sencillos del Wizard se pueden crear interfaces para la entrada y modificación de datos de una tabla, algo mucho menos engorroso que la introducción de sentencias SQL.

También se pueden crear programas en Visual Basic, Visual C++ o Delphi y crear el entorno que se desee. Además de ello se destacan características específicas que fueron utilizadas en la implementación de la Base de Datos tales como: **[Polodori, 2005]**

- ❖ Permitir el ingreso de datos de tipos: Numéricos, Texto, Fecha, Sí/No, OLE, Moneda, Memo, con los cuales se satisfacen los requerimientos de SaceTIC.



Epígrafe III.1.-Implementación de la Base de Datos de SaceTIC



The screenshot shows the Microsoft Access interface for a table named 'RESULTADOS_X_HABILIDAD'. A list of data types is displayed, including 'Número', 'Fecha/Hora', 'Texto', 'Memo', 'Moneda', 'Autonumérico', 'Sí/No', 'Objeto OLE', 'Hipervínculo', and 'Asistente para búsqueda'. The 'General' tab is selected in the 'Propiedades del campo' pane.

Nombre del campo	Tipo de datos
IdCarrera	Número
NuMero_Grupo	Número
ID_Habilidad	Número
Num_Est_Exam	Número
Fecha_Evaluacion	Fecha/Hora
Nota	Número

Figura III.1.1.- Tipos de Datos de MSAccess utilizados en la tabla Resultados X Habilidad de SaceTIC.

- ❖ Posibilitar realizar consultas directas a las tablas contenidas mediante instrucciones SQL o mediante su potente generador de consultas que construye las mismas de manera visual con solo arrastrar las tablas a usar y los campos a utilizar de estas tablas. Brindando la posibilidad de utilización de gran variedad de consultas como las de selección, empleadas en la implementación de SaceTIC (Figura III.1.1), actualización, de datos anexados y de eliminación.



Epígrafe III.1.-Implementación de la Base de Datos de SaceTIC

Campo:	Nivel	Cant: Cantidad	IdFacultad	Ano
Tabla:	CantXNivel	CantXNivel	CantXNivel	CantXNivel
Total:	Agrupar por	Suma	Dónde	Dónde
Orden:				
Mostrar:	☒	☒	☐	☐
Criterios:			VDevol()	DevolNoAnoEsc()
o:				

Figura III.1.2.- Ejemplo de una consulta de selección de SaceTIC

```
SELECT CantXNivel.Nivel, Sum(CantXNivel.Cantidad) AS Cant
FROM CantXNivel
WHERE (((CantXNivel.IdFacultad)=VDevol()) AND ((CantXNivel.Ano)=DevolNoAnoEsc()))
GROUP BY CantXNivel.Nivel;
```

Figura III.1.3.- Vista en SQL de la consulta de selección de la Figura III.1.2

- ❖ Garantiza por si mismo las integridades de llave y referencial así como las operaciones de actualizado y borrado en cascada, todas ellas utilizadas en SaceTIC, facilitándose así mucho más el trabajo.



Epígrafe III.1.-Implementación de la Base de Datos de SaceTIC

RESULTADOS_X_HABILIDAD : Tabla

Nombre del campo	Tipo de datos	Descripción
IdCarrera	Número	
NuMero_Grupo	Número	
ID_Habilidad	Número	
Num_Est_Exam	Número	
Fecha_Evaluacion	Fecha/Hora	
Nota	Número	

Índices: RESULTADOS_X_HABILIDAD

Nombre de índice	Nombre del campo	Orden
PrimaryKey	IdCarrera	Ascendente
	NuMero_Grupo	Ascendente
	ID_Habilidad	Ascendente
	Num_Est_Exam	Ascendente
	Fecha_Evaluacion	Ascendente

Propiedades del índice

Principal	Sí
Única	Sí
Ignorar Nulos	No

No se puede modificar esta pro tablas vinculadas.

Figura III.1.4.- Empleo de Integridad de Llave en la tabla Resultado X Habilidad de SaceTIC.

Tabla o consulta: GRUPO Tabla o consulta relacionada: RESULTADOS_X_HABILIDAD

Tabla o consulta	Campos
GRUPO	NUMERO_GRUPO
GRUPO	ID_CARRERA
RESULTADOS_X_HABILIDAD	NuMero_Grupo
RESULTADOS_X_HABILIDAD	IdCarrera

☒ Exigir integridad referencial

☒ Actualizar en cascada los campos relacionados

☒ Eliminar en cascada los registros relacionados

Tipo de relación: Uno a varios

Figura III.1.5.- Empleo de la Integridad Referencial y de las operaciones de actualizado y eliminado en cascada en SaceTIC.



Epígrafe III.1.-Implementación de la Base de Datos de SaceTIC

- ❖ Brindar diferentes niveles y métodos de protección de los datos, entre ellos **Seguridad a nivel de objetos** uno de los modos más fuerte y flexible de protección de una aplicación, el mismo fue utilizado para implementar la seguridad de la Base de Datos de SaceTIC. En el epígrafe 3 del presente capítulo puede ver la implementación del diseño de la seguridad de SaceTIC utilizando las herramientas ofrecidas por MSAccess para tales fines.

En el Capítulo I al caracterizar la Estrategia del empleo de las TIC en la UPR fue creado su Modelo Conceptual de las TIC el que puede ver en la Figura I.2.1 del mencionado capítulo. Este modelo se obtuvo con la herramienta CASE Power Designer (Ver Anexo 2), del que se generó el Modelo de Datos o Modelo Lógico de los Datos, el que puede ver en la figura III.1.6 implementado con uso del gestor seleccionado y las tablas añadidas al Modelo Lógico de Datos, para mayor detalle de estas tablas vea el (Anexo 3) del “Modelo Lógico de Datos Extendido”.

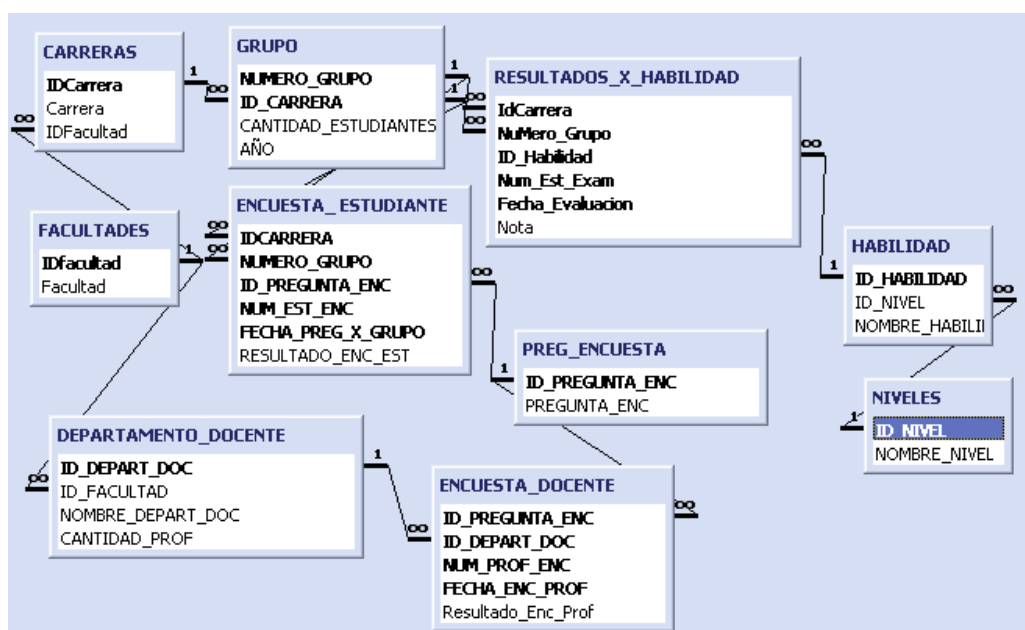


Figura III.1.6.- Modelo Lógico de los Datos de SaceTIC implementado.



III.2.- Implementación de la Interfaz Usuario de SaceTIC.

La opción de la herramienta a elegir para la implementación del diseño de la interfaz usuario de un sistema es una tarea a veces compleja, lógicamente el factor de mayor peso deben ser las necesidades de desarrollo que se tengan pesando las características tanto de Hardware como de software del lugar donde se le dará uso a la aplicación.

En el caso del sistema SaceTIC, que es una aplicación de gestión, se seleccionaría una herramienta de desarrollo que permita emplear más tiempo en el análisis, planificación y diseño y menos en tareas como la creación de interfaces, gestión de mensajes del sistema, etc. Por su buena preparación para el trabajo con Bases de Datos se podría elegir entre Visual Basic, Delphi y C++ Builder y en particular **Visual Basic para Aplicaciones** embebido en el gestor aprovechando a su vez los objetos brindados por este que permiten realizar un diseño visual de la interfaz usuario con rapidez y sencillez.

A continuación se destacan con ejemplos concretos algunas de las características empleadas de MSAccess y del lenguaje "Visual Basic para Aplicaciones" en la implementación de la Interfaz de Usuario de SaceTIC.

En la Implementación de SaceTIC se utilizó los objetos formularios e informes de MSAccess así como los controles:

- Etiqueta
- Cuadro de Texto
- Cuadro Combinado
- Botón de Comando
- Botón de Radio
- Grupo de Opciones
- SubFormularios
- ActiveX



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC

➤ Data Picker

Detalle

Fecha Evaluación: 01-07-06

Independiente

Indeper

Facultad: Independiente

Carrera: Independiente

Grupo: dependient

No Estudiante: dependient

Lista de Facultades

Lista de Carreras

Lista de Grupos

Lista de Habilidades

Guardar

Cancelar

Salir

Llenar Evaluación

Detalle

Habilidades:

ID_Habilidad

Nota

NOTA

IDCARRERA

NUMERO_GRUPO

FECHA_EVALUACION

Num_Est_Exam

Fig. III.2.1.-Ejemplo de controles usados en un objeto formulario para Implementar la Interfaz-Usuario del Caso de Uso: Registrar Evaluación de la Estrategia

Para coordinar el trabajo de los objetos y controles utilizados se realizó dando respuesta a los eventos de interés ocurrido en estos.



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC

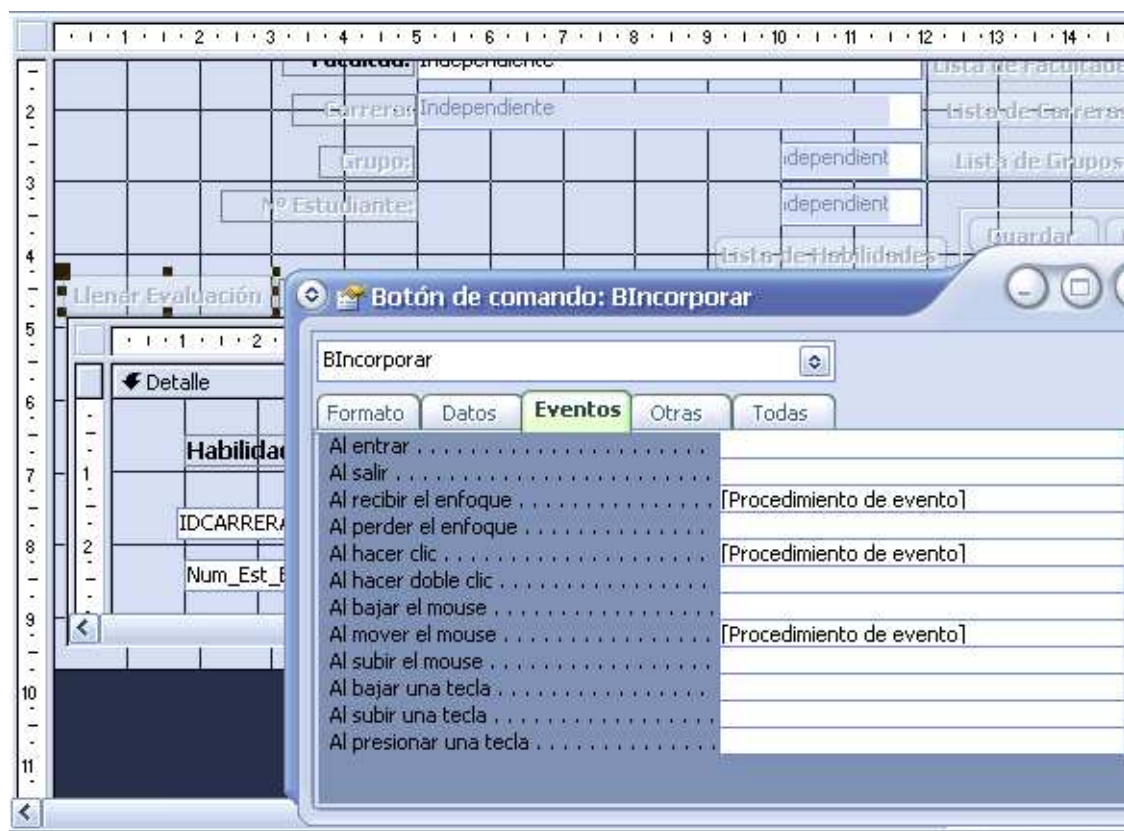


Fig. III.2.2.-Eventos a los que se les da respuesta mediante código Visual Basic para Aplicaciones ocurridos en el control BIncorporar del formulario “Registrar Evaluación de la Estrategia”.

El Lenguaje Visual Basic para Aplicaciones (Lenguaje Residente en todas las aplicaciones de Microsoft Office) es una de las posibilidades para dar respuesta a los eventos de interés, la otra son las macros. Resumiendo lo consultado sobre este Lenguaje de Programación Visual Basic para Aplicaciones se tiene:



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC

- Posee un potente editor y depurador de código

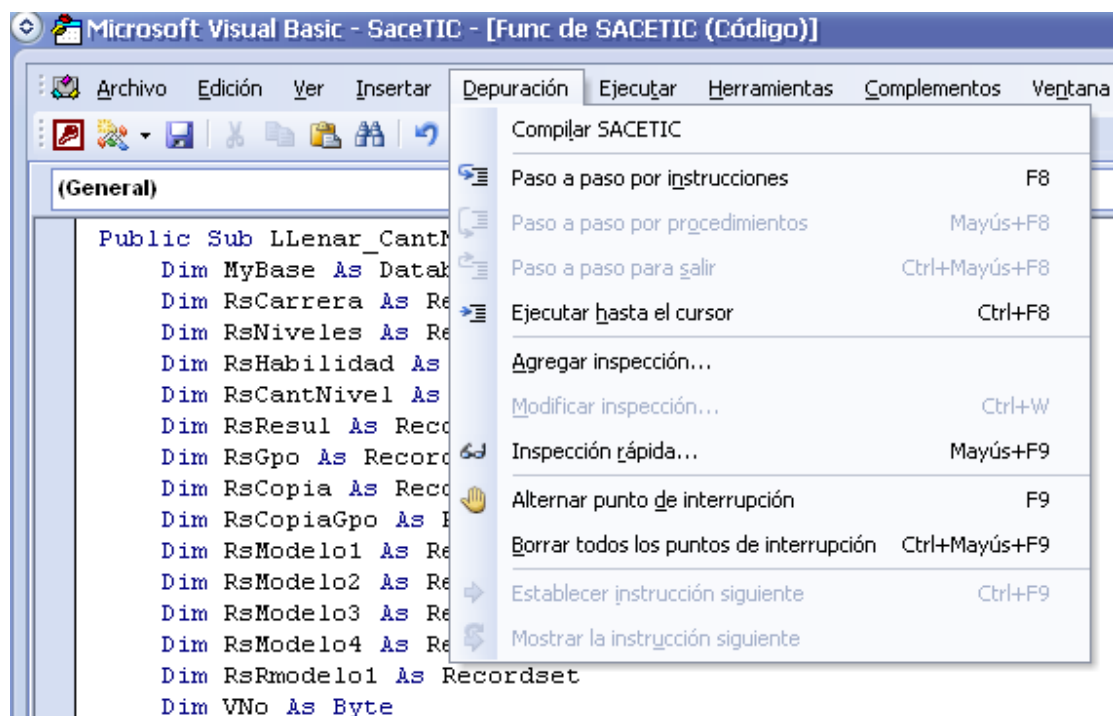


Fig. III.2.3.- Ventana de Edición-Depuración de código de Visual Basic

- Permite la creación de Sub y Function

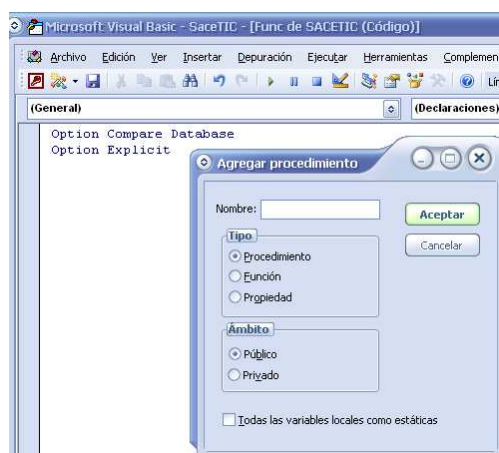
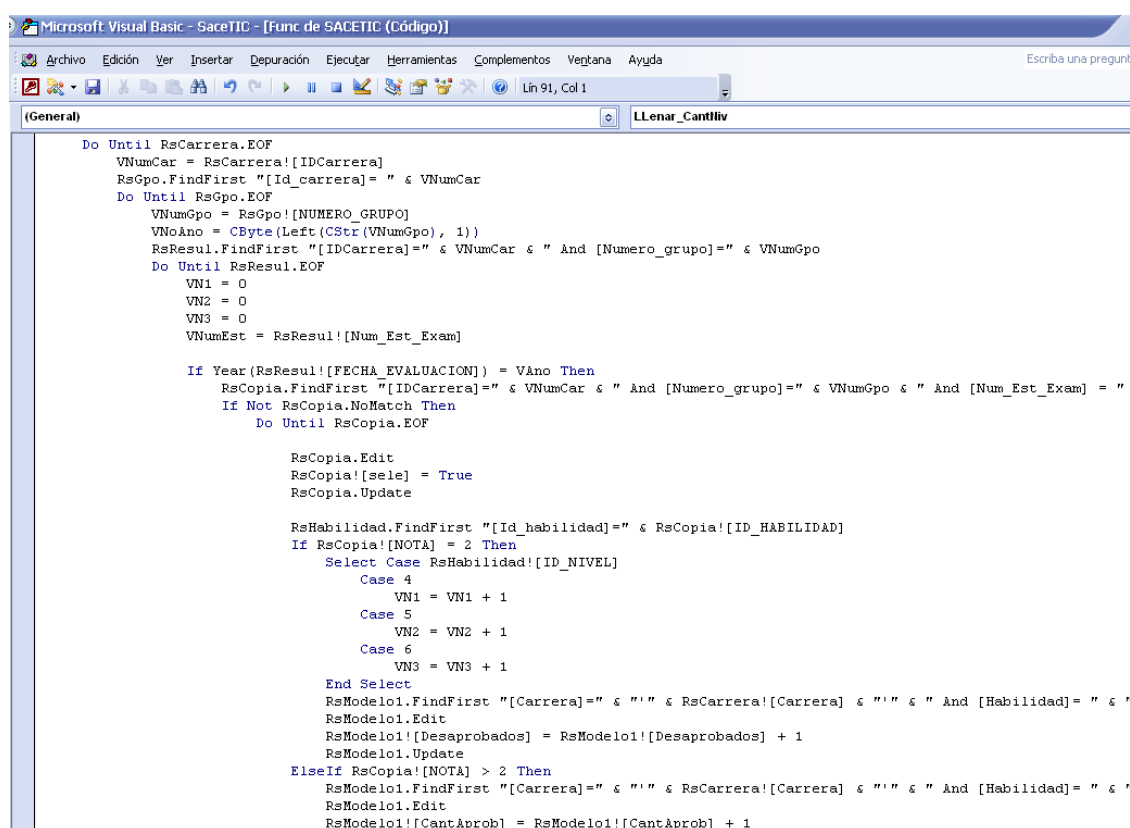


Fig. III.2.4.- Ventana para insertar un procedimiento



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC

- Posee estructuras de control, repetitivas y selectivas iguales a las de cualquier lenguaje moderno como:
 - IF ----- End If
 - Select ----- End Select
 - For ----- Next
 - Do ----- Loop
 - While ----- Whend



```
Do Until RsCarrera.EOF
    VNumCar = RsCarrera![IDCarrera]
    RsGpo.FindFirst "[ID_carrera]= " & VNumCar
    Do Until RsGpo.EOF
        VNumGpo = RsGpo![NUMERO_GRUPO]
        VNoAño = CByte(Left(CStr(VNumGpo), 1))
        RsResul.FindFirst "[IDCarrera]=" & VNumCar & " And [Numero_grupo]=" & VNumGpo
        Do Until RsResul.EOF
            VN1 = 0
            VN2 = 0
            VN3 = 0
            VNumEst = RsResul![Num_Est_Exam]

            If Year(RsResul![FECHA_EVALUACION]) = VNoAño Then
                RsCopia.FindFirst "[IDCarrera]=" & VNumCar & " And [Numero_grupo]=" & VNumGpo & " And [Num_Est_Exam] = "
                If Not RsCopia.NoMatch Then
                    Do Until RsCopia.EOF

                        RsCopia.Edit
                        RsCopia![sele] = True
                        RsCopia.Update

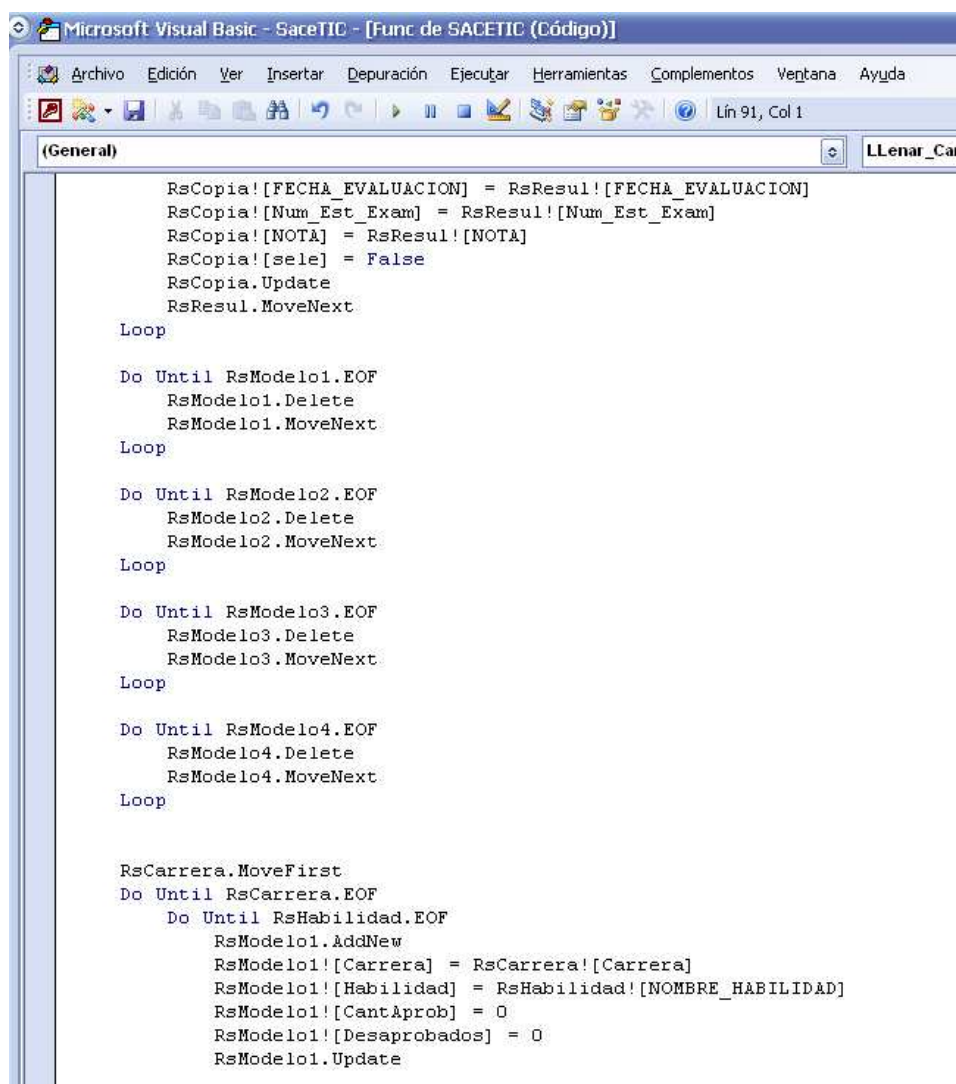
                        RsHabilidad.FindFirst "[Id_habilidad]=" & RsCopia![ID_HABILIDAD]
                        If RsCopia![NOTA] = 2 Then
                            Select Case RsHabilidad![ID_NIVEL]
                                Case 4
                                    VN1 = VN1 + 1
                                Case 5
                                    VN2 = VN2 + 1
                                Case 6
                                    VN3 = VN3 + 1
                            End Select
                        RsModelo1.FindFirst "[Carrera]=" & "'" & RsCarrera![Carrera] & "'" & " And [Habilidad]= " & "
                        RsModelo1.Edit
                        RsModelo1![Desaprobados] = RsModelo1![Desaprobados] + 1
                        RsModelo1.Update
                    End Until
                ElseIf RsCopia![NOTA] > 2 Then
                    RsModelo1.FindFirst "[Carrera]=" & "'" & RsCarrera![Carrera] & "'" & " And [Habilidad]= " & "
                    RsModelo1.Edit
                    RsModelo1![CantAprob] = RsModelo1![CantAprob] + 1
                End If
            End Do
        End Do
    End Do
End Do
```

Fig. III.2.5.- Ejemplo del uso de estructuras selectivas y repetitivas en SaceTIC

- Permite el trabajo con registros y campos utilizando los modelos: Objetos de Acceso de Datos (DAO) y ActiveX Data Objects (ADO).



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC



```
RsCopia![FECHA_EVALUACION] = RsResul![FECHA_EVALUACION]
RsCopia![Num_Est_Exam] = RsResul![Num_Est_Exam]
RsCopia![NOTA] = RsResul![NOTA]
RsCopia![sele] = False
RsCopia.Update
RsResul.MoveNext
Loop

Do Until RsModelo1.EOF
    RsModelo1.Delete
    RsModelo1.MoveNext
Loop

Do Until RsModelo2.EOF
    RsModelo2.Delete
    RsModelo2.MoveNext
Loop

Do Until RsModelo3.EOF
    RsModelo3.Delete
    RsModelo3.MoveNext
Loop

Do Until RsModelo4.EOF
    RsModelo4.Delete
    RsModelo4.MoveNext
Loop

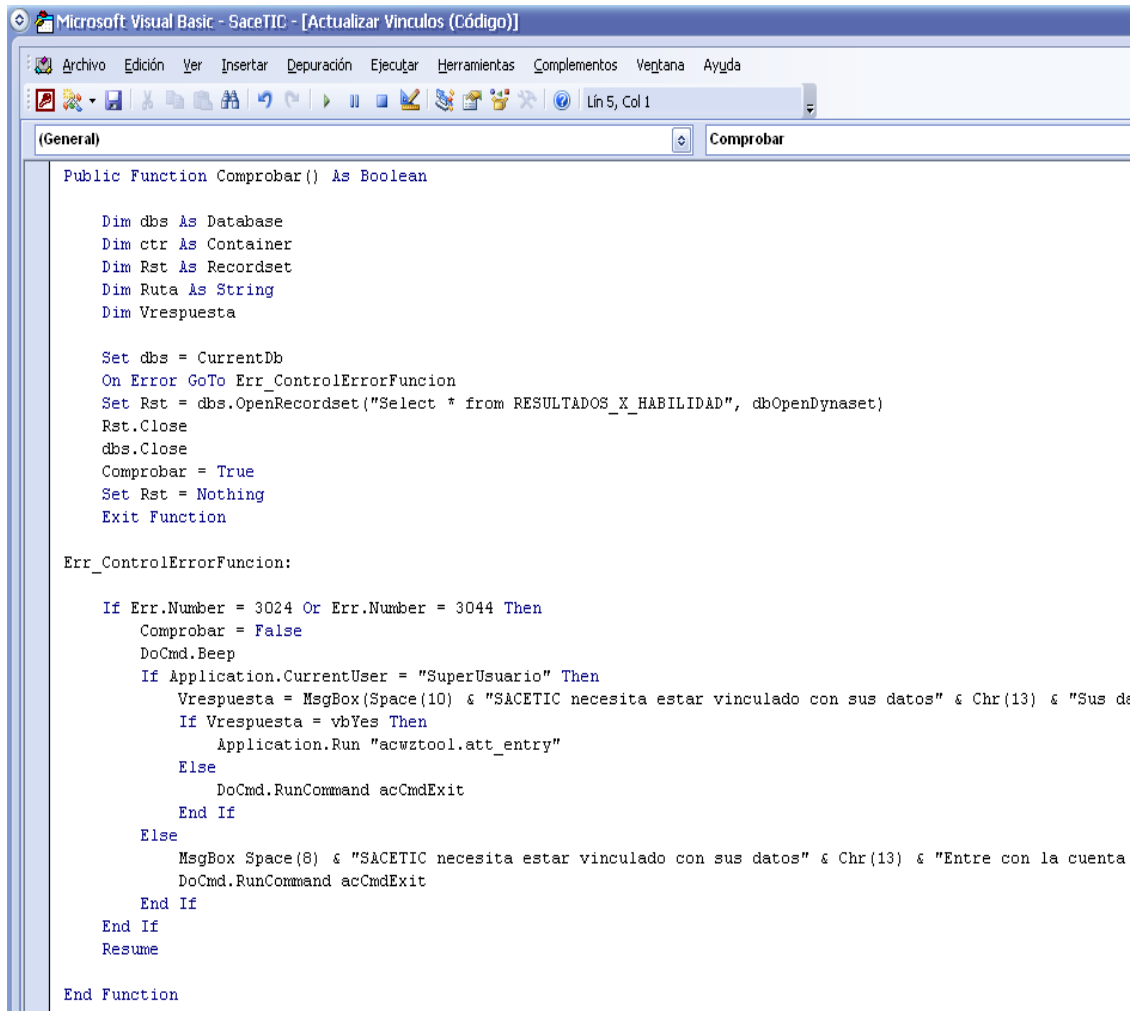
RsCarrera.MoveFirst
Do Until RsCarrera.EOF
    Do Until RsHabilidad.EOF
        RsModelo1.AddNew
        RsModelo1![Carrera] = RsCarrera![Carrera]
        RsModelo1![Habilidad] = RsHabilidad![NOMBRE_HABILIDAD]
        RsModelo1![CantAprob] = 0
        RsModelo1![Desaprobados] = 0
        RsModelo1.Update
```

Fig. III.2.6.- Ejemplo de trabajo en SaceTIC con registros y campos de objetos Recordset



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC

- Permite manipular los errores en tiempo de ejecución



```
Public Function Comprobar() As Boolean

    Dim dbs As Database
    Dim ctr As Container
    Dim rst As Recordset
    Dim Ruta As String
    Dim Vrespuesta

    Set dbs = CurrentDb
    On Error GoTo Err_ControlErrorFunction
    Set rst = dbs.OpenRecordset("Select * from RESULTADOS_X_HABILIDAD", dbOpenDynaset)
    rst.Close
    dbs.Close
    Comprobar = True
    Set rst = Nothing
    Exit Function

Err_ControlErrorFunction:

    If Err.Number = 3024 Or Err.Number = 3044 Then
        Comprobar = False
        DoCmd.Beep
        If Application.CurrentUser = "SuperUsuario" Then
            Vrespuesta = MsgBox(Space(10) & "SACETIC necesita estar vinculado con sus datos" & Chr(13) & "Sus da
            If Vrespuesta = vbYes Then
                Application.Run "acwztool.att_entry"
            Else
                DoCmd.RunCommand acCmdExit
            End If
        Else
            MsgBox Space(8) & "SACETIC necesita estar vinculado con sus datos" & Chr(13) & "Entre con la cuenta
            DoCmd.RunCommand acCmdExit
        End If
    End If
    Resume

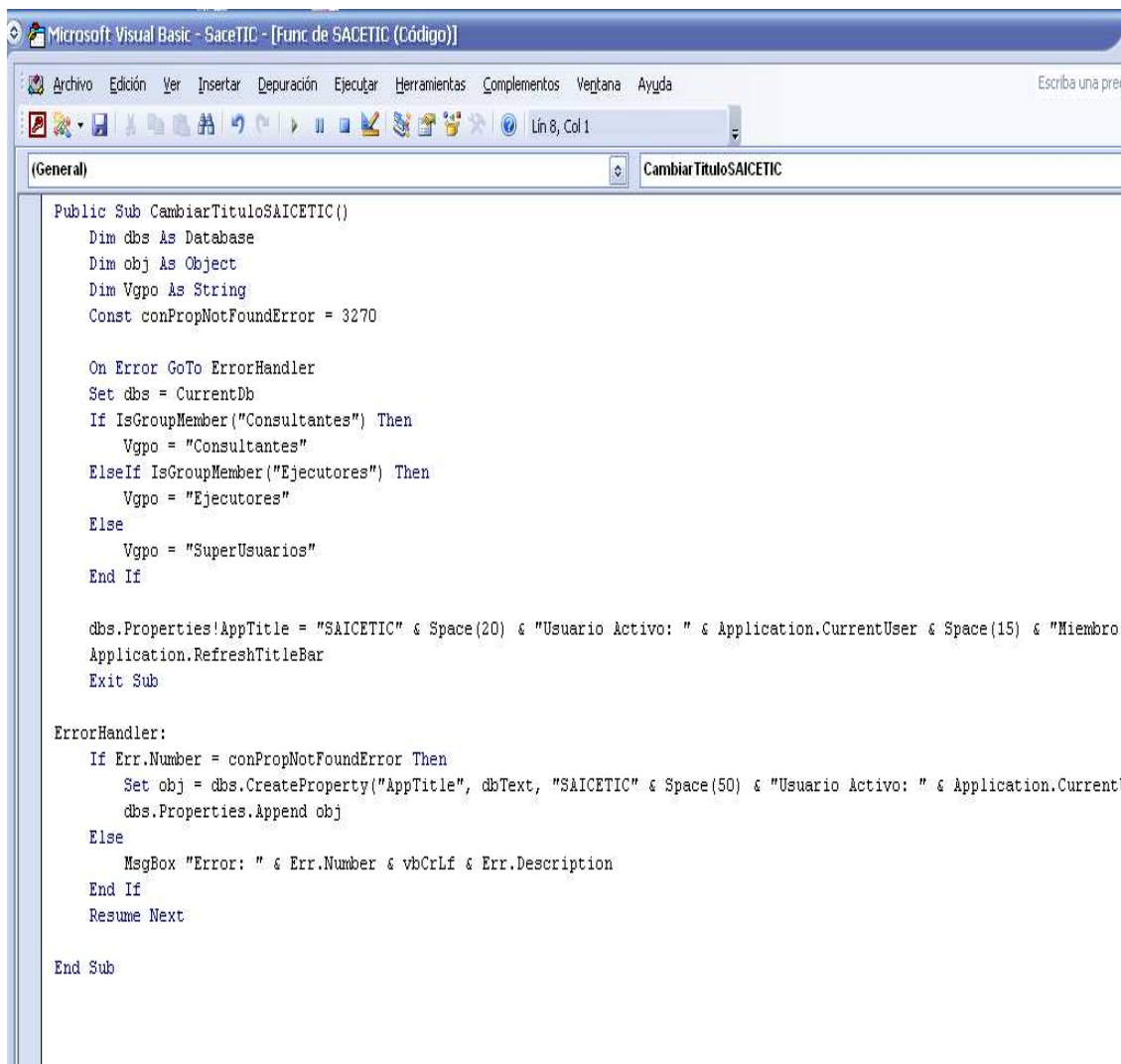
End Function
```

Fig. III.2.7.- Ejemplo de Manipulación de Errores en SaceTIC para detectar si esta vinculado a sus datos.



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC

- Permite cambiar las propiedades de los objetos en tiempo de la ejecución.



```
Public Sub CambiarTituloSAICETIC()  
    Dim dbs As Database  
    Dim obj As Object  
    Dim Vgpo As String  
    Const conPropNotFoundError = 3270  
  
    On Error GoTo ErrorHandler  
    Set dbs = CurrentDb  
    If IsGroupMember("Consultantes") Then  
        Vgpo = "Consultantes"  
    ElseIf IsGroupMember("Ejecutores") Then  
        Vgpo = "Ejecutores"  
    Else  
        Vgpo = "SuperUsuarios"  
    End If  
  
    dbs.Properties!AppTitle = "SAICETIC" & Space(20) & "Usuario Activo: " & Application.CurrentUser & Space(15) & "Miembro  
    Application.RefreshTitleBar  
    Exit Sub  
  
ErrorHandler:  
    If Err.Number = conPropNotFoundError Then  
        Set obj = dbs.CreateProperty("AppTitle", dbText, "SAICETIC" & Space(50) & "Usuario Activo: " & Application.CurrentUser  
        dbs.Properties.Append obj  
    Else  
        MsgBox "Error: " & Err.Number & vbCrLf & Err.Description  
    End If  
    Resume Next  
  
End Sub
```

Fig. III.2.8.- Cambio del título de la aplicación SaceTIC según usuario conectado



Epígrafe III.2.-Implementación de la Interfaz Usuario de SaceTIC

- Permite personalizar los mensajes de MSAccess.

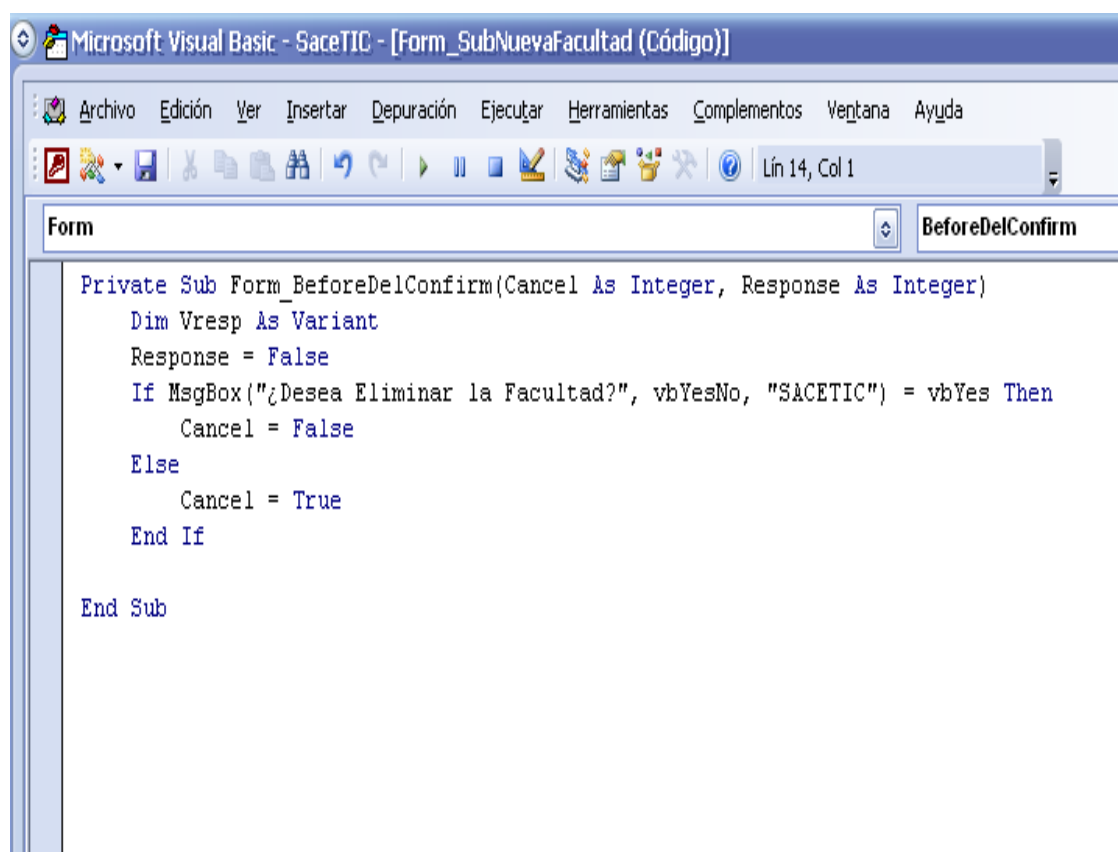


Fig. III.2.9.- Ejemplo de personalización del mensaje de Access en tiempo de ejecución

Al dar valor a Response de False se evita el mensaje de MSAccess y en su lugar se muestra el establecido en la función MsgBox.





III.3.- Implementación de la Seguridad de la Base de Datos.

Para asegurar la Base de Datos de SaceTIC se utilizó la opción de seguridad a nivel de objeto pues constituye el modo más fuerte de protección de una aplicación, aprovechando para ello el gestor utilizado.

Para su implementación es necesario primero crear un Grupo de Trabajo (vea Figura III.3.1).

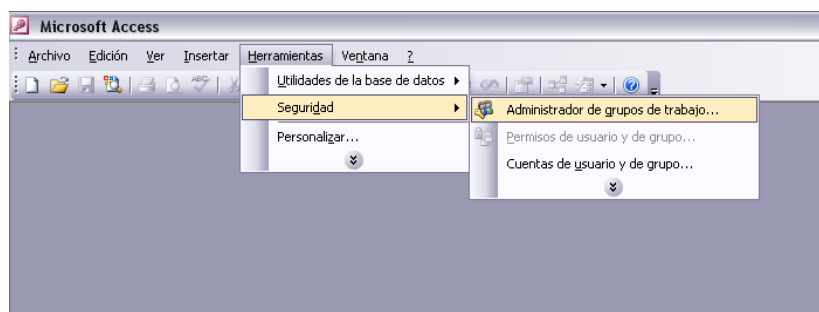


Figura III.3.1.- Uso del administrador de Grupo de Trabajo de MSAccess.

Creado el grupo de trabajo el gestor queda unido este y todas las Bases de Datos creadas pertenecerán a este grupo de trabajo (mientras este se encuentre activo) en vez de pertenecer al grupo System.mdw que es el que por defecto utiliza MSAccess. El grupo de trabajo contiene las cuentas de grupo, de usuarios y sus contraseñas.

En el caso de SaceTIC se crea el grupo de trabajo SaceTIC.mdw y en el se crearon las cuentas de usuarios Ejecutor y Consultante (vea Figura II.3.1 del capítulo II) utilizando para ello la herramientas de seguridad brindadas por MSAccess que se muestra en la figura III.3.2.



Epígrafe III.3.-Implementación de la Seguridad de la Base de Datos



Figura III.3.2.- Creando las cuentas de usuario de SaceTIC

Las cuentas de grupo de SaceTIC son las de MSAccess, (estas no pueden ser eliminadas):

- ❖ **Administradores**, al que pertenece la cuenta Administrador de MSAccess que es la que por defecto usada para la conexión al no tenerse implementada la seguridad. Esta posee todos los permisos sobre los objetos (tablas, consultas, formularios, informes y macros) al ser propietario de estos y de la Base de Datos y se le adiciona los permisos que tenga los grupos a que pertenece, así también podrá administrar la Base de Datos (esto le permite crear y eliminar cuentas de grupos y de usuarios, borrar contraseña de usuario, editar los permisos otorgados a las cuentas sobre los objetos)
- ❖ **Usuarios** al cual pertenecen todas las cuentas de usuario creadas, como puede verse en la figura III.3.3. A este grupo por defecto le es otorgado todos los permisos sobre la Base de Datos y objetos de esta excepto los de



Epígrafe III.3.-Implementación de la Seguridad de la Base de Datos

administrarla y Modificar su diseño por lo que obliga tener que quitarle todos los permisos incluyendo el de abrir la Base de Datos (vea figura III.3.4).



Figura III.3.3.- Grupos a los que pertenece la cuenta Ejecutor de SaceTIC.

Posteriormente se otorgan a las cuentas creadas los permisos que tendrán sobre la Base de Datos y sus objetos. Destaquemos que cuando se tenga varios usuarios pertenecientes a un grupo (que no es nuestro caso) sera más productivo establecer los permisos al grupo y no a los usuarios pues así se evita tener que ir dándolos a cada usuario por separado, al otorgarlo al grupo todos los usuarios pertenecientes a este tendrán estos permisos.



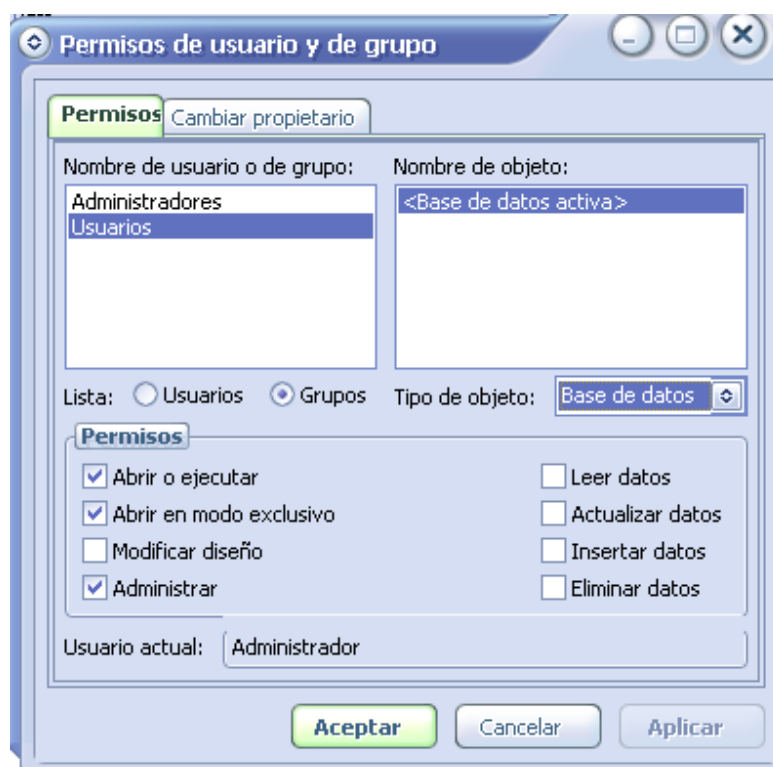


Figura III.3.4.-Editando los permisos sobre la Base de Datos del grupo usuarios

Los permisos posibles otorgar por Objetos son:

➤ **Base de Datos**

- ❖ Abrirla o no
- ❖ Abrirla en Modo Exclusivo o no
- ❖ Modificar Diseño o no
- ❖ Administrarla o no

➤ **Tablas y consultas**

- ❖ Abrirla o no.
- ❖ Leer o no el diseño.
- ❖ Modificar o no el diseño.



Epígrafe III.3.-Implementación de la Seguridad de la Base de Datos

- ❖ Leer o no los datos.
- ❖ Eliminar o no los datos.
- ❖ Modificar o no los datos.
- ❖ Añadir o no los datos.
- ❖ Administrarlas o no (con vista a poder importarlas o exportarlas a otra Base de Datos).

➤ Formularios, Informes y macros

- ❖ Abrir y Ejecutar o no.
- ❖ Leer Diseño o no.
- ❖ Modificar diseño o no.
- ❖ Administrarlos o no (con vista a poder importarlas o exportarlas a otra Base de Datos).

Los que pueden ser asignados a las cuentas de grupos y de usuarios.

Debe tener en cuenta siempre tener una cuenta que administre la Base de Datos en el caso de SaceTIC la cuenta es la de Administrador que trae Access (puede crearse una nueva) de no hacerlo no podrá incorporar nuevas cuentas o eliminarlas como tampoco editar los permisos otorgados a estas y eliminar su contraseña.

Para activar la seguridad es necesario ponerle una contraseña al usuario administrador de MSAccess que es el que en ese momento esta activo.

En el caso que sea necesario cambiar el propietario de la Base de Datos y de sus objetos por un nuevo usuario administrador diferente al que trae Access es necesario después de cerrar el grupo de trabajo creado (SaceTIC.mdw) realizar una nueva conexión con la cuenta de administrador establecida. En este caso al estar activada la seguridad se mostrara la ventana de conexión la que le exigirá la



Epígrafe III.3.-Implementación de la Seguridad de la Base de Datos

autenticación, es decir, nombre la cuenta y contraseña en este caso bastará con poner el nombre de la cuenta pues no posee esta contraseña; posteriormente podrá si así desea editar la contraseña de la cuenta con que entro (o ponerle una en caso que la cuenta no posea contraseña como es el caso). Estando conectado con la cuenta que de administrador tomada abrir una nueva Base de Datos la cual pertenecerá al grupo creado. La Base de Datos creada y todos los objetos que sean creados en ella o importados tendrán como propietario la cuenta que esta activada teniendo por ende este usuario permiso total sobre la Base de Datos y sus objetos. Señalemos que en el caso que se halla tomado para administrar la cuenta Administrador de Access no será necesario realizar éste paso ya que está cuenta es propietaria de la Base de Datos y de todos sus objetos no necesitándose cambiar de propietario.

Por último haciendo nuevamente uso del administrador de grupo de trabajo de MSAccess nos unimos nuevamente al grupo de trabajo System.mdw que por defecto trae MSAccess y la conexión al nuevo grupo de trabajo debe ser realizada por la aplicación para poder hacer uso de la Base de Datos.

Como puede apreciarse el proceso de proteger la Base de Datos resulta engorroso y difícil para un usuario normal y con escasos conocimientos sobre el gestor pero esto se compensa con creces por la eficiente protección que se logra a nivel de objeto (Base de Datos, Tablas, Consultas, Formularios Informes, etc.).

