INSTITUTO METROPOLITANO DE EDUCACIÓN
PROGRAMACIÓN DE COMPUTADORES
GUIA #6 DE VISUAL FOXPRO
DOCENTE: MAURICIO CANO

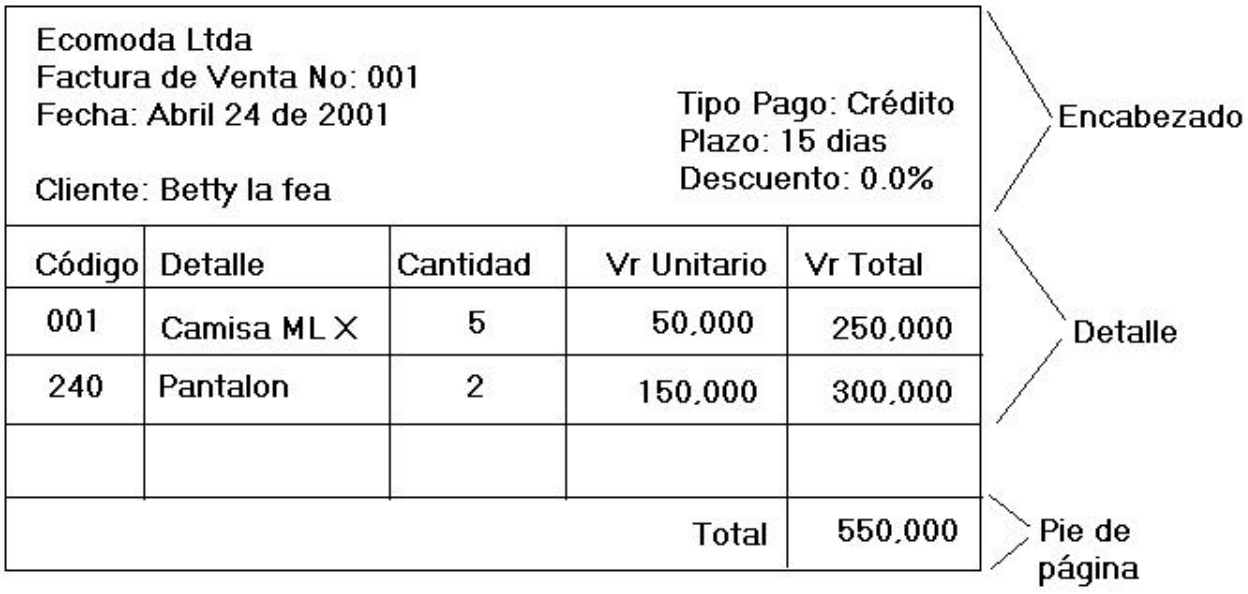
TÉCNICA DE PROGRAMACIÓN #3 – FORMULARIOS TIPO FACTURA

Existe una rutina muy popular entre los programadores y la cual se constituye en el "*quebradero de cabeza*" de más de un iniciado en la programación: el diseño de formularios que permitan la captura de información en documentos tales como la factura.

Un documento como la factura (bien sea de compra o de venta) tiene un encabezado y un detalle. También tienen esta característica los pedidos, los documentos soporte de inventario, una lista de grupos, un proceso de calificaciones, etc.

Cualquier documento empleado por una empresa y que tenga una presentación que incluya encabezado del documento y una descripción de detalle, debe ser automatizado empleando la técnica número 3. Es de aclarar que tal situación de programación se puede solucionar de muchas formas, planteamos en este curso, un método sencillo y práctico de automatización.

La siguiente gráfica muestra un posible documento de factura de compra. Aunque es bastante sencillo ilustra la característica de los documentos que requieren aplicar la técnica de programación propuesta en esta guía.



En donde reside lo complicado de programar un formulario para este documento? Cuando se efectúa el diseño de la base de datos, se descubre que es necesario dos tablas para solucionar el requerimiento del documento: la principal y el detalle, relacionando ambas tablas por un campo común, usualmente el campo clave de la tabla principal, en este caso *Número de factura.* Lo cual indica, que en la parte del detalle, el programa deberá visualizar solamente a los registros que posean el mismo número de factura del documento visualizado en la parte del encabezado. Otro asunto importante, es el de permitir manipular en forma ágil el detalle, permitiendo agregar, modificar y retirar información de las líneas que constituyen el detalle, y actualizando el total del documento en el pie de página del documento. Es decir, cada vez que se actualice la cantidad o el valor unitario de una de las líneas del detalle, se debe reflejar tales cambios en todo el documento.

FORMULARIOS TIPO FACTURA PROPUESTO

Como modelo de programación utilizaremos el control de los grupos de clase de una institución. Como parte principal del documento aparecerá la información relevante a un grupo de clases y como detalle, se colocarán todos los estudiantes pertenecientes a cada grupo. Utilizaremos además, uno de los mejores objetos incluidos en el Visual FoxPro, el *Grid (cuadrícula)*. Como parámetro de los elementos a mostrar en el *grid*, utilizaremos una instrucción *Select de SQL*.

El diseño utilizará las siguientes tablas: Programas, Asignaturas, Profesores, Hoja de vida, Matrícula, Grupos y Estudiantes por Grupo.

1. Grupos (*Grupos*)		
Nombre Campo	*Tipo de datos*	*Ancho*
Codigo	Carácter	10
Nombre	Carácter	50
Asignatura	Carácter	10
Horario	Carácter	50
Aula	Carácter	50
Profesor	Carácter	15
Campo clave ascendente: Codigo		

2. Estudiantes por Grupo (*EstGrupo*)		
Nombre Campo	*Tipo de datos*	*Ancho*
Codigo	Carácter	10
CodEstud	Carácter	10
ExParcial	Numerico	4,2
Seguim	Numerico	4,2
ExFinal	Numerico	4,2
Wfinal	Numerico	4,2
NotaFinal	Numerico	4,2
Estado	Carácter	1
Campo clave ascendente: Codigo + CodEstud		

3. Hojas de vida (Hvida)		
Nombre Campo	*Tipo de datos*	*Ancho*
Codigo	Carácter	10
Nombre	Carácter	50
Apellidos	Carácter	50
Programa	Carácter	2
TelResiden	Carácter	20
DirResiden	Carácter	40
FechaNac	Fecha	8
Campo clave ascendente: Codigo		

4. Matrícula (Matric)		
Nombre Campo	*Tipo de datos*	*Ancho*
Codigo	Carácter	10
Fecha	Fecha	8
Nivel	Carácter	2
Jornada	Carácter	1
Estado	Carácter	1
Observa	Memo	10
Campo clave ascendente: Codigo		

5. Maestro de Profesor (*Profe*)		
Nombre Campo	*Tipo de datos*	*Ancho*
Codigo	Carácter	15
Nombre	Carácter	50
TelResiden	Carácter	15
TelTrabajo	Carácter	15
Asignatura	Carácter	100
Campo clave ascendente: Codigo		

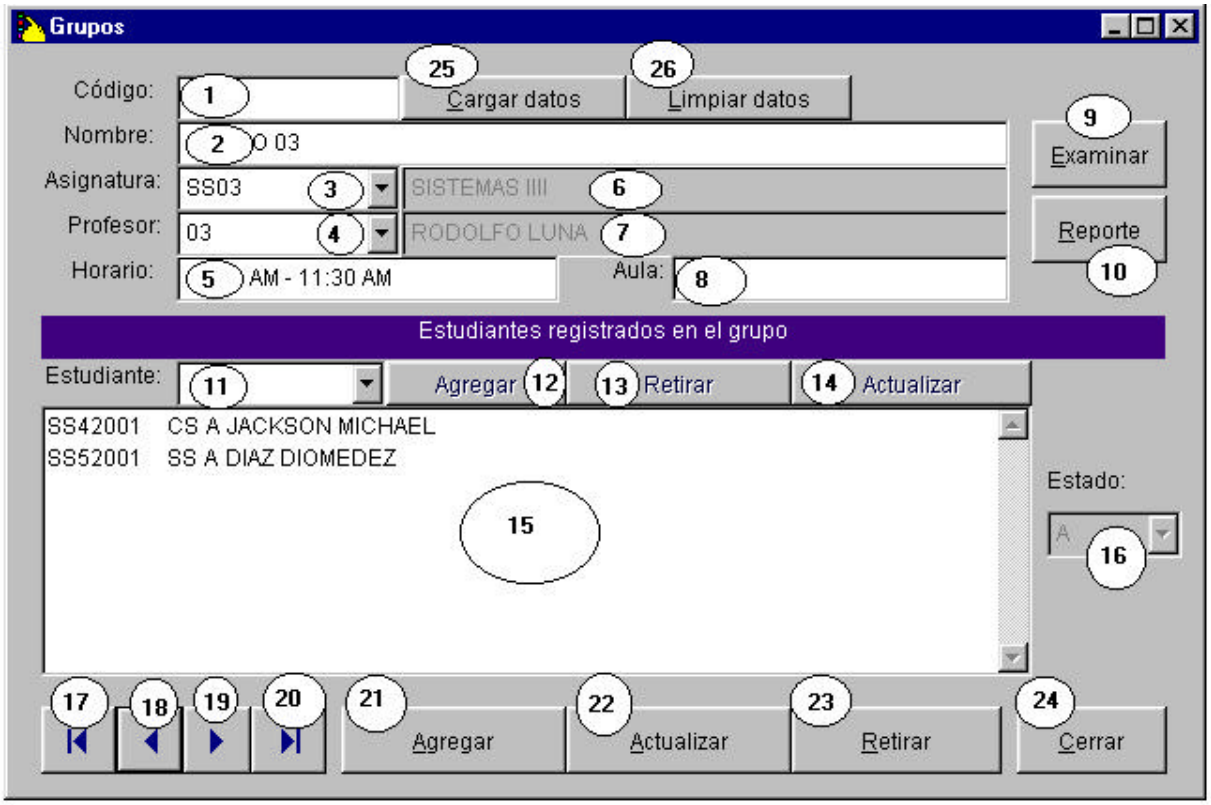
6. Maestro de Programas (Programa)		
Nombre Campo	*Tipo de datos*	*Ancho*
Codigo	Carácter	2
Nombre	Carácter	50
Campo clave ascendente: Codigo		

7. Maestro de Asignaturas (Asigna)		
Nombre Campo	*Tipo de datos*	*Ancho*
Codigo	Carácter	10
Nombre	Carácter	50
Programa	Carácter	2
Campo clave ascendente: Codigo		

Esta clase de rutina es un poco compleja debido al gran número de tablas que se requieren para su funcionamiento. Observe que para la rutina de grupos se requiere tener almacenados datos de programas, asignaturas, profesores, hojas de vida y datos de matrícula de los estudiantes. Si no se tienen datos en tales tablas, el diseño no funcionará debido a los controles de datos que efectúa sobre los datos entrados por el usuario.

EL FORMULARIO TIPO FACTURA

Utilizaremos la tabla de grupos y el detalle de grupos para ilustrar el funcionamiento de la técnica de programación en cuestión. Note que para esta clase de formularios, es normal combinar gran cantidad de tablas a la hora de la programación. Verifique el código visto en rutinas anteriores, debido a que cambia gran cantidad de instrucciones, para dar soporte a la nueva técnica.



1.mCodigo	2. mNombre	3. mAsignatura	4. mProfesor
5. mHorario	6. mNomAsigna	7. mNomProfesor	8. mAula
9. btnExaminar	10. btnReporte	11. mCodEstMaestro	12. btnAgregarDet
13. btnRetirarDet	14. btnModificarDet	15. DetalleGrupo	16. mEstado
17. btnPrimero	18. btnAnterior	19. btnSiguiente	20. btnUltimo
21. btnAgregar	22. btnActual	23. btnRetirar	24. btnCerrar
25. btnCargar	26. btnLimpiar		

Para crear los objetos 3, 4 y 11 utilice la técnica enseñada en clases anteriores para definir listas desplegables basadas en otras tablas.

Objeto 3 indica la asignatura para el grupo:

ControlSource: m.asignatura
RowSource: asigna.codigo, nombre
RowSourceType: 2-Alias
ColumnCount: 2
ColumnWitdh: 40,300

Objeto 4 indica el profesor asignado al grupo:

ControlSource: m.profesor
RowSource: profe.codigo,nombre
RowSourceType: 2-Alias
ColumnCount: 2
ColumnWitdh: 40,300

Objeto 11 sirve para seleccionar estudiantes e integrarlos al grupo activo en el formulario:

RowSource: hvida.codigo, apellidos, nombre
RowSourceType: 2-Alias
ColumnCount: 3
ColumnWitdh: 70,200,200

El objeto 16 es una lista basada en valores entrados a la hora de programar que indica el estado dentro del grupo para el estudiante:

ControlSource: m.estado
RowSource: A,Activo,S,Suspendido,C,Cancelado,N,No Matriculado
RowSourceType: 1-Valor
ColumnCount: 2
ColumnWitdh: 40, 100

Desactive los siguientes objetos en el formulario (propiedad enabled igual a .F.-Falso): 6, 7, 12, 13, 14 y 16.

LA RUTINA QUE INICIA TODO EL PROCESO

Como caso particular de este formulario, es que no se puede ejecutar directamente. Por problemas técnicos del Visual FoxPro, cuando se inicializan tablas en un formulario y a la vez en este se incluyen elementos de cuadrícula (*grid*) o cuadro de listas (*combo list*) simplemente no funcionan. Por tal razón, debemos crear una rutina que abra todas las tablas a usar en el sistema y llamar al formulario al final de la rutina.

Como origen de datos para una *cuadrícula* o *combo list*, resulta muy útil el declarar una instrucción *SELECT - SQL*.

Este es el contenido de la rutina *I_GRUPOS.PRG* (no olvide almacenarla en el sitio destinado para los programas):

I_GRUPOS.PRG

```
* Esta rutina invoca al formulario Grupos

*abrir entorno para el formulario grupos utilizando
*las variables publicas

*tabla de grupos
mFile1 = _dircia +"grupos.dbf"
if !file( mFile1)
        do cGrupos in prg\creatbl
endif
sele 1
use &mFile1 order tag codigo alias grupos

*tabla estudiantes por grupo
mFile2 = _dircia +"estgrupo.dbf"
if !file( mFile2)
        do cEstGrupo in prg\creatbl
endif
sele 2
use &mFile2 order tag codigo alias estgrupo

*tabla maestro asignaturas
mFile3 = _dircia +"asigna.dbf"
if !file( mFile3)
        do cAsigna in prg\creatbl
endif
sele 3
use &mFile3 order tag codigo alias asigna

*tabla maestro programas
mFile4 = _dircia +"programa.dbf"
if !file( mFile4)
        do cPrograma in prg\creatbl
endif
sele 4
use &mFile4 order tag codigo alias programa
```

```
*tabla maestro profesores
mFile5 = _dircia +"profe.dbf"
if !file( mFile5)
        do cProfe in prg\creatbl
endif
sele 5
use &mFile5 order tag codigo alias profe

*tabla hojas de vida
mFile6 = _dircia +"hvida.dbf"
if !file( mFile6)
        do cHvida in prg\creatbl
endif
sele 6
use &mFile6 order tag codigo alias hvida

*tabla matricula
mFile7 = _dircia +"matric.dbf"
if !file( mFile7)
        do cMatric in prg\creatbl
endif
sele 7
use &mFile7 order tag codigo alias matric

sele 1

*apuntar a la carpeta de los formularios
do form forms\grupos

return
```

Lo único que hace esta rutina es abrir las tablas requeridas en diferentes áreas de trabajo, asignar los respectivos alias y por supuesto, pasar el control de ejecución, al formulario tipo factura.

RUTINAS EN LOS OBJETOS DEL FORMULARIO

Formulario.INIT

```
*continuación del codigo iniciado en i_grupos
public pCodEstud
pCodEstud = ""

set delete on

sele 1
go top
scatter memvar memo

*ir al objeto inicial
thisform.mCodigo.setfocus

*actualizar grid (cuadrícula)
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
        "hvida.programa, estgrupo.estado, " +;
        "alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
        "from estGrupo join hvida " +;
        "on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
        "where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
        ".and. !deleted() order by codEstud noconsole nowait " + ;
        "into cursor miconsulta "

thisform.DetalleGrupo.refresh
thisform.refresh
```

En esta rutina se hace uso de la instrucción SELECT-SQL y se asigna como parámetro al objeto DetalleGrupo. Cada vez que se requiera actualizar el detalle de cada grupo, simplemente invocaremos esta sección de código y refrescaremos el objeto en el formulario.

Observe que la instrucción se arma como una cadena. Esa es la razón del porqué de tantas " y ; en la instrucción. Esta orden podría colocarse en una sola línea, pero resulta conveniente dividirla en secciones, para poderla entender. El docente guía le indicará que hace la orden SQL empleada.

Objeto BtnPrimero.Click

```
*ir al primero
wait window "Primer registro..." nowait
sele 1
go top
scatter memvar memo

sele asigna
seek alltrim( m.asignatura)

sele profe
seek alltrim( m.profesor)

*actualizar grid (cuadrícula)
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
                "hvida.programa, estgrupo.estado, " +;
                "alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
                "from estGrupo join hvida " +;
                "on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
                "where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
                ".and. !deleted() order by codEstud noconsole nowait " + ;
                "into cursor miconsulta "

thisform.DetalleGrupo.refresh

*controlar botones del detalle
pCodEstud = ""
thisform.btnRetirarDet.enabled = .f.
thisform.btnModificarDet.enabled = .f.
thisform.mEstado.enabled = .f.

thisform.refresh
return
```

Objeto BtnAnterior.Click

```
*ir al anterior
sele 1
if !bof()
	skip -1
endif

if bof()
	go top
endif

scatter memvar memo

sele asigna
seek alltrim( m.asignatura)

sele profe
seek alltrim( m.profesor)

*actualizar grid (cuadrícula)
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
		"hvida.programa, estgrupo.estado, " +;
		"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
		"from estGrupo join hvida " +;
		"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
		"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " +;
		".and. !deleted() order by codEstud noconsole nowait " + ;
		"into cursor miconsulta "

thisform.DetalleGrupo.refresh

*controlar botones del detalle
pCodEstud = ""
thisform.btnRetirarDet.enabled = .f.
thisform.btnModificarDet.enabled = .f.
thisform.mEstado.enabled = .f.

thisform.refresh
return
```

Objeto btnSiguiente.click

```
*ir al siguiente
sele 1
if !eof()
	skip 1
endif

if eof()
	go bottom
endif

scatter memvar memo

sele asigna
seek alltrim( m.asignatura)

sele profe
seek alltrim( m.profesor)

*actualizar grid (cuadrícula)
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
		"hvida.programa, estgrupo.estado, " +;
		"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
		"from estGrupo join hvida " +;
		"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
		"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
		".and. !deleted() order by codEstud noconsole nowait " + ;
		"into cursor miconsulta "

thisform.DetalleGrupo.refresh

*controlar botones del detalle
pCodEstud = ""
thisform.btnRetirarDet.enabled = .f.
thisform.btnModificarDet.enabled = .f.
thisform.mEstado.enabled = .f.

thisform.refresh
return
```

Objeto btnUltimo.click

```
*ir al ultimo
wait window "Último registro..." nowait
sele 1
go bottom
scatter memvar memo

sele asigna
seek alltrim( m.asignatura)

sele profe
seek alltrim( m.profesor)

*actualizar grid (cuadrícula)
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
                "hvida.programa, estgrupo.estado, " +;
                "alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
                "from estGrupo join hvida " +;
                "on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
                "where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
                ".and. !deleted() order by codEstud noconsole nowait " + ;
                "into cursor miconsulta "

thisform.DetalleGrupo.refresh

*controlar botones del detalle
pCodEstud = ""
thisform.btnRetirarDet.enabled = .f.
thisform.btnModificarDet.enabled = .f.
thisform.mEstado.enabled = .f.

thisform.refresh
return
```

Objeto BtnAgregar.clic

```
*agregar registro con los datos en pantalla
sele 1

*verificar si hay datos en blanco
if empty( thisform.mCodigo.value)
	wait window "Código incorrecto..." nowait
	thisform.mCodigo.setfocus
	return
endif

if empty( thisform.mNombre.value)
	wait window "Nombre incorrecto..." nowait
	thisform.mNombre.setfocus
	return
endif

*verificar campo clave
mCodigo = alltrim( thisform.mCodigo.value)
sele 1
seek mCodigo
if found()
	mMens1 = "Imposible agregar, el código " +mCodigo +" ya existe!" +chr(13)
	mMens2 = "Utilice otro código e intente de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	thisform.mCodigo.setfocus
	return
endif

*validar la asignatura entrada por el usuario
mAsignatura = alltrim( thisform.mAsignatura.value)
sele asigna
seek mAsignatura
if !found()
	mMens1 = "El código de Asignatura " +mAsignatura +" no existe!" +chr(13)
	mMens2 = "Utilice un código válido e intente de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	thisform. mAsignatura.setfocus
	return
endif
```

```
*validar el profesor entrado por el usuario
mProfesor = alltrim( thisform.mProfesor.value)
sele profe
seek mProfesor
if !found()
	mMens1 = "El código de Profesor " +mAsignatura +" no existe!" +chr(13)
	mMens2 = "Utilice un código válido e intente de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	thisform.mProfesor.setfocus
	return
endif

*agregar nuevo registro con los datos en pantalla
sele 1
insert into grupos from memvar

*cargar nuevos datos en blanco
scatter memvar blank

*ir al primer campo
thisform.mCodigo.setfocus

*actualizar nuevo detalle
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
	"hvida.programa, estgrupo.estado, " +;
	"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
	"from estGrupo join hvida " +;
	"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
	"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " +;
	".and. !deleted() order by codEstud noconsole nowait " + ;
	"into cursor miconsulta "

thisform.refresh
```

Objeto btnActual.click

```
*actualizar el contenido de un campo

*verificar si hay datos en blanco
if empty( thisform.mCodigo.value)
	wait window "Código incorrecto..." nowait
	thisform.mCodigo.setfocus
	return
endif

if empty( thisform.mNombre.value)
	wait window "Nombre incorrecto..." nowait
	thisform.mNombre.setfocus
	return
endif

if empty( thisform.mAsignatura.value)
	wait window "Código de Asignatura incorrecto..." nowait
	thisform.mAsignatura.setfocus
	return
endif

if empty( thisform.mProfesor.value)
	wait window "Código de Profesor incorrecto..." nowait
	thisform.mProfesor.setfocus
	return
endif

*validar asignatura entrada por el usuario
mAsignatura = alltrim( thisform.mAsignatura.value)
sele asigna
seek mAsignatura
if !found()
	mMens1 = "El código de Asignatura " +mAsignatura +" no existe!" +chr(13)
	mMens2 = "Utilice un código válido e intente de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	thisform.mAsignatura.setfocus
	return
endif

*validar profesor entrado por el usuario
mProfesor = alltrim( thisform.mProfesor.value)
sele profe
seek mProfesor
if !found()
	mMens1 = "El código de Profesor " +mProfesor +" no existe!" +chr(13)
	mMens2 = "Utilice un código válido e intente de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	thisform.mProfesor.setfocus
	return
endif
```

```
*actualizar datos del campo clave
sele 1
mCodigo = alltrim( thisform.mCodigo.value)
seek mCodigo
if found()  && actualizar registro hallado
	gather memvar memo
	wait window "Datos actualizados..." nowait
else
	mMens1 = "Imposible actualizar, el código " +mCodigo +" no existe!" +chr(13)
	mMens2 = "Utilice otro código e intente actualizar de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	thisform.mCodigo.setfocus
	return
endif

*actualizar detalle
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
	"hvida.programa, estgrupo.estado, " +;
	"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
	"from estGrupo join hvida " +;
	"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
	"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " +;
	".and. !deleted() order by codEstud noconsole nowait " + ;
	"into cursor miconsulta "

sele 1

thisform.refresh
return
```

Objeto btnRetirar.click

```
*retirar el registro clave hallado
sele 1
mCodigo = alltrim( thisform.mCodigo.value)
seek mCodigo
if !found()  && no existe codigo
	mMens1 = "Imposible retirar, el código " +mCodigo +" no existe!" +chr(13)
	mMens2 = "Utilice otro código e intente retirar de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	thisform.mCodigo.setfocus
	return
endif

mMens1 = "Desea retirar el código " + mCodigo +" del sistema!" +chr(13)
mOpc = messagebox( mMens1 , 1+32, "Retirar registro!")

if mOpc = 1 && Clic en botón Aceptar

	sele 1
	delete
	wait window "Registro retirado del sistema..." nowait

	*mover puntero al próximo disponible
	if !eof()
		skip 1
	endif
	if eof()
		go bottom
	endif
	*retirar estudiantes del grupo eliminado
	sele estGrupo
	dele all for alltrim( estGrupo.codigo) = mCodigo
	sele 1
endif

sele 1
scatter memvar memo

*actualizar detalle
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
	"hvida.programa, estgrupo.estado, " +;
	"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
	"from estGrupo join hvida " +;
	"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
	"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " +;
	".and. !deleted() order by codEstud noconsole nowait " + ;
	"into cursor miconsulta "

thisform.refresh
return
```

Objeto btnExaminar.click

```
*examinar registros
sele 1
browse fields codigo, nombre ;
        noedit noappend nodelete

*activar el seleccionado
scatter memvar memo

*actualizar grid (cuadrícula)
sele estGrupo
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
        "hvida.programa, estgrupo.estado, " +;
        "alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
        "from estGrupo join hvida " +;
        "on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
        "where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
        ".and. !deleted() order by codEstud noconsole nowait " + ;
        "into cursor miconsulta "

thisform.DetalleGrupo.refresh

thisform.refresh
return
```

Objeto btnLimpiar.clic

```
*limpiar el contenido de los campos
wait window "Datos en blanco..." nowait

sele 1
scatter memvar memo blank

*actualizar grid (cuadrícula)
sele estGrupo
thisform.DetalleGrupo.rowsource = ""
thisform.DetalleGrupo.refresh

thisform.refresh
```

Objeto btnCargar.click

```
*cargar el contenido de los campos con la clave entrada

*verificar si hay datos en blanco
if empty( thisform.mCodigo.value)
	wait window "Código incorrecto..." nowait
	thisform.mCodigo.setfocus
	return
endif

*cargar datos del campo clave
sele 1
mCodigo = alltrim( thisform.mCodigo.value)
seek mCodigo
if found()  && actualizar registro hallado
	scatter memvar memo
	wait window "Datos cargados..." nowait
else
	mMens1 = "Imposible cargar datos, el código " +mCodigo +" no existe!" +chr(13)
	mMens2 = "Utilice otro código e intente cargar datos de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
endif

*actualizar grid (cuadrícula)
sele estGrupo
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
	"hvida.programa, estgrupo.estado, " +;
	"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
	"from estGrupo join hvida " +;
	"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
	"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
	".and. !deleted() order by codEstud noconsole nowait " + ;
	"into cursor miconsulta "

thisform.DetalleGrupo.refresh

thisform.refresh
thisform.mCodigo.setfocus

return
```

Objeto mCodEstMaestro.valid

```
*activar los otros botones para el detalle

thisform.btnAgregarDet.enabled = .t.
thisform.btnRetirarDet.enabled = .t.
thisform.btnModificarDet.enabled = .t.
```

mCodEstMaestro.when

```
return !empty( m.codigo)
```

Objeto btnAgregarDet.when

```
return !empty( m.codigo) .and. !empty( thisform.mCodEstMaestro.value)
```

BtnAgregarDet.valid

```
*agregar codigo seleccionado al detalle del grupo
	mCodEstud = alltrim( thisform.mCodEstMaestro.value)

	*validar código en maestro estudiantes
	sele hvida
	seek mCodEstud
	if !found()
		mMens1 = "Imposible agregar al detalle, el código " +mCodEstud +" no existe en el maestro!" +chr(13)
		mMens2 = "Utilice otro código e intente agegar al detalle de nuevo..."
		mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
		thisform.mCodEstMaestro.setfocus
		return
	endif

	*verificar estudiante en el grupo
	sele estgrupo
	mGrupo = alltrim( m.codigo)
	seek mGrupo + mCodEstud
	if found()
		mMens1 = "Imposible agregar estudiante al grupo, el código " +mCodEstud +" ya existe!" +chr(13)
		mMens2 = "Utilice otro código e intente agegar al detalle de nuevo..."
		mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
		thisform.mCodEstMaestro.setfocus
		return
	endif

	*agregar linea al detalle
	insert into estGrupo ;
		( codigo, codEstud, estado) ;
	values ;
		( mGrupo, mCodEstud, "A")

	*actualizar grid (cuadrícula)
	thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
		"hvida.programa, estgrupo.estado, " +;
		"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
		"from estGrupo join hvida " +;
		"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
		"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " +;
		".and. !deleted() order by codEstud noconsole nowait " + ;
		"into cursor miconsulta "

	thisform.DetalleGrupo.refresh

	*keyboard "{ctrl+w}"
	thisform.refresh
```

Objeto BtnRetirarDet.valid

```
*retirar el registro seleccionado en el detalle
*utilizar pCodEstud como clave de seleccion

if empty( pCodEstud)
	mMens1 = "Seleccione un código del detalle e intente retirar de nuevo..."
	mOpc = messagebox( mMens1, 0+64, "Atención!")
	thisform.DetalleGrupo.setfocus
	return
endif

sele estGrupo
mCodigo = alltrim( m.codigo) +alltrim( pCodEstud)
seek mCodigo
if !found()  && no existe codigo
	mMens1 = "Imposible retirar, el código " + pCodEstud +" no existe en el detalle!" +chr(13)
	mMens2 = "Seleccione un código del detalle e intente retirar de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	pCodEstud = ""
	thisform.DetalleGrupo.setfocus
	return
endif

mMens1 = "Desea retirar el código " + pCodEstud +" del grupo!" +chr(13)
mOpc = messagebox( mMens1 , 1+32, "Retirar estudiante del Grupo!")

if mOpc = 1 && Clic en botón Aceptar

	sele estGrupo
	delete

	pCodEstud = ""

	wait window "Registro retirado del sistema..." nowait

endif

sele 1
scatter memvar memo
```

```
*actualizar detalle
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
        "hvida.programa, estgrupo.estado, " +;
        "alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
        "from estGrupo join hvida " +;
        "on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
        "where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
        ".and. !deleted() order by codEstud noconsole nowait " + ;
        "into cursor miconsulta "

thisform.DetalleGrupo.refresh
thisform.refresh
return
```

BtnRetirarDet.when

```
*habilitar cuando hagan clic en el detalle del grupo
return !empty( pCodEstud)
```

Objeto btnModificarDet.valid

```
*actualizar el registro seleccionado en el detalle
*utilizar pCodEstud como clave de seleccion

if empty( pCodEstud)
	mMens1 = "Seleccione un código del detalle e intente actualizar de nuevo..."
	mOpc = messagebox( mMens1, 0+64, "Atención!")
	thisform.DetalleGrupo.setfocus
	return
endif

sele estGrupo
mCodigo = alltrim( m.codigo) +alltrim( pCodEstud)
seek mCodigo
if !found()  && no existe codigo
	mMens1 = "Imposible actualizar, el código " + pCodEstud +" no existe en el detalle!" +chr(13)
	mMens2 = "Seleccione un código del detalle e intente retirar de nuevo..."
	mOpc = messagebox( mMens1 +mMens2, 0+64, "Atención!")
	pCodEstud = ""
	thisform.DetalleGrupo.setfocus
	return
endif

mMens1 = "Desea actualizar los datos del código " + pCodEstud +"!" +chr(13)
mOpc = messagebox( mMens1 , 1+32, "Actualizar datos del estudiante!")

if mOpc = 1 && Clic en botón Aceptar

	sele estGrupo
	replace estGrupo.estado with alltrim( thisform.mEstado.value)

	wait window "Datos actualizados..." nowait

endif
sele 1
scatter memvar memo
*actualizar detalle
thisform.DetalleGrupo.rowsource = "select all estgrupo.codEstud, " + ;
	"hvida.programa, estgrupo.estado, " +;
	"alltrim( hvida.apellidos) +space(1)+alltrim( hvida.nombre) " +;
	"from estGrupo join hvida " +;
	"on alltr( estGrupo.codEstud) = alltr( hvida.codigo) " + ;
	"where alltrim( estGrupo.codigo) = alltrim( m.codigo) " + ;
	".and. !deleted() order by codEstud noconsole nowait " + ;
	"into cursor miconsulta "

thisform.DetalleGrupo.refresh
thisform.refresh
return
```

BtnModificarDet.when

```
*habilitar cuando hagan clic en el detalle del grupo
return !empty( pCodEstud)
```

Objeto DetalleGrupo.click

```
*codigo seleccionado para borrar o modificar
pCodEstud = alltrim( this.value)

*buscar codigo en detalle para que funcione
*modificacion de datos
mCodigo = alltrim( m.codigo) +alltrim( pCodEstud)
sele estGrupo
seek mCodigo

*datos disponibles para actualizar
m.estado = estgrupo.estado
thisform.mEstado.enabled = .t.

thisform.btnRetirarDet.enabled = .t.
thisform.btnModificarDet.enabled = .t.

thisform.refresh

return
```

Observaciones:

- El objeto DetalleGrupo debe definirse como un Combo List y debe contemplar los siguientes parámetros:

  RowSourceType: 3-SQL
  ColumnCount: 5
  ColumnLines:..F.-Falso

- El objeto *mNomAsigna* debe tener la propiedad *ControlSource* igual a: *asigna.nombre*

- El objeto *mNomProfesor* debe tener la propiedad *ControlSource* igual a: *profe.nombre*

ADICION DE SUBPROGRAMAS A LA RUTINA CREATBL

El archivo de procedimientos CREATBL.PRG, debe contener dos subprogramas que permitan crear las tablas de *Grupos.dbf* y *EstGrupo.dbf*.

Debe insertar el siguiente código en la el archivo de procedimientos CREATBL.PRG:

```
procedure cGrupos
	*definir tabla grupos
	mFile = _dircia + "grupos.dbf"
	create table &mFile ;
		( codigo     c ( 10), ;
		  nombre     c ( 50), ;
		  asignatura c ( 10), ;
		  profesor   c ( 15), ;
		  horario    c ( 50), ;
		  aula       c ( 50))
	index on codigo tag codigo for !deleted()
	index on nombre tag nombre for !deleted()
	close data
return

procedure cEstGrupo
	*definir tabla grupos
	mFile = mFile = _dircia + "estgrupo.dbf"
	create table &mFile ;
		( codigo     c ( 10), ;
		  codestud   c ( 10), ;
		  exparcial  n (4,2), ;
		  seguim     n (4,2), ;
		  exfinal    n (4,2), ;
		  wfinal     n (4,2), ;
		  estado     c (  1))
	index on alltrim( codigo) + alltrim( codestud) tag codigo for !deleted()
	close data
return
```

Nota: Observe que el campo índice de la tabla Estudiantes por Grupo es de tipo compuesto. Revise la teoría ofrecida en las guías de clase anteriores para comprender mejor la situación y el manejo de campos claves compuestos. Revise en el código de la rutina Grupos para analizar en que puntos se hace el uso de los índices compuestos y su forma correcta de utilizarlos.