

Haciendo un resumen de lo que hemos visto hasta ahora tenemos un formulario que nos permite ver una relación varios a varios y otros formularios que nos permiten agregar registros a la tabla puente de la relación varios a varios mencionada. Ahora vamos a relacionar cada uno de los registros de la tabla puente con otros formularios que nos amplían información. Volvemos a echar un vistazo al esquema de relaciones de la aplicación, y, al origen de datos del formulario.

```
SELECT llamada.IDTRABAJADOR, llamada.IDRODAJE, llamada.FECHA_LLAMADA,  
llamada.FECHA_RODAJE, llamada.HORA, llamada.INCIDENCIAS  
FROM llamada  
WHERE (((llamada.IDTRABAJADOR)=[Forms]![participa3]![participa_IDPERSONAL]) AND  
((llamada.IDRODAJE)=[Forms]![participa3]![participa_IDRODAJE]))  
ORDER BY llamada.FECHA_LLAMADA DESC , llamada.HORA DESC;
```

[Ver el formulario.]

La tabla "participa" es la que interviene en el origen de datos del formulario que muestra la relación varios a varios. Las otras tablas puente son una ampliación de información de cada uno de sus registros. A primera vista puede parecer que sería mejor agrupar todas las tablas puente en una sola pero hay que tener en cuenta que esto provocaría muchas redundancias y campos en blanco, cosa que hay que evitar a toda costa según nos dicen en la teoría de diseño de bases de datos todos los autores.

Si alguna vez has tenido o tienes que concertar cientos de entrevistas todos los días agradecerás tener una herramienta como este formulario que vamos a comentar.

[Ver el formulario.]

Como ves se recogen las incidencias de cada llamada para cada trabajador y proyecto. No deja de asombrarme la facilidad con que se almacena la información y puede ser utilizada posteriormente. El usuario guarda las incidencias haciendo dos clic y cada vez que necesite la información el formulario, que tiene RecordSource dinámico dependiendo del registro que esté seleccionado en el formulario que muestra la relación varios a varios, le muestra los registros que permiten al usuario tomar una decisión. Se distinguen dos zonas en el formulario, Encabazado y Detalle.

[Ver el formulario.]

La llamada se realiza con un solo clic.

[Ver el formulario.]

La sección encabezado copia los campos más descriptivos del registro que tenemos seleccionado en el formulario que muestra la relación varios a varios mediante el código del siguiente procedimiento. Vemos también que el formulario tiene RecordSource dinámico dependiendo del registro seleccionado en el formulario que muestra la relación varios a varios.

Private Sub Form\_Activate()

[La función de estas tres variables la explicaré en otro artículo, de momento su presencia no debe preocuparte.]

```
abrir_faltas_llamadas = True  
abrir_faltas_agregarparametros = False  
abrir_faltas_agregaropciones = False
```

```
elect = "SELECT [llamada].[IDTRABAJADOR], [llamada].[IDRODAJE],  
[llamada].[FECHA_LLAMADA], " & _  
" [llamada].[FECHA_RODAJE], [llamada].[HORA], [llamada].[INCIDENCIAS]"
```

```
rom = " FROM llamada"
```

```
here = " WHERE [llamada].[IDRODAJE]=[Forms]![participa3]![participa_IDRODAJE] And" & _  
" [llamada].[IDTRABAJADOR]=[Forms]![participa3]![participa_IDPERSONAL]"
```

```
rder = " ORDER BY [llamada].[FECHA_LLAMADA] DESC , [llamada].[HORA] DESC"
```

```
fin = ";
```

```
Me.RecordSource = elect + rom + here + rder + fin
```

[los controles independientes muestran datos del registro activo del formulario que muestra la relación varios a varios.]

```
Me.[Texto10] = [Forms]![participa3]![NOMBRE]  
Me.[Texto12] = [Forms]![participa3]![APELLIDOS]  
Me.[Texto14] = [Forms]![participa3]![PRODUCCION]  
Me.[Texto16] = [Forms]![participa3]![FECHA]  
Me.Texto30 = [Forms]![participa3]![TELEFONO]  
Me.Texto28 = [Forms]![participa3]![TELEFONO_1]  
Me.Texto26 = [Forms]![participa3]![TELEFONO_2]
```

End Sub

Hacemos uso de las funciones del sistema HoraActual() y Fecha() para pasarlas como valor predeterminado a los campos "HORA" y "FECHA\_LLAMADA" de la tabla "llamadas" respectivamente.

[Ver el formulario.]

A continuación pego y comento el resto del código.

```
Option Compare Database  
Option Explicit
```

[variables que intervienen en el RecordSource de este formulario.]

Dim elect As String  
Dim rom As String  
Dim here As String  
Dim rder As String  
Dim fin As String

Private Sub FECHA\_RODAJE\_GotFocus()

[Añade un registro al formulario.]

[Ver el formulario.]

Recordset.AddNew

Me.IDRODAJE = [Forms]![participa3]![participa\_IDRODAJE]  
Me.IDTRABAJADOR = [Forms]![participa3]![participa\_IDPERSONAL]  
Me.FECHA\_RODAJE = [Forms]![participa3]![participa\_FECHA]

End Sub

Private Sub INCIDENCIAS\_Change()

[Abre el outlook.]

[Ver el formulario.]

If INCIDENCIAS = "mensaje buzón" Then

FollowHyperlink "mailto:" & [Forms]![participa3]![CORREO\_ELECTRONICO]

End If

End Sub

[Las siguientes funciones nos permiten abrir otros formularios con información adicional. Todos ellos tienen una estructura similar al que estamos estudiando y por supuesto su origen de datos está relacionado con el registro seleccionado en el formulario que muestra la relación varios a varios.]

[Ver el formulario.]

Private Function composit()

Dim stDocName As String  
Dim stLinkCriteria As String

```
stDocName = "composit"  
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

End Function

Private Function i\_rapida()

```
Dim stDocName As String  
Dim stLinkCriteria As String
```

```
stDocName = "participa3"  
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

End Function

Private Function parametros()

```
Dim stDocName As String  
Dim stLinkCriteria As String
```

```
stDocName = "parametros"  
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

End Function

Private Function rodajes()

```
Dim stDocName As String  
Dim stLinkCriteria As String
```

```
ecordSource_llamada2_rodaje3 = True  
ecordSource_personal1_rodaje3 = False  
ecordSource_productora2_rodaje3 = False  
ecordSource_participa3_rodaje3 = False
```

```
stDocName = "rodaje3"  
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

End Function

Private Function trabajadores()

```
Dim stDocName As String  
Dim stLinkCriteria As String
```

```
stDocName = "personal1"  
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

End Function

Private Function FALTAS()

Dim stDocName As String

Dim stLinkCriteria As String

stDocName = "faltas"

DoCmd.OpenForm stDocName, , , stLinkCriteria

End Function

Para terminar puedes ver dos formularios de un diseño practicamente igual utilizados para registrar operaciones contables y su origen de datos.

```
SELECT rodaje.PRODUCCION, cuenta_trabajadores.FECHA_RODAJE,
productora.IDPRODUCTORA, productora.NOMBRE_PRODUCTORA,
cuenta_trabajadores.IDPERSONAL, cuenta_trabajadores.IDRODAJE,
cuenta_trabajadores.ABONO, cuenta_trabajadores.COMENTARIOS,
cuenta_trabajadores.FECHA_APUNTE FROM (productora INNER JOIN rodaje ON
productora.IDPRODUCTORA=rodaje.IDPRODUCTORA) INNER JOIN cuenta_trabajadores ON
rodaje.IDRODAJE=cuenta_trabajadores.IDRODAJE WHERE
(((cuenta_trabajadores.IDPERSONAL)=Forms!participa3!participa_IDPERSONAL) And
((cuenta_trabajadores.IDRODAJE)=Forms!participa3!participa_IDRODAJE));
```

[Ver el formulario.]

```
SELECT cuenta_productoras.IDPERSONAL, cuenta_productoras.IDRODAJE,
cuenta_productoras.CARGO, cuenta_productoras.COMENTARIOS, rodaje.PRODUCCION,
cuenta_productoras.FECHA_RODAJE, productora.IDPRODUCTORA,
productora.NOMBRE_PRODUCTORA, cuenta_productoras.FECHA_APUNTE FROM
(productora INNER JOIN rodaje ON productora.IDPRODUCTORA=rodaje.IDPRODUCTORA)
INNER JOIN cuenta_productoras ON rodaje.IDRODAJE=cuenta_productoras.IDRODAJE
WHERE (((cuenta_productoras.IDPERSONAL)=Forms!participa3!participa_IDPERSONAL) And
((cuenta_productoras.IDRODAJE)=Forms!participa3!participa_IDRODAJE));
```

[Ver el formulario.]