

Universidad Tecnológica Nacional  
Facultad Regional Córdoba

Departamento de  
Ingeniería en Sistemas de Información

Cátedra  
Programación aplicaciones Visuales I

Trabajo Práctico 1



## TRABAJO PRÁCTICO 7 “MODELO FACTURA/REMITO/ORDEN DE COMPRA/ORDEN DE PEDIDO/TICKET”

### OBJETIVO:

1. Entender el diseño y programación del modelo Factura.
2. Comprender el concepto de grabación de multi-tablas.
3. Desarrollar la programación de recuperación de información de una Factura existente.
4. Desarrollar la programación de modificación y grabación de una Factura existente.
5. Utilización de los componentes vistos en el trabajo práctico 6, clase de acceso a datos.

### DISEÑE UN FORMULARIO CON LAS SIGUIENTES CARACTERÍSTICAS

Orden de Pedidos

Fecha

Sucursal  Nro Pedido

Cliente  ...

Artículo  ...

Cantidad  Precio  Total  Agregar

Cantidad	Id	Artículo	Precio	Subtotal
----------	----	----------	--------	----------

SubTotal

Decuento

Total



### CARACTERÍSTICAS:

1. En el Modelo Factura el ABM se realizar con dos tablas o más, según la sofisticación alcanzada en modelo, en contraposición con los trabajos prácticos anteriores donde el ABM se realizaba sobre una única tabla. La cabecera de factura, llamada habitualmente tabla "Factura", y el detalle de ésta, denominado "Detalle de Factura", son las dos tablas sobre las cuales se realizará el proceso de ABM.
2. El modelo factura comparte las mismas características, con sutiles variantes y formas, que los siguientes modelos:
  - 2.1. Remitos.
  - 2.2. Pedidos de compras o Nota de pedidos u Orden de Compra.
  - 2.3. Ticket de supermercado, o farmacia, o bar, etc.
3. Distribución de los datos en las tablas, en relación a la distribución de datos dentro del documento factura:
  - 3.1. Si observa una factura, dos áreas definidas:
    - 3.1.1. La que se denomina **cabecera de factura**, que también incluye el pie de la factura, que contiene los datos:
      - 3.1.1.1. número de sucursal.
      - 3.1.1.2. número de factura.
      - 3.1.1.3. fecha.
      - 3.1.1.4. tipo de factura.
      - 3.1.1.5. cliente.
      - 3.1.1.6. domicilio del cliente.
      - 3.1.1.7. condiciones fiscales.
      - 3.1.1.8. cuit-cuil.
      - 3.1.1.9. remito.
      - 3.1.1.10. condiciones de venta.
      - 3.1.1.11. en el pie, subtotal, impuestos, % iva, otros y total
    - 3.1.2. La denominada **detalla de factura**, que tiene un formato de planilla en donde se puede ingresar n líneas, que contiene la siguiente información:
      - 3.1.2.1. cantidad.
      - 3.1.2.2. descripción del artículo.
      - 3.1.2.3. precio unitario.
      - 3.1.2.4. precio total.
    - 3.1.3. En función de las dos distribuciones anteriores, se tendrá dos tablas claramente definidas, una denominada "**Factura**" y otra denominada "**Detalle de Factura**".

Cant.	DESCRIPCION	Precio U.	Precio Total
1000	AZUCAR (Kg)	1,28	1280
1000	FIDEOS (Kg)	1,14	1140
1000	LECHE (Lts)	6,30	630
1000	ACEITE (0.800Lts)	1,79	1790
1000	YERBA (Kg)	1,41	1410
1000	MANZANA (Kg)	0,80	800
1000	LENTIGAS (Kg)	1,70	1700
1000	ARROZ	0,69	690
1000	CROBUCA - (CARNE)	1,40	670
			2
Sub-Total			
Impuestos			
Sub-Total			
I.V.A. Insc. %			
I.V.A. No Insc. %			
TOTAL \$			10110.00

3.1.4. Dentro del modelo a estudiar, se desarrollará un **ABM Pedidos**, por consiguiente la tabla de cabecera se llamará **t\_pedidos**, siendo la tabla de detalle **t\_detalle\_pedidos**.

#### 4. Mecánica del formulario:

Cuando oprime el botón de 3 puntos, se busca el cliente, cuyo código está en la caja de textos de la izquierda

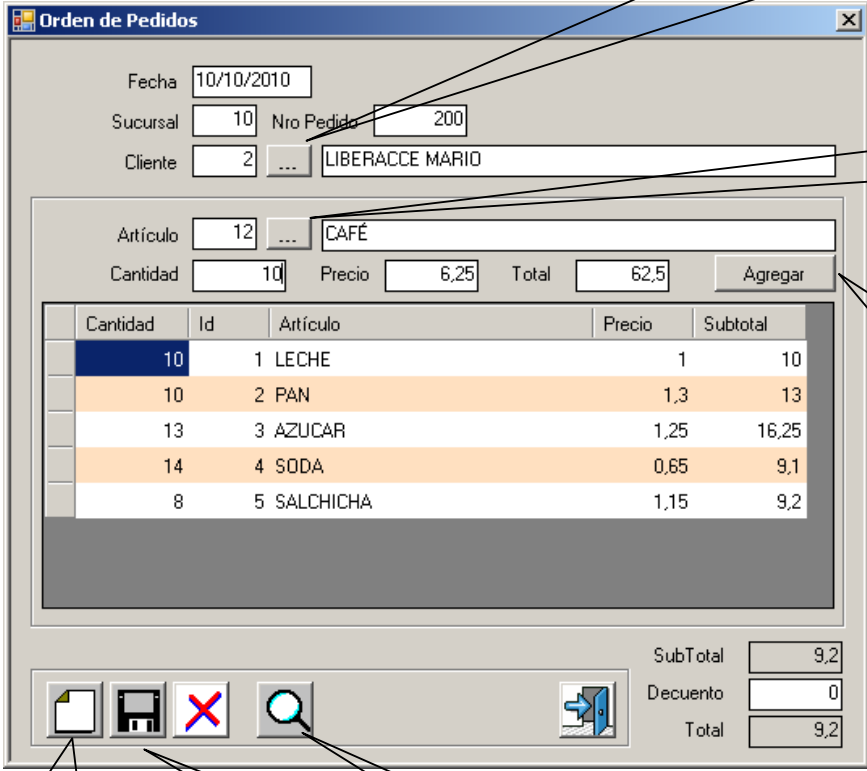
Igual que el botón de búsqueda de clientes, pero ahora con artículos. Se debe poder cargar varios artículos en la lista

Grilla que contiene los artículos que se cargaron en este pedido. (DataGridView)

Botón nuevo, debe poder blanquear todos los objetos que se utilicen para cargar datos.

Botón buscar, permite cargar otro formulario para realizar búsqueda de

Botón grabar, permite grabar los datos cargados en pantalla.



Cantidad	Id	Artículo	Precio	Subtotal
10	1	LECHE	1	10
10	2	PAN	1,3	13
13	3	AZUCAR	1,25	16,25
14	4	SODA	0,65	9,1
8	5	SALCHICHA	1,15	9,2

SubTotal: 9,2  
Descuento: 0  
Total: 9,2

- 4.1. Los objetos debe estar ordenados según un orden de desplazamiento de cursor, de arriba hacia abajo, de izquierda a derecha.
- 4.2. Cuando oprima el botón de búsqueda (botón de tres puntos), correspondiente al cliente, debe buscar a éste de la siguiente forma:
  - 4.2.1. Se debe cargar en la caja de texto que está ubicada a la izquierda del botón un número.
  - 4.2.2. Se oprime el botón de tres puntos.
  - 4.2.3. En caso de encontrar un cliente con ese código se debe cargará el nombre del cliente en la caja de texto que se encuentra a la derecha de botón de tres puntos, correspondiente a la columna **n\_cliente** de la tabla **t\_clientes**.

- 4.2.4. En caso de no encontrar un cliente con ese código, debe emitir un mensaje que diga, “el código ingresado no corresponde a un cliente”.
- 4.3. Al oprimir el botón de búsqueda (de tres puntos) del artículo, buscará los datos de uno, del mismo modo que se hizo con los datos del cliente.
- 4.4. La grilla debe poder almacenar n registros de detalle de pedido. Estos datos se almacenarán en la grilla, cuando oprima el botón agregar.
- 4.5. Botón nuevo: Blanquea los objetos del formulario.
- 4.6. Botón grabar: Permite la grabación de un registro nuevo o de una modificación de un registro existente.
  - 4.6.1. La grabación de los registro de la **pedido** y el **detalle de pedido** se deben realizar en una sola acción dentro del botón grabar.
  - 4.6.2. Primero. Debe grabar el **pedido**, utilice la clase de acceso a datos que ya se estudio en el trabaja práctico anterior.
  - 4.6.3. Segundo. Grabe el **detalle de pedido**, recuerde que se deben grabar n registros, tantos como los que figuran en la grilla.
  - 4.6.4. Cuando grave el **detalle de un pedido**, como medida restrictiva de integridad de la base de datos, no puede grabar dos registros con el mismo código de artículo. **Piense en una programación que le permita salvar esta situación.**
- 4.7. Botón buscar: debe disparar el funcionamiento de un formulario que le permita recuperar los datos. Este formulario le debe permitir elegir en una lista un pedido ya realizado, debe tener las características de que se muestra a continuación.



Nro Pedido	Nro Sucursal	Fecha	Cliente	Total
20	1	09/10/200...	LIBERACCE MARIO	38,5
30	1	09/10/200...	QUITO SERGIO	41,25
200	200	10/10/201...	LIBERACCE MARIO	37
200	10	10/10/201...	LIBERACCE MARIO	9,2
*				

5. Hasta este punto, el de recuperar la información, se le entregará la programación, siendo ésta una de las posibilidades, de tantas, para lograr esta tarea.



## 1. TRABAJO PRÁCTICO A REALIZAR POR USTED:

- 1.1. Mostrar los datos recuperados en el formulario principal.
- 1.2. Realizar la programación que corresponda para permitir que estos datos se puedan modificar.
  - 1.2.1. Contemplar que no se puedan modificar ni el número de sucursal, ni pedido, que forman parte de la PK de ambas tablas "t\_pedidos" y "t\_detalle\_pedido".
  - 1.2.2. Para poder diferenciar si se encuentra grabando un nuevo registro, o modificando un ya existente, debe implementar un mecanismo de control que le permita saber cual es la tarea que esta realizando. Por ejemplo una bandera de tipo lógico (boolean).
  - 1.2.3. Grabar la modificación del pedido y también la del detalle modificado. Recuerde que el comando SQL para realizar esta tarea es "Update".
- 1.3. Cuando realice doble clic sobre un elemento de la grilla que contiene el detalle del pedido dentro el formulario principal, se debe borrar ese elemento de la grilla, antes de borrarlo, se debe realizar una pregunta con un mensaje, solicitando que se confirme esta acción o no.

**IMPORTANTE:** Recuerde que los eventos, como `_cmd_buscar_cliente_click`, no debe escribirlos Ud., sino, deben resultar de la interacción con el formulario, al hacer doble clic sobre el objeto correspondiente. Con esto disminuirá sensiblemente los errores de referencia de código. Los métodos como `_calcular_total`, si los debe escribir Ud., pues son programación específica del diseño del programador.

### PROGRAMACIÓN DE FORMULARIO PRINCIPAL:

```
Public Class frm_orden_pedido
    Enum _resultado
        exitoso
        erroneo
    End Enum
    Dim _acceso As New acceso_a_datos("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=F:\utn\PAV1_VB.NET\clases\tp07_modelo_factura\DBF.mdb", acceso_a_datos.motores.access)
    Private Sub _cmd_buscar_cliente_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
_cmd_buscar_cliente.Click

        If Me._txt_id_cliente.Text = "" Then
            MsgBox("falta el código de cliente", MsgBoxStyle.Critical, "Importante")
            Exit Sub
        End If

        buscar_cliente(Me._txt_id_cliente.Text)

    End Sub

    Private Sub buscar_cliente(ByVal id As String)

        Dim tabla As New Data.DataTable
        Me._acceso._tabla = "t_clientes"
        tabla = Me._acceso.leo_tabla("id_cliente = " & Me._txt_id_cliente.Text)

        If tabla.Rows.Count > 0 Then
            Me._txt_cliente.Text = tabla.Rows(0)("n_cliente").ToString
        End If
    End Sub
End Class
```



```
Else
    Me._txt_cliente.Text = ""
    MsgBox("Ese código de cliente no existe", MsgBoxStyle.Critical, "Importante")
End If

End Sub

- - - - -
Private Sub _cmd_buscar_articulo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
_cmd_buscar_articulo.Click
    If Me._txt_id_articulo.Text = "" Then
        MsgBox("El Código del artículo esta vacío", MsgBoxStyle.Critical, "Importante")
        Exit Sub
    End If

    Dim tabla As New Data.DataTable
    Me._acceso._tabla = "t_articulos"
    tabla = Me._acceso.leo_tabla("id_articulo = " & Me._txt_id_articulo.Text.Trim())

    If tabla.Rows.Count > 0 Then
        Me._txt_articulo.Text = tabla.Rows(0)("n_articulo")
        Me._txt_precio.Text = tabla.Rows(0)("precio")
    Else
        MsgBox("No existe ese código de artículos", MsgBoxStyle.Critical, "Importante")
    End If

End Sub

- - - - -
Private Sub _cmd_agregar_grilla_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
_cmd_agregar_grilla.Click
    Me.grilla1.Rows.Add(Me._txt_cantidad.Text, Me._txt_id_articulo.Text, Me._txt_articulo.Text, Me._txt_precio.Text,
Me._txt_Subtotal.Text)
    Me._txt_sub_total.Text = Me._calcular_total()
    Me._txt_total.Text = Me._txt_sub_total.Text
End Sub

- - - - -
Private Function _calcular_total() As Double
    Dim c As Integer
    Dim calculo As Double
    calculo = 0
    For c = 0 To Me.grilla1.RowCount - 1
        calculo = calculo + Convert.ToDouble(Me.grilla1.Rows(c).Cells(4).Value)
    Next
    Return calculo
End Function

- - - - -
Private Sub _graba_pedido()
    Dim txt As String

    txt = "nro_pedido=" & Me._txt_pedido.Text
    txt &= ", nro_sucursal=" & Me._txt_sucursal.Text
    txt &= ", fecha=" & Me._txt_fecha.Text
    txt &= ", id_cliente=" & Me._txt_id_cliente.Text
    txt &= ", sub_total=" & Me._txt_Subtotal.Text.Replace(",", ".")
    txt &= ", descuento=" & Me._txt_descuento.Text.Replace(",", ".")
    txt &= ", total=" & Me._txt_total.Text.Replace(",", ".")

    Me._acceso._tabla = "t_pedidos"
```



```
Me._acceso._insertar(txt)
```

```
End Sub
```

```
Private Sub _grabar_detalle()
```

```
Dim c As Integer
```

```
Dim txt_insert As String = ""
```

```
c = 0
```

```
Me._acceso._tabla = "t_detalle_pedidos"
```

```
For c = 0 To Me.grilla1.RowCount() - 1
```

```
txt_insert &= "nro_pedido=" & Me._txt_pedido.Text
```

```
txt_insert &= ", nro_sucursal=" & Me._txt_sucursal.Text
```

```
txt_insert &= ", id_articulo=" & Me.grilla1.Rows(c).Cells("id").Value
```

```
txt_insert &= ", cantidad=" & Me.grilla1.Rows(c).Cells("cantidad").Value.ToString.Replace(",", ".")
```

```
txt_insert &= ", precio=" & Me.grilla1.Rows(c).Cells("precio").Value.ToString.Replace(",", ".")
```

```
Me._acceso._insertar(txt_insert)
```

```
txt_insert = ""
```

```
Next
```

```
End Sub
```

```
Private Sub cmd_grabar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmd_grabar.Click
```

```
Me._graba_pedido()
```

```
Me._grabar_detalle()
```

```
End Sub
```

```
Private Sub _txt_cantidad_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles _txt_cantidad.TextChanged
```

```
Dim a, b As Double
```

```
a = Val(Me._txt_precio.Text.Replace(",", "."))
```

```
b = Val(Me._txt_cantidad.Text.Replace(",", "."))
```

```
Me._txt_Subtotal.Text = a * b
```

```
End Sub
```

```
Private Sub frm_orden_pedido_FormClosing(ByVal sender As System.Object, ByVal e As
```

```
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
```

```
If MsgBox("Esta seguro que desea cerrar el formulario", MsgBoxStyle.Critical + MsgBoxStyle.OkCancel, "Importante") =
```

```
MsgBoxResult.Ok Then
```

```
e.Cancel = False
```

```
Else
```

```
e.Cancel = True
```

```
End If
```

```
End Sub
```

```
Private Sub limpiar(ByVal de_donde As Object)
```

```
Dim gri As DataGridView
```

```
For Each obj As System.Windows.Forms.Control In de_donde
```

```
'MsgBox(obj.Name)
```

```
Me.grilla1.Rows.Clear()
```

```
Select Case obj.GetType().ToString
```

```
Case "System.Windows.Forms.TextBox"
```

```
obj.Text = ""
```





```
Case "System.Windows.Forms.MaskedTextBox"
    obj.Text = ""
Case "System.Windows.Forms.DataGridView"
    gri = obj
    gri.Rows.Clear()
Case "System.Windows.Forms.GroupBox"
    Me.limpiar(obj.Controls)
End Select
Next obj
End Sub

Private Sub _cmd_buscar_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles _cmd_buscar.Click
    Me.limpiar(Me.Controls)
    frm_buscar_pedido.ShowDialog()
    If Me._txt_pedido.Text <> "" Then
        Me.recuperar_datos()
    End If
End Sub

Private Sub cmd_nuevo_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmd_nuevo.Click
    Me.limpiar(Me.Controls)
    Me._txt_fecha.Focus()
End Sub

Private Sub cmd_salir_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmd_salir.Click
    Me.Close()
End Sub
End Class
```

#### **PROGRAMA DE FORMULARIO DE BÚSQUEDA DE PEDIDOS EXISTENTES:**

```
Private Sub frm_buscar_pedido_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Me.buscar_datos()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Me.Close()
End Sub

Private Sub _grid1_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles _grid1.DoubleClick

    frm_orden_pedido._txt_sucursal.Text = Me._grid1.CurrentRow.Cells(1).Value
    frm_orden_pedido._txt_pedido.Text = Me._grid1.CurrentRow.Cells(0).Value

    Me.Close()

End Sub

Private Sub buscar_datos()
    Dim _acceso1 As New acceso_a_datos("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=F:\utn\PAV1_VB.NET\clases\tp07_modelo_factura\DBF.mdb", acceso_a_datos.motores.access)

    Dim sql As String
    Dim tablaL As New Data.DataTable

    sql = "SELECT t_pedidos.nro_pedido, t_pedidos.nro_sucursal, t_pedidos.fecha, t_clientes.n_cliente, t_pedidos.total " & _
```



```
" FROM (t_pedidos INNER JOIN " & _  
"t_clientes ON t_pedidos.id_cliente = t_clientes.id_cliente)"  
  
tablaL = _acceso1._consulta(sql)  
  
'Me._grid1.DataSource = tablaL  
  
Dim c As Integer  
  
Me._grid1.Rows.Clear()  
  
For c = 0 To tablaL.Rows.Count() - 1  
    Me._grid1.Rows.Add(tablaL.Rows(c)("nro_pedido"), tablaL.Rows(c)("nro_sucursal"), tablaL.Rows(c)("fecha") _  
        , tablaL.Rows(c)("n_cliente"), tablaL.Rows(c)("total"))  
Next  
  
End Sub
```