



Cuadernillo de examen

ASIGNATURA	PROGRAMAS DE APLICACIÓN II PROGRAMAS DE APLICACIÓN III	CÓDIGO	305 305
CONVOCATORIA	Junio 1999 (2º Parcial)	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Mañana	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1998/99
DURACIÓN APROXIMADA	2 horas		

APELLIDOS: NOMBRE:

GRUPO: NÚMERO DE EXPEDIENTE:

Parte teórica

1. ¿Para que se utiliza la etiqueta de HTML OBJECT. Explique la importancia de esta etiqueta para utilizar componentes ActiveX. Comente el contenido de los distintos parámetros que la acompañan.
2. ¿Qué son los controles ActiveX? Indique todos los elementos que caracterizan la creación de un control ActiveX.
3. Explique las posibles formas de hacer que una aplicación desarrollada en Visual Basic 5.0 pueda acceder a Internet.
4. Explique las dos formas posibles de insertar elementos multimedia en un proyecto de Visual Basic 5.0.

Parte práctica

1. Una empresa mantiene sus datos en una base de datos Access (comercial.mdb) y desea gestionar los pedidos de sus clientes. Para ello, entre otras, dispone de dos tablas:

Tabla: Pedidos		
Nombre Campo	Formato	Observaciones
IDPedido	Entero largo	Clave primaria. Identifica unívocamente el pedido.
IDCliente	Texto, 5 posiciones	Identifica al cliente que ha realizado el pedido. Es el campo del índice idxCliente. Admite duplicados (un cliente puede tener varios pedidos)
FechaPedido	Fecha/Hora	Fecha de realización del pedido
FechaEntrega	Fecha/Hora	Fecha de la entrega del pedido
FechaPago	Fecha/Hora	Fecha del pago de la factura

Tabla: DetallesPedidos		
Nombre Campo	Formato	Observaciones
IDPedido	Entero largo	Identifica al pedido al que pertenece el detalle. Es el campo del índice idxPedido. Admite duplicados (un pedido puede tener varias líneas de detalle)
IDProducto	Entero largo	Identifica al producto del detalle del pedido. Es el campo del índice idxProducto. Admite duplicados (un producto puede aparecer en varias líneas de detalle)
PrecioUnidad	Monetario	Precio unitario del producto en el momento de hacer el pedido
Cantidad	Entero	Unidades pedidas
Pagado	Lógico	Verdadero si ese detalle se ha pagado

La empresa desea realizar un formulario como parte de una aplicación que gestione los pagos de los pedidos por parte del cliente. El formulario de entrada de datos tendrá el siguiente aspecto:



Cuadernillo de examen

ASIGNATURA	PROGRAMAS DE APLICACIÓN II PROGRAMAS DE APLICACIÓN III	CÓDIGO	305 305
CONVOCATORIA	Junio 1999 (2º Parcial)	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Mañana	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1998/99
DURACIÓN APROXIMADA	2 horas		

APELLIDOS: NOMBRE:

GRUPO: NÚMERO DE EXPEDIENTE:

Notas importantes: Sólo existen los controles que aparecen en el formulario.

Todos los accesos a datos se deben hacer mediante objetos del DAO.

Se deben codificar los procedimientos y declaraciones necesarios para:

- A. Buscar todos los pedidos del cliente introducido en el cuadro de texto txtIDCliente. Si no tiene pedidos, un mensaje de error nos informará de la situación. En caso contrario los datos de los pedidos aparecerán en los cuadros de texto del *frame* Pedidos.

Puntuación: 1,25 puntos.

- B. Gestionar el movimiento por los pedidos del cliente mediante los botones Primero, Ultimo, Siguiente y Anterior.

Puntuación: 1 punto.

- C. Cuando el usuario pulse sobre el botón Pagar, el campo Pagado de cada uno de los detalles del pedido deberá ponerse a Verdadero. Así mismo se deberá actualizar la Fecha de pago, con el valor de la fecha actual.

Puntuación: 1,25 puntos.

Solución

```
Option Explicit
Dim db As Database
Dim rs As Recordset
```

```
Private Sub cmdAnterior_Click()
    rs.MovePrevious
    If rs.BOF Then
        MsgBox "No hay más pedidos", vbInformation, "Pagos de pedidos"
        rs.MoveFirst
    Else
```



Cuadernillo de examen

ASIGNATURA	PROGRAMAS DE APLICACIÓN II PROGRAMAS DE APLICACIÓN III	CÓDIGO	305 305
CONVOCATORIA	Junio 1999 (2º Parcial)	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Mañana	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1998/99
DURACIÓN APROXIMADA	2 horas		

APELLIDOS: NOMBRE:

GRUPO: NÚMERO DE EXPEDIENTE:

```
        cargarcampos
    End If

End Sub

Private Sub cmdBuscar_Click()
    Dim Qry As String

    Qry = "SELECT IdPedido, IdCliente, FechaPedido, FechaEntrega, FechaPago "
    Qry = Qry & " FROM Pedidos WHERE IdCliente = '" & txtIDCliente & "'"
    Set rs = db.OpenRecordset(Qry)

    If rs.RecordCount = 0 Then
        MsgBox "No se encontraron pedidos", vbInformation, _
            "Pagos de pedidos"
    Else
        cargarcampos
    End If

End Sub

Private Sub cmdPagar_Click()
    Dim sql As String

    'Modificar los detalles de pedidos
    sql = "UPDATE DetallesPedidos SET Pagado=True "
    sql = sql & "WHERE IdPedido = " & txtIDPedido
    db.Execute sql

    'Actualizar la fecha de pago
    rs.Edit
    rs("FechaPago") = Date
    rs.Update
    cargarcampos

End Sub

Private Sub cmdPrimero_Click()
    rs.MoveFirst
    cargarcampos
End Sub

Private Sub cmdSiguiente_Click()
    rs.MoveNext
    If rs.EOF Then
        MsgBox "No hay más pedidos", vbInformation, "Pagos de pedidos"
    rs.MoveLast
```



Cuadernillo de examen

ASIGNATURA	PROGRAMAS DE APLICACIÓN II PROGRAMAS DE APLICACIÓN III	CÓDIGO	305 305
CONVOCATORIA	Junio 1999 (2º Parcial)	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Mañana	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1998/99
DURACIÓN APROXIMADA	2 horas		

APELLIDOS: NOMBRE:

GRUPO: NÚMERO DE EXPEDIENTE:

```

Else
    cargarcampos
End If
End Sub

Private Sub cmdUltimo_Click()
    rs.MoveLast
    cargarcampos
End Sub

Private Sub Form_Load()
    Set db = DBEngine.Workspaces(0).OpenDatabase("\\luis\bd\comercial.mdb")
End Sub

Sub cargarcampos()
    txtIDPedido = rs("IDPedido")
    txtFechaPedido = rs("FechaPedido")
    txtFechaEntrega = rs("FechaEntrega")
    If Not IsNull(rs("FechaPago")) Then
        txtFechaPago = rs("FechaPago")
    Else
        txtFechaPago = Empty
    End If
End Sub

```

2. Implemente un control ActiveX (XCombo) que amplie las capacidades de un control ComboBox en este control, cuando el usuario introduzca una letra, deberá buscar si es la letra inicial de cualquiera de los elementos del combo. De ser así aparecerá el primer elemento con dicha que posea dicha letra inicial en el cuadro de texto. En caso contrario se permitirá que el usuario introduzca el texto que desee.

Por ejemplo, si el combo contiene los valores "Amarillo", "Azul", "Naranja", "Verde" y "Violeta", si se introduce en el control la letra "V" aparecerá en el cuadro de texto la palabra "Verde" (primer elemento de la lista que empieza por "V"). Si se introduce cualquier letra distinta, por ejemplo una "L", el usuario del control podrá introducir el texto que desee. Además, siempre será posible desplegar el combo para seleccionar de la lista.

El alumno deberá crear una propiedad Text que corresponda a la propiedad del control constituyente, los métodos AddItem y RemoveItem que se corresponden con los del control y los eventos Click y Change, así como la función privada que permita buscar el elemento entre los elementos del combo.

Puntuación: 2,5 puntos

Solución

(Nota: El nombre del control combo del objeto UserControl es cmbExtended. El nombre del UserControl es Xcombo)

```

Option Explicit
Option Compare Text

```



Cuadernillo de examen

ASIGNATURA	PROGRAMAS DE APLICACIÓN II PROGRAMAS DE APLICACIÓN III	CÓDIGO	305 305
CONVOCATORIA	Junio 1999 (2º Parcial)	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Mañana	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1998/99
DURACIÓN APROXIMADA	2 horas		

APELLIDOS: NOMBRE:

GRUPO: NÚMERO DE EXPEDIENTE:

'Event Declarations:

Event Change()

Event Click()

```
Private Sub UserControl_Resize()
    cmbExtended.Move 0, 0, ScaleWidth
End Sub
```

```
Public Sub AddItem(Item As String, Optional Index As Variant)
    cmbExtended.AddItem Item, Index
End Sub
```

```
Public Sub RemoveItem(Index As Integer)
    cmbExtended.RemoveItem Index
End Sub
```

```
Public Property Get Text() As String
    Text = cmbExtended.Text
End Property
```

```
Public Property Let Text(ByVal New_Text As String)
    cmbExtended.Text() = New_Text
    PropertyChanged "Text"
End Property
```

```
' Cargar valores de propiedades desde el almacenamiento
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    cmbExtended.Text = PropBag.ReadProperty("Text", "")
End Sub
```

```
' Escribir valores de propiedades en el almacenamiento
Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    Call PropBag.WriteProperty("Text", cmbExtended.Text, "")
End Sub
```

```
Private Sub cmbExtended_Change()
    Dim p As Integer

    p = FindItem(cmbExtended.Text & "*")
    cmbExtended.ListIndex = p

    RaiseEvent Change
End Sub
```

```
Private Sub cmbExtended_Click()
    RaiseEvent Click
End Sub
```



Cuadernillo de examen

ASIGNATURA	PROGRAMAS DE APLICACIÓN II PROGRAMAS DE APLICACIÓN III	CÓDIGO	305 305
CONVOCATORIA	Junio 1999 (2º Parcial)	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Mañana	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1998/99
DURACIÓN APROXIMADA	2 horas		

APELLIDOS: NOMBRE:

GRUPO: NÚMERO DE EXPEDIENTE:

End Sub

```
Private Function FindItem(Item As String) As Integer
    Dim i As Integer

    For i = 0 To cmbExtended.ListCount - 1
        If cmbExtended.List(i) Like Item Then
            Exit For
        End If
    Next

    If i = cmbExtended.ListCount Then
        FindItem = -1
    Else
        FindItem = i
    End If

End Function
```