



ASIG

NATURA	Programas de Aplicación II/III	CÓDIGO	305
CONVOCATORIA	Septiembre de 2000	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNOS	Tarde	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1999/2000
DURACIÓN APROXIMADA	2 horas		

Soluciones al examen

Pruebas teórico-prácticas

1. El control CommonDialog ¿Qué funciones tiene? Propiedades y métodos más importantes.

Práctica

Escriba el código necesario para que, mediante un control CommonDialog se pueda seleccionar un archivo gráfico. En el diálogo, si se pulsa el botón Aceptar, se cargará el archivo seleccionado en el control Picture1, si se pulsa el botón Cancelar se cargará en dicho control el archivo Default.jpg.

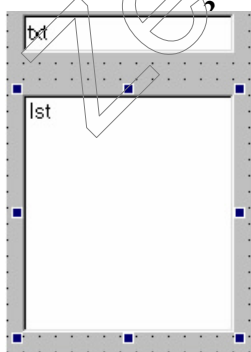
```
.  
. .  
On Error Resume Next  
dlg.CancelError = True  
dlg.ShowOpen  
If Err <> 0 then  
    Picture1.Picture = LoadPicture("Default.jpg")  
Else  
    Picture1.Picture = LoadPicture(dlg.FileName)  
End If  
. .  
.
```

2. Búsqueda de registros en el modelo DAO. Explique todos los métodos de búsqueda sobre objetos DAO con su formato e indicando cuando habría que aplicar cada uno de ellos.
3. Describa las analogías y diferencias entre los Documentos ActiveX y los controles ActiveX.

Puntuación: 1 punto cada pregunta

Pruebas prácticas

1. Se desea diseñar un control ActiveX (TextBoxListBox) compuesto de un control ListBox y un control TextBox.



El control debe utilizar la propiedad Text y el evento Click del control ListBox. Se implementará en el control un método AddText que insertará en el ListBox el contenido del control txt y un método RemoveText que eliminará, buscándolo en la lista, el contenido del control txt, siempre y cuando dicho valor exista en la lista. Si no existe no hará nada.

Puntuación: 3 puntos.

```
Option Explicit  
'Declaraciones de eventos  
Event Click()
```



Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

```
Private Sub UserControl_Resize()
    txt.Move 0, 0, ScaleWidth
    lst.Move 0, txt.Height + 100, ScaleWidth, ScaleHeight - txt.Height + 100
End Sub

Private Sub lst_Click()
    RaiseEvent Click
End Sub

Public Property Get Font() As Font
    Set Font = lst.Font
End Property

Public Property Set Font(ByVal New_Font As Font)
    Set lst.Font = New_Font
    PropertyChanged "Font"
End Property

Public Property Get Text() As String
    Text = txt.Text
End Property

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

Public Sub AddText()
    lst.AddItem txt.Text
End Sub

Public Sub RemoveText()
    Dim i As Integer
    i = 0
    Do While i < lst.ListCount - 1 And lst.List(i) <> txt.Text
        i = i + 1
    Loop

    If lst.List(i) = txt.Text Then
        lst.RemoveItem i
    End If
End Sub

' Cargar valores de propiedad desde el almacén
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    Set lst.Font = PropBag.ReadProperty("Font", Ambient.Font)
    txt.Text = PropBag.ReadProperty("Text", "")
End Sub

' Escribir valores de propiedad en el almacén
Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    Call PropBag.WriteProperty("Font", lst.Font, Ambient.Font)
    Call PropBag.WriteProperty("Text", txt.Text, "")
End Sub
```

3. Una inmobiliaria mantiene los datos tanto de los pisos en venta como de sus clientes en una base de datos Access (casas.mdb) que, entre otras, tiene las siguientes tablas:

Tabla: Zonas (contiene información de sobre las distintas zonas de la ciudad)

Campo	Formato	Observaciones
IDZona	Entero largo	Identificador de la zona donde está el piso
NombreZona	Texto	Nombre de la zona
....	Existirían más datos describiendo la zona que no interesan a la aplicación

- 4.

Tabla: Pisos (contiene un registro por cada piso que ha entrado en la agencia)

Campo	Formato	Observaciones
-------	---------	---------------



Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

IDPiso	Autonumérico	Clave primaria de la tabla
IDZona	Entero largo	Identificador de la zona donde está el piso
Calle	Texto	
Número	Texto	
Piso	Texto	Contiene la escalera, el piso y la puerta del inmueble
Metros	Entero	
Precio	Entero	
Comentarios	Texto	Más informaciones sobre el piso
IDPropietario	Entero largo	Clave del propietario del piso. Lo relacionaría con la tabla Propietarios que no interesa a la aplicación

Tabla: Ofertas (contiene el identificador de los pisos que la inmobiliaria actualmente están en oferta)

Campo	Formato	Observaciones
IDPiso	Entero largo	Clave primaria
....	Existirían más datos que no interesan a la aplicación

Tabla: Clientes (contiene información de las personas que alguna vez se han interesado por un piso)

Campo	Formato	Observaciones
IDCliente	Autonumérico	Clave primaria de la tabla
Apellidos	Texto	
Nombre	Texto	
Telefono	Texto	
Email	Texto	

Tabla: Interesados (contiene información de las personas que se han interesado por algún piso)

Campo	Formato	Observaciones
IDInteres	Autonumérico	Clave primaria de la tabla
IDCliente	Entero largo	Identificador del cliente que se ha interesado por un piso
IDPiso	Entero largo	Identificador del piso por el que se ha interesado
Fecha	Fecha/Hora	Fecha en la que el cliente mostró interés por el piso

Utilizando estos datos se desea implementar un formulario que permita gestionar los pisos y los clientes. El formulario tendrá el siguiente formato:

Notas:

- Sólo existen los controles que aparecen en el formulario.
- La aplicación se debe hacer utilizando el modelo ADO.
- Los cuadros de texto sólo tienen establecida la propiedad Name. Las etiquetas tienen ya establecida la propiedad Caption. Los botones de órdenes tienen establecida tanto la propiedad Name como la propiedad Caption. El DataCombo únicamente tiene establecida la propiedad Name. El DataGrid tiene establecidas las propiedades Name y Caption, así como las propiedades Caption y DataField de las cinco columnas que se asocian a los campos Calle, Numero, Piso, Metros y Precio de la tabla Pisos.



- Si se desea enlazar cualquier control a una fuente de datos habrá que establecer las propiedades apropiadas, ya sea indicando cómo se haría en tiempo de diseño o en tiempo de ejecución mediante código.

Funcionalidades de la aplicación

- A. Cuando se cargue el formulario, deberán aparecer en el DataCombo dcZonas, todas las zonas a partir de la tabla Zonas.

```
Private Sub Form_Load()  
    'Crear la conexión  
    Set cn = New ADODB.Connection  
    cn.Provider = "Microsoft.Jet.OLEDB.3.51"  
    cn.ConnectionString = "\\luis\bd\casas.mdb"  
    cn.CursorLocation = adUseClient  
    cn.Open  
  
    'Crea un recorset con las zonas  
    Set rsZonas = New ADODB.Recordset  
    rsZonas.ActiveConnection = cn  
    rsZonas.Source = "Zonas"  
    rsZonas.Open  
  
    'Relaciona rsZonas con el control dcZonas  
    dcZonas.ListField = "NombreZona"  
    dcZonas.BoundColumn = "IDZona"  
    dcZonas.DataField = "IDZona"  
    Set dcZonas.RowSource = rsZonas
```

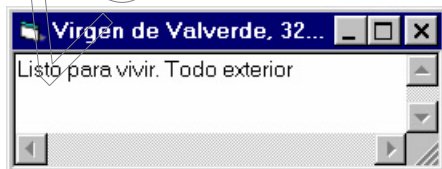
End Sub

- B. Al pulsar sobre el botón cmdBuscar, se cargarán en el DataGrid dgPisos todos los pisos en oferta en la zona seleccionada en el DataCombo (sólo aparecerán los pisos que estén incluidos en la tabla Ofertas).

```
Private Sub cmdBuscar_Click()  
    'Busca los pisos de la zona  
    Set rsPisos = New ADODB.Recordset  
    rsPisos.ActiveConnection = cn  
    rsPisos.Source = "SELECT * FROM Pisos,Ofertas " & _  
        "WHERE Pisos.IDPiso = Ofertas.IDPiso AND " & _  
        "IDZona = " & dcZonas.BoundText  
  
    rsPisos.Open  
  
    'Relaciona rsPisos con el DataCombo  
    Set dgPisos.DataSource = rsPisos
```

End Sub

- C. Al pulsar dos veces sobre cualquier fila del DataGrid, aparecerá un formulario frmComentarios. En el título del formulario aparecerá la dirección del piso. Además tendrá un cuadro de texto donde aparecerán los comentarios que pueda tener el piso.



```
Private Sub dgPisos_DblClick()  
    frmComentarios.Caption = rsPisos!Calle & ", " & rsPisos!Numero & _  
        " - " & rsPisos!Piso  
    frmComentarios.txtComentarios = rsPisos!Comentarios  
    frmComentarios.Show 1  
End Sub
```

- D. Cuando el control txtIDCliente pierda el foco se buscará el identificador de cliente de la tabla Clientes. Si el cliente existe aparecerán sus datos en los controles txtApellidos, txtNombre, txtTelefono y txtEmail,



bloqueándose al mismo tiempo los controles. Si el identificador de cliente no existe, esos controles se desbloquearán y aparecerán vacíos.

```
Private Sub txtIDCliente_LostFocus()  
    Set rsClientes = New ADODB.Recordset  
    rsClientes.ActiveConnection = cn  
    rsClientes.LockType = adLockOptimistic  
    rsClientes.CursorType = adOpenDynamic  
    rsClientes.Source = "SELECT * FROM Clientes " & _  
                        "WHERE IDCliente = " & txtIDCliente  
    rsClientes.Open  
    If rsClientes.RecordCount = 0 Then  
        'Se desbloquean los campos  
        txtApellidos.Locked = False  
        txtNombre.Locked = False  
        txtTelefono.Locked = False  
        txtEmail.Locked = False  
    Else  
        'Se bloquean los campos  
        txtApellidos.Locked = True  
        txtNombre.Locked = True  
        txtTelefono.Locked = True  
        txtEmail.Locked = True  
        'Se cargan los campos  
        txtApellidos = rsClientes!Apellidos  
        txtNombre = rsClientes!Nombre  
        txtTelefono = rsClientes!Telefono  
        txtEmail = rsClientes!Email  
    End If  
End Sub
```

- E. Al pulsar sobre el botón cmdIncluir, se incluirá un registro en la tabla Interesados. En dicho registro el valor del campo IDPiso será el del piso seleccionado en el DataCombo, el campo Fecha tomará el valor de la fecha actual y el campo IDCliente tomará el valor del campo IDCliente buscado en el paso anterior. Si en el paso anterior el cliente no existiera se dará también un alta en la tabla Clientes con los valores de los campos que tengan los controles txtApellidos, txtNombre, txtTelefono y txtEmail.

```
Private Sub cmdIncluir_Click()  
    Dim sql As String  
    Dim lIDCliente As Long  
  
    'Cliente nuevo  
    If rsClientes.RecordCount = 0 Then  
        'rsClientes.Close  
        'rsClientes.Source = "Clientes"  
        'rsClientes.Open  
        rsClientes.AddNew  
        lIDCliente = rsClientes!IDCliente  
        rsClientes!Apellidos = txtApellidos  
        rsClientes!Nombre = txtNombre  
        rsClientes!Telefono = txtTelefono  
        rsClientes!Email = txtEmail  
        rsClientes.Update  
        'Cierro y abro el recordset para  
        'averiguar el ID del último cliente  
        rsClientes.Close  
        rsClientes.Source = "Clientes"  
        rsClientes.Open  
        rsClientes.MoveLast  
    End If  
  
    'Se da un alta en Interesado  
    sql = "INSERT INTO Interesados " & _  
          "(IDCliente, IDPiso, Fecha) " & _  
          "VALUES (" & rsClientes!IDCliente & _  
          ", " & rsPisos("Pisos.IDPiso") & ",#" & Now() & "#)"
```



```
Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software  
cn.Execute sql  
End Sub
```

Otras declaraciones y procedimientos

```
Option Explicit  
Private cn As ADODB.Connection  
Private rsZonas As ADODB.Recordset  
Private rsPisos As ADODB.Recordset  
Private rsClientes As ADODB.Recordset
```

ZEON PDF DRIVER TRIAL
www.zeon.com.tw