



### Cuadernillo de examen

ASIGNATURA	Programas de Aplicación II/III	CÓDIGO	305
CONVOCATORIA	Ordinaria de junio de 2000 (Final)	PLAN DE ESTUDIOS	1994/1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Tarde	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	1999/2000
DURACIÓN APROXIMADA	2 horas		

### Soluciones al Examen

#### Pruebas teórico-prácticas

- Modelos de acceso a datos desde Visual Basic. Describa los distintos modelos de acceso a datos utilizando Visual Basic, explicando detalladamente las distintas capas existentes entre la aplicación y los datos en cada uno de los modelos.  
*Habría que explicar los modelos DAO, RDO y OLEDB con las distintas capas de cada uno de ellos.*
- Eventos del objeto UserControl. Enumere y describa los eventos que se producen al crear o destruir instancias de un control ActiveX.  
*Habría que explicar cuando se producen los eventos Initialize, InitProperties, ReadProperties, WriteProperties, Terminate, Paint y Resize cuando se pasa de un modo a otro (al colocar una instancia en el formulario, el paso de diseño a ejecución, de ejecución a diseño, etc.)*
- El modelo de objetos ADO. Describa los distintos objetos del modelo ADO, indicando sus relaciones y características.  
*Los siete objetos de ADO son Connection, Recordset, Command, Parameters, Fields, Error y Properties.*
- Indique las distintas formas en las que podemos utilizar Internet Explorer desde Visual Basic. Codifique de que forma podríamos utilizar el objeto Internet Explorer desde Visual Basic.

*Características más importantes del control Web Browser y el objeto Internet Explorer*

```
Dim ie as InternetExplorer
Set ie = New InternetExplorer

o

Dim ie as Object
Set ie = CreateObject("InternetExplorer.Application")
```

**Puntuación: 0,75 puntos cada pregunta**

#### Pruebas prácticas

- Se desea diseñar un control ActiveX (Numeric TextBox) compuesto únicamente de un cuadro de texto que permite introducir valores numéricos enteros positivos comprendidos entre dos límites (Max y Min). El control deberá exportar la propiedad Text del TextBox y el evento Click del mismo. Así mismo deberá tener dos propiedades Min y Max, con unos valores por omisión de 0 y 32767 respectivamente que indicarán los límites entre los que puede estar el valor del TextBox. Si el valor no estuviera entre dichos límites, un mensaje advertiría de la incidencia.

El control sólo podrá admitir los dígitos entre 0 y 9. Al introducir los valores de las propiedades Max y Min, deberemos comprobar que no sean negativos y que el valor de Min siempre sea menor o igual que el de Max.

**Puntuación: 2 puntos.**

```
Option Explicit
'Valores por omisión de las propiedades
Const m_def_Max = 32767
Const m_def_Min = 0
'Variables de las propiedades
Dim m_Max As Long
Dim m_Min As Long
```



```
'Declaraciones de evento
Event Click()

Private Sub txt_KeyPress(KeyAscii As Integer)
    'Si la tecla no está entre 0 y 9, no se escribe nada
    If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
        KeyAscii = 0
    End If
End Sub

Private Sub txt_LostFocus()
    If txt < Min Or txt > Max Then
        MsgBox "El valor debe estar entre " & Min & " y " & Max, _
            vbInformation, "Numeric TextBox"
        txt.SelStart = 0
        txt.SelLength = Len(txt)
        txt.SetFocus
    End If
End Sub

Private Sub UserControl_Resize()
    txt.Move 0, 0, ScaleWidth, ScaleHeight
End Sub

Private Sub txt_Click()
    RaiseEvent Click
End Sub

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 Property Get Max() As Long
    Max = m_Max
End Property

Public Property Let Max(ByVal New_Max As Long)
    If New_Max < 0 Then
        MsgBox "El valor mínimo debe ser un entero positivo", _
            vbInformation, "Numeric TextBox"
    ElseIf New_Max < Min Then
        MsgBox "El valor máximo debe ser un mayor que el mínimo", _
            vbInformation, "Numeric TextBox"
    Else
        m_Max = New_Max
    End If
    PropertyChanged "Max"
End Property

Public Property Get Min() As Long
    Min = m_Min
End Property

Public Property Let Min(ByVal New_Min As Long)
    If New_Min < 0 Then
        MsgBox "El valor mínimo debe ser un entero positivo", _
            vbInformation, "Numeric TextBox"
    ElseIf New_Min > Max Then
        MsgBox "El valor mínimo debe ser un menor que el máximo", _
            vbInformation, "Numeric TextBox"
    End If
End Property
```



```

Else
    m_Min = New_Min
End If
PropertyChanged "Min"
End Property

'Inicializar propiedades para control de usuario
Private Sub UserControl_InitProperties()
    m_Max = m_def_Max
    m_Min = m_def_Min
End Sub

'Cargar valores de propiedad desde el almacén
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    txt.Text = PropBag.ReadProperty("Text", "")
    m_Max = PropBag.ReadProperty("Max", m_def_Max)
    m_Min = PropBag.ReadProperty("Min", m_def_Min)
End Sub

'Escribir valores de propiedad en el almacén
Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    Call PropBag.WriteProperty("Text", txt.Text, "")
    Call PropBag.WriteProperty("Max", m_Max, m_def_Max)
    Call PropBag.WriteProperty("Min", m_Min, m_def_Min)
End Sub
  
```

2. Una comunidad de propietarios en la que existen una serie de apartamentos en alquiler pretende gestionar los alquileres. Para ello dispone de una base de datos Access (alquiler.mdb) que dispone, entre otras de las siguientes tablas.

Tabla Propietarios		
Campo	Formato	Observaciones
IDPropietario	Autonumérico	Clave primaria de la tabla
Nombre	Texto de 50 posiciones	

Tabla Apartamentos		
Campo	Formato	Observaciones
IDApartamento	Autonumérico	Clave primaria de la tabla
IDPropietario	Entero largo	Puede tener repeticiones (un propietario puede tener varios apartamentos)
Metros	Entero	Metros cuadrados del apartamento

Tabla Alquileres (se genera un registro cuando se alquila un apartamento)		
Campo	Formato	Observaciones
IDAlquiler	Autonumérico	Clave primaria de la tabla
IDApartamento	Entero largo	Identificador del apartamento alquilado
Mensualidad	Entero largo	Importe mensual del apartamento
FechaInicio	Fecha/Hora	Fecha de inicio del alquiler
FechaFin	Fecha/Hora	Fecha de terminación del alquiler



Tabla Inquilinos		
Campo	Formato	Observaciones
IDInquilino	Autonumérico	Clave primaria de la tabla
IDAlquiler	Entero largo	Identificador del alquiler que ha realizado el inquilino
Nombre	Texto de 50 posiciones	
Moroso	Lógico	Indica si le quedan recibos por pagar

Tabla Pendientes (incluye un registro por cada recibo pendiente de pagar)		
Campo	Formato	Observaciones
IDPendiente	Autonumérico	Clave primaria de la tabla
IDAlquiler	Entero largo	Identificador del alquiler que ha generado el recibo
FechaRecibo	Fecha/Hora	Fecha de expedición del recibo
Importe	Entero largo	Importe del recibo

Tabla Pagos (Incluye un registro por cada recibo que haya satisfecho el inquilino)		
Campo	Formato	Observaciones
IDPago	Autonumérico	Clave primaria de la tabla
IDAlquiler	Entero largo	Identificador del alquiler que ha generado el recibo
FechaRecibo	Fecha/Hora	Fecha de expedición del recibo
FechaPago	Fecha/Hora	Fecha de pago del recibo
Importe	Entero largo	Importe del recibo

Se desea diseñar un formulario que gestione los pagos de los alquileres. El formulario tendrá el siguiente aspecto:

Notas:

- Sólo existen los controles que aparecen en el formulario
- Se puede utilizar cualquier modelo de acceso a datos (DAO o ADO)
- El DataGrid (o DBGrid) dgDetalle ya tiene definidos las propiedades DataField de las columnas a los campo Referencia y Unidades respectivamente.

### Funcionalidad del formulario

1. Al introducir el número de un apartamento en el control txtApartamento, buscará dicho apartamento en la tabla de apartamentos. Si el apartamento no existiera, aparecerá un mensaje advirtiéndolo. Si el apartamento existe, aparecerá el nombre de su propietario y los metros cuadrados del mismo.
2. También se buscará si dicho apartamento está alquilado actualmente en la tabla de alquileres. De no ser así, los datos del inquilino aparecerán en blanco. Si el apartamento está alquilado, aparecerá la mensualidad, la fecha de inicio del alquiler, el nombre del inquilino y el control chkMoroso se marcará según el valor del campo moroso de la tabla de inquilinos.



3. También debe aparecer en el DataGrid dgPendientes los pagos pendientes que tenga el inquilino. Si no tiene pagos pendientes, la casilla chkMoroso aparecerá sin marcar.
4. Al pulsar sobre el botón Actualizar, se actualizarán los campos de la tabla Inquilinos (sólo se podrá actualizar la casilla chkMoroso).
5. Al pulsar sobre el botón Pagar, se pagará el recibo que tengamos seleccionado en el control dgPendientes. Para ello se generará un nuevo registro en la tabla Pagos con los campos IDAlquiler, FechaRecibo e Importe del elemento seleccionado del control dgPendientes. La fecha del pago será la fecha actual. Al mismo tiempo se eliminará el registro de la tabla Pendientes.

**Puntuación: 3 puntos.**

```
Option Explicit
Private cn As ADODB.Connection
Private rsAlq As ADODB.Recordset
Private rsInq As ADODB.Recordset
Private rsPen As ADODB.Recordset

Private Sub cmdActualizar_Click()
    If chkMoroso = 0 Then
        rsInq("Moroso") = False
    Else
        rsInq("Moroso") = True
    End If
    rsInq.Update
End Sub

Private Sub cmdPagar_Click()
    Dim SQL As String
    SQL = "INSERT INTO Pagos (IDAlquiler, FechaRecibo, FechaPago, Importe) " & _
        "VALUES (" & rsPen("IDAlquiler") & ",#" & rsPen("FechaRecibo") & _
        "#,#" & _
        Date & "#," & rsPen("Importe") & ")"
    cn.Execute SQL
    rsPen.Delete
    If rsPen.RecordCount = 0 Then
        chkMoroso = 0
    Else
        chkMoroso = 1
    End If
End Sub

Private Sub Form_Load()
    'Abrir la conexión
    Set cn = New ADODB.Connection
    cn.Provider = "Microsoft.Jet.OLEDB.3.51"
    cn.ConnectionString = "\\luis\bd\alquiler.mdb"
    cn.CursorLocation = adUseClient
    cn.Open
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set cn = Nothing
    Set rsAlq = Nothing
    Set rsInq = Nothing
    Set rsPen = Nothing
End Sub

Private Sub txtApto_LostFocus()
    Set rsAlq = New ADODB.Recordset
    rsAlq.Source = "SELECT Apartamentos.IDApartamento, Alquileres.IDAlquiler, _
        "Nombre, Metros, Mensualidad, FechaInicio, FechaFin " & _
        "FROM Apartamentos, Propietarios, Alquileres " & _
```



```
        " WHERE Apartamentos.IDPropietario = " & _  
        "Propietarios.IDPropietario " & _  
        "AND Apartamentos.IDApartamento = Alquileres.IDApartamento " & _  
        " AND FechaFin > #" & Format(Date, "mm/dd/yy") & "#" & _  
        " AND Alquileres.IDApartamento = " & txtApto  
rsAlq.ActiveConnection = cn  
rsAlq.CursorType = adOpenStatic  
rsAlq.Open  
If rsAlq.RecordCount = 0 Then  
    Frame1.Enabled = False  
    Frame2.Enabled = False  
    txtPropietario = Empty  
    txtMetros = Empty  
    txtMensualidad = Empty  
Else  
    Frame1.Enabled = True  
    Frame2.Enabled = True  
    txtPropietario = rsAlq("Nombre")  
    txtMetros = rsAlq("Metros")  
    txtMensualidad = rsAlq("Mensualidad")  
    txtFecha = rsAlq("FechaInicio")  
    'Buscar el inquilino  
    Set rsInq = New ADODB.Recordset  
    rsInq.Source = "SELECT * FROM Inquilinos WHERE IDAlquiler = " & _  
        rsAlq("IDAlquiler")  
    rsInq.ActiveConnection = cn  
    rsInq.CursorType = adOpenDynamic  
    rsInq.LockType = adLockOptimistic  
    rsInq.Open  
    txtInquilino = rsInq("Nombre")  
    If rsInq("Moroso") Then  
        chkMoroso = 1  
    Else  
        chkMoroso = 0  
    End If  
    'Comprobar los pagos  
    Set rsPen = New ADODB.Recordset  
    rsPen.Source = "SELECT * FROM Pendientes WHERE IDAlquiler = " & _  
        rsInq("IDAlquiler") & " ORDER BY FechaRecibo"  
    rsPen.ActiveConnection = cn  
    rsPen.CursorType = adOpenDynamic  
    rsPen.LockType = adLockOptimistic  
    rsPen.Open  
    Set dgPagos.DataSource = rsPen  
    'Comprobar si es un moroso  
    If rsPen.RecordCount = 0 Then  
        chkMoroso = 0  
    Else  
        chkMoroso = 1  
    End If  
End If  
End Sub
```

3. Sobre la misma base de datos, se desea gestionar los nuevos alquileres. Para ello se dispone de un formulario con el siguiente formato:



fraAlquiler

cmdAlquiler

#### Notas:

- El formulario dispone además de tres controles de datos ocultos (`datAlquileres`, `datApartamentos` y `datInquilinos`) para gestionar las tablas de Alquileres, Apartamentos e Inquilinos
- La gestión de las bases de datos se debe hacer utilizando esos controles de datos
- No existen más controles que los que aparecen en el formulario.
- No se ha establecido ninguna propiedad a los controles además de la propiedad `Name` de cada uno de ellos y las propiedades `Caption` de los botones de órdenes y las etiquetas.

#### Funcionalidades de la aplicación

1. Cuando el control `txtApartamento` pierda el foco, se buscará el apartamento que aparezca en el control. El nombre del propietario y sus metros cuadrados aparecerán en los controles `txtPropietario` y `txtMetros` respectivamente.
2. Se comprobará que dicho apartamento esté actualmente libre en la tabla de alquileres. Si el apartamento está actualmente ocupado, se deshabilitará el marco `fraAlquiler`.
3. Una vez rellenos los datos del alquiler (nombre del inquilino, mensualidad y la fecha del fin del alquiler), al pulsar el botón `Alquilar`, se procederá a alquilar el apartamento.
4. Para alquilar el apartamento se dará un alta en la tabla de Alquileres, relleno del campo `IDApartamento` con el valor del control `txtApartamento`, la mensualidad con el valor del control `txtMensualidad`, la fecha de inicio del alquiler con la fecha actual y la fecha de fin del alquiler con el valor del control `txtFechaFin`. Al mismo tiempo se dará un alta al inquilino con el nombre del inquilino y el campo `IDAlquiler` del registro de alquileres que se acaba de dar de alta.
5. Al mismo tiempo, se generará un primer recibo que se almacenará en la tabla de Pendientes con el importe de la primera mensualidad.

#### Puntuación: 2 puntos.

```
Begin VB.Data datInquilinos
    Caption           = "datInquilinos"
    Connect           = "Access"
    DatabaseName      = "D:\Bases de Datos\Alquiler.mdb"
    ...
End
Begin VB.Data datAlquileres
    DatabaseName      = "D:\Bases de Datos\Alquiler.mdb"
    ...
End
Begin VB.Data datApartamentos
    DatabaseName      = "D:\Bases de Datos\Alquiler.mdb"
    ...
End
Option Explicit

Private Sub cmdAlquiler_Click()
```



```
Dim IDAlquiler As Long
Dim sql As String

'Da de alta al alquiler
datAlquileres.Recordset.AddNew
IDAlquiler = datAlquileres.Recordset("IDAlquiler")
datAlquileres.Recordset("Mensualidad") = txtMensualidad
datAlquileres.Recordset("FechaInicio") = Date
datAlquileres.Recordset("FechaFin") = txtFechaFin
datAlquileres.Recordset("IDApartamento") = txtApartamento
datAlquileres.Recordset.Update

'Da de alta el inquilino
datInquilinos.Recordset.AddNew
datInquilinos.Recordset("IDAlquiler") = IDAlquiler
datInquilinos.Recordset("Nombre") = txtInquilino
datInquilinos.Recordset("Moroso") = False
datInquilinos.Recordset.Update

'Da de alta el primer pago
sql = "INSERT INTO Pendientes " & _
      "(IDAlquiler, FechaRecibo, Importe) " & _
      "VALUES (" & IDAlquiler & ",#" & Date & "#," & _
      txtMensualidad & ")"
datInquilinos.Database.Execute sql

End Sub

Private Sub Form_Load()
Dim qry As String
qry = "SELECT Apartamentos.IDApartamento, Nombre, Metros " & _
      "FROM Apartamentos, Propietarios " & _
      "WHERE Apartamentos.IDPropietario = Propietarios.IDPropietario"
datApartamentos.RecordSource = qry
datApartamentos.Refresh

datInquilinos.RecordSource = "Inquilinos"
datInquilinos.Refresh
End Sub

Private Sub txtApartamento_LostFocus()
Dim qry As String
'Busca el apartamento
datApartamentos.Recordset.FindFirst "IDApartamento = " & txtApartamento
txtMetros = datApartamentos.Recordset("Metros")
txtPropietario = datApartamentos.Recordset("Nombre")
'Busca el apartamento en alquileres
qry = "SELECT * FROM Alquileres " & _
      " WHERE IDApartamento = " & txtApartamento & _
      " ORDER BY FechaFin "
datAlquileres.RecordSource = qry
datAlquileres.Refresh
'datAlquileres contiene alquileres de ese apartamento
'Si no está vacío puede que esté alquilado
If datAlquileres.Recordset.RecordCount > 0 Then
datAlquileres.Recordset.MoveLast
'Si la última fecha de fin del alquiler es
'mayor que la actual, está alquilado
If datAlquileres.Recordset("FechaFin") > Date Then
MsgBox "El apartamento está alquilado"
txtInquilino = Empty
txtMensualidad = Empty
txtFechaFin = Empty
fraAlquiler.Enabled = False
```



```
    Else
        fraAlquiler.Enabled = True
    End If
Else
    fraAlquiler.Enabled = True
End If
End Sub
```

Zeon PDF Driver Trial  
www.zeon.com.tw