UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID
Facultad de Informática
Escuela Universitaria de Informática
Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

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	**Mañana**	CENTRO	**Facultad**
CARÁCTER	**Anual**	CURSO ACADÉMICO	**1999/2000**
DURACIÓN APROXIMADA	**2 horas**		

Soluciones al examen

Pruebas teórico-prácticas

1. Explique los distintos tipos de objetos recordset del modelo DAO. Explique sus características más importantes, así como los distintos tipos de búsqueda que se pueden utilizar en cada uno de ellos.

Explicar los 5 tipos de recordset de DAO (Table, Dynaset, Snapshot, ForwardOnly y Dynamic) e indicar en cada uno de ellos como utilizar los métodos de búsquda Find... y Seek.

2. Componentes ActiveX. Explique los distintos tipos de componentes ActiveX que se pueden realizar desde Visual Basic 6.

Explicar los componentes de código ActiveX, Documentos ActiveX y controles ActiveX.

3. Objetos Command y Parameter en el modelo de objeto ADO. ¿Para qué se utilizan? Describa sus principales propiedades.

Explicar el uso de estos objetos y sus propiedades más importantes (ActiveConnection, CommandText, CommandType, etc.)

Dada la siguiente consulta SQL "SELECT * FROM Clientes WHERE Provincia = sProvincia", donde sProvincia es un argumento de tipo cadena que se le pasa a la sentencia, codifique las líneas necesarias para realizar la consulta utilizando objetos Command y Parameter de ADO.

```
Set cmd = New ADODB.Command
cmd.ActiveConnection = conexiónActiva
cmd.CommandType = adCmdText
cmd.CommandText = "SELECT * FROM Clientes WHERE Provincia = sProvincia"

Set prm = New ADODB.Parameter
prm = cm.CreateParameter(sProvincia,adChar,adParamInput,50,txtProvincia)
cm.Parameter.Append prm

Set rs = cmd.Execute
```

4. El control Web Browser. ¿Qué es y para qué se utiliza? Describa los eventos más importantes que se producen en el control.

Es un envoltorio de la aplicación InternetExplorer que se puede utilizar en un formulario. Explicar los enventos BeginNavigate, NavigateComplete, BeginDownload, DownloadComplete).

Puntuación: 0,75 puntos cada pregunta

Pruebas prácticas

1. Se desea diseñar un control ActiveX compuesto únicamente de un control ListBox (Arithmetic ListBox). El control deberá contemplar los métodos necesarios para gestionar los elementos del ListBox (AddItem, RemoveItem, Clear), la propiedad Text y el evento Click.

El control deberá realizar operaciones aritméticas con los elementos del ListBox mediante el método Calcule. Al llamar a este método se operará con todos los elementos de la lista según el valor de la propiedad OperationType que también deberá implementarse. Esta propiedad contendrá valores enteros 0 (suma), 1 (resta), 2 (multiplicación) o 3 (división). El método Calcule devolverá el resultado de las operaciones.

Si alguno de los elementos del ListBox no es numérico, se advertirá de la incidencia y el método Calcule devolverá 0.

Zeon PDF Driver Trial
www.zeon.com.tw

Puntuación: 2 puntos

```
Option Explicit
'Valores por omisión de las propiedades
Const m_def_OperationType = 0
'Variables de propiedades
Dim m_OperationType As Integer
'Enumeraciones
Enum OperationTypeConstants
    Add
    Subtract
    Multiply
    Divide
End Enum
'Declaraciones de eventos
Event Click()

Private Sub UserControl_Resize()
    lst.Move 0, 0, ScaleWidth, ScaleHeight
End Sub
Public Sub AddItem(ByVal Item As String, Optional ByVal Index As Variant)
    lst.AddItem Item, Index
End Sub

Public Sub Clear()
    lst.Clear
End Sub

Private Sub lst_Click()
    RaiseEvent Click
End Sub

Public Sub RemoveItem(ByVal Index As Integer)
    lst.RemoveItem Index
End Sub

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

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

Public Property Get OperationType() As OperationTypeConstants
    OperationType = m_OperationType
End Property

Public Property Let OperationType(ByVal New_OperationType As
OperationTypeConstants)
    m_OperationType = New_OperationType
    PropertyChanged "OperationType"
End Property

Public Function Calcule() As Variant
    Dim resultado As Variant
    Dim i As Integer
    resultado = 0
    For i = 0 To lst.ListCount - 1
        If Not IsNumeric(lst.List(i)) Then
            MsgBox "Los elementos deben ser numéricos", vbInformation, _
                   "Arithmetic ListBox"
            resultado = 0
            Exit For
```

Zeon PDF Driver Trial
www.zeon.com.tw

```
        Else
            Select Case m_OperationType
                Case 0
                    resultado = resultado + lst.List(i)
                Case 1
                    If i = 0 Then
                        resultado = lst.List(i)
                    Else
                        resultado = resultado - lst.List(i)
                    End If
                Case 2
                    If i = 0 Then
                        resultado = lst.List(i)
                    Else
                        resultado = resultado * lst.List(i)
                    End If
                Case 3
                    If i = 0 Then
                        resultado = lst.List(i)
                    Else
                        resultado = resultado / lst.List(i)
                    End If
            End Select
        End If
    Next
    Calcule = resultado
End Function

'Inicializar propiedades para control de usuario
Private Sub UserControl_InitProperties()
    m_OperationType = m_def_OperationType
End Sub

'Cargar valores de propiedad desde el almacén
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    lst.Text = PropBag.ReadProperty("Text", "")
    m_OperationType = PropBag.ReadProperty("OperationType",
m_def_OperationType)
End Sub

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

Zeon PDF Driver Trial
www.zeon.com.tw

2. Un taller de reparaciones de automóviles mantiene informatizadas sus operaciones. La información está almacenada en una base de datos Access (Taller.mdb) que, entre otras, contiene las siguientes tablas:

Tabla Clientes (contiene información sobre los clientes)		
Campo	**Formato**	**Observaciones**
DNI	Texto de 11 posiciones	Clave primaria de la tabla
Nombre	Texto de 50 posiciones	

Tabla coches (contiene información sobre los vehículos que han entrado en el taller)		
Campo	**Formato**	**Observaciones**
Matricula	Texto de 10 posiciones	Clave primaria de la tabla
DNI	Texto de 11 posiciones	DNI del propietario del vehículo. Un propietario puede tener varios coches
Foto	Objeto OLE	Foto del vehículo

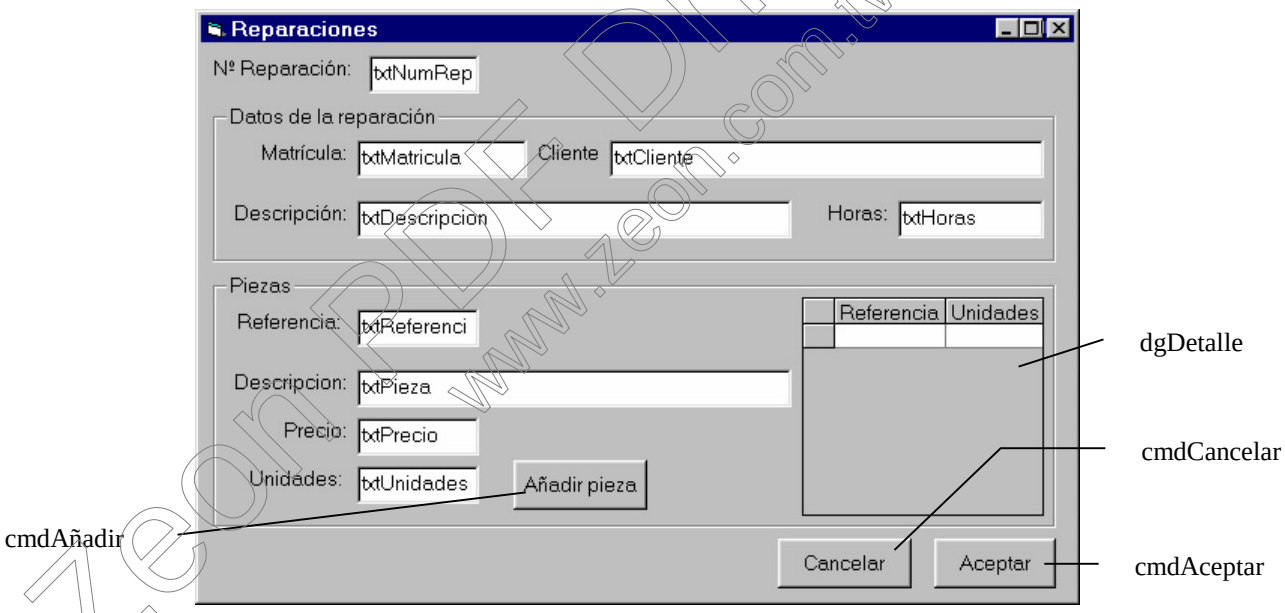
Tabla Reparaciones (cada vez que un vehículo entra en taller, se genera un registro de esta tabla)

Campo	Formato	Observaciones
NumReparacion	Entero largo	Clave primaria de la tabla
Matricula	Texto de 10 posiciones	Matrícula del vehículo. Puede aparecer varias veces en la tabla
Descripción	Texto de 50 posiciones	Descripción de la avería
FechaEntrada	Fecha/Hora	Fecha de entrada en el taller
FechaSalida	Fecha/Hora	Fecha salida del taller
Horas	Entero	Horas de mano de obra que se facturarán

Tabla DetallesReparacion (un registro por cada pieza utilizada en la reparación)		
Campo	**Formato**	**Observaciones**
NumReparacion	Entero largo	Número de la reparación en la que se ha empleado la pieza. Puede tener duplicados
Referencia	Texto de 10 posiciones	Referencia de la pieza empleada
Unidades	Entero	

Tabla Piezas (Almacena las piezas del almacén)		
Campo	**Formato**	**Observaciones**
Referencia	Texto de 10 posiciones	Clave primaria de la tabla. Referencia de la pieza empleada
Descripción	Texto de 50 posiciones	Descripción de la pieza
Precio	Entero largo	

Se desea realizar un formulario en el que se vayan añadiendo las piezas utilizadas en una reparación, así como actualizando la mano de obra. El formulario presenta 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. Cuando el control txtNumRep pierda el foco, deberá buscar en la tabla Reparaciones la matrícula del vehículo, la descripción de la avería y las horas empleadas. Además deberá sacar el nombre del propietario del coche. También, si ya se hubieran utilizado piezas en el DataGrid aparecerán la referencia y las unidades de las piezas empleadas.

Zeon PDF Driver Trial www.zeon.com.tw

2. Cuando el control txtReferencia pierda el foco, se deberá buscar en la tabla Piezas la descripción y el precio de la misma. Si la referencia no existiera aparecerá un mensaje advirtiéndolo.

3. Al pulsar sobre el botón Añadir se añadirá un registro en la tabla DetallesReparacion (el nuevo registro deberá verse en el DataGrid).

4. Al pulsar sobre el botón Aceptar, se actualizará la tabla Reparación, guardando el valor de las horas empleadas almacenado en el control txtHoras.

5. Al pulsar sobre el botón Cancelar se eliminarán todos los detalles de la reparación.

Puntuación: 3 puntos.

```
Option Explicit
Private cn As ADODB.Connection
Private rsRep As ADODB.Recordset
Private rsPieza As ADODB.Recordset
Private rsDetalle As ADODB.Recordset

Private Sub cmdAceptar_Click()
    rsRep("Horas") = txtHoras
    rsRep.Update
End Sub

Private Sub cmdAñadir_Click()
    rsDetalle.AddNew
    rsDetalle("NumReparacion") = txtNumRep
    rsDetalle("Referencia") = txtReferencia
    rsDetalle("Unidades") = txtUnidades
    rsDetalle.Update
    rsDetalle.Requery
    Set dgDetalles.DataSource = rsDetalle
End Sub

Private Sub cmdCancelar_Click()
    cn.Execute "DELETE * FROM DetallesReparacion WHERE NumReparacion = " & _
               txtNumRep
    rsDetalle.Requery
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\taller.mdb"
    cn.CursorLocation = adUseClient
    cn.Open
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set cn = Nothing
    Set rsRep = Nothing
    Set rsPieza = Nothing
    Set rsDetalle = Nothing
End Sub

Private Sub txtNumRep_LostFocus()
    'Buscar la reparación
    Set rsRep = New ADODB.Recordset
    rsRep.Source = "SELECT *  FROM (Clientes INNER JOIN Coches " & _
                   "ON Clientes.DNI = Coches.DNI) INNER JOIN Reparaciones " & _
                   "ON Coches.Matricula = Reparaciones.Matricula" & _
                   " WHERE Reparaciones.NumReparacion =" & txtNumRep
    rsRep.CursorType = adOpenDynamic
    rsRep.LockType = adLockOptimistic
    rsRep.ActiveConnection = cn
```

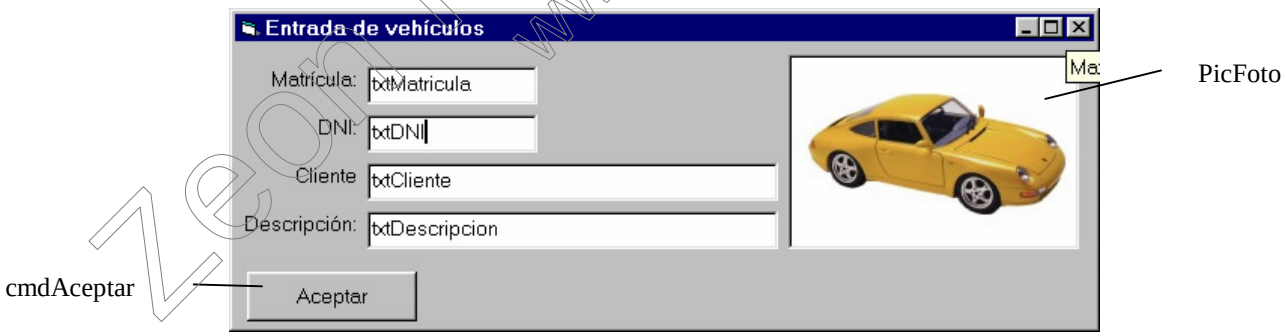
Zeon PDF Driver Trial
www.zeon.com.tw

```
    rsRep.Open
    'Mover los campos al formulario
    If rsRep.RecordCount <> 0 Then
        txtMatricula = rsRep("Reparaciones.Matricula")
        txtCliente = rsRep("Nombre")
        txtDescripcion = rsRep("Descripcion")
    Else
        MsgBox "No existe"
    End If

    'Buscar los detalles de la reparación
    Set rsDetalle = New ADODB.Recordset
    rsDetalle.Source = "SELECT * from DetallesReparacion"
    rsDetalle.CursorType = adOpenDynamic
    rsDetalle.LockType = adLockOptimistic
    rsDetalle.ActiveConnection = cn
    rsDetalle.Open
    Set dgDetalles.DataSource = rsDetalle
End Sub

Private Sub txtReferencia_LostFocus()
    Set rsPieza = New ADODB.Recordset
    rsPieza.Source = "SELECT *  FROM Piezas WHERE Referencia = '" & _
                     txtReferencia & "'"
    rsPieza.CursorType = adOpenDynamic
    rsPieza.LockType = adLockOptimistic
    rsPieza.ActiveConnection = cn
    rsPieza.Open
    If rsPieza.RecordCount = 0 Then
        MsgBox "No existe la pieza"
    Else
        txtDescripcionPieza = rsPieza("Descripcion")
        txtPrecio = rsPieza("Precio")
    End If
End Sub
```

3. El taller del apartado anterior desea gestionar la entrada de vehículos a reparación. Para ello dispone de un formulario con el siguiente formato:



Notas:

- El formulario dispone además de tres controles data ocultos (datCoches, datClientes y DatReparaciones) para gestionar las tablas de coches, clientes y reparaciones.
- La gestión de las bases de datos se debe hacer utilizando esos controles Data
- El formulario dispone además de un cuadro de diálogo común llamado dlg.
- 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.

Zeon PDF Driver Trial
www.zeon.com.tw

Funcionalidades de la aplicación

1. Escribir el valor de las propiedades que considere necesario.
2. Al hacer doble Click sobre el control PicFoto, se abrirá un cuadro de diálogo que permitirá seleccionar un archivo gráfico y cargarlo en el control.
3. Al perder el foco del control txtMatrícula, se buscará el coche en la tabla Coches. Si el coche ya exisitiera, aparecerá el DNI y el nombre del Cliente. En caso contrario un mensaje advertirá de la incidencia y el usuario deberá rellenar dichos campos.
4. Al pulsar sobre el botón Aceptar, se dará un alta a la reparación con la matrícula, la descripción de la avería y la fecha de entrada, que será la fecha actual. Al mismo tiempo, si el coche no exisitía se dará un alta del vehículo en la tabla de coches, así como a su propietario en la tabla de clientes.

Puntuación: 2 puntos.

```
Begin VB.Data datReparaciones
      Caption         =   "datReparaciones"
      Connect         =   "Access"
      DatabaseName    =   "D:\Bases de Datos\taller.mdb"
      RecordSource    =   "Reparaciones"
Begin VB.Data datClientes
      Caption         =   "datClientes"
      Connect         =   "Access"
      DatabaseName    =   "D:\Bases de Datos\taller.mdb"
      RecordSource    =   "Clientes"
      Top             =   2280
Begin VB.Data datCoches
      Caption         =   "datCoches"
      Connect         =   "Access"
      DatabaseName    =   "D:\Bases de Datos\taller.mdb"
      RecordSource    =   "Coches"
Begin VB.PictureBox picFoto
      DataField       =   "Foto"
      DataSource      =   "datCoches"

Option Explicit

Private Sub cmdAceptar_Click()
    'Si el coche no existe, se da de alta el coche y el cliente
    If datCoches.Recordset.NoMatch Then
        'Se da de alta el cliente
        datClientes.Recordset.AddNew
        datClientes.Recordset("DNI") = txtDNI
        datClientes.Recordset("Nombre") = txtCliente
        datClientes.Recordset.Update
        'Se da de alta el coche
        datCoches.Recordset.AddNew
        datCoches.Recordset("Matricula") = txtMatricula
        datCoches.Recordset("DNI") = txtDNI
        datCoches.Recordset("Foto") = Nothing
        datCoches.Recordset.Update
    End If

    'Se da de alta la reparación
    datReparaciones.Recordset.AddNew
    datReparaciones.Recordset("Matricula") = txtMatricula
    datReparaciones.Recordset("Descripcion") = txtDescripcion
    datReparaciones.Recordset("FechaEntrada") = Date
    datReparaciones.Recordset.Update

End Sub

Private Sub picFoto_Click()
```

Zeon PDF Driver Trial
www.zeon.com.tw

```
    dlg.ShowOpen
    picFoto.Picture = LoadPicture(dlg.FileName)
End Sub

Private Sub txtMatricula_LostFocus()
    datCoches.Recordset.FindFirst "Matricula = '" & txtMatricula & "'"
    If datCoches.Recordset.NoMatch Then
        'El coche no existe
        MsgBox "El coche no existe, debe introducir los datos del propietario y
la foto"
    Else
        txtDNI = datCoches.Recordset("DNI")
        'Buscar al propietario
        datClientes.Recordset.FindFirst "DNI ='" & txtDNI & "'"
        txtCliente = datClientes.Recordset("Nombre")
    End If
End Sub
```

Zeon PDF Driver Trial
www.zeon.com.tw