



Cuadernillo de examen

ASIGNATURA		CÓDIGO	305
CONVOCATORIA	Ordinaria de Junio de 2002 (segundo parcial)	PLAN DE ESTUDIOS	1996
ESPECIALIDAD	Común	CURSO	3º
TURNO	Mañana	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	2001/2002
DURACIÓN APROXIMADA	2 horas 15 minutos		

Soluciones propuestas al examen

Preguntas teórico-prácticas

(Nota: las referencias al “libro de texto” son del primer libro de la bibliografía básica: **BALENA, F.:** *Programación avanzada con Microsoft Visual Basic*. Madrid: McGraw-Hill, 2000

1. Explique las características y componentes de la arquitectura OLEDB.

Páginas 353-354 del libro de texto y apuntes de clase

Puntuación 1 punto

2. El objeto Command. Explique las características, propiedades y métodos más importantes. Señale además las situaciones en las que es conveniente utilizar este tipo de objetos.

Puntuación 1 punto

Páginas 610-614 y 649-651 del libro de texto y apuntes de clase

3. Conceptos de documento bien formado y documento validado en XML. Señale además las características que debe tener un documento bien formado.

Puntuación 1 punto

Apuntes de clase

Parte práctica

Ejercicio 1

Una agencia de viajes de dedica a vender paquetes de vacaciones completos. La información sobre destinos, clientes y reservas está almacenada en la tabla con las siguientes tablas.

Tabla: Paquetes		
Guarda información sobre los paquetes de vacaciones que ofrece la agencia		
Campo	Formato	Observaciones
idPaquete	Texto	Clave primaria de la tabla
Descripción	Texto	
FechaSalida	Fecha/Hora	
Días	Numérico	Número de días del viaje
Precio	Numérico	
PlazasTotales	Numérico	
PlazasLibres	Numérico	



Tabla: Reservas		
Guarda información de las reservas efectuadas		
Campo	Formato	Observaciones
idReserva	Autonumérico	Clave primaria de la tabla
idPaquete	Texto	
idCliente	Texto	Clave del cliente que ha realizado la reserva
Importe	Numérico	
Fecha	Fecha/Hora	Fecha en la que se ha realizado la reserva
ACuenta	Numérico	Cantidad entregada a cuenta
Pagado	Lógico	Verdadero si el importe se ha pagado en su totalidad
Tabla: Clientes		
Guarda información de las clientes/viajeros que han utilizado en algún momento la agencia		
Campo	Formato	Observaciones
idCliente	Texto	Clave primaria de la tabla
Nombre	Texto	
Tabla: ClientesReservas		
Se utiliza para la relación n a n de los clientes y las reservas		
Campo	Formato	Observaciones
idReserva	Numérico	Junto con el campo idCliente forman la clave primaria de la tabla
idCliente	Texto	Junto con el campo idReserva forman la clave primaria de la tabla

Para la realización de nuevas reservas se utiliza el siguiente formulario:

Notas: Sólo están disponibles los controles que se visualizan. Las únicas propiedades establecidas en tiempo de diseño son las referentes al nombre, la posición y el tamaño de los controles, menos para el DataGrid dgPaquetes en los que se ha establecido en tiempo de diseño el tamaño y los títulos de las columnas y los campos asociados a ellas.

Declaraciones iniciales de formulario

```
Option Explicit
Public cn As New ADODB.Connection
Private rsPaq As New ADODB.Recordset
Private rsCli As New ADODB.Recordset
Private rsRes As New ADODB.Recordset
```



```
Dim cm As New ADODB.Command
```

1. Al cargar el formulario, se cargarán en el DataGrid todos los viajes disponibles (aquellos en los que queden plazas y cuya fecha de salida sea posterior a la fecha actual). Al pulsar sobre cualquier viaje, aparecerá en el control txtFecha la fecha del día y se pasará el foco al control txtIDCliente.

Puntuación: 1 punto.

```
Private Sub Form_Load()  
    'Abrir la conexión  
    Set cn = New ADODB.Connection  
    cn.CursorLocation = adUseClient  
    cn.Provider = "Microsoft.JET.OLEDB.4.0"  
    cn.Open "\\luis\bd\viajes.mdb"  
  
    'Crear objeto Command  
    Set cm = New ADODB.Command  
    cm.ActiveConnection = cn  
    cm.CommandType = adCmdText  
  
    'Crear el recordset de reservas  
    Set rsRes = New ADODB.Recordset  
    rsRes.Open "Reservas", cn, adOpenDynamic, adLockOptimistic  
  
    'Buscar Viajes disponibles actualmente  
    Dim sql As String  
    Set rsPaq = New ADODB.Recordset  
    sql = "SELECT * FROM Paquetes WHERE " & _  
        "PlazasLibres > 0 AND " & _  
        "FechaSalida > Now()" & " ORDER BY FechaSalida"  
    rsPaq.Open sql, cn, adOpenDynamic, adLockOptimistic  
  
    'Asignar el recordset al DataGrid  
    Set dgPaquetes.DataSource = rsPaq  
  
End Sub  
  
Private Sub dgPaquetes_Click()  
    'Al pulsar sobre un viaje se carga la fecha actual  
    'y el foco va al identificador del cliente que ha hecho  
    'la reserva  
    txtFecha = Format(Now(), "dd/mm/yyyy")  
    txtIDCliente.SetFocus  
End Sub
```

2. Al salir del control txtIDCliente, se buscará en la tabla Clientes si existe un cliente con dicho código. Si se encuentra aparecerá en la etiqueta lblNombreCliente el nombre del cliente. En caso contrario aparecerá otro formulario para introducir el nombre y se dará de alta un nuevo cliente (este identificador de cliente es el del cliente que realiza la reserva).

The image shows a standard Windows dialog box with a blue title bar that says "Nuevo cliente". Inside the dialog, there are two text boxes. The first one is labeled "Identificador cliente" and contains the text "lblIDCliente". The second one is labeled "Nombre cliente" and contains the text "txtNombreCliente". At the bottom of the dialog, there is a button labeled "Aceptar".

Puntuación: 1,5 puntos.



```
Private Sub txtIDCliente_LostFocus()  
    Dim rsAux As ADODB.Recordset  
    'Busca el identificador del cliente  
    cm.CommandText = "SELECT * FROM Clientes " & _  
        "WHERE idCliente = '" & txtIDCliente & "'"<br>  
    Set rsAux = cm.Execute  
    If rsAux.RecordCount > 0 Then  
        'Si lo encuentra carga el nombre en la etiqueta  
        lblNombreCliente = rsAux!Nombre  
    Else  
        'Si no saca un formulario para darlo de alta  
        frmNuevoCliente.lblIDCliente = txtIDCliente  
        frmNuevoCliente.Show 1  
        txtIDCliente.SetFocus  
    End If  
End Sub
```

Código del formulario para un nuevo cliente

```
Private Sub cmdAceptar_Click()  
    Dim sql As String  
    sql = "INSERT INTO Clientes (idCliente,Nombre) " & _  
        "VALUES ('" & lblIDCliente & _  
        "', '" & txtNombreCliente & "'"<br>  
    frmReservas.cn.Execute sql  
    Unload Me  
End Sub
```

3. A continuación se cargarán los viajeros que se incluyen en la reserva (la persona que hace la reserva no tiene porqué ir al viaje). Al pulsar sobre el botón “cmdAñadirViajero”, se buscará el cliente con el identificador que aparezca en el control txtIDViajero. Si el cliente existe, se añadirá su nombre en lstViajeros y su identificador en lstIDViajero. En caso contrario se procederá como en el apartado 2.

Puntuación: 1 punto.

```
Private Sub cmdAñadirViajero_Click()  
    Dim rsAux As ADODB.Recordset  
    'Busca el identificador del cliente  
    cm.CommandText = "SELECT * FROM Clientes " & _  
        "WHERE idCliente = '" & txtIDViajero & "'"<br>  
    Set rsAux = cm.Execute  
    If rsAux.RecordCount > 0 Then  
        'Si lo encuentra carga el nombre en la etiqueta  
        lstViajeros.AddItem rsAux!Nombre  
        lstIDViajero.AddItem rsAux!idCliente  
    Else  
        'Si no saca un formulario para darlo de alta  
        frmNuevoCliente.lblIDCliente = txtIDViajero  
        frmNuevoCliente.Show 1  
        txtIDViajero.SetFocus  
    End If  
End Sub
```

4. Al pulsar el botón “cmdRealizarReserva”, de dará de alta una nueva reserva y se saldrá del formulario. Para dar alta una reserva, se tomarán los campos idPaquete, idCliente y Fecha del formulario; el importe se calculará a partir del precio de la reserva y de el número de viajeros. Es necesario también insertar nuevos registros en la tabla ClientesReservas para cada uno de los viajeros. Por último se modificará el campo PlazasLibres de la tabla Paquetes, restando al valor actual el número de plazas reservadas.

Puntuación: 1,5 puntos

```
Private Sub cmdRealizarReserva_Click()  
    'Añadir una nueva reserva
```



```
rsRes.AddNew
rsRes!idCliente = txtIDCliente
rsRes!idPaquete = rsPaq!idPaquete
rsRes!Fecha = txtFecha
rsRes!Importe = lstViajeros.ListCount * rsPaq!Precio
rsRes.Update

'Añadir los viajeros
Dim i As Integer
Dim sql As String
sql = "INSERT INTO ClientesReservas (idCliente,idReserva) "
For i = 0 To lstViajeros.ListCount - 1
    cn.Execute sql & "VALUES ('" & lstIDViajero.List(i) & _
        "' , " & rsRes!idReserva & ")"
Next

'Disminuir el número de plazas
rsPaq!PlazasLibres = rsPaq!PlazasLibres - lstViajeros.ListCount
rsPaq.Update

'Cerrar el formulario
Unload Me
End Sub
```

Ejercicio 2

1. Escribir una DTD y un archivo XML correspondiente a dicha definición para almacenar las personas que trabajan en el departamento de desarrollo web de una empresa y los proyectos que se han llevado a cabo. Por cada persona es necesario almacenar la información de:

- Nombre
- Área de trabajo. Ésta podrá ser “Programador”, “Diseño”, “Contenidos”.
- Cada persona deberá tener también un identificador único.

Por cada proyecto debe aparecer:

- Cada proyecto deberá contener un atributo que indique su estado. Éste podrá ser “Finalizado”, “Suspendido”. Los proyectos en ejecución no contendrán dicho atributo.
- Descripción del proyecto.
- Fecha de comienzo
- Duración estimada
- Personas que intervienen o han intervenido en el proyecto.
- Los proyectos finalizados tendrán información sobre el coste total del mismo. Los proyectos suspendidos o en ejecución tendrán información sobre su presupuesto.

Una persona puede estar trabajando en varios proyectos. En un proyecto pueden participar varias personas.

Puntuación: 1 punto.

Archivo empresa.dtd

```
<!ELEMENT empresa (personas,proyectos)>
<!ELEMENT personas (persona+)>
<!ELEMENT persona (#PCDATA)>
<!ATTLIST persona
    idPers ID #REQUIRED
    area (Programador|Contenidos|Diseño) #REQUIRED>
<!ELEMENT proyectos (proyecto+)>
<!ELEMENT proyecto
    (descripcion,fechaInicio,duracion,(coste|presupuesto),participantes)>
<!ATTLIST proyecto
    idProy ID #REQUIRED
    estado (CDATA) #IMPLIED>
<!ELEMENT descripcion (#PCDATA)>
<!ELEMENT fechaInicio (#PCDATA)>
<!ELEMENT duracion (#PCDATA)>
```



```
<!ELEMENT coste (#PCDATA)>
<!ELEMENT presupuesto (#PCDATA)>
<!ELEMENT participantes (persona+)>
<!ELEMENT participante EMPTY>
<!ATTLIST participante
    id IDREF #REQUIRED>
```

Archivo empresa.xml

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE empresa SYSTEM "empresa.dtd">
<?xml-stylesheet type="text/xsl" href="proyectos.xsl"?>
<empresa>
  <personas>
    <persona idPers="E001" area="Programador">
      Bautista Armendi, Lázaro
    </persona>
    <persona idPers="E002" area="Programador">
      Carpio Belmonte, Fernanda del
    </persona>
    <persona idPers="E003" area="Diseño">
      Babilonia Gui, Mauricio
    </persona>
    <persona idPers="E004" area="Dontenidos">
      Bienvenido O, Antonia
    </persona>
    <persona idPers="E005" area="Programador">
      Estibano González, Juan José
    </persona>
    <persona idPers="E006" area="Contenidos">
      Bazalote de la Hoz, Fermín
    </persona>
    <persona idPers="E007" area="Contenidos">
      Benjalí Oz, Ana
    </persona>
    <persona idPers="E008" area="Diseño">
      Fuenlabrada Móstoles, Juana
    </persona>
  </personas>
  <proyectos>
    <proyecto idProy="P001" estado="finalizado">
      <descripcion>Rediseño web Madrid</descripcion>
      <duracion>5</duracion>
      <fechaInicio>3/5/1999</fechaInicio>
      <coste>13232</coste>
      <participantes>
        <participante id="E003"/>
        <participante id="E004"/>
        <participante id="E007"/>
      </participantes>
    </proyecto>
    <proyecto idProy="P002" estado="suspendido">
      <descripcion>Implantación servidor Rumasa</descripcion>
      <duracion>1</duracion>
      <fechaInicio>2/6/2001</fechaInicio>
      <presupuesto>54345</presupuesto>
      <participantes>
        <participante id="E001"/>
        <participante id="E002"/>
        <participante id="E006"/>
      </participantes>
    </proyecto>
    <proyecto idProy="P003">
```



```
<descripcion>Sitio Puebla del Río</descripcion>
<duracion>10</duracion>
<fechaInicio>8/5/2002</fechaInicio>
<presupuesto>13232</presupuesto>
<participantes>
  <participante id="E001"/>
  <participante id="E003"/>
  <participante id="E008"/>
</participantes>
</proyecto>
<proyecto idProy="P004">
  <descripcion>Patagon Internet Bank</descripcion>
  <duracion>13</duracion>
  <fechaInicio>6/6/2002</fechaInicio>
  <presupuesto>43231</presupuesto>
  <participantes>
    <participante id="E002"/>
    <participante id="E006"/>
    <participante id="E007"/>
  </participantes>
</proyecto>
</proyectos>
</empresa>
```

2. Escriba un archivo XSL que permita obtener un documento HTML en el que aparezcan los proyectos en ejecución. Cada proyecto aparecerá en una tabla separada donde deberá constar el identificador del proyecto, la fecha de comienzo, la duración estimada, su presupuesto y los nombres y áreas de cada persona que interviene en él.

Puntuación: 1 punto.

```
<?xsl version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" indent="yes"/>
<xsl:template match="/">
<html><head><title>Listado de proyectos</title></head>
<body>
  <xsl:for-each select="empresa/proyectos/proyecto">
    <xsl:if test="not(@estado)">
      <table border="1">
        <tr>
          <td colspan="2">
            <b>Proyecto:</b>(<xsl:value-of select="@idProy"/>)
            <xsl:value-of select="descripcion"/></td>
          </tr>
          <tr>
            <td><b>Fecha:</b><xsl:value-of select="fechaInicio"/></td>
            <td><b>Presupuesto:</b><xsl:value-of select="presupuesto"/></td>
          </tr>
          <tr>
            <th colspan="2">Participantes</th>
          </tr>
          <tr>
            <td><b>Participante</b></td>
            <td><b>Area</b></td>
          </tr>
          <xsl:for-each select="participantes/participante">
            <tr>
              <xsl:for-each select="id(@id)">
                <td><xsl:value-of select="."/></td>
                <td><xsl:value-of select="@area"/></td>
              </xsl:for-each>
            </tr>
          </xsl:for-each>
        </table>
      </xsl:if>
    </xsl:for-each>
  </body>
</html>
```



```
        </tr>
    </xsl:for-each>
</table>
<p> </p>
</xsl:if>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```