



### Cuadernillo de examen

Asignatura: **Interacción Hombre-Máquina / Interacción Persona-Computadora**  
Titulación: **Ingeniero en Informática / Ingeniero Técnico en Informática**  
Curso: **2º** Carácter: **Optativa /Obligatoria**  
Convocatoria: **Final Junio 2006** Curso académico: **2005/2006**  
Duración aproximada: **3 horas**

Código: **208**  
Plan de estudios: **2000/2002**  
Especialidad: **Común**  
Turno: **Tarde**

## Solución propuesta

### Programación en VB.NET

#### Parte teórica

Seleccione la opción correcta en el cuadernillo de examen. Sólo existe una opción verdadera, si considera que hay más de una, justifíquelo junto a la pregunta. Cada pregunta bien contestada valdrá 0,2 puntos. Cada opción mal contestada restará 0,1 puntos. Las preguntas no contestadas no restan puntos.

- Para añadir un elemento en un objeto de la clase `ListBox`...**
  - Hay que llamar al método `Add` de la colección `Items`.
  - Hay que llamar al método `AddNew` del objeto.
  - Hay que llamar al método `Add` del objeto.
- En VB.NET, para visualizar un formulario...**
  - Sólo hay que crear una instancia del formulario.
  - Se llama al método `Show`
  - Es necesario poner la propiedad `visible` a `True`.
- En un formulario modal...**
  - Se abre en el mismo hilo de ejecución desde dónde se ha abierto.
  - Se abre llamando al método `ShowModal`
  - Se abre un nuevo hilo de ejecución que se termina al cerrar el formulario.
- Un objeto de la clase `RadioButton`...**
  - Puede presentar tres estados..
  - Puede presentar dos estados.
  - Tiene una propiedad `Value` que toma los valores `True` o `False`
- El evento `ValueChanged` de un control `HScrollBar`...**
  - Se produce cuando se modifica el valor de la propiedad `Value` interactuando con la interfaz.
  - Se produce cuando se modifica el valor de la propiedad `Value` mediante código
  - Las dos respuestas son correctas.
- Una de las diferencias entre un objeto de la clase `ListBox` y otro de la clase `ComboBox`...**
  - Es que el `ComboBox` no tiene la propiedad `Sorted`.
  - Es que el `ComboBox` puede tomar 5 estilos distintos.
  - Es que el `ComboBox` no puede tener selección múltiple.
- Dentro de ADO.NET...**
  - El objeto `Connection` es independiente del gestor de base de datos utilizado.
  - El objeto `Command` es independiente del gestor de base de datos utilizado.
  - El objeto `DataSet` es independiente del gestor de base de datos utilizado.
- El método `Find`...**
  - Precisa de la existencia de una clave primaria.
  - Permite buscar las filas que cumplan una condición.
  - Las dos respuestas son correctas.



9. Para enlazar un **DataGrid** con una tabla de un **DataSet**...
- Es necesario establecer la propiedad `DataSource` del control.
  - Es necesario establecer la propiedad `DataTable` del control.
  - Es necesario establecer la propiedad `DataBase` del control.
10. En **ADO.NET**, el método **Update** del adaptador de datos...
- Se utiliza para actualizar el origen de datos.
  - En modo conectado se utiliza para actualizar una tabla.
  - Se utiliza para actualizar una fila de una tabla del `DataSet`.

**Responda brevemente a las siguientes preguntas (0,75 puntos cada una)**

1. **Arquitectura de .NET Framework ¿Qué componentes tiene? ¿Cuál es la función de cada uno de ellos?**
2. **La clase Array ¿Cómo se declara un array? ¿Cómo se instancia un array? Explique las distintas formas de asignar el número de elementos de un array. Arrays dinámicos. Modificación del tamaño en un array dinámico.**

**Parte práctica**

1. Se tienen almacenados en un control `ListBox` una serie de elementos de tipo `Persona`. Por cada persona se almacenan los siguientes datos:
  - DNI, una cadena.
  - Nombre, una cadena
  - Sexo, una cadena (contiene una "H" para hombre y una "M" para mujer).

Los datos del `ListBox` se gestionan desde el siguiente formulario:

Se pide:

- a) Codificar la estructura `Persona`.

```
Structure Persona
    Dim DNI As String
    Dim nombre As String
    Dim sexo As String

    Sub New(ByVal d As String, ByVal n As String, ByVal s As String)
        DNI = d
        nombre = n
        sexo = s
    End Sub

    Overrides Function toString() As String
        Return nombre
    End Function
End Structure
```

**Puntuación: 0,5 puntos**

- b) Codificar el evento necesario para que al seleccionar una persona de la lista aparezcan sus datos en los controles `txtDNI` y `txtNombre`, y se marque la opción `Hombre` o `Mujer`.



```
Private Sub lstPersonas_SelectedIndexChanged( _  
    ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) _  
    Handles lstPersonas.SelectedIndexChanged  
    'Si no hay elementos seleccionados no hago nada  
    If lstPersonas.SelectedIndex = -1 Then  
        Exit Sub  
    End If  
  
    txtDNI.Text = lstPersonas.SelectedItem.DNI  
    txtNombre.Text = lstPersonas.SelectedItem.nombre  
    If lstPersonas.SelectedItem.sexo = "H" Then  
        Hombre.Checked = True  
    Else  
        Mujer.Checked = True  
    End If  
End Sub
```

**Puntuación: 0,5 puntos**

- c) Codificar el evento necesario para que al pulsar el botón Añadir, se añada un nuevo elemento en el ListBox. El evento debe comprobar que no existe ninguna otra persona con ese DNI, en caso contrario apareciera un mensaje indicándolo.

```
Private Sub btnAñadir_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles btnAñadir.Click  
    Dim p As New Persona  
    p.DNI = txtDNI.Text  
    p.nombre = txtNombre.Text  
    If Hombre.Checked Then  
        p.sexo = "H"  
    Else  
        p.sexo = "M"  
    End If  
    'Buscar el DNI de la persona  
    For Each item As Persona In lstPersonas.Items  
        If item.DNI = p.DNI Then  
            MessageBox.Show("La persona ya existe", "Ejercicio1", _  
                MessageBoxButtons.OK, _  
                MessageBoxIcon.Information)  
            Exit Sub  
        End If  
    Next  
    lstPersonas.Items.Add(p)  
End Sub
```

**Puntuación: 0,75 puntos**

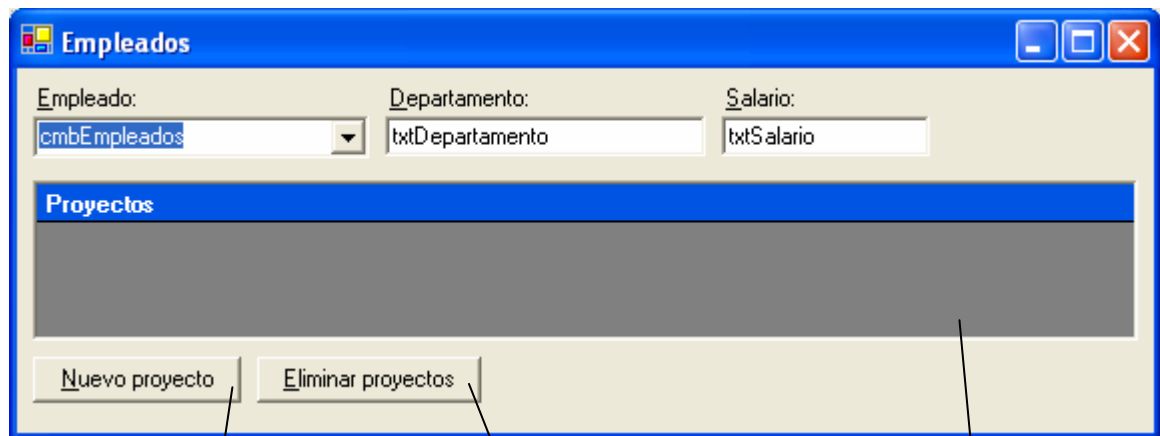
2. Una empresa gestiona los proyectos que realizan sus empleados en una base de datos Access llamada empresa.mdb. Entre otras, contiene las siguientes tablas:

Tabla Empleados			
Nombre del campo	Tipo	Contenido	Observaciones
IdEmpleado	Autonumérico	Código del empleado	Clave primaria
Nombre	Texto	Nombre del empleado	
Departamento	Texto	Nombre del Departamento	
Salario	Numérico	Sueldo del empleado	
Plus	Numérico	Plus del empleado	Se calcula a partir de los proyectos que ha finalizado, a razón de un 5% del presupuesto de cada proyecto terminado



Tabla Proyectos			
Nombre del campo	Tipo	Contenido	Observaciones
IdProyecto	Texto	Código del proyecto	Junto con el campo IdEmpleado constituye la clave primaria.
IdEmpleado	Autonumérico	Código del empleado	Clave externa (Empleados)
Nombre	Texto	Nombre del proyecto	
Inicio	Fecha	Fecha de inicio del proyecto	
Fin	Fecha	Fecha de finalización del proyecto	
Presupuesto	Numérico	Presupuesto total del proyecto	

Los proyectos de los empleados se gestionan con el siguiente formulario:



btnNuevo

btnCalcularPlus

dgProyectos

Se pide:

- a) Realizar las declaraciones necesarias para poder realizar las operaciones con las tablas de la base de datos.

```
Imports System.Data
Imports System.Data.OleDb
...
Private cn As New OleDbConnection
Private daEmpleado As OleDbDataAdapter
Private daProyecto As OleDbDataAdapter
Private ds As New DataSet
Private nombreArchivo = "..\empresa.mdb"
...
'Crear y abrir la conexión
cn.ConnectionString = "PROVIDER=Microsoft.Jet.OleDB.4.0; " & _
    "Data Source = " & nombreArchivo
cn.Open()
'Llenar el DataSet
daEmpleado = New OleDbDataAdapter("SELECT * FROM Empleados", cn)
daProyecto = New OleDbDataAdapter("SELECT * FROM Proyectos", cn)
daEmpleado.Fill(ds, "Empleados")
daProyecto.Fill(ds, "Proyectos")
cn.Close()

'Crear las claves primarias
'...para empleados
Dim claveEmp(0) As DataColumn
claveEmp(0) = ds.Tables("Empleados").Columns("IdEmpleado")
ds.Tables("Empleados").PrimaryKey = claveEmp

'...para Proyectos
```



```
Dim claveProy(1) As DataColumn
claveProy(0) = ds.Tables("Proyectos").Columns("IdProyecto")
claveProy(1) = ds.Tables("Proyectos").Columns("IdEmpleado")
ds.Tables("Proyectos").PrimaryKey = claveProy

'Generar órdenes de actualización automáticamente
Dim cbProy As OleDbCommandBuilder = New OleDbCommandBuilder(daProyecto)
Dim cbEmp = New OleDbCommandBuilder(daEmpleado)
```

**Puntuación: 0,5 puntos**

- b) Al arrancar el programa, se deberán cargar en el ComboBox cmbEmpleados todos los empleados de la empresa.

```
Private Sub Form1_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load
...
...
'Cargar los empleados en el ComboBox
For Each dr As DataRow In ds.Tables("Empleados").Rows
    cmbEmpleados.Items.Add(New Empleado( _
        dr.Item("Nombre"),
        dr.Item("IdEmpleado")))
Next
End Sub
```

**Puntuación: 0,25 puntos**

- c) Al seleccionar un empleado del ComboBox, se cargarán en los cuadros de texto los datos de ese empleado. Al mismo tiempo, en el DataGrid dgProyectos se cargarán todos los proyectos del empleado.

```
Private Sub cmbEmpleados_SelectedIndexChanged( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles cmbEmpleados.SelectedIndexChanged

' Cargar los datos del empleado seleccionado
Dim Empleado As DataRow = _
    ds.Tables("Empleados").Rows.Find(cmbEmpleados.SelectedItem.IdEmpleado)
txtDepartamento.Text = Empleado.Item("Departamento")
txtSalario.Text = Empleado.Item("Salario")

' Cargar los proyectos
Dim dv As DataView = ds.Tables("Proyectos").DefaultView
dv.RowFilter = "IdEmpleado = " & Empleado.Item("IdEmpleado")
dgProyectos.DataSource = dv
End Sub
```

**Puntuación: 0,5 puntos**

- d) Al pulsar sobre el botón btnNuevoProyecto, se creará un nuevo proyecto. Para cargar los datos del proyecto se abrirá una ventana secundaria modal llamada frmNuevoProyecto,



El IdEmpleado y el nombre del proyecto se tomarán del empleado actual, y al pulsar en esta ventana el botón Aceptar se insertará una nueva fila en la tabla Proyectos. El IdEmpleado será el del empleado actual y el resto de datos se tomarán de la ventana Nuevo Proyecto.

```
Private Sub btnNuevo_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnNuevo.Click
    If cmbEmpleados.SelectedIndex = -1 Then
        Exit Sub
    End If
    Dim frm As New frmNuevoProyecto
    frm.txtDepartamento.Text = txtDepartamento.Text
    frm.txtIdEmpleado.Text = cmbEmpleados.SelectedItem.idEmpleado
    If frm.ShowDialog() = DialogResult.OK Then
        'Crear un nuevo proyecto
        Dim nuevo As DataRow = ds.Tables("Proyectos").NewRow
        nuevo.Item("IdProyecto") = frm.txtIdProyecto.Text
        nuevo.Item("IdEmpleado") = frm.txtIdEmpleado.Text
        nuevo.Item("Nombre") = frm.txtNombreProyecto.Text
        nuevo.Item("Inicio") = frm.txtFechaInicio.Text
        nuevo.Item("Fin") = frm.txtFechaFin.Text
        nuevo.Item("Presupuesto") = frm.txtPresupuesto.Text
        Try
            ds.Tables("Proyectos").Rows.Add(nuevo)
        Catch ex As System.Data.OleDb.OleDbException
            'Si se da un error de base de datos,
            'aparece un cuadro de diálogo advirtiendo del error
            MessageBox.Show(ex.Message, Me.Text, _
                MessageBoxButtons.OK, _
                MessageBoxIcon.Information)
        End Try
    End If
End Sub
```

Puntuación: 1 punto

### Interacción hombre-máquina

Seleccione la opción correcta en el cuadernillo de examen. Sólo existe una opción verdadera, si considera que hay más de una, justifíquelo junto a la pregunta. Cada pregunta bien contestada valdrá 0,15 puntos. Cada opción mal contestada restará 0,05 puntos. Las preguntas no contestadas no restan puntos.

1. Para conseguir la *affordance* en una interfaz de usuario...
  - Es necesario centrarse en los conceptos de flexibilidad y consistencia.
  - Es necesario centrarse en los conceptos de visibilidad y evidencia.



- Hay que aplicar las reglas de oro de Mandel.
2. **Para conseguir la usabilidad de un sistema interactivo...**  
 Hay que seguir los principios de facilidad de aprendizaje, flexibilidad y consistencia o solidez.  
 Basta con seguir las leyes de agrupación de objetos.  
 No es cierta ninguna de las dos preguntas anteriores.
3. **La flexibilidad de un sistema interactivo...**  
 Hace referencia a su solidez.  
 Permite que el sistema tenga capacidad de síntesis.  
 Hace referencia a las distintas formas en que el usuario y el sistema intercambian información.
4. **La interfaz de línea de órdenes**  
 Es un estilo de interacción.  
 Es un paradigma de interacción.  
 Es una metáfora.
5. **Las interfaces de manipulación directa**  
 Permiten manipular directamente con las bases de datos.  
 Crean una representación visual de las acciones que se pueden realizar a partir de la manipulación de los objetos de la interfaz.  
 Se utilizan en las interfaces dirigidas por agentes.
6. **Los asistentes**  
 Guían al usuario en la ejecución de una tarea.  
 Trabajan sin intervención del usuario.  
 Aprenden el comportamiento del usuario mediante técnicas de inteligencia artificial.
7. **Una ventana modal**  
 Impide que el usuario realice otra tarea hasta que se cierra la ventana.  
 Quita el control al usuario.  
 Las dos respuestas anteriores son correctas.
8. **La retroalimentación o *feedback* de un sistema interactivo**  
 Da a los usuarios un mayor control del sistema.  
 Reduce la carga de memoria a largo plazo..  
 Proporciona una interfaz de usuario consistente.
9. **Los botones en una interfaz Windows pueden ser**  
 Botones de órdenes y botones de menú.  
 Botones de órdenes, botones de radio y casillas de verificación.  
 Sólo botones de órdenes.
10. **Según las recomendaciones de Microsoft para la interfaz Windows, un cuadro de lista (ListBox)**  
 No debería tener más de 7 opciones.  
 Permite que el usuario elija entre alguna de las opciones que presenta o que el introduzca por teclado su propia opción.  
 Sólo permite que el usuario seleccione una de las opciones que presenta.

**Conteste brevemente a estas preguntas (1 punto)**

1. Métodos de evaluación por indagación. ¿En qué consiste? ¿Quiénes intervienen? Explique los métodos por indagación que conozca.