



Cuadernillo de examen

ASIGNATURA	INTERACCIÓN HOMBRE MÁQUINA INTERACCIÓN PERSONA-COMPUTADORA	CÓDIGO	208
CONVOCATORIA	Septiembre 2004	PLAN DE ESTUDIOS	2000/2002
ESPECIALIDAD	Común	CURSO	2º
TURNO	Mañana	CURSO ACADÉMICO	2003/2004
CARÁCTER	Anual	PROGRAMA	Ingeniería Informática Ingeniería Técnica en Informática

Soluciones al examen

PRIMERA PARTE: 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 0,2 puntos. Cada opción mal contestada restará 0,1 puntos. Las preguntas no contestadas no restan puntos.

- La Ergonomía se encarga de**
 - Adaptar el sistema y toda la interfaz al perfil del usuario
 - Ajustar bien el significado de los símbolos a las tareas
 - Transmitir información sobre objetos utilizando metáforas.
- La interfaz de usuario por línea de órdenes:**
 - Ha sido el último de los estilos de interfaz que se ha utilizado
 - Ha sido el primero de los estilos de interfaz que se ha utilizado
 - Ninguna de las opciones anteriores es cierta.
- Los tipos de paradigmas de interacción actuales son:**
 - El ordenador se sobremesa y la realidad virtual.
 - La computación ubicua y la realidad aumentada.
 - Las dos respuestas anteriores son ciertas.
- La computación ubicua pretende:**
 - Extender la capacidad de computación al entorno del usuario mediante pequeños dispositivos que permiten interactuar conectados a una red o sistema de información.
 - Reducir las interacciones con el ordenador, utilizando la información del entorno como una entrada implícita.
 - Trasladar información adicional al mundo real, en lugar de trasladarla al ordenador.
- La usabilidad es:**
 - La capacidad de una interfaz gráfica para transmitir metáforas.
 - La medida en la que un producto se puede utilizar por determinados usuarios para conseguir objetivos específicos con efectividad.
 - Ninguna de las opciones anteriores es correcta.
- Las leyes de agrupación:**
 - Incluyen, entre otros, los principios de proximidad y simetría.
 - Incluyen, entre otros, los principios de similitud y transparencia.
 - Intentan identificar las claves más relevantes de la percepción de objetos en conjuntos.
- Las metáforas pueden ser de los siguientes tipos:**
 - Visual, sonora y acústica.
 - Visual, verbal y global.
 - Ninguna de las respuestas anteriores es correcta.
- Los estándares de facto:**
 - Se crean a partir de comités avalados por gobiernos o instituciones acreditadas.
 - Son, por ejemplo, los desarrollados por la agencia ISO.
 - Se crean a partir de productos de gran aceptación por el mercado o realizados a partir de desarrollos de amplia difusión realizados por grupos industriales o de investigación
- Una guía de estilo**
 - Es una regla de obligado cumplimiento en el diseño de una interfaz.
 - Ayuda a mantener la consistencia entre un sistema o familia de sistemas.
 - Puede ser comercial, corporativa o estatal.
- La evaluación de un sistema interactivo:**
 - Debe ser realizada por especialistas en la materia.



- Desarrolla principalmente como fase previa a la comercialización y puesta en marcha de un sistema interactivo.
- Se debe desarrollar a lo largo de todo el ciclo de vida del sistema.

Indique en el cuadernillo de examen si cada afirmación si es verdadera o falsa. Cada pregunta bien contestada valdrá 0,2 puntos. Cada pregunta mal contestada restará 0,1 puntos. Las preguntas no contestadas no restan puntos.

1. El objetivo del diseño grafico de una interfaz es mostrar la información de forma atractiva y legible.
 V F
2. Se define como interacción a todos los intercambios que se producen entre los dispositivos de entrada y salida de un sistema.
 V F
3. Las ventanas en un sistema WIMP son áreas de la pantalla que actúan como si fueran terminales independientes.
 V F
4. La evaluación heurística es uno de los métodos de test existentes.
 V F
5. Una guía de estilo comercial es una norma de obligado cumplimiento.
 V F
6. Entre los métodos de evaluación por inspección se encuentra el recorrido cognitivo.
 V F
7. Las metáforas son recursos del lenguaje que se utilizan para comunicar un concepto abstracto de un modo familiar y accesible.
 V F
8. El principio de consistencia de un sistema interactivo implica proporcionar mecanismos de interacción tanto a usuarios noveles como a usuarios expertos.
 V F
9. Las interfaces WIMP están intrínsecamente relacionadas con la computación ubicua.
 V F
10. La *affordance* consiste en hacer evidente la parte del objeto sobre la que se puede realizar una acción y como realizar dicha acción
 V F



3. Principios generales de la usabilidad.

4. Estilos de interacción: definición, tipos y características principales



SEGUNDA PARTE: LENGUAJES DE PROGRAMACIÓN VISUAL

Preguntas teórico-práctica

1. Procedimientos y funciones de usuario en VB.NET. Declaración. Paso de argumentos. Tipos de argumentos.
2. Tipos de datos en VB.NET. Describa los distintos tipos de datos y sus características principales.
3. Tratamientos de listas en Windows Forms. Describa las similitudes y diferencias más importantes entre la clase ListBox y la clase ComboBox. Describa sus propiedades, métodos y eventos más importantes.

Puntuación: 1 punto cada pregunta

Parte práctica

Ejercicio 1

Se desea gestionar una agenda de citas en memoria. Por cada cita se almacenará la fecha de la cita y un texto con las actividades a realizar dicho día. Para ello se utilizará el siguiente formulario (se indican los nombres de los controles):

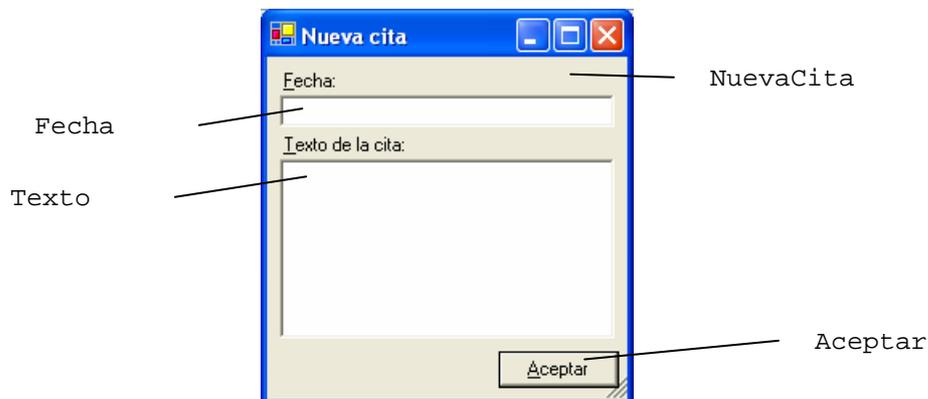


1. Realice las declaraciones necesarias para almacenar los datos en memoria.

Puntuación: 0,5 puntos

```
Public Structure cita
    Public textoCita As String
    Public fechaCita As Date
    Public Sub New(ByVal t As String, ByVal f As Date)
        textoCita = t
        fechaCita = f
    End Sub
    Public Overrides Function toString() As String
        Return fechaCita
    End Function
End Structure
```

2. Al pulsar el botón Agregar, se cargará una nueva cita. Para ello se utilizará el siguiente formulario:



El cuadro de texto `Texto`, podrá incluir todas las citas de ese día. Para agregar una nueva cita, se deberá comprobar que el dato introducido en el cuadro de texto `Fecha`, se trata de una fecha válida, en caso contrario, no se almacenará



la cita y aparecerá un cuadro de diálogo indicando la incidencia. Además en el ComboBox Citas del primer formulario, se añadirá la fecha de la cita.

Puntuación: 1 punto.

```
'Botón Agregar del formulario Agenda
Private Sub Agregar_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles Agregar.Click
    Dim frm As New NuevaCita
    frm.ShowDialog()
    If Not IsDate(frm.Fecha.Text) Then
        MessageBox.Show("Formato de fecha incorrecto", _
            "Agenda", MessageBoxButtons.OK, _
            MessageBoxIcon.Exclamation)
    Else
        'Si es una fecha correcta...
        'almacena los datos del formulario NuevaCita en el ComboBox
        Dim c As cita = New cita(frm.texto.Text, _
            frm.Fecha.Text)
        citas.Items.Add(c)
    End If
End Sub

'Botón Aceptar del formulario NuevaCita
Private Sub Aceptar_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Aceptar.Click
    Me.Close()
End Sub
```

3. Al pulsar sobre el botón Editar, se volverá a mostrar el formulario NuevaCita pero con los datos del elemento seleccionado del ComboBox Citas. En el caso de que no se hubiera seleccionado ninguna fecha aparecerá un mensaje indicando la incidencia. El título de la ventana será la fecha de la cita. Al pulsar el botón Aceptar, se cerrará el formulario y se modificarán los datos de la cita tanto en memoria como en el ComboBox.

Puntuación: 1 punto.

```
Private Sub Editar_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Editar.Click
    'Si hay algún elemento seleccionado en el Combo...
    If citas.SelectedIndex <> -1 Then
        Dim frm As New NuevaCita
        Dim c As cita = citas.SelectedItem
        'Carga los datos de la cita en el formulario NuevaCita
        frm.Text = c.fechaCita
        frm.Fecha.Text = c.fechaCita
        frm.texto.Text = c.textoCita
        frm.ShowDialog()
        If Not IsDate(frm.Fecha.Text) Then
            MessageBox.Show("Formato de fecha incorrecto", _
                "Agenda", MessageBoxButtons.OK, _
                MessageBoxIcon.Exclamation)
        Else
            'Si la fecha es correcta...
            Dim i As Integer = citas.SelectedIndex
            'Modifica el texto de la cita,
            c.textoCita = frm.texto.Text
            'elimina el elemento antiguo
            citas.Items.RemoveAt(i)
            'y añade en la misma posición el elemento modificado
            citas.Items.Insert(i, c)
        End If
    Else
        MessageBox.Show("Debe seleccionar una cita", "Agenda", _
            MessageBoxButtons.OK, _
            MessageBoxIcon.Exclamation)
    End If
End Sub
```

Ejercicio 2º



La Agencia de Viajes *Viajes Bonilla* ofrece nuevos destinos para el verano del 2004. Para ello cuenta con la base de datos *Viajes.mdb*, constituida por las siguientes tablas:

DESTINOS

CAMPO	TIPO	DESCRIPCION
Cod Destino	Texto	Código del destino. Campo clave
Nombre	Texto	Nombre del destino
Descripcion	Texto	Descripción breve del viaje
Max personas	Entero	Nº máximo de personas en ese viaje
Precio	Real	Precio por persona del paquete

RESERVAS

CAMPO	TIPO	DESCRIPCION
Cod Destino	Texto	Código del destino
Dni Cliente	Texto	Dni o pasaporte del cliente
Salida	Fecha	Fecha de salida
Num personas	Entero	Nº de personas que viajan
Confirmado	Lógico	Reserva confirmada o no

Para gestionar las reservas tenemos el siguiente formulario:

Labels pointing to the form elements:

- Fecha: points to the 'FECHA SALIDA' field.
- Destinos: points to the 'CODIGO DESTINO' dropdown menu.
- DNI: points to the 'DNI CLIENTE' text input field.
- NumPersonas: points to the 'Nº PERSONAS' text input field.
- Confirmada: points to the 'CONFIRMADA' checkbox.
- VerDestino: points to the 'Ver Destino' button.
- Agregar: points to the 'Agregar Reserva' button.

Como formulario auxiliar utilizamos el siguiente, en caso de pulsar el botón VerDestino:

Labels pointing to the form elements:

- DatosDestino: points to the window title bar.
- CodDestino: points to the 'CODIGO:' field.
- Nombre: points to the 'NOMBRE:' field.
- Descripción: points to the 'DESCRIPCIÓN' field.
- Precio: points to the 'PRECIO PERSONA' field.
- NumPersonas: points to the 'Nº PERSONAS' field.
- Volver: points to the 'Volver' button.

Se pide lo siguiente:



1. Declarar los objetos necesarios para gestionar la base de datos, almacenando ésta en memoria del cliente. Se incluirá el enlace de los controles con los campos de la tabla Reservas.

Puntuación: 1,5 puntos

```
Private cn As New OleDbConnection
Private daReservas As OleDbDataAdapter
Private daDestinos As OleDbDataAdapter
Private ds As New DataSet
Private Const nombreBBDD = "..\Viajes.mdb"

...
'Configurar acceso a datos
'Crear la conexión
cn.ConnectionString = "PROVIDER=Microsoft.jet.oledb.4.0; " & _
"Data Source= " & nombreBBDD
cn.Open()

'Crear los adaptadores de datos
daDestinos = New OleDbDataAdapter("SELECT * FROM Destinos", cn)
daReservas = New OleDbDataAdapter("SELECT * FROM Reservas", cn)

'Rellenar el dataset
daDestinos.Fill(ds, "Destinos")
daReservas.Fill(ds, "Reservas")

'Una vez cargados, se cierra la conexión
cn.Close()

'Generar órdenes automáticamente (necesario para agregar una reserva)
Dim cb As OleDbCommandBuilder = New OleDbCommandBuilder(daReservas)

'Para establecer como clave principal la columna Cod_Destino
'(necesario para buscar el destino mediante Find
Dim clave(0) As DataColumn
clave(0) = New DataColumn
clave(0) = ds.Tables("Destinos").Columns("Cod_Destino")
ds.Tables("Destinos").PrimaryKey = clave
```

2. Implementar el código del botón VerDestino, que muestra el formulario DatosDestino con los datos del destino cuyo código se encuentra en el combobox del formulario RESERVAS.

Puntuación: 1,5 puntos

```
Private Sub VerDestino_Click(ByVal sender As System.Object, _
                             ByVal e As System.EventArgs) _
                             Handles VerDestino.Click
'Buscar el código en la tabla Destinos
Dim encontrado As DataRow = ds.Tables("Destinos").Rows.Find(Destinos.Text)

'Crear una nueva instancia del formulario DatosDestino
'y mostrarlo
Dim frm As New DatosDestino
frm.CodDestino.Text = encontrado.Item("Cod_Destino")
frm.Nombre.Text = encontrado.Item("Nombre")
frm.Descripción.Text = encontrado.Item("Descripción")
frm.Precio.Text = encontrado.Item("Precio")
frm.NumPersonas.Text = encontrado.Item("Max_Personas")

frm.Show()
End Sub
```

3. Implementar el código del botón Agregar, que agrega a la tabla Reservas un nuevo registro conteniendo los datos de los controles del formulario RESERVAS. Los códigos de los viajes deben cargarse desde la tabla Destinos al iniciarse la aplicación.

Puntuación: 1,5 puntos

...



```
'En el procedimiento Reservas_Load
'para cargar al comienzo los códigos de los destinos
...
'Cargar el comboBox con los destinos
For Each dr As DataRow In ds.Tables("Destinos").Rows
    Destinos.Items.Add(dr.Item("Cod_Destino"))
Next

Private Sub Agregar_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles Agregar.Click
    'Crear un registro de reservas vacío
    'y rellenarlo con los datos del formulario
    Dim NuevaFila As DataRow = ds.Tables("Reservas").NewRow
    NuevaFila.Item("Cod_Destino") = Destinos.Text
    NuevaFila.Item("Salida") = Fecha.Text
    NuevaFila.Item("DNI_Cliente") = DNI.Text
    NuevaFila.Item("Num_Personas") = NumPersonas.Text
    If Confirmada.CheckState = CheckState.Checked Then
        NuevaFila.Item("Confirmado") = True
    Else
        NuevaFila.Item("Confirmado") = False
    End If

    'Agregar la nueva fila a la colección Rows
    ds.Tables("Reservas").Rows.Add(NuevaFila)

    'Actualizar el DataSet
    daReservas.Update(ds, "Reservas")
End Sub
```