Se trata de realizar un programa que simule el funcionamiento de un cajero automático. Las funciones que debe incluir son las siguientes:

- Ingresar / retirar dinero en múltiplos de 1000 obteniendo un resguardo de la operación con la fecha y hora, tipo de la operación (ingreso o retirada), importe y saldo.

- Cambiar el número secreto (el nuevo se pide 2 veces).

- Consultar saldo (con un resguardo como el de los ingresos pero sin tipo de operación ni importe).

- Sacar entradas para un espectáculo del que se hace publicidad en el cajero.

- Un botón para apagar y encender el cajero.

- Que al entrar en la pantalla aparezca la foto del cliente

NOTAS:

- La base de datos de clientes se entrega con el examen y consta de los siguientes campos:

Nombre	Texto	
Dni (#)	Texto	
Teléfono	Texto	
Foto	OLE	
Saldo	Moneda	
Clave	Numérico	(es el número secreto)

- La acción de meter la tarjeta se simula pidiendo el dni del cliente antes de su número secreto.

- La acción de expulsar la tarjeta se simula con un mensaje "Retire su tarjeta"

- La acción de ingresar dinero se simula con un mensaje "Introduzca el dinero"

- La forma de realizar el ejercicio así como el diseño de las pantallas se deja a gusto del programador.

- Los tickets para el espectáculo y los recibos de retirada de dinero se imprimirán en papel.

La empresa de neumáticos Sakamoto tiene una plantilla de comerciales que viajan alrededor del mundo vendiendo ruedas. Esta empresa necesita para ellos un programa, denominado "Cartas", que les permita, a través de sus portátiles, preparar y enviar cartas a los clientes de la ciudad y país que van a visitar. Este programa debe incluir las siguientes características:

- El programa solicita un nombre de usuario y contraseña. Si estos son correctos, se comienza la ejecución del programa.
- Un menú principal con las siguientes opciones: Inicio (sin pulsar aquí el programa no funciona), Salir (visualiza un mensaje de despedida personalizado para el comercial, espera 3 segundos y termina el programa) y Acerca de (con el nombre del programador).
- El programa trabaja con la base de datos "Cartas", compuesta por dos tablas:
    - Comerciales: con usuario, clave y nombre completo de los comerciales.
    - Clientes: con los campos nombre, ciudad, país y teléfono de contacto en el que están todos los clientes que los comerciales pueden visitar.
- Cuando un comercial viaja a una ciudad necesita saber quienes son los clientes que viven en ella y por lo tanto el programa debe permitirle realizar búsquedas por ciudad. Estas búsquedas también deben poder realizarse por país.
- Una vez realizada la consulta, se presentan los resultados y se da la opción al comercial de eliminar de la consulta aquellos clientes a los que por alguna razón no quiere enviar la carta de presentación.
- Cuando ha finalizado esta tarea, el comercial tiene en pantalla los clientes a los que va a enviar la carta de presentación. Pulsando un botón "Imprimir" se generará una carta por cada cliente con el siguiente texto:

> "Estimado/a <nombre>:
> En breve visitaré <ciudad> en <país>. Desearía poder contactar con usted. Según mis datos su teléfono de contacto es el <teléfono>. A mi llegada a <ciudad> le llamaré para tratar los asuntos que conciernan.
> Atentamente: <nombre del comercial>.

- Con el botón derecho se puede elegir el color del texto de la carta entre 4 opciones.
- Al medio minuto de entrar a utilizar el programa aparecerá un mensaje "Batería baja" que no debe interferir en la ejecución del programa. Medio minuto después se finalizará la ejecución del mismo.

La Compañía Estatal de Ferrocarriles de Guatemala tiene guardada en una base de datos todo lo referente a los recorridos de sus trenes. Tomándola como base, los responsables de la compañía desean realizar una aplicación que permita:

- Visualizar conjuntamente todos los recorridos de los trenes del país.
- Que pueda sobre ese listado elegir el precio de un trayecto y modificarlo.
- Subir el precio de todos los billetes de la compañía un porcentaje determinado.
- Comprar billetes. Para ello se permitirá:
    - Elegir ciudad de origen y de destino.
    - Se muestran todos los trayectos entre las dos ciudades y de entre ellos, el usuario elegirá el horario que quiera. Si el usuarios tarda más de 10 segundos en elegir un horario, la acción se cancelará.
    - El usuario puede en ese momento elegir entre ida o ida/vuelta. El precio del billete de ida/vuelta tiene un 20% de descuento.
- Cuando se elija un trayecto, se imprimirá un billete con: Fecha de emisión, ciudad de origen, ciudad de destino, precio del trayecto y horario. La operación de imprimir el billete se simulará visualizando el billete y un mensaje "Billete emitido" cuando se pulse un botón "Aceptar".

NOTAS:

- La base de datos se ofrece con el enunciado del examen. Los campos de la tabla son: Origen, Destino, Distancia, Precio y Horario
- La presentación de los trayectos se puede realizar utilizando cualquier control que se considere oportuno.

Origen	Destino	Distancia	Precio	Horario
Quezalteuango	Antigua	70	210	09.20-11.00
Quezalteuango	San José	75	220	11.20-15.00
Quezalteuango	Zacapa	105	315	09.20-11.00
Flores	Antigua	175	3200	19.20-01.00
Flores	Champerico	187	3300	21.00-22.30
San José	Zacapa	63	205	09.20-11.35
Huelhuetango	Quezalteuango	31	95	22.00-00.30
Huelhuetango	Antigua	48	130	21.00-22.55
Quezalteuango	Antigua	80	230	01.00-03.30
Flores	Antigua	155	3000	01.00-04.30
Flores	Antigua	133	2000	02.00-06.30
Quezalteuango	Antigua	66	200	12.00-16.30

Inicio

Visualizar recorridos

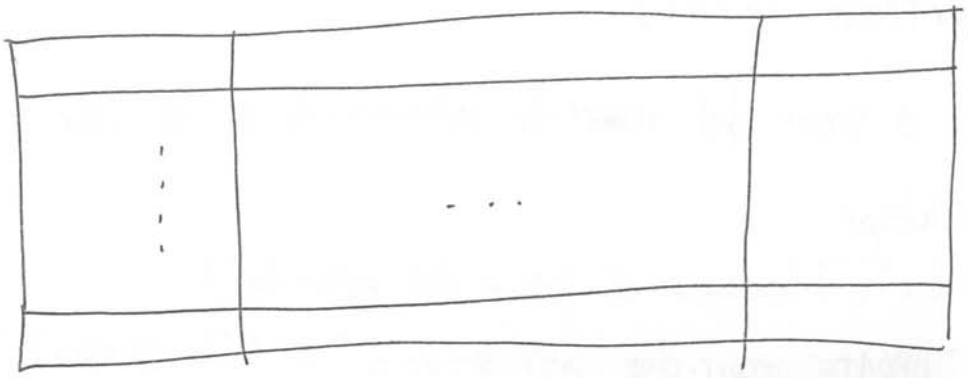
Modificar precio

Subir precios compañía

Comprar billetes

Salir

• Enabled = False



• Interval = 10000

• Enabled = False

```
Dim db As Database

mnuVisualizarRecorridos_Click()
    AbreBD
    VisualizarFlex "recorridos"
    mnuModificarPrecio.Enabled = True
    " SubirPrecios " "
    " ComprarBilletes " "
```

← abre el recordset crea el Flex con la consulta que se le pasa y cierra el recordset El Flex lo borra (.Clear) para poder usarlo en varias consultas.

```
mnuModificarPrecio_Click()

    'Modifica el precio del recorrido seleccionado en el Flex
Dim precio As Integer
precio = InputBox("Introduzca el precio del recorido")
db.Execute "UPDATE recorridos SET precio = " & CStr(precio) &
    "WHERE Origen = '" & flx.TextMatrix(flx.Row, 0) &
    "'AND destino = '" &                          1 &
    "'AND horario = '" &                          4 & Chr(34)

'Esta consulta es asi pq no hay clave y considero que la
clave es origen+destino+horario.

VisualizarFlex "recorridos"
```

```
mnuSubirPrecios_Click()
    Dim pctaje As Integer
    pctaje = InputBox ("Introduzca el % la subida de precios")
    db.Execute "UPDATE recorridos SET precio = precio *
        (1 + "&Str(pctaje)& " / 10)"

    VisualizarFlex "recorridos"
```

```
mnuComprarBilletes_Click()
    Dim origen As String
    .  destino ..  "
    origen = InputBox ("Introduzca origen")
    destino =  ..                destino
    VisualizarFlex " SELECT * FROM recorridos WHERE
       origen like '" & origen & "' AND destino LIKE '" &
       destino & "'"
    MsgBox "Elija el horario haciendo click"
    tmrCancelar.Enabled = True
```

```
flx_Click()
    Dim opc As String
    Dim precio As Currency
    opc = InputBox("Ida (i) o Ida y vuelta (v)")
    If opc = "i" Then
        precio = flx.TextMatrix(flx.Row, 3)
    Else                                        * 1.20
        "   =          "
    End If
    [illegible]
    frmBillete.Show
    frmBillete.Fecha.Caption = Date
        "      Origen    "  = txt.flx.TextMatrix(flx.Row, 0)
        "      Destino   "  =                    "          1)
        "      Precio    "  = precio
        "      Horario   "  =                    "          , 4)

tmrCancelar_Timer()
    VisualizarFlex "recorridos"
    tmrCancelar.Enabled = False


mnuSalir_Click()
    db.Close
    End
```


```
cmdImprimir_Click()
    cmdImprimir.Visible = False
    Me.PrintForm
    Unload Me
```

En la biblioteca del barrio de Hortaleza se desea crear una aplicación para que los usuarios de la misma puedan escribir mensajes entre ellos. En la biblioteca hay un único ordenador en el que los usuarios pueden escribir y leer sus mensajes. La forma de utilizar la aplicación es la siguiente: cada usuario tiene un nombre de usuario y una contraseña. Una vez que el usuario se valida en el ordenador, puede escribir mensajes a los demás y leer los mensajes que los demás usuarios del sistema han escrito para él y. Tras leer sus mensajes, el sistema le ofrecerá la opción de

guardar sus mensajes en un fichero, cuyo nombre será: usuario.txt

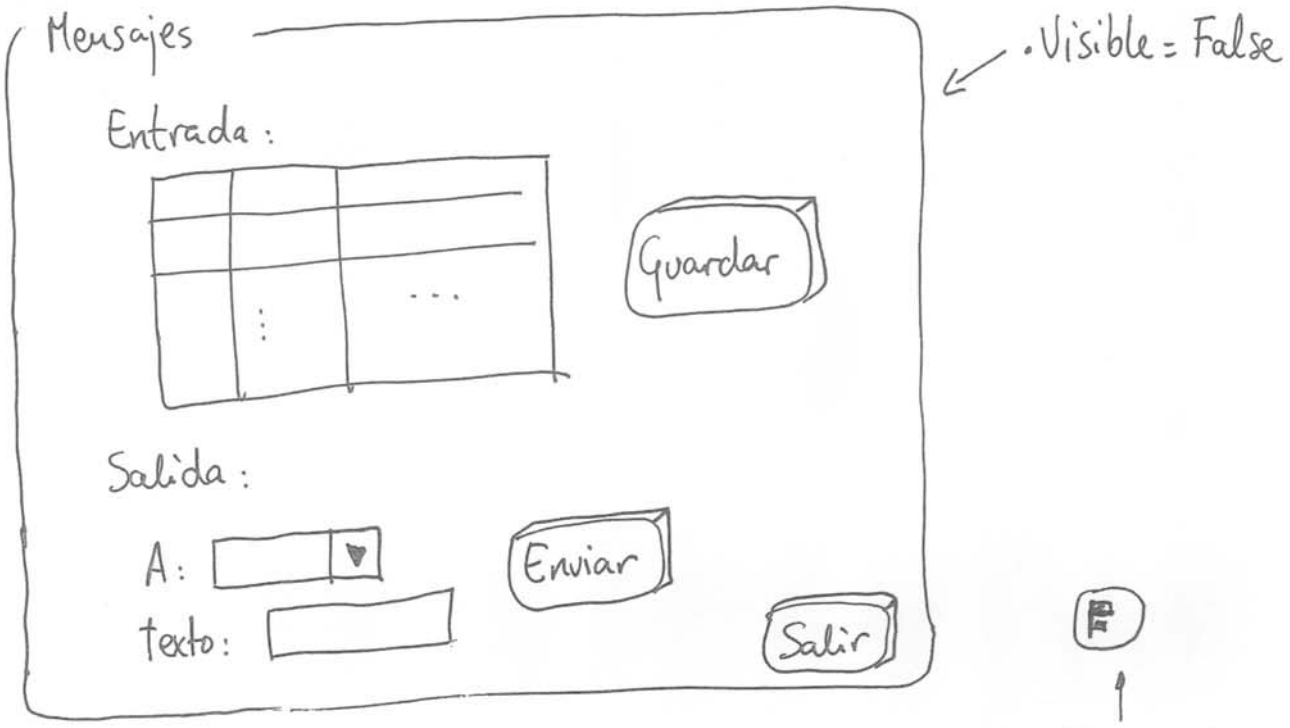
NOTAS:

- El programador creará la(s) tabla(s) que necesite para realizar la aplicación.

- La aplicación no debe crear ni borrar usuarios. Estos ya existen en la base de datos y no es necesario ocuparse de ellos.

- No es necesario borrar mensajes antiguos ni presentar sólo los mensajes no leídos. Se presentarán todos los mensajes que pertenezcan al usuario.

- El diseño de las pantallas y la forma de realizar la aplicación es de libre elección por parte del programador.



• Visible = False

dlgGuardar

Tablas:

USUARIOS (usuario, clave

MENSAJES (emisor, receptor, mensaje

```
Dim db As Database
cmdAceptar_Click()
    Dim db As Database
    Dim rst As Recordset
    Set db = Opendatabase(---)
    Set rst = db.OpenRecordset("select * from usuarios
        where usuario = '" & txtUsuario.Text & "' and clave = '"
        & txtClave.Text & chr(34)
    If rst.EOF And rst.BOF then
        MsgBox "Datos incorrectos"
    Else ——— rst.Close
             db.Close
        fraEntrada.Visible = False
        fraMensajes.Visible = True
        VisualizarFlex "SELECT emisor, mensaje FROM
            mensajes WHERE receptor = '" & txtUsuario.Text
            & "'"
        CargarCombo "SELECT usuario FROM usuarios"
```

abre bd y rst
y crea flex
pone cabecera
pone datos

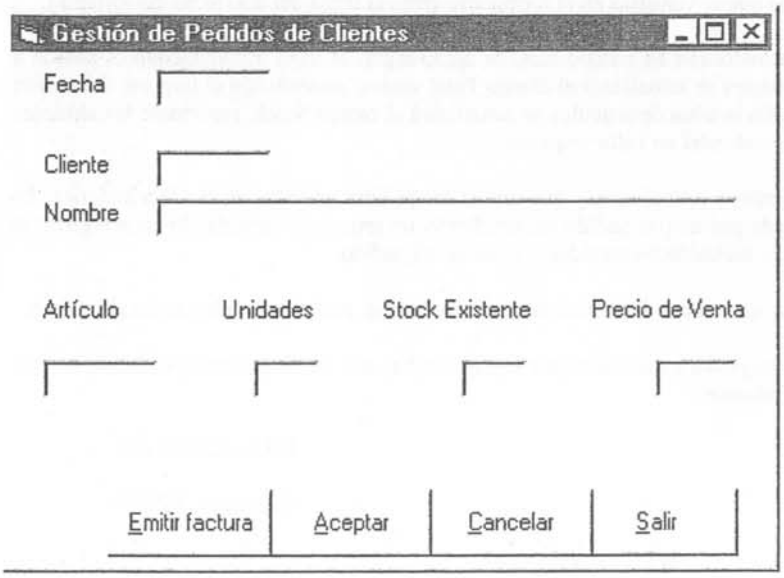
abre bd y rst
carga combo

```
cmdSalir_Click()
    db.Close
    End
```

```
cmdEnviar_Click ()

    db.Execute "INSERT INTO mensajes VALUES ('" &
      txtUsuario.Text & "','" & cmbA.text & "','" &
      txtTexto.Text & "')"

    'tb se puede hacer abriendo un recordset con la
    'tabla de mensajes y luego
    '   .AddNew
    '   asignar valor a los .Fields
    '   .Update


cmdGuardar_Click ()
    Dim i As Integer
    dlgGuardar."nFich"Filename = txtUsuario.Text & ".txt"
      "          .Filter = "Ficheros de texto|*.txt"
      "          .ShowSave
    nFich = FreeFile
    Open dlgGuardar.FileName For Output As nFich

    For i = 1 To flx.Rows - 1

        Print # nFich, flx.textMatrix(i,0) & ":" & flx.textMatrix(i,1)

    Next i

    Close nFich


tb  Print #nFich, flx.TextMatrix(i,0)    <- pondria el emisor en
     "        "        "          1           una linea y el msje
                                              en otra

tb  Write #nFich, flx.TextMatrix(i,0), flx.TextMatrix(i,1)
 (pondria los datos con el formato:  "pepe", "hola, que tal"
                                     "juan", "Mañana quedamos"
```

Examen Profesores Técnicos FP.

En un almacén se quiere mecanizar la captura de los pedidos realizados por los clientes. Para ello, se ha diseñado el siguiente formulario:



Los datos de los clientes y artículos se encuentran en una Base de Datos con las siguientes tablas:

- Tabla de Clientes. Tiene los siguientes campos:

  . Código de cliente. 6 caracteres alfanuméricos. Clave primaria.
  . N.I.F. 9 caracteres alfanuméricos.
  . Nombre. 50 caracteres alfabéticos.
  . Domicilio. 35 caracteres alfanuméricos.
  . Localidad. 35 caracteres alfabéticos.
  . Código Postal. 5 caracteres alfanuméricos.
  . Teléfono. 9 caracteres alfanuméricos.
  . Fax. 9 caracteres alfanuméricos.
  . Total ventas. 9 caracteres numéricos.

- Tabla de Artículos. Tiene los siguientes campos:

  . Código de artículo. 6 caracteres alfanuméricos. Clave primaria.
  . Descripción. 25 caracteres alfanuméricos.
  . Stock. 4 caracteres numéricos.
  . Stock mínimo. 3 caracteres numéricos.
  . Precio de compra. 5 caracteres numéricos.
  . Precio de venta. 5 caracteres numéricos.

Además de éstas dos tablas, la Base de Datos contiene una tercera tabla:

- Tabla histórica. Tiene los siguientes campos:

  . Cliente. 6 caracteres alfanuméricos.
  . Artículo. 6 caracteres alfanuméricos.
  . Unidades. 4 caracteres numéricos.
  . Fecha. Variable de tipo fecha.

Las tablas no poseen ninguna relación entre ellas.

Para capturar los pedidos de los clientes, en primer lugar hay que introducir el código del cliente y, posteriormente, se introducen los códigos de todos los artículos solicitados por el mismo cliente y las unidades pedidas de cada uno de los artículos. El nombre del cliente y el stock y precio de venta de cada artículo, se visualizarán de forma automática. Hay que controlar la inexistencia de los códigos de cliente y artículo en sus respectivas tablas.

Cuando se pidan unidades de un artículo, puede ocurrir que el stock de dicho artículo esté a 0, en cuyo caso se genera un pedido pendiente de servir con todas las unidades pedidas, o que no haya stock suficiente para servir, en cuyo caso se sirven las unidades existentes y se genera un pedido pendiente de servir con las unidades que no se hayan servido.

Los pedidos pendientes se grabarán en un fichero secuencial, con los campos código del cliente, código del artículo, unidades pendientes de servir y fecha del pedido.

Tanto la fecha del pedido como la fecha que se visualiza en el formulario serán la fecha del día (fecha del sistema).

Las tablas de clientes y artículos se actualizarán en tiempo real, es decir, según se vaya introduciendo el pedido a través del formulario. En la tabla de clientes se actualizará el campo Total ventas, acumulando el importe del pedido (unidades servidas * Precio de venta). En la tabla de artículos se actualizará el campo Stock, restándole las unidades pedidas, teniendo en cuenta que nunca contendrá un valor negativo.

Además de actualizar las tablas de clientes y artículos, hay que grabar los pedidos servidos en la tabla histórica. En ésta tabla se grabará un registro por cada pedido (un pedido es: un cliente, un artículo), conteniendo cada registro el código del cliente, el código del artículo, las unidades servidas y la fecha del pedido.

Cuando se pulsa el botón *Cancelar,* hay que anular las actualizaciones realizadas, pidiendo confirmación al usuario.

Una vez finalizada la introducción de los pedidos realizados por un cliente, hay que emitir la correspondiente factura. Para ello, hay que utilizar el siguiente informe:

Nombre de la empresa N.I.F.: xxxxxxxxx

Fecha: 99.99.99 Factura nº: 999999

Cliente: xxxxxx N.I.F.: xxxxxxxxx
Nombre: xx ..... 50 xx

Artículo	Unidades	Precio	Importe
xxxxxx	9.999	99.999	999.999.999

Total factura (I.V.A. inc.): 999.999.999

Para imprimir la factura no se podrá utilizar ningún generador de informes.

El número de factura comenzará por el 1.

Cuando se pulsa el botón *Salir* y hay alguna factura sin emitir, hay que preguntar al usuario si desea emitirla. Si no desea emitir la factura, habrá que anular las actualizaciones realizadas.

Al cargar el formulario hay que controlar la inexistencia de la Base de Datos y de todas las tablas. Si alguna de ellas no existe, se suspenderá la ejecución del proyecto. También se supenderá la ejecución del proyecto si las tablas de Clientes y Artículos existen pero están vacías.

Firma del alumno.

```
Dim db As Database
Dim rstClientes As Recordset
Dim rstArticulos As Recordset
Dim rstHistorico As Recordset
Dim wrkEspacio As WorkSpace
Dim nfactura As Integer
Dim totalFactura As Long
Dim facturaEnCurso As Boolean


Form_Load()
    On Error Goto ErrorCarga
    Set db = Opendatabase ("pedidos.mdb")
    Set rstClientes = db.OpenRecordset ("clientes", dbOpenDynaset)
    Set rstArticulos = db.OpenRecordset ("articulos", dbopenDynaset)
    Set rstHistorico = db.OpenRecordset ("historico", dbOpenDynaset)
    On Error Goto 0
    If (rstClientes.EOF And rstClientes.BOF) Or
       (rstArticulos.EOF And rstArticulos.BOF) then
            MsgBox " Tabla vacía"
            End
    End If
    Set wrkEspacio = DBEngine.Workspaces (0)
    txtFecha.Text = Date
    Exit Sub
ErrorCarga:
    MsgBox " BD errónea"
    End
End Sub
```

```
txtCliente_LostFocus()
    If      Len(txtCliente.text) > 0 then
        rstClientes.FindFirst "codigo = '" & txtCliente.text & "'"
        If rstClientes.NoMatch then
            MsgBox "Cliente no existe"
        Else
            txtNombre.text = rstClientes.Fields("nombre").Value
        End If
        wrkEspacio.BeginTrans
        PreparaFactura
End Sub

txtArticulo_LostFocus()
    If Len(txtArticulo.text) > 0 then
        rstArticulos.FindFirst "codigo = '" & txtArticulo.text & "'"
        If rstArticulos.NoMatch then
            MsgBox "Artículo no existe"
        Else
            txtStock.text = rstArticulos.Fields("stock").Value
            txtPVP.text =       "          ("pvp")    "
        EndIf

End Sub
```

```
' Acepta articulo
cmdAceptar_Click()
    If CInt(txtUnidades.text) <= 0 then
        MsgBox "Introduzca el nº de unidades"
    ElseIf CInt(txtStock.text) = 0 then
        PedidoPendiente CInt(txtUnidades.text)
    ElseIf CInt(txtUnidades.text) > CInt(txtStock.text) then
        PedidoPendiente CInt(txtUnidades.text) - _
                        CInt(txtStock.text)
        ActualizarStock 0
        ActualizarTotalVentas CLng(txtStock.text) * _
                              CLng(txtPrecioVenta.text)
        AnadirLineaFactura
        GrabarHistorico
    Else
        ActualizarStock CInt(txtStock.text) - _
                        CInt(txtUnidades.text)
        ActualizarTotalVentas CLng(txtUnidades.text) * _
                              CLng(txtPrecioVenta.Text)
        AnadirLineaFactura
        GrabarHistorico
    End If
End Sub
```

```
Sub Preparar Factura                                    "N.I.F.:  43214314"

                    "SALESA"
Printer.Print                        & vbCrLf & vbCrLf & vbCrLf
    & vbCrLf & vbCrLf & vbCrLf &

Printer.Print       "Fecha: " & Date & vbCrLf & vbCrLf &
    vbCrLf & vbCrLf & vbCrLf & vbCrLf & "Factura nº: " &
        CStr(nFactura)

Printer.Print  "Cliente: " & txtCliente.text &   vbCrLf &
    vbCrLf & vbCrLf & "N.I.F.: " & rstClientes.Fields("NIF").Value

Printer.Print              txtNombre.Text   'lo de limitarlo a
    '50 (en la factura) sería poniendo .MaxLength: 50 en la textBox


Printer.FontUnderLine = true                          'subrayado

Printer.Print  vbCrLf & vbCrLf & "Artículo" & vbCrLf &
    "Unidades" & vbCrLf & "Precio" & VbCrLf & "Importe"

Printer.FontUnderLine = False                         'desactiva
                                                       subrayado


facturaEnCurso = true

End Sub
```

```
Sub AnadirLineaFactura
  Dim importe As Long
  importe = CLng(txtUnidades.text) * CLng(txtPrecioVenta.text)

  Printer.Print vbCrLf & vbCrLf & txtArticulo.text &
      vbCrLf & txtUnidades.Text & vbCrLf &
      txtPrecioVenta.text & vbCrLf & CStr(importe)

  totalFactura = totalFactura + importe


End Sub
```

```
Sub ActualizarStock (stock As Integer)

    rstArticulos.Edit
    rstArticulos.Fields("stock").Value = stock
    rstArticulos.Update

End Sub


Sub ActualizarTotalVentas (ventas As Long)

    .

    rstClientes.Edit
    rstClientes.Fields("totalventas").Value = ventas
    rstClientes.Update

End Sub
```

```
cmdEmitirFactura_Click()
```

```
Printer.Print vbCrLf & ... & vbCrLf
& "Total Factura (IVA. inc.): " &
totalFactura * 1.16
```

```
    Printer.EndDoc                    'Imprime
    nFactura = nFactura + 1
    totalFactura = 0
    wrkEspacio.CommitTrans
    facturaEnCurso = False
End Sub

cmdCancelar_Click()
    Printer.KillDoc
    txtCliente.Text = ""
        Nombre
        Articulo
        Unidades
        Stock
        PVP
    totalFactura = 0
    wrkEspacio.Rollback
    facturaEnCurso = False
End Sub
```

```
Sub PedidoPendiente (pedido As Integer)

    Dim nFich As Integer

    nFich = FreeFile
    Open "pedidospendientes.txt" For Append As #nFich
    Print #nFich, txtCliente.Text, txtArticulo.Text, pedido,
          txtFecha.Text

    Close nFich

End Sub


Sub GrabarHistorico

    rstHistorico.AddNew
    rstHistorico.Fields("cliente").Value = txtCliente.text
                        ("articulo")     : txtArticulo.text
                        ("unidades")     : txtUnidades.text
                        ("fecha")        : txtFecha.text

    rstHistorico.AddNew

End Sub
```

```
cmdSalir_Click
    Dim resp As Integer

    If facturaEnCurso then
        resp = MsgBox ("¿Desea imprimir la factura?", vbYesNo)
        If resp = vbYes then
            cmdEmitirFactura_Click
        Else
            cmdCancelar_Click
        EndIf
    Else
        End
    EndIf
End Sub
```

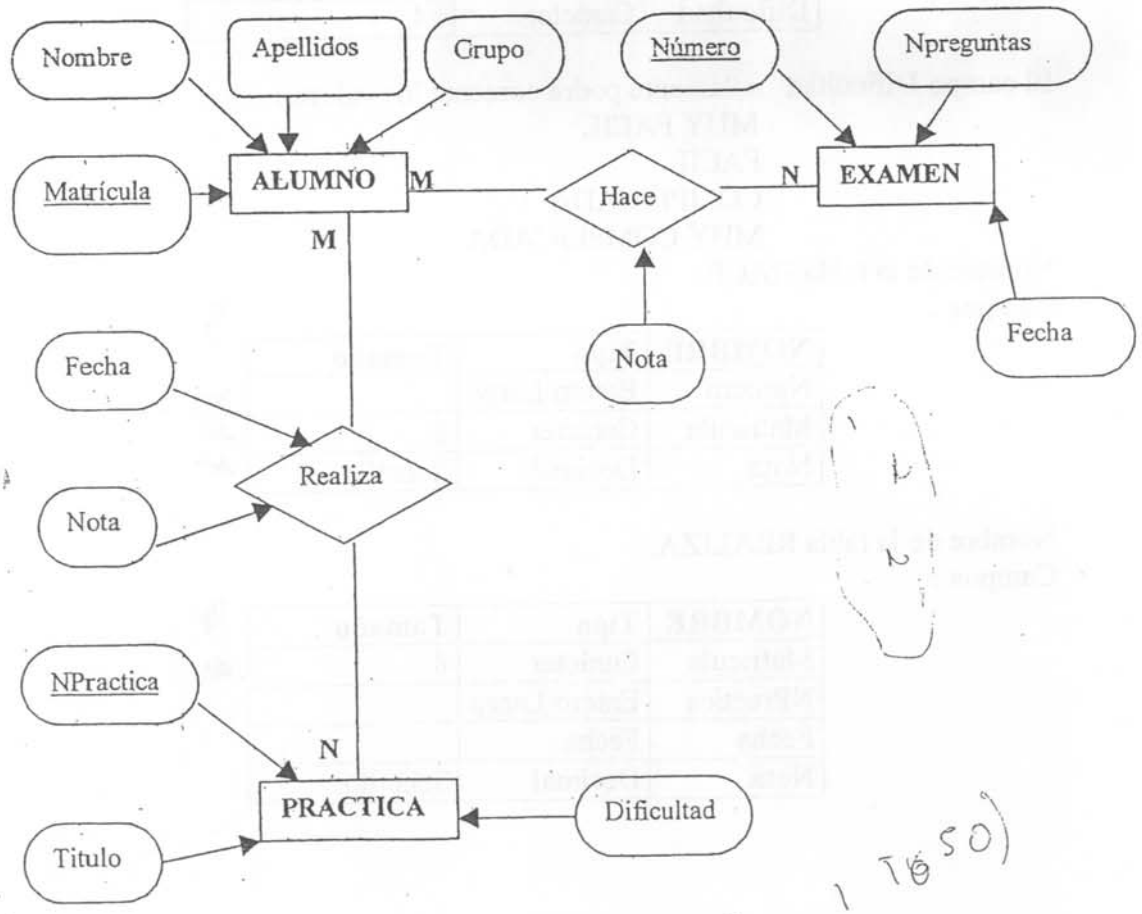
PRUEBA DE CONTENIDO DE CARÁCTER PRÁCTICO
DEL CUERPO DE PROFESORES TÉCNICOS DE FORMACIÓN PROFESIONAL
DE LA ESPECIALIDAD DE SISTEMAS Y APLICACIONES INFORMATICAS

Ejercicio 1

Los profesores del departamento de informática de cierto instituto de enseñanza secundaria de la Comunidad de Madrid han creado una base de datos para cada asignatura del departamento con la siguiente estructura con la información de los resultados de las pruebas realizadas a los alumnos en cada asignatura. Para realizar el diseño se sabe que:

- Los alumnos están definidos por su nº de matrícula, nombre y el grupo al que asisten a clase.
- Dichos alumnos realizan dos tipos de pruebas a lo largo del curso académico:
  1. Exámenes escritos: cada alumno realiza varios a lo largo del curso, y se definen por el nº de examen, el nº de preguntas de que consta y la fecha de realización (la misma para todos los alumnos que realizan el mismo examen y solamente se realizará un examen el mismo día). Evidentemente, importa almacenar la nota de cada alumno por examen.
  2. Prácticas: se realizan un nº indeterminado de ellas durante el curso académico, algunas serán en grupo y otras individuales. Se definen por un código de práctica, título y el grado de dificultad. En este caso, los alumnos pueden examinarse de cualquier práctica cuando lo deseen, debiéndose almacenar la fecha de realización y la nota obtenida.

El diagrama Entidad/Relación resultante es el siguiente:



Las tablas resultantes son las siguientes:
Nombre de la Tabla ALUMNO:
Campos :

NOMBRE	Tipo	Tamaño
Matricula	Carácter	8
Nombre	Carácter	20
Apellidos	Carácter	40
Grupo	Carácter	2

Nombre de la Tabla EXAMEN
Campos :

NOMBRE	Tipo	Tamaño
Numero	Entero Largo	Autoincrementado
NPreguntas	Entero	
Fecha	Fecha	

Nombre de la tabla PRACTICA
Campos :

NOMBRE	Tipo	Tamaño
Npractica	Entero Largo	Autoincrementado
Titulo	Carácter	20
Dificultad	Carácter	14

El campo Dificultad solamente podrá contener los valores:
MUY FACIL
FACIL
COMPLICADA
MUY COMPLICADA

Nombre de la tabla HACE:
Campos :

NOMBRE	Tipo	Tamaño
Numero	Entero Largo	
Matricula	Carácter	8
Nota	Decimal	Sencillo

Nombre de la tabla REALIZA:
Campos :

NOMBRE	Tipo	Tamaño
Matricula	Carácter	8
NPractica	Entero Largo	
Fecha	Fecha	
Nota	Decimal	Sencillo

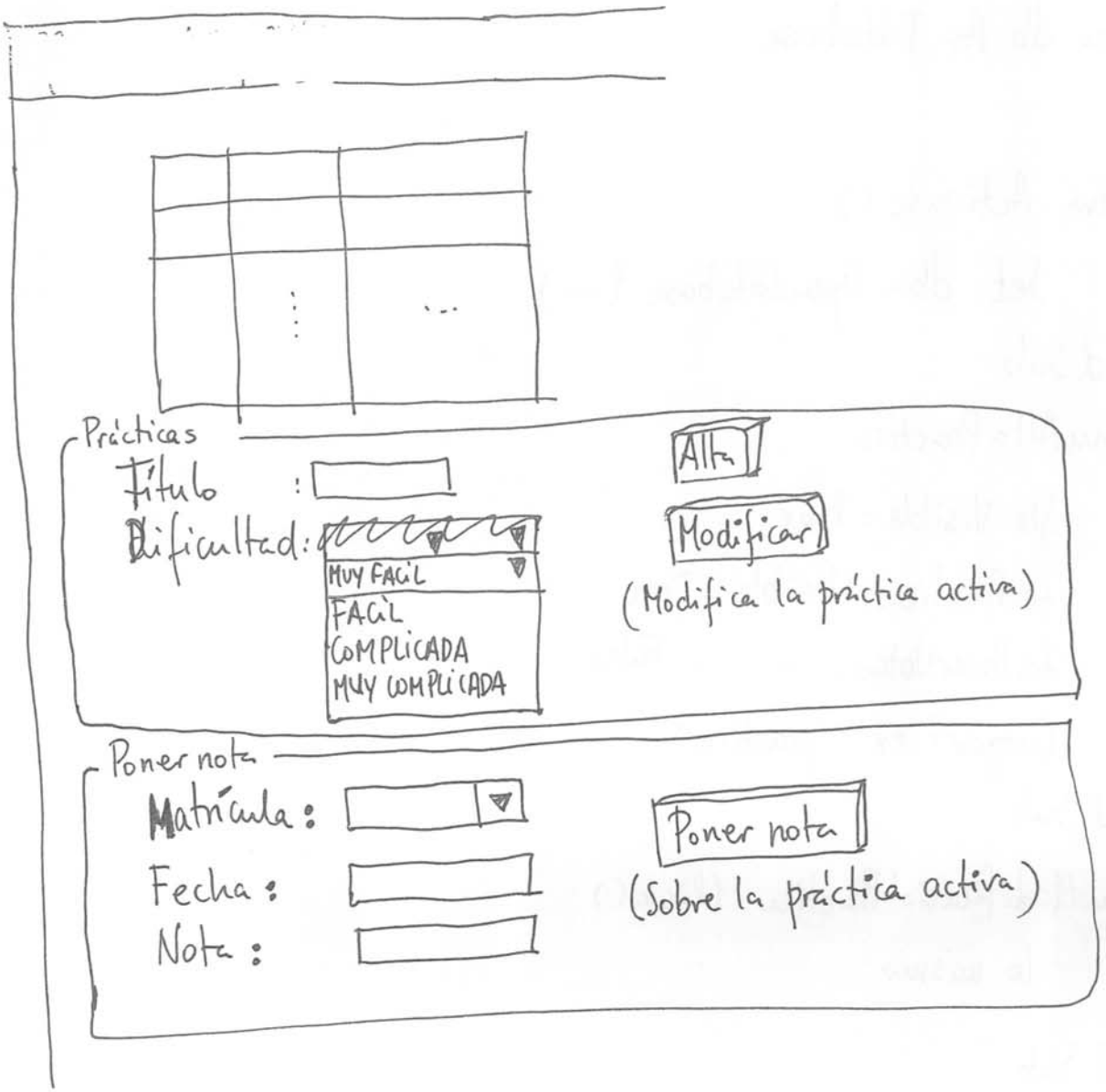
2

Las matrículas se considerarán validadas y cada alumno tiene una única matrícula. Se pide realizar las soluciones necesarias para resolver los problemas planteados a continuación utilizando un lenguaje de programación Visual, indicando los elementos que van a intervenir en cada solución, objetos, controles, propiedades, métodos, procedimientos y funciones creados por el opositor, tipos y estructuras de datos, etc. *La base de datos es una base de datos local, instalada en un directorio de una única máquina:*

1. Realice altas y modificaciones de la tabla Practica.
   - Implemente el alta del resultado de una práctica realizada por un alumno en concreto.
2. Algunos profesores a la hora de poner las notas de exámenes de algunos alumnos optan por crear un fichero de texto de acceso secuencial que se llama "ExamenesRealizadosPorAlumnos.txt". La información asociada será:
   - Matricula, nombre del alumno, apellidos del alumno, grupo al que pertenece el alumno, número del examen, la fecha y la nota. Esta información estará separada por dos puntos (:).
   - Ejemplo: 1243:Minombre:Miapellido1 Miapellido2:AD:12:12/06/2002:5
   - Este fichero se vuelca a la base de datos en la tabla o tablas correspondientes
   - Si el alumno no existe se dará de alta y se sigue con el proceso.
   - Si no existe el examen se generará un mensaje de error y se detiene el proceso
   - Una vez terminado el proceso se borrará el fichero "ExamenesRealizadosPorAlumnos.txt".
3. Muestre el listado de los exámenes de un alumno en concreto pidiendo al usuario que introduzca el nombre y apellidos del alumno.

NOTA: con respecto a los diseños tanto de las pantallas como de los listados, hay que indicar los elementos que intervienen. Relacionando los controles utilizados con sus propiedades correspondientes. Tener también en cuenta el control de errores y la validación de datos.

3



Puede hacerse este form sin el flex y sin el combo, pidiendo los datos en textbox o en InputBox. Es + rápido.

La validación de datos se puede hacer poniendo MaskedEdit en vez de TextBox

```
Dim db As Database

Form_Activate()
    Set db = OpenDatabase(...)
End Sub
mnuAltaPractica
    flx.Visible = true
    fraPracticas.Visible = true
    fraPonerNotas.  "    = False
    CargarFlex "practica"
End Sub
mnuModificaciónPractica_Click()
    lo mismo
End Sub

mnuPonerNotaPractice
    flx.Visible = False
    fraPracticas.Visible = False
    fraPonerNotas.  "    = True
    CargarComboMatriculas
End Sub
```

```
cmdAltaPractica_Click()

    Dim rst As Recordset
    Set rst = db.OpenRecordset("practica", ...)
    rst.AddNew
    rst.Fields("titulo").Value = txtTitulo.Text
    rst.Fields("dificultad") = cmbDificultad.Text
    rst.Update
    rst.Close
End Sub
```

No deben dar error. Si se quiere controlar, On Error...

```
cmdModificacionPractica_Click()

    Dim rst As Recordset
    Set rst = db.OpenRecordset("SELECT *
      FROM practica WHERE Npractica = "&
                        flx.TextMatrix(flx.Row, 0), ...)

    rst.Edit
    rst.Fields("titulo").Value = txtTitulo.Text
    rst.Fields("dificultad") = cmbDificultad.text
    rst.Update
    rst.Close

End Sub
```

```
cmdPonerNota_Click()
    On Error Goto ErrorPonerNota
    If Not IsNumeric(txtNota.text) Or CDec(txtNota.Text)<0 _
       Or CDec(txtNota.Text) > 10 then
          MsgBox "Error al escribir la nota"
    Elself Not IsDate(txtFecha.text) then
          MsgBox "Error en la fecha"
    Else
       db.Execute "INSERT INTO realiza VALUES ('" &
          cmbMatricula.text & "', " &
          Clng(flx.textMatrix(flx.Row, 0)) &
          ", #" & txtFecha.text & "#, " &
          CDec(txtNota.text) & ")"
    End If
    On Error Goto 0
    Exit Sub
ErrorPonerNota:
    MsgBox "Error al poner la nota"
End Sub

Public Sub CargarCombo()
'Carga el combo de las matriculas de la tabla alumnos
```

frmPpal:

```
mnuDesdeArchivo_Click()
    Dim rst      As Recordset

    Dim nFich As Integer
    Dim linea As String                    On Error Goto ErrorLectura
    .. campos() " "
    Dim error As Boolean
    Me.MousePointer = vbHourGlass
    nFich = FreeFile
    Open "ExamenesRealizadosPorAlumnos.txt" For Input As #nFich


    error = False
    While Not EOF(nFich)   And  Not error
        LineInput #nFich, linea
        campos = Split(linea,      ":")
        Set rst = db.OpenRecordset("alumno",...)
        rst      .FindFirst "matricula = '" & campos(0) & "'"
        If rst      .NoMatch then   ' ∄ alumno
           rst      .   .AddNew
              "      .Fields("matricula").Value = campos(0)
                  "          ("nombre").Value =    .. 1
                  ..         ("apellidos").  ..  =    .. 2
                  ..         ("grupo").  "  =    .. 3
           rstAlumnos.Update
        End If
        rst.    Close
```

```
        Set rst = db.OpenRecordset ("examen", ...)
        rst.FindFirst "numero = " & CLng (campos (4))
        If rst.NoMatch then
            error = true
        Else 'Alta
            If IsNumeric (campos(6)) And
               CDec(campos(6)) >= 0 And
               <= 10 then
            db.Execute " INSERT INTO hace VALUES ("&
                CLng (campos (4)) & ", '" & campos (0) & "', " &
                CDec (campos (6)) & ")"
            'la fecha no se guarda, no hay sitio en la
            'tabla para ella
            Else
                MsgBox "Error en la nota"
        EndIf
        rst.Close

    Loop
    If error then
        Msgbox "Examen no existente"
    EndIf
    kill "ExamenesRealizadosPorAlumnos.txt"
    Me.MousePointer = vbDefault
    Close (nFich)
    On    Error Goto 0
    Exit Sub
ErrorLectura:
    MsgBox "Error al leer las notas"
    Resume Next
End Sub
```

```
mnuListado_Click()
    Dim nombre As String
     " apellidos " "
     " rst       " Recordset
    nombre = InputBox ("Introduzca el nombre del alumno")
    apellidos =     "                "  los apellidos "   "
    Set rst = db.OpenRecordset ("alumno", ...)
    rst.FindFirst "nombre LIKE '" & nombre & "*' AND
       apellidos LIKE '" & apellidos & "*'"
    If rst.NoMatch then
        MsgBox "Alumno no existe"
    Else
        CargarFlex "SELECT * FROM hace WHERE matricula = '" &
        rst.Fields("matricula").Value
    EndIf
    rst.Close

End Sub
```