



Ejercicio Ficheros Binarios

- ✓ Realizar un programa que permita el uso y la gestión de una agenda de contactos teniendo en cuenta los siguientes aspectos:
 - Los datos de la agenda se almacenarán en un fichero binario
 - Cada entrada de la agenda guardará un nombre y un teléfono
 - El programa deberá mostrar un menú con las opciones siguientes análogo al que se muestra a continuación:



```
C:\WINDOWS\system32\cmd.exe - turbo
Agenda de telefonos....
1.- Insertar un contacto
2.- Mostrar todos los contactos de la agenda
3.- Buscar un telefono
4.- Ver numero de contactos en la agenda
5.- Actualizacion del numero de telefono de un contacto
6.- Salir
```

74



Ejercicio Ficheros Binarios

- ✓ Se definirán las siguientes funciones y procedimientos:
 - Función que visualiza el menú del programa
 - Función que visualiza y devuelve el número de contactos de la agenda
 - Procedimiento que muestra todos los contactos de la agenda
 - Procedimiento que inserta los datos de un nuevo contacto en la agenda al final del fichero binario
 - Función que busca y devuelve el teléfono asociado a un nombre en la agenda
 - Función que comprueba si existe o no un nombre en la agenda
 - Procedimiento que actualiza el teléfono de un contacto (si existe en la agenda)



75



Ejercicio Ficheros Binarios - Solución

```
PROGRAM AgendaFBinario;
CONST
    maxCaracNombre = 50;
    NombreAgenda = 'agenda.bin';
TYPE
    tNombre = String[maxCaracNombre];
    tAmigo = RECORD
        nombre: tNombre;
        telefono: longint;
    END;
    tAgenda = FILE OF tAmigo;
VAR
    fAgenda: tAgenda;
    opc: integer;
    tel: longint;
    nombre: tNombre;
    repetido: boolean;
...
```

MTP2006-2007

76



Ejercicio Ficheros Binarios - Solución

```
BEGIN {Programa principal}
    ASSIGN (fAgenda, NombreAgenda);
    REPEAT
        opc := Menu;
        CASE opc OF
            1: BEGIN
                writeln('Introduzca el nombre');
                readln(nombre);
                writeln('Introduzca el telefono');
                readln(tel);
                repetido := buscaNombre(fAgenda, nombre);
                IF (repetido) THEN writeln('Contacto repetido')
                ELSE insertarNuevoRegistro(fAgenda, nombre, tel);
            END;
            2: imprimeAgenda(fAgenda);
            3: BEGIN
                writeln('Introduzca el nombre');
                readln(nombre);
                tel := buscaTelefono(fAgenda, nombre);
                IF (tel <> -1) THEN writeln(nombre, ' ', tel)
                ELSE writeln('No encuentra en la agenda');
            END;
        END;
    ...
```

MTP2006-2007



Ejercicio Ficheros Binarios - Solución

```
... {Programa principal}
    4: devuelveNumeroContactos(fAgenda);
    5: BEGIN
        writeln('Introduzca el nombre del contacto');
        readln(nombre);
        writeln('Introduzca el nuevo numero de tel');
        readln(tel);
        actualizaTelefono(fAgenda, nombre, tel);
    END;
    END;
    UNTIL opc = 6;
END.
```

MTP2006-2007

78



Ejercicio Ficheros Binarios - Solución

```
FUNCTION Menu:integer;
VAR
    opcion: integer;
BEGIN
    writeln('*****');
    writeln('Agenda de telefonos.....');
    writeln('*****');
    writeln('1.- Insertar un contacto');
    writeln('2.- Mostrar todos los contactos de la agenda');
    writeln('3.- Buscar un telefono');
    writeln('4.- Ver numero de contactos en la agenda');
    writeln('5.- Actualizacion del numero de telefono de un contacto');
    writeln('6.- Salir');
    readln(opcion);
    Menu:=opcion;
END;
```

MTP2006-2007

79



Ejercicio Ficheros Binarios - Solución

```
FUNCTION devuelveNumeroContactos (VAR f: tAgenda): integer;
VAR
    num: integer;
BEGIN
    num := 0;
    {$I-}
    RESET(f);
    {$I+}
    IF IORESULT = 0 THEN
    BEGIN
        num := FILESIZE(f);
        writeln('Numero de contactos en la agenda: ', num);
        CLOSE(f);
    END;
    devuelveNumeroContactos := num;
END;
```

MTP2006-2007

80



Ejercicio Ficheros Binarios - Solución

```
PROCEDURE imprimeAgenda (VAR f: tAgenda);
VAR
    r: tAmigo;
    num: integer;
BEGIN
    num := devuelveNumeroContactos(f);
    IF num <> 0 THEN BEGIN
        RESET(f);
        WHILE NOT EOF(f) DO
        BEGIN
            read(f, r);
            writeln('Nombre: ', r.nombre, ' Telefono: ', r.telefono);
        END;
        CLOSE(f);
    END
    ELSE
        writeln('Todavía no hay contactos en su agenda');
END;
```

MTP2006-2007

81



Ejercicio Ficheros Binarios - Solución

```
PROCEDURE insertarNuevoRegistro(VAR f: tAgenda; n:tNombre; t:longint);
VAR
    r,aux: tAmigo;
    pos: longint;
BEGIN
    r.nombre := n;
    r.telefono := t;
    {Se intenta abrir el fichero para escribir al final el registro}
    {Si no existe el fichero se crea y se inserta el primer registro}
    {$I-}
    RESET(f);
    {$I+}
    IF IOResult = 0 THEN
        SEEK(f, FILESIZE(f))
    ELSE REWRITE(f);
    write(f,r);
    CLOSE(f);
END;
```

MTP2006-2007

82



Ejercicio Ficheros Binarios - Solución

```
FUNCTION buscaTelefono(VAR f: tAgenda; nom: tNombre): integer;
VAR
    r: tAmigo;          encontrado: boolean;          t: integer;
BEGIN
    t:= -1;
    encontrado := false;
    {$I-}
    RESET(f);
    {$I+}
    IF IORESULT = 0 THEN BEGIN
        WHILE NOT EOF(f) AND (encontrado = false) DO BEGIN
            read(f, r);
            IF (r.nombre = nom) THEN BEGIN
                encontrado:= true;
                t:= r.telefono;
            END;
        END;
    END;
    CLOSE (f);
    buscaTelefono:= t;
END;
```

MTP2006-2007

83



Ejercicio Ficheros Binarios - Solución

```
FUNCTION buscaNombre (VAR f: tAgenda; nom:tNombre): boolean;
VAR
    r: tAmigo;
    encontrado: boolean;
BEGIN
    encontrado := false;
    {$I-}
    RESET(f);
    {$I+}
    IF IORESULT = 0 THEN
    BEGIN
        WHILE NOT EOF(f) AND (encontrado = false) DO
        BEGIN
            read(f, r);
            IF (r.nombre = nom) THEN encontrado:= true;
        END;
        CLOSE(f);
    END;
    buscaNombre:= encontrado;
END;
```

MTP2006-2007

84



Ejercicio Ficheros Binarios - Solución

```
PROCEDURE actualizaTelefono (VAR f: tAgenda; nom: tNombre; tel: longint);
VAR
    encontrado: boolean;    r: tAmigo;
BEGIN
    encontrado := false;
    {$I-}
    RESET(f);
    {$I+}
    IF IORESULT = 0 THEN BEGIN
        WHILE NOT EOF(f) AND (encontrado = false) DO BEGIN
            read(f, r);
            IF (r.nombre = nom) THEN encontrado:= true;
        END;
        IF encontrado THEN BEGIN
            r.nombre := nom;    r.telefono := tel;
            SEEK(f, FILEPOS(f)-1); write(f, r);
        END
    ELSE writeln(nom, ' no existe en la agenda');
    CLOSE(f);
END;
```

MTP2006-2007

85