

# ALGORITMOS Y ESTRUCTURA DE DATOS



## Trabajo práctico, archivo binario

### Enunciado

Se dispone de un archivo binario cuya estructura se describe más abajo. Se requiere desarrollar dos programas:

1. **Mostrar**, que recorre el archivo mostrando su contenido por consola. Además, este programa debe actualizar la fecha de último acceso (ver estructura) con la fecha actual del sistema.
2. **Crear**, que el usuario podrá utilizar para generar un archivo de este tipo con la información que él desee almacenar.

Ambos programas recibirán como argumento en línea de comandos el nombre del archivo con el que van a trabajar.

Por ejemplo: sea el archivo DEMO.dat; entonces, al ejecutar el programa **Mostrar** veremos la siguiente salida por pantalla:

```
C:\>Mostar.exe DEMO.dat [ENTER]
```

```
----[CONTENIDO DEL ARCHIVO]-----
Nro. de Serie: 43112
Full Filename: C:/DEMO.dat
Fecha Modificacion: 2014/9/20
Cantidad de Campos Customizados: 4
Campo [codigo: 1, descripcion: Nombre]
Campo [codigo: 2, descripcion: Telefono]
Campo [codigo: 3, descripcion: Direccion]
Campo [codigo: 4, descripcion: EMail]
Cantidad de Registros: 5
-----
Nombre: Rogelio Roldan
Telefono: 4532-2411
Direccion: Rulo 126
EMail: roge@roldan.com.ar
-----
Nombre: Mongo Piccio
Telefono: 4223-2345
EMail: mongo@piccio.com.uy
-----
Nombre: Donald Patto
EMail: pattodonald@gmail.com
-----
Nombre: Roviralta Cenizza
```

```
Telefono: 5423-7732
Direccion: Pje. Lulupo 4645
EMail: cenicero@rovi.com
-----
Nombre: Momo Lulubell
Telefono: 4212-6623
Direccion: Formosa 4112
EMail: momo@saynomore.com
----[FIN CONTENIDO DEL ARCHIVO]-----
```

**NOTA:** Los valores pueden diferir según cual sea el archivo que se provea junto con este enunciado.

#### Estructura interna del archivo

La información que contiene el archivo binario está almacenada como se indica en la tabla que se muestra más abajo.

Esto es: los primeros 2 bytes describen el número de serie del archivo; los siguientes  $n$  bytes representan una cadena de caracteres que contiene el nombre completo del archivo, incluyendo la ruta. Los próximos 2 bytes representan la fecha de último acceso, etcétera.

Bytes	Tipo de Dato	Descripción
2	Integer	Número de serie del archivo.
$n$	CString	Nombre completo del archivo, incluyendo el <i>path</i> .
2	Date	Fecha de último acceso.
2	Integer	Cantidad de campos que tienen los registros.
$m^*$ (1)	RegType	Definición de campo.
2	Integer	Cantidad de registros que contiene el archivo.
$t^*$ (2)	RegData	Registro que contiene datos concretos.

(1) Se repite según la cantidad de campos que tienen los registros.

(2) Se repite según la cantidad de registros que tiene el archivo.

## Tipos de dato

A continuación analizaremos la estructura interna de cada uno de los tipos de dato antes mencionados.

### Integer

Este tipo representa 2 bytes que contienen un valor numérico entero positivo.

### RegType

Los datos de este tipo respetan la estructura que describe la siguiente tabla.

Bytes	Tipo de Dato	Descripción
2	Integer	Código del campo.
$n$	CString	Descripción del campo.

### RegData

Los datos de este tipo respetan la estructura que se describe a continuación.

Bytes	Tipo de Dato	Descripción
2	Integer	Cantidad de campos completos.
$(2+n)* (1)$	Integer	Código del campo.
	CString	Contenido del campo.

(1) Se repite tantas veces como indique la cantidad de campos completos.

### CString

Representa una cadena de caracteres cuya estructura interna dependerá de su longitud. Sea  $L$  la longitud de la cadena, entonces:

Si  $0 < L < 255$  entonces:

Bytes	Tipo de Dato	Descripción
1	Integer	Longitud de la cadena.
$n*(1)$	Character	Carácter.

(1) Son los caracteres que componen la cadena. Se repite tantas veces como indique su longitud.

Si  $L=0$  entonces:

Bytes	Tipo de Dato	Descripción
1	Flag	Valor: 0x00 para indicar que la cadena es vacía.

Si  $L \geq 255$  entonces:

Bytes	Tipo de Dato	Descripción
1	Flag	Valor: 0xFF que indica que la cadena es larga.
2	Integer	Longitud de la cadena.
$n*(1)$	Character	Carácter.

(1) Son los caracteres que componen la cadena. Se repite tantas veces como indique su longitud.

## Character y Flag

Estos tipos representan un byte; en esencia son lo mismo. La única diferencia es que el valor que contiene un **Character** corresponderá a un código ASCII. En cambio el valor que contiene un **Flag** estará definido previamente.

## Date

Este tipo de datos representa una fecha almacenada en 2 bytes.

1er. byte								2do. byte							
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Año							Mes				Día				
$2^7-1 = 127$							$2^4-1 = 15$				$2^5-1 = 32$				

Los primeros 7 bits, comenzando desde la izquierda, representan el valor del año. Los próximos 4 bits indican el mes y los últimos 5 bits marcan el día.

Sea  $a$  el valor del año contenido en una fecha; debe interpretarse del siguiente modo:

- si  $a < 100$ ; representa al año  $2000+a$ .
- si  $a \geq 100$ ; representa al año  $1999-a+100$ .

Por ejemplo:

Si  $a = 125$  entonces representa al año:  $1999-a+100 = 1974$ .

Si  $a = 4$  entonces representa al año  $2000-a = 2004$ .