

Funciones para manipular fechas

Escribe unos ficheros que contenga la definición de una estructura para almacenar fechas y las funciones apropiadas para manipular esta estructura.

```
//fecha.h

//Manipulacion de fechas
////////////////////////////////////

#include <fstream.h>

//Definicion de tipo
////////////////////////////////////
struct fecha
{
    unsigned int dia;
    unsigned int mes;
    unsigned int anyo;
};

//Operaciones
////////////////////////////////////

//Crear a partir de 3 enteros
fecha CrearFecha(unsigned int, unsigned int, unsigned int);

//entrada y salida
//dispositivos estandar cin y cout

void leerFecha (fecha&);
void escribirFecha(fecha);

//Archivos
bool fleerFecha (ifstream&, fecha&);
bool fescribirFecha (ofstream&, fecha);

//Operaciones Relacionales
bool igualFecha(fecha, fecha);
bool menorFecha(fecha, fecha);
bool mayorFecha(fecha, fecha);
```

```
//fecha.cpp
//mplementacion de las operaciones de "fecha.h"

#include <fstream.h>
#include <iostream.h>
#include "fecha.h"

//crear a partir de 3 enteros

fecha crearFecha (unsigned int d, unsigned int m, unsigned
int a)
{
    bool error = false;
    fecha f;

    //Comprobar que la fecha es válida

    //comprobar valores inferiores
    if (( d < 0 ) || ( m < 0) || ( a < 0) ) error = true;
    else //valores demasiado grandes
        if ( ( d > 31) || ( m > 12) ) error = true;
        else // ok
            if ( ( m == 4) || ( m == 6) || (m == 9) || (m
== 11) )
                //meses de 30 dias
                if ( m == 2) //fecbrero
                    if ( ( a % 4 ) == 0) //bisiesto
                        if( d <= 29 ) error = false;
                        else error = true;
                    else // normal, 28 dias
                        if ( d <= 28 ) error = false;
                        else error = true;
                else // meses de 31 dias
                    error = false;
    if ( error == true)
        cerr << "ERROR: fecha incorrecta " << endl;
    else
    {
        f.dia = d;
        f.mes = m;
        f.anyo =a;
    }
    return f;
}
```

```
//entrada y salida
void leerFecha( fecha& f)
{
    unsigned int d, m, a;

    cin >> d >> m >> a;
    f=crearFecha(d,m,a);

    return;
}

void escribirFecha(fecha f)
{
    cout << f.dia << '/' << f.mes << '/' << f.anyo;
}

bool fleerFecha (ifstream& ent, fecha& f)
{
    bool error = false;

    unsigned int d, m, a;

    if (! ent) error = true; // archivo no existente
    else
    {
        if ( ent >> d)
            if ( ent >> m)
                if ( ent >> a) error = false;
                else error = true; //fin fichero
            else error = true; // fin fichero
        else error = true; // fin fichero

        if ( error == false ) f = crearFecha(d, m, a);
    }
    return error;
}

bool fescribirFecha( ofstream& sal, fecha f)
{
    bool error = false;

    if ( ! sal ) error = true;
    else sal << f.dia << '/' << f.mes << '/' << f.anyo;

    return error;
}
```

```
//operaciones relacionales

bool igualFecha ( fecha f1, fecha f2)
{
    bool iguales;

    if (f1.dia != f2.dia) iguales = false;
    else
        if ( f1.mes != f2.mes ) iguales = false;
        else
            if (f1.anyo !=f2.anyo ) iguales = false;
            else iguales =true;
    return iguales;
}

bool menorFecha (fecha f1, fecha f2)
{
    bool menor;

    if (f1.anyo < f2.anyo ) menor = true;
    else
        if ( f1.anyo > f2.anyo) menor = false;
        else //mismo anyo
            if (f1.mes < f2.mes ) menor = true;
            else
                if (f1.mes > f2.mes) menor = false;
                else //mismo mes
                    if ( f1.dia < f2.dia) menor = true;
                    else menor = false;

    return menor;
}

bool mayorFecha(fecha f1, fecha f2)
{
    bool mayor;

    mayor = ! ( igualFecha(f1, f2) || menorFecha(f1, f2)
);
    return mayor;
}
```