

Programación 2

Examen de teoría (junio 2011)

31 de mayo de 2011



Instrucciones

- **Duración: 3 horas**
- El fichero del primer ejercicio debe llamarse `ej1.cc`. Para el segundo problema es necesario entregar cuatro ficheros, llamados `ciudad.cc`, `ciudad.h`, `partido.cc`, `partido.h`. Pon tu DNI y tu nombre en un comentario al principio de los ficheros fuente.
- La entrega se realizará a través del servidor de prácticas del DLSI (<http://pracdlsi.dlsi.ua.es>), del mismo modo que si fuera una práctica. Puedes realizar varias entregas, aunque sólo se corregirá la última.

Problemas

1. (6 puntos)

Queremos gestionar pedidos de piezas de los transbordadores espaciales de la NASA. Para ello, implementaremos un programa que recibirá tres parámetros¹: un nombre de un fichero de texto, el nombre de un transbordador y una fecha. A continuación se muestra un ejemplo de sintaxis correcta:

```
./ej1 datos.txt Atlantis 10/2/2012
```

El fichero de texto contiene la siguiente información por cada línea²:

```
transbordador:pieza:fecha
```

Por ejemplo, el fichero podría tener lo siguiente:

```
Endeavour:junta de la trocola:10/2/2015
Atlantis:bujia:05/6/2011
Atlantis:inyector motor:05/2/2013
Endeavour:tubo de escape:1/1/2012
Atlantis:asiento de eyeccion:05/2/2012
```

Se trata de imprimir por pantalla los nombres de las piezas que estarán disponibles en una determinada fecha para el transbordador pedido, es decir, aquellas cuya fecha sea menor que la introducida por parámetro. Con el ejemplo de llamada del principio y el fichero anterior, el programa debería imprimir:

```
bujia
asiento de eyeccion
```

Nota: Este problema debe implementarse usando programación procedural (no orientada a objetos).

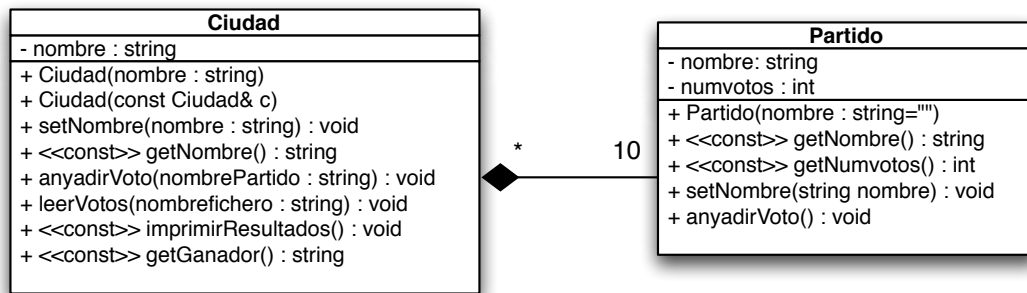
2. (4 puntos)

Queremos hacer un programa para gestionar los resultados electorales en una ciudad. Para ello, partimos del siguiente diagrama de clases³:

¹Hay que comprobar que el usuario introduce tres datos, y en caso contrario mostrar un mensaje de error. Se supondrá que el orden de los parámetros será correcto, así como el formato de la fecha.

²Se supondrá que la sintaxis del fichero será siempre correcta.

³A pesar de que la relación de composición debería implementarse con un vector de punteros, se hará mediante un vector de objetos, como en la práctica 3.



A las elecciones de una ciudad concurren varios partidos. Dado el siguiente ejemplo de fichero `elecciones.txt`, que contiene una linea por voto con el nombre del partido:

```

MOTA
EMC
PACC
DDA
EMC
MOTA
PACC
MOTA
  
```

... y el siguiente fichero `main.cc`:

```

#include "ciudad.h"
int main()
{
    Ciudad c("Villa arriba");
    c.leerVotos("elecciones.txt");
    c.imprimirResultados();
    cout << "Ganador: " << c.getGanador() << endl;
}
  
```

...el programa debería imprimir por pantalla:

```

Villa arriba
-----
MOTA:3
EMC:2
PACC:2
DDA:1
Ganador: MOTA
  
```

No es necesario ordenar los partidos por número de votos. Antes de ejecutar el programa, no se conocen los partidos que se pueden presentar (los nombres se leerán directamente del fichero mediante el método `leerVotos`). Los métodos `anyadirVoto` deben sumar un voto al partido correspondiente.

Ayuda: Puedes usar el siguiente `makefile` para compilar el programa:

```

main : main.o ciudad.o partido.o
    g++ -o main main.o ciudad.o partido.o
main.o: main.cc partido.h ciudad.h
    g++ -c -g -Wall main.cc
ciudad.o: ciudad.cc ciudad.h partido.h
    g++ -c -g -Wall ciudad.cc
partido.o: partido.cc partido.h
    g++ -c -g -Wall partido.cc
  
```