



## Metodología de la Programación - Febrero 2003 - Parte I

Dpto. Ing. Electrónica, Sistemas Informáticos y Automática  
E.P.S. La Rábida (Universidad de Huelva)

### Sección informativa :

- El examen se compone de **dos partes**, siendo la presente la primera de ellas.
- La duración de la primera parte es de **1h y 45 m.**
- Debe entregar cada uno de los dos ejercicios en folios independientes.
- Cada folio llevará **Apellidos y Nombre, DNI y Grupo (Gestión/Sistemas).**
- Solo se podrán realizar consultas durante los primeros **20 minutos.**
- Se agotará la convocatoria transcurridos los primeros **20 minutos.**

### Ejercicio 1: (4 Puntos)

Realice en los apartados en los que se requiera código, un programa completo:

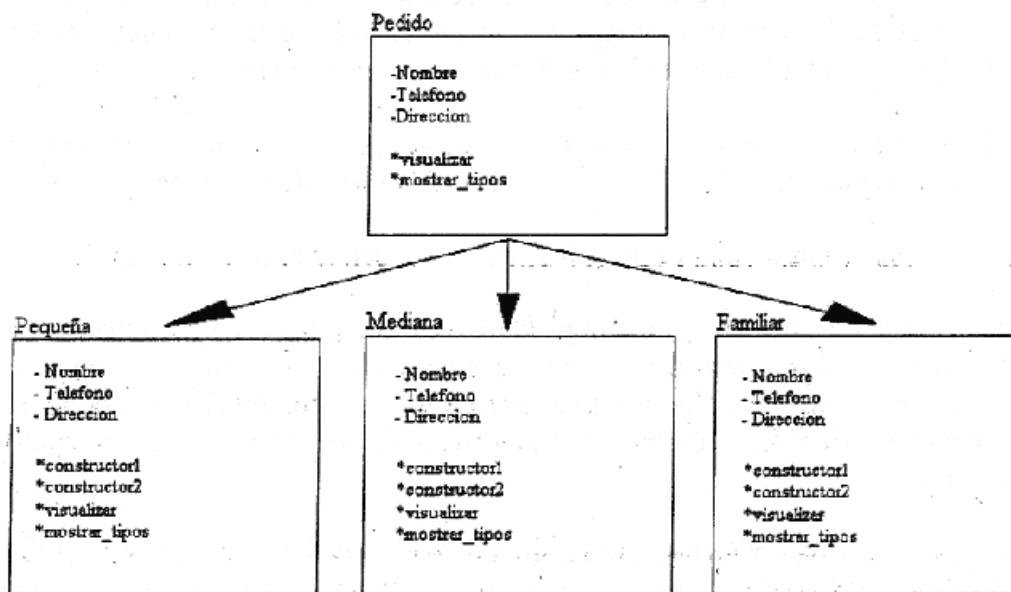
- a) En C estándar, declare en memoria estática una tabla bidimensional de 3x2 de cadenas de 25 caracteres y cargue con los valores de las cadenas introducidas por teclado. ¿Cuál es su orden de ejecución ? (0.75 Puntos).
- b) En C estándar, haga lo mismo que el apartado a) pero completamente en memoria dinámica. (No olvide liberar la memoria). (0.50 Puntos).
- c) Empleando instrucciones específicas de C++, realice la sección b). (0.25 Puntos).
- d) En C estándar, declare en memoria dinámica una tabla bidimensional de 3x2 de registros de dos campos: un campo cadena de 25 caracteres y el otro real de simple precisión. Realice la rutina de captura de datos por teclado como en los apartados anteriores. Libere la memoria antes de finalizar el programa. (1 Punto).
- e) Dado un fichero binario de reales de doble precisión ("data.bin"), genere un fichero de texto con la misma información ("data.txt") en el que se encuentre la misma información pero separando cada cifra con un espacio en blanco. (1 Punto).
- f) Razone en tres o cuatro renglones el motivo por el que un programa que implementa un algoritmo determinado es o no más rápido, ejecutado en un mismo ordenador, en Java que en C++. (0.5 Puntos).

## Ejercicio 2: (2 Puntos)

Se desea realizar un programa en Java para gestionar las peticiones telefónicas de pizzas de un restaurante. Las peticiones deben introducirse en una cola para ser atendidas por orden de llegada. Los datos que se toman por teléfono comunes a todas las peticiones son el nombre, la dirección y el teléfono. Además, cada pedido puede ser de un tamaño de pizza diferente: pequeña, mediana y familiar. El tamaño condiciona los tipos de los que puede ser, es decir, la pequeña sólo puede ser de tipo 1 y 2, la mediana de tipo 2, 3 y 4 y la familiar de tipo 4 y 5.

Se pide:

- Siguiendo el esquema de la figura, realice la interfaz de las cuatro clases y la implementación de sus dos constructores, sabiendo que queremos tener un constructor que inicializa todos los parámetros, y otro que trabaja parcialmente por defecto a unos valores predefinidos en cuanto al tipo (tipo\_1 para la pequeña, tipo\_3 para la mediana y tipo\_5 para la grande). (1 Punto).
- La interfaz de la clase cola, que es la encargada de atender los pedidos. (1 Punto).





# Metodología de la Programación - Febrero 2003 - Parte II

*Dpto. Ing. Electrónica, Sistemas Informáticos y Automática  
E.P.S. La Rábida (Universidad de Huelva)*

## Sección informativa :

- El examen se compone de dos partes, siendo la presente la segunda de ellas.
- La duración de la segunda parte es de 45 m.
- Cada folio llevará Apellidos y Nombre, DNI y Grupo (Gestión/Sistemas).

## Ejercicio 3: (4 Puntos)

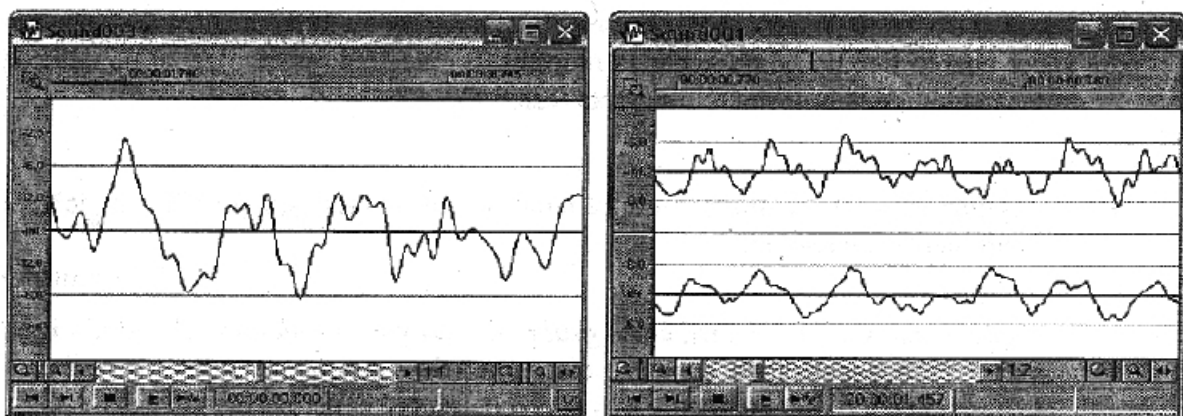
Se desea construir un programa que permita manipular sonidos. Se pretende que la aplicación admita, entre otras operaciones:

- Grabar un sonido.
- Reproducir un sonido.

Los sonidos pueden ser mono o estéreo y ambos comparten las siguientes propiedades:

- Número de muestras.
- Frecuencia de muestreo.
- Número de bits de resolución de una muestra.

Un sonido mono se diferencia de un sonido estéreo en que el primero sólo posee un canal mientras que el segundo posee dos canales, ver Figura 1.



**Figura 1. Sonido mono y estéreo respectivamente.**

Ejemplo numérico de un sonido mono :

	FM=44Khz .			BitsResolución=8				Número de muestras=10		
C	3	255	253	200	100	50	40	3	14	17

Ejemplo numérico de un sonido estéreo :

	FM=44Khz .			BitsResolución=8				Número de muestras=10		
CD	3	255	253	200	100	50	40	3	14	17
CI	4	5	32	45	67	50	56	7	3	1

Un canal se implementa mediante una tabla dinámica de números enteros sin signo que representan el valor de la intensidad o amplitud del sonido en un determinado instante de tiempo.

Se dispone de una librería en C denominada *music.h* que proporciona las siguientes funciones:

```
void reproducemuestramono (      unsigned int muestra,
                                unsigned int FM,
                                unsigned int BitsResol);
void grabamuestramono (      unsigned int *muestra, unsigned
                                int FM, unsigned int BitsResol);
void reproducemuestraestereo ( unsigned int muestracanalizquierdo,
                                unsigned int muestracanalizquierdo,
                                unsigned int FM,
                                unsigned int BitsResol);
void grabamuestraestereo (unsigned int *muestracanalizquierdo, unsigned
                                int *muestracanalizquierdo, unsigned int FM,
                                unsigned int BitsResol);
```

Para reproducir un sonido es necesario tomar cada muestra del canal y utilizar las diferentes funciones *reproducemuestra* (descritas en el cuatro anterior), pasándole como parámetro, además de la muestra, o muestras (estéreo), la frecuencia de muestreo para saber cuanto tiempo ha de sonar la muestra, o muestras (estéreo), y el número de bits de resolución de la muestra, o muestras (estéreo).

Se pide:

1. Diseñar en C++ la interfaz de la jerarquía de clases que permite modelar e implementar un sonido. **(2 puntos)**
2. Implementar en C++ los métodos *grabarsonido* y *reproducirsonido* de la forma más adecuada para cada subclase diseñada. (Se supone que todos los atributos se han rellenado correctamente tras la ejecución del constructor). **(2 puntos)**