



# Convocatoria extraordinaria de septiembre de 2006 FUNDAMENTOS DE LA PROGRAMACIÓN

Nombre, apellidos y DNI

---

## EJERCICIO 1 (3 puntos)

El número de Reynolds es un número adimensional utilizado en mecánica de fluidos que viene dado por la siguiente expresión:

$$Re = \frac{\rho v_s L}{\mu}$$

donde  $\rho$  es la densidad del fluido,  $v_s$  la velocidad característica del fluido,  $L$  la longitud característica del problema (p.ej. el diámetro de una tubería) y  $\mu$  la viscosidad del fluido.

Si el número de Reynolds es menor que 2100 el flujo es laminar. Si es mayor de 4000, el flujo es turbulento.

- Implemente una clase que permita representar las características de un problema de mecánica de fluidos e incluya un método que nos devuelva si el flujo de un fluido tiene carácter laminar o turbulento.
- Diseñe un conjunto completo de casos de prueba que nos sirva para comprobar el funcionamiento del método anterior. Construya una tabla de casos de prueba de la siguiente forma:

$\rho$	$v_s$	$L$	$\mu$	Resultado	Justificación
...	...	...	...	...	(el porqué de cada caso de prueba)

## EJERCICIO 2 (2 puntos)

Dada una clase `Vector` declarada de la siguiente forma:

```
public class Vector
{
    private int datos[];

    public int size ()
    {
        return datos.length;
    }

    public int rec (int j, int k)
    {
        int m = (j+k)/2;
        if (j<=k) {
            return (datos[m]%2)
                + rec(j, m-1)
                + rec(m+1, k);
        } else {
            return 0;
        }
    }
}
```



Convocatoria extraordinaria de septiembre de 2006  
**FUNDAMENTOS DE LA PROGRAMACIÓN**

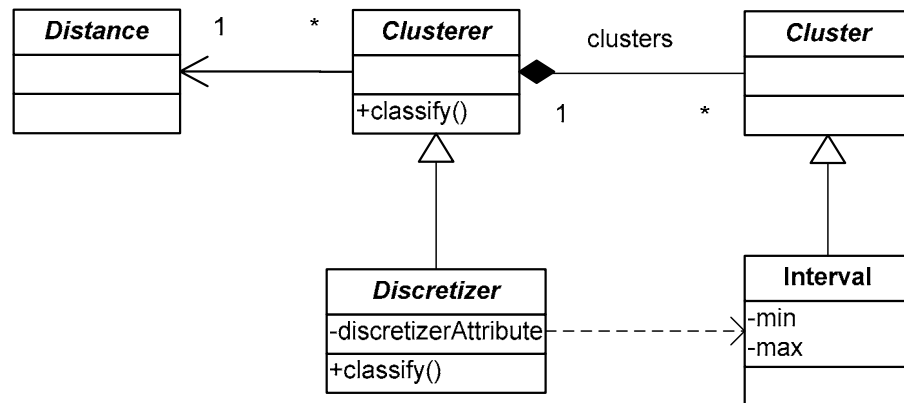
Nombre, apellidos y DNI

---

- a) ¿Qué función realiza la llamada al método `vector.rec(0, vector.size()-1)`? ¿Y la llamada a `vector.rec(a, b)` para dos valores cualesquiera a y b?
- b) Implemente un método equivalente sin utilizar recursividad.

**EJERCICIO 3** (2 puntos)

Declare e implemente adecuadamente las clases en Java que se derivan del siguiente diagrama de clases UML:



**EJERCICIO 4** (3 puntos)

Diseñe e implemente en Java un servidor TCP **multihebra** que sea capaz de recibir conexiones a través del puerto TCP número 21.

Cuando un cliente se conecta al servidor, le envía una línea de texto que contiene el nombre de un fichero al que desea acceder. Entonces, el servidor lee el fichero que haya solicitado el cliente y le devuelve el contenido del mismo a través de la conexión establecida con el cliente a través de un socket.

NOTA: Cuando el cliente intenta acceder a un fichero que no existe en el servidor, el servidor simplemente cierra la conexión sin enviarle nada al cliente.