

## Hoja de ejercicios 65\_65:

- Interfaces.

65.- Supongamos la siguiente *interface*:

```
public interface IAlumno
{
    /** * @return Devuelve el nombre del alumno */
    public String getNombre();

    /** Añade una nota al alumno
     * @param nota
     * @return true si se puede añadir una nota más, false en caso contrario, ya sea
     * porque
     * no caben más notas o porque la nota no es válida (nota >=0 y <=10). */
    public boolean anadeNota(int nota);

    /** @return array con las notas existentes del alumno */
    public int[] getNotas();

    /** @return número de notas que tiene el alumno */
    public int numNotas();

    /** @return nota media de las notas que tiene. Si no hay notas devuelve -1.*/
    public float getNotaMedia();

    /** @return número de notas >= 5 */
    public int getNumeroAprobados();

    /** Elimina las notas del alumno */
    public void borrarNotas();
}
```

**Crea una clase llamada *Alumno*** que implemente la interface *IAlumno*:

- Al constructor se le indicará el nombre del alumno. El alumno inicialmente no tendrá ninguna nota.
- El comentario predecesor de cada método en la interface indica su funcionamiento.
- El **número máximo de notas** que puede tener cada alumno es **10**.

**Crea un programa principal** que cargue desde teclado un array de 30 Alumnos, con sus nombres y sus notas, y posteriormente los liste. La entrada y salida por consola se realizará de la siguiente manera:

Introduzca nombre: Luis Introduzca notas separadas por espacios: 1 2 3 5 10 22 56 9 8 5 4 3 La nota 22 no se pudo añadir La nota 56 no se pudo añadir Introduzca nombre: Roberto Introduzca notas separadas por espacios: 5 6 Introduzca nombre: Alfonso Introduzca notas separadas por espacios: 7 8 9 10 11 10 10 5 6 5 7 9 La nota 11 no se pudo añadir La nota 9 no se pudo añadir	Listado de alumnos: Nombre Número Aprobados Media Luis 10 5 5.0 Roberto 2 2 5.5 Alfonso 10 10 7.7
---	---

- Las notas no válidas no se añadirán.
- Sólo se añadirán las 10 notas primeras válidas.