

# Examen de las Unidades 4b, 5 y 6

23 de febrero de 2015

---

Apellidos y nombre:

---

## Instrucciones:

- Cada ejercicio tiene una puntuación inicial indicada en el enunciado.
  - Los fallos cometidos en su realización restarán puntos según este baremo: fallos leves, entre el 5 y el 25% de la puntuación total; fallos graves, entre el 25 y el 50%; fallos muy graves, entre el 50 y el 100%.
  - Cualquier suposición adicional sobre los casos prácticos debe consignarse por escrito en la resolución del problema si se desea que sea tenida en cuenta en la corrección del mismo.
  - Escriba con tinta azul o negra. No se admitirán exámenes escritos a lápiz.
- 

## EJERCICIO 1 (2,5 PUNTOS)

Escribir un programa en Java que abra un archivo de texto y genere otro fichero idéntico, pero en el que **todas las mayúsculas se hayan cambiado por minúsculas y al revés**. Al terminar, debe informar del **número de caracteres** que se han modificado.

Los nombres de los dos ficheros deben pedirse por teclado.

Se recuerda que los códigos ASCII de las letras mayúsculas van del 65 ('A') al 90 ('Z'), y que las minúsculas van del 97 ('a') al 122 ('z'). No es necesario tener en cuenta las ñes.

Un ejemplo de ejecución del programa podría ser:

```
$ Introduzca el nombre del fichero de origen: prueba.txt
$ Introduzca el nombre del fichero de destino: destino.txt
Fichero destino.txt generado con éxito.
Se han modificado 2983 caracteres.
```

A continuación se muestran los métodos más habituales de las clases que pueden utilizarse para resolver el problema (no significa que haya que usarlos todos, por supuesto):

### Clase FileWriter:

- `FileWriter(String nombreFichero, boolean añadir)`: constructor.
- `write(char c)`: escribe el carácter c en el flujo, a continuación del último carácter escrito.
- `write(char[] c)`: escribe una cadena de caracteres.
- `write(String s)`: escribe el contenido de un String.
- `close()`: cierra el flujo.

### Clase FileReader:

- `FileReader(String nombreFichero)`: construye el objeto a partir del nombre del fichero.
- `read()`: lee un carácter del flujo.
- `read(char[] c)`: lee todos los caracteres disponibles en el flujo de entrada y los coloca en el array c. Devuelve el número de caracteres leídos, o -1 si no pudo leer nada.

### Clase FileOutputStream:

- `FileOutputStream(String nombreFichero, boolean añadir)`: Constructor.
- `write(byte b)`: escribe el byte b en el flujo.
- `close()`: cierra el flujo.

### Clase FileInputStream:

- `FileInputStream(String nombreFichero)`: abre un flujo de entrada desde el fichero indicado.
- `available()`: devuelve el número de bytes que restan por leer en lectura secuencial.
- `read()`: lee un byte del flujo. Apropiado para el acceso secuencial. Devuelve -1 si no hay datos en el flujo.
- `skip(int n)`: salta n bytes adelante en la lectura del flujo.
- `close()`: cierra el flujo.

## EJERCICIO 2 (2,5 PUNTOS)

Tenemos una **clase Damas** para el juego de las damas. El tablero del juego tiene 8x8 casillas (como el del ajedrez) que pueden estar ocupadas por fichas blancas o negras.

El tablero se ha modelado como una variable de instancia declarada así:

```
short[] tablero = new short[64];
```

Cada posición del array contiene un 0 (casilla vacía), un 1 (casilla ocupada por ficha blanca) o un 2 (casilla ocupada por una ficha negra).

Se pide escribir el código de dos métodos de la clase **Damas**:

a) **void mostrarTablero()**: este método imprimirá el contenido de la variable tablero en la pantalla de forma legible, más o menos así:

o	o	o	o	o	o	o	o
o	o	o			o		o
				o		o	
	x		o				
							x
			x	x			
x		x			x	x	
x	x	x	x	x	x	x	x

La esquina superior izquierda se corresponde con la posición 0 del array tablero, y la esquina inferior derecha es la posición 63. Una "O" simboliza una ficha blanca y una "X", una ficha negra.

b) **short getContenido(int fila, int columna)**: este método devolverá el contenido de la posición (fila, columna) del tablero, teniendo en cuenta que la posición (0,0) es la esquina superior izquierda y la (7,7) es la esquina inferior derecha.

Por ejemplo, en el tablero anterior, una llamada a `getContenido(0,3)` devolverá un 1, porque en esa posición hay una ficha blanca.

Obsérvese que este método puede ser usado para programar más fácilmente el anterior.

### EJERCICIO 3 (2,5 PUNTOS)

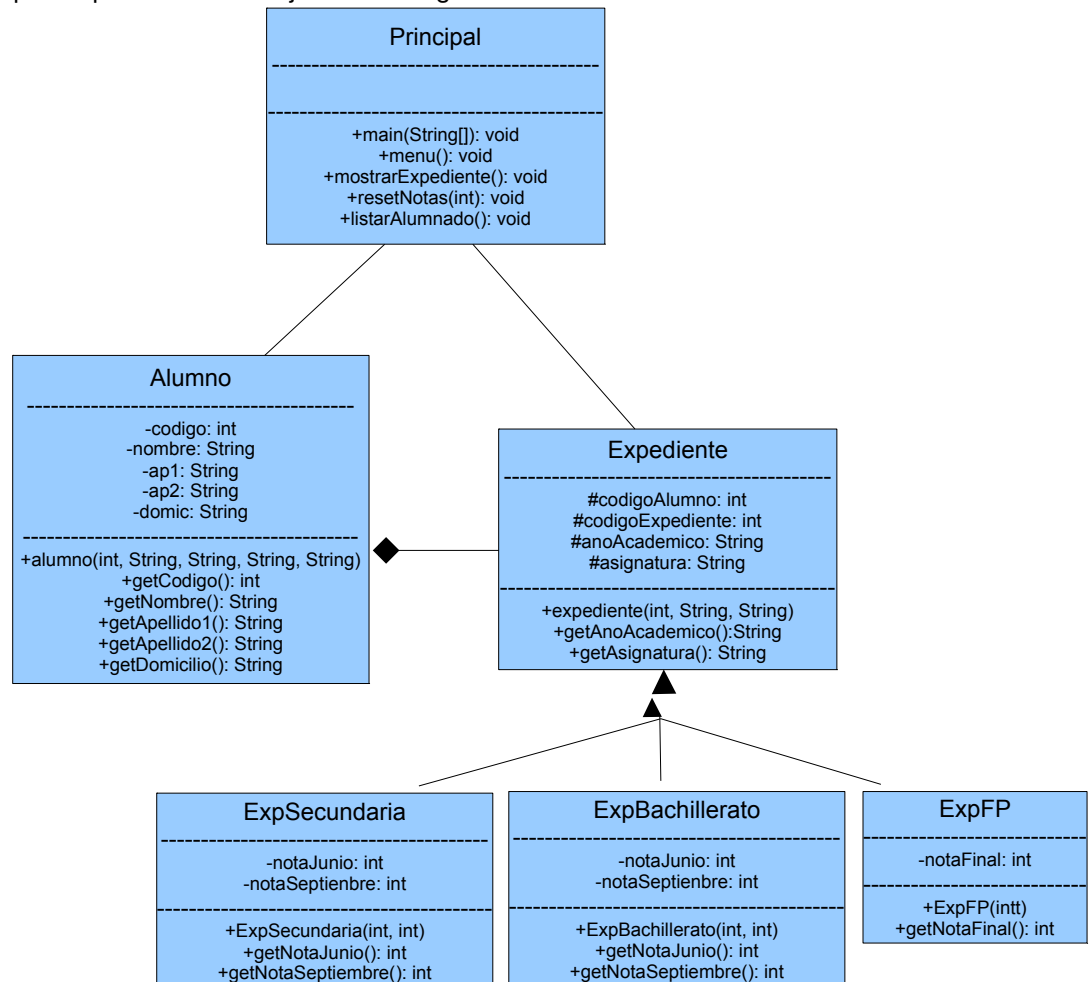
En un instituto utilizan dos ficheros para almacenar la información de los alumnos y su expediente académico:

A) Los alumnos son **objetos serializados** en un archivo llamado **alumnos.dat**. Los atributos de cada objeto se pueden ver en el diagrama de clases adjunto.

B) Los expedientes académicos son objetos serializados en el archivo **expedientes.dat**. Los atributos figuran en el diagrama de clases. Las calificaciones son diferentes según el alumno sea de secundaria, de bachillerato o de formación profesional.

Como es lógico, cada alumno puede tener varias entradas en el archivo de expedientes (una por cada asignatura y cada año)

El diagrama de clases al que responden estos objetos es el siguiente:



**Se pide** escribir el código del método `mostrarExpediente()` de la clase `Principal`. Este método debe **mostrar por la pantalla el expediente académico completo de un alumno** cuyo código introducirá el usuario por teclado.

Una ejecución del método podría resultar en algo así:

\$ Introduzca el código del alumno: 784

Datos del alumno 784:

- Nivel: ESO
- Nombre y apellidos: Juan Pérez Pérez
- Domicilio: C/ Rue del Percebe 13, 2º A
- Fecha de nacimiento: 31/02/1991 (continúa en pág. siguiente)

Asignatura	Curso	Nota Junio	Nota Septiembre
Matemáticas	14/15	6,7	0,0
Filosofía	14/15	7,1	0,0
Ed. Física	14/15	3,5	8,0
...etc...			

#### EJERCICIO 4 (2,5 PUNTOS)

Supongamos que hemos detectado **un error en el archivo expedientes.dat**. El error consiste en que alguien introdujo erróneamente las notas de un alumno que en realidad todavía no se ha examinado de nada y por lo tanto no puede tener ninguna nota asignada.

El ejercicio consiste en escribir el método `resetNotas(int)` de la clase `Principal`. Este método se encargará de subsanar ese error, es decir, **buscará las notas del expediente pasado como parámetro y las pondrá todas a 0,0**.