

1. Flujos de caracteres: (FileReader, FileWriter) (1,5 puntos)

Escribe un programa en Java que lea una línea de texto desde un archivo y devuelva la misma línea, pero con las palabras invertidas individualmente. Es decir, si el archivo contiene "Hola Mundo", el archivo de salida debería contener "aloH odnuM".

2. Flujos de caracteres: (BufferedReader, BufferedWriter) (1,5 puntos)

Crea un programa en Java que lea un archivo con nombres y apellidos separados por espacios, y escriba en un nuevo archivo solo los nombres que tienen exactamente cinco letras.

3. (Flujos binarios:InputStream) (1,5 puntos)

Realiza un programa en Java que lea la cabecera de un fichero PDF y verifique si realmente se trata de un archivo PDF válido. Para que un archivo sea un PDF válido, debe comenzar con la siguiente secuencia de bytes: {37, 80, 68, 70}. Si la cabecera no coincide con esta secuencia, el programa debe informar al usuario de que el archivo no es válido (1 punto)

4. (Acceso Aleatorio: RandomAccessFile)

Desde la editorial Marvel te han contratado para hacer una aplicación que gestione los datos de sus superhéroes y supervillanos. Para almacenar la información han decidido utilizar ficheros de distintas clases.

a). Realiza un programa en java para guardar datos de personajes en un fichero aleatorio, dale el nombre **Marvel.dat**. Introduce la información de los personajes a partir de los arrays que se te proporcionan en la plataforma Moodle. Cuando termine de realizar la carga de datos deberá informar al usuario de que la carga se ha realizado satisfactoriamente o no. Los datos por cada personaje son: (1,5 puntos)

| | |
|--------------------|----------------|
| Id: | Numero Entero. |
| DNI: | String [9]. |
| Nombre: | String[10]. |
| Identidad secreta: | String[20]. |
| Tipo: | String[10] |
| Peso: | Numero Entero. |
| Altura: | Numero Entero |

```
|La carga de los personajes ha terminado correctamente.
```

b). Desde la editorial quieren tener controlado el peso de sus personajes, ya que últimamente han hecho algún exceso que otro. Realiza un programa en java que te permita

modificar los datos de un personaje. El programa recibe desde la línea de comandos el dni y el peso del último mes. Si el personaje no existe devolverá un mensaje de error, sino mostrará en la consola el nombre del personaje y cuantos kilos ha engordado, adelgazado o si se mantiene en su peso. (1,5 puntos)

```
Introduzca el DNI (con letra) del personaje para control de peso:
01010101A
Introduzca su peso actual:
76
Spiderman se mantiene en su peso anterior.
```

```
Introduzca el DNI (con letra) del personaje para control de peso:
01010101A
Introduzca su peso actual:
80
Spiderman ha engordado 4 kilos.
```

```
Introduzca el DNI (con letra) del personaje para control de peso:
01010101A
Introduzca su peso actual:
74
Spiderman ha adelgazado 6 kilos.
```

c). Realiza un programa en Java que te permita visualizar los personajes de un tipo concreto (héroe o villano). El programa recibe desde la línea de comandos el tipo de personaje y visualiza cuantos personajes hay de dicho tipo y todos los datos de dichos personajes. Verifica que exista dicho tipo en el fichero, si no existe saca un mensaje de error por pantalla. (1,5 puntos)

Nota: Hay que pensar que el fichero puede crecer en un futuro y aparecer nuevos tipos

```
Introduce un tipo de personaje: heroe
Se han encontrado 4 heroes.
Personaje [dni=01010101A, nombre=Spiderman , identidad=Peter Parker , tipo=heroe , peso=74, altura=178]
Personaje [dni=04040404D, nombre=Iron Man , identidad=Tony Stark , tipo=heroe , peso=102, altura=182]
Personaje [dni=05050505E, nombre=Storm , identidad=Ororo Munroe , tipo=heroe , peso=66, altura=156]
Personaje [dni=07070707G, nombre=Wolverine , identidad=James Howlett , tipo=heroe , peso=136, altura=152]
```

```
Introduce un tipo de personaje: patrulla X
No existen patrulla Xs en el fichero.
```