

EJERCICIO USO DE FICHEROS.

1. Realizar un programa que a partir de un fichero llamado nota.txt que contenga un listado de notas de alumnos, realizará las lecturas de todas las notas y nos muestre la nota media.

En el fichero nota.txt las notas estarán separadas por la barra vertical ("|"). Las notas serán notas enteras.

Un ejemplo del contenido del fichero es el siguiente con 10 notas: 5|6|7|8|4|10|6|5|7|8

2. Aumentar la funcionalidad del ejercicio anterior se establecerá un menú con el siguiente formato:

1. Registrar notas. Solicitará al usuario cuantas notas va a introducir y recogerá todas las notas. Estas notas se registrarán en un fichero "notas.txt" utilizando el formato explicado en el ejercicio anterior utilizando como separador la barra vertical "|".

2. Obtener media. Llama al algoritmo desarrollado en el ejercicio anterior.

-1. Salir. El menú saldrá constantemente hasta que el usuario escoge la opción -1 en el menú principal para salir del programa.

3. Escribe una clase llamada "Fichero" que tenga los siguientes métodos estáticos:

- **obtenerFicheros(): String[]**. Método para obtener una lista de todos los ficheros dado la ruta de un directorio.
- **detectarRuta(String ruta): int**. Método que indica si una ruta dada existe o no y en caso de existir si un fichero o un directorio. Retornos
 - -1: No existe.
 - 1: Es un fichero.
 - 2: Es un directorio.
- **leerFichero(String ruta)**. Dado un fichero de texto plano, muestra por terminal contenido integro (línea por línea de dicho fichero).
- **contadorVocales(String ruta): int**. Dado un fichero de texto plano. Retorna el número de vocales detectadas dentro de todo el contenido del fichero.
- **contadorConsonantes(String ruta): int**. Dado un fichero de texto plano. Retorna el número de consonantes detectadas dentro de todo el contenido del fichero.
- **contadorMayusculas(String ruta): int**. Dado un fichero de texto plano. Retorna el número de caracteres mayúsculas detectados dentro de todo el contenido del fichero.

4.- Crea clase llamada "Personaje" con las siguientes características:

- **nombrePersonaje** : String.
- **vida**: int. Número entero entre 0 y 100.
- **herir(int cantidad)**. Disminuye la vida del personaje hasta un mínimo de 0.
- **curar(int cantidad)**. Aumenta la vida del personaje hasta un máximo de 100.

- **toString()** : String. Retorna el estado actual del personaje en un String.

Tras instanciar un objeto de la clase Personaje, guardamos el estado actual de inicio del personaje guardando el objeto en un fichero "personaje.dat".

Realiza una serie de acciones hiriendo o curando el personaje y tras mostrar su estado, reiniciamos el estado del objeto personaje leyendo del objeto registrado en el fichero "personaje.dat".