

Desafío - Escritura y lectura

En este desafío validaremos nuestros conocimientos vistos en las sesiones anteriores.

Lee todo el documento antes de comenzar el desarrollo individual o grupal, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

En el siguiente desafío debes crear un archivo y un fichero. Se necesita escribir y leer desde el archivo creado por el usuario. Para ello, se deberá validar la información de entrada.

Formato del archivo

El archivo tendrá múltiples líneas y en cada línea un texto aleatorio.

Cada elemento debe corresponder a un texto precargado dentro de ArrayList que se entregará en el requerimiento, esto debe implementarse para llenar el archivo.

```
Perro  
Gato  
Juan  
Daniel  
Juan  
Gato  
Perro  
Camila  
Daniel  
Camila
```

En el ejemplo anterior, el fichero está llenado con información extraída desde un ArrayList.

Requerimientos

1. Crear un método llamado `crearArchivo(directorio, archivo archivo)`, este método recibe el nombre del directorio y el ~~archivo~~ ^{archivo} como parámetros de entrada.
(0,5 Puntos)
2. El nombre del ~~archivo~~ ^{archivo} debe terminar con ".txt", para indicar que es un archivo de texto.
(0,25 Puntos)
3. Validar que el nombre del directorio no exista ~~dentro del programa~~.
(1,5 Puntos)
 - Si el directorio no existe, se debe crear.
 - Si existe se mostrará el siguiente mensaje por consola El directorio ya existe
 - Si existe algún otro problema al crear el directorio, se mostrará el siguiente mensaje por consola:

Error al crear directorio

4. Escribir en el archivo con un salto de línea lo que se encuentra en el siguiente ArrayList.
(1,5 Puntos)

Considerar utilizar Iterator para recorrer la lista.

```
ArrayList<String> lista = new ArrayList<String>();
    lista.add("Perro");
    lista.add("Gato");
    lista.add("Juan");
    lista.add("Daniel");
    lista.add("Juan");
    lista.add("Gato");
    lista.add("Perro");
    lista.add("Camila");
    lista.add("Daniel");
    lista.add("Camila");
```

5. Crear un método llamado `buscarTexto(nombreArchivo nombreArchivo, nombreFichero archivo, texto)`, este método recibe el nombre del ~~fichero~~ ^{archivo} y el texto a buscar como parámetros de entrada.
(1,75 Puntos)
6. Validar que el ~~fichero~~ ^{archivo} exista.
(1,5 Puntos)
 - Si el ~~fichero~~ ^{archivo} no existe, se mostrará el siguiente mensaje por consola.

El ~~fichero~~^{archivo} ingresado no existe

7. Si el ~~fichero~~^{archivo} existe, buscar el texto ingresado dentro del archivo.

(1,5

Puntos)

8. Mostrar por consola la cantidad de veces que la palabra ingresada se encuentra en el ~~texto~~^{archivo}.

(1,5 Puntos)

cantidad de repeticiones del texto -> 2



¡Mucho éxito!

Consideraciones y recomendaciones

Ejemplo de cómo debe quedar el directorio

