

**ACTIVIDAD Y TIPO: “Excepciones, Flujos y Ficheros”/ AEF**

**LUGAR DE REALIZACIÓN:** Centro docente/Casa

Unidad Didáctica	Criterios de Evaluación	Resultados de Aprendizaje
UD14	RA3.d) Se ha escrito código utilizando control de excepciones.	RA3
UD15	RA5.a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.	RA5
UD15	RA5.b) Se han aplicado formatos en la visualización de la información.	RA5
UD15	RA5.c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.	RA5
UD16	RA5.d) Se han utilizado ficheros para almacenar y recuperar información.	RA5
UD16	RA5.e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros	RA5

Objetivos Generales	Competencias Profesionales, Personales y Sociales
a) e) f) h) i) j) n) ñ) q) r) w)	a) e) g) h) i) j) n) ñ) q) r) w)

- **Individual/ Equipo:** Individual
- **Descripción de la actividad:** Realizar la actividad propuesta.
- **Fecha y medio de entrega:** La fecha y medio de entrega se indicará en la publicación de la tarea en Classroom. Si se incumplen las especificaciones indicadas en la actividad tanto de contenido como de fecha, formato, nombre o forma de entrega, la actividad será calificada con un 0.
- **Formato:** Se deberá entregar un archivo **Ape1Ape2Nom.zip** conteniendo:
  - un proyecto Netbeans de nombre **Ape1Ape2Nom** conteniendo la implementación de la actividad con la especificación indicada en la misma.
  -

### INSTRUMENTOS DE EVALUACIÓN:

La calificación de la competencia INICIATIVA supondrá el 7.5% de la calificación de cada CE.

CE	INSTRUMENTO DE EVALUACIÓN	% CE Actividad
RA3.d) Se ha escrito código utilizando control de excepciones.	Checklist	75% del 18%
RA5.a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.	Checklist	100% del 25%
RA5.b) Se han aplicado formatos en la visualización de la información.	Checklist	100% del 8%
RA5.c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.	Checklist	100% del 15%
RA5.d) Se han utilizado ficheros para almacenar y recuperar información.	Checklist	100% del 33%
RA5.e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros	Checklist	100% del 19%

---

**IMPORTANTE:** ya **NO** se puede usar la clases **Leer** para la entrada de datos.

En todos los programas habrá que garantizar que sea multiplataforma, garantizando la elección del carácter separador de ruta correcto.

**Ud1456Act1.** Realice un programa en Java de nombre *Micat.java* que funcione exactamente igual que el comando de Linux cat, esto es, el programa debe recibir en la línea de comandos como único argumento el nombre de un fichero, y posteriormente visualizar en pantalla el contenido del mismo.

**Ud1456Act2.** Realice un programa en Java de nombre *Mimore.java* que funcione exactamente igual que el comando de Linux more, esto es, el programa debe recibir en la línea de comandos como único argumento el nombre de un fichero, y posteriormente visualizar en pantalla el contenido del mismo página a página. (páginas de 10 líneas)

**Ud1456Act3.** Realice un programa en Java de nombre *LinCarPal.java* que dado el nombre de un fichero de texto introducido por teclado, muestre por pantalla el total de líneas, caracteres y palabras que contiene.

**Ud1456Act4.** Realice un programa en Java de nombre *Tam.java* que pida por teclado el nombre de un fichero y muestre la cantidad en bytes que ocupa, contando los bytes y usando la información del sistema de archivos.

**Ud1456Act5.** Realizar un programa en Java de nombre *PruebaTest* que permita realizar las correcciones de exámenes de tipo test de manera automática y presente los resultados en pantalla. El programa debe permitir cargar un fichero de resultados en memoria, cargar ficheros de respuestas y evaluarlos. Así pues, presentará una interfaz en pantalla con las siguientes opciones:

*CORRECCION DE TEST*

-----

*1.Cargar fichero de resultados.*

*2.Evaluar fichero de respuestas*

*3.Salir*

*Elija opción (1-3):*

La **OPCIÓN 1** se encarga de pedirle al usuario el nombre del fichero de texto que contiene los resultados del test y cargar éstos en memoria. El formato de este fichero es el siguiente:

*RESULTADOS*

-----

*20*

*1 a*

*2 b*

*3 b*

*4 c*

*...*

*20 c*

**Línea 1 y 2:** Cabecera del fichero de resultado.

**Línea 3:** Un entero que indica el número de preguntas/respuestas que tiene el test.

**Línea 4, 5 y sucesivas:** Un entero que indica la pregunta y un carácter que indica la opción correcta a esa pregunta.

La **OPCIÓN 2** necesita tener cargados en memoria los resultados de los tests (la opción 1 fue utilizada anteriormente) y le pedirá al usuario el fichero que contiene las respuestas del alumno para evaluarlo. El fichero de respuesta tendrá el siguiente formato:

*Nombre: XXXXXXXXXXXXXXXX*

*Apellidos: YYYYYYYYYYYYYY*

*20*

*1 a*

*2 b*

*3 b*

*4 c*

*...*

*20 c*

**Línea 1:** Nombre del alumno.

**Línea 2:** Apellidos del alumno.

**Línea 3:** Entero con el número de preguntas/respuestas que tiene el test.

**Línea 4, 5 y sucesivas:** Un entero que indica la pregunta y un carácter que indica la respuesta del alumno a esa pregunta (puede ser mayúscula o minúscula).

Al finalizar la lectura del fichero de respuesta del alumno, el programa mostrará el nombre y apellidos del alumno así como el número de respuestas correctas e incorrectas que tiene el alumno en cuestión.

#### **NOTAS:**

- El número de preguntas/respuestas dependerá del tipo de test (variable).
- Si los ficheros respuesta de los alumnos tienen un tamaño distinto al número anterior, el programa mostrará un error y no seguirá leyendo respuestas.
- El formato de los ficheros es correcto y salvo lo anterior supondremos que están escritos como nos dice el enunciado.
- Se valorará una correcta modularización del programa, la claridad del código (incluyendo los comentarios) y una acertada utilización de la memoria dinámica.

**Ud1456Act6.** Queremos implementar en Java una aplicación, *CriptoFich*, que se encargue de encriptar/desencriptar ficheros de texto. La interfaz que presentará el programa será similar al siguiente menú:

#### **CRIPTOFICH**

-----

1. *Encriptar (Codificar).*

2. *Desencriptar (Decodificar).*

3. *Salir*

*Elija opción (1-3):*

Las opciones deben realizar las siguientes acciones:

**Encriptar:** Pedirá al usuario que introduzca el nombre del fichero de texto que queremos codificar y el fichero destino que contendrá el texto encriptado. Esta opción se encargará de codificar el texto en claro para crear un fichero codificado.

**Desencriptar:** Pedirá al usuario que introduzca el nombre del fichero de texto codificado y en dónde queremos almacenar el resultado (fichero con el texto en claro). Esta opción se encargará de leer el fichero codificado y desencriptarlo.

### Codificación/Decodificación

El algoritmo que seguiremos para codificar los mensajes de texto en claro se basa en la permutación de filas por columnas de una tabla bidimensional. Por ejemplo, el fichero de texto en claro *texto.txt* presenta el siguiente formato:

```
8
10
de lo que sucedió al ingenioso hidalgo en la venta que él imaginaba ser castillo
```

El fichero de texto especifica en las dos primeras líneas el número de filas(8) y columnas(10) que debe tener la matriz para codificar el mensaje. En memoria, por tanto, se cargaría la tabla:

d	e		l	o		q	u	e	
s	u	c	e	d	i	ó		a	l
	i	n	g	e	n	i	o	s	o
	h	i	d	a	l	g	o		e
n		l	a		v	e	n	t	a
	q	u	e		é	l		i	m
a	g	i	n	a	b	a		s	e
r		c	a	s	t	i	l	l	o

Una vez cargada en memoria, el resultado, de la codificación consiste en cambiar filas por columnas, siendo el fichero codificado *codif.txt*:

```
10
8
ds n areuih qg cniluiclegdaenaodea as inlvébtqóigelaiu oon leas lisl loeamneo
```

Donde las dos primeras líneas corresponden al número de filas(10) y columnas(8) de la matriz encriptada.

Para desencriptar tendríamos que cargar en memoria la matriz desencriptada y realizar los cambios de filas por columnas. Es decir, en memoria cargaríamos la tabla siguiente y realizaríamos la operación inversa:

d	s			n		a	r
e	u	i	h		q	g	
	c	n	i	l	u	i	c
l	e	g	d	a	e	n	a
o	d	e	a			a	s
	i	n	l	v	é	b	t
q	ó	i	g	e	l	a	i
u		o	o	n			l
e	a	s		t	i	s	l
	l	o	e	a	m	e	o

**Se pide:** Implementar la aplicación **Criptofich** suponiendo que los ficheros de texto en claro contienen el formato especificado anteriormente y una línea de texto que tendrá como tamaño máximo el número de filas por columnas especificados en el mismo formato del fichero.

**Ud1456Act7.** Realice un programa en Java de nombre *Midir.java* que funcione exactamente igual que el comando de Windows dir.