

PRÁCTICA 3. SOPA DE LETRAS AUTOMATIZADA

E	R	Í	A	Ü	R	O	B	O	T	C	B
Y	Z	F	B	D	X	X	É	D	E	A	E
S	O	R	W	Y	K	L	Z	Ú	W	B	Ü
C	N	O	L	O	G	Í	A	I	I	L	P
M	Í	D	Í	J	F	M	U	K	T	E	R
O	Ñ	O	Ü	R	C	K	K	B	J	S	O
T	Y	M	Ó	F	Ñ	H	V	É	E	I	Y
O	O	C	Í	X	P	Ñ	I	L	Á	A	E
R	Y	É	Ü	Y	S	D	Ñ	P	P	Ñ	C
U	Í	V	A	P	A	T	D	Z	Á	Í	T
V	G	Á	H	V	D	F	J	É	N	Y	O

→ INTRODUCCIÓN. El método *matches()* vol. 2

Como ya sabemos, el método *matches()* de la clase *String* nos permite comprobar que el formato de una cadena introducida es correcto gracias a las expresiones de formato que es capaz de validar. En este caso, como lo que va a introducir el usuario son los datos de una sopa de letras, deberemos controlar que los valores son todos caracteres.

La forma de validar letras con el método *matches*, dando por buenas tanto minúsculas como mayúsculas, es la siguiente:

```
lectura.matches("[a-zA-Z]+")
```

En caso de no corresponder con el formato esperado, el método devuelve el valor *false*. Utilízalo para implementar el programa de la práctica que se explica a continuación.

→ PROBLEMA A RESOLVER

Dada una matriz de M x N letras y una palabra, tenemos que programar un código que encuentre la ubicación en la matriz en la que se puede localizar dicha palabra.

La palabra debe coincidir con una línea recta e ininterrumpida de letras en la matriz (mayúsculas y minúsculas se consideran equivalentes). Es decir, si buscamos *sopa*, podemos considerar *Sopa* o *sOPa* como coincidencias válidas.

La coincidencia solamente se puede dar en dirección horizontal derecha y dirección verticales abajo (las coincidencias hacia arriba o hacia la izquierda y las diagonales no se deben tener en cuenta).

ENTRADA

El programa recibirá una entrada con el siguiente formato:

1. La primera línea preguntará y leerá un par de números enteros M (filas) y N (columnas).
2. Las siguientes M líneas de entrada contendrán N letras cada una, representando la matriz de letras donde debe buscarse la palabra. Las letras pueden estar en mayúsculas o minúsculas.

```
Introduce el número de filas (:
3
Introduce el número de columnas:
4
Introduce las letras de la fila 1: aGfj
Introduce las letras de la fila 2: hola
Introduce las letras de la fila 3: RthV
a G f j
h o l a
R t h V
```

En caso de no haber introducido **N letras**, se debe mostrar un error y finalizar el programa.

```
Introduce el número de filas:
2
Introduce el número de columnas:
3
Introduce las letras de la fila 1: loo
ERROR. Introduce datos válidos: 3 letras.
```

3. La siguiente línea pedirá una palabra a buscar. Esta palabra sólo puede contener letras mayúsculas y minúsculas (sin espacios, guiones u otros caracteres no alfabéticos).

```
Introduce el número de filas (:
3
Introduce el número de columnas:
4
Introduce las letras de la fila 1: aGfj
Introduce las letras de la fila 2: hola
Introduce las letras de la fila 3: RthV
a G f j
h o l a
R t h V
Introduce la palabra a buscar:
hola
```

SALIDA

El programa devolverá un par de números enteros que representan la ubicación en la matriz de la palabra (estarán separados por un espacio):

- El primer entero es la fila de la matriz donde se puede encontrar la primera letra de la palabra dada.
- El segundo entero es la columna de la matriz donde se puede encontrar la primera letra de la palabra dada.

Si una palabra se puede encontrar más de una vez en la matriz se debe devolver la ubicación de la ocurrencia más alta de la palabra, es decir, aquella que sitúa la primera letra de la palabra más cerca de la parte superior izquierda de la matriz. Se asumirá que la palabra se encuentra al menos una vez en la matriz.

EJEMPLOS

```
Introduce el número de filas (:
3
Introduce el número de columnas:
4
Introduce las letras de la fila 1:aGfj
Introduce las letras de la fila 2:hola
Introduce las letras de la fila 3:RthV
a G f j
h o l a
R t h V
Introduce la palabra a buscar:
hola
Encontrada!!! En la posición 1 0
```

```
Introduce el número de filas:
8
Introduce el número de columnas:
11
Introduce las letras de la fila 1:abcDEFGhigg
Introduce las letras de la fila 2:hEbKWaLDorf
Introduce las letras de la fila 3:FtaAwaldORm
Introduce las letras de la fila 4:FtmimrLqsrc
Introduce las letras de la fila 5:byBArBeTTYv
Introduce las letras de la fila 6:KLlbqwikomk
Introduce las letras de la fila 7:stREBGadhrb
Introduce las letras de la fila 8:yUiqlxcnBjk
a b c D E F G h i g g
h E b k W a l D o r f
F t a A w a l d O R m
F t m i m r L q s r c
b y B A r B e T T Y v
K L l b q w i k o m k
s t r E B G a d h r b
y U i q l x c n B j k
Introduce la palabra a buscar:
bambi
Encontrada!!! En la posición 1 2
```

Realiza un programa en *Java* que implemente la lógica del juego explicada anteriormente, usando matrices.

→ REALIZACIÓN DE LA PRÁCTICA

Sigue los siguientes pasos para realizar la práctica. **¡Ve guardando tu trabajo de vez en cuando para evitar que se borre el avance si se cierra el editor de textos u ocurre cualquier problema en tu equipo!**

1. Programa en Java la aplicación requerida
2. Plan de pruebas. Realiza las pruebas necesarias para comprobar que el programa funciona bien



ENTREGA

REALIZA UN INFORME EN PDF CON LA INFO GENERADA Y LOS PASOS SEGUIDOS PARA REALIZAR ESTA PRÁCTICA. EXPLICA TU CÓDIGO. SÚBELO TODO A LA TAREA DE AULES DISPONIBLE.

ADEMÁS, PEGA LA URL DE TU PROYECTO EN GITHUB.