

Programación

Sopa de letras

T	I	T	E	S	P	I	N	A	C	A	S	M	A
A	M	S	D	R	A	N	O	J	I	F	P	C	Z
L	C	E	B	O	L	L	A	R	S	O	A	E	H
M	O	T	L	S	Ñ	O	J	E	L	H	Y	B	I
O	Z	A	S	R	O	Ñ	F	I	L	O	C	H	O
Z	A	S	S	A	N	E	P	F	J	A	C	L	A
I	N	A	T	D	E	I	R	I	X	C	V	L	P
N	A	R	A	N	J	A	E	P	Ñ	L	E	A	L
E	H	B	E	R	E	N	J	E	N	A	P	T	A
J	O	S	I	O	N	I	L	S	O	R	Z	I	T
A	R	W	A	S	O	Z	E	I	L	N	I	S	A
C	I	R	U	E	L	A	R	A	M	D	F	E	N
K	A	Z	T	F	D	O	T	N	O	M	L	O	J
S	J	E	I	M	A	N	D	A	R	I	N	A	J

```

int n_palabras = 0; //El numero de palabras que introducen por teclado
int maximo = 0; //La palabra mas larga
char relleno = '-'; //Relleno de la sopa de letras
char caracter; //Letra de las palabras
boolean si_entra = false;
int x = 0;
int y = 0;
int ix = 0; //Derecha izquierda
int iy = 0; //Parriba pabajo
int generar = 500; //Las veces que va a volver generar coordenadas cuando la palabra no entre
int intentos = 0;
int finalx = 0, finaly = 0;
boolean libre = true;
boolean parar = false;

//Contamos el numero de palabras que vamos a introducir por teclado
for (int i = 0; i < args.length; i++) {

    n_palabras = Integer.parseInt(args[i]);
}
String[] palabras = new String[n_palabras]; //Vector donde guardamos las palabras que introducimos por teclado.
//Introducimos las palabras por teclado y las guardamos en un vector de string
for (int i = 0; i < n_palabras; i++) {
    System.out.println("Introduzca la palabra " + i);
    Scanner teclado = new Scanner(System.in);
    String palabra = teclado.nextLine();
    palabras[i] = palabra;
    if (palabra.length() > maximo) {
        maximo = palabra.length();
    }
}
//Creamos y declaramos la matriz
char sopa[][] = new char[maximo * 3][maximo * 3];
for (int i = 0; i < sopa.length; i++) {
    System.out.println();

    //Rellenar la matriz
    for (int j = 0; j < sopa.length; j++) {
        sopa[i][j] = relleno;
    }
}

String palabra = palabras[0];
for (int i = 0; i < palabras.length && parar == false; i++) { //Va cogiendo las palabras del vector
    palabra = palabras[i];
    si_entra = false;
    x = (int) (Math.random() * maximo * 3);
    y = (int) (Math.random() * maximo * 3);
    int temporalx = x;
    int temporaly = y;
    if (intentos < generar) {

        while (si_entra == false && intentos < generar) { //Saca las direcciones aleatorias
            intentos++;
            iy = 0;
            ix = 0;

            if (Math.random() < 0.5) { //Selecciona X o Y

                if (Math.random() < 0.5) { //Izquierda
                    ix = -1;

                } else if (Math.random() < 0.5) { //Derecha
                    ix = 1;

                } else {
                    ix = +1; //Diagonal abajo
                    iy = +1;

                }
            } else if (Math.random() < 0.5) { //Arriba
                iy = -1;
            } else if (Math.random() < 0.5) { //Abajo
                iy = 1;
            } else {
                ix = -1; //Diagonal arriba
                iy = -1;
            }
        }
        //Calcula el final de la palabra
        finaly = 0;
        finalx = 0;
        if (iy == 1) {

```

```

        finaly = y + palabra.length();
    }
    if (iy == -1) {
        finaly = y - palabra.length();
    }
    if (ix == 1) {
        finalx = x + palabra.length();
    }
    if (ix == -1) {
        finalx = x - palabra.length();
    }
    if (finalx >= 0 && finalx < sopa[0].length && finaly >= 0 && finaly < sopa.length) {
        for (int k = 0; k < palabra.length() && libre == true; k++) {
            caracter = palabra.charAt(k);

            if (sopa[temporalx][temporalx] == '-' || sopa[temporalx][temporalx] == caracter) { //Comprueba si la palabra cruza con otras

                temporalx += iy;
                temporalx += ix;
                libre = true;
            } else {
                libre = false;
            }
        }
        if (libre == true) {
            for (int k = 0; k < palabra.length(); k++) { //Va cogiendo las letras de las palabras
                caracter = palabra.charAt(k);
                sopa[y][x] = caracter;
                x += ix;
                y += iy;
                si_entra = true;
            }

            System.out.println();
        }
        } else {
            System.out.println("La palabra no entra...");
        }
    }

} else {
    System.out.println("Limite de intentos alcanzados,vuelva a intentarlo");
    parar = true;
}

}

//Imprimir matriz
if (parar == false) {
    for (int i = 0; i < sopa.length; i++) {
        System.out.println();

        for (int j = 0; j < sopa.length; j++) {
            System.out.print(sopa[i][j]);
        }

        System.out.println();
        System.out.println();
    }
    //Ordena el vector de palabras alfabeticamente
    System.out.println("-----LISTA DE PALABRAS ORDENADA ALFABETICAMENTE-----");

    java.util.Arrays.sort(palabras);
    for (int i = 0; i < n_palabras; i++) {
        System.out.print(palabras[i] + " ");
        System.out.println();
    }
    System.out.println();
}
}
}

```