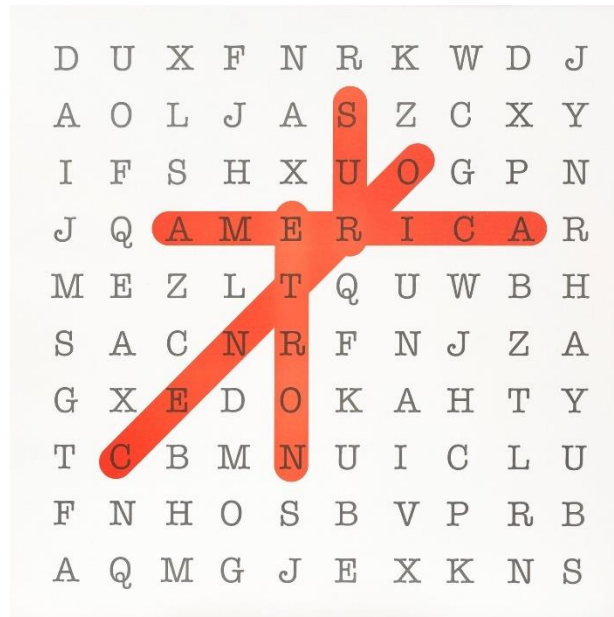


## ACTIVIDAD 5.18 - Sopa de letras

### Enunciado

Vamos a desarrollar una versión especial del famoso pasatiempo conocido como la **SOPA DE LETRAS**.



Para ello, nos vamos a hacer valer de un **array bidimensional cuadrado** (mismas filas que columnas) el cual almacenará de manera aleatoria en cada una de sus casillas **letras del alfabeto inglés en minúsculas**.

Este programa no se dedicará a resolver el pasatiempo, sino simplemente a analizar el contenido de la sopa de letras que se generará tras cada ejecución.

Al arrancar el programa, se solicitará el tamaño de la sopa de letras a generar. Una vez generada, éste deberá mostrar la matriz y dar como salida los siguientes resultados, teniendo en cuenta que, para cada punto, si existe más de un resultado, deberás mostrar una lista de todos ellos:

1. Índice de la/s fila/s con mayor número de vocales.
2. Índice de la/s columna/s con mayor número de consonantes.
3. Índice de la/s fila/s que tenga alguna letra repetida.
4. Índice de la/s columna/s que tengan la misma letra en todas sus celdas.
5. Si la diagonal principal tiene todo vocales o todo consonantes.
6. Si la diagonal secundaria tiene todo vocales o todo consonantes.

Diagonal Principal

$$\begin{pmatrix} 1 & 2 & 0 \\ 3 & 1 & 4 \\ 3 & 0 & 1 \end{pmatrix}$$

Diagonal Secundaria

$$\begin{pmatrix} 1 & 2 & 0 \\ 3 & 1 & 4 \\ 3 & 0 & 1 \end{pmatrix}$$

## Método de trabajo y normas de implementación

- Al solicitar el tamaño de la matriz, ésta debe ser un número entero positivo (mayor que 0). Si se introduce algo que no cumple con esas características (algo que no sea un número o un número que sea negativo) se mostrará de nuevo el texto Introduce el tamaño de la sopa de letras.
- La información a mostrar de **cada uno de los apartados** debe cumplir con el siguiente formato:

### 1. Índice de las filas con mayor número de vocales

Por cada fila que cumpla con el criterio, se debe mostrar el texto (puede haber más de una, por tanto, se mostrará este mensaje tantas veces como coincidencias haya):

#### 1. Fila índice X con mayor número de vocales

Si no hubiese ninguna fila con vocales, se mostraría el texto:

##### 1. La matriz sólo tiene consonantes

siendo X el valor de un índice de la matriz.

### 2. Índice de las columnas con mayor número de consonantes

Por cada columna que cumpla con el criterio se debe mostrar el texto (puede haber más de una, similar al punto anterior):

#### 2. Columna índice X con mayor número de consonantes

Si no hubiese ningún consonante en la matriz se mostraría el texto:

##### 2. La matriz sólo tiene vocales

siendo X un valor de índice válido.

### 3. Índice de las filas que tengan alguna letra repetida

Por cada fila que cumpla con el criterio se mostrará (puede haber más de una, similar a los puntos anteriores):

#### 3. Fila índice X con letras repetidas

Si no hay ninguna fila/columna que cumpla con los criterios se mostrará el texto:

#### 3. No hay filas con letras repetidas

siendo X un valor de índice válido

### 4. Índice de las columnas que tengan la misma letra en todas sus celdas

Por cada columna que cumpla con el criterio se mostrará (puede haber más de una, similar a los puntos anteriores):

#### 4. Columna índice X con sólo la letra Y

Si no hay ninguna fila/columna que cumpla con los criterios se mostrará el texto:

#### 4. No hay columnas con la misma letra

siendo X un valor de índice válido e Y la letra repetida.

#### 5. Diagonal principal con sólo un tipo de letra y

#### 6. Diagonal secundaria con sólo un tipo de letra

Si la diagonal principal o secundaria cumple con los criterios se mostrará el texto:

#### 5. Diagonal principal con el mismo tipo de letra

*o bien*

#### 6. Diagonal secundaria con el mismo tipo de letra

Si la diagonal principal o secundaria no cumple con los criterios se mostrará el texto:

#### 5. Diagonal principal con mezcla de vocales y consonantes

*o bien*

#### 6. Diagonal secundaria con mezcla de vocales y consonantes

- Debes hacer un programa que directamente muestre la información requerida y acabe mostrando la información indicada, debiendo **RESPETAR EL FORMATO QUE SE TE PRESENTA** en el siguiente **ejemplo de ejecución**:

```
Introduce el tamaño de la sopa de letras: 0
Introduce el tamaño de la sopa de letras: -2
Introduce el tamaño de la sopa de letras: hola
Introduce el tamaño de la sopa de letras: 4
```

```
r u u q
t y a t
t g q t
a s x u
```

1. Fila índice 0 con mayor número de vocales
1. Fila índice 3 con mayor número de vocales
2. Columna índice 0 con mayor número de consonantes
2. Columna índice 1 con mayor número de consonantes
2. Columna índice 3 con mayor número de consonantes
3. Fila índice 0 con letras repetidas
3. Fila índice 1 con letras repetidas
3. Fila índice 2 con letras repetidas
4. No hay columnas con la misma letra
5. Diagonal principal con mezcla de vocales y consonantes
6. Diagonal secundaria con mezcla de vocales y consonantes

## Métodos a implementar

- A modo de facilitar la implementación del programa, tu programa debe implementar como mínimo y de manera **obligatoria** los métodos indicados a continuación.
- Fíjate bien en el comentario que precede a cada uno, ya que tiene que cumplir con la especificación propuesta.

## Métodos que no deben imprimir nada por pantalla, sólo analizar el contenido de la sopa de letras y retornar el valor indicado

```
/**  
  
* Retorna una sopa de Letras aleatoria del tamaño indicado.  
  
* Si se recibe como @tamanyo un valor menor a 0, se debe devolver null (aunque esto  
nunca debería pasar en nuestro programa, de esta forma, el método quedaría más robusto  
y podría ser usado en otras aplicaciones)  
  
**/  
  
public static char[][] generarSopaLetras(int tamanyo)  
  
/**  
  
* Retorna la lista de los índices de las filas con mayor número de vocales  
  
**/  
  
public static int[] getIndicesFilasConMasVocales(char[][] sopaLetras)  
  
/**  
  
* Retorna la lista de los índices de las columnas con mayor número de consonantes **/  
  
public static int[] getIndicesColumnasConMasConsonantes(char[][] sopaLetras)  
  
/**  
  
* Devuelve si la fila indicada de la sopa tiene alguna letra repetidam, sea la que  
* sea.  
  
**/  
  
public static boolean isLetraRepetidaEnFila(char[][] sopaLetras, int indiceFila)  
  
/**  
  
* Retorna la lista de los índices de las columnas que tengan todas sus celdas con  
* una misma letra  
  
**/  
  
public static int[] getIndicesColumnasConMismaLetra(char[][] sopaLetras)  
  
/**  
  
* Retorna true si la diagonal principal de la sopa de letras tiene el mismo tipo de  
* letra en todas sus celdas ya sean vocales o consonantes  
  
**/  
  
public static boolean isDiagonalPrincipalDelMismoTipo(char[][] sopaLetras)
```

```
/**
```

```
* Retorna true si la diagonal secundaria de la sopa de letras tiene el mismo tipo  
* de letra en todas sus celdas, ya sean vocales o consonantes.
```

```
**/
```

```
public static boolean isDiagonalSecundariaDelMismoTipo(char[][] sopaLetras)
```

### ***Método que SÍ debe imprimir por pantalla***

```
/**
```

```
* Imprime los resultados de los 6 puntos exigidos por el ejercicio en función de  
* la sopa de letras recibida como parámetro
```

```
**/
```

```
public static void visualizarResultados(char[][] sopaLetras)
```

- Se valorará la implementación de **métodos adicionales** además de los obligatorios que hagan el código más legible, mantenible y reutilizable.
- Para recabar la información de los índices de las diferentes filas/columnas que cumplen con una cierta condición, por ejemplo, en el método

```
public static int[] getIndicesFilasConMasVocales(char[][] sopaLetras),
```

se puede optar por dos soluciones:

- 1 Crear un array de enteros del tamaño máximo de índices posibles, en nuestro caso, un array del tamaño de la sopa de letras (solución menos óptima ya que pueden quedar elementos del array sin sentido alguno).
- 2 Crear un array que tendrá fijado y ajustado el tamaño exacto del número de filas que tienen más vocales (solución más óptima). Necesitaríamos un paso previo adicional para conocer la cantidad de filas que tienen ese valor máximo.
- 3 Crear un array vacío que irá creciendo conforme vayamos encontrando índices que cumplan con la condición deseada (solución también más óptima ya que se acabará teniendo un array del tamaño necesario).

- En nuestro caso, vamos a optar por la **opción 2 o 3**. Para la opción 3, fíjate en la siguiente ayuda. Intenta aplicar la misma estrategia para tu solución:



## Ayuda para el manejo de arrays que crecen de tamaño en tiempo de ejecución

- Para llevar a cabo la **recolección del grupo de índices que cumplen con un determinado criterio** se puede hacer uso del método `Arrays.copyOf`. Este nos permite crear un nuevo array de tamaño determinado a partir de otro, copiando los elementos del array anterior en el nuevo que se creará. Dicho método, recibe dos parámetros:

1. El array original que se desea copiar
2. El nuevo tamaño del array creado, réplica del original.

- De esta forma, **se simulará que el array crece** en tamaño conforme se va encontrando nuevos elementos a añadir en él.
- El objetivo es dar al *nuevo array* un tamaño de **una unidad más** que el *array original* para que en el *array réplica* se permita poder introducir el nuevo elemento que queramos añadir (en nuestro caso, el índice de la nueva fila).
- En el siguiente ejemplo se muestra un método que recolecta los índices de las filas de la sopa de letras que **contienen en alguna de sus celdas la letra minúscula a**:

```
public static int[] indicesFilasQueTienenUnaA(char[][] sopaLetras) {  
  
    int[] indices = new int[0]; // Partimos de un array de tamaño 0  
  
    for (int i = 0; i < sopaLetras.length; i++) {  
        for (int j = 0; j < sopaLetras[i].length; j++) {  
  
            if (matriz[i][j] == 'a') {  
                indices = Arrays.copyOf(indices, indices.length + 1);  
                indices[indices.length-1] = i; // Se añade nuevo índice en la nueva celda  
creada para tal fin  
                break; // Rompemos el bucle interno para pasar a la siguiente fila  
            }  
        }  
    }  
    return indices;  
}
```

- Al final se devolverá un array del tamaño del número de índices encontrados.

## Más ejemplos de ejecución:

Introduce el tamaño de la sopa de letras: **6**

f v n i q o  
q o j c r u  
c t s r l r  
v r t r u z  
g f m w k z  
q p v d n q

1. Fila índice 0 con mayor número de vocales
1. Fila índice 1 con mayor número de vocales
2. Columna índice 0 con mayor número de consonantes
2. Columna índice 2 con mayor número de consonantes
3. Fila índice 2 con letras repetidas
3. Fila índice 3 con letras repetidas
3. Fila índice 5 con letras repetidas
4. No hay columnas con la misma letra
5. Diagonal principal con mezcla de vocales y consonantes
6. Diagonal secundaria con mezcla de vocales y consonantes

--

Introduce el tamaño de la sopa de letras: **2**

e c  
k l

1. Fila índice 0 con mayor número de vocales
2. Columna índice 1 con mayor número de consonantes
3. No hay filas con letras repetidas
4. No hay columnas con la misma letra
5. Diagonal principal con mezcla de vocales y consonantes
6. Diagonal secundaria con el mismo tipo de letra.

--

Introduce el tamaño de la sopa de letras: **1**

q

1. La matriz sólo tiene consonantes
2. Columna índice 0 con mayor número de consonantes
3. No hay filas con letras repetidas
4. Columna índice 0 con sólo la letra q
5. Diagonal principal con el mismo tipo de letra.
6. Diagonal secundaria con el mismo tipo de letra.

Introduce el tamaño de la sopa de letras: **8**

n r o l k a a z  
x b e n a o s e  
v e m c u w w f  
h i m v g h i i  
u c n q w b i n  
m l t p w m f b  
c r i m t z g i  
q c e w g l f c

1. Fila índice 1 con mayor número de vocales
2. Columna índice 3 con mayor número de consonantes
3. Fila índice 0 con letras repetidas
3. Fila índice 1 con letras repetidas
3. Fila índice 2 con letras repetidas
3. Fila índice 3 con letras repetidas
3. Fila índice 4 con letras repetidas
3. Fila índice 5 con letras repetidas
3. Fila índice 6 con letras repetidas
3. Fila índice 7 con letras repetidas
4. No hay columnas con la misma letra
5. Diagonal principal con el mismo tipo de letra.
6. Diagonal secundaria con el mismo tipo de letra.