

AlphabetSoup

El proyecto consiste en un Juego de Sopa de Palabras.

Tecnologías

Fue desarrollado en las siguientes tecnologías:

- Lenguaje de programación: Java
- Marco de Trabajo: Spring Boot

Respuestas al Ejercicio Sitrack

1 - Problema

Explicación del proyecto AlphabetSoup y del algoritmo realizado.

El proyecto consiste en una Sopa de Palabras, en la cual las palabras pueden estar ubicadas: Horizontalmente de Izquierda a Derecha, Horizontalmente de Derecha a Izquierda, Verticalmente de Arriba hacia Abajo, Verticalmente de Abajo hacia Arriba, Diagonal de Izquierda a Derecha de Arriba hacia Abajo, Diagonal de Izquierda a Derecha de Abajo hacia Arriba, Diagonal de Derecha a Izquierda de Arriba hacia Abajo y Diagonal de Derecha a Izquierda de Abajo hacia Arriba. La Sopa de palabras puede ser de distintos tamaños tanto de alto como de ancho, específicamente está configurada para que ambos valores no puedan ser menores a 15 y mayores a 80.

El algoritmo que crea la Sopa de Palabras puede ser probado completamente mediante la consola una vez ejecutado el proyecto, para lo cual se deben seguir los siguientes pasos:

- 1- Primeramente, se debe ingresar los valores del Alto y el Ancho (Un número entre 15 y 80 incluyendo a ambos).
- 2- Posteriormente se deben ingresar las Ubicaciones de las palabras en la sopa (Especificando Si a la que desee que esté y No a la que no desee que esté).
- 3- Por último se debe ingresar la Categoría de las palabras con las que se va a jugar en la sopa (Especificando un número del 1 al 11).

Una vez insertados estos parámetros se genera correctamente la Sopa de Palabras, mostrando el Ancho y Alto de la Sopa, las palabras a encontrar en la Sopa generadas aleatoriamente y en correspondencia con la Categoría seleccionada y la Sopa de Palabras generada.

Importante!!!: Con el objetivo de que el algoritmo pueda ser probado fácilmente se muestran las Coordenada par inicial (fila, columna) y par final (fila, columna) de las palabras a encontrar en la sopa, los valores de estos pares deben comenzar desde el 1, 2, 3, 4, 5, ... hasta el alto o ancho que se ingresó, correspondiéndose el Alto a las Filas y el Ancho a las Columnas.

Para comenzar a Jugar en la Sopa de Palabras:

1- Se debe buscar alguna de las palabras mostradas, en la Sopa generada y cuando se encuentre determinar su par fila y columna inicial y su par fila y columna final e ir ingresando estos valores al sistema siguiendo los pasos que se le van indicando, primero la Fila Inicial, después la Columna Inicial, después la Fila Final y por último la Columna Final.

2- El sistema le indicará si los valores ingresados son correctos o incorrectos. En caso de ser correctos la palabra encontrada es transformada a sus correspondiente mayúscula y en caso de ser incorrectos se le notifica con un mensaje.

3- Así se debe ir encontrando de esta forma todas las palabras que se encuentran ocultas en la sopa y una vez que todas hayan sido encontradas el sistema le notifica que ha ganado el juego y que ya no quedan palabras por descubrir.

2 - Web Services

Explicación del Web Service AlphabetSoup.

Endpoint 1

Endpoint para crear la sopa de palabras:

POST <http://localhost:8085/alphabetSoup>

```
{
  "w": 15,    //Por defecto tiene valor 15
  "h": 15,    //Por defecto tiene valor 15
  "ltr": true, //Por defecto tiene valor true
  "rtl": false, //Por defecto tiene valor false
  "ttb": true, //Por defecto tiene valor true
  "btt": false, //Por defecto tiene valor false
  "d" : false, //Por defecto tiene valor false
}
```

Este Endpoint permite ingresar los parámetros para generar la Sopa de Palabras.

"w" - constituye el Alto de la Sopa de Palabras.

"h" - constituye el Ancho de la Sopa de Palabras.

"ltr" - permite habilitar (true) o deshabilitar (false) las palabras que van de Izquierda a Derecha (Horizontal de Izquierda a Derecha), teniendo valor true por defecto.

"rtl" - permite habilitar (true) o deshabilitar (false) las palabras que van de Derecha a Izquierda (Horizontal de Derecha a Izquierda), teniendo valor false por defecto.

"ttb" - permite habilitar (true) o deshabilitar (false) las palabras que van de Arriba hacia Abajo (Vertical de Arriba hacia Abajo), teniendo valor true por defecto.

"btt" - permite habilitar (true) o deshabilitar (false) las palabras que van de Abajo hacia Arriba (Vertical de Abajo hacia Arriba), teniendo valor false por defecto.

"d" - permite habilitar (true) o deshabilitar (false) las palabras que van en Diagonal (Diagonal de Izquierda a Derecha de Arriba hacia Abajo, Diagonal de Izquierda a Derecha de Abajo hacia Arriba, Diagonal de Derecha a Izquierda de Arriba hacia Abajo y Diagonal de Derecha a Izquierda de Abajo hacia Arriba), teniendo valor false por defecto.

Este Endpoint crea la Sopa de Palabras teniendo en cuenta los parámetros ingresados y devuelve el id de la Sopa de Palabras creada en formato UUID en caso que se cree correctamente y en caso de no crearse correctamente se muestra un mensaje con el error y se lanza el error 400.

En caso correcto se muestra el id en formato UUID y se lanza el (HttpStatus.CREATED):

```
{  
  "id": "44f09bae-d089-440c-81d1-34890886fc67"  
}
```

En caso incorrecto se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el mensaje:

```
{  
  "mensaje": "No se pudo crear la Sopa de Palabras"  
}
```

Endpoint 2

Endpoint para visualizar la lista de palabras:

GET <http://localhost:8085/alphabetSoup/list/44f09bae-d089-440c-81d1-34890886fc67>

El Endpoint visualiza la lista de palabras que se encuentran ocultas en la Sopa de Palabras.

En caso correcto muestra la lista de palabras y se lanza el (HttpStatus.OK):

```
[  
  "periodista",  
  "matanzas",  
  "mesa",  
  "amigo",  
  "prima",  
  "peru",  
  "hermana"  
]
```

En caso que el id pasado sea incorrecto se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```
[  
  "El Id es incorrecto"  
]
```

En caso que no haya una Sopa de palabras creada se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el siguiente mensaje:

```
[  
  "No existe una Sopa de Palabras creada"  
]
```

Endpoint 3

Endpoint para visualizar la Sopa de Palabras:

GET <http://localhost:8085/alphabetSoup/view/44f09bae-d089-440c-81d1-34890886fc67>

El Endpoint visualiza la Sopa de Palabras.

En caso correcto muestra la Sopa de Palabras en texto plano y se lanza el (HttpStatus.OK):

```
f|s|l|e|ñ|o|e|w|n|v|c|a|r|p|v|e|r|x|b|j|i|x|q|z|m|
b|u|e|x|f|h|s|f|h|y|q|k|y|z|i|g|b|w|i|g|b|q|k|d|f|
b|t|r|a|p|q|j|p|f|k|h|r|h|x|f|u|u|y|f|z|k|w|p|j|y|
y|l|t|e|j|t|b|i|u|r|n|l|i|ñ|c|b|d|s|m|x|o|v|y|t|j|
q|u|e|ñ|p|j|w|y|i|a|o|c|f|e|r|q|e|y|p|o|f|g|a|ñ|v|
s|r|g|r|e|ñ|v|y|u|w|g|f|a|v|y|r|y|n|a|e|t|w|u|h|f|
r|o|s|m|s|w|f|a|j|p|i|y|h|c|t|b|m|f|x|q|i|v|r|z|v|
c|z|h|s|ñ|p|n|b|u|j|m|l|m|l|u|z|s|z|o|m|d|m|s|e|t|
k|b|w|y|p|a|c|b|m|e|a|n|d|w|s|s|y|e|c|w|r|z|p|i|f|
f|v|v|n|m|r|h|n|h|l|e|t|p|v|y|s|a|z|n|a|t|a|m|f|ñ|
n|f|f|r|s|q|i|t|s|b|d|o|ñ|m|n|a|h|m|f|h|i|x|c|l|e|
e|x|e|n|b|t|d|m|v|b|e|r|k|x|r|p|u|g|z|p|n|l|y|d|t|
v|h|b|i|l|n|k|l|a|p|k|d|b|y|k|f|ñ|v|o|q|f|l|y|ñ|s|
y|a|z|z|t|ñ|w|z|x|o|z|a|s|y|v|r|n|x|s|y|c|k|z|j|w|
b|o|u|m|a|b|r|q|f|t|h|u|i|u|u|h|u|u|e|m|t|a|s|b|t|
k|x|h|l|p|d|ñ|u|x|i|P|E|R|I|O|D|I|S|T|A|e|w|p|m|j|
m|i|u|q|r|d|e|f|j|v|w|s|p|c|v|a|y|p|w|y|p|y|z|e|u|
m|d|c|e|m|q|g|k|l|p|v|o|v|y|k|e|s|o|e|m|t|k|x|s|p|
b|b|v|r|z|l|b|r|u|g|m|h|b|d|y|o|k|ñ|d|f|y|a|q|a|u|
d|u|z|z|w|d|h|s|r|s|q|i|i|v|a|e|w|p|o|q|l|w|a|c|o|
```

En caso que el id pasado sea incorrecto se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```
[
  "El Id es incorrecto"
]
```

En caso que no haya una Sopa de palabras creada se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el siguiente mensaje:

```
[
  "No existe una Sopa de Palabras creada"
]
```

Endpoint 4

Endpoint para indicar que hemos encontrado una Palabra en la Sopa:

PUT <http://localhost:8085/alphabetSoup/44f09bae-d089-440c-81d1-34890886fc67>

```
{  
  "sr": 16,  
  "sc": 11,  
  "er": 16,  
  "ec": 20  
}
```

Este Endpoint permite ingresar las coordenadas par inicial y par final de la palabra encontrada en la Sopa de Palabras.

"**sr**" - contituye la fila inicial.

"**sc**" - contituye la columna inicial.

"**er**" - contituye la fila final.

"**ec**" - contituye la columna final.

En caso que las coordenadas sean válidas se lanza el (HttpStatus.OK) y se muestra el siguiente mensaje:

```
[  
  "Las coordenadas insertadas son válidas, se ha modificado correctamente la Sopa de Palabras "  
]
```

En caso que las coordenadas sean inválidas o ya se hayan encontrado anteriormente se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```
[  
  "Las coordenadas insertadas son inválidas o ya han sido encontradas anteriormente, la Sopa de Palabras no ha sido modificada "  
]
```

En caso que el id pasado sea incorrecto se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```
[  
  "El Id es incorrecto"  
]
```

En caso que no haya una Sopa de palabras creada se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el siguiente mensaje:

```
[
  "No existe una Sopa de Palabras creada"
]
```

En este Endpoint si las coordenadas son Válidas además de notificarlo con un mensaje, la Sopa de Palabras es modificada transformando la palabra encontrada a su correspondiente mayúscula. En este caso la palabra periodista en las coordenadas par Inicial (16,11) y par Final (16, 20) es transformada a PERIODISTA.

```

      11      20
f|s|l|e|ñ|o|e|w|n|v|c|a|r|p|v|e|r|x|b|j|i|x|q|z|m|
b|u|e|x|f|h|s|f|h|y|q|k|y|z|i|g|b|w|i|g|b|q|k|d|f|
b|t|r|a|p|q|j|p|f|k|h|r|h|x|f|u|u|y|f|z|k|w|p|j|y|
y|l|t|e|j|t|b|i|u|r|n|l|i|ñ|c|b|d|s|m|x|o|v|y|t|j|
q|u|e|ñ|p|j|w|y|i|a|o|c|f|e|r|q|e|y|p|o|f|g|a|ñ|v|
s|r|g|r|e|ñ|v|y|u|w|g|f|a|v|y|r|y|n|a|e|t|w|u|h|f|
r|o|s|m|s|w|f|a|j|p|i|y|h|c|t|b|m|f|x|q|i|v|r|z|v|
c|z|h|s|ñ|p|n|b|u|j|m|l|m|l|u|z|s|z|o|m|d|m|s|e|t|
k|b|w|y|p|a|c|b|m|e|a|n|d|w|s|s|y|e|c|w|r|z|p|i|f|
f|v|v|n|m|r|h|n|h|l|e|t|p|v|y|s|a|z|n|a|t|a|m|f|ñ|
n|f|f|r|s|q|i|t|s|b|d|o|ñ|m|n|a|h|m|f|h|i|x|c|l|e|
e|x|e|n|b|t|d|m|v|b|e|r|k|x|r|p|u|g|z|p|n|l|y|d|t|
v|h|b|i|l|n|k|l|a|p|k|d|b|y|k|f|ñ|v|o|q|f|l|y|ñ|s|
y|a|z|z|t|ñ|w|z|x|o|z|a|s|y|v|r|n|x|s|y|c|k|z|j|w|
b|o|u|m|a|b|r|q|f|t|h|u|i|u|u|h|u|u|e|m|t|a|s|b|t|
16 k|x|h|l|p|d|ñ|u|x|i|PERIODISTA|e|w|p|m|j|
m|i|u|q|r|d|e|f|j|v|w|s|p|c|v|a|y|p|w|y|p|y|z|e|u|
m|d|c|e|m|q|g|k|l|p|v|o|v|y|k|e|s|o|e|m|t|k|x|s|p|
b|b|v|r|z|l|b|r|u|g|m|h|b|d|y|o|k|ñ|d|f|y|a|q|a|u|
d|u|z|z|w|d|h|s|r|s|q|i|i|v|a|e|w|p|o|q|l|w|a|c|o|
```

3 - Repositorio

Explicación de cómo configurar el entorno para correr el código de AlphabetSoup y poder analizarlo.

Configuración del entorno para correr el proyecto:

Para configurar el entorno del proyecto y poder probarlo se debe realizar lo siguiente:

Primera variante

- 1- Instalar la jdk de Java, específicamente fue desarrollado con la versión jdk1.8.0_181.
- 2- Configurar su Variable de entorno en el sistema, para lo cual se debe realizar lo siguiente:

2.1- Para configurar la variable de entorno de la jdk de Java, primero se busca en la barra de búsqueda del menú Inicio "Variables de entorno" y selecciona "Editar las variables de entorno del sistema", una vez ahí se selecciona el botón "Variable de entorno" que se encuentra en la parte de abajo de la ventana.

2.2- Una vez abierta la nueva ventana "Variables de entorno", se va al apartado "Variables del Sistema" y selecciona el botón "Nueva".

2.3- En el formulario que se abre se debe digitar como nombre de la variable "JAVA_HOME" y como valor de la variable la ruta donde está instalada la jdk de Java, ejemplo: (C:\Program Files\Java\jdk1.8.0_181).

2.4- Por último se debe editar la Variable del Sistema con nombre "Path" y añadir a su lista seleccionando el botón "Nuevo" "%JAVA_HOME%\bin", al terminar se da Aceptar.

2.5- Para probar que funciona correctamente se puede ejecutar en la consola del sistema "java -versión" y como respuesta se debe ver la versión de Java instalada.

3- Instalar Apache-Maven en cualquiera de sus versiones. Fue probado con la versión apache-maven-3.6.3-bin.zip descargada de <https://maven.apache.org/download.cgi?Preferred=ftp://ftp.osuosl.org/pub/apache/>, solo hay que descomprimir el archivo en "C:\Program Files" o en otra ruta y seguir los pasos siguientes para configurar su variable de entorno del sistema.

3.1- Para instalar Apache-Maven se deben configurar las variables de entorno de la jdk de Java primero y del Apache-Maven después.

3.2- Para configurar la variable de entorno de apache-maven, primero se busca en la barra de búsqueda del menú Inicio "Variables de entorno" y selecciona "Editar las variables de entorno del sistema", una vez ahí se selecciona el botón "Variable de entorno" que se encuentra en la parte de abajo de la ventana.

3.3- Una vez abierta la nueva ventana "Variables de entorno", se va al apartado "Variables del Sistema" y selecciona el botón "Nueva".

3.4- En el formulario que se abre se debe digitar como nombre de la variable "MAVEN_HOME" y como valor de la variable la ruta donde está la carpeta del maven, ejemplo: (C:\Program Files\apache-maven-3.6.3).

3.5- Por último se debe editar la Variable del Sistema con nombre "Path" y añadir a su lista seleccionando el botón "Nuevo" "%MAVEN_HOME%\bin", al terminar se da Aceptar.

3.6- Para probar que funciona correctamente se puede ejecutar en la consola del sistema "mvn -v" y como respuesta se debe ver la versión de Maven instalada.

4- Una vez instalada correctamente la jdk de Java y el apache-maven con sus respectivas variables de entorno del sistema para correr el proyecto solo se debe abrir la consola del sistema (con "Ctr-R" y se escribe "cmd") y se va a la carpeta raíz del proyecto.

5- Una vez ahí se debe ejecutar el comando "mvn clean install" obteniéndose una salida de éxito "BUILD SUCCESS".

6- Por último, se debe ejecutar el comando "mvn spring-boot:run" el cual inicia el proyecto en el puerto 8085.

Segunda variante

1- Instalar la jdk de Java, específicamente fue desarrollado con la versión jdk1.8.0_181.

2- Configurar su Variable de entorno en el sistema, como se especificó en la variante anterior.

3- Instalar un IDE para levantar el proyecto, se recomienda "IntelliJ IDEA", que fue el utilizado en el desarrollo del proyecto, específicamente en su versión 2019.3.2.

4- Una vez abierto el proyecto con el IDE, se le da "Run" al proyecto y el mismo inicia en el puerto 8085.

Como parte de la respuesta a este Ejercicio también se encuentra la documentación del proyecto que se ha realizado en este documento y la publicación en GitHub del proyecto, específicamente en la url: <https://github.com/ssjimenez89/AlphabetSoup>.

4 - Docker

No he trabajado antes con Docker, aunque estoy informado teóricamente de que es una herramienta de virtualización. La cual a diferencia de las máquinas virtuales suele utilizarse para levantar máquinas independientes con sistemas operativos ligeros y en muy poco tiempo.

Otra de sus principales ventajas por encima de las máquinas virtuales es que los contenedores Docker no instalan el Sistema operativo completo, sino que comparten el núcleo y algunas otras bibliotecas core con el sistema anfitrión.

Los contenedores pueden desplegarse fácilmente en entornos de integración continua, son ideales para obtener un entorno distribuido y clusterizado.

Reducen considerablemente el tiempo de despliegue y actualización de las aplicaciones.

Nota: Intenté descargar Docker Desktop on Windows de la página oficial <https://docs.docker.com/docker-for-windows/install/> para instalarlo y me da el siguiente mensaje:

"403 Forbidden

Since Docker is a US company, we must comply with US export control regulations. In an effort to comply with these, we now block all IP addresses that are located in Cuba, Iran, North Korea, Republic of Crimea, Sudan, and Syria. If you are not in one of these cities, countries, or regions and are blocked, please reach out to <https://support.docker.com> "

Todo parece indicar que está bloqueado para Cuba, si hubiese conseguido la instalación por otra vía hubiera intentado hacer el ejercicio.

5 - Propuesta

Con el objetivo de agregarle un valor adicional al Juego se definieron 11 Categorías para que el usuario seleccione con cual tipo de palabra desea jugar la Sopa de Palabras.

- 1- **"Todas las palabras"**: Permite generar aleatoriamente palabras de Todas las Categorías.
- 2- **"Animales"**: Permite generar aleatoriamente nombres de Animales como León, gato, Perro, etc.
- 3- **"Naturaleza"**: Permite generar aleatoriamente palabras relacionadas con la Naturaleza como árbol, hoja, aire, agua, etc.
- 4- **"Figuras"**: Permite generar aleatoriamente los nombres de las Figuras geométricas como cuadrado, triangulo, circulo, etc.
- 5- **"Hogar"**: Permite generar aleatoriamente palabras relacionadas con el Hogar como casa, ventana, habitación, etc.
- 6- **"Familia"**: Permite generar aleatoriamente palabras relacionadas con la Familia como padre, madre, hermano, hermana, etc.
- 7- **"Países"**: Permite generar aleatoriamente nombres de Países como cuba, españa, argentina, francia, etc.
- 8- **"Ciudades"**: Permite generar aleatoriamente nombres de Ciudades como habana, madrid, moscu, paris, etc.

- 9- **"Deporte"**: Permite generar aleatoriamente nombres de Deportes como ciclismo, football, basket, etc.
- 10- **"Profesiones"**: Permite generar aleatoriamente nombres de Profesiones como maestro, ingeniero, medico, etc.
- 11- **"Colores"**: Permite generar aleatoriamente nombres de Colores como rojo, azul, verde, etc.

Cuando se prueba el Algoritmo mediante la consola, una vez ingresado los valores del Alto y el Ancho y seleccionadas las Ubicaciones de las palabras, el Sistema le indica que seleccione la Categoría de Palabras con la que desea Jugar y lo único que hay que hacer es ingresar un número del 1 al 11 según la Categoría deseada.

EL sistema va guiando perfectamente la consecutividad de estos pasos iniciales sin complejidad alguna y con toda la explicación necesaria para ir guiando al usuario.

Como otro valor agregado al Ejercicio se definieron otros Endpoint para apoyar la interacción con el Algoritmo, los cuales se especifican a continuación:

Primero especificar que en el **Endpoint 1** realizado para crear la Sopa de Palabras también se puede especificar la Categoría.

Endpoint para crear la sopa de palabras:

POST <http://localhost:8085/alphabetSoup>

```
{  
  "w": 15,    //Por defecto tiene valor 15  
  "h": 15,    //Por defecto tiene valor 15  
  "ltr": true, //Por defecto tiene valor true  
  "rtl": false, //Por defecto tiene valor false  
  "ttb": true, //Por defecto tiene valor true  
  "btt": false, //Por defecto tiene valor false  
  "d" : false, //Por defecto tiene valor false  
  "c" : 1      // Por defecto tiene valor 1, generando palabras de Todas las Categorías  
}
```

Este Endpoint permite ingresar también el parámetro "c" indicando la Categoría de las Palabras a generar en la Sopa de Palabras.

"c" - permite indicar el número de la Categoría con un número del 1 - 11, teniendo valor de 1 por defecto.

Otros Endpoint realizados:

Endpoint 5

Endpoint para Visualizar la lista de palabras que ya han sido encontradas en la Sopa de Palabras:

GET <http://localhost:8085/alphabeticSoup/list/found/44f09bae-d089-440c-81d1-34890886fc67>

En caso correcto muestra la lista de palabras encontradas y se lanza el (HttpStatus.OK):

```
[  
  "periodista"  
]
```

En caso que el id pasado sea incorrecto se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```
[  
  "El Id es incorrecto"  
]
```

En caso que no haya una Sopa de palabras creada se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el siguiente mensaje:

```
[  
  "No existe una Sopa de Palabras creada"  
]
```

Endpoint 6

Endpoint para Visualizar la Coordenadas de las palabras que se encuentran en la Sopa de Palabras. Este Endpoint tiene como único objetivo mostrar las Coordenadas par Inicial (fila, columna) y par Final (fila, columna) para que el juego se pueda jugar de forma rápida y verificar su correcto funcionamiento. Además, se va especificando las Coordenadas que ya han sido encontradas anteriormente.

GET <http://localhost:8085/alphabeticSoup/coordinate/44f09bae-d089-440c-81d1-34890886fc67>

En caso correcto muestra enumeradas las Coordenadas de las palabras en la Sopa y lanza el (HttpStatus.OK):

```
[
  "1-(16,11) (16,20) !!!Coordenada encontrada!!! ",
  "2-(10,23) (10,16)",
  "3-(16,24) (19,24)",
  "4-(9,11) (5,11)",
  "5-(9,5) (13,9)",
  "6-(5,5) (2,2)",
  "7-(13,2) (7,8)"
]
```

En caso que el id pasado sea incorrecto se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```
[
  "El Id es incorrecto"
]
```

En caso que no haya una Sopa de palabras creada se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el siguiente mensaje:

```
[
  "No existe una Sopa de Palabras creada"
]
```

Endpoint 7

Endpoint para Visualizar Datos de la Sopa de Palabras. Puede ser útil para conocer las especificaciones de la Sopa de Palabras que está creada y con la que se está jugando en ese momento.

GET <http://localhost:8085/alphabetSoup/data/44f09bae-d089-440c-81d1-34890886fc67>

En caso correcto muestra los siguientes Datos y lanza el (HttpStatus.OK):

```
{
  "alto": 20,
  "ancho": 25,
  "categoriasPalabras": "TODAS_LAS_PALABRAS",
  "ubicaciones": [
    "HORIZONTAL_IZQUIERDA_DERECHA",
    "HORIZONTAL_DERECHA_IZQUIERDA",
```

```

        "VERTICAL_ARRIBA_ABAJO",
        "VERTICAL_ABAJO_ARRIBA",
        "DIAGONAL_IZQUIERDA_DERECHA_ARRIBA_ABAJO",
        "DIAGONAL_DERECHA_IZQUIERDA_ABAJO_ARRIBA",
        "DIAGONAL_IZQUIERDA_DERECHA_ABAJO_ARRIBA",
        "DIAGONAL_DERECHA_IZQUIERDA_ARRIBA_ABAJO"
    ],
    "totalDePalabras": "(7)",
    "palabras": [
        "periodista",
        "matanzas",
        "mesa",
        "amigo",
        "prima",
        "peru",
        "hermana"
    ],
    "totalDePalabrasEncontradas": "(1/7)",
    "palabrasEncontradas": [
        "periodista"
    ]
}

```

En caso que el id pasado sea incorrecto se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```

[
    "El Id es incorrecto"
]

```

En caso que no haya una Sopa de palabras creada se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el siguiente mensaje:

```

[
    "No existe una Sopa de Palabras creada"
]

```

Endpoint 8

Endpoint para Eliminar la Sopa de Palabras que se encuentra creada.

DELETE <http://localhost:8085/alphabetSoup/44f09bae-d089-440c-81d1-34890886fc67>

En caso correcto lanza el (HttpStatus.OK) y muestra el siguiente mensaje:

```
{  
  "mensaje": "La Sopa de Palabras ha sido eliminada correctamente"  
}
```

En caso que el id pasado sea incorrecto se lanza el (HttpStatus.NOT_FOUND) y se muestra el siguiente mensaje:

```
[  
  "El Id es incorrecto"  
]
```

En caso que no haya una Sopa de palabras creada se lanza el error 400 (HttpStatus.BAD_REQUEST) y se muestra el siguiente mensaje:

```
[  
  "No existe una Sopa de Palabras creada"  
]
```