Datos Técnicos

Autor:
Jesús Gabriel Peña

Caracas, Octubre, 2017

Introducción

La aplicación se ejecuta bajo la estructura MVC (*Modelo-Vista-Controlador*) del framework Laravel, el cual es un framework de PHP de código abierto que añade funcionalidades extras, y al mismo tiempo, mantiene el código simple y fácil de actualizar.

La aplicación no solo hace uso de funcionalidades de Laravel, sino que también hace uso de Bootstrap 4.0, el cual es un framework de CSS para crear interfaces profesionales y responsivas de manera sencilla, para el diseño de la interfaz de la aplicación. También hace uso de JQuery 3.2.1, el cual es un framework de Javascript, para manejar las validaciones, peticiones AJAX y animaciones de la aplicación.

La aplicación implementa las siguientes funcionalidades de Laravel:

-Rutas
-Controladores
-Validaciones
-Lenguaje
-Modelo de interfaz Blade

para mayor facilidad en el manejo de la información y en la visualización de la misma.

sopa_letras.blade.php

El front-end de la aplicación se encuentra en la ruta **`resources/views/`** bajo el nombre de **`sopa_letras.blade.php`**. El archivo contiene todo el código HTML de la aplicación, junto con la carga de los scripts y los estilos.

La carga de los scripts y estilos se ejecuta mendiante el *include asset* del Modelo Blade, para cargar los archivos que se encuentran en la carpeta **`public/js`** y **`public/css`** respectivamente. También se encuentra el **`csrf_field()`**, para validar el token generado por Laravel.

El formulario está encapsulado en un contenedor fluido de Bootstrap para lograr un diseño responsive para el usuario, y cada campo usa clases propias de Bootstrap para lograr uan buena apriencia.

El código contiene un **`<div>`** invisible para el usuario bajo la clase **`errormsg`**. Esta clase sirve como identificador para el script **`SopaQuery.js`** para mostrar los mensajes de error en caso de que exista un error en el formulario, como datos vacíos, incompletos o erróneos.

Al final del código, se encuentra una tabla invisible para el usuario. Esta tabla será usada por el script **`SopaQuery.js`** para generar la sopa de letras y mostrarla por pantalla, junto con el número de coincidencias encontradas. En cada iteración, el diseño de la tabla, junto con el mensaje de coincidencias encontradas, se reinicia a su valor original para una nueva operación. La tabla está bajo la clase t**`able-responsive`** y un estilo **`overflow:auto`**, para garantizar un diseño responsive. Si la tabla llegase a exceder el tamaño del contenedor, se mostrarán barras de navegación para navegar en la tabla.

SopaQuery.js

El script de la aplicación se encuentra en la dirección **public/js** bajo el nombre de **SopaQuery.js**, que contiene todo el código Javascript de la aplicación. El archivo es cargado por la vista **sopa_letras.blade.php** para validar, procesar y mostrar la data del formulario, junto con el control de animaciones variadas para los contenedores y mensajes.

Las validaciones para el campo **Filas** y el campo **Columnas** del formulario son las mismas: JQuery verifica los campos cuando ocurren cambios, para verificar que los datos sean correctos. Si los campos poseen valores mayores o menores a los permitidos, el script lo corrige automáticamente colocando los valores permitidos.

La validación para el campo **Datos** se efectua en base a los valores de los campos **Filas** y **Columnas.** Como la cantidad de letras de la sopa la determina el largo y ancho de la misma, la data debe tener exactamente esa cantidad de carácteres. Si la data introducida es mayor o menor a la cantidad requerida, se despliegará un mensaje para que el usuario lo corriga. El script elimina todo espacio en blanco, junto con saltos de linea, para garantizar que la data sea la misma sin importar como sea introducida por el usuario.

La validación para el campo **Palabra** se efectua en base a su longitud. Si se introduce una letra nada más, se despliegará un mensaje de error para que el usuario lo corrija.

Cuando todos los datos estén en orden, se efectua una petición **AJAX** usando el método **POST** a la url **/ValidarSopa**. Si el controlador que procesa la data devuelve un array con mensaje de errores, se despliegará los mensajes pertinentes y se para la ejecución. Caso contrario, JQuery usará los datos del formulario para crear la sopa de letras y mostrarla en pantalla. Una vez que la sopa esté terminada, mostrará la cantidad de coincidencias encontradas que el controlador devuelve en caso de éxito.

SopaController.php

El controlador principal de la aplicación se encuentra en la dirección **app/Http/Controllers** bajo el nombre **SopaController.php**. El controlador es el responsable de efectuar chequeos extras a la data del formulario, y procesar la información para buscar coincidencias en la sopa.

El controlador hace uso de la clase **Validator,** que permite validar la información de manera más sencilla. El procesamiento de la información depende de la respuesta de **Validator.** En caso de error, le devuelve a **SopaQuery.js** un array con los errores encontrados, en caso contrario, **SopaController** sigue con el procesamiento de la información.

Si los datos del formulario son correctos, **SopaController** crea un array de dos dimensiones para poder efectuar la búsqueda de la palabra clave en la sopa de letras. El array es creado recorriendo la data y almacenando cada letra en una posición del array. Cada vez que el array añade una letra, esta es eliminada de la data para proseguir a la siguiente letra.

Durante el recorrido del array, si se encuentra la primera letra de la palabra clave, esa coordenada será usada como punto clave para efectuar la búsqueda en ocho(8) sentidos. Cada búsqueda es un ciclo independiente y se ejecutan de manera secuencial. Cada vez que se encuentre una coincidencia, la variable **$coincidencias** aumentará, y cuando el recorrido del array termine, **SopaController** devuelve la variable como mensaje de éxito en la operación.