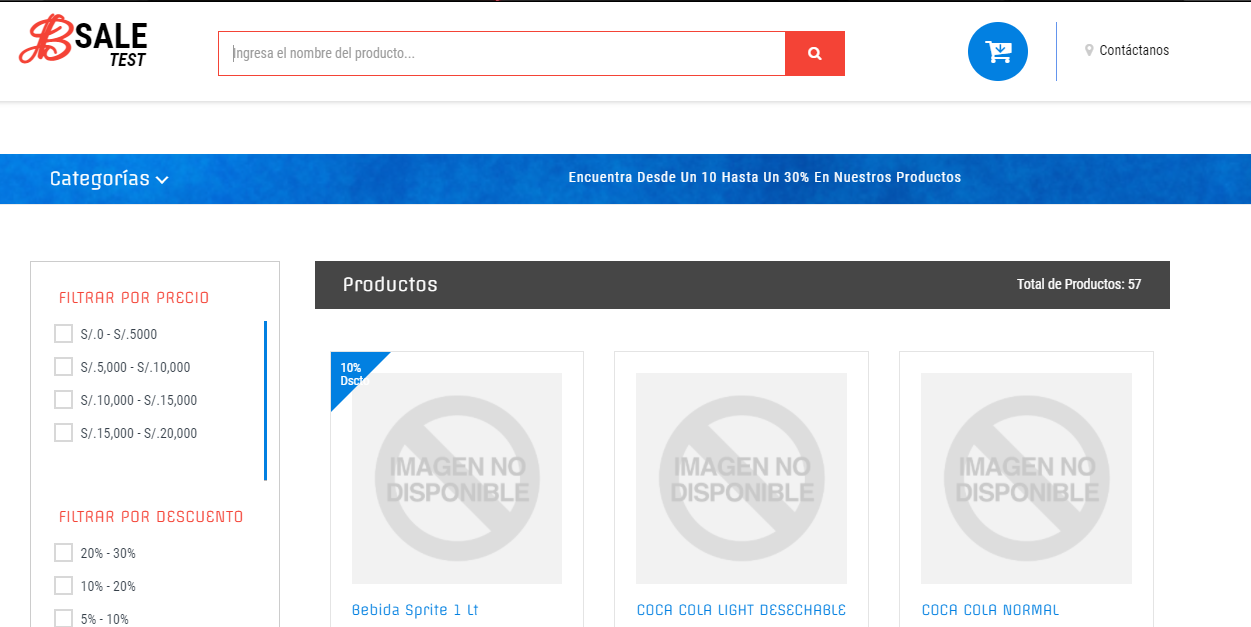
**DOCUMENTACION DE PÁGINA WEB DE PRUEBA BSALE**

Se procederá a realizar la explicación de la página web desarrollada como prueba para la empresa BSALE:

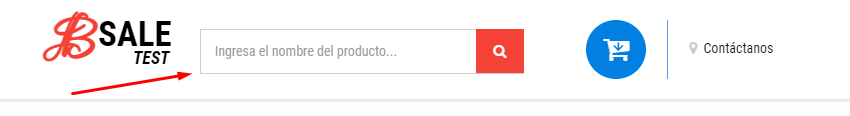
1. **EXPLICACIÓN DEL EJERCICIO**

El ejercicio consiste en la construcción de una tienda online donde se visualice los productos agrupados por categoría y ordenados por nombre.

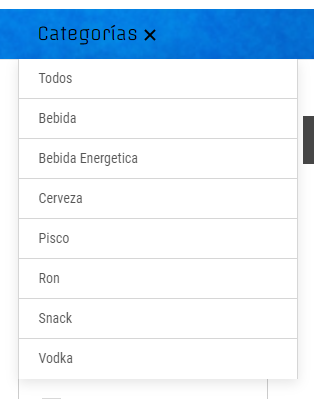


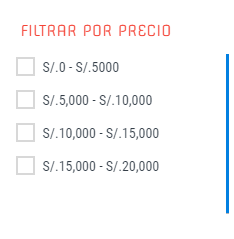
Contiene:

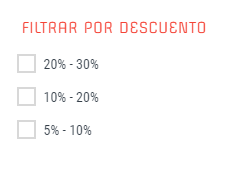
* Buscador: Se puede realizar la búsqueda de los productos por nombre de categoría o nombre del producto.



* Menú Categorías: Se utiliza para obtener las categorías y filtrar los productos por cada una.



* Filtro por Precio: Se utiliza para obtener los productos por intervalos de precios. Son:
* Filtro por Descuento: Se utiliza para obtener los productos por intervalos de porcentaje de descuento. Son:

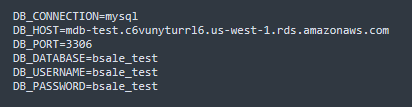


1. **API REST**

Para generar el Api Rest se realizó mediante el framework LARAVEL que utiliza la estructura Modelo – Vista – Controlador (MVC) a través del cual se desarrolló de la siguiente manera:

* Conexión de Base de Datos: La conexión de la base de datos fue realizada en el archivo **“.env”** utilizando parámetros (host, puerto, base de datos, usuario y contraseña) que fueron brindados por la empresa para el desarrollo de éste ejercicio. De ésta forma, se puede acceder a las tablas y campos para la creación, diseño y estructuración de los modelos y controladores generando las funciones correspondientes para el llamado de datos.

Base de Datos: **bsale\_test**



* Creación de Modelos (Producto y Categoria): Se crearon los modelos que asisten a la declaración de los campos de las tablas a utilizar. Es importante incluir todos los campos, ya que serán utilizados en los controladores y sus funciones.

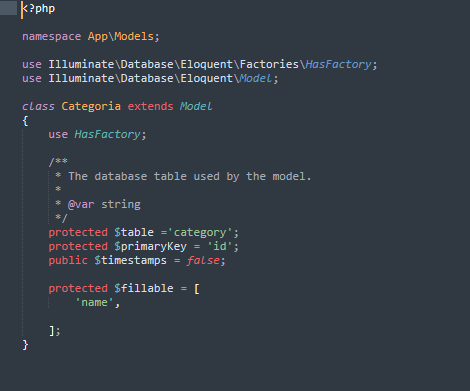
Tablas: **product, category**.

Ubicación de archivos: **app/Models**

**MODELO** **PRODUCTO**



**MODELO** **CATEGORIA**



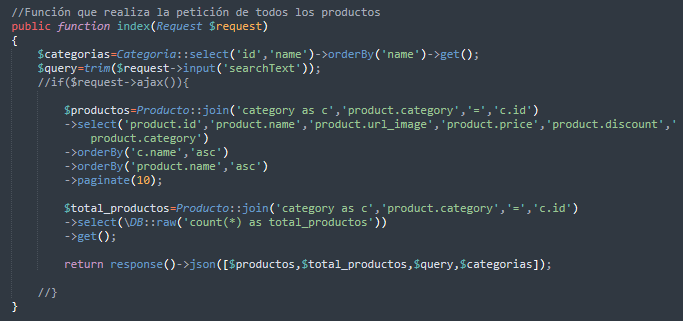
* Creación de Controlador (ProductoController): Una vez creados los modelos, se realiza la creación del controlador donde se generan las funciones que se utilizarán para las peticiones a nivel de vista. Dentro de las funciones se puede realizar el uso de consultas SQL que sirve para obtener los datos a través de peticiones específicas.

Ubicación de archivos: **app/Http/Controllers**.

Nombre de Controlador: **ProductoController**.

Las funciones son:

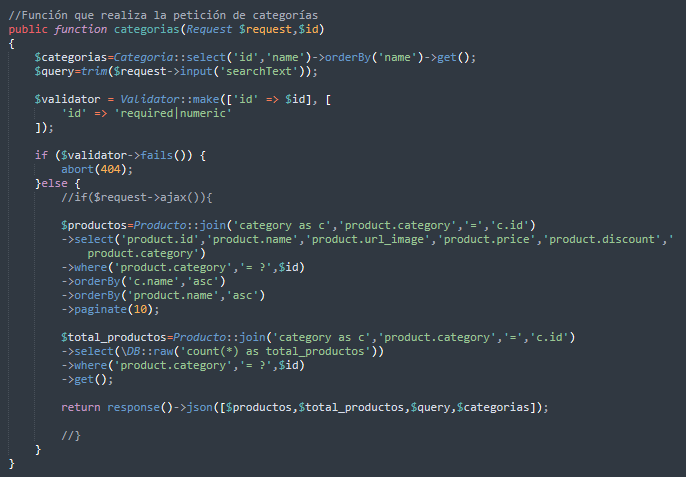
**FUNCION INDEX():**



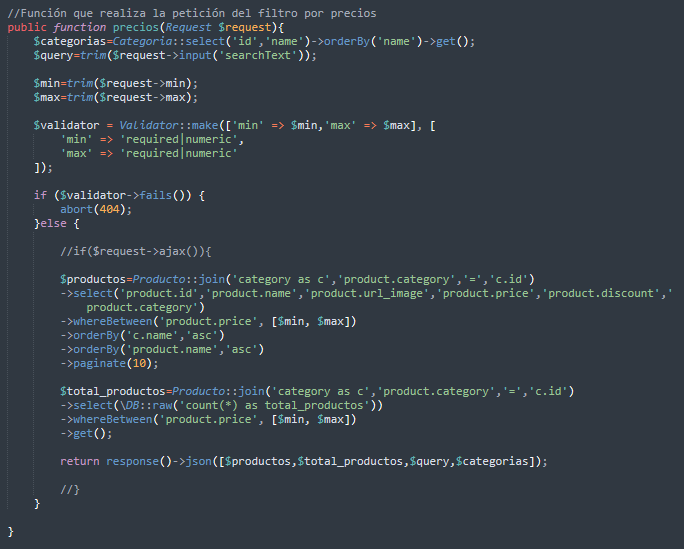
**FUNCIÓN BUSCAR():**



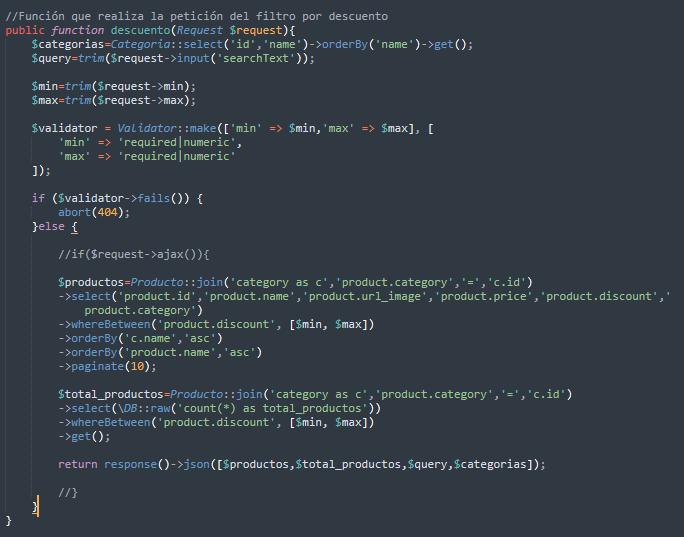
**FUNCIÓN CATEGORIAS():**



**FUNCIÓN PRECIOS():**



**FUNCIÓN DESCUENTO():**



* Creacion de Vista (Welcome): La vista es la interfaz gráfica donde se realiza el diseño e interacción con el usuario final. Se implementa las funciones creadas en el controlador y se define la llamada de éstas mediante rutas generadas para el llamado de datos y peticiones del usuario. Las rutas son:
* PRODUCTOS:

Ruta: **Route::get('productos', 'ProductoController@index');**

Link: **http://localhost:8000/api/productos**

* FILTRO BUSCAR

Ruta: **Route::get('buscar', 'ProductoController@buscar');**

Link local: **http://localhost:8000/api/buscar?searchText=texto**

Texto: Categoria o producto a buscar.

* FILTRO CATEGORIAS

Ruta: **Route::get('categorias/{id}', 'ProductoController@categorias');**

Link: **http://localhost:8000/api/categorias/id**

Id: Llave principal de la fila donde se encuentre la categoria a buscar.

* FILTRO PRECIOS

Ruta: **Route::get('precios', 'ProductoController@precios’);**

Link: **http://localhost:8000/api/precios?min=min&max=max**

Min: Valor mínimo del intervalo de precios del producto

Max: Valor máximo del intervalo de precios del producto

* FILTRO DESCUENTO

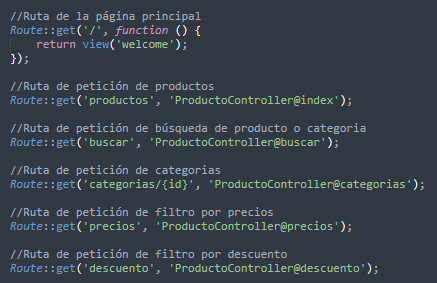
Ruta: **Route::get(‘descuento', 'ProductoController@descuento');**

Link: **http://localhost:8000/api/descuento?min=min&max=max**

Min: Valor mínimo del intervalo de porcentaje de descuento

Max: Valor máximo del intervalo de porcentaje de descuento

Ubicación de archivo: **routes/api.php**



Ubicación de vistas: **resources/views**

Nombre de Vista: **welcome.blade.php**

1. **FRONTEND**

Para la parte de fronteNd (Usuario Final) se creó el archivo BSALE.JS ubicado en “public/js”. Éste archivo fue desarrollado utilizando el lenguaje JavaScript, éste lenguaje es compilado bajo la premisa Just In Time (JIT) o Justo a Tiempo, que permite brindarle la obtención de datos al usuario sin necesidad de generar un tiempo de espera prolongado o recargar la página, apoyado en las líbrerias como JQuery, Bootstrap, Css, entre otros, para dar soporte al diseño e interacción entre la página y el usuario. Dentro de éste archivo se crearon las funciones que sirven para realizar el llamado de datos y mostrarlos al usuario de forma inmediata. Tenemos las siguientes funciones:

* Principal: Mediante ésta función se obtiene el total de productos agrupados por categoría y ordenado por su nombre de forma ascendente.
* Buscar: Filtro que utiliza una caja de texto (input) en el cual se debe ingresar el nombre del producto o nombre de la categoría a buscar. Los resultados que se obtienen son ordenados por nombre del producto y de forma ascendente.
* Categorías: Genera la petición de las categorías para cargarlas en un menú interactivo en el cual realiza el filtro correspondiente por cada una de ellas obteniendo como resultado los productos ordenados por nombre y de forma ascendente.
* Oferta: Ésta función se utiliza para calcular el precio de oferta. La fórmula es:

**PO= PP-(PP\*D/100)**

Donde:

**PO:** Precio Oferta del producto.

**PP:** Precio del producto.

**D:** Porcentaje de Descuento.

* Precios: Filtro utilizado para la obtención de datos de los productos entre intervalos de precios establecidos.
* Descuento: Filtro utilizado para la obtención de datos de los productos entre intervalos de descuento establecidos.

1. **README**

Archivo que contiene la descripción e indicaciones sobre el proyecto realizado.

