Informe técnico ejercicio Laravel creando un módulo para docentes

Desarrollado por: María Cecilia Álvarez Martínez

Presentando a: Carlos Andrés Mora Agudelo

SENA CENTRO DE DISEÑO E INNOVACIÓN

TECNOLOGICA INDUSTRIAL

DOSQUEBRADAS

Versión 1.0

Continuando con el mismo proyecto de academia, pasamos a crear un nuevo módulo para docentes, el cual tendrá la tarea de almacenar los datos de los docentes de la academia. Tales como:

Nombre

Apellido

Título Universitario

Edad

Fecha de contrato

Foto

Documento de identidad

Para iniciar este módulo creamos un nuevo controlador con el siguiente comando:

php artisan make:controller PhotoController -–resource

Se trabaja como ejercicio una aplicación básica para una academia. Se crean los controladores respectivos Docentes.

Una vez creado el controlador para los Docentes, se realizaron los siguientes pasos:

1. Se abre la carpeta http
2. Posteriormente se abre la carpeta controllers que contiene todos los controladores
3. Se abre el controlador llamado: “DocenteController”



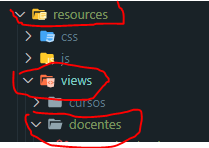
1. Se ingresa al web. Php y se crea la clase con su ruta



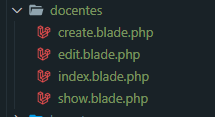
1. Se crea la ruta dentro del mismo web.php



1. Paso seguido para crear la vista, creamos una carpeta con el nombre “docentes” dentro del archivo “views” de la carpeta “resources”

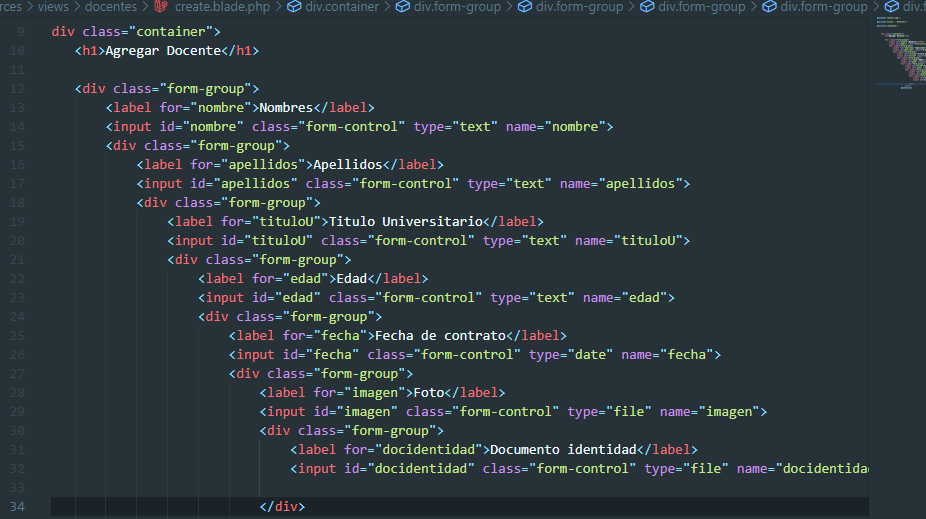


1. Dentro de la carpeta “cursos” creamos los archivos con las extensiones de Laravel sugeridas blade.php los cuales vamos a usar (créate, edit., índex, show)

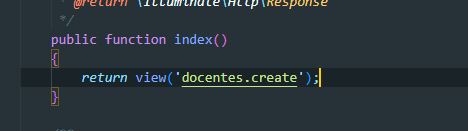


1. Dentro del archivo create, pasamos a crear un formulario

y utilizamos el comando b-form-group ayudados por el pluggin y nos dio la creación de los campos en los que almacenamos los datos del docente.



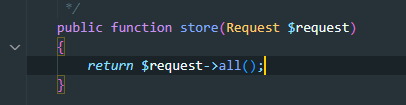
1. En el archivo DocenteController.php creamos el return en el método create:



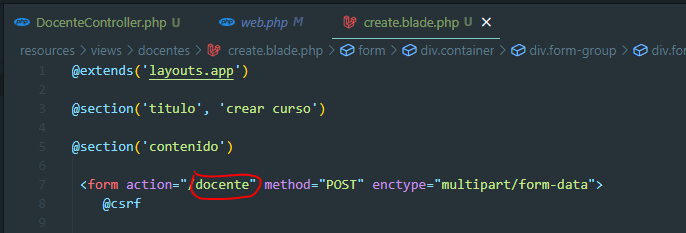
En el navegador cursos/create podremos ver el resultado:



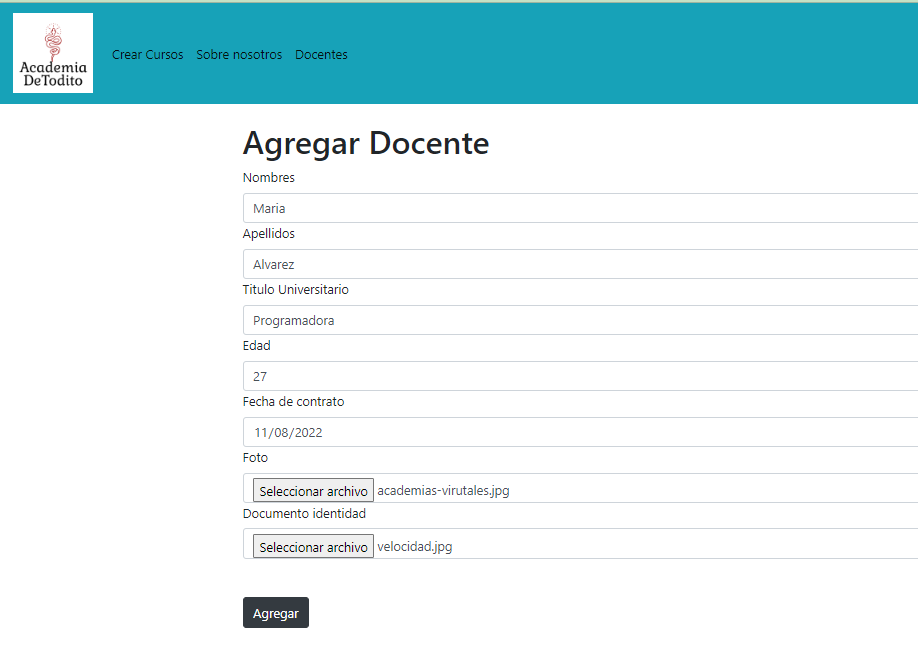
1. Ingresamos al archivo “DocenteController” y creamos el return $request:



1. En el archivo crate.blade.php le damos al formulario una acción /docente



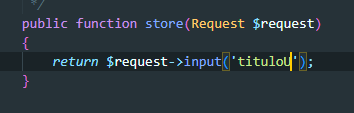
A continuación, con este ejercicio diligenciando los campos del formulario, puede verse el request generado



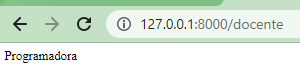


**Comprobando el envío de datos del formulario**

1. Se hizo la siguiente prueba para verificar que se recibe cada dato por separado del formulario:



Se obtuvo el siguiente resultado:



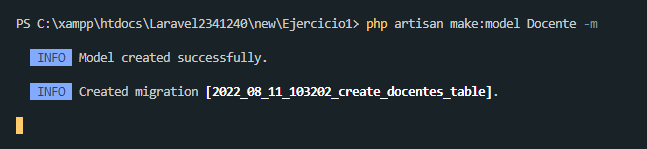
Por lo cual, se deduce que el envió de información se almacenó correctamente en el servidor de Laravel.

**Creación de bases de datos en MysQL**

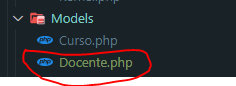
Se siguieron los siguientes pasos:

Usamos la base de datos anteriormente creada academia2

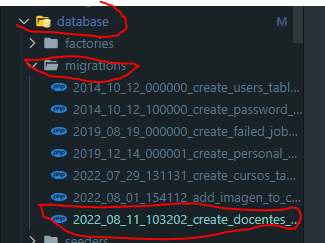
Para la creación de la nueva tabla usamos el comando: php artisan make:model Docente –m

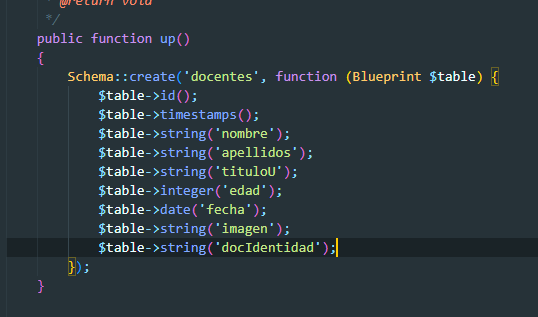


Se aloja en la carpeta modelo como se observa en la siguiente captura.

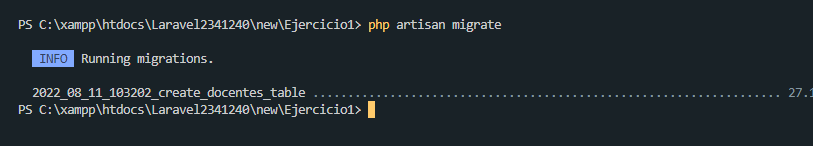


1. Ingresamos a la carpeta “database”, y posteriormente a “migrations” para acceder al modelo archivo que creamos, con anterioridad.



Continuando dentro del archivo del método creado pegamos el código extraído de la documentación por el número de veces que lo requerimos, modificando los campos ‘name’ por ‘Nombre’, ‘apellidos, ‘tituloU. Edad, fecha, imagen y documento de identidad.

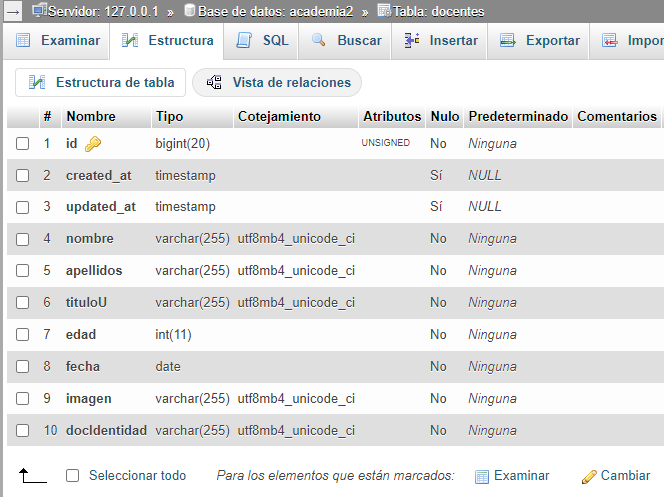
Hacemos la migración con el comando php artisan migrate



De esta manera queda creada la tabla en la base de datos

Una vez lanzado el comando entramos a la base de datos y revisamos su estructura.

Observamos que las columnas estén como deben ser y el tipo de dato como se muestra en la captura.



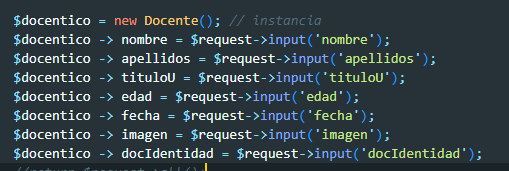
**Trabajando con el “store”**

1. Al volver al VSC, ubicamos la carpeta “http”, luego la de “controllers” y damos clic en “CursoController”.
2. En el método “store” se ingresó las siguientes líneas de código como se muestra en la siguiente captura.



se creó la instancia que se llamó $docentico, el cual actuara como un array, por eso se le agregan nombres de posición, los cuales deben coincidir con los nombres de las columnas de la tabla cursos: nombre, apellidos, TutiloU.

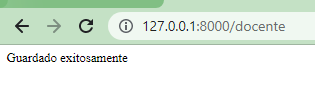
* 1. En cada nombre de posición este array $docentico se almacenó lo que viene desde el formulario. Para eso se usó el igual (=). Se igualo con el input (‘nombrecampoformulario’), como se ve en la siguiente captura.



* 1. Finalmente guaramos la tabla cursos, los datos almacenados en el array instancia $docentico, usando el comando (save) unido con -> a la instancia.



En el negador diligenciamos los campos del formulario y retornó un mensaje ‘Guardado exitosamente’



Se verificó que haya quedado guardado, fuimos a la base de datos y en “examinar” se pudo observar los datos almacenados.

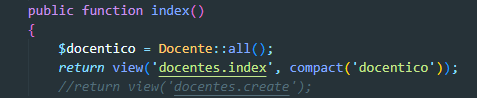


**Programación vista Index**

La vista index permite ver todos los registros que se encuentran en la tabla en cuestión. Para este caso Docentes.

Se siguieron estos pasos:

1. se adjuntó este array instancia a la vista, con el método compact, como se aprecia aquí:

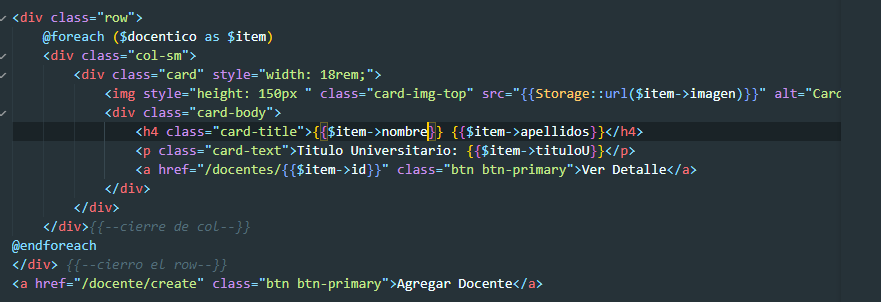


1. Luego en la vista index, se llamó este array, iterándolo mediante @foreach y asignando un alias llamada $item

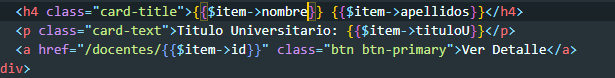


**Añadiendo estilo a la información mostrada en el index**

1. Se optó por usar un card de Bootstrap, según se muestra en la página <https://getbootstrap.com/>
2. Se copió el código respectivo en el index, dentro del foreach. Como se muestra en esta captura.



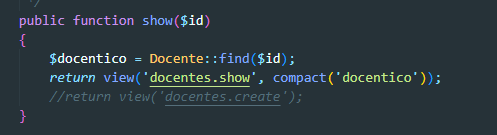
1. Se interpoló la información con el nombre del alias del array, dentro de cada sección del card, así:

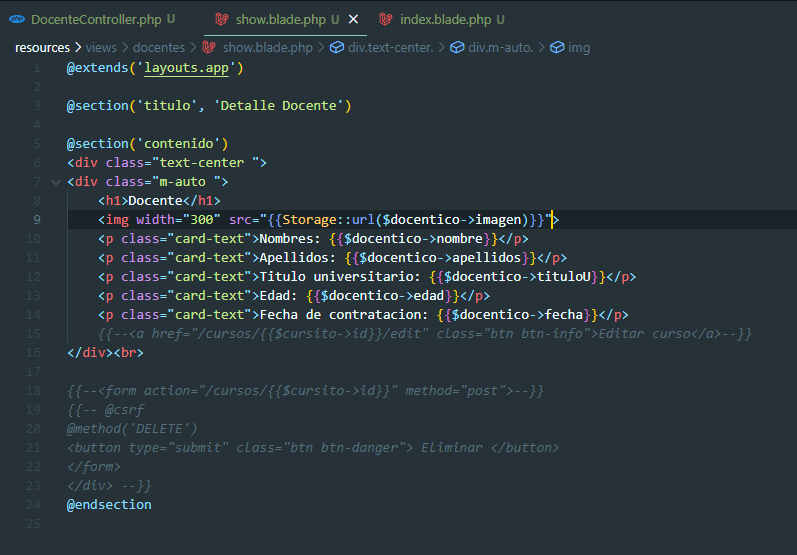


**Programación vista show**

La vista show permite ver todos los detalles que se encuentran en la tabla en cuestión. Para este caso Docentes.

Se siguieron los siguientes pasos:

1. En el archivo DocenteController nos ubicamos en el método de show y creamos las líneas de código como se muestra en la captura: 
2. En el archivo show.blade.php se realizaron las siguientes líneas de código interpolando los datos de la tabla como se muestra en la siguiente captura:



Nos dio el siguiente resultado en el motor de búsqueda

