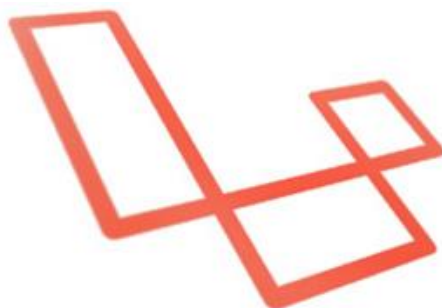


Desarrollo de Aplicaciones Empresariales

LABORATORIO N° 9

Laravel – Funcionalidad del Proyecto III

CODIGO DEL CURSO:



laravel

Alumno(s)		Nota
Grupo		
Ciclo		
Fecha de entrega		

I.- OBJETIVOS:

- Crear funcionalidad para eliminar álbumes y fotos.

II.- SEGURIDAD:**Advertencia:**

En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.

III.- FUNDAMENTO TEÓRICO:

Revise sus diapositivas del tema antes del desarrollo del laboratorio.

IV.- NORMAS EMPLEADAS:

No aplica

V.- RECURSOS:

- En este laboratorio cada alumno trabajará con un equipo con Windows 8.

VI.- METODOLOGÍA PARA EL DESARROLLO DE LA TAREA:

- El desarrollo del laboratorio es individual.

VII.- PROCEDIMIENTO:**Nota:**

Las secciones en cursivas son demostrativas, pero sirven para que usted pueda instalar las herramientas de desarrollo en un equipo externo.

CREANDO LA FUNCIONALIDAD PARA ELIMINAR ÁLBUMES Y FOTOS

Vamos a trabajar con la última parte del CRUD, que viene a ser la eliminación.

Si un álbum va a ser eliminado desde la vista: Mostrar de los álbumes.

¿Será necesario usar un método `getEliminarAlbum`?

1. Diríjase al método `getEliminarAlbum` en el controlador de Álbum y proceda a eliminarlo.
2. Ahora nos centraremos en la función `postEliminarAlbum`.
3. Cree un Request nuevo de nombre: “EliminarAlbumRequest”. Posterior a eso, no se olvide de referenciarlo en el controlador de Álbum.
4. En el Request creado, modifique las funciones `authorize` y `rules`, para que queden de la siguiente manera:

```

1  <?php namespace GestorImagenes\Http\Requests;
2
3  use GestorImagenes\Http\Requests\Request;
4  use Illuminate\Support\Facades\Auth;
5
6  class EliminarAlbumRequest extends Request {
7
8      /**
9       * Determine if the user is authorized to make this request.
10      *
11      * @return bool
12      */
13      public function authorize()
14      {
15          $user=Auth::user();
16          $id=$this->get('id');
17          $album=$user->albumes()->find($id);
18          if($album){
19              return true;
20          }
21          return false;
22      }
23
24      /**
25       * Get the validation rules that apply to the request.
26      *
27      * @return array
28      */
29      public function rules()
30      {
31          return [
32              'id'=>'required|exists:albumes,id'
33          ];
34      }
35
36  }

```

5. Ahora diríjase al controlador de Álbum en su acción “postEliminarAlbum” y modifíquelo de la siguiente manera:

```

59  public function postEliminarAlbum(EliminarAlbumRequest $request){
60
61      $album=Album::find($request->get('id'));
62      $album->fotos()->delete();
63      $album->delete();
64      return redirect('/validado/albumes')->with('eliminado','El album fué eliminado!');
65  }

```

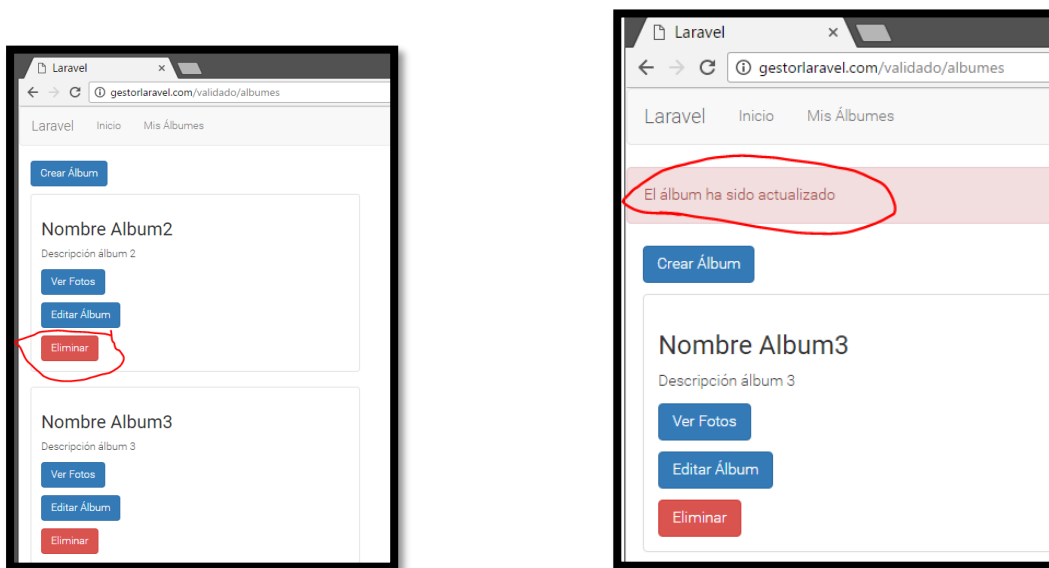
6. Ahora diríjase a la vista de mostrar álbumes, y agregue una porción de código para poner el botón eliminar el álbum:

```

16
17  @if(Session::has('eliminado'))
18      <div class="alert alert-danger">
19          <p>El álbum ha sido actualizado</p>
20      </div>
21  @endif
22
23  <div class="container-fluid">
24      <p><a href="/validado/albumes/crear-album" class="btn btn-primary" role="button">Crear Álbum</a></p>
25      @if(sizeof($albumes)>0)
26          @foreach($albumes as $album)
27              <div class="row">
28                  <div class="col-sm-6 col-md-12">
29                      <div class="thumbnail">
30                          <div class="caption">
31                              <h3>{{ $album->nombre }}</h3>
32                              <p>{{ $album->descripcion }}</p>
33                              <p><a href="/validado/fotos?id={{ $album->id }}" class="btn btn-primary" role="button">Ver Fotos</a></p>
34                              <p><a href="/validado/albumes/actualizar-album/{{ $album->id }}" class="btn btn-primary" role="button">Editar Álbum</a></p>
35                              <form action="/validado/albumes/eliminar-album" method="POST">
36                                  <input type="hidden" name="token" value="{{ csrf_token() }}" required>
37                                  <input type="hidden" name="id" value="{{ $album->id }}" required>
38                                  <input class="btn btn-danger" role="button" type="submit" value="Eliminar">
39                              </form>
40                          </div>
41                      </div>
42                  </div>
43              </div>
44          @endforeach
45      @else
46          <div class="alert alert-danger">

```

Pruebe el programa. Intente eliminar un álbum y verifique los resultados.



Ahora continuaremos con la eliminación de Fotos

1. Por las mismas razones al igual que el controlador de los álbumes, en el controlador de las fotos, elimine la función `getEliminarFoto`.
2. Cree un Request de nombre "EliminarFotoRequest".
3. Referéncielo en el controlador de las fotos.
4. Modifique el request creado, para que quede de la siguiente manera:

```

1  <?php namespace GestorImagenes\Http\Requests;
2
3  use GestorImagenes\Http\Requests\Request;
4  use Illuminate\Support\Facades\Auth;
5  use GestorImagenes\Foto;
6  class EliminarFotoRequest extends Request {
7
8      /**
9       * Determine if the user is authorized to make this request.
10      *
11      * @return bool
12      */
13     public function authorize()
14     {
15         $user=Auth::user();
16         $id=$this->get('id');
17         $foto=Foto::find($id);
18         $album=$user->albumes()->find($foto->album_id);
19
20         if($album){
21             return true;
22         }
23         return false;
24     }
25
26     /**
27      * Get the validation rules that apply to the request.
28      *
29      * @return array
30      */
31     public function rules()
32     {
33         return [
34             'id'=>'required|exists:fotos,id'
35         ];
36     }
37
38 }
39

```

Explique: ¿Qué hacen las dos funciones importantes en este request?

5. Modifique la función `postEliminarFoto`, para que quede de la siguiente manera:

```
80
81     public function postEliminarFoto(EliminarFotoRequest $request){
82         $foto=Foto::find($request->get('id'));
83         $rutaanterior=getcwd().$foto->ruta;
84
85         if (file_exists($rutaanterior)) {
86             unlink(realpath($rutaanterior));
87         }
88         $foto->delete();
89         return redirect("/validado/fotos?id=$foto->album_id")->with('eliminada','La foto fue eliminada');
90     }
```

Investigue:

Cuál es la finalidad de la función `unlink`, `file_exists` y `realpath`.

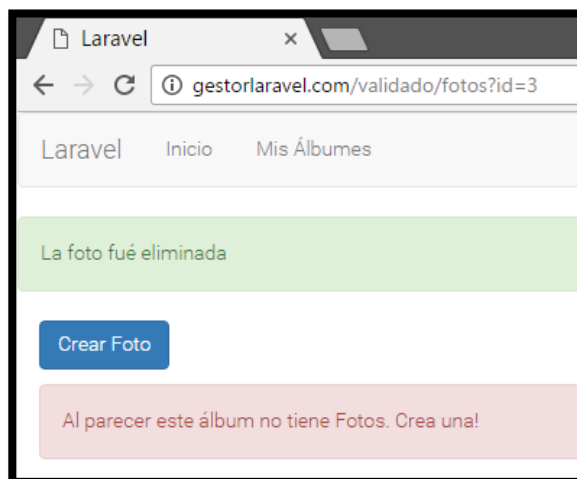
Explique línea a línea la funcionalidad del código mostrado en la función.

Ahora para crear el botón para eliminar una foto en la vista de mostrar fotos:

6. Ingrese a la vista mostrar de Foto, y agregue el siguiente código:

```
8     </div>
9     @endif
10
11     @if(Session::has('eliminada'))
12         <div class="alert alert-success">
13             <p>La foto fué eliminada</p>
14         </div>
15     @endif
16
17     <div class="container-fluid">
18         <p><a href="/validado/fotos/crear-foto?id={{ $id }}" class="btn btn-primary" role="button">Crear Foto</a></p>
19         @if(sizeof($fotos)>0)
20             @foreach($fotos as $foto)
21                 <div class="row">
22                     <div class="col-sm-6 col-md-12">
23                         <div class="thumbnail">
24                             
25                             <div class="caption">
26                                 <h3>{{ $foto->nombre }}</h3>
27                                 <p>{{ $foto->descripcion }}</p>
28                             </div>
29                             <p><a href="/validado/fotos/actualizar-foto/{{ $foto->id }}" class="btn btn-primary" role="button">Editar Foto</a></p>
30                             <form action="/validado/fotos/eliminar-foto" method="POST">
31                                 <input type="hidden" name="_token" value="{{ csrf_token() }}" required>
32                                 <input type="hidden" name="id" value="{{ $foto->id }}" required>
33                                 <input class="btn btn-danger" role="button" type="submit" value="Eliminar">
34                             </form>
35                         </div>
36                     </div>
```

PRUEBE LA ELIMINACIÓN DE FOTOS



This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.