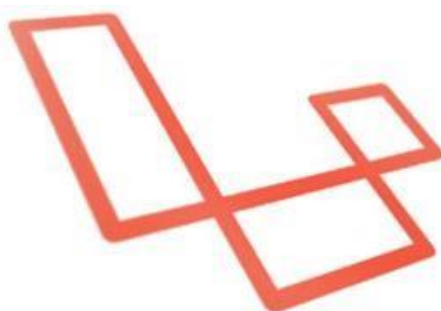


Desarrollo de Aplicaciones Empresariales

LABORATORIO N° 0

Laravel – Controladores y Vistas

CODIGO DEL CURSO:



laravel

<i>Alumno(s)</i>		<i>Nota</i>
Victor Calvo Guzmán		
<i>Grupo</i>	<i>C-15-A</i>	
<i>Ciclo</i>	<i>IV</i>	
<i>Fecha de entrega</i>		

	Desarrollo de Aplicaciones Empresariales – Laravel	Nro. DD-106
		Página 1 de 13

I.- OBJETIVOS:

- Instalar las dependencias en Laravel.
- Instalar Composer.
- Configuración general del IDE de desarrollo para Laravel.

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.

1. En la carpeta **Controllers**, cree un controlador de nombre **FotoController.php**
2. En la carpeta **Controllers**, cree un controlador de nombre **AlbumController.php**
3. En la carpeta **Controllers**, cree un controlador de nombre **UsuarioController.php**
4. Copie el contenido de **HomeController.php** y pegue en el controlador de **Foto**.
5. En la línea 3, modifique el nombre de la clase por **"FotoController"**
6. Ubique la función **getIndex** y modifique su valor de retorno por: **"mostrando fotos del usuario"**.
7. Cree la función **getCrearFoto**, y en su valor de retorno coloque: **"formulario de crear fotos"**.
8. Cree la función **postCrearFoto**, y en su valor de retorno coloque: **"almacenando foto"**.
9. De la misma manera, cree funciones para Actualizar y eliminar fotos: El resultado debería ser como en la imagen siguiente:

```
FotoController.php AlbumController.php ValidacionControlador.php
1 <?php namespace GestorImagenes\Http\Controllers;
2
3 class FotoController extends Controller {
4     /**
5      * Create a new controller instance.
6      *
7      * @return void
8      */
9     public function __construct()
10    {
11        $this->middleware('auth');
12    }
13    public function getIndex(){
14        return 'mostrando fotos del usuario';
15    }
16
17    /**
18     * Show the application dashboard to the user.
19     *
20     * @return Response
21     */
22    public function getCrearFoto()
23    {
24        return 'formulario de crear fotos';
25    }
26    public function postCrearFoto(){
27        return 'almacenando foto';
28    }
29    public function getActualizarFoto()
30    {
31        return 'formulario de actualizar fotos';
32    }
33    public function postActualizarFoto(){
34        return 'actualizar foto';
35    }
36    public function getEliminarFoto()
37    {
38        return 'formulario de eliminar fotos';
39    }
40    public function postEliminarFoto(){
41        return 'eliminando foto';
42    }
43
44 }
45
```

10. Copie el código de **FotoController** y pegue el mismo en **AlbumController**.
11. De manera análoga, el código debería quedar de la siguiente manera:

```
AlbumController.php x FotoController.php ValidationCont
1 <?php namespace GestorImagenes\Http\Controllers;
2
3 class AlbumController extends Controller {
4     /**
5      * Create a new controller instance.
6      *
7      * @return void
8      */
9     public function __construct()
10    {
11        $this->middleware('auth');
12    }
13    public function getIndex(){
14        return 'mostrando Albums del usuario';
15    }
16    /**
17     * Show the application dashboard to the user.
18     *
19     * @return Response
20     */
21    public function getCrearAlbum()
22    {
23        return 'formulario de crear Albums';
24    }
25    public function postCrearAlbum(){
26        return 'almacenando Album';
27    }
28    public function getActualizarAlbum()
29    {
30        return 'formulario de actualizar Album';
31    }
32    public function postActualizarAlbum(){
33        return 'actualizar Album';
34    }
35    public function getEliminarAlbum()
36    {
37        return 'formulario de eliminar Album';
38    }
39    public function postEliminarAlbum(){
40        return 'eliminando Album';
41    }
42
43 }
```

12. Cree el código necesario para el controlador del usuario, de modo que, solo pueda ver y editar su perfil. El código debería quedar de la siguiente manera:

```
UsuarioController.php x AlbumController.php x FotoCont
1 <?php namespace GestorImagenes\Http\Controllers;
2
3 class UsuarioController extends Controller {
4     /**
5      * Create a new controller instance.
6      *
7      * @return void
8      */
9     public function __construct()
10    {
11        $this->middleware('auth');
12    }
13
14    public function getEditarPerfil(){
15        return 'mostrando formulario';
16    }
17    public function postEditarPerfil(){
18        return 'generando actualización de perfil';
19    }
20
21 }
22
```


13. En el controlador de Album, agregue el siguiente método:

```
public function postEliminarAlbum(){  
    return 'eliminando Album';  
}  
public function missingMethod($parameters=array()){  
    abort(404);  
}
```

El método que acabamos de crear sirve para que lanzar el error 404 (notFound), en caso de que el usuario coloque una ruta (ACTION) que inexistente.

14. Copie y pegue la misma función en los controladores de: Validación, Foto, Usuario e Inicio.

FOTO

```
        return 'generando actualización de perfil';  
    }  
    public function missingMethod($parameters=array()){  
        abort(404);  
    }  
}
```

USUARIO

```
3     public function postEditarPerfil(){  
9         return 'generando actualización de perfil';  
9     }  
1    public function missingMethod($parameters=array()){  
2        abort(404);  
3    }  
4 }
```

VALIDACION

```
{  
    $this->auth = $auth;  
    $this->registrar = $registrar;  
  
    $this->middleware('guest', ['except' => 'getLogout']);  
}  
public function missingMethod($parameters=array()){  
    abort(404);  
}
```

INICIO

```
33         return 'página de inicio validado';  
34     }  
35     public function missingMethod($parameters=array()){  
36         abort(404);  
37     }  
38 }  
39 }
```

CREANDO RUTAS EN LARAVEL

1. Abra el archivo routes.php ubicado en el directorio app/http.
2. Modifique el contenido del archivo para que quede de la siguiente manera:

```
Route::controllers(  
[  
    'validacion' => 'Validacion\ValidacionController',  
    'validado/fotos' => 'FotoController',  
    'validado/albumes' => 'AlbumController',  
    'validado/usuario' => 'UsuarioController',  
    'validado' => 'InicioController',  
    '/' => 'BienvenidaController'  
]);
```

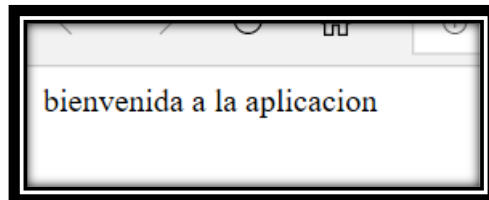
3. Probaremos el Index de la aplicación. Ingrese en la URL, la siguiente dirección:

<http://gestorlaravel.com>

El resultado debería ser:



4. En este caso se hace referencia a la ruta: "/", ya que la carpeta, el controlador que responde es BienvenidaController, y el action por defecto a ejecutarse es: getIndex, ya que la página está siendo cargada por primera vez.



Probando la ruta para **Validado**

1. Abra el archivo: **GestorImagenes/app/Http/Middleware/Authenticate.php**.
2. Ubique la función **handle**.
- 3.

--

```

public function handle($request, Closure $next)
{
    if ($this->auth->guest())
    {
        if ($request->ajax())
        {
            return response('Unauthorized.', 401);
        }
        else
        {
            return redirect()->guest('auth/login');
        }
    }

    return $next($request);
}

```

4. El archivo **Authenticate** se encarga de verificar que el usuario esté autenticado o no.
5. Como hemos cambiado nuestros controladores a castellano, modifique la redirección marcada por: "validacion/inicio".
6. Abra el archivo: **GestorImagenes/app/Http/Middleware/RedirectIfAuthenticated.php**.
7. Este archivo se encarga de enviar a una página de Inicio (autenticado), en caso de que el usuario sea autenticado de manera correcta.

```

public function handle($request, Closure $next)
{
    if ($this->auth->check())
    {
        return new RedirectResponse(url('/home'));
    }

    return $next($request);
}

```

8. Modifique la ruta de home por: "/validado" Ingrese a la siguiente URL: <http://gestorlaravel.com/validado>

El resultado será ser el siguiente:



9. El texto que debería mostrarse es lo que tenemos en la función **INDEX** del controlador **InicioController**, es decir el texto: "página de inicio validado".

```

29  @return response
30  */
31  public function getIndex()
32  {
33      return 'página de inicio validado';
34  }
35  public function missingMethod($parameters=array()){
36      abort(404);
37  }

```

```

Route::controllers(
[
    'validacion' => 'Validacion\ValidacionController',
    'validado/fotos' => 'FotoController',
    'validado/albumes' => 'AlbumController',
    'validado/usuario' => 'UsuarioController',
    'validado' => 'InicioController',
    '/' => 'BienvenidaController'
]);

```



```

33 public function handle($request, Closure $next)
34 {
35     if ($this->auth->guest())
36     {
37         if ($request->ajax())
38         {
39             return response('Unauthorized.', 401);
40         }
41         else
42         {
43             return redirect()->guest('validacion/inicio');
44         }
45     }
46     return $next($request);
47 }
48
49

```

```

85  * @return \Illuminate\Http\Response
86  */
87  public function getInicio()
88  {
89      return "mostrando formulario de inicio de sesión";
90  }
91
92  /**

```

Como se observa la redirección se hizo de manera automática al action “getInicio” del controlador **validación**. Esto se debe a que el usuario no está “**logueado**” correctamente y siempre se le mostrará el formulario de **inicio de sesión** en caso de que la función **handle** del Middleware **Authenticate** no lo valide como tal.

Ingrese a la siguiente URL: <http://gestorlaravel.com/validado/fotos>

El resultado debería ser acceder al controlador
El resultado será el siguiente:



validación en su action **fotos**. Sin embargo y de manera análoga al caso anterior, la redirección se hizo automáticamente a la página del formulario para iniciar sesión, y esto sucederá hasta que le usuario se encuentre validado.

Para omitir el control de validación de usuario (para temas de desarrollo) haremos lo siguiente:

1. Ubique el archivo **GestorImagenes/app/Http/Middleware/Authenticate.php**, y comente el código de validación en la función **handle**, de modo que quede de la siguiente manera:

De esta manera estamos diciendo que, sin

```

32  */
33  public function handle($request, Closure $next)
34  {
35      /*if ($this->auth->guest())
36      {
37          if ($request->ajax())
38          {
39              return response('Unauthorized.', 401);
40          }
41          else
42          {
43              return redirect()->guest('validacion/inicio');
44          }
45      }*/
46
47      return $next($request);
48  }

```

important que el usuario se encuentre “logueado”, se acceda a la petición recibida en los parámetros recibidos.

Ahora intente ingresar a la siguiente URL:

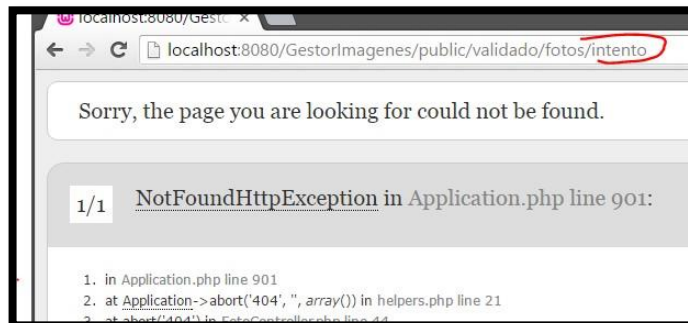
<http://gestorlaravel.com/validado/fotos>



Ahora intente ingresar a la siguiente URL (probaremos otro action del controlador de Foto):



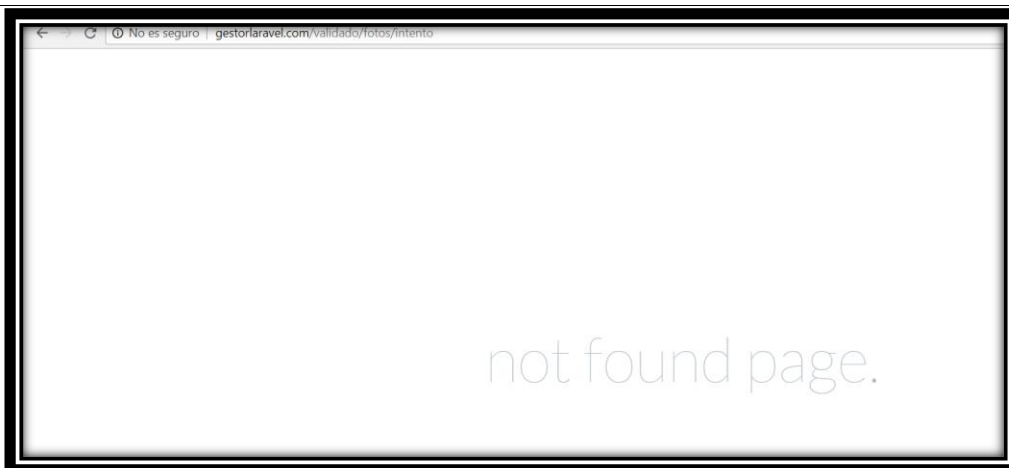
Ahora intente ingresar una URL inexistente.



En efecto se produce una excepción, sin embargo, no es un error **404 (not found page)**, que es lo que tenemos en el controlador de foto en su función **missingMethod**.

1. Haga una copia del archivo **GestorImagenes/Resources/Views/errors/503.blade.php**, en el mismo directorio, pero renombre el nuevo archivo a **"404.blade.php"**.
2. Abra el archivo copiado y cambie el texto **"be right back"** por **"not found page"**
3. Nuevamente, intente introducir una URL inexistente.





CREANDO LAS PRIMERAS VISTAS

1. Ubique el archivo app.blade.php y modifíquelo en las siguientes líneas (cambio de idioma unicamente)

- a. Línea 36 modifique "Home" por "Inicio".
- b. Línea 41 modifique "Login" por "Iniciar Sesión".
- c. Línea 42 modifique "Register" por "Registrarse".
- d. Línea 47 modifique "Logout" por "Salir".

```
class="nav navbar-nav /
<li><a href="/">Inicio</a>
</li>
</li>

<li class="nav navbar-nav
/navbar-right">
@if (Auth::guest())
<li><a href="/auth/
login">Iniciar Sesión</
a></li>
<li><a href="/auth/
register">Registrarse</
a></li>
@else
<li class="dropdown">
<a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button"
aria-expanded="
false">{{
Auth::user()->name
}} <span class="
caret"></span></a>
<ul class="
dropdown-menu" role
="menu">
<li><a href="/
auth/logout">
Salir</a></li>
</ul>
```

pas

2. Ahora modifique las siguientes líneas para modificar los controladores, el archivo debería quedar de la siguiente manera:

```
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav">
    <li><a href="{{ url('/') }}">Inicio</a></li>
  </ul>

  <ul class="nav navbar-nav navbar-right">
    @if (Auth::guest())
      <li><a href="{{ url('/validacion/inicio') }}">Iniciar Sesión</a></li>
      <li><a href="{{ url('/validacion/registro') }}">Registrarse</a></li>
    @else
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
          class="caret"></a>
        <ul class="dropdown-menu" role="menu">
          <li><a href="{{ url('/validacion/salida') }}">Logout</a></li>
        </ul>
      </li>
    @endif
  </ul>
</div>
```

```
<div class="collapse navbar-collapse" id="
bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav">
    <li><a href="{{ url('/') }}">Inicio</a></li>
  </ul>

  <ul class="nav navbar-nav navbar-right">
    @if (Auth::guest())
      <li><a href="{{ url('/validacion/inicio') }}">
        Iniciar Sesión</a></li>
      <li><a href="{{ url('/validacion/registro') }}">
        Registrarse</a></li>
    @else
      <li class="dropdown">
        <a href="#" class="dropdown-toggle"
          data-toggle="dropdown" role="button"
          aria-expanded="false">{{
            Auth::user()->name }} <span class="caret"
          ></span></a>
        <ul class="dropdown-menu" role="menu">
          <li><a href="{{ url('/validacion/
            salida') }}">Salir</a></li>
        </ul>
      </li>
    @endif
  </ul>
</div>
```

- Abra la vista **home.blade.php** y modifique el nombre a **"inicio.blade.php"**
- Abra la vista **welcome.blade.php** y modifique el nombre a **"bienvenida.blade.php"**
- Borre el código de **bienvenida.blade.php**.
- Copie el código de **inicio.blade.php** a **bienvenida.blade.php**
- En **bienvenida.blade.php**, cambie el texto **"Home"** por el texto **"Inicio"**
- En **bienvenida.blade.php**, modifique el texto **"you are logged in"** por **"Por favor inicie sesión para usar el sistema."**
- En **inicio.blade.php**, modifique el texto **"you are logged in"** por **"Bienvenido (usuario)"**, más adelante usaremos el verdadero nombre del usuario logeado.

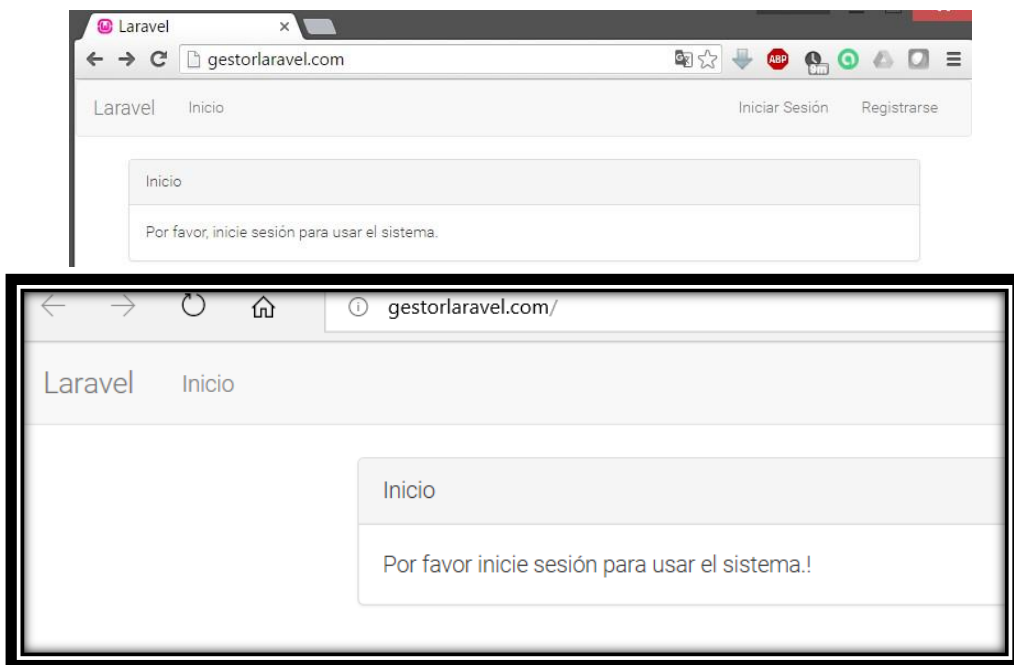
Renderizando las vistas

Hasta ahora, tenemos 3 vistas creadas, aunque 2 funcionales solamente.

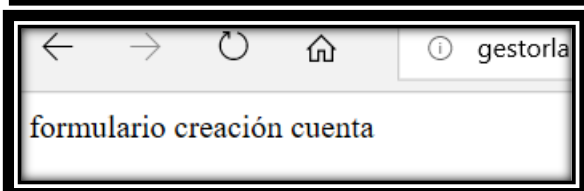
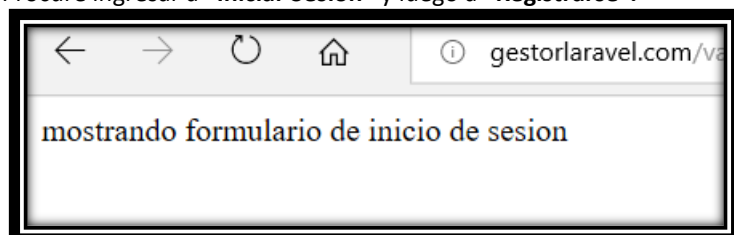
1. Abra **BienvenidaController.php** y `return view('bienvenida');` ubique el action **getIndex()**.
2. Cambie el mensaje de retorno a
3. Abra **InicioController.php** y `return view('inicio');` ubique el action **getIndex()**.
4. Cambie el mensaje de retorno a

Probando la funcionalidad del código.

1. Abra el buscador y coloque la siguiente URL: <http://gestorlaravel.com> El resultado debería ser el siguiente:



2. Procure ingresar a “Iniciar Sesión” y luego a “Registrarse”.





- Abra el archivo **"ValidacionController.php"**.
- Ubique la función **getRegistro()**, y modifique su valor de retorno a: `return view('validacion.registro');`
El "." (punto) se debe a que la vista está en una carpeta llamada **"validacion"**.
- Ubique la función **getInicio()**, y modifique su valor de retorno a: `return view('validacion.inicio');`
El "." (punto) se debe a que la vista está en una carpeta llamada **"validacion"**.

Creando las vistas

- Ingresa a la **sub-carpeta** de la carpeta **VIEWS** de nombre **auth**.
- Modifique la vista de nombre **"login.blade.php"** por **"inicio.blade.php"**.
- Modifique la vista de nombre **"register.blade.php"** por **"registro.blade.php"**.
- Elimine las vistas **"reset.blade.php"** y **"password.blade.php"**.
- Abra el archivo **"inicio.blade.php"** y modifique la **línea 21** como en la figura de abajo, indica que el **action** del formulario para iniciar sesión deberá redireccionar a nuestra página **inicio**.

```

</div>
@endif

<form class="form-horizontal" role="form" method="POST" action="/validacion/inicio">
  <input type="hidden" name="_token" value="{{ csrf_token() }}">

```

- En la línea 51, modifique la palabra **"Login"** por **"Iniciar Sesión"**, líneas abajo modifique el texto **"Forgot your password"** por **"Olvidé mi contraseña"**.
- Modifique el texto de error por **"Al parecer algo está mal"**.
- Modifique también el link de **olvidar contraseña** al action **"validacion/recuperar"**.

```

<div class="panel-heading">Iniciar Sesión</div>
<div class="panel-body">
  @if (count($errors) > 0)
    <div class="alert alert-danger">
      <strong>Whoops!</strong> Al parecer algo está mal.<br>
      <ul>
        @foreach ($errors->all() as $error)
          <li>{{ $error }}</li>
        @endforeach
      </ul>
    </div>
  @endif

```

```

</div>

<div class="form-group">
  <div class="col-md-6 col-md-offset-4">
    <div class="checkbox">
      <input type="checkbox" name="remember"> Recordarme
    </div>
  </div>
</div>

<div class="form-group">
  <div class="col-md-6 col-md-offset-4">
    <button type="submit" class="btn btn-primary" style="margin-right: 15px;">
      Iniciar Sesión
    </button>
    <a href="/validacion/recuperar">Olvidé mi contraseña</a>
  </div>
</div>

```

- De manera análoga en la vista **register.blade.php**, realice los cambios resaltados a continuación:


```

3  @section('content')
4  <div class="container-fluid">
5      <div class="row">
6          <div class="col-md-8 col-md-offset-2">
7              <div class="panel panel-default">
8                  <div class="panel-heading">Registrarse</div>
9                  <div class="panel-body">
10                     @if (count($errors) > 0)
11                         <div class="alert alert-danger">
12                             <strong>Whoops!</strong> Al parecer algo está mal.<br><br>
13                             <ul>
14                                 @foreach ($errors->all() as $error)
15                                     <li>{{ $error }}</li>
16                                 @endforeach
17                             </ul>
18                         </div>
19                     @endif
20
21                     <form class="form-horizontal" role="form" method="POST" action="/validacion/registrad">
22                         <input type="hidden" name="_token" value="{{ csrf_token() }}">
23
24                         <div class="form-group">
25                             <label class="col-md-4 control-label">Nombre</label>
26                             <div class="col-md-6">
27                                 <input type="text" class="form-control" name="nombre" value="{{ old('name') }}">
28                             </div>
29                         </div>
30
31                         <div class="form-group">
32                             <label class="col-md-4 control-label">Correo Electrónico</label>
33                             <div class="col-md-6">
34                                 <input type="email" class="form-control" name="email" value="{{ old('email') }}">
35                             </div>
36                         </div>
37
38                         <div class="form-group">
39                             <label class="col-md-4 control-label">Contraseña</label>
40                             <div class="col-md-6">
41                                 <input type="password" class="form-control" name="password">
42                             </div>
43                         </div>
44
45                         <div class="form-group">
46                             <label class="col-md-4 control-label">Confirm Password</label>
47                             <div class="col-md-6">
48                                 <input type="password" class="form-control" name="password_confirmation">
49                             </div>
50                         </div>
51
52                         <div class="form-group">
53                             <label class="col-md-4 control-label">Pregunta</label>
54                             <div class="col-md-6">
55                                 <input type="text" class="form-control" name="pregunta">
56                             </div>
57                         </div>
58
59                         <div class="form-group">
60                             <label class="col-md-4 control-label">Respuesta</label>
61                             <div class="col-md-6">
62                                 <input type="text" class="form-control" name="respuesta">
63                             </div>
64                         </div>
65
66                         +
67
68                         <div class="form-group">
69                             <div class="col-md-6 col-md-offset-4">
70                                 <button type="submit" class="btn btn-primary">
71                                     Registrarse
72                                 </button>
73                             </div>

```

Aumentar estos campos para que coincidan con el modelo creado para usuario.

10. Ubique la sub-carpeta dentro de Vistas de nombre "Auth" y renombre a: "validacion".
11. En el navegador ponga la siguiente dirección:

<http://gestorlaravel.com/validacion/inicio>

El resultado debería ser el siguiente:

Laravel Inicio Iniciar Sesión Registrarse

Iniciar Sesión

E-Mail Address

Password

☐ Recordarme

Iniciar Sesión Olvidé mi contraseña

Login

E-Mail Address victor.calvo

Password

☒ Recordarme

Iniciar Sesión Olvide mi contraseña

De la misma manera intente entrar al link registrarse, el resultado debería ser el siguiente:

Laravel Inicio Iniciar Sesión Registrarse

Registrarse

Nombre

Correo Electrónico

Contraseña

Confirm Password

Pregunta

Respuesta

Registrarse

Registrarse

Nombre Victor

Correo Electrónico victor.calvo

Contraseña

Confirm Password

Pregunta ..

Respuesta ..

Registrarse

Creando Vista para recuperar la contraseña

1. Cree una nueva vista en la subcarpeta “**validacion**” de nombre: “**recuperar.blade.php**”.
2. Copie el código de la vista “**registro**” a la vista “**recuperar**” y haga las modificaciones indicadas en la imagen a continuación:

```

3 @section('content')
4 <div class="container-fluid">
5   <div class="row">
6     <div class="col-md-8 col-md-offset-2">
7       <div class="panel panel-default">
8         <div class="panel-heading">Recuperar Contraseña</div>
9         <div class="panel-body">
10           @if (count($errors) > 0)
11             <div class="alert alert-danger">
12               <strong>Whoops!</strong> Al parecer algo está mal.<br><br>
13               <ul>
14                 @foreach ($errors->all() as $error)
15                   <li>{{ $error }}</li>
16                 @endforeach
17               </ul>
18             </div>
19           @endif
20
21           <form class="form-horizontal" role="form" method="POST" action="/validacion/recuperar">
22             <input type="hidden" name="_token" value="{{ csrf_token() }}">
23
24
25             <div class="form-group">
26               <label class="col-md-4 control-label">Correo Electrónico</label>
27               <div class="col-md-6">
28                 <input type="email" class="form-control" name="email" value="{{ old('email') }}">
29               </div>
30             </div>
31
32             <div class="form-group">
33               <label class="col-md-4 control-label">Nueva Contraseña</label>
34               <div class="col-md-6">
35                 <input type="password" class="form-control" name="password">
36               </div>
37             </div>
38
39             <div class="form-group">
40               <label class="col-md-4 control-label">Repita la contraseña</label>
41               <div class="col-md-6">
42                 <input type="password" class="form-control" name="password_confirmation">
43               </div>
44             </div>
45
46             <div class="form-group">
47               <label class="col-md-4 control-label">Pregunta</label>
48               <div class="col-md-6">
49                 <input type="text" class="form-control" name="pregunta">
50               </div>
51             </div>
52
53             <div class="form-group">
54               <label class="col-md-4 control-label">Respuesta</label>
55               <div class="col-md-6">
56                 <input type="text" class="form-control" name="respuesta">
57               </div>
58             </div>
59
60             <div class="form-group">
61               <div class="col-md-6 col-md-offset-4">
62                 <button type="submit" class="btn btn-primary">
63                   Recuperar Contraseña
64                 </button>
65               </div>
66             </div>

```

3. Abra el
modifique su

```
return view('validacion.recuperar');
```

controlador de “Validacion”, busque la acción **getRecuperar()** y
valor de retorno por:

Recuperar Contraseña

Nueva Contraseña

Repita la contraseña

Pregunta

Respuesta

Recuperar Contraseña

OBSERVACIONES

- Vamos a crear diferentes tipos de controladores ya que vamos a ver diferentes tipos de peticiones.
- Al trata de correr el programa, se ha producido errores ya que no se escribe bien la ruta de la pag web o está mal llamada en el código ya sea por un espacio, una letra, mayúsculas o un simple "\".
- Nos fijamos que en los controladores vamos a tener la función si se pierde el método nos va a mandar el error 404.
- Vamos a utilizar un autenticado, el cual va a verificar si está el usuario autenticado o no, en este punto se ha tenido problemas ya que se debe tener en cuenta el cambio de las palabras resaltadas en el laboratorio.
- En el laboratorio en la parte de la validación, el link que nos indica está mal, ya que nos lleva a otra dirección porque faltaba al inicio el gestorlaravel.

CONCLUSIONES

- Vamos a crear controladores para poder realizar el mecanismo para agrupar la lógica de la petición.
- En la carpeta routes, debemos tener mucho cuidado, ya que si cambiamos o editamos o está mal escrito alguna ruta no va a correr al momento de probar el programa.
- Vamos a cambiar los textos en idioma español para que se entienda más los cambios que se hacen además por el cambio de idioma único que nos indica en el laboratorio.
- Después de que nos vaya a validar de que podemos entrar al inicio de sesión y también a la creación de una cuenta vamos a interactuar con su contenido el cual va a ser muy parecido al del inicio, ya que de ahí sacamos los datos para estos dos.
- Con el punto de recuperar contraseña vamos a acceder a él mediante la página inicio, al costado del botón antes de ingresar, vamos a observar que se acede mediante una frase resaltada y no de una que está en una esquina o en un botón.

