**CLASE FS**

**MAILS AUTOMÁTICOS**

* Configurar los datos en el archivo .env

*MAIL\_DRIVER=smtp*

*MAIL\_HOST=smtp.gmail.com*

*MAIL\_PORT=465*

*MAIL\_USERNAME=algo@gmail.com*

*MAIL\_PASSWORD=clavegmail*

*MAIL\_ENCRYPTION=ssl*

* Crear mail en la ventana de comandos: *php artisan make:mail UsuarioCreado*
* Nos crea en app/Mail/UsuarioCreado.php y en la función build es dónde se genera el mail

*class UsuarioCreado extends Mailable{*

*use Queueable, SerializesModels;*

*public $email;*

*public $nombre;*

*public $pass;*

*public function \_\_construct($nombre,$email,$pass) {*

*$this->email = $email;*

*$this->nombre = $nombre;*

*$this->pass = $pass;*

*}*

*public function build() {*

*return $this->from('medicon@masterweb.la')*

*->subject('Se ha registrado en Medicon')*

*->view('mails.usuario');*

*}*

*}*

* Crear la vista mails y allí vamos a crear factura.blade.php (poner un HTML básico)

*<h1>Registro Exitoso</h1>*

*<p>Hola {{$nombre}}, ha sido registrado en Medicon con estos datos: </p>*

*<ul>*

*<li>Email: {{$email}}</li>*

*<li>Password: {{$pass}}</li>*

*</ul>*

*<p>Gracias</p>*

* En UsuariosController.php vamos a poner inicialmente para probar, en index
  + Incluimos la librería: *use App\Mail\FacturaCreada;*
  + Incluimos el Facade: *use Illuminate\Support\Facades\Mail y Auth;*
  + Enviamos el mail:

$email = $request->email;

$nombre = $request->name;

$pass = $request->password;

Mail::to($email)->send(new UsuarioCreado($nombre,$email,$pass));

**REPORTES EN PDF**

Librería Dompdf: <https://github.com/barryvdh/laravel-dompdf>

Ruta: *Route::get('pdf',['as'=>'pdf', 'uses'=>'PdfController@generarpdf']);*

Controlador Prueba simple:

*public function generarpdf(){*

*$pdf = \App::make('dompdf.wrapper');*

*$pdf->loadHTML('<h1>Test</h1><p>Hola, soy un PDF</p>');*

return $pdf->stream('factura');

*}*

Controlador prueba 2 con vista

*public function generarpdf(){*

*$facturas = Factura::all();*

*$pdf = \App::make('dompdf.wrapper');*

*$view = \View::make('pdf.facturas', compact('facturas'))->render();*

*$pdf->loadHTML($view);*

*return $pdf->stream('facturas');*

*}*

Si queremos que el documento se descargue usamos:

*return $pdf->download(facturas.pdf');*

En las vistas, creamos una carpeta pdf y vamos a montar la vista

*<h1>Listado de Facturas</h1>*

*@foreach ($facturas as $factura)*

*{{$factura->cliente->nombre}}<br>*

*{{$factura->numero}}<br>*

*${{number\_format($factura->valor)}}*

*<hr>*

*@endforeach*

**EXPORTAR EXCEL**

<http://www.maatwebsite.nl/laravel-excel/docs/getting-started>

* En composer.json colocamos: "maatwebsite/excel": "~2.1.0"
* Ejecutamos en la consola: composer update
* Incluimos los providers y los aliassess
* Ejecutamos en la consola: *php artisan vendor:publish --provider="Maatwebsite\Excel\ExcelServiceProvider"*

**Controlador:**

*php artisan make:controller ExcelController*

*<?php*

*use Maatwebsite\Excel\Facades\Excel;*

*use App\Cliente;*

*class ExcelController extends Controller{*

*public function index(){*

*Excel::create('Clientes', function($excel) {*

*$excel->sheet('Clientes', function($sheet) {*

*$clientes = Cliente::all();*

*$sheet->fromArray($clientes);*

*});*

*})->export('xlsx');*

*}*

*}*

**Ruta**

*Route::resource('excel','ExcelController');*

**Vista (en clientes.index)**

*<div class="text-center">*

*<a class="btn btn-info" href="{{route('excel.index')}}" target="blank">Exporta a Excel</a>*

*</div>*

**IMPORTAR EXCEL**

**Ruta:** *Route::get('importar',['as'=>'importar','uses'=>'ImportarController@importar']);*

**Controlador**

*<?php*

*use Maatwebsite\Excel\Facades\Excel;*

*use App\Cliente;*

*class ImportarController extends Controller*

*{*

*public function importar(){*

*Excel::load('clientes8.xlsx', function($reader) {*

*foreach ($reader->get() as $cliente) {*

*Cliente::create([*

*'nombre' => $cliente->nombre,*

*'nit' => $cliente->nit,*

*'direccion' => $cliente->direccion,*

*'telefono' => $cliente->telefono,*

*'email' => $cliente->email,*

*'ciudad' => $cliente->ciudad,*

*]);*

*}*

*});*

*return Cliente::all();*

*}*

*}*

**MIDDLEWARES**

Los middleware proporcionan un mecanismo conveniente para filtrar peticiones HTTP que ingresan a la aplicación.

La finalidad de este componente es disminuir la carga de trabajo en los controladores y proporcionar una solución mucho más simple y estándar a la hora de aplicar las restricciones necesarias en el proyecto. Los middleware te permiten agregar restricciones por zona horaria, geolocalización, login, edad, genero, rol o tipo de usuario, y básicamente todo lo que quieras. Esto te permite bloquear acceso a ciertas áreas de tu aplicación para un usuario o un grupo de usuarios en específico

Por ejemplo, Laravel incluye un middleware que verifica que el usuario de tu aplicación está autenticado. Si el usuario no está autenticado, el middleware redirigirá al usuario a la pantalla de inicio de sesión. Sin embargo, si el usuario es autenticado, el middleware permitirá que la solicitud continúe en la aplicación.

**Validar autenticación**

Middleware En la ruta:

*Route::get('/', function () { return view('index'); })->middleware('auth');*

Middleware en el Controller

*public function \_\_construct(){*

*$this->middleware('auth');*

*}*

Variación con una execepción en una función del Controlador

*public function \_\_construct(){*

*$this->middleware('auth',['except'=>'index']);*

*}*

**Validar roles de usuario:**

\* Loguerme – desloguearme y explicar que los ítems de menús no se están mostrando si no estoy logueado, pero puedo entrar a las rutas sin problema, entonces la idea es proteger la ruta y validar que a algunas de esas rutas solo puedan ingresar los usuarios tipo admin

Comando para crear Middleware: *php artisan make:middleware Admin*

Se ha creado **Admin.php** en el directorio app/Http/Middleware

*public function handle($request, Closure $next) {*

*if ($request->user()->idperfil != 1) {*

*return redirect()->back()->with('mensaje','No tiene permisos');*

*}*

*return $next($request);*

*}*

En protected $routeMiddleware de app/Http **Kernel.php** ingresamos:

'admin' => \App\Http\Middleware\Admin::class,

En Facturas Controller ponemos:

*public function \_\_construct(){*

*$this->middleware('admin',['except'=>'index']);*

*}*

*Variación*

*$this->middleware(['auth','admin'],['except'=>'index']);*

*Variación*

*$this->middleware('auth');*

*$this->middleware('admin',['except'=>'index']);*