\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Banner**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Tipo: Integración de modulo.***

***Requeridos Composer:***

$ composer require yajra/laravel-datatables-oracle:"~9.0"

**Service Provider & Facade (Optional on Laravel 5.5+)**

Register provider and facade on your config/app.php file.

'providers' => [

...,

Yajra\DataTables\DataTablesServiceProvider::class,

]

'aliases' => [

...,

'DataTables' => Yajra\DataTables\Facades\DataTables::class,

]

**Configuration (Optional)**

$ php artisan vendor:publish --provider="Yajra\DataTables\DataTablesServiceProvider"

**Archivos creados y agregados**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*IMPORTANTE: CSS y JS copiar y pegar carpetas css\_files y js\_files en public**

**\*IMPORTANTE: Copiar y pegar carpeta banner en resources\views\staff\**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. banner Controller en la ruta **App\Controllers\Staff\BannerController**

<?php

namespace App\Http\Controllers\Staff;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

use Illuminate\Support\Facades\Storage;

use App\Http\Controllers\Controller;

use App\Models\Banner;

use Yajra\Datatables\Datatables;

use Auth;

use App\Models\User;

class BannerController extends Controller

{

    protected $table;

    function \_\_construct(){

        $this->table = new Banner();

    }

        /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

        return view('Staff.banner.index');

    }

    /\*\*

     \* Show the form for creating a new resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function create()

    {

        return view('Staff.banner.create');

    }

    /\*\*

     \* DataTable POST

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return \Illuminate\Http\Response

    \*/

    public function dataTable\_store(Request $request)

    {

        $table = Datatables::of($this->table::all());

        $table->addColumn('action', function($row){

            return view('Staff.banner.datatables.config',['option'=>'buttons','id'=>$row['id\_banner']]);

        })->addColumn('status', function($row){

            return view('Staff.banner.datatables.config',['option'=>'enabled','data'=>$row['enabled'],'id'=>$row['id\_banner']]);

        })->addColumn('image', function($row){

            return view('Staff.banner.datatables.config',['option'=>'img','data'=>$row['image']]);

        })->rawColumns(['action','status','image']);

        return $table->make(true);

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @return \Illuminate\Http\Response

     \*/

    public function store(Request $request)

    {

        $validator = Validator::make($request->all(),[

            'title' => 'required',

            'image' => 'required',

            'priority' => 'required|numeric',

        ]);

        if ($validator->fails()) {

            return response()->json(['error'=>$validator->errors()->all()]);

        }else{

            $allowedfileExtension=['jpg','png']; // Extensiones de imagenes permitidas

            $data\_all = $request->except(['\_token']);

            if ($request->hasFile('image')) {

                if (!is\_dir(public\_path().'/storage')) {

                    mkdir(public\_path().'/storage');

                }

                if(!Storage::disk('public\_uploads')->has('banners')){

                    Storage::disk('public\_uploads')->makeDirectory('banners');

                }

                $file = $request->file('image');

                $name = $file->store('banners',['disk' => 'public\_uploads']);

                $data\_all['image'] = $name;

            }

            $data\_all['enabled'] = (!empty($data\_all['enabled'])) ? $data\_all['enabled'] : 0 ;

            $this->table->fill($data\_all)->save();

            $request->session()->flash('create', 'Registro agregado con exito!');

            return response()->json(['success'=>'Registro agregado con exito','url'=>url("dashboard/banners")]);

        }

    }

    /\*\*

     \* Display the specified resource.

     \*

     \* @param  int  $id

     \* @return \Illuminate\Http\Response

     \*/

    public function show($id)

    {

        //

    }

    /\*\*

     \* Show the form for editing the specified resource.

     \*

     \* @param  int  $id

     \* @return \Illuminate\Http\Response

     \*/

    public function edit($id)

    {

        $data = $this->table::find($id);

        return view('Staff.banner.edit',['item'=>$data]);

    }

    /\*\*

     \* Update the specified resource in storage.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  int  $id

     \* @return \Illuminate\Http\Response

     \*/

    public function update(Request $request, $id)

    {

        if($request->has('action')){

            $data\_all = $request->all();

            switch ($data\_all['action']) {

                case 'status':

                    $row = $this->table::find($id);

                    if(!empty($row)){

                        $status = ($row['enabled']== 1) ? 0 : 1;

                        $row->enabled = $status;

                        $row->save();

                    }

                    return response()->json(['success'=>'Status cambiado','tipe'=>$status]);

                break;

                case 'update\_item':

                    $validator = Validator::make($request->all(),[

                        'title' => 'required',

                        'priority' => 'required|numeric',

                    ]);

                    if ($validator->fails()) {

                        return response()->json(['error'=>$validator->errors()->all()]);

                    }else{

                        $item = $this->table::find($id); // Obtenemos el registro de la base de datos para actualizar

                        $allowedfileExtension=['jpg','png']; // Extensiones de imagenes permitidas

                        $data\_all = $request->except(['\_token']);

                        if ($request->hasFile('image')) {

                            if (!is\_dir(public\_path().'/storage')) {

                                mkdir(public\_path().'/storage');

                            }

                            if (Storage::disk('public\_uploads')->exists($item->image)) {

                                Storage::disk('public\_uploads')->delete($item->image);

                            }

                            $file = $request->file('image');

                            $name = $file->store('banners',['disk' => 'public\_uploads']);

                            $data\_all['image'] = $name;

                        }else{

                            $data\_all['image'] = $item->image;

                        }

                        $data\_all['enabled'] = (!empty($data\_all['enabled'])) ? $data\_all['enabled'] : 0 ;

                        $item->fill($data\_all)->save();

                        $request->session()->flash('create', 'Registro Actualizado con exito!');

                        return response()->json(['success'=>'Registro Actualizado con exito','url'=>url("dashboard/banners")]);

                    }

                break;

            }

        }

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*

     \* @param  int  $id

     \* @return \Illuminate\Http\Response

     \*/

    public function destroy($id)

    {

        $item = $this->table::find($id);

        if (!empty($item)) {

            if (Storage::disk('public\_uploads')->exists($item->image)) {

                Storage::disk('public\_uploads')->delete($item->image);

            }

            $item->delete();

            return response()->json(['success'=>'Registro eliminado con exito']);

        }

    }

}

**2.** banner Model en la ruta **App\Models\Banner**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Banner extends Model

{

    use HasFactory;

    protected $primaryKey = 'id\_banner';

    protected $fillable = [

        'title', 'description', 'image','priority','link','enabled'

    ];

    public function get\_rand\_banners(){

        $get\_id = $this->priority\_rand();

        $get = $this::find($get\_id);

        return $get;

    }

    public function priority\_rand(){

        $nElementos = 1;

        $arr = array();

        $priority = array(

            1 => 70,

            1 => 70,

            1 => 70,

            2 => 30,

            3 => 35,

        );

        while (count($arr) < $nElementos) {

            $n = rand(1,3);

            $p = rand(1,100);

            if ($p < $priority[$n]) $arr[] = $n;

        }

        $r = $arr[0];

        if ($r == 1) {

            $sql = $this::where('enabled','1')->orderBy('priority','DESC')->limit(3)->get();

        }else if ($r == 2) {

            $b = $this::count();

            if ($b <= 4) {

                $c = 0;

            }else{

                $b = $b-3;

                $c = ceil($b/2);

            }

            $sql = $this::where('enabled','1')->orderBy('priority','ASC')->offset($c)->limit($b)->get();

        }else if ($r == 3) {

            $b = $this::count();

            $c = ceil($b/2);

            $sql = $this::where('enabled','1')->orderBy('priority','ASC')->offset(0)->limit($c)->get();

        }

        if(count($sql) > 0){

            $rad = $this->rand\_array($sql);

            $order = array\_rand($rad);

            return $rad[$order]->id;

        }

    }

    public static function rand\_array($sql){

        $rad = array();

        foreach ($sql as $key =>  $c) {

            $rad[] = (Object)array('priority'=>$c->priority,'id'=>$c->id\_banner);

        }

        return $rad;

    }

}

**3.** Migración de banner en la ruta **Database\Migrations\2020\_11\_06\_141031\_banners.php**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class Banners extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('banners', function (Blueprint $table) {

            $table->bigIncrements('id\_banner');

        $table->string('title',80);

            $table->text('description')->nullable();

            $table->text('image',150);

            $table->integer('priority');

            $table->string('link')->nullable();

            $table->integer('enabled');

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('banners');

    }

}

**Archivos modificados:**

1. Modificado AppServiceProvider.php en **App\Providers\AppServiceProvider.php**

Código usado y requeridos en la modificación

use App\Models\Banner;

use App\Models\Article;

use Auth;

Código en **public function boot()**

        //banners rand and notice function

        view()->composer('\*', function(View $view){

            //section banners and notice

            $banner = new Banner();

            $banners = $banner->get\_rand\_banners();

            $url\_excerpt = url()->current();

            $url\_parts = explode('/',$url\_excerpt);

            $article\_read = array();

            if (Auth::check()){

                $user = User::find(Auth::id());

                if (in\_array('users', $url\_parts) || in\_array('create', $url\_parts) || in\_array('edit', $url\_parts) || $user->group->name == 'SuperUser') {

                    $banners = '';

                }

                //article sections

                if(!in\_array('dashboard',$url\_parts)){

                    $article = new Article();

                    //$article\_read = $article->get\_article\_important(Auth::id());

                }

            }

            $view->with('banners\_rand',$banners)->with('article\_read',$article\_read);

        });

        //Fin

2. Modificada la vista general de layout para agregar el banner en **resources\layout \default.blade.php**

  @include('partials.banner\_and\_notice')

3. Scripts para banner y noticias en **resources\layout\default.blade.php**

       <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

        <!--\*\*\*\*Javascript para datatables\*\*\*\*\*-->

        <script type="text/javascript" src="{{  asset('js\_files/plugins/datatables.min.js')  }}" crossorigin="anonymous"></script>

        <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

        <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

        <!--\*\*\*\*Javascript para datatables\*\*\*\*\*-->

        <script type="text/javascript" src="{{  asset('js\_files/plugins/fileinput.min.js')  }}" ></script>

        <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

        <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

        <!--\*\*\*\*Funcion de ajax para efectos\*\*-->

        <script type="text/javascript" src="{{  asset('js\_files/views\_functions/ajax\_functions.js')  }}" ></script>

        <!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

        @include('partials.banner\_and\_notice\_scripts')

# 4. Agregar opcion al menu de dashboard en resources/views/partials/dashboardmenu.blade.php

            <li>

                <a href="{{ route('staff.banners.index') }}">

                    <i class="{{ config('other.font-awesome') }} fa-newspaper"></i> Banners

                </a>

            </li>

# 5. Agregar estilos en el header resources/views/partials/head.blade.php

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<!--\*\*\*\*Estilos para datatables\*\*\*\*\*-->

<link rel="stylesheet" href="{{ asset('css\_files/plugins/datatables.min.css') }}" crossorigin="anonymous">

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<!--\*\*\*\*Estilos para datatables\*\*\*\*\*-->

<link rel="stylesheet" href="{{ asset('css\_files/plugins/fileinput.min.css') }}" crossorigin="anonymous">

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<!--\*\*\*\*Estilos para ajaxs\*\*\*\*\*-->

<link rel="stylesheet" href="{{ asset('css\_files/plugins/css\_content.css') }}" crossorigin="anonymous">

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fin de banners\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Noticias**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Tipo: Integración de modulo.***

**Creación de nuevos archivos**

1. Copiar y pegar el archivo en la ruta correspondiente **App/Helpers/PaginationHelper.php**

2. Copiar y pegar los siguientes archivos en sus rutas correspondientes:

# resources/views/article/article\_list\_read.blade.php

# resources/views/partials/modals\_read\_article.blade.php

# resources/views/vendor/pagination/bootstrap\_jquery.blade.php

* resources/views/partials/banner\_and\_notice.blade.php
* resources/views/partials/ banner\_and\_notice\_scripts.blade.php

**Modificaciones**

2. Modificar y agregar algunos utilidades a utilizar en **App/ Http/Controllers/Staff/ArticleController.php**

**Contenido de ArticleController.php**

use Auth;

use DB;

use Route;

**Dentro de la clase agregar la variable $table\_pivot**

class ArticleController extends Controller

{

    protected $table\_pivot = 'users\_articles\_required'; //Agregamos esta variable protegida

**Agregamos las siguientes funciones:**

    /\*\*

     \* List Article from ajax

     \*

     \* @return view

    \*/

    public function read\_article\_list(){

        $article = new Article();

        $route\_name =explode('.',Route::currentRouteName());

        $data = $article->get\_article\_important(Auth::id(),$route\_name[0]);

        return \view('article.article\_list\_read',['article\_read\_json'=>$data]);

    }

    /\*\*

     \* Check read article.

     \*

     \* @param \Illuminate\Http\Request $request

     \*

     \* @return Illuminate\Http\RedirectResponse

    \*/

    public function read\_article(Request $request){

        $data\_all = $request->all();

        $find = DB::table($this->table\_pivot)

        ->where('articles\_id',$data\_all['id'])

        ->where('users\_id',Auth::id())->first();

        if(!empty($find)){

            DB::table($this->table\_pivot)

            ->where('articles\_id', $data\_all['id'])

            ->where('users\_id',Auth::id())

            ->update([

                'read\_article' => 1,

                'fech\_article' => now()

            ]);

        }else{

            DB::table($this->table\_pivot)->insert([

                ['articles\_id' => $data\_all['id'],

                 'users\_id' => Auth::id(),

                 'read\_article'=>'1',

                 'fech\_article'=>now()

                ],

            ]);

        }

    }

Dentro de la función **public function store** agregamos el siguiente código

        $article->read\_required = (!empty($request->input('read\_required'))) ? $request->input('read\_required') : 0 ;

Dentro de la función **public function update** agregamos el siguiente código

        $article->read\_required = (!empty($request->input('read\_required'))) ? $request->input('read\_required') : 0 ;

        //colocamos los articulos leidos por el usuario en 0 nuevamente

        DB::table($this->table\_pivot)

        ->where('articles\_id', $id)

        ->update([

            'read\_article' => 0

        ]);

Dentro de la función **public function destroy** agregamos el siguiente código

        //Eliminamos también el contenido de la tabla pivot

        DB::table($this->table\_pivot)->where('articles\_id', $id)->delete();

3. Modificar y agregar algunos utilidades a utilizar en **App/ Http/Models/Article.php**

**Contenido de ArticleController.php**

use App\Helpers\PaginationHelper;

use DB;

**Dentro de la clase agregar la variable $table\_pivot**

class Article extends Model

{

    use HasFactory;

    use Auditable;

    protected $table\_pivot = "users\_articles\_required";

**Agregar las siguientes funciones**

       /\*\*

     \* Status articles important

     \*

     \* @return data

    \*/

    public function get\_article\_important($user\_id,$route=''){

        $data = $this::where('read\_required','1')->get();

        $result = $this->returnCollection($data,$user\_id);

        if($route == 'read\_article\_list'){

            return PaginationHelper::paginate($result,count($result),10,'');

        }else{

            return PaginationHelper::paginate($result,count($result),10,1);

        }

    }

    public function returnCollection($data,$user\_id){

        $rr = collect();

        foreach ($data as $c) {

            $db = DB::table($this->table\_pivot)

                    ->where('articles\_id',$c->id)

                    ->where('users\_id',$user\_id)

                    ->where('read\_article',1)->first();

            if (empty($db)) {

                $rr[] = $c;

            }

        }

        return $rr;

    }

Agregar la columna **read\_required - Tabla = articles**

* En el archivo **database/migrations/2017\_12\_10\_020753\_create\_articles\_table.php Agregar**

  $table->boolean('read\_required');

Agregar la siguiente migración con **php artisan make:migration users\_articles\_required o copiar y pegar lo siguiente y renombrar**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class UsersArticlesRequired extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('users\_articles\_required', function (Blueprint $table) {

            $table->increments('id');

            $table->integer('users\_id');

            $table->integer('articles\_id');

            $table->boolean('read\_article');

            $table->dateTime('fech\_article');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('users\_articles\_required');

    }

}

**\*IMPORTANTE: Una vez agregada la migración correr el comando php artisan migrate para crear la tabla y sus campos**

4. Modificar el archivo **resources/views/Staff/article/create.blade.php** para poder seleccionar si un articulo será y aparecerá en noticias o no

            <div class="row">

                <div class="col-lg-10">

                    <div class="form-group">

                        <label for="title">@lang('common.title')</label>

                        <label>

                            <input type="text" class="form-control" name="title" required>

                        </label>

                    </div>

                </div>

                <div class="col-sm-6 col-md-2 col-lg-2 text-center">

                    <label for="enlacee">Important article</label>

                    <b>Enabled :</b> <input type="checkbox" name="read\_required" id="check\_enabled\_box" value="1" >

                </div>

            </div>

5. Modificar el archivo **resources/views/Staff/article/edit.blade.php** para poder seleccionar si un articulo será y aparecerá en noticias o no

    <div class="row">

                <div class="col-lg-10">

                    <div class="form-group">

                        <label for="title">@lang('common.title')</label>

                        <label>

                            <input type="text" class="form-control" name="title" value="{{$article->title}}" required >

                        </label>

                    </div>

                </div>

                <div class="col-sm-6 col-md-2 col-lg-2 text-center">

                    <label for="enlacee">Important article</label>

                    <b>Enabled :</b> <input type="checkbox"  name="read\_required" id="check\_enabled\_box" value="1" {{($article->read\_required == 1) ? 'checked' : ''}} >

                </div>

            </div>

6. Agregar las rutas para leer y paginar **routes/web.php**

 /\*

    |---------------------------------------------------------------------------------

    | Route For notice read and list pagination

    |---------------------------------------------------------------------------------

    \*/

    Route::group(['middleware' => ['auth','twostep', 'banned']], function (){

        Route::post('/read\_article', 'Staff\ArticleController@read\_article')->name('read\_article');

        Route::get('read\_article\_list', 'Staff\ArticleController@read\_article\_list')->name('read\_article\_list');

    });