



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**MANUAL TÉCNICO**

**TEMA:**

**“SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE  
PARQUEADEROS PRIVADOS UTILIZANDO EL FRAMEWORK PHONEGAP Y  
METODOLOGÍA DE DESARROLLO MOBILE-D”**

**AUTOR:**

Diego Manuel Pérez Carvajal

**DIRECTOR:**

Ing. Fernando Garrido, Msc.

**Ibarra – Ecuador**

**2016**

# Tabla de contenido

1.	EL Proyecto.....	4
2.	Herramientas para el desarrollo .....	4
2.1.	IDE NetBeans.....	4
2.2.	Editor de texto Atom.....	4
2.3.	Navicat.....	4
2.4.	JDK.....	4
2.5.	PhoneGap.....	4
2.6.	Cordova .....	4
2.7.	Android SDK.....	4
2.8.	Laravel .....	4
2.9.	Google Chrome .....	5
2.10.	MySql.....	5
2.11.	PhpMyAdmin.....	5
2.12.	Servidor web local .....	5
2.13.	Servidor de producción .....	5
2.14.	Computador .....	5
2.15.	Smarphone .....	5
3.	Diagrama Entidad Relación .....	5
3.1.	Definición de Tablas .....	7
4.	Estructura del Proyecto Laravel .....	12
4.1.	Models - Modelos.....	12
4.2.	Views - Vistas.....	13
4.3.	Middleware .....	16
4.4.	Controllors – Controladores.....	17
4.5.	Routes – Rutas.....	19
4.6.	Request – Peticiones .....	20
4.7.	Directorio Public.....	21
4.8.	Composer.json.....	21

# Tabla de Ilustraciones

Ilustración 1: Estructura MVC del proyecto .....	12
Ilustración 2: Directorio Public.....	21

## 1. EL Proyecto

EL proyecto “SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE PARQUEADEROS PRIVADOS UTILIZANDO EL FRAMEWORK PHONEGAP Y METODOLOGÍA DE DESARROLLO MOBILE-D”, consta de dos partes, la primera se enfoca en el desarrollo de una aplicación web que gestione y administre los servicios de estacionamiento, además tiene herramientas para la configuración de horarios y tarifas, gráficos de estado, control de usuarios, y la segunda parte es la aplicación móvil híbrida que consume la información de la aplicación web y la plasma en un mapa.

## 2. Herramientas para el desarrollo

### 2.1. IDE NetBeans

Se utilizó el entorno de desarrollo integrado NetBeans en su versión 8.0.1.

### 2.2. Editor de texto Atom

Atom es un editor de texto y código, para OS X, Linux, Windows, se utilizó la versión 1.6.0.

### 2.3. Navicat

Navicat es un modelador de base de datos, versión 11.0.8.

### 2.4. JDK

EL Java Developer Kit en sus versiones 7.0 y 8.0\_60

### 2.5. PhoneGap

PhoneGap es el framework intermediario entre el lenguaje HTML y el lenguaje nativo de las aplicaciones móviles, se utilizó la versión 5.4.1.

### 2.6. Cordova

Cordova al igual que PhoneGap son intermediarios entre el lenguaje web y el lenguaje nativo de las aplicaciones móviles, Cordova cuenta con repositorio de plugins más completo y actualizado. EL proyecto inicial fue Cordova, PhoneGap se basa en Cordova. Versión 6.0.0

### 2.7. Android SDK

EL kit de desarrollo de Android:

- Android SDK tolos 24.4.1
- Android API 23
- Android API 22
- Android API 21
- Android API 20
- Android API 19

### 2.8. Laravel

EL framework Laravel para el diseño MVC de la aplicación web, versión 5.0.

### 2.9. Google Chrome

Se utilizó las herramientas para desarrollo del navegador Google Chrome, versión 49.0.2623.112 m.

### 2.10. MySQL

EL gestor de base de datos MySQL versión 5.6.24 - MySQL Community Server (GPL).

### 2.11. PhpMyAdmin

Para la gestión de la base de datos se utilizó PhpMyAdmin 4.3.11.

### 2.12. Servidor web local

Se utilizó un servidor web local Apache/2.4.12 (Win32) OpenSSL/1.0.1l PHP/5.5.24.

### 2.13. Servidor de producción

Se utilizó el servicio PAAS de Openshift, small gear, con soporte para el framework Laravel, php 5.4 y base de datos MySQL 5.5, especificaciones de hardware: 1 cpu, 512 mb en ram y 1 gb de almacenamiento.

### 2.14. Computador

Nombre del SO: Microsoft Windows 10 Pro

Versión: 10.0.10586 compilación 10586

Fabricante del SO: Microsoft Corporation

Nombre del sistema: DESKTOP-BD8C42C

Fabricante del sistema: TOSHIBA

Modelo del sistema: Satellite L635

Tipo de sistema PC basado en x64

Procesador: Intel(R) Core(TM) i3 CPU M 350 @ 2.27GHz, 2266 Mhz, 2 procesadores principales, 4 procesadores lógicos

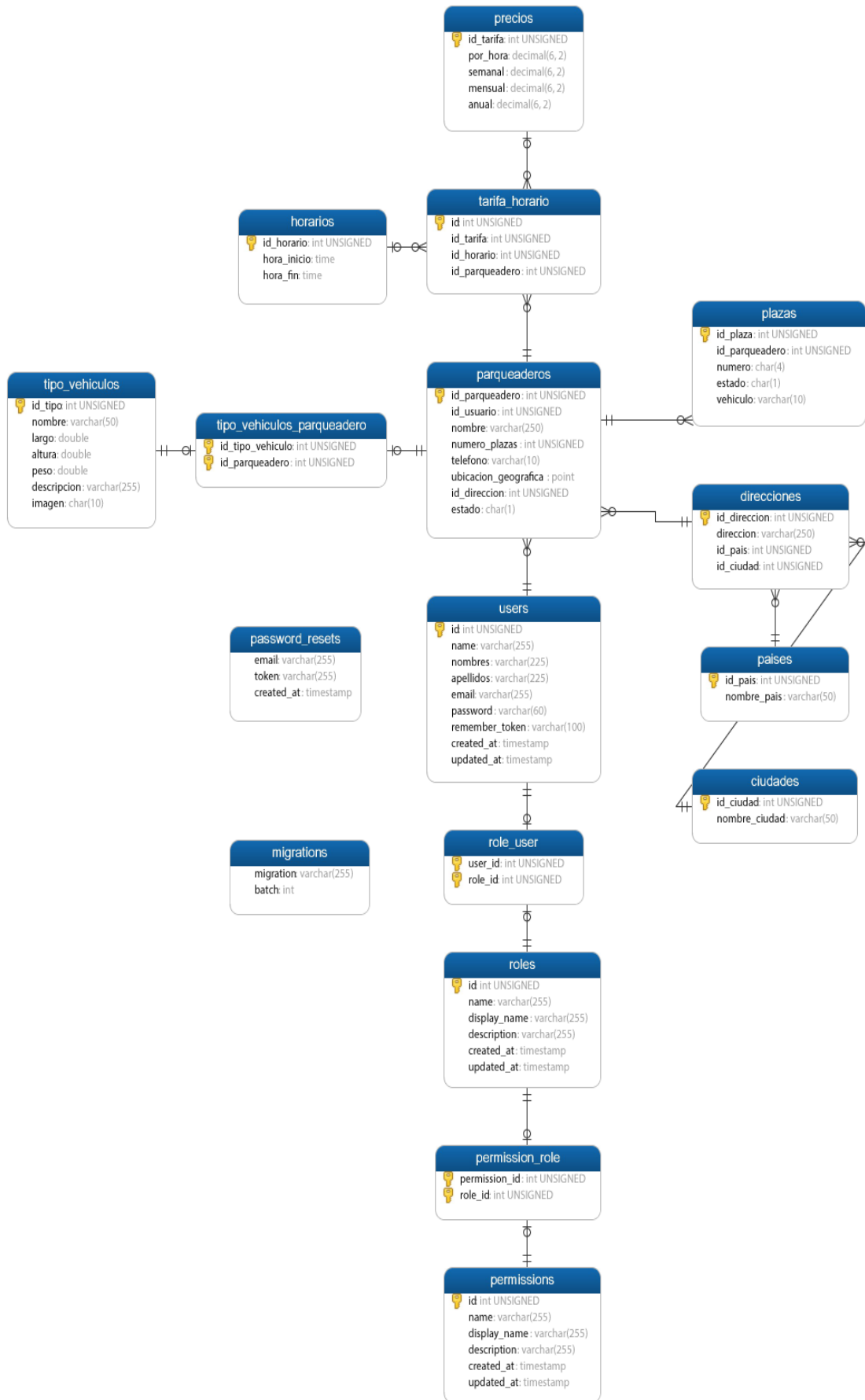
Ram: 4 GB

Disco duro: SSD

### 2.15. Smartphone

Motorola moto g, 8GB/16GB memoria interna, 1GB RAM, procesador Qualcomm Snapdragon 400 quad-core 1.2 GHz, GPU Adreno 305, Android lollipop 5.1.

## 3. Diagrama Entidad Relación



### 3.1. Definición de Tablas

Lista de Tablas:

- 1 ciudades
- 2 direcciones
- 3 horarios
- 4 migrations
- 5 paises
- 6 parqueaderos
- 7 password\_resets
- 8 permissions
- 9 permission\_role
- 10 plazas
- 11 precios
- 12 roles
- 13 role\_user
- 14 tarifa\_horario
- 15 tipo\_vehiculos
- 16 tipo\_vehiculos\_parqueadero
- 17 users

---

#### ciudades

Field	Type	Extra
P id_ciudad	int(10) UNSIGNED COMMENT 'Clave única de registro'	Auto I
increment		
nombre_ciudad	varchar(50) COMMENT 'Nombre de ciudad'	Allow Null

---

#### direcciones

Field	Type	Extra
P id_direccion	int(10) UNSIGNED COMMENT 'Clave única de registro'	Auto
Increment		
direccion	varchar(250) COMMENT 'Dirección del parqueadero'	
id_pais	int(10) UNSIGNED COMMENT 'Clave foranea de paises'	
id_ciudad	int(10) UNSIGNED COMMENT 'Clave foranea de ciudades'	

---

Index	Fields	Extra
fk_direcciones_paises_1	id_pais	
fk_direcciones_ciudades_1	id_ciudad	

---

### horarios

Field	Type	Extra
P id_horario	int(10) UNSIGNED COMMENT 'Clave única de registro'	Auto Increment
hora_inicio	time COMMENT 'hora de apertura del servicio'	Allow Null
hora_fin	time COMMENT 'hora de cierre del servicio'	Allow Null

---

Index	Fields	Extra
fk_horarios_horas_1	hora_inicio	
fk_horarios_horas_2	hora_fin	

---

### migrations

Field	Type	Extra
migration	varchar(255)	
batch	int(11)	

---

### países

Field	Type	Extra
P id_pais	int(10) UNSIGNED COMMENT 'Clave única de registro'	Auto Increment
nombre_pais	varchar(50) COMMENT 'Nombre de país'	Allow Null

---

### parqueaderos

Field	Type	Extra
P id_parqueadero	int(10) UNSIGNED COMMENT 'Clave única de registro'	Auto Increment
id_usuario	int(10) UNSIGNED	
nombre	varchar(250) COMMENT 'Nombre del parqueadero'	
numero_plazas	int(4) UNSIGNED COMMENT 'Número de plazas del parqueadero'	
telefono	varchar(10) COMMENT 'Teléfono del parqueadero'	Allow Null
ubicacion_geografica	point COMMENT 'Ubicación geográfica del parqueadero'	
id_direccion	int(10) UNSIGNED	
estado	char(1) COMMENT 'Estado del parqueadero, 1 : Activo, 0: Fuera de servicio'	

---

Index	Fields	Extra
fk_direcciones	id_direccion	
fk_usuario	id_usuario	

---



### password\_resets

Field	Type	Extra
email	varchar(255)	
token	varchar(255)	
created_at	timestamp	

---

Index	Fields	Extra
password_resets_email_index	email	
password_resets_token_index	token	

1

### permissions

	Field	Type	Extra
P	id	int(10) UNSIGNED	Auto Increment
	name	varchar(255)	
	display_name	varchar(255)	Allow Null
	description	varchar(255)	Allow Null
	created_at	timestamp	
	updated_at	timestamp	

---

Index	Fields	Extra
permissions_name_unique	name	Unique

### permission\_role

	Field	Type	Extra
P	permission_id	int(10) UNSIGNED	
P	role_id	int(10) UNSIGNED	

---

Index	Fields	Extra
permission_role_role_id_foreign	role_id	

### plazas

	Field	Type	Extra
P	id_plaza	int(10) UNSIGNED	COMMENT 'Clave única' Auto Increment
	id_parqueadero	int(10) UNSIGNED	COMMENT 'Clave foranea de parqueaderos'
	numero	char(4)	COMMENT 'Número de la plaza que se muestra en la i nterfaz de usuario'
	estado	char(1)	COMMENT 'Identifica el estado de la plaza, D (disp onible), O (ocupada), M (matenimiento)'
	vehiculo	varchar(10)	COMMENT 'Registra un identificador del vehícul o que ocupa la plaza.' Allow Null

---

Index	Fields	Extra
-------	--------	-------

fk\_parqueaderos\_plazas id\_parqueadero

### precios

Field	Type	Extra
P id_tarifa	int(10) UNSIGNED COMMENT 'Clave única de registro'	Auto Increment
por_hora	decimal(6,2) COMMENT 'Precio por hora'	Allow Null
semanal	decimal(6,2) COMMENT 'Precio semanal'	Allow Null
mensual	decimal(6,2) COMMENT 'Precio mensual'	Allow Null
anual	decimal(6,2) COMMENT 'Precio anual'	Allow Null

### roles

Field	Type	Extra
P id	int(10) UNSIGNED	Auto Increment
name	varchar(255)	
display_name	varchar(255)	Allow Null
description	varchar(255)	Allow Null
created_at	timestamp	
updated_at	timestamp	

Index	Fields	Extra
roles_name_unique	name	Unique

### role\_user

Field	Type	Extra
P user_id	int(10) UNSIGNED	
P role_id	int(10) UNSIGNED	

Index	Fields	Extra
role_user_role_id_foreign	role_id	

2

### tarifa\_horario

Field	Type	Extra
P id	int(10) UNSIGNED COMMENT 'Clave única de registro de tarifas - horarios'	Auto Increment
id_tarifa	int(10) UNSIGNED COMMENT 'Clave foranea de Tarifas'	Allow Null
id_horario	int(10) UNSIGNED COMMENT 'Clave foranea de Horarios'	Allow Null
id_parqueadero	int(10) UNSIGNED COMMENT 'Clave foranea de Horarios'	

Index	Fields	Extra
id_tarifa	id_tarifa	
id_horario	id_horario	

id\_parqueadero id\_parqueadero

### tipo\_vehiculos

Field	Type	Extra
P id_tipo	int(10) UNSIGNED	COMMENT 'Clave única de registro'
Auto Increment		
nombre	varchar(50)	COMMENT 'nombre del tipo de vehículo'
Allow Null		
largo	double(6,2)	COMMENT 'Largo máximo de un vehículo'
altura	double(6,2)	COMMENT 'Altura del vehículo'
peso	double(6,2)	COMMENT 'Peso del vehículo en toneladas'
descripcion	varchar(255)	COMMENT 'Descripción del tipo de vehiculo'
Allow Null		
imagen	char(10)	COMMENT 'Imagen que representa el tipo de vehículo'

### tipo\_vehiculos\_parqueadero

Field	Type	Extra
P id_tipo_vehiculo	int(10) UNSIGNED	COMMENT 'Clave foranea de tipo de vehiculos'
P id_parqueadero	int(10) UNSIGNED	COMMENT 'Clave foranea de parqueaderos'

Index	Fields	Extra
fk_tipo_vehiculos_parqueadero_parqueaderos_1	id_parqueadero	

### users

Field	Type	Extra
P id	int(10) UNSIGNED	Auto Increment
name	varchar(255)	
nombres	varchar(225)	
apellidos	varchar(225)	
email	varchar(255)	
password	varchar(60)	
remember_token	varchar(100)	Allow Null
created_at	timestamp	
updated_at	timestamp	

Index	Fields	Extra
users_email_unique	email	Unique

## 4. Estructura del Proyecto Laravel

La estructura del proyecto tiene un diseño Modelo Vista Controlador MVC, y está compuesto por los siguientes directorios:

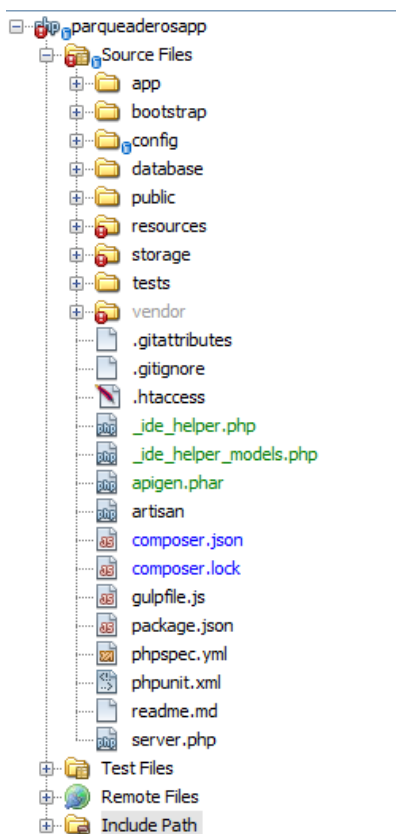


Ilustración 1: Estructura MVC del proyecto

### 4.1. Models - Modelos

Este es un modelo base del proyecto, para este modelo se utiliza la tabla Ciudades. Para representar un modelo de la base de datos se hereda la estructura de la clase base Model.

El parámetro \$table identifica en nombre de la base de datos.

El parámetro \$primaryKey identifica la clave primaria de la tabla.

El parámetro \$fillable identifica a las columnas sobrantes de la tabla.

\$timestamp, esta parámetro se establece por defecto, guardando en la base de datos la fecha en la que se crea el registro.

direcciones(), esta función es la relación con la tabla direcciones.

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```

class Ciudad extends Model {
    protected $table = 'ciudades';
    protected $primaryKey = 'id_ciudad';
    protected $fillable = ['nombre_ciudad'];
    public $timestamps = false;

    public function direcciones() {
        return $this->hasMany('App\Direccion', 'id_direccion');
    }
}

```

## 4.2. Views - Vistas

Son las pantallas usadas para mostrar datos además de la interacción con el usuario. Para este proyecto se ha creado una página maestra que contiene el menú e identificadores de sección para el relleno de datos.

Secciones. Se establecieron secciones:

- Para el título de la página 'title'
- Para inserción de scripts en la cabecera de la página, con nombre 'linktop'.
- Para el contenido de la página 'content'
- Para inserción de scripts en el pie de la página, con nombre 'linktop'.

### Página Maestra

```

<!DOCTYPE html>
<html lang="es">
    <head>
        <link rel="shortcut icon" type="image/x-icon" href="{{
asset('/img/p.ico') }}" />
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <title>@section('title')PARQUEANDO @show</title>
        <link href="{{ asset('/css/app.css') }}" rel="stylesheet" />
        <link href="//fonts.googleapis.com/css?family=Roboto:400,300"
rel='stylesheet' type='text/css' />
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scri
pt>
        <script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scr
ipt>

        @section('linktop')@show
        <script>

            $(document).ready(function () {

                $(".nav li").removeClass("active");
                @if(isset($ruta))
                    $("#{{$ruta}}").addClass("active");
                @endif

            });

        </script>
    </head>
    <body>
        <!--Prevenir el botón atras después de cerrar la sesión-->
        <?php

```

```

    echo
    header("Cache-Control: no-store, no-cache, must-revalidate, max-
age=0");
    header("Cache-Control: post-check=0, pre-check=0", false);
    header("Pragma: no-cache");
    header('Content-Type: text/html');
    ?>

    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed"
data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
                    <span class="sr-only">Toggle Navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a href="{{ asset('/index') }}"></a>
                <a class="navbar-brand" href="{{ url('/')
}}">PARQUEANDO</a>
            </div>

            <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
                <ul class="nav navbar-nav">
                    @if(isset($parqueadero))
                    <li id="parqueaderos"><a href="{{ url('parqueaderos')
}}">Parqueaderos</a></li>
                    @endif
                    @if(Entrust::hasRole('admin'))
                    <li id="tipov"><a href="{{ url('tipovehiculos')
}}">Tipo de Vehículos</a></li>
                    @endif
                    @if(Entrust::hasRole('admin'))
                    <li id="usuarios"><a
href="{{URL::to('usuarios')}}">Usuarios</a></li>
                    @endif
                    @if(Entrust::hasRole('admin'))
                    <li id="roles"><a
href="{{URL::to('roles')}}">Roles</a></li>
                    @endif
                    @if(Entrust::hasRole('admin'))
                    <li id="permisos"><a
href="{{URL::to('permisos')}}">Permisos</a></li>
                    @endif
                    @if(Entrust::hasRole('admin'))
                    <li id="app"><a href="{{URL::to('parqueando')}}">WEB
Parqueando</a></li>
                    @endif
                </ul>
                <ul class="nav navbar-nav navbar-right">

                    @if (Auth::guest())
                    <li><a href="{{ url('/auth/login') }}">Iniciar
sesión</a></li>
                    <li><a href="{{ url('/auth/register')
}}">Registrarse</a></li>
                    @else
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-expanded="false"><i class="glyphicon
glyphicon-user"></i> {{ Auth::user()->name }} <span
class="caret"></span></a>
                        <ul class="dropdown-menu" role="menu">

```

```

        <li><a href="{ { url('/auth/logout')
}}">Salir</a></li>
    </ul>
</li>
    @endif
</ul>
</div>
</div>
</nav>
<div class="content">
    @yield('content')
</div>
<!-- Scripts -->

@yield('linkbot')
<div class="footer">
    <p style="margin-right: 25px; margin-top: 5px; font-family:
monospace; text-align: right"> v. 0.7.5 </p>
</div>

</body>

</html>

```

## Página Hija

Se determina de qué página maestra se requiere extender y se establecen las etiquetas de sección para insertar el código.

```

@extends('app')
@section('title')
usuarios
@stop
@section('content')
<div class="panel panel-info">
    <div class="panel-heading">Lista de Usuarios </div>
    <div class="panel-body" >

        <div class="table-responsive">
            <table class="table table-striped">
                <thead>
                    <tr><th>Nombre de
Usuario</th><th>Rol</th><th>Email</th><th>Nombres</th><th>Apellidos</th><th>Ed
itar</th><th>Eliminar</th></tr>
                </thead>
                @if(isset($users))
                <tbody>
                    <!--//este es un comentario-->
                    @foreach($users as $user)
                    <tr><td>{!! $user->name !!}</td>
                        <td>{!! $user->rol->name !!}</td>
                        <td>{!! $user->email !!}</td>
                        <td>{!! $user->nombres !!}</td>
                        <td>{!! $user->apellidos !!}</td>
                        @if(Entrust::can('editar_usuarios'))
                        <td>{!! Form::open(array('method' => 'GET', 'route' =>
array('usuarios.edit', $user->id))) !!}
                            <button class="btn btn-warning"><span
class="glyphicon glyphicon-pencil" aria-hidden="true"></span></button>
                            {!! Form::close() !!}</td>
                        <td>
                            {!! Form::open(array('id' => $user->id,'method' =>
'DELETE', 'route' => array('usuarios.destroy', $user->id))) !!}

```

```

                {!! Form::button('', array('class' => 'open btn
btn-danger glyphicon glyphicon-remove','id' => 'btn-eliminar','data-toggle' =>
'modal', 'data-target' => '#myModal', 'data-username' => $user-> nombres,
'data-apellidos' => $user-> apellidos, 'data-id' => $user-> id)) !!}
                {!! Form::close() !!}
            </td>
        @endif
    </tr>
@endforeach
</tbody>
@endif
</table>
</div>
</div>
</div>
</div>

<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                <h4 class="modal-title" id="myModalLabel">Confirmación</h4>
            </div>
            <div class="modal-body">
                <p id="usuario"></p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-
dismiss="modal" id="btn-cancel">Cancelar</button>
                <button type="button" class="btn btn-primary" id="btn-
ok">Aceptar</button>
            </div>
        </div>
    </div>
</div>
@endsection
@section('linkbot')
<script>
    $(document).on('click', '.open', function () {
        var $nombres = $(this).data('username');
        var $apellidos = $(this).data('apellidos');
        var $id = $(this).data('id');
        $('#usuario').html("¿Está seguro de eliminar el usuario <strong>" +
$nombres + " " + $apellidos + "</strong>?");

        $('#btn-ok').click(function () {
            $('#' + $id).submit();
        });
    });
</script>

```

### 4.3. Middleware

El middleware se ejecuta antes de que la petición sea procesada por el controlador, existe un middleware para la verificación de inicio de sesión y chequear los roles del usuario.



```

<?php

namespace App\Http\Middleware;

// First copy this file into your middleware directory
use Closure;

class CheckRole {

    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next) {

        // Get the required roles from the route
        $roles = $this->getRequiredRoleForRoute($request->route());
        // Check if a role is required for the route, and
        // if so, ensure that the user has that role.

        $user = $request->user();

        if (!empty($user)) {
            if ($user->hasRole($roles) || !$roles) {
                return $next($request);
            }
            return view('errors/403');
        }else{
            return view('auth/login');
        }
    }

    private function getRequiredRoleForRoute($route) {
        $actions = $route->getAction();
        return isset($actions['roles']) ? $actions['roles'] : null;
    }
}

```

#### 4.4. Controllers – Controladores

Un ejemplo de controlador:

```

<?php

namespace App\Http\Controllers;

use App\User;
use App\Http\Requests;
use Response;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Role;
use App\Permission;
use App\Http\Requests\RoleRequest;
use App\Http\Requests\RoleRequest_Update;
use Validator;
use Illuminate\Support\Facades\Redirect;

class RolesController extends Controller {

```

```

public function __construct() {
    $this->middleware('auth');
}

/**
 * Display a listing of the resource.
 *
 * @return Response
 */
public function index() {
    $parqueadero = false;
    if (count(\Request::user()->parqueaderos()->get()) > 0) {
        $parqueadero = true;
    }

    $data = array(
        'roles' => Role::all(),
        'permisos' => Permission::all(),
        'parqueadero' => $parqueadero,
        'ruta' => 'roles'
    );
    return view('role.role')->with($data);
}

/**
 * Show the form for creating a new resource.
 *
 * @return Response
 */
public function create() {
    return view('role.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @return Response
 */
public function store(RoleRequest $roleRequest) {

    $nombreRol = \Request::input('name');
    $nombreRolVisual = \Request::input('display_name');
    $descripcion = \Request::input('description');

    Role::create([
        'name' => $nombreRol,
        'display_name' => $nombreRolVisual,
        'description' => $descripcion
    ]);

    return redirect('roles');
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return Response
 */
public function show($id) {
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id

```

```

* @return Response
*/
public function edit($id) {
    try {
        $rol = Role::find($id);
        return view('role.edit')->with('rol', $rol);
    } catch (\PDOException $exc) {
        abort(500);
    }
}

/**
 * Update the specified resource in storage.
 *
 * @param int $id
 * @return Response
 */
public function update($id, RoleRequest_Update $roleRequest) {

    $rolReal = Role::find($id);

    $roleName = \Request::input('name');
    $rolVisual = \Request::input('display_name');
    $rolDescripcion = \Request::input('description');
    $rolReal->display_name = $rolVisual;
    $rolReal->description = $rolDescripcion;

    if ($rolReal->name !== $roleName) {
        $validator = Validator::make($roleRequest->all(), [
            'name' => 'unique:roles', ['name.unique' => 'El rol
ya existe'
        ]);
        if ($validator->fails()) {
            return Response::json($validator->messages(), 422);
        }
    }

    $rolReal->name = $roleName;
    $rolReal->save();
    return view('role.role');
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return Response
 */
public function destroy($id) {
    $rol = Role::destroy($id);
    return Redirect::route('roles.index');
}

```

## 4.5. Routes – Rutas

Se crearon rutas para GET, PUT y Resource, este último tipo admite servicios web restfull.

```
//PARQUEADEROS
```

```
Route::get('/parqueaderos/success', 'ParqueaderoController@success');
Route::put('/parqueaderos/activar', 'ParqueaderoController@activar');

Route::resource('/parqueaderos', 'ParqueaderoController', ['only' => ['index',
'create', 'store', 'edit', 'destroy', 'update', 'activar', 'mensaje']]);
```

## 4.6. Request – Peticiones

Aquí se realizan procesos de validación de datos enviados al servidor, se establecen reglas de validación y mensajes:

```
<?php

namespace App\Http\Requests;

use App\Http\Requests\Request;

class ParqueaderoForm extends Request {

    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize() {

        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules() {
        return [
            "nombre" => "required|max:250|unique:parqueaderos",
            "numero" => "required|numeric|min:1|max:999",
            // "email" => "required|email|unique:users",
            "telefono" => "required|numeric|min:8|max:9999999999",
            "lat" => "required|numeric|regex:/^[+-]?\d+\.\d+$/",
            "lng" => "required|numeric|regex:/^[+-]?\d+\.\d+$/",
            "pais" => "required|min:2|max:50",
            "ciudad" => "required|min:2|max:50",
            "direccion" => "required|min:2|max:250",
        ];
    }

    public function messages() {
        return [
            "nombre.required" => "El Nombre del parqueadero es requerido",
            "nombre.unique" => "Este nombre de parqueadero ya existe",
            "nombre.max" => "El número de caracteres máximo del Nombre es
100",

            "numero.numeric" => "El número de plazas es entero",
            "numero.required" => "El Número de plazas es requerido",
            "numero.min" => "El mínimo de plazas es uno",
            "numero.max" => "El máximo de plazas es 999",
            "ciudad.required" => "La ciudad es requerida",
            "telefono.required" => "El número de telefono es requerido",
            "telefono.numeric" => "Solo números en el campo teléfono",
            "telefono.max" => "Máximo 10 números en el campo teléfono",
            "telefono.min" => "Mínimo 7 números en el campo teléfono",
```

```
"lat.regex" => "Latitud no tiene el formato correcto",  
"lng.regex" => "Longitud no tiene el formato correcto",  
];
```

#### 4.7. Directorio Public

En este directorio van los archivos, librerías, multimedia que será compartida públicamente en la página web.

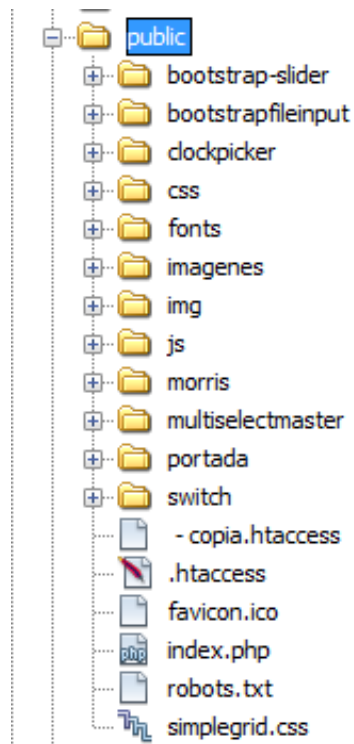


Ilustración 2: Directorio Public

#### 4.8. Composer.json

En este archivo se especifican los parámetros descriptivos del proyecto, además se establecen las librerías que son utilizadas por el proyecto.

```
{  
    "name": "DiegoMPerez/parqueando",  
    "description": "Aplicación para la búsqueda de parqueadeers  
privados.",  
    "keywords": ["parqueando", "laravel"],  
    "license": "MIT",  
    "type": "project",  
    "require": {  
        "laravel/framework": "5.0.*",  
        "laravelcollective/html": "~5.0",  
        "illuminate/html": "5.0.*@dev",  
    }  
}
```

```

        "guzzlehttp/guzzle": "~4.0",
        "mitulgolakiya/laravel-api-generator": "1.3.*",
        "zizaco/entrust": "1.4.1",
        "barryvdh/laravel-cors": "^0.7.3",
        "barryvdh/laravel-ide-helper": "^2.1",
        "doctrine/dbal": "^2.5"
    },
    "require-dev": {
        "phpunit/phpunit": "~4.0",
        "phpspec/phpspec": "~2.1",
        "f2m2/apidocs": "dev-master"
    },
    "autoload": {
        "classmap": [
            "database"
        ],
        "psr-4": {
            "App\\": "app/"
        }
    },
    "autoload-dev": {
        "classmap": [
            "tests/TestCase.php"
        ]
    },
    "scripts": {
        "post-install-cmd": [
            "php artisan clear-compiled",
            "php artisan optimize"
        ],
        "post-update-cmd": [
            "php artisan clear-compiled",
            "php artisan optimize"
        ],
        "post-create-project-cmd": []
    },
    "config": {
        "preferred-install": "dist"
    }
}

```