UNIVERSIDAD POLITÉCNICA DE TULANCINGO

ORTIZ OSORIO CRISTIAN URIEL
FRANCISCO MELO MARIO ETZAEL

U

P

T

Ingeniería en Sistemas Computacionales

Manual de creación de un serivicio Restful en PHP

LARAVEL. Es un framework de código abierto de php para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Para poder crear un proyecto en laravel debemos contar con composer el cual permite descargar las librerías y paquetes necesarios para el desarrollo de un

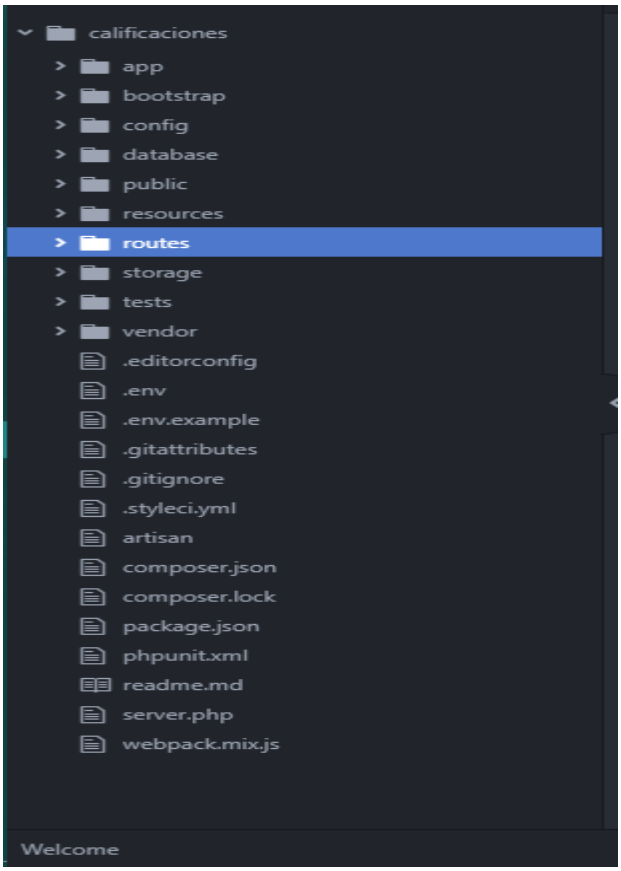
proyecto de laravel, composer está disponible para su descarga desde su página oficial, dejamos un pequeño tutorial para su instalación:

https://styde.net/instalacion-de-composer-y-laravel-en-windows/

Después de su instalación vamos a proceder a crear un proyecto con composer en la versión más actualizada de laravel (6.0) con el siguiente comando por consola dentro de la carpeta que queremos que se cree:

composer create-project laravel/laravel nameproyect

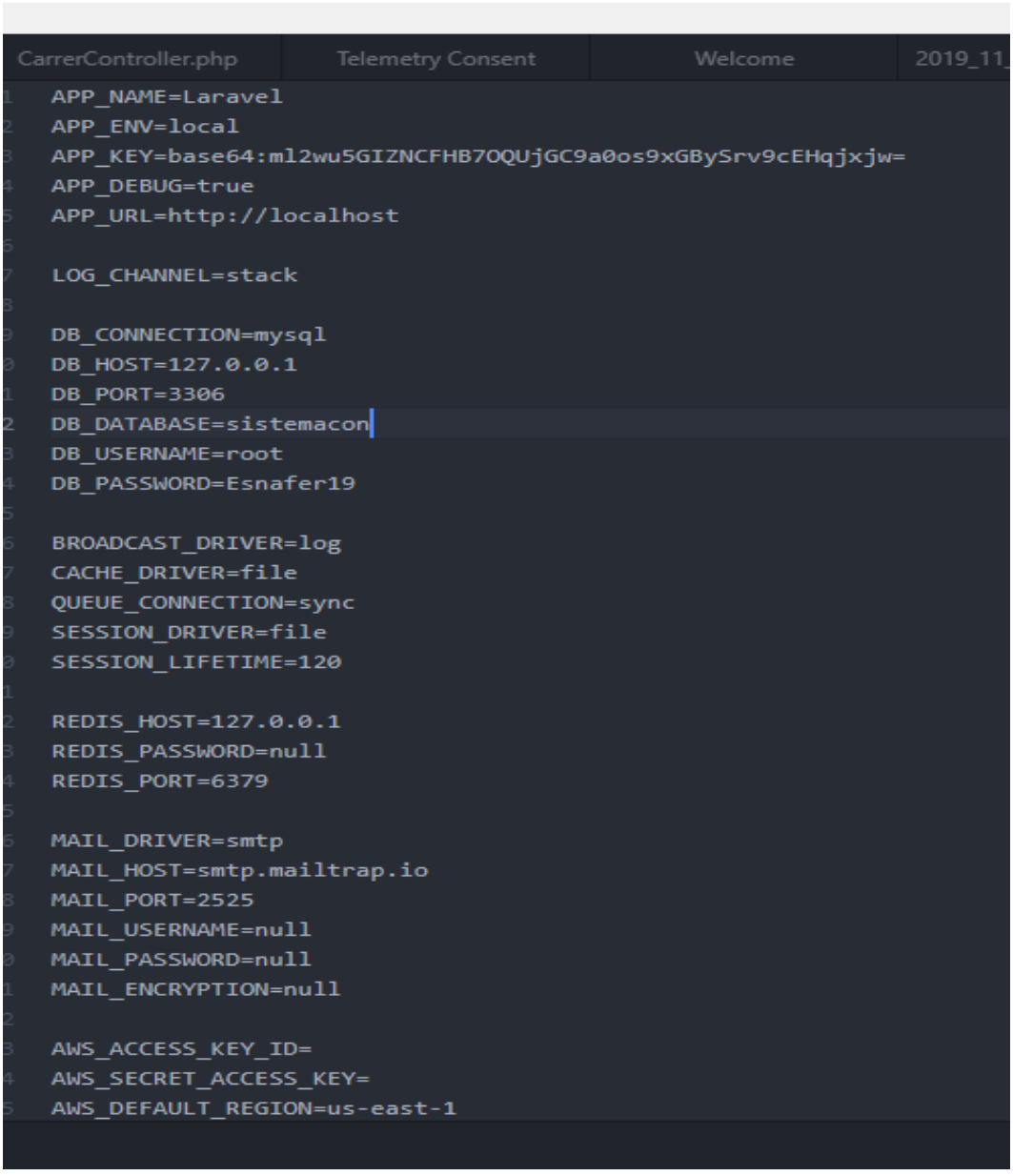
Después de esto composer empezara a descargar paquetes y librerías necesarias para laravel automáticamente y generara el proyecto. Al abrir nos encontramos con los directorios que en principio no sabremos como funcionana en principio los directorios y archivos más importantes que debemos aprender son los siguientes



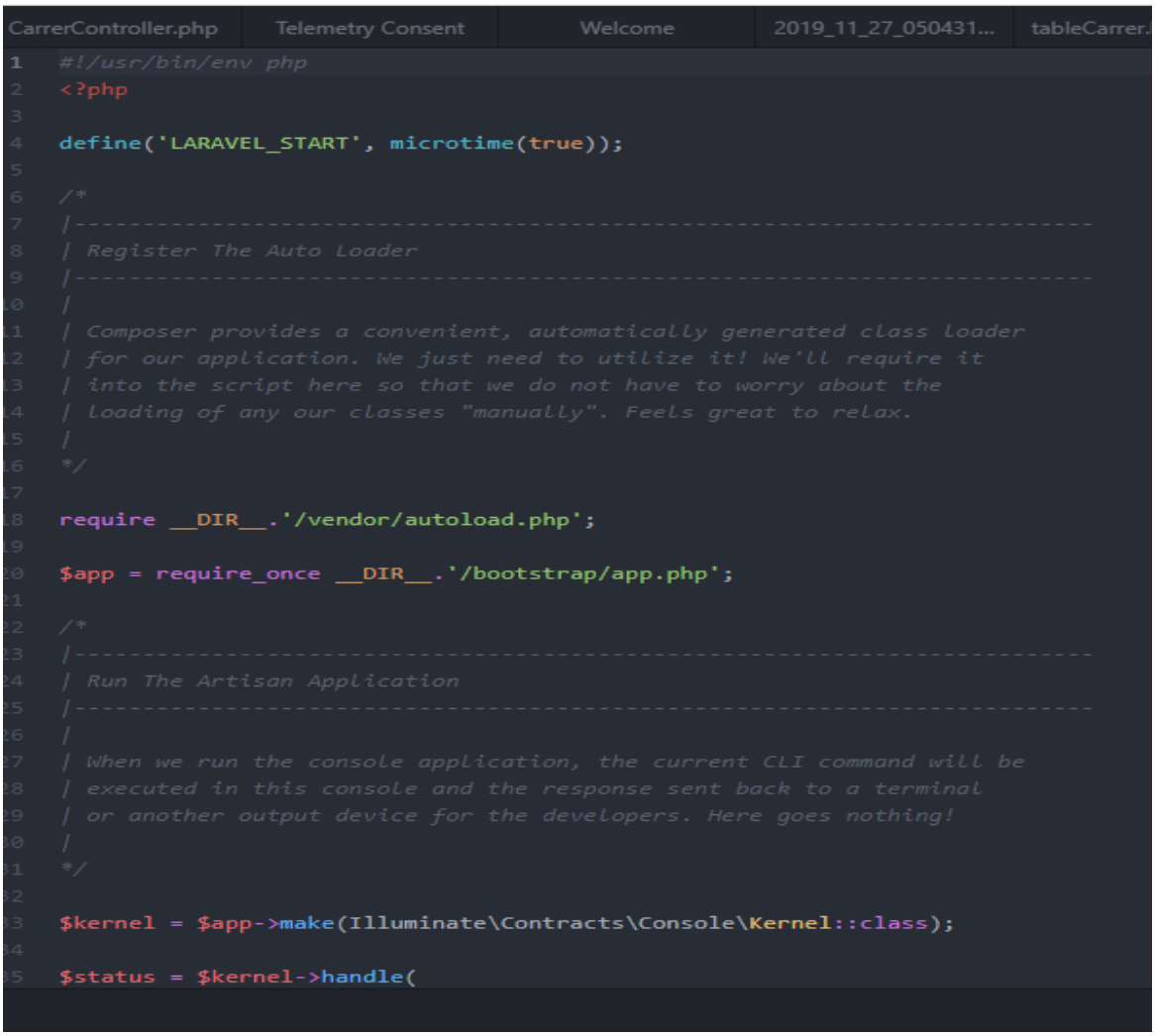
Los archivos mas importantes son

- .env
- Artisan

El primer archivo artisan es acerca de la configuracion de todo el proyecto, en muhcas de las ocasciones el unico cambio que debemos ejercer dentro del archivo es el cambio de la base de datos o de la conexión a la base de datos con una base de datos ya creada.



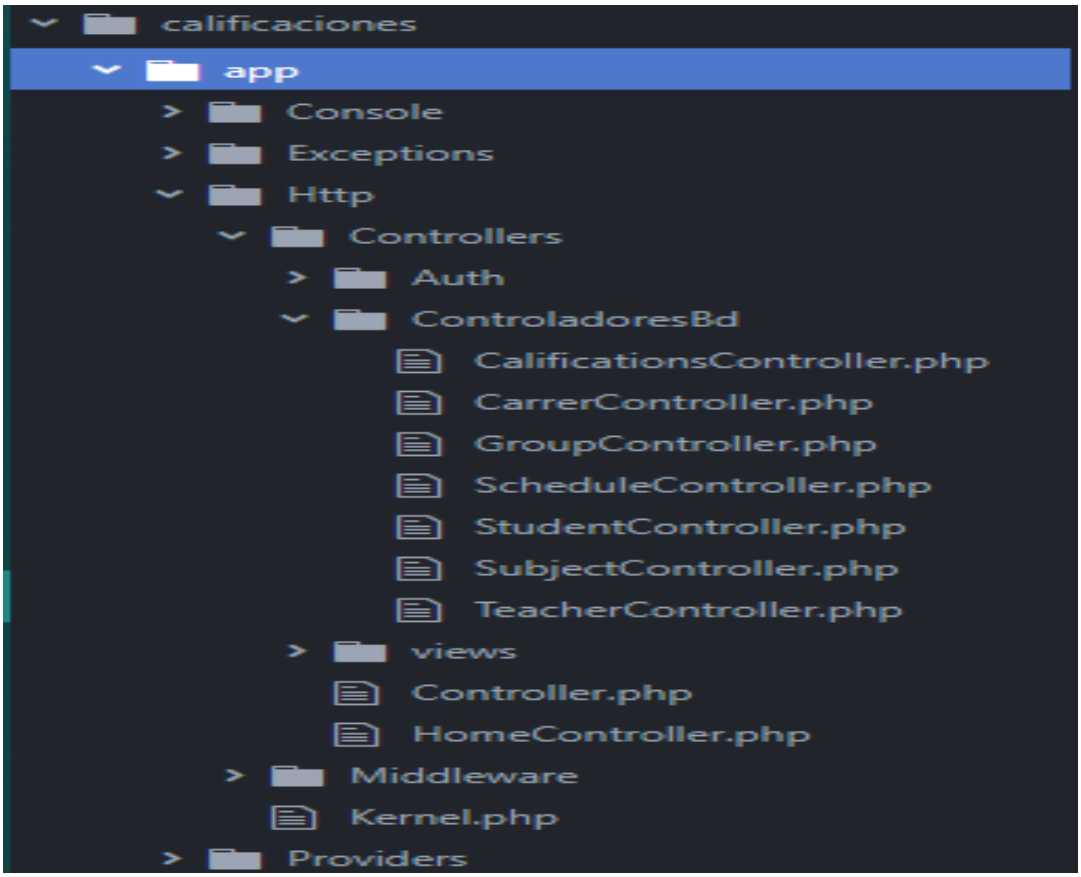
El siguiente archivo artisan contiene las propiedades del comando fundamental de laravel ‘artisan’ para crear o modificar dentro de la terminal

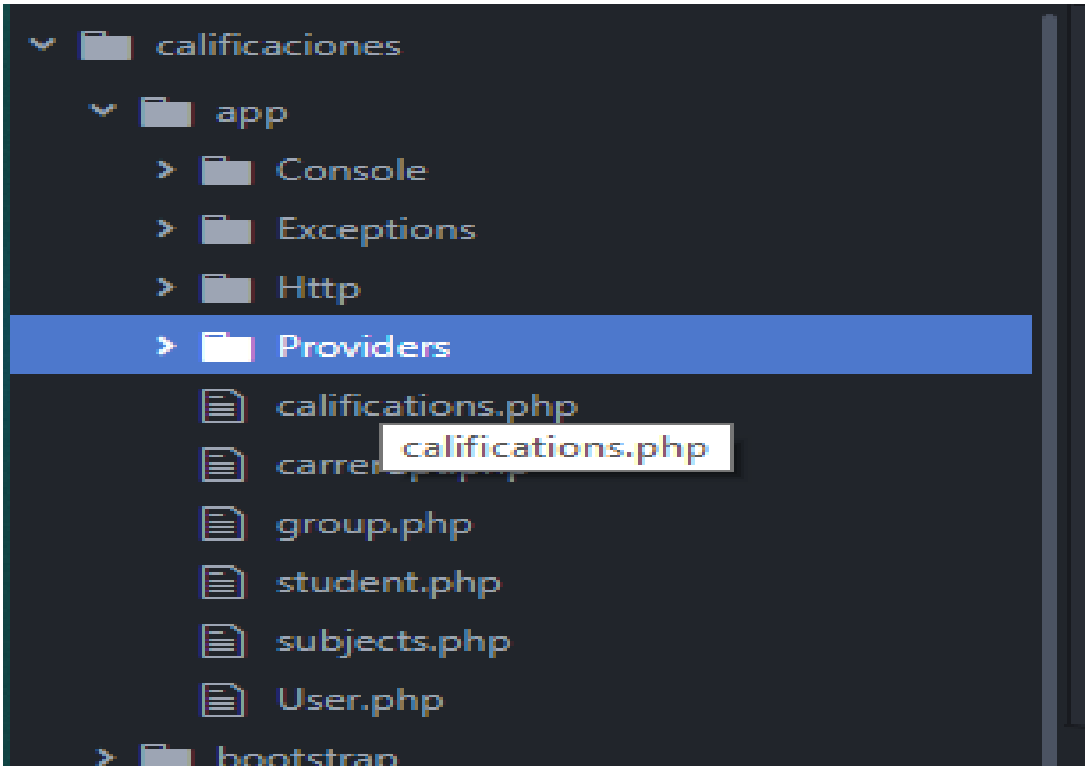


Las carpetas mas importantes son:

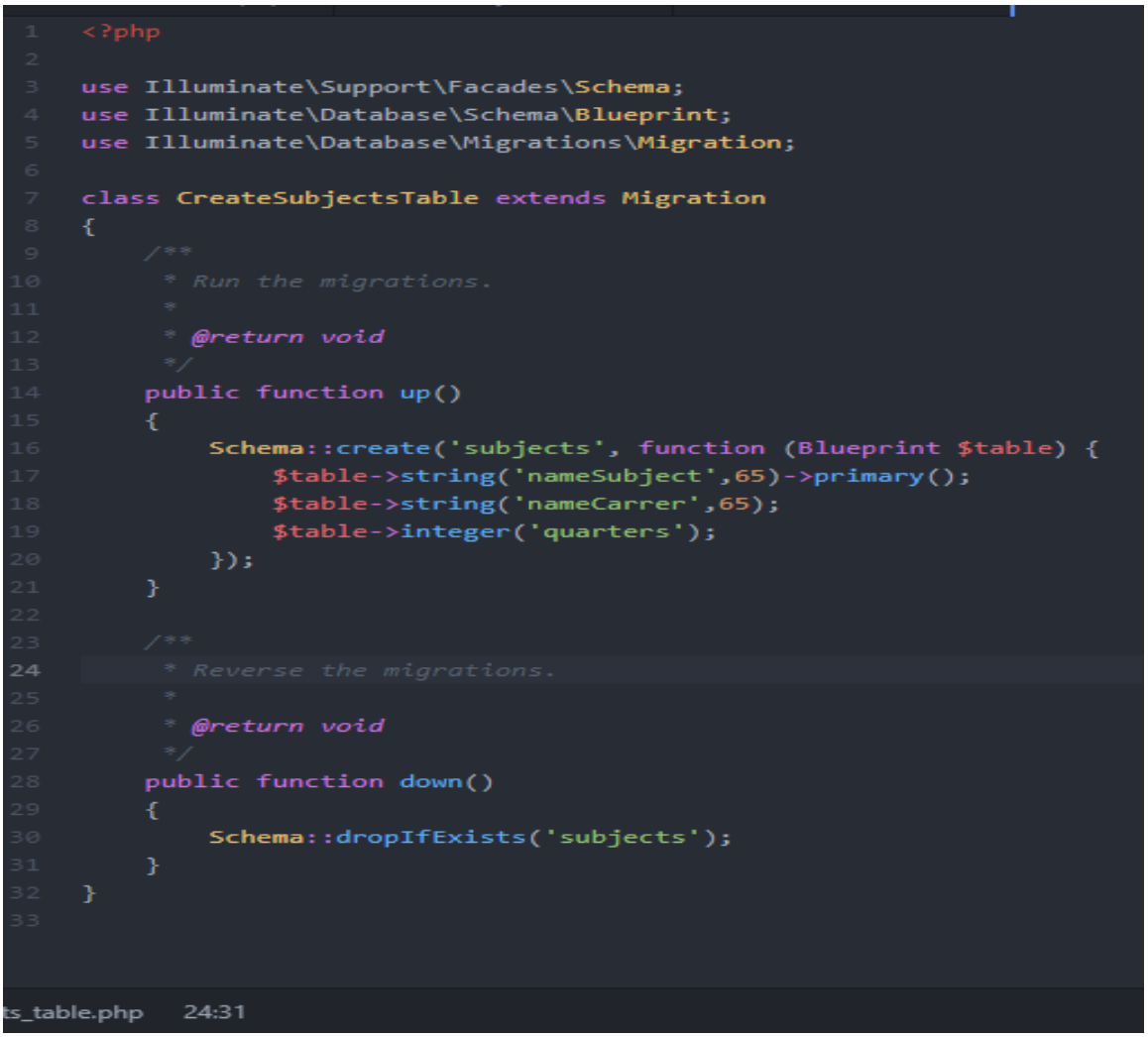
- App
  - Http
    - Controllers
- Database
- Resources
- Routes

Laravel es un framework que trabaja estrictamente con el modelo MVC para podert enteder un poco dentro de la carpeta App-Http se guardan los controladores y todas las funciones que estan implementadas para el desarrollo del proyecto, de la misma manera App contiene los modelos de cada tabla.


calificaciones
app
Console
Exceptions
Http
Controllers
Auth
ControladoresBd
CalificationsController.php
CarrerController.php
GroupController.php
ScheduleController.php
StudentController.php
SubjectController.php
TeacherController.php
views
Controller.php
HomeController.php
Middleware
Kernel.php
Providers


calificaciones
app
Console
Exceptions
Http
Providers
calificaciones.php
calificaciones.php
group.php
student.php
subjects.php
User.php

La carpeta es Database en donde se guardan lso scripts de la base de datos, gracias al potente framework de laravel podemos crear tablas a partir de una base de datos existentes dejando de lado el lenguaje sql y partiendo del lenguaje php para el desarrollo de las tablas de la siguiente forma



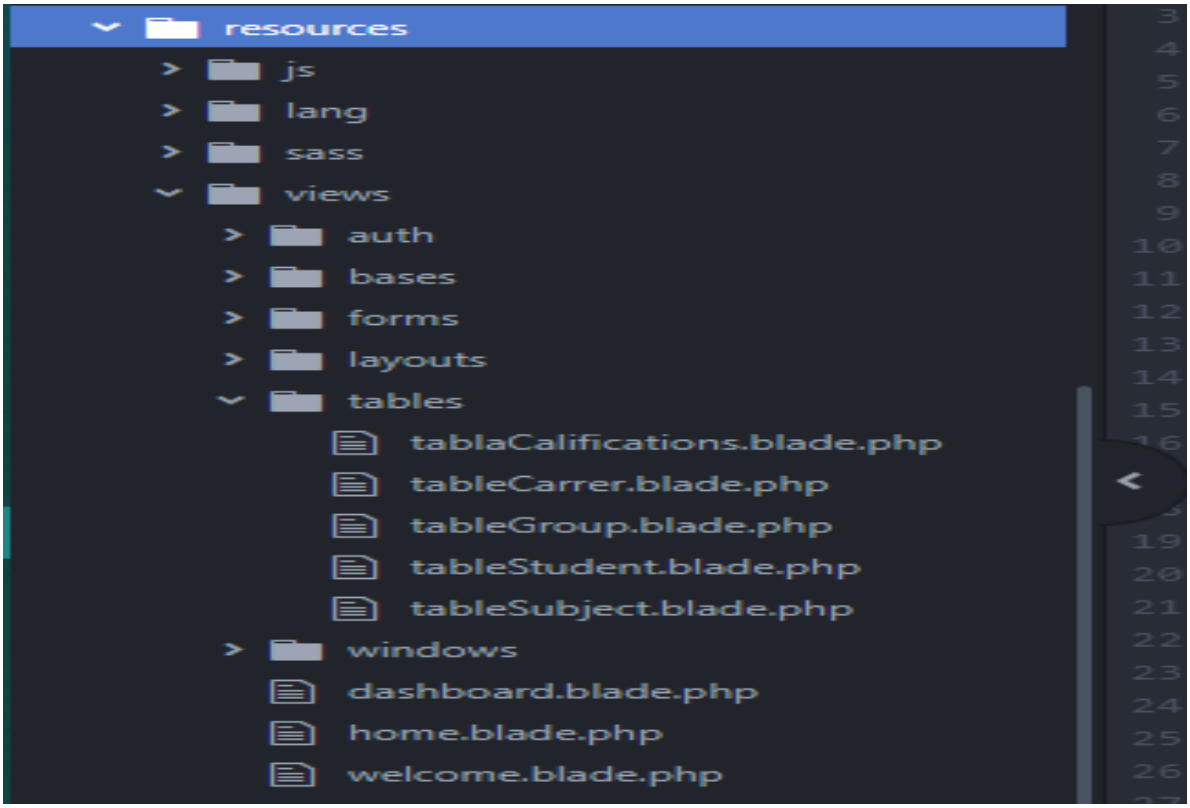
Para poder crear la tabla que se muestra en este script debemos ejectuar el siguiente comando en consola dentro del directorio del proyecto:

php artisan migrate

Este comando migra los scripts a tablas dentro de la base de datos. En caso de que queramos modifcar las tablas debemos de modificar el script y ejecutar el siguiente comando

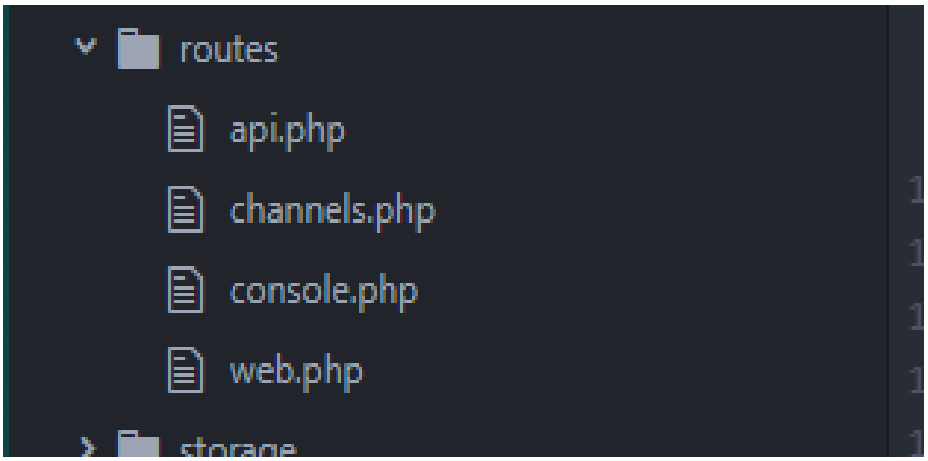
php artisan migrate refresh

La carpeta Resources contiene toda la vista del proyectos, archivos como js, css, html, php y Blade.php que es generado gracias al fremework, la diferencia entre archivo es su maleabilidad de html con php.



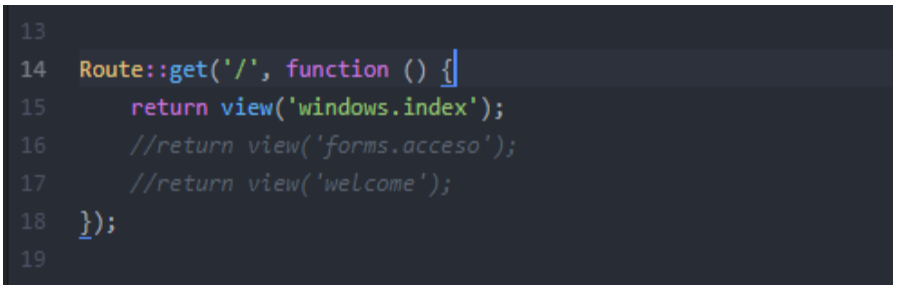
```
@extends('layouts.app')
@section('content')
<div class="container">
  <div class="container col-md-8 col-md-offset-2">
       <div class="panel panel-default">
           <div class="panel-heading">
               <h2>Alumnos</h2>
           </div>
           @if ($student->isEmpty())
               <div>No hay Registros</div>
           @else
               <table class="table">
                   <thead>
                       <tr>
                           <th>Número de cuenta</th>
                           <th>Nombre de estudiante</th>
                           <th>Apellido paterno</th>
                           <th>Apellido materno</th>
                           <th>Nombre carrera</th>
                           <th>Cuatrimestre</th>
                           <th>grupo</th>
                           <th>Modificar</th>
                           <th>Eliminar</th>
                       </tr>
                   </thead>
                   <tbody>
                       @foreach($student as $student)
                           <tr>
                               <td>{!! $student->numberAccount !!}</td>
                               <td>{!! $student->nameStudent !!}</td>
                               <td>{!! $student->lastNameP !!}</td>
                               <td>{!! $student->lastNameM !!}</td>
                               <td>{!! $student->nameCarrer !!}</td>
                               <td>{!! $student->quarters !!}</td>
```

Dentro de la carpeta routes es donde toda la magia empieza ya que permite el enrutamiento de todo el proyecto y esto es gracias a diferencia con otros frameworks que yo considero el mejor.



La carpeta contiene 4 arvhivos, solo se pueden confiugrar como archivos de ruta 2, los cuales son api.php y web.php. Existen 2 formas de crear rutas:

La primera consta de agregar una funcion dentro de la ruta sin necesidad de usar un controlador y retornando un valor.

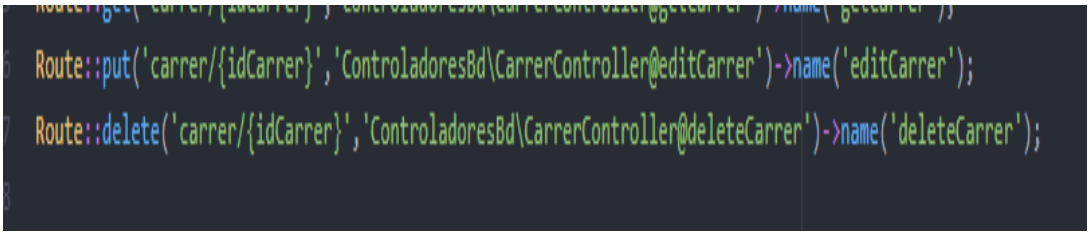


```
13
14  Route::get('/', function () {
15      return view('windows.index');
16      //return view('forms.acceso');
17      //return view('welcome');
18  });
19
```

La segunda consta de una ruta que va hacia un controlador especifico para realizar cierta función.

```
10  Auth::routes();
11
12  Route::get('vistaAcceso','views\vistas@inicioSesion')->name('vistaAcceso');
13  Route::get('registerStudent','views\vistas@registerStudent')->name('registerStudent');
```

Para poder hacer que el proyecto consuma recursos restful es agregando el siguiente prefijo a la ruta



```
6   Route::put('carrer/{idCarrer}','ControladoresBd\CarrerController@editCarrer')->name('editCarrer');
7   Route::delete('carrer/{idCarrer}','ControladoresBd\CarrerController@deleteCarrer')->name('deleteCarrer');
8
```

Los prefijos put para modificar, delete para eliminar dirigida a un controlador con una función específica.

Los navegadores solo permiten las peticiones por http post y get, como podemos hacer que un formulario mande peticiones put y delete a través de http, el siguiente formulario muestra un ejemplo en laravel de cómo llevarlo a cabo.

```
<div class="card-header">Grupos</div>

<div class="card-body">
    <form method="POST" action="{{ route('ediGroup',$group->groupA) }}">
      {{csrf_field()}}
      {{ method_field('PUT') }}
```

```
      <td>
        <form action="{{ route('delCarrer',$carrer->idCarrer) }}" method="POST">
          @csrf
          @method('delete')
          <button type="submit" class="btn btn-danger">Eliminar</button>
        </form>
```

El formulario tiene que llevar el método de POST, pero en el formulario se genera un input hiden con el method desesado, en este caso los archivos generados con Blade permiten generar componentes html, gracias a esto el método llevara un put o delete dependiendo el caso.

Para poder crear controladores necesitamos el comando:

php artisan make:controller nombre del controlador

Generamos un controlador con una estructura de código php el cual puede tener la siguiente estructura

```php
<?php

namespace App\Http\Controllers\ControladoresBd;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\carrerupt;

class CarrerController extends Controller
{

    public function getAllCarrer(){
        $carrer=carrerupt::all();
        return view('tables.tableCarrer')->with('carrer',$carrer);
    }
    //Agregar
    public function addCarrer(Request $request){
        $carrer=carrerupt::create($request->all());
        $carrer=carrerupt::all();
        return view('tables.tableCarrer')->with('carrer',$carrer);

    }
    public function obtenerCarr($idCarrer){
      $c=carrerupt::select('nameCarrer')->where('idCarrer','=',$idCarrer)->get();
      return $c;
    }
    //obtener un solo registro
    public function getCarrer($idCarrer){
        $carrer=carrerupt::where('idCarrer',$idCarrer)->first();
        return view('forms.editCarrer')->with('carrer',$carrer);

    }
    //Modificar
    public function editCarrer($idCarrer,Request $request){

r.php  24:81
```

El ejemplo recién mostrado es un controlador de base de datos el cuál ocupa un modelo que se ha generado y de él se obtienen las consultas. Para generar un modelo dentro accedemos al directorio del proyecto en la terminal escribimos el comando

php artisan make:model nombre del modelo

Genera un archivo con el nombre dado y su contenido es el siguiente

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class subjects extends Model
{

}
```

Y dentro pondremos la tabla a la que va a hacer referencia y sus respectivos campos.

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class subjects extends Model
{
    protected $table='subjects';
    public $timestamps = false;
    protected $fillable=array('nameSubject','nameCarrer','quarters');
}

```

Conclusiones

Podemos decir que laravel es un framework muy útil cundo de usar servicios restful se trata, es bastante manejable y fácil de entender, tiene cosas muy interesantes, sobre todo las rutas.

El servicio restful ayuda a los programadores a tener un mejor manejo de los datos y buen retorno con los archivos json.