UNIVERSIDAD MARIANO GÁLVEZ.

**Ingeniería en Sistemas de Información y Ciencias de la Computación.**

**Ingeniería de Proyectos.**

**Ing. Ivan de leon**

**Onceavo Semestre.**

****

**“AutoParking v1.0”**

Eddy Estuardo Agustin Marroquin 0903-13-13796

Erwin Jonatan Navarro Fuentes 0903-09-5855  
Julio Noe de Leon Soc 0903-13-12163  
Alma Maria Cordon 0903-13-8610

**San Pedro Sacatepéquez, San Marcos, 20 de Marzo de 2018.**

INDICE

[Rutas: 3](#__RefHeading___Toc166_936851535)

[Modelo 4](#__RefHeading___Toc168_936851535)

[Métodos 4](#__RefHeading___Toc170_936851535)

[Método GET 4](#__RefHeading___Toc172_936851535)

[Método POST 5](#__RefHeading___Toc174_936851535)

[Método PUT 8](#__RefHeading___Toc176_936851535)

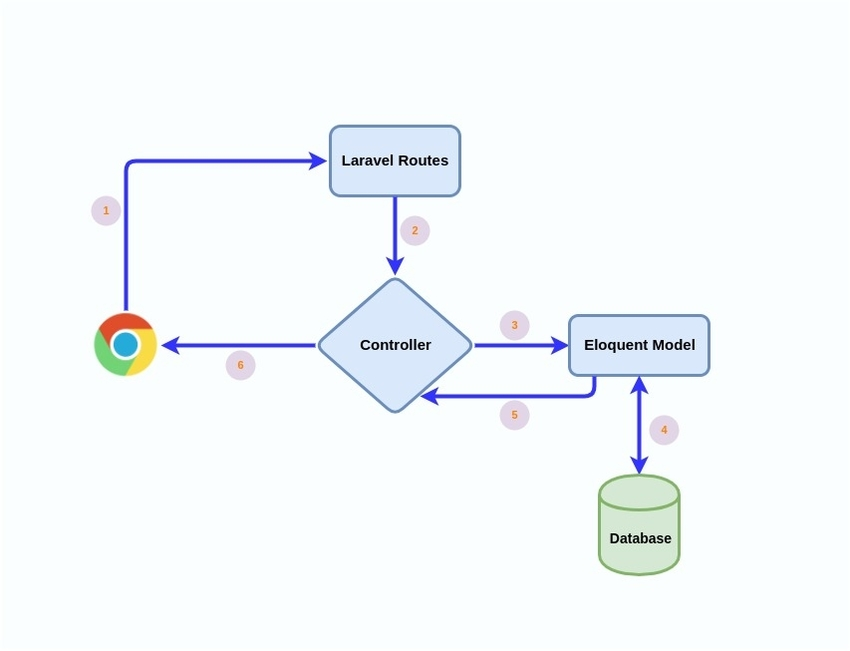
[Método DELETE 9](#__RefHeading___Toc178_936851535)

FASE 3

AUTOPARKING

Métos de la API AUTOPARKING.

Laravel es un framework que sigue la arquitectura Modelo Vista Controlador. MVC ayuda a separar las consultas de la base de datos (el modelo), de la lógica relacionada con la forma en que se deben procesar las solicitudes (el Controlador)



# Rutas:

Cuando el servidor recibe una solicitud HTTP, laravel intenta hacer coincidir con una ruta registrada dentro de cualquiera de los archivos de ruta. Todos los archivos de ruta se encuentran dentro del directorio de rutas **routes/api.php.**

GET, POST, PUT Y DELETE son los verbos HTTP (más conocidos popularmente como métodos HTTP), especialmente conocidos para construir un servicio RESTful.

Los métodos HTTP solicitan al servidor que realice la acción deseada en un recurso determinado.

Route::apiresource('datas', 'DataController');

Route::apiresource('userParkings', 'UserParkingController');

Route::apiresource('locations', 'LocationController');

Route::apiresource('cities', 'CityController');

Route::apiresource('countries', 'CountryController');

# Modelo

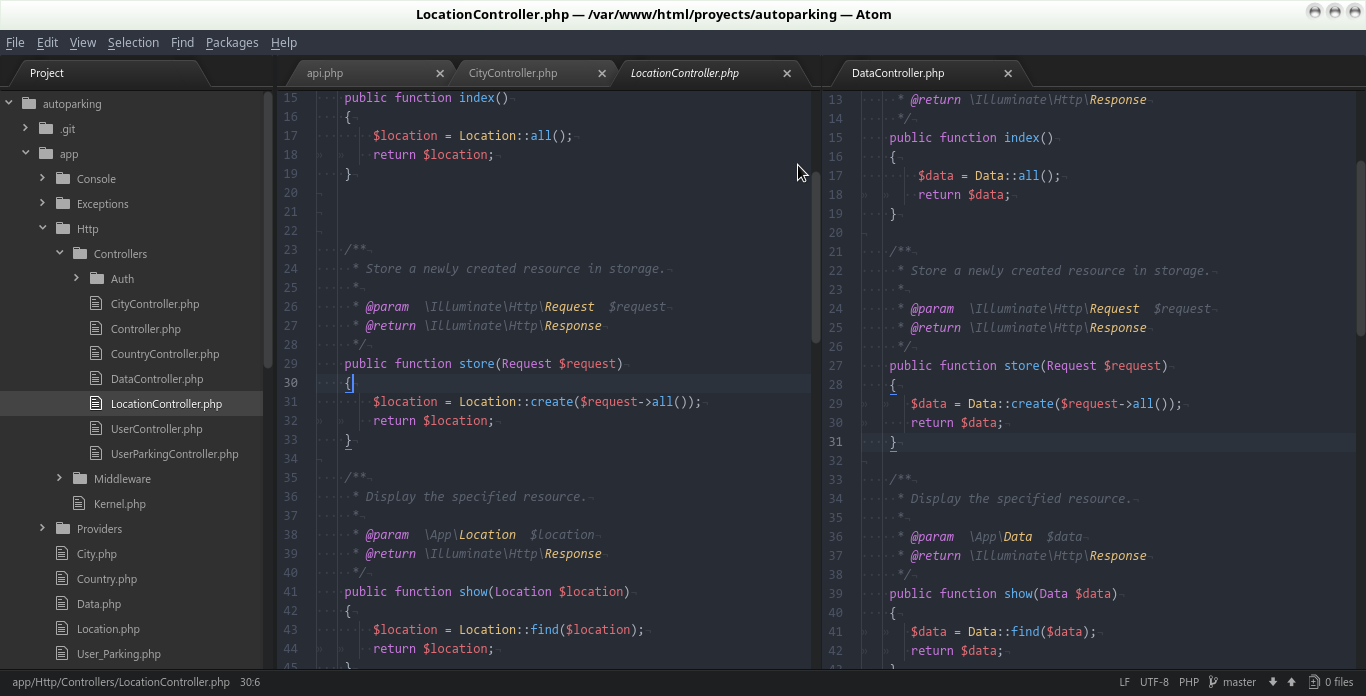
Los recursos necesitan un modelo para interactuar con la base de datos. El modelo es la capa que se encuentra en la parte superior de la base de datos, ocultando toda las operaciones específicas de la base de datos.

# Métodos

## Método GET

Este método recupera todos los datos que estamos consultando en la base de datos, podemos realizar consultas generales o especificas dandole un ID concreto.

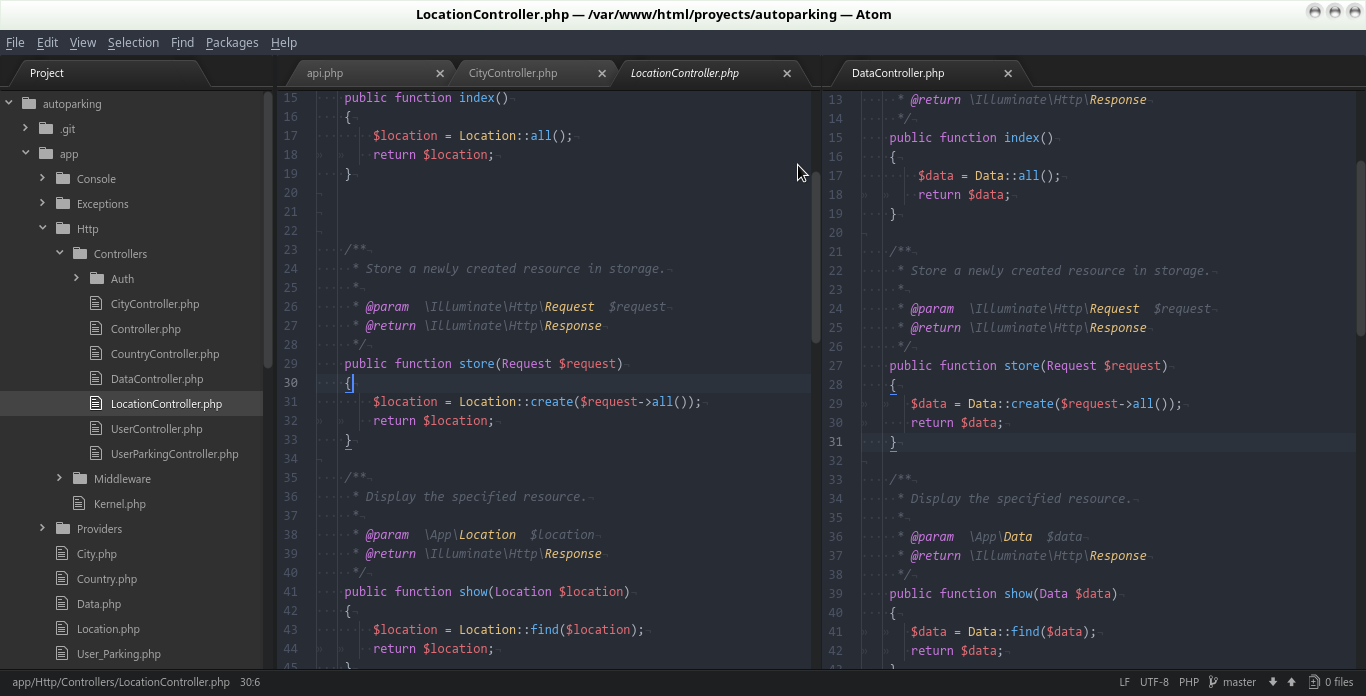
Este método lo empleamos para realizar las consultas a nuestros diferentes recursos que tenemos definidos.



Con los métodos INDEX Y SHOW, nosotros realizamos la acción de mostrar las consultas realizadas por medio de las peticiones http, puede ser que una petición sea general y no especifique un dato en especial, así como pasar como parametro un ID, para obtener especialmente los datos de un registro.

## Método POST

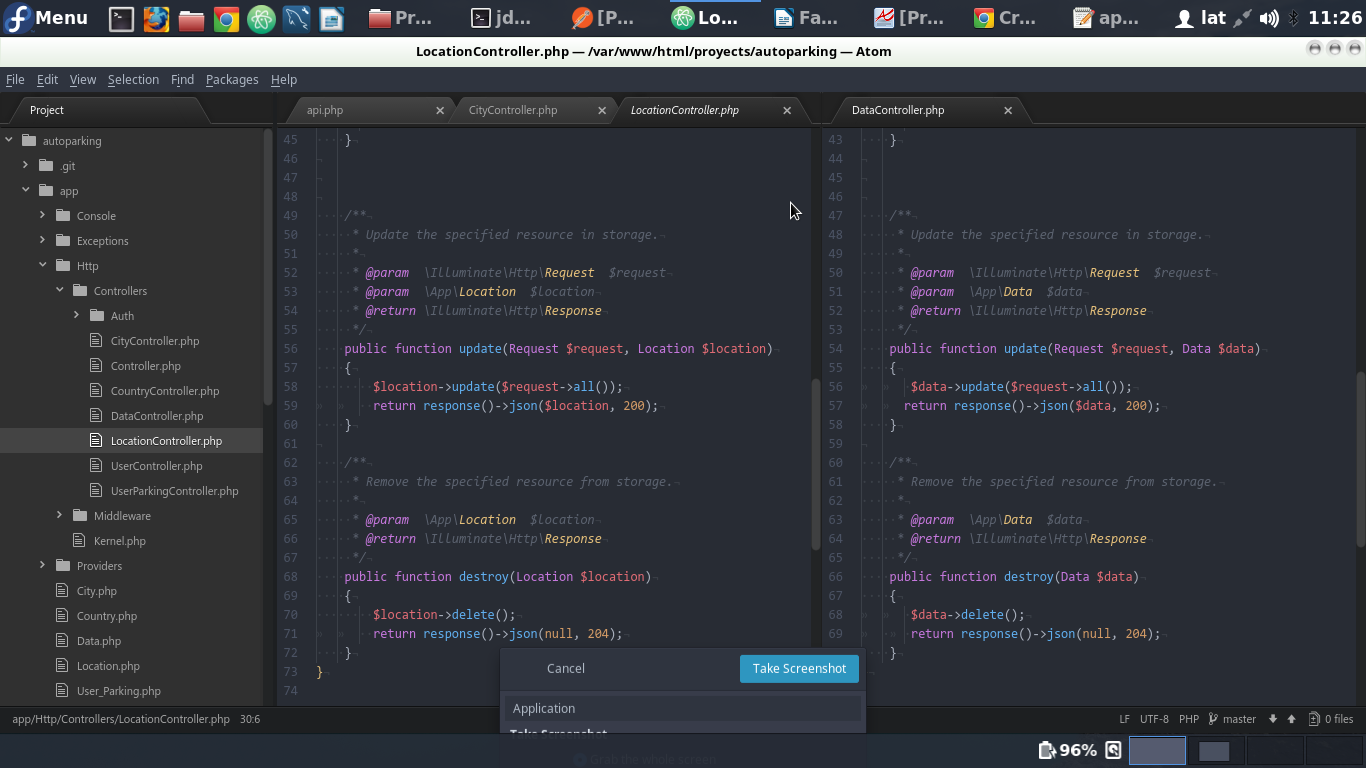
Este método permite crear y almacenar datos en nuestra base de datos.



El método store nos permite realizar la acción de insertar o crear un nuevo registro en la base de datos, cuando realizamos una petición por medio de la ruta GET, laravel realizar el trabajo de buscar que método hace la acción requerida de acuerdo con la petición que se realice.

## Método PUT

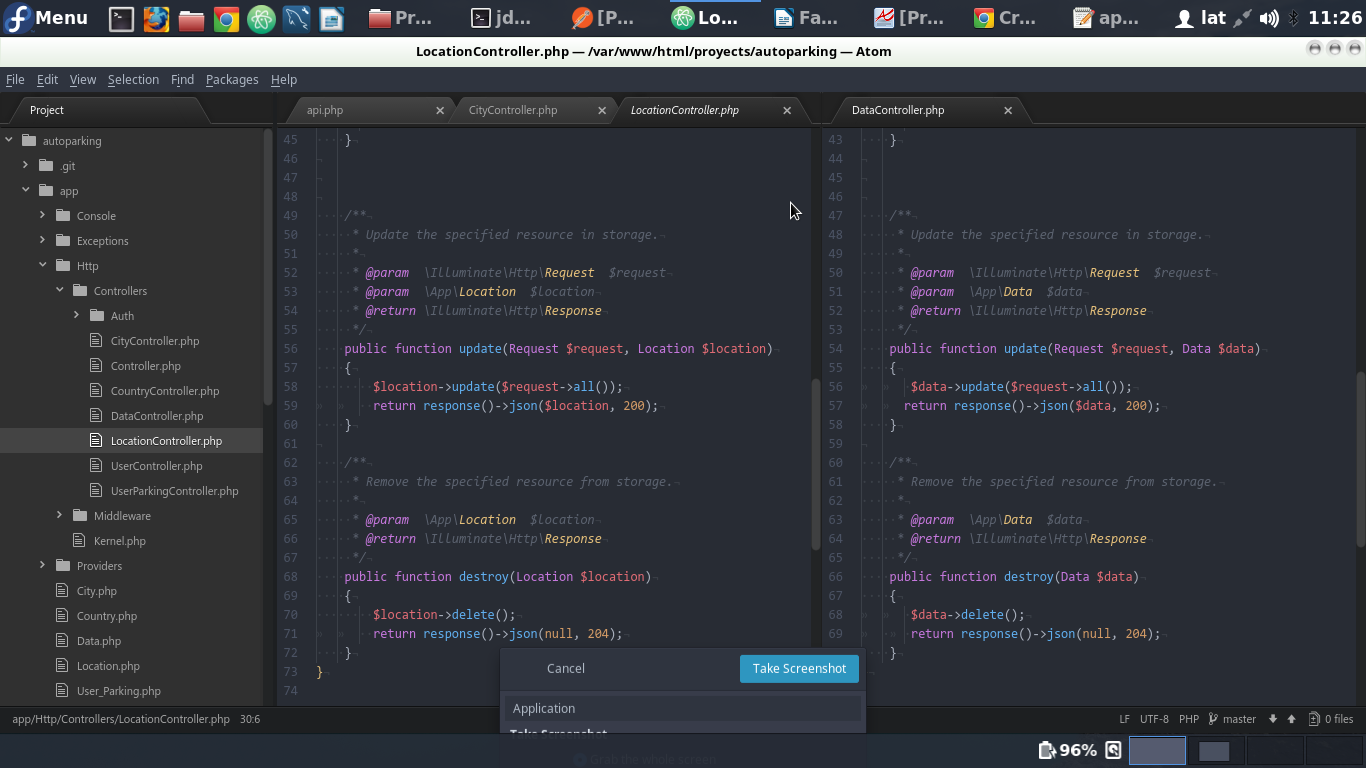
Con este método podemos actualizar datos dentro de nuestra base de datos, pasandole como parametro un ID de identificación de lo que deseamos actualizar.



Para actualizar datos dentro de nuestra base de datos empleamos el Método update, al cual le pasamos un parametro que identifica el registro que deseamos modificar.

## Método DELETE

Con este método eliminamos datos dentro de la base de dato pasandole como parametro un id de identificación de lo que deseamos eliminar.



Para eliminar registros de nuestra base de datos, empleamos el método DESTROY de nuestro controlador del recurso requerido, pasandole como parametro un ID, para saber que registro queremos eliminar.

Cabe mensionar que estos métodos dentro de nuestros controladores son generados automaticamente por laravel, estos métodos pueden ser generados manualmente con nombres personalizados, pero empleamos ell framework de laravel para hacer esta tarea mas sencilla.