

```
public function update(Request $request, Receta $receta)
```

```
// Revisar el policy  
$this->authorize('update', $receta);
```

```
// validación  
$data = $request->validate([  
    'titulo' => 'required|min:6',  
    'preparacion' => 'required',  
    'ingredientes' => 'required'
```

# Eloquent

## El ORM de Laravel



```
$img->save();  
  
// Asignar al objeto  
$receta->imagen = $ruta_imagen;  
}
```

Juan Pablo De la torre Valdez

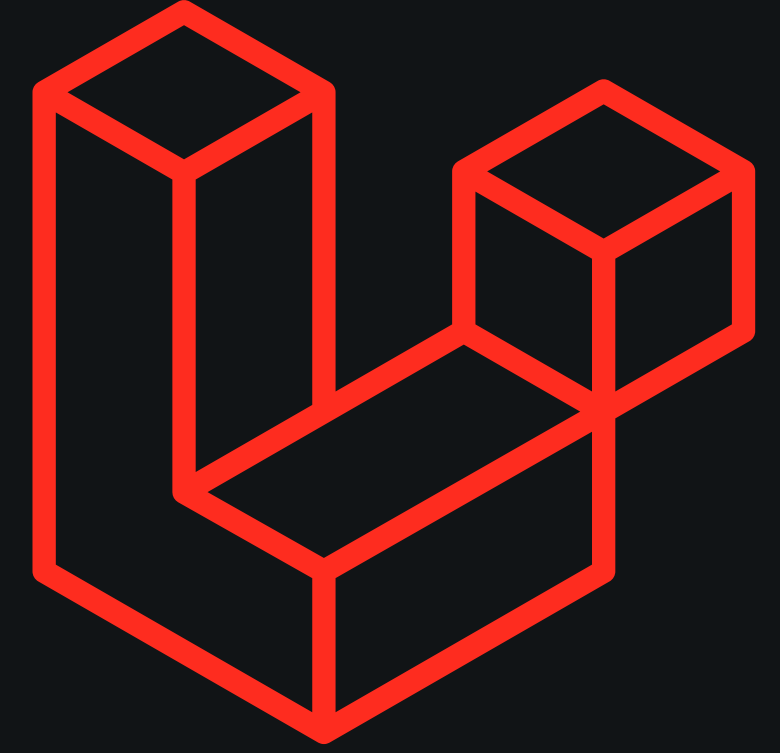
[www.codigoconjuan.com](http://www.codigoconjuan.com)



```
// redireccionar  
return redirect()->action('RecetaController@index');
```

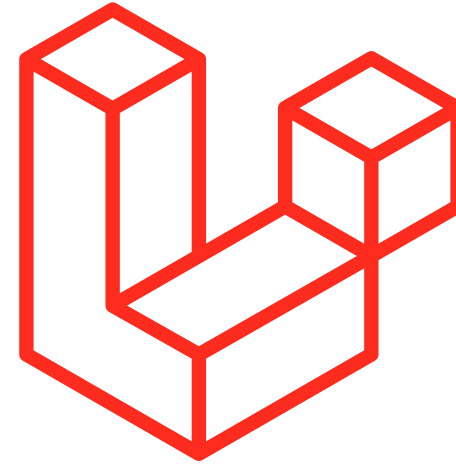
# Eloquent

## ORM



Laravel incluye su propio ORM (Object Relational Mapper) que hace muy sencillo interactuar con tu base de datos.

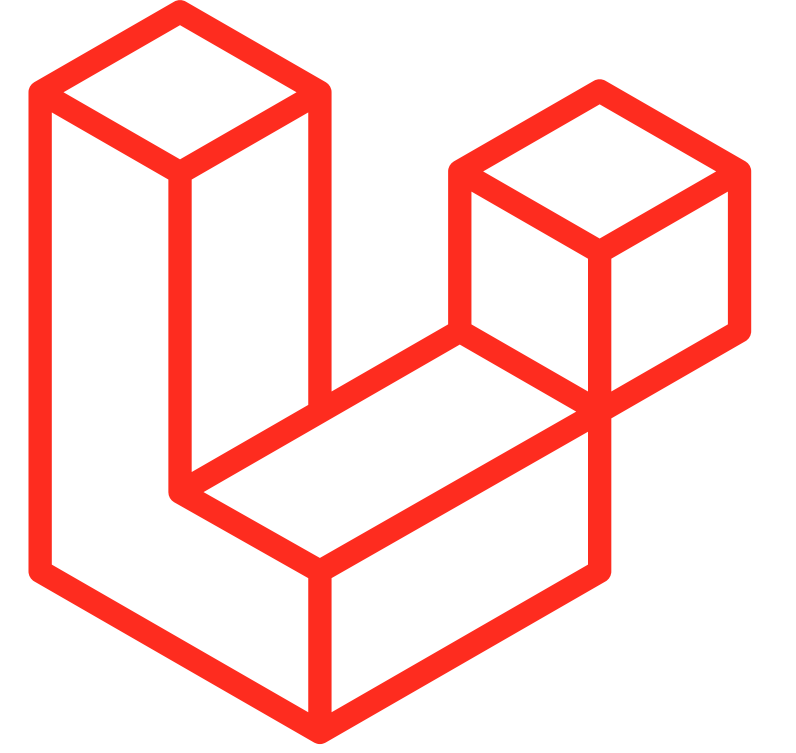
En Eloquent cada tabla, tiene su propio modelo; ese modelo interactúa únicamente con esa tabla y tiene las funciones necesarias para crear registros, obtenerlos, actualizarlos y eliminarlos.



**sail php artisan make:model Cliente**

# Convenciones

## En Modelos



Cuando creas el Modelo Cliente, Eloquent asume que la tabla se va a llamar clientes

Si el Modelo se llama Producto; Eloquent espera una tabla llamada productos

Puede ser un problema llamar tu modelo Proveedor, porque Eloquent espera la tabla llamada provedors, pero se puede reescribir en el modelo.