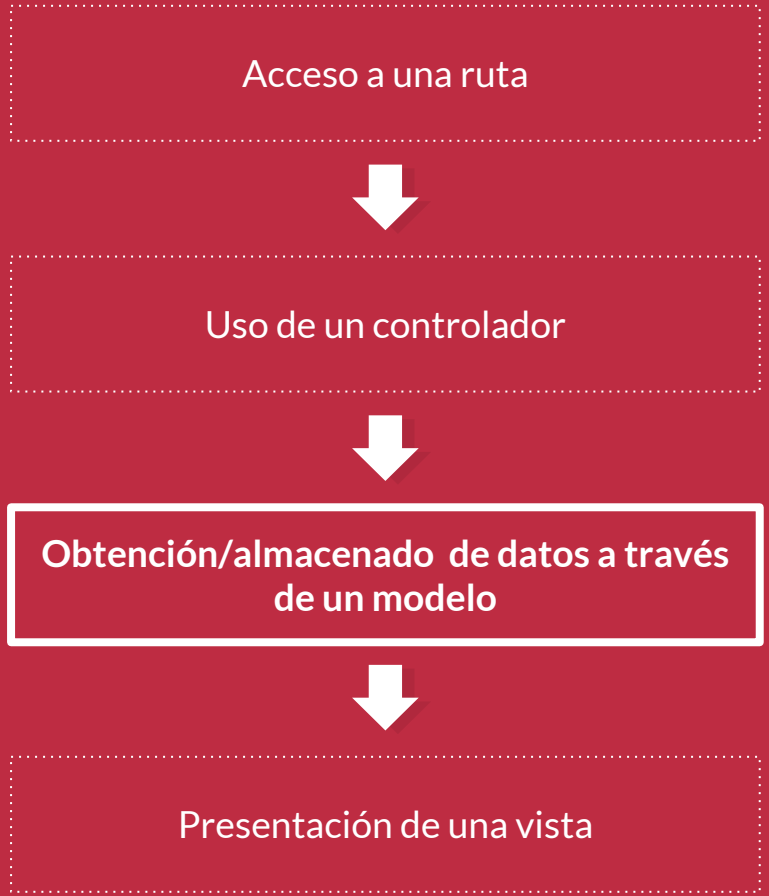
4.

MODELOS

FLUJO DE EJECUCIÓN



4.1 ELOQUENT ORM

Se incluye en Laravel con el fin de manejar más fácilmente los procesos correspondientes al uso de las bases de datos. **Sin escribir código SQL seremos capaces de crear, leer, actualizar y eliminar datos en nuestra aplicación**.

4.1 ELOQUENT ORM: MODELOS

Eloquent ORM hace uso de los Modelos para recibir o enviar la información a la base de datos. En nuestro caso, necesitamos un modelo Post para gestionar los artículos del CMS.

```
$ php artisan make:model Post
```

4.1 ELOQUENT ORM: MODELOS

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    //
}
```

4.1 ELOQUENT ORM: CONVENCIONES

Nombre de la tabla

```
/**
 * The database table used by the model.
 *
 * @var string
 */
 protected $table = 'users';
```

4.1 ELOQUENT ORM: CONVENCIONES

Claves primarias

Eloquent ORM asume por defecto que cada tabla tiene una columna con clave primaria llamada '`id`'. Si queremos modificar ese valor por defecto, debemos definir la variable `$primaryKey`.

4.1 ELOQUENT ORM: CONVENCIONES

Mass assignment

Utilizando el método '`create`' podemos crear un nuevo modelo en una sola línea de código, pero antes de poder hacer esto, debemos primero especificar el atributo `$fillable` o `$guarded`.

4.1 ELOQUENT ORM: CONVENCIONES

Mass assignment: `$fillable`

```
/**
 * The attributes that are mass assignable.
 *
 * @var array
 */
 protected $fillable = ['title', 'body'];
```

4.1 ELOQUENT ORM: CONVENCIONES

Mass assignment: `$guarded`

Mientras que `$fillable` es una lista de los atributos que pueden ser asignados en masa, `$guarded` es una lista de atributos que no pueden serlo.