

4.3 SEEDERS

Los *seeders* se crean con el comando `make:seeder` de Artisan y sirven para introducir registros en la base de datos.

Para introducir posts, crearemos el *seeder*
`PostsTableSeeder`:

```
$ php artisan make:seeder PostsTableSeeder
```

4.3 SEEDERS

Dentro del método run, podemos introducir el siguiente contenido:

```
DB::table('posts')->insert([
    'title' => 'Title 1',
    'body' => 'Content of the post #1',
]);
```

4.3 SEEDERS

Para ejecutarlo, podemos utilizar Artisan de nuevo:

```
$ php artisan db:seed --class=PostsTableSeeder
```

Si quisiéramos ejecutar todos los seeds sin especificar la clase habría que hacer una llamada a cada *seeder* dentro de `DatabaseSeeder::run()`:

```
$this->call(PostsTableSeeder::class);
```

4.4 ACCESO A DATOS

Para los siguientes apartados crearemos una ruta por cada método. De esta manera podremos ver el resultado, el cual devolveremos como JSON (por ahora).

Todos los métodos están dentro de `PostController`.

4.4 ACCESO A DATOS

`all()`

```
Route::get('/', 'PostController@index');
```

```
public function index()
{
    $posts = Post::all();
    return $posts->toJson();
}
```

4.4 ACCESO A DATOS

find()

```
Route::get('post/{id}', 'PostController@show');
```

```
public function show($id)
{
    $post = Post::find($id);
    return $post->toJson();
}
```

4.4 ACCESO A DATOS

`findOrFail()`: lanza 404

```
Route::get('post/{id}', 'PostController@show');
```

```
public function show($id)
{
    $post = Post::findOrFail($id);
    return $post->toJson();
}
```