

## Hoja06\_MVC\_08

Seguimos desarrollando la web para la gestión de un zoológico y configurando la base de datos.

### 1.- Modelo de datos

En este ejercicio vamos a crear el modelo de datos asociado con la tabla animales. Para esto usaremos el comando apropiado de Artisan para crear el modelo llamado Animal.

Una vez creado este fichero lo abriremos y comprobaremos que el nombre de la clase sea el correcto y que herede de la clase Model.

También debemos comprobar si el nombre de la clase en plural coincide con el nombre de la tabla de la base de datos (siempre en inglés).

Y ya está. No es necesario hacer nada más. El cuerpo de la clase puede estar vacío ({}), ¡todo lo demás se hace automáticamente!

### 2.- Uso de la base de datos

Vamos a actualizar los métodos del controlador AnimalController para que obtengan los datos desde la base de datos. Seguiremos los siguientes pasos:

- Modificar el método **index** para que obtenga toda la lista de animales desde la base de datos usando el modelo Animal y que se lo pase a la vista.
- Modificar el método **show** para que le pase el animal pasado por parámetro a la vista.
- Modificar el método **edit** para que le pase el animal pasado por parámetro a la vista.

Ya no necesitaremos más el array de animales que habíamos puesto en el controlador, así que lo podemos eliminar.

Ahora tendremos que **actualizar las vistas** para que en lugar de acceder a los datos del array los obtenga del objeto con el animal. Para esto cambiaremos en todos los sitios donde hayamos **puesto \$animal['campo'] por \$animal->campo**.

Por último, para calcular la edad utilizaremos el paquete Carbon de Laravel. En la clase Animal podemos crear el siguiente método:

```
public function getEdad()
{
    $fechaFormateada=Carbon::parse($this->fechaNacimiento);
    return $fechaFormateada->diffInYears(Carbon::now());
}
```

En la misma clase tenemos que poner la línea **use Carbon\Carbon;** (ya que utilizamos la clase Carbon

### 3.- Creación de semillas para la base de datos

Sirven para rellenar tablas de la base de datos con datos iniciales. Para esto seguiremos los siguientes pasos:

- Crearemos una clase seeder llamada AnimalSeeder
- Desde el método run de DatabaseSeeder.php lo llamaremos de la forma:
- Para que reconozca DB use Illuminate\Support\Facades\DB;

```
public function run()
{
    DB::table('animales')->delete();
    $this->call(AnimalSeeder::class);
}
```

- Movemos el array de animales que se facilitaba en los materiales y que habíamos copiado dentro del controlador AnimalController a la clase de semillas (AnimalSeeder.php), guardándolo de la misma forma, como variable privada de la clase.
- Dentro del método run() de AnimalSeeder realizamos las siguientes acciones y recuerda para que reconozca Animal deberás llamar a su modelo y averigua a que espacio pertenece Str

```
foreach ($this->animales as $animal)
{
    $a = new Animal();
    $a->especie = $animal['especie'];
    $a->slug = Str::slug($animal['especie']);
    $a->peso = $animal['peso'];
    $a->altura = $animal['altura'];
    $a->fechaNacimiento = $animal['fechaNacimiento'];
    $a->imagen = $animal['imagen'];
    $a->alimentacion = $animal['alimentacion'];
    $a->descripcion = $animal['descripcion'];
    $a->save();
}
$this->command->info('Tabla animales inicializada con datos');
```

- Por último tendremos que ejecutar el comando de Artisan que procesa las semillas (**php artisan db:seed**) y una vez realizado abriremos PHPMyAdmin para comprobar que se ha rellenado la tabla