

Modificamos la web para la gestión de un zoológico para añadir más relaciones

### 1.- Añadir dos relaciones uno a muchos

Cada cuidador debe tener obligatoriamente 2 titulaciones.

Crea el modelo **Titulacion**. Únicamente tendrá un campo id, un nombre y un slug.

Vamos a crear dos campos a la migración de cuidadores. Cada cuidador tendrá un `id_titulacion1` y un `id_titulacion2`.

Cada una de estas será una relación uno a muchos

Crearemos 20 titulaciones automáticamente utilizando factory.

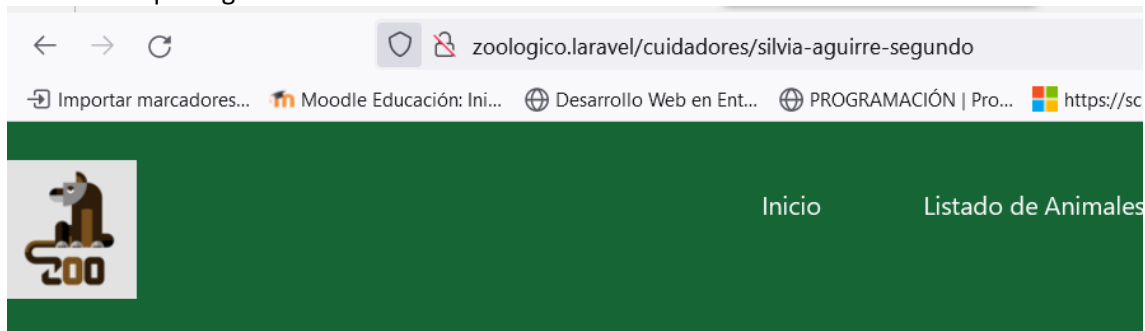
Modificamos el factory de cuidadores, para que introduzca las titulaciones.

Crearemos una ruta **cuidadores/{cuidador}** de tipo GET que reciba un cuidador. Haremos un controlador **CuidadorController** con un método **show** que reciba un cuidador y se lo pase a la vista **cuidadores.show**.

Dentro de la vista se mostrará el nombre del cuidador y sus titulaciones en forma de lista y El método titulaciones puede ser algo así:

```
public function titulaciones()
{
    // cada belongsTo devuelve uno. Por eso los uno
    return $this->belongsTo(Titulacion::class, 'id_titulacion1')->get()
        ->merge($this->belongsTo(Titulacion::class, 'id_titulacion2')->get());
}
```

Y un botón que regrese al listado



## vista detalle del cuidador Silvia Aguirre Segundo

Titulación: Vel optio quisquam iste eveniet et consectetur et.

Titulación: Beatae tempore id recusandae doloremque sint soluta.

[Volver al listado](#)

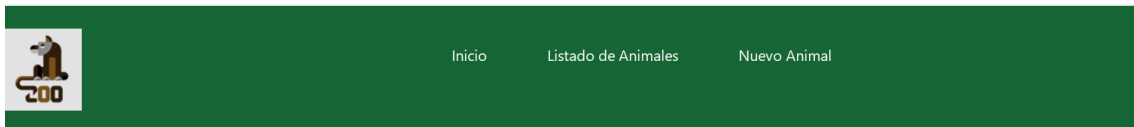
**Hoja06\_MVC\_13**

Modificar la vista listado de animales para que si pinchan en un cuidador nos muestre sus titulaciones.

También haremos la ruta **titulaciones/{titulacion}** de tipo GET que recibirá una titulación y mostrará todos los cuidadores que posean dicha titulación.

El método cuidadores del modelo será algo así:

```
public function cuidadores(){  
    // cada hasMany me devuelve muchos. Por eso le pongo otra condicion.  
    return $this->hasMany(Cuidador::class,'id_titulacion1')->get()  
    ->merge($this->hasMany(Cuidador::class,'id_titulacion2')->get());  
}
```



**vista de los cuidadores con titulo Vel optio quisquam iste eveniet et consectetur et.**

Nombre: Silvia Aguirre Segundo  
Nombre: Cristian Toledo  
Nombre: Ariadna Aguilar Hijo  
Nombre: Srta. Julia Galván Hijo  
Nombre: Lic. Margarita Enríquez

[Volver al listado animales](#)

© 2022 CIC. Reservado todos los derechos

## 2.- Unificación de rutas

Si hemos realizado una CRUD de Animales siguiendo las pautas marcadas por Laravel, es decir, creado el controlador con -r comando **php artisan make:controller AnimalController -r**

Crea el contenedor con los métodos:

- index() – listado de animales
- create()- crear un animal
- store(Request \$request) -guardar la información de un animal
- show(\$id) – mostrar la información de un animal
- edit(\$id) – editar la información de un animal
- update(Request \$request, \$id) – guardar la información modificada de un animal
- destroy(\$id) – borrar la información de un animal

las rutas creadas con esos métodos se pueden unificar creando **la ruta resource**

**Route::resource('animales',AnimalController::class);**

Y ahora con el comando de listar las rutas podemos ver que nos ha creado

**php artisan r:list**

Nos crea las siguientes rutas:

Tipo	Ruta	Nombre Ruta / Método
GET HEAD	animales	animales.index
POST	animales	animales.store
GET HEAD	animales/create	animales.create
GET HEAD	animales/{animal}	animales.show
PUT PATCH	animales/{animal}	animales.update
DELETE	animales/{animal}	animales.destroy
GET HEAD	animales/{animal}/edit	animales.edit