

Tasca M11

Problema:

Descripció

En aquesta pràctica aprendràs a crear peticions HTTP amb un CRUD (Create, Read, Update, Delete) al complet sobre una base de dades SQL. Per això implementarem un patró l'arquitectura MVC amb repositoris.

Nivell 1

- Exercici 1

Crear una aplicació, per la gestió d'empleats, aplicant el patró de disseny de software MVC(Model-Vista-Controlador).

Definir les rutes principals que tindrà el nostre lloc web. El domini ha de tindre el CRUD al complet (Create, Read, Update, Delete), utilitzant els verbs HTTP associats.

- Exercici 2

Crea les vistes php y associa-la a cadascuna de las rutes .

Defineix els Models de dades, middleware i controllers que consideres necessaris. Recorda que la interacció amb la base de dades es realitzarà per mitjà del ORM Eloquent i per tant els objectes seran persistits únicament en memòria.

- Exercici 3

Crear els formularis necessaris per poder realitzar el CRUD sobre el sistema de gestió d'empleats. Heu de validar que la informació introduïda a l'usuari sigui correcta.

Nivell 2

- Exercici 4

Crea una petició HTTP especial que busqui empleats per feina.

- Exercici 5

Defineix la vista home de benvinguda a l'aplicació e introdueix una navbar.

Crea un sistema que gestiona l'error 404 a nivell de projecte. Podeu utilitzar la funció Resposta i la redirecció de Laravel.

Nivell 3

- Exercici 6

Afegeix la vista de login, registre i recuperació de contrasenya que l'usuari necessita per accedir a l'aplicació.

- Exercici 7

Agafeu les dades d'inici de sessió al sistema de galetes de Laravel. En cadascuna de les vistes has d'incrustar al navbar el nom d'usuari registrat.

Solucion:

1.- Rutas del proyecto:

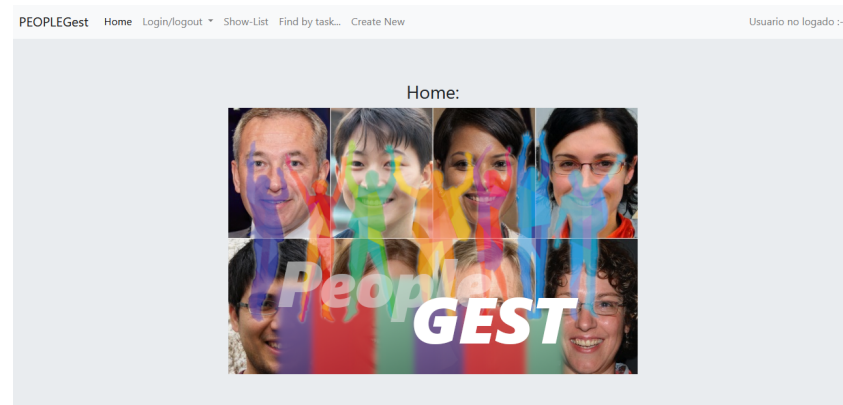
Estas son las rutas que se han generado para el proyecto.

Metodo	Uri:	Name:	Action:	Middleware:
GET	/	home	App\Http\Controllers\peopleGestController@home	
GET	empleados/show	show	App\Http\Controllers\peopleGestController@show	
GET	empleados/create	empleados.create	App\Http\Controllers\peopleGestController@create	
POST	empleados/create	empleados.createStore	App\Http\Controllers\peopleGestController@createStore	
GET	empleados/edit/{id}	empleados.edit	App\Http\Controllers\peopleGestController@edit	
PUT	empleados/edit/{id}	empleados.editStore	App\Http\Controllers\peopleGestController@editStore	
GET	empleados/show/{id}	empleados.showid	App\Http\Controllers\peopleGestController@showid	
DELETE	empleados/delete/{id}	empleados.destroy	App\Http\Controllers\peopleGestController@destroy	
GET	empleados/find	empleados.find	App\Http\Controllers\peopleGestController@find	

Ademas se añaden todas las rutas creadas por breeze.

4.- Crud + Validacion:

Se crean todos los formularios para obtener un CRUD:



Pagina de Home: En la parte superior se muestra el **navBar** de navegación con las opciones de menu:

Home: pagina de inicio.

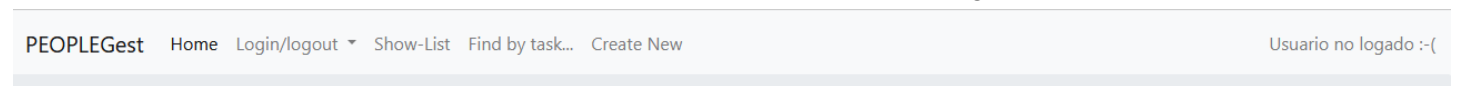
Menu login con: **Register new user**, **login/logout**, y **password recovery**.

Show-list: para mostrar todos los registros,

Find by task: para buscar por una tarea específica y

Create New: para generar un nuevo registro.

Además indica el estado del usuario en la parte derecha del navbar: no logado/nombre de usuario.



PEOPLEGest Home Login/logout Show-List Find by task... Create New Usuario no logado :-)

Employee list:

#	Name:	Surname:	Date of birth:	Phone number:	Task:	Department:	Detail:	Edit:	Delete:
1	Jose	Ramirez	1970-01-15	931111111	programador junior	sistemas	Show	Edit	Delete
2	Antonio	Pascual	1971-02-15	932222222	programador senior	sistemas	Show	Edit	Delete
3	Maria	Mengual	1972-03-15	933333333	responsable	sistemas	Show	Edit	Delete
4	Rosa1	Vilafranca1	1995-07-20	888888881	programador	Sistemas	Show	Edit	Delete
6	Jose Maria	Fernandez	1995-11-15	999999999	abc abc abc defe casa	mantenimiento	Show	Edit	Delete

Pagina de **Employee-list**: muestra todos los registros. En esta página tenemos las opciones de **Show**: mostrar el detalle del registro, **Edit**: editar el registro, y **Delete**: para borrarlo.

Employee detail:

id: 1

Name: Jose

Surname: Ramirez

Date of birth: 1970-01-15

Phone number: 931111111

Task: programador junior

Department: sistemas

Additional info: Lorem ipsum dolor sit amet consectetur adipisicing elit. In doloremque ex labore expedita saepe voluptate quam, minus perspiciatis repudiandae omnis.

Pagina **Employee detail**: para mostrar el detalle especifico del empleado seleccionado.

Edit an employee:

Ruta envio form por post: http://192.168.1.12/m1101/public/index.php/empleados/edit/1

Employee ID: 1

Name: *(min 2, max 50)

Jose

Surname: *(min 2, max 50)

Ramirez

Date of birth: *(fecha valida)

15 / 01 / 1970

Phone number: *(9 carac.)

931111111

Task: *(min 2, max 50)

programador junior

Department: *(min 2, max 50)

sistemas

Additional info: *(min 0, max 100)

Lorem ipsum dolor sit amet consectetur adipisicing elit. In doloremque ex labore expedita saepe voluptate quam, minus perspiciatis repudiandae

Fields with * are mandatory

Save

Pagina **Edit an employee**: para editar el registro seleccionado. El id no se puede cambiar. El resto de datos, se muestra el dato a modificar. También se realiza la validación. En caso de error de validación, se muestra un mensaje de advertencia del campo incorrecto, y se guardan los valores anteriores para que el usuario no tenga que introducirlos de nuevo.

Employee eliminated:

El registro con id: 4 ha sido eliminado correctamente...

Regresar al listado de empleados.

En la pagina **Employee eliminated** nos muestra una vista que nos indica que registró ha sido eliminado correctamente.

Create New Employee:

Info: Ruta envio form por post: <http://192.168.1.12/m1101/public/index.php/empleados/create>

Name: *(min 2, max 50)

Surname: *(min 2, max 50)

Date of birth: *(fecha valida)

Phone number: *(9 carac.)

Task: *(min 2, max 50)

Department: *(min 2, max 50)

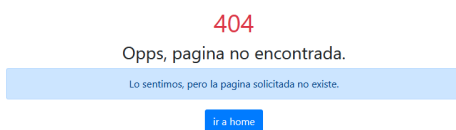
Additional info: *(min 0, max 100)

Fields with * are mandatory

[Save](#)

Pagina **Create new Employee**: para crear un nuevo registro, que también realiza las validaciones.

5.- Error 404 personalizado:



Se implementa la pagina de error personalizada, desde la que podemos regresar a home.

6.- Inicio registro/inicio de sesion/restauracion:

Se implementa el paquete **Breeze**, que realiza todas estas tareas de Login/logout/restauración de password por email.



Dashboard

Welcome... jorge
You have logged in.

[Regresar al home...](#)

Se realiza una modificación de: `html/M1001/resources/views/dashboard.blade.php` para que aparezca el mensaje de bienvenida, el nombre de **usuario logeado** y un link a la página de home.

7.- Cookies

Se añaden gestión de cookies del usuario logeado para que en el navbar se muestre si el usuario esta logado.

Usuario logado: jorge

Mensaje de ejemplo del navbar cuando se está logueado.

Usuario no logado :-(

Mensaje del navbar cuando no se está logado.