

EJERCICIO 10: Validación de formularios

1. En el archivo web.php, creamos una ruta tipo get llamada “mostrarFormu” que redirige a un objeto vista llamado “FormularioUsuario”.

```
Route::get('/mostrarFormu', function () {  
    return view("FormularioUsuario");  
});
```

2. En la vista “FormularioUsuario”, creamos el formulario asignando a los campos “DNI, email y código” un elemento “@error” que mostrará el mensaje de error si se produce.

```
<form action="/registrarFormu" method="POST">  
    @csrf  
    <table>  
        <tr><td><label>Nombre:</label></td></tr>  
        <tr><td><input name="nombre" type="text"></td></tr>  
        <tr><td><label>Apellido:</label></td></tr>  
        <tr><td><input name="apellido" type="text"></td></tr>  
        <tr><td><label>DNI:</label></td></tr>  
        <tr><td><input name="dni" type="text"></td><td><div id="alerts"></div></td></tr>  
        @error('dni')  
            {{-- {{alertError($message)}} --}}  
            <tr><td><div class="alert alert-danger" style="color: red">{{ $message }}</div></td></tr>  
        @enderror  
        <tr><td><label>email:</label></td></tr>  
        <tr><td><input name="email" type="text"></td><td id="alerts"></td></tr>  
        @error('email')  
            <tr><td><div class="alert alert-danger" style="color: red">{{ $message }}</div></td></tr>  
            {{-- {{alertError($message)}} --}}  
        @enderror  
        <tr><td><label>Fecha:</label></td></tr>  
        <tr><td><input name="fecha" type="date"></td></tr>  
        <tr><td><label>Codigo seguridad</label></td><td id="alerts"></td></tr>  
        <tr><td><input name="codigo" type="text"></td></tr>  
        @error('codigo')  
            <tr><td><div class="alert alert-danger" style="color: red">{{ $message }}</div></td></tr>  
            {{-- {{alertError($message)}} --}}  
        @enderror  
        <tr><td><input type="submit" value="Mostrar"></td></tr>  
    </table>  
</form>
```

3. Una vez pulsado el botón mostrar, redirigirá a una ruta llamada “registrarFormu”. Esta ruta llama a la función “comprobarFormu” del controlador.

```
Route::post('/registrarFormu',[Controlador1::class,"comprobarFormu"]);
```

4. En la función “comprobarFormu”, creamos una instancia de un “Validator” y con el metodo “make” pasamos como campos el “request” que se va a validar, las reglas de validación con los required y los regex y, los mensajes de error personalizados dependiendo del tipo de error producido.

```
1 reference | 0 overrides
public function comprobarFormu(Request $req){

    $validator = Validator::make($req->all(), [
        'dni' => 'required|regex:/^[0-9]{8}[A-Z]$/',
        'email' => array('required','regex:/^ik012108[0-9]{2}@plaiaundi.(com|net)$/'),
        'codigo' => 'required|regex:/^ab+\d*cc+\d*$/',
    ], $messages = [
        'dni.required' => 'Necesitamos saber tu DNI',
        'email.required' => 'Necesitamos saber tu email',
        'codigo.required' => 'Introduce un codigo',
        'dni.regex' => 'Introduzca un :attribute valido',
        'email.regex' => 'Introduzca un :attribute valido',
        'codigo.regex' => 'Introduzca un :attribute valido',
    ]->validate());
}
```

5. Casos:

- No introducir nada en ningún campo:

Nombre:

Apellido:

DNI:

Necesitamos saber tu DNI

email:

Necesitamos saber tu email

Fecha:

Codigo seguridad

Introduce un codigo

- DNI mal introducido:

DNI:

Introduzca un dni valido

- Email mal introducido

Introduzca un email valido

- Código mal introducido

Codigo seguridad

ab

Introduzca un codigo valido

Mostrar