



# UNIVERSIDAD GERARDO BARRIOS

SEDE CENTRAL SAN MIGUEL /CENTRO REGIONAL USULUTAN

Datos Generales	
Facultad	Ciencia y tecnología
Asignatura	Programación Computacional IV
Docente	Inga. Gisela Espinoza
No. de Unidad	2
Contenido a desarrollar	Validación de datos (rutas, vistas, formularios y controladores)

## COMPETENCIAS DE LA ASIGNATURA

Aplicar los lenguajes de programación y tecnologías para la Web con el propósito de utilizarlos en el desarrollo de aplicaciones basándose en estándares internacionales de desarrollo de software trabajando de forma individual y colaborativa.

# CONTENIDO

## Validaciones en Laravel

Existen varias formas de validar una aplicación para cubrir aspectos de seguridad como SQL Injection, ataques XSS o CSRF, algunas de ellas son:

- Validación de lado del cliente (Javascript y etiquetas HTML).
- Validación a nivel de base de datos (Migraciones y modelos).
- Validación de formularios (Request).

### Validación del lado del cliente

Se puede validar que los campos de un formulario sean requeridos al agregar el atributo **required**

```
<form action="#">
  Username:
  <input type="text" name="username" required>
  <input type="submit">
</form>
```

El atributo required es un atributo booleano. Cuando esta presente, este especifica que un campo debe ser rellenado antes de ser enviado el contenido del formulario.

El atributo required trabaja con los siguientes tipos de input:

- text
- search
- url
- tel
- email
- password
- data pickers
- number
- checkbox

## El atributo pattern

Con required solo se necesita de cualquier valor en el elemento <input> para ser válido, pero utilizando el atributo pattern en conjunto, se logra que se verifique no solo la presencia de un valor, sino que este valor debe contener un formato, una longitud o un tipo de dato específico. Esto último se logra definiendo un patrón con expresiones regulares.

```
<label for="tel">Teléfono 10 dígitos empezando por 503</label>
<input type="text" pattern="^503\d{8}$">
```

## Validación de formularios con plugins JQuery

El mejor plugin JQuery para validar formularios es formvalidation. Sin embargo formvalidation es un plugin que tiene un costo dependiendo la licencia que se requiera utilizar. Formvalidation es compatible con los formularios de los frameworks css más populares:

- Bootstrap
- Foundation
- Pure
- Semantic UI
- Otros

## Validación del lado del servidor.

Pero podemos especificar que solo queremos aceptar números haciendo una condición de la siguiente manera:

```
web.php x
routes > web.php
23
24 Route::get('/usuario/{id}', function ($id=0) {
25     return 'ID Usuario: '. $id;
26 })->where('id', '[0-9]+');
```

En este ejemplo estamos haciendo uso de expresiones regulares el nos brinda un mundo de posibilidades.

Rutas con parámetros opcionales:

```
web.php x enlaces.tutoriales
routes > web.php
23
24 Route::get('/usuario/{id}/{nombre}/{apellido?}', function ($id=0, $nombre = '', $apellido= '') {
25     return 'ID USuario: '. $id. ' ; Nombre: '. $nombre .'; Apellido: '.$apellido;
26 })->where('id','[0-9]+');
```

Y tendremos como salida los siguiente:

```
localhost:8000/usuario/683/Luis/Hernandez
```

ID USuario: 683; Nombre: Luis; Apellido: Hernandez

Pero si omitimos el "apellido" tendremos la siguiente salida:

```
localhost:8000/usuario/683/Luis
```

ID USuario: 683; Nombre: Luis; Apellido:

Si deseamos establecer posibles valores a los parámetros de las rutas, por ejemplo, en "acción" que las posibilidades sean: créate, actualizar o eliminar tendríamos que ponerlo de la siguiente manera:

```
web.php x enlaces.tutoriales
routes > web.php
23
24 Route::get('/usuario/{accion}/{id}/{nombre}/{apellido?}', function ($accion='', $id=0, $nombre = '', $apellido= '') {
25     return 'ID USuario: '. $id. ' ; Nombre: '. $nombre .'; Apellido: '.$apellido.'; ACCION: '.$accion;
26 })->where(['id'=>'[0-9]+', 'accion'=>'create|update|delete']);
```

Ahora iremos más a fondo con la validación.

Retomaremos el ejemplo anterior y vamos a validar el campo nombre de la siguiente manera:

```
web.php x
routes > web.php
23
24 Route::get('/usuario/{accion}/{id}/{nombre}/{apellido?}', function ($accion='', $id=0, $nombre = '', $apellido= '') {
25     return 'ID USuario: '. $id. ' ; Nombre: '. $nombre .'; Apellido: '.$apellido.'; ACCION: '.$accion;
26 })->where(['id'=>'[0-9]+', 'accion'=>'create|update|delete', 'nombre'=>'[a-zA-ZñÑ ]+']);
```

Fíjate que ahora hemos puesto una expresión regular mas compleja que solo acepte caracteres comprendidos entre "a-z" en minúsculas y "A-Z" en mayúsculas y también puedes ver que hemos puesto la "ñ" tanto en minúscula, como la "Ñ" en mayúscula pues no forma parte del alfabeto.

Y tendremos la siguiente salida, pues en el nombre he puesto un numero entonces no coincide con la expresión regular que hemos hecho:

localhost:8000/usuario/create/683/Luis2/Hernandez

404 | Not Found

Ahora hemos validado el DUI para que acepte solo 10 caracteres los primero 8 números y un "-" y un numero al final:

```
web.php x
routes > web.php
23
24 Route::get('/usuario/{accion}/{id}/{nombre}/{apellido?}/{dui}', function ($accion='', $id=0, $nombre = '', $apellido= '', $dui='') {
25     return 'ID Usuario: '. $id. ' Nombre: '. $nombre .'; Apellido: '.$apellido.' ACCION: '.$accion.' Tel: '. $dui;
26 }->where(['id'=>'[0-9]+', 'accion'=>'create|update|delete', 'nombre'=>'[a-zA-ZñÑ ]+', 'dui'=>'[0-9]{8}-[0-9]{1}']);
27
```

Si establecemos un valor no valido tendremos lo siguiente:

localhost:8000/usuario/create/683/Luis/Hernandez/0000000000

404 | Not Found

Pero si establecemos los valores correctos entonces mostrara los siguiente:

localhost:8000/usuario/create/683/Luis/Hernandez/12345678-9

ID USuario: 683; Nombre: Luis; Apellido: Hernandez; ACCION: create; Tel: 12345678-9

Y así podemos validar cualquier tipo de información que deseemos.

ACTIVIDAD DE EVALUACIÓN DE LA SEMANA	
Nombre de la Actividad	Validación de datos
Tipo de Actividad	Evaluada
Tipo de Participación	Grupal
Competencia específica de la asignatura	Aplicar los lenguajes de programación y tecnologías para la Web con el propósito de utilizarlos en el desarrollo de aplicaciones basándose en estándares internacionales de desarrollo de software trabajando de forma individual y colaborativa.
Instrucciones para la actividad	En grupos de trabajo de 3 integrantes cree una aplicación web que valide los campos del estudiante, como el código, nombre, dirección, teléfono, y email.
Fecha de Entrega	Domingo al final de la semana
Instrumento de evaluación	Rúbrica
Ponderación	50% laboratorio II

RECURSOS COMPLEMENTARIOS		
Recurso	Título	Cita referencial
Web	Validación de datos	<a href="https://styde.net/validacion-de-datos-en-laravel-6/">https://styde.net/validacion-de-datos-en-laravel-6/</a>
Web	Validación en Laravel	<a href="https://diegooo.com/validacion-en-laravel/">https://diegooo.com/validacion-en-laravel/</a>