



Universidad de
Oviedo



VALIDACIÓN

Enol García González
Universidad de Oviedo
10 de noviembre de 2025

VALIDACIÓN DE CAMPOS

La validación de campos es algo muy importante durante el desarrollo. Nos ayuda a garantizar que los datos que va a procesar nuestra aplicación son coherentes.

EJEMPLOS DE VALIDACIÓN

- **Email.** Donde se vaya a procesar un email debería haber una cadena del tipo “usuario@dominio.com”.
- **Tarjetas de crédito.** 16 dígitos, de los que se pueden validar dos cosas:
 - Emisor. Los 6 primeros dígitos identifican a la entidad emisora de la tarjeta. Para las más comunes como Visa o Mastercard con el primer número ya es suficiente.
 - Validez del número. Mediante el algoritmo de Luhn se puede verificar si es un número válido para una tarjeta. para ello se multiplican por los los caracteres impares y se suman todos los dígitos. El resultado debe ser múltiplo de 10.
Por ejemplo, con la tarjeta 0123 4567 8910 1112 se sumaria $0 + 1 + 4 + 3 + 8 + 5 + 1 + 2 + 7 + 1 + 6 + 9 + 2 + 0 + 2 + 1 + 2 + 2 = 56$. Como no es múltiplo de 10, no puede existir ninguna tarjeta con ese número

EJEMPLOS DE VALIDACIÓN

- **Matrícula.** Múltiples reglas:
 - *P-0000-LL*, P tiene que ser un código de ciudad de matriculación válido.
 - *0000-LLL*.
 - *PPP-0000-L* Para vehículos especiales. PPP tiene que representar el organismo: PGC (Guardia civil), CNP (Policía nacional), ET (Ejercito de tierra), EA (Ejercito del aire), FN (Fuerzas navales), FAE (Cuarteles internacionales), etc.
 - *P-0000-LLL* para matriculaciones provisionales o temporales. P puede ser: P (Provisional a particulares), S (Vehículo en pruebas no matriculado) o V (Vehículo en pruebas matriculado), E (Especial, p.e. tractor o cosechadora), R (Remolque), H (Histórico), C (Ciclomotores).
 - *PP-000-000* para algunas matriculaciones especiales según lo que se indique en PP: CD (Cuerpo Diplomático), CC (Cuerpo Consular), TA (Técnicos Administrativos), OI (Organismos Internacionales).

EJEMPLOS DE VALIDACIÓN

- **DNI.** Donde se vaya a introducir un DNI español se debe introducir.
 - Una cadena formada por 7 u 8 dígitos y una letra.
 - La letra es un carácter de validación. Al hacer la división con resto del número entre 23, el resto debería indicar la letra según la tabla. Por ejemplo al número 12345678 le corresponde la letra Z porque $12345678 \% 23 = 14$

0	T	6	Y	12	N	18	H
1	R	7	F	13	J	19	L
2	W	8	P	14	Z	20	C
3	A	9	D	15	S	21	K
4	G	10	X	16	Q	22	E
5	M	11	B	17	V		

¿DÓNDE REALIZAR LA VALIDACIÓN?

La validación se puede realizar en dos momentos:

- En el cliente antes de hacer la petición al servidor.
- En el servidor al recibir la petición.

¿DÓNDE REALIZAR LA VALIDACIÓN?

La validación en el servidor SIEMPRE es OBLIGATORIA.

La validación en el cliente es opcional. Se hace solo por cuestiones de usabilidad.

OJO: Cuidado con hacer validaciones excesivas.

Ejemplo de matricula diplomática francesa

¿COMO IMPLEMENTAR LA VALIDACIÓN?

- ① En proyectos como JSF donde el código se escribe utilizando una plantilla:
 - **NORMALMENTE** el motor de plantillas/vistas separa el código de validación en el servidor y en el cliente.
 - Si el motor de plantillas no hace un gran procesamiento de la plantilla hay que implementarlo a mano como en la segunda opción.
 - Si el motor si lo hace, con usar las etiquetas apropiadas es suficiente.
- ② En proyectos hechos a mano con HTML+JS o en motores de plantillas que no procesen las validaciones:
 - Los métodos del controlador tienen que tener las comprobaciones necesarias con if al principio del método.

VALIDADORES DE JSF

- Las etiquetas de tipo input
 - Todas tienen la capacidad de la validación required (implementada en el servidor).
 - Pueden ser con una única etiqueta (sin validaciones adicionales) o con etiqueta de apertura y de cierre. En el caso de que tenga dos etiquetas de apertura y cierre se coloca en medio las validaciones que se quieran hacer.
 - Pueden tener los atributos requiredMessage y validatorMessage para detallar el mensaje que mostrar en caso de que no se cumpla alguna condición.

VALIDADORES DE JSF

- Algunos validadores ya vienen implementados:
 - `f:validateLength` `minimum="..."` `maximum="..."`
 - `f:validateLongRange` `minimum="..."` `maximum="..."`
 - `f:validateDoubleRange` `minimum="..."` `maximum="..."`
 - `f:validateRegex` `pattern="..."`
- Se pueden crear validadores personalizados.
 - Crear una clase que implemente la interfaz `Validator`. Eso implica que la clase que creamos tendrá un método con la siguiente firma:
 - `public void validate(FacesContext context, UIComponent component, Object value)`
 - El contexto es el contexto de la aplicación para poder recuperar información sobre ella,
 - El `UIComponent` es el input sobre el que se está haciendo la validación
 - El `Object` es el valor introducido por el usuario.

VALIDADORES DE JSF

- Se pueden crear validadores personalizados.
 - Crear una clase que implemente la interfaz Validator. Eso implica que la clase que creemos tendrá un método con la siguiente firma:
 - `public void validate(FacesContext context, UIComponent component, Object value)`
 - El contexto es el contexto de la aplicación para poder recuperar información sobre ella,
 - El UIComponent es el input sobre el que se está haciendo la validación
 - El Object es el valor introducido por el usuario.
 - El método lanzara una excepción de tipo `ValidatorException` si la entrada no es válida.
- Anotar la clase con la etiqueta `@Validator("NombreDelValidador")`
- Utilizar la etiqueta `f:validator validatorId="NombreDelValidador"` en el campo deseado