

RESUMEN FORMULARIOS (2ª parte)

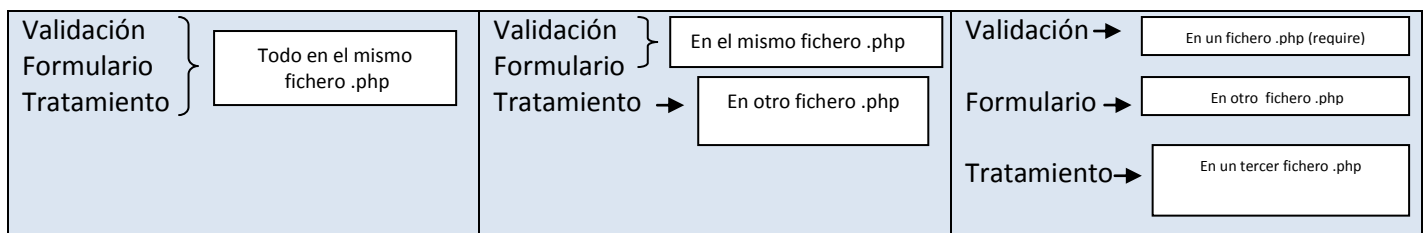
- Atributos: https://www.w3schools.com/html/html_forms_attributes.asp
- Elementos: https://www.w3schools.com/html/html_form_elements.asp

Validación datos

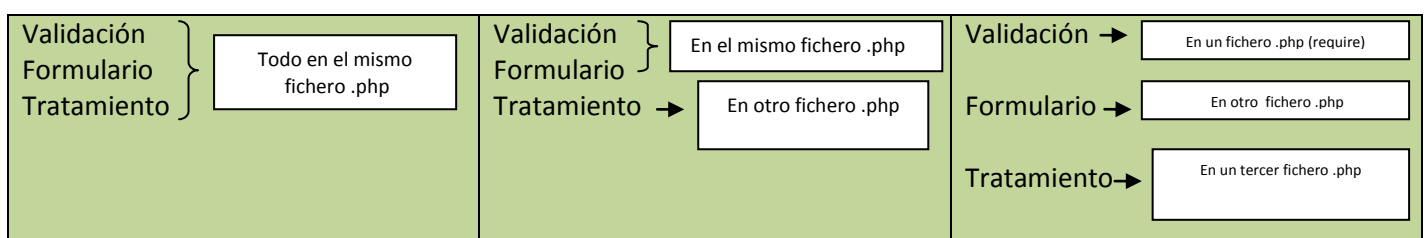
Puede realizar en:

- **El propio formulario.** Muchos tipos de elementos (contraseña, email, texto, fecha, teléfono,...) disponen del atributo *pattern*, que acepta un patrón o expresión regular (https://www.w3schools.com/tags/att_input_pattern.asp)
- **En el lado del cliente web** (utilizando Javascript y expresiones regulares)
- **En el lado del servidor web** (evitarla si es posible ya que con cualquiera de las dos formas anteriores se evitan accesos innecesarios al servidor). En este caso lo habitual:
 - Diseñar funciones para la validación (la función `preg_match()` para validad string es muy típica; maneja expresiones regulares).
 - La sanitización de los datos (eliminar caracteres no deseados utilizando funciones como `htmlspecialchars()`)
 - La utilización de filtros específicos de php que permiten validar o satinizar datos utilizando la función `filter_var()` entre otras (<https://diego.com.es/filtrado-de-datos-de-entrada-en-php>)

Ejemplos de estructuración de una aplicación web (hasta ahora)



Cliente web (sólo una pestaña: introducción de datos y muestra de resultados tras el tratamiento)



Cliente web (2 pestañas; una para introducción de datos y otra para mostrar resultados tras tratamiento utilizando el atributo target)