



ANGULARJS

FORMULARIOS

FORMULARIOS

- ▶ ANGULAR PERMITE SABER SI LA INFORMACIÓN DE UN FORMULARIO HTML
 - ▶ HA SIDO MODIFICADA O NO
 - ▶ SUS CAMPOS SON VÁLIDOS O NO
- ▶ CUANDO EN NUESTRO HTML CREAMOS UN FORM, PODEMOS ACCEDER A EL DESDE EL SCOPE:
 - ▶ `$scope.nombreForm`
- ▶ EL OBJETO ACCEDIDO ES UNA INSTANCIA DE LA CLASE FORMCONTROLLER DE ANGULAR

ENTRADA DE USUARIO - FORMULARIOS

- ▶ PARA SABER SI UN FORMULARIO HA SIDO MODIFICADO O NO, CONSULTAMOS SU ATRIBUTO \$DIRTY
 - ▶ `$scope.form.$dirty`: VALE TRUE SI EL FORMULARIO HA SIDO MODIFICADO YA. TAMBIÉN ES APLICABLE SOBRE CUALQUIERA DE SUS CAMPOS . SU OPUESTO ES `$pristine`
 - ▶ [EJEMPLO EJ12](#)

ENTRADA DE USUARIO - FORMULARIOS

- ▶ PARA SABER SI LOS CAMPOS DE UN FORMULARIO SON O NO VÁLIDOS, ESTÁ EL ATRIBUTO \$VALID / \$INVALID
- ▶ ANGULAR DISPONE DE 9 TIPOS DE VALIDACIONES PARA CAMPOS, EN FUNCIÓN DEL TIPO DE CAMPO
 - ▶ EMAIL: EL CAMPO DEBE TENER EL FORMATO DE UN CORREO ELECTRÓNICO (type=email)
 - ▶ MAX: EL CAMPO DEBE TENER UN VALOR MÁXIMO (type=number)
 - ▶ MAXLENGTH: EL CAMPO DEBE TENER UN N° MÁXIMO DE CARACTERES (type=text, textarea, number)
 - ▶ MIN: EL CAMPO DEBE TENER UN VALOR MÍNIMO (type=number)
 - ▶ MINLENGTH: EL CAMPO DEBE TENER UN N° MÍNIMO DE CARACTERES (type=text, textarea, number)
 - ▶ NUMBER: EL CAMPO DEBE SER UN NÚMERO
 - ▶ PATTERN: EL CAMPO DEBE SEGUIR UNA EXPRESIÓN REGULAR
 - ▶ REQUIRED: EL CAMPO ES REQUERIDO
 - ▶ URL: EL CAMPO DEBE TENER EL FORMATO DE UNA URL

ENTRADA DE USUARIO - FORMULARIOS

- ▶ PARA SABER SI UNA VALIDACIÓN HA FALLADO, HAY 2 MÉTODOS
 - ▶ COMPROBAR POR CAMPOS (UNO A UNO)
 - ▶ `$scope.form.campo.$error.*`
 - ▶ COMPROBAR POR TIPO DE VALIDACIÓN
 - ▶ `$scope.form.$error.*`
- ▶ [EJEMPLO EJ12_PATTERNS](#)

ENTRADA DE USUARIO - FORMULARIOS

- ▶ SE PUEDE ESPECIFICAR `novalidate` EN LA ETIQUETA DE FORM PARA EVITAR QUE EL NAVEGADOR HAGA SUS PROPIAS VALIDACIONES Y CHOQUEN CON LAS DE ANGULARJS
- ▶ LOS MENSAJES PARA INFORMAR AL USUARIO CUANDO HA INTRODUCIDO UN CAMPO MAL SE SUELEN DEJAR A IMPLEMENTAR LIBREMENTE POR EL DESARROLLADOR, CON LA TECNOLOGÍA QUE PREFIERA

ENTRADA DE USUARIO - FORMULARIOS

- ▶ PODREMOS TAMBIÉN CREAR NUESTRAS PROPIAS VALIDACIONES PARA CAMPOS DE FORMULARIOS RELACIONADOS CON EL MODELO (QUE TENGAN EL CAMPO `ng-model`)
- ▶ LA EXISTENCIA DE `ng-model` COMO ATRIBUTO DE LA DIRECTIVA ES IMPORTANTE. LO QUE VALIDAMOS DESDE ANGULAR ES EL VALOR DEL MODELO, NO EL ELEMENTO DEL DOM
- ▶ TENDREMOS QUE CREAR UNA DIRECTIVA ESPECIAL PARA APLICARLE AL CAMPO DE FORMULARIO QUE QUEREMOS VALIDAR

ENTRADA DE USUARIO - FORMULARIOS

- ▶ LA DIRECTIVA QUE CREEMOS HA DE CUMPLIR DOS CONDICIONES
 - ▶ require: 'ngModel'. CON ESTO FORZAMOS A QUE EL ELEMENTO DEL DOM TENGA UN ATRIBUTO NG-MODEL
 - ▶ TENDREMOS QUE IMPLEMENTAR LA VALIDACIÓN DENTRO DE LA FUNCIÓN APUNTADA POR link

ENTRADA DE USUARIO - FORMULARIOS

- ▶ USAREMOS EL CUARTO ARGUMENTO DE LA FUNCIÓN link: EL ARGUMENTO controller
 - ▶ EN CONCRETO, USAREMOS EL CAMPO \$parser DEL OBJETO PASADO COMO CUARTO ARGUMENTO.
 - ▶ \$parser CONTIENE UN ARRAY DE FUNCIONES A EJECUTAR CADA VEZ QUE SE ALMACENA EL VALOR EN EL MODELO
 - ▶ AÑADIMOS NUESTRA PROPIA FUNCIÓN, DESDE DONDE HAREMOS LA VALIDACIÓN, Y PODREMOS ESTABLECER EL VALOR DE LOS FLAGS \$valid Y \$error DEL CAMPO DEL FORMULARIO QUE ESTEMOS VALIDANDO

ENTRADA DE USUARIO - FORMULARIOS

- ▶ [EJEMPLO EJ12_CUSTOMVAL](#)
- ▶ EN ESTE EJERCICIO SE VE EL USO DE LOS ATRIBUTOS \$PARSER Y \$FORMATTER DEL CUARTO ARGUMENTO DE LA FUNCIÓN \$LINK
 - ▶ \$PARSER ES UN ARRAY DE FUNCIONES A EJECUTAR CADA VEZ QUE SE CAMBIA UN OBJETO DEL MODELO DESDE LA VISTA (EJ: ENTRADA DE USUARIO). VIEW → MODEL
 - ▶ \$FORMATTER ES UN ARRAY DE FUNCIONES A EJECUTAR CADA VEZ QUE UN CAMBIO EN UN OBJETO DEL MODELO REQUIERE QUE SE ACTUALICE LA VISTA (CAMBIOS EN EL DOM DEBIDOS A CAMBIOS EN EL MODELO). MODEL → VIEW
- ▶ SE APRECIA MEJOR ESTE EFECTO EN EL [EJEMPLO EJ13](#)

ENTRADA DE USUARIO - FORMULARIOS

- ▶ SI NO QUEREMOS COMPLICARNOS IMPLEMENTANDO DIRECTIVAS, TAMBIÉN PODEMOS USAR EL MÓDULO `ui.directives`, DE ANGULARUI
- ▶ TENEMOS UN EJEMPLO EN [EJEMPLOS_LIBRERIAS_EXTERNAS/ANGULARUI/VALIDATE_FORMS](#)