

## Relación 5: Introducción de información.

Crear la carpeta aplicacion/relacion5 dentro de la cual se creará un fichero index.php donde se hará la comprobación de todo.

Se deben tener en cuenta todos los requisitos indicados en prácticas anteriores como modelo-vista-contralador, comentarios, barra de ubicación, uso de las funciones mb\_, definición de tipos en funciones, control de errores, etc

Validación de datos. Todo dato enviado en un formulario debe ser validado en el servidor. Para validar los datos se pueden usar dos métodos:

- Comprobar que es de un tipo determinado mediante las funciones is\_xxx (is\_array, is\_integer, etc, no hace falta si definimos el tipo en parámetros de funciones/valor devuelto) y que su valor cumple una condición bien con comparaciones (ej > o <=) o con expresiones regulares.
- Validar que el contenido de una variable cumple con una condición con funciones de la librería FILTER (<http://php.net/manual/es/book.filter.php>). Además, esta librería nos permite sanear el valor (eliminar caracteres que no sean correctos).

1.- Crea dos librerías de funciones para validar variables de distintos tipos. Llamar las librerías val\_normal.php y val\_filter.php. Cada librería corresponde con una forma de validación de las indicadas anteriormente. Para que podamos usar las dos librerías a la vez le vamos a indicar a cada una un espacio de nombres distinto (VALNORMAL y VALFILTER respectivamente). Si dentro de la librería FILTER no hay una validación concreta para un caso determinado, copiar lo hecho en la librería sin FILTER. Cada librería tendrá las siguientes funciones:

- function validaEntero(int &\$var, int \$min, int \$max, int \$defecto):bool Esta función comprueba que \$var contiene un entero cuyo valor está entre \$min y \$max. En \$var se devuelve el entero saneado (en caso de no cumplir las condiciones devuelve \$defecto). La función devuelve true si es correcto y false en caso contrario.
- function validaReal(float &\$var, float \$min, float \$max, float \$defecto):bool Esta función comprueba que \$var contiene un real cuyo valor está entre \$min y \$max. En \$var se devuelve el real saneado (en caso de no cumplir las condiciones devuelve \$defecto). La función devuelve true si es correcto y false en caso contrario.
- function validaFecha(string &\$var, string \$defecto):bool Esta función comprueba que \$var contiene una fecha correcta en el formato dd/mm/aaaa. En \$var se devuelve la fecha saneada (2 cifras para día y mes y cuatro para año- ej fecha 1/2/2023 es válida aunque se sana y se convierte en 01/02/2023). En caso de no cumplir las condiciones devuelve \$defecto en \$var. La función devuelve true si es correcta y false en caso contrario.
- function validaHora(string &\$var, string \$defecto):bool Esta función comprueba que \$var contiene una hora correcta en el formato hh:mm:ss . En \$var se devuelve la hora saneada (2 cifras para hora, min, segundos - Ej 0:5:1, hora valida aunque se sana y se convierte a 00:05:01). En caso de no cumplir las condiciones devuelve \$defecto en \$var. La función devuelve true si es correcta y false en caso contrario
- function validaEmail(string &\$var, string \$defecto):bool Esta función comprueba que \$var contiene un email correcto en el formato aaaaa@bbbb.ccc. En \$var se devuelve el email saneado (en caso de no cumplir las condiciones devuelve \$defecto). La función devuelve true si es correcto y false en caso contrario.
- function validaCadena(string &\$var, int \$longitud, string \$defecto):bool Esta función comprueba que \$var contiene una cadena de longitud máxima \$longitud. En caso de no cumplir las condiciones se asigna a \$var el valor \$defecto. La función devuelve true si es correcto y false en caso contrario.
- function validaExpresion(string &\$var, string \$expresion, string \$defecto):bool Esta función comprueba que \$var cumple con la expresión regular \$expresion. En caso de no cumplir las

condiciones se asigna a \$var el valor por \$defecto. La función devuelve true si es correcto y false en caso contrario.

- function validaRango(mixed \$var, array \$posibles, int \$tipo=1):bool Esta función comprueba que \$var sea igual a uno de los elementos del array \$posibles (\$tipo=1) o a una de las claves del array \$posibles (\$tipo=2). La función devuelve true si es correcta y false en caso contrario.

Una vez creadas y comprobadas las librerías elegimos una de ellas y la copiamos a \scripts\librerias. La llamaremos validacion.php y le quitamos el espacio de nombres. Ya la tenemos preparada para reutilizarla.

2.- Crear un formulario para introducir los siguientes elementos:

- Nombre: cadena máximo 25 caracteres. No puede estar vacío ni empezar por H. Se debe pasar a mayúsculas.
- Contraseña: cadena máximo 15 caracteres. Con elemento de formulario password. Tiene que cumplir los requisitos: longitud mínima 8 caracteres, incluye al menos una Mayúscula, una minúscula, una cifra y un carácter especial (\$\_-;<>@)
- Fecha de nacimiento: Se debe recoger una única cadena y comprobar que es una fecha válida. (se puede usar un input type date o un input type text, usad el tipo text)
- Fecha de carnet: Se deben recoger tres elementos (día, mes, año) independientes. Cuando se recojan se creará una única fecha compuesta de los tres. Hay que validar que la fecha que es mayor de edad, es decir no es anterior a la fecha de nacimiento + 18 años (usar DateTime, sub y comprobar en el intervalo que te devuelve).
- Hora de levantarse: Se debe recoger una cadena y comprobar que es una hora válida. Se corresponde con la hora del día en la que normalmente se levanta.
- Estado: a partir de varios radiobuttons (1 - estudiante, 2 - en paro, 3 - trabajando, 4 - jubilado). Se recogerá un entero y se mostrará el texto apropiado. Debe indicarse obligatoriamente uno. Inicialmente no hay ninguno marcado. Hay que validar que se ha escogido uno correcto. A efectos de probarlo añadimos una opción que no es válida, por ejemplo, 5 - incorrecta.
- Estudios: varios checkbox que se pueden marcar a la vez (0 - sin estudios, 1 - primaria, 2 - secundaria, 3 - bachillerato, 4 - ciclo formativo, 5 - universitarios). Se recogerá los códigos marcados. Debe seleccionarse al menos uno. Si ha marcado 0 - Sin estudios, no puede haber ningún otro seleccionado (desmarcamos el resto). Hay que validar que se tiene marcado opciones correctas. A efectos de probarlo, añadimos una opción que no es válida, 6 - incorrecta.
- Hermanos: entero que debe estar entre 0 y 20. Por defecto, 0.
- Sueldo: real que debe estar entre 1000 y 150000 €. Valor por defecto 1100€.

Se debe hacer:

- Un formulario que use post como método y que llame al propio fichero.
- Una función separada de la vista para mostrar el formulario a la que se le pasan como parámetros un array con los valores a mostrar y un array con los errores. Se llamará desde la vista.
- Comprobar todos los valores que ha enviado el usuario en la parte del controlador. Se usa la librería que hemos puesto en \scripts\librerias siempre que se pueda.

El funcionamiento deberá ser (según lo indicado en la ESTRUCTURA DE LA APLICACIÓN):

- Se muestra el formulario al usuario para que introduzca los datos.
- Una vez que se envíe el formulario, se comprueban los datos. Si hay algún error, se vuelve a mostrar el formulario con los datos que ha introducido el usuario (datos validados y corregidos) y se muestran los errores.
- Si todos los datos son correctos, no se muestra el formulario y se muestra un resumen con los datos introducidos.