

Instrucciones	Consejos
<ul style="list-style-type: none"> - Crear una carpeta que se llamará “Apellidos_Nombre”, donde guardaréis las soluciones de los problemas - Para cada problema crearéis una carpeta que llamaréis “Problema1”, “Problema2”, etc, donde guardaréis las soluciones. - Una vez hayáis finalizado debéis subir el examen al aula virtual comprimido en un fichero con el formato “Apellidos_Nombre”. - El examen puntúa sobre 10, selecciona los ejercicios como quieras. 	<ul style="list-style-type: none"> - Guardar cada poco el trabajo que estáis realizando. - Mirar la puntuación de los problemas y centraros primero en el aprobado. Luego incluís mejoras. - El programa es personal e intransferible, por lo que no deberíais tener que mirar para nada al monitor del compañero. - Escoger en primer lugar los problemas que tienen fácil solución, asegurad el aprobado lo primero. - Recordad que podéis hacer uso de código que ya tenéis creado

Problema 1 – Filtrado de campos – 5 Ptos

El nombre del fichero con el que arranca el programa será **index.php** En este fichero incluiréis al comienzo un comentario en el que indicaréis para que sirven el resto de ficheros utilizados en la solución.

Se desea realizar una página que nos permita leer datos de personas, para que sean filtrados y luego mostrar el resultado.

Los campos a leer serán los siguientes:

- Fecha de nacimiento: Debe ser una fecha válida y la edad debe ser mayor que 18 años. Se mostrará un mensaje de error diferente para ambas condiciones
- Provincia de nacimiento: Será una de las provincias de España que se encuentran en la lista que se adjunta en el fichero “provincias.php”
- Estado civil se puede indicar o no, dependiendo del campo checkbox seleccionado al efecto. En el caso de seleccionarlo se mostrará una nueva opción que permitirá seleccionarlo.

El programa mostrará la siguiente secuencia de pantallas

- PD1 – Pantalla de datos 1, la primera que se visualiza. Los errores se mostrarán si los datos no son correctos
- PD2 – Pantalla de datos 2, si han seleccionado la opción deseo indicar estado civil.
- PR1 – Pantalla de resultados 1, si han enviado la PD1 sin errores
- PR2 – Pantalla de resultados 2, si han enviado la PD2 sin errores

PD1	Introducción de datos - 1 Fecha de nacimiento (dd/mm/aaaa): <input type="text" value="XXXXXXXXXXXX"/> La fecha debe ser válida / Debe tener más de 18 años Provincia: <input type="text" value="Seleccione provincia ▼"/> Seleccione una <input type="checkbox"/> Deseo indicar mi estado civil <div style="text-align: right;">[Enviar]</div>
PD2	Introducción de datos - 2 Fecha de nacimiento (dd/mm/aaaa): <input type="text" value="15/12/1982"/> Provincia: <input type="text" value="Sevilla ▼"/> Estado civil : <input type="radio"/> Soltero <input type="radio"/> Casado Seleccione un estado civil <div style="text-align: right;">[Enviar]</div>
PR1	Resultado datos - 1 Fecha de nacimiento: 15/2/1982 Provincia: Sevilla – Código 41 Estado civil: No desea indicar su estado civil Volver a introducir datos
PR2	Resultado datos - 2 Fecha de nacimiento: 15/2/1982 Provincia: Sevilla – Código 41 Estado civil: Soltero Volver a introducir datos

Notas:

- Funciones que os pueden resultar interesantes: checkdate(), str_split()
- En el fichero *provincias.php* tenéis la lista de provincias y sus códigos.

Puntuación (sobre 10):

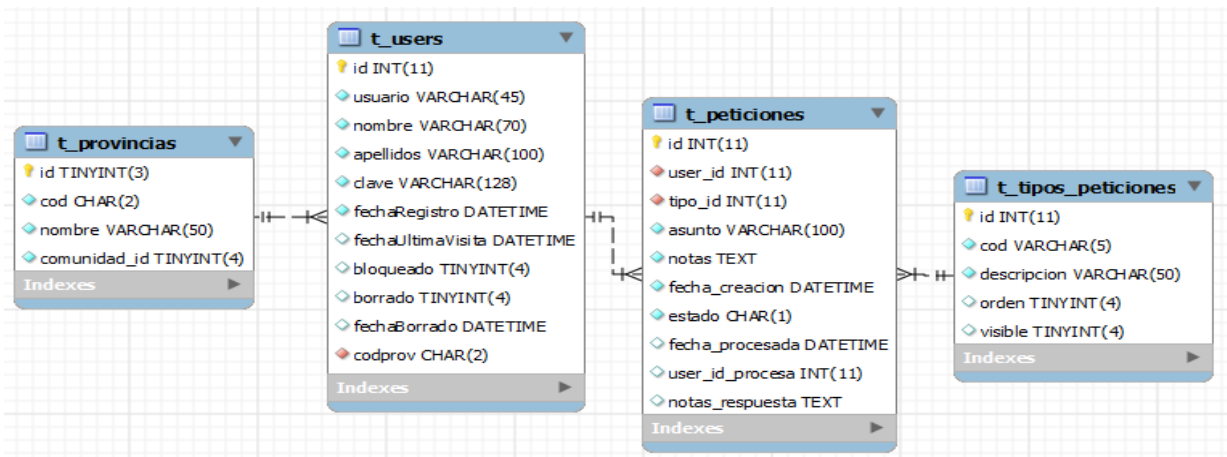
- Incluir provincias en select con código PHP a partir de lista: 1 ptos
- Filtrar edad: 1,5 pto.
- Filtrar provincia: 1 pto.
- Filtrar estado civil: 1 pto.

- Mostrar después de filtrar PR1: 1,5 pts.
- Mostrar después de filtrar PR2: 2 pts.
- Mostrar código de provincia en PR1 y PR2: 0,5 pts
- Programa completo: 1,5 pto.

Problema 2 – Consultas B.D. (8 pts)

El nombre del fichero con el que arranca el programa será **index.php**. En este fichero incluiréis al comienzo un comentario en el que indicaréis para que sirven el resto de ficheros utilizados en la solución.

Utilizando la base de datos *Peticiones* que tenéis en el fichero *Peticiones.sql*, y que almacena datos acorde al siguiente esquema.



Relaciones existentes entre las tablas

t_provincias.cod – t_users.codprov	t_users.id – t_peticiones.user_id
t_peticiones.user_id – t_tipos_peticiones.id	

Se desea realizar una aplicación que nos proporcione listados sobre las peticiones realizadas y nos permita realizar operaciones sobre las mismas. El programa arrancará con una pantalla que nos permitirá mostrar las peticiones filtrando por su estado y/o provincia. Si dejamos los campos en blanco, se entenderá que no tenemos interés en filtrar por ese campo y que se mostrarán todos los valores.

Listado de peticiones
 Estado de la petición: (blanco = todas)
 Provincia: (blanco = todas)
 [**Mostrar peticiones**]

P1

De las peticiones solamente se mostrará el campo “asunto” y “estado”, paginando los resultados en 15 elementos por página.

Una vez listadas las peticiones, se permitirá para las mismas borrarlas u obtener más información. El listado tendrá una presentación como el cuadro de la derecha.

En el título, en caso de estar el listado filtrado, se mostrará la provincia o el estado por el cual se está filtrando.
 Los valores posibles para el campo “estado” son, las siguientes letras, con su significado:

P=>Pendiente, R=>Revisándose, C=>Completa, D=>Denegada

Al pulsar el enlace “Borrar” se borrará la petición (sin confirmar) y se mostrará de nuevo la lista con las condiciones iniciales que la habían generado (Provincia, estado).

LISTADO DE PETICIONES

Filtrado: Provincia=XXXX, Estado=X

Asunto	Estado	Acción
xxxx	Revisándose	Detalle / Borrar
....

P2

[Volver a página inicio](#)

Al pulsar el enlace “Detalle” se mostrarán los siguientes campos de la petición:

- Id: Id de la petición (campo 'id')
- Nombre usuario: Nombre y apellidos del usuario que ha creado la petición (campos 'nombre' y 'apellidos' de la tabla t_users)
- Asunto: Texto del asunto (campo 'asunto')
- Tipo de petición: Descripción de la petición (campo 'descripcion' de la tabla 't_tipos_peticiones')

DETALLE DE PETICIÓN

P3

Id: X
 Usuario: XXXXXXXX
 Asunto: XXXXXX
 Tipo petición: XXXXX

[Volver a página inicio](#) [Volver a lista](#)

El enlace “Volver a página de inicio” nos llevará a la página inicial, mientras que el enlace “Volver a lista” nos mostrará la lista que estábamos viendo aplicando los mismos criterios de filtrado, que teníamos.

Podéis consultar el modelo e imágenes del mismo en la carpeta en la que está ubicada el fichero del ejercicio.

Puntuación (sobre 10):

- Mostrar lista de peticiones (sin paginar): 1 ptos.
- Mostrar descripción en campo estado 0,5 ptos.
- Mostrar lista de peticiones (paginado): 1,5 ptos.
- Filtrar por provincia: 0,5 ptos.
- Filtrar por estado: 0,5 ptos.
- Mostrar texto de filtro en P2: 0,5 pto.
- Borrar petición: 1 ptos.
- Detalle de petición: 2 ptos.
- Mostrar lista con mismos criterios después de borrar o ver detalle: 1 ptos.
- Programa completo: 1,5 pto.

Problema 3 - 2 ptos.

El nombre del fichero con el que arranca el programa será **index.php**. En este fichero incluiréis al comienzo un comentario en el que indicaréis para que sirven el resto de ficheros utilizados en la solución.

Nos han encargado que realicemos el formulario de control de acceso a una aplicación. Para el acceso a dicha aplicación, que aun no está construida, nos han dicho que debemos hacer un formulario en el que se valide el usuario y la clave. Para saber que



usuarios y claves son válidos en nuestra aplicación existirá la función “*EsUnUsuarioValido(\$usuario, \$clave)*” que nos indicará si el usuario y clave son correctos, devolverá True o False. Dicha función aun no está implementada, y luego se realizará. De momento vosotros crearéis una que valide al usuario “juan”, con clave “kk” para permitirnos validar la parte que tenéis que desarrollar.

Nuestra página de entrada a la aplicación, si no no hemos validado previamente mostrará algo parecido a la imagen de la izquierda.

- Ya está dentro -

[Cerrar Sesión](#)

Una vez dentro, se mostrará el mensaje “Ya esta dentro” y se incluirá un enlace que permita al usuario cerrar la sesión y volver a validarse de nuevo.

Cuando se pulse el enlace “Cerrar sesión” se mostrará de nuevo la página de inicio pidiendo los datos de acceso.

En la pantalla inicial se dispone de una opción “Recordar usuario” que si se activa hará que la próxima vez que accedamos a nuestro formulario de entrada se cargará automáticamente el último usuario en el campo. Dicho usuario se recordará solamente durante 5 minutos, pasados los cuales no se considerará. Más adelante cuando la aplicación este funcionando este valor sera de 15 días o un mes, deberéis planificarlo, indicando que habría que cambiar, para esto.

Vuestro desarrollo debe cumplir los requisitos siguientes:

- Cuando se accede a la página inicial “index.php” se mostrará el formulario si el usuario no se ha validado, o el mensaje “Ya está dentro”, con la opción de cerrar la sesión.
- Cuando se pulse el enlace “Cerrar sesión” automáticamente se visualizará la página de inicio con el formulario. Recordad que este enlace posiblemente se incluya en el resto de páginas de la aplicación, y debe seguir mostrando lo mismo.
- Se debe utilizar la función “*EsUnUsuarioValido(\$usuario, \$clave)*” que más adelante se redefinirá.
- Se debe soportar la opción de recordar el usuario, y mostrarlo en el formulario, 5 minutos, que será ampliable a más días durante el normal funcionamiento de la aplicación.

Puntuación (sobre 2 ptos):

- Validar si está dentro o fuera utilizando la función que indican nuestras especificaciones: 0,5 ptos
- Recordar usuario: 0,5 pto.
- Recordar usuario solamente 5 minutos: 0,5 ptos.
- Implementar funcionalidad de cerrar sesión, valido para todas las páginas de la aplicación 0,5 ptos.

Problema 4 – 2,25 ptos.

El nombre del fichero con el que arranca el programa será **index.php**. En este fichero incluiréis al comienzo un comentario en el que indicaréis para que sirven el resto de ficheros utilizados en la solución.

Texto:

Contar palabras Borrar

Se desea realizar una página que nos permita contar el número de palabras que introducimos en un cuadro de texto.

Entenderemos que es una palabra todo aquello que esté separado por un espacio, tabulador, salto de línea, coma, punto, y punto y coma.

Cuando se escribe texto y se pulsa el botón “Contar palabras” se muestra a continuación el número de palabras existentes y la lista de palabras separadas por el símbolo | como se muestra en la siguiente imagen:

El texto introducido se seguirá mostrando en el cuadro de texto, por si queremos realizar modificaciones y volver a contar palabras.
El botón “Borrar” nos mostrará la pantalla vacía que teníamos inicialmente.

Notas:

- La función *preg_split* puede resultaros muy útil para resolver el problema.
- Recordad que algo entre corchetes [] en una expresión regular indicaba opcionalidad. Y que con el + podáis indicar repetición []+.

Puntuación:

- Contar palabras utilizando sólo el espacio como separados 0,5 ptos.
- Contar palabras utilizando los caracteres indicados 0,5 ptos.
- Mostrar lista de palabras y contarlas 1 pto.
- Hacer que se borre el contenido con el botón “Borrar” 0,25 ptos.

Texto:

una, dos, tres, cuatro
cinco

Contar palabras Borrar

Nº de palabras: 5

Palabras:

una | dos | tres | cuatro | cinco