Instrucciones	***Consejos***
- **Crear una carpeta** que se llamará **"Apellidos, Nombre Nº PC",** 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 subiréis comprimida la carpeta al aula virtual.	- 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. - El examen puntúa sobre 10, selecciona los ejercicios como quieras.

Problema 1 – Procesado de XML (3 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. No es necesario, ni se recomienda utilizar CodeIgniter

Nos han proporcionado el fichero "subgrupos.xml" que contiene los datos resultados de la exportación de la tabla *subrupos* definida en el fichero "subgrupos_estructura.sql".

Formato Fichero XML descrito informalmente	Estructura tabla

`<DATA>`
`<ROWS>`
`<ROW>`
`<id>...</id>`
`<cod>..</cod>`
`<nombre>...</nombre>`
`</ROW>`
`...`
`</ROWS>`
`</DATA>`
	`CREATE TABLE `subgrupos` (`
	`  `id` int(11) NOT NULL,`
	`  `cod` smallint(5) unsigned NOT NULL,`
	`  `nombre` varchar(255) DEFAULT NULL,`
	`) ENGINE=InnoDB DEFAULT CHARSET=utf8;`

Nos solicitan que realicemos un script PHP que nos permita insertar los datos contenidos en el fichero "subgrupos.xml" en la tabla "subgrupos" ubicada en la base de datos "subgrupos".
Se desea además que el script permita seleccionar el fichero XML que contiene los datos en un formulario. De esta forma se podrá utilizar en futuras ocaciones.

Podéis utilizar el fichero "subgrupos_estructura.sql" para crear la tabla y base de datos.

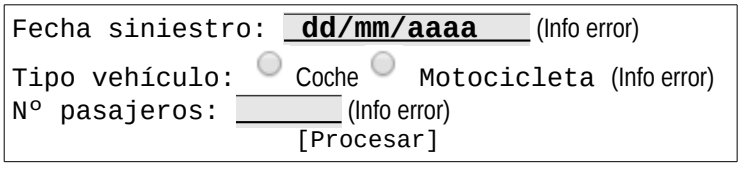
Puntuación:
- Inserción de datos en tabla desde fichero XML - 2 ptos
- Lectura de fichero de formulario – 1 pto

Problema 2 – Filtrado de datos en formulario – 2,5 Ptos

Utilizaremos los ficheros que se incluyen en la carpeta "ej1" para resolver el problema. Solo es necesario que copiéis el fichero del controlador

Nos han encargado la realización de un formulario de captura de datos para una compañía de seguros. Dicho formulario debe recoger una serie de datos y filtrarlos atendiendo a las siguientes reglas que a continuación indicamos. La solución que implementemos debe utilizar el Framework Codeigniter y la librería "form_validation"

- Fecha de siniestro: Campo de tipo texto que lee la fecha del siniestro. Es un campo obligatorio y precisa que se introduzca una fecha válida y anterior a la actual.
- Tipo de vehículo: Será un campo de tipo radio con las opciones "Coche" o "Motocicleta". Por defecto no aparecerá ninguno selecionado. Para que el campo se consideré correcto deberá tener seleccionada alguna opción.
- N.º de pasajeros: Este será un campo de tipo numérico que permitirá los siguientes valores dependiendo del tipo de vehículo:
  - Coche: Será valido un número entre 1 y 5
  - Motocicleta: Será valido un número entre 1 y 2



Cuando se envíe el formulario se validarán los datos y en el caso de que haya error se mostrará junto al campo el motivo del error. Permitiendo al usuario subsanar dicho error, se mostrará el valor introducido en los campos para facilitar la corrección. Cuando el envío haya sido correcto se mostrará el mensaje "Datos introducidos correctamente"

La solución implementada deberá filtrar los datos utilizando el estilo definido por el framework CodeIgniter.

Como nuestra solución irá dentro de una aplicación mayor, nos solicitan que nuestra solución utilice como plantilla el fichero "template_ej2.php". De momento solo rellenaremos información en la sección "tmpl_cuerpo", más adelante ya se nos proporcionará más información.

Puntuación :

- Filtrar error en fecha de siniestro. (0,5 ptos) / Mostrar error y valor de campo en error (0,25 ptos)
- Filtrar error en "Tipo vehículo" (0,25 ptos) / Mostrar selección en error (0,25 ptos)
- Mostrar error en N.º pasajeros (0,5 pto) Mostrar valor en error (0,25 ptos)
- Usar plantilla 0,25 ptos (para mostrar algo del programa)
- Programa completo 0,25 ptos.

Problema 3 – Cliente servicio web SOAP (1 ptos)

Utilizando el servicio web que se describe en el siguiente fichero WSDL (http://www.webservicex.net/geoipservice.asmx?WSDL), y del cual tenéis una copia en la carpeta "ej3" se desea que realicéis un script en php que usando el servición SOAP nos muestre información de geolocalización sobre una IP pública, por ejemplo 84.127.229.40 . Método (`GetGeoIP`)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://www.webservicex.net/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:GetGeoIP>
      <!--Optional:-->
      <web:IPAddress>?</web:IPAddress>
    </web:GetGeoIP>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetGeoIPResponse xmlns="http://www.webservicex.net/">
      <GetGeoIPResult>
        <ReturnCode>1</ReturnCode>
        <IP>84.127.229.40</IP>
        <ReturnCodeDetails>Success</ReturnCodeDetails>
        <CountryName>Spain</CountryName>
        <CountryCode>ESP</CountryCode>
      </GetGeoIPResult>
    </GetGeoIPResponse>
  </soap:Body>
</soap:Envelope>
```

Deberéis mostrar los a siguiente información: País (CountryName), y Código de País (CountryName)

Nota: No se garantiza que el servicio funcione correctamente durante el examen se recomienda probar antes son SOAP UI

Problema 4 – Cliente / Servidor Servicios Web (2 ptos)

Deberéis incluir los fichero del servidor en la carpeta "servidor" y los ficheros del cliente en la carpeta "cliente". Las rutas que utilicéis deberán ser relativas para que sea sencillo su ejecución en distintos entornos. Podéis declarar la ruta en una constante al inicio del fichero, igualmente.

Una empresa de alquiler de coches está diseñando un sistema distribuido que permitirá centralizar la gestión de su parque de coches. Nos solicitan que realicemos un servicio web, y que problemos dicho servicio realizando un cliente que lo pruebe.
La API a desarrollar en el servidor será la siguiente.

Servidor:
Implementará las siguientes funciones:

CochesEnCatalogo()	Devuelve un array con las marcas de coches disponibles en el servicio. Por ejemplo: Renault Clio, Opel Corsa, Ford K, Seat Ibiza
NumCochesDisponibles($marca)	Devuelve un valor entero indicando el número de coches disponibles de alguna de las marcas devueltas en el método anterior. En caso de no existir la marca devolverá el valor -1

Supongamos, para hacer pruebas, que el stock de las marcas anteriores es: Renault Clio=5, Opel Corsa=10, Ford K=2, Seat Ibiza 50, aunque los valores devueltos son irrelevantes, pues en un entorno real se leerían de la base de datos.

Cliente:
En un formulario se mostrará un campo select en el que se permitirá seleccionar alguno de los coches del catalogo. Al pulsar el botón "Mostrar stock" mostraremos el número de coches existentes.

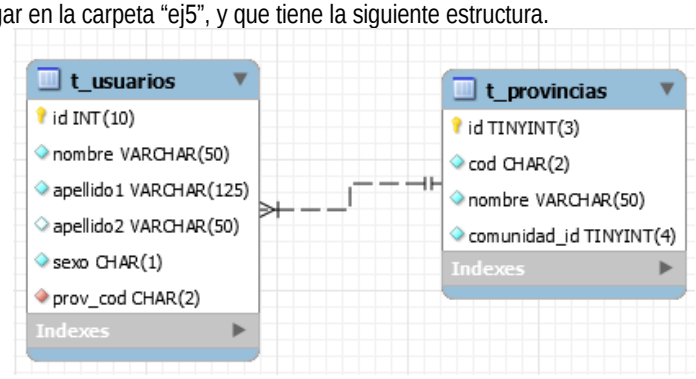
No os preocupéis por la presentación o el filtrado de datos. Lo importante es la definición y uso correcto del servicio.
Nota: Podréis utilizar la librería SOAP o la vista como ejemplo en clase, eso queda a vuestro criterio.
Puntuación :

- Plantear cliente y servidor, aunque no operativos 1 pto
- Problema completo 1pto

Problema 5 – Acceso a base de datos – 4 Ptos

Utilizaremos la base de datos "paises_bd", de la cual tenéis información en la carpeta que se entrega con el examen. Aquí podréis consultar la estructura de la base de datos y los campos de las tablas. Esta base de datos debéis instalarla en vuestro sistema ejecutando el script paises.*sql*

Utilizando el framework CodeIgniter, se desea realizar una aplicación que nos muestre los elementos existentes en la base de datos "usuarios", la cual podéis cargar en la carpeta "ej5", y que tiene la siguiente estructura.



La aplicación tendrá las siguientes características:

- Al arrancarla mostrará la lista de todos los usuarios.
- Se mostrará en cada página el número de usuarios total que tiene la lista **que se está mostrando**.
- Al pulsar el nombre de provincia de un usuario, que será un enlace, se mostrará la lista de usuarios de dicha provincia y se cambiará el campo de encabezado que dice porqué está filtrado.

El formato de página será algo similar al siguiente:

Listado usuarios
No se aplica filtro / Solo de provincia [Nombre]
Número usuarios: XX

Nombre	Apellido1	Apellido2	Provincia (Nombre)
XXXX	XXXX	XXXX	XXXX *(es enlace)*
...			...

<< 1 2 3 ... >> *(Paginador de vuestro gusto con 20 elementos por página)*

En el encabezado se indicará si no se está aplicando ningún filtro, o si se están filtrando los datos por provincia.

Puntuación:

- Mostrar lista usuarios 0,75 ptos.
- Paginar resultados 0,75 ptos.
- Mostrar texto: No se aplica filtro o Sólo de provincia XXXX 0,5 ptos.
- Mostrar lista usuarios filtrado por provincia 1 ptos.
- Incluir paginación filtrando por provincia 0,5 ptos.
- Programa completo 0,5 ptos.

Problema 6 – Generar un PDF. (4 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 puede utilizar CodeIgniter o no, como mejor veáis

Utilizando la base de datos de usuarios, proporcionada para el ejercicio anterior, el departamento de estadística desea generar un informe en PDF, en formato A4 vertical que mostrará la siguiente estructura:

Encabezado			Estadística de	
	Nº	Apellido1º	Total	%
Detalle	1	XXXXX	XX	XX%
	2	XXXXX	XX	XX%
	….			
Pie página	Página X de N			

La aplicación mostrará directamente el informe, mostrando un encabezado y pie para cada página. En el encabezado se mostrará el nombre de cada columna, y se repetirá en todas las páginas. Las columnas que tendremos serán las siguientes:

- N.º: Número correlativo que empieza por 1
- Apellido: Valor para cada apellido diferente, ordenado alfabéticamente
- Total: N.º de apellidos existentes de cada apellidos
- %: Porcentaje sobre el total que representan los apellidos

Notas:

- No os preocupéis mucho por la presentación. Lo que importa es que demostréis que sabríais como hacer las cosas.
- Las consultas de agrupamiento en SQL (group by) simplifican la solución del problema.

Puntuación :

- Generar informe sin porcentaje: 1,25 ptos.
- N.º correlativo: 0,25 ptos (solo si hay un informe válido)
- Incluir porcentaje: 0,5 ptos
- Encabezado 0,75 ptos.
- Pie y números de página (actual / Total) 0,75 ptos
- Programa completo: 0,5 pto.