

UT03 – Supuesto Práctico: Desarrollo de aplicaciones Web dinámicas

1.- Ejercicio. XML, AJAX y Sesiones. (4 ptos)

Se pide realizar una verificación de inicio de sesión en un sistema. Para ello utilizaremos una fuente de datos que consistirá en un archivo de configuración denominado “configuracion.xml” con la siguiente estructura y los siguientes datos:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<configuracion>
```

```
  <usuario>
```

```
    <alias>admin</alias>
```

```
    <clave>1234</clave>
```

```
    <perfil>Admin</perfil>
```

```
  </usuario>
```

.....

En el sistema “usuario” dispone de diferentes datos “Alias”, “Clave” y “Perfil”.

Existen diferentes usuarios con diferentes tipos de perfiles. Los perfiles disponibles son: “Admin”, “Cliente” y “Gestor”.

Se pide realizar en el fichero “index.html” un formulario “Login” con los campos “usuario” y “clave”. Mediante peticiones AJAX, se debe comprobar que el usuario tiene acceso a la aplicación.

Una vez verificado que el usuario tiene acceso a la aplicación, se debe iniciar sesión y guardar los datos en la sesión, los datos serán “usuario” y “perfil”.

Sin movernos de la misma página “index.html” se debe ocultar el formulario de “Login” y mostrar el resto de las secciones dependiendo del perfil del usuario. (Se facilita el código de las estructuras html).

Se dispone de un menú en el que aparecerá el alias de usuario y un link denominado “Cerrar Sesión” que debe realizar una petición AJAX que elimine la sesión y devuelva al usuario al Login inicial.

A continuación, mostramos algunas pantallas de ejemplo del resultado esperado:

Usuario Perfil Admin:

Usuario Clave

Usuario: admin [Cerrar sesión](#)

Contenido de usuario admin

Usuario Perfil Cliente:

Usuario Clave

Usuario: cliente [Cerrar sesión](#)

Contenido de usuario cliente

2.- Ejercicio.XML y AJAX (6 ptos)

Se pide realizar una aplicación que muestre información de una tienda online de discos. Para ello se debe mostrar una serie de tablas con información de los albums, las tiendas que venden los discos y la posibilidad de comprar las canciones sueltas de los Albums. La fuente de datos que se va a utilizar es un fichero xml denominado “albums.xml” con la siguiente estructura y los siguientes datos:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<albums>
```

```
  <album>
```

```
    <id></id>
```

```
    <titulo></titulo>
```

```
    <portada></portada>
```

```
    <tiendas>
```

```
      <tienda>
```

```
        <nombre></nombre>
```

```
        <precio></precio>
    </tienda>
</tiendas>
<artistas></artistas>
<plataformas></plataformas>
<canciones>
    <cancion>
        <id></id>
        <titulo></titulo>
        <duracion></duracion>
        <colaboraciones></colaboraciones>
        <letra></letra>
        <precio></precio>
    </cancion>
</canciones>
</albums>
```

Se pide, Crear tres clases denominadas “Album”, una clase denominada “Tienda” y otra clase denominada “Canción” con la estructura de atributos del fichero xml.

Se pide crear una serie tablas de álbumes mediante una petición AJAX de tipo GET asíncrona, para poder mostrar la relación de álbumes, tiendas donde se venden los álbumes y las canciones pertenecientes a esos álbumes.

Este conjunto de tablas se debe generar dinámicamente en el fichero “albums.html” al realizar un click sobre el link de la barra de menú “Ver Albums”, como se muestra en las siguientes capturas:


Menú:

[Ver Albums](#)

Menú:

[Ver Albums](#)

Listado de Albums y Canciones

Título	Artistas	Plataformas	Portada
And Justice for All	James Hetfield,Kirk Hammett,Lars Ulrich,Cliff Burton,Dave Mustaine,Robert Trujillo	Spotify	

Tienda	Precio
El corte Ingles	22,50€
Amazon	18,90€

Título	Colaboraciones	Letra	Duración	Precio
One	Hetfield, Ulrich	...	7:26	0,75€
...And Justice for All	Hetfield,Hetfield, Ulrich, Kirk Hammett	...	9:44	1€
To Live Is to Die	Cliff Burton	...	9:48	1,5€

Título	Artistas	Plataformas	Portada
--------	----------	-------------	---------

La estructura de la aplicación será la siguiente (la estructura es obligatoria y no se puede modificar, se pueden incluir ficheros nuevos dentro de cada carpeta dónde correspondan):

/bootstrap-> incluyen los ficheros del Bootstrap.

/js -> incluirán los ficheros AJAX

/clases -> incluirán los ficheros de las clases “álbum”, “tienda” y “canción”.

/datos -> incluirán el fichero “bd” y “albums.xml”

/portadas-> dónde se incluyen los ficheros de las portadas de los albums.

albums.html

Indicaciones:

- La funcionalidad de cada aplicación debe ser completa para obtener la puntuación establecida.
- No se podrán modificar la estructura de los ficheros proporcionados.
- Los elementos que se incluyen en los ficheros deben permanecer inalterados.
- No se pueden modificar parámetros, ni nombre de funciones, ni elementos HTML, etc.
- Se puede utilizar AJAX dado en clase. (No se pueden utilizar Frameworks)
- Se pueden añadir nuevos ficheros.

- El uso inadecuado de elementos del código que no estén correctamente establecidos puede dar lugar a penalizaciones en la nota. (Nombre de variables que no corresponden conceptualmente con los datos, elementos agregados como HTML que no corresponden con la lógica del programa, etc.).
- Se pueden añadir funciones auxiliares dentro de las clases o ficheros.
- Pueden utilizar internet, documentos, manuales, etc, durante el desarrollo de la prueba.
- Si se detecta alguna copia. El alumno y/o alumnos implicados en dicha copia, tendrán un 0 en la nota del supuesto, lo que implica la suspensión de la unidad.
- El trabajo y entrega del supuesto es una tarea individual.

Indicaciones de la entrega:

La tarea se debe entregar en CAMPUS en tiempo y forma.

Entregar los códigos en carpetas separadas que se han proporcionado previamente:

"/1-login-perfiles", "/2-cargar-albums" comprimidas en un único archivo .zip

El nombre de la entrega debe ser:

apellidos_nombre_dsw_ut03_supuesto.zip