**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**DESARROLLO DE APLICACIONES WEB**

**TERCERA EVALUACIÓN - I TÉRMINO 2016**

***Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matrícula: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

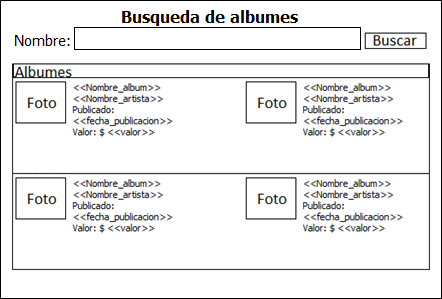
**TEMA 1 (30 puntos)**

1. Explique cómo funciona el modelo cliente-servidor?
2. Qué son los *media queries*?
3. Mencione al menos 3 razones del porqué desarrollar basado en estándares.
4. Explique el proceso de colocar a un usuario en sesión, y donde se almacena la sesión?
5. Cual es la diferencia entre asignar al *Cache-Control* el valor de public o private?
6. Para qué sirve el Content-Type cuando se responde un requerimiento?

**TEMA 2 (70 puntos)**

La empresa *Charmender S.A.* se dedica a la venta de albumes mp3 y necesitan el desarrollo de una aplicación web que permita a sus clientes adquirir los productos a través del portal. Han decidido que la aplicación sea desarrollado bajo los estándares Web, el backend sea con node.js y el framework express, y utilizarán una base de datos no relacional (mongo). A continuación se le solicita lo siguiente:

1. **(12 puntos)** Cree una estructura html con tags semánticos para que un cliente pueda buscar albumes de musica a traves de un formulario sencillo. A continuación, se muestra un mockup del sitio.



1. **(13 puntos)** Se requieren aplicar las siguientes reglas de CSS al siguiente código para que luzca como la figura adjunta. Se necesita:
   1. El ancho de todo el albumPopup sea de 500px.
   2. Un padding de 5px en la foto del álbum
   3. Los textos de la tabla estén centrados.
   4. Remover el subrayado en links, además de aplicarles el color brown y negritas.
   5. Y cualquier otro cambio visual que usted identifique.



|  |
| --- |
| <div id="albumPopup">  <div class="fotoAlbum"><img src="Californication.jpg" height="100px"></div>  <div class="info">  <h2>Red Hot Chili Peppers</h2>  <h4>Californication</h4>  <p>Fecha de publicación: 8 de junio de 1999</p>  </div>  <div class="canciones">  <table>  <thead>  <tr>  <th>#</th>  <th>Canción</th>  <th>Duración</th>  </tr>  </thead>  <tbody>  <tr>  <td>1</td>  <td><a href="#">Around the World</a></td>  <td>3:58</td>  </tr>  <tr>  <td>2</td>  <td><a href="#">Parallel Universe</a></td>  <td>4:30</td>  </tr>  </tbody>  </table>  </div>  </div> |

1. **(15 puntos)** Se necesita implementar la función **cargarDatos** que recibe un idalbum y deberá hacer un requerimiento AJAX a <http://localhost/album/:idalbum/canciones> para obtener todas las canciones a partir del idalbum. La función deberá crear una lista y usted deberá hacerle un append a un div llamado “albumCanciones”.

La respuesta del servidor es un JSON con la siguiente estructura:

{

idAlbum: 5,

album: Californication,

canciones: [

"Around the world",

"Parallel Universe",

....

]

}

1. **(15 puntos)** Se le solicita crear los modelos (utilizando Mongoose) para representar las siguientes entidades:
   1. Un artista tiene un nombre, un país de origen, y álbumes publicados.
   2. Un álbum está compuesto de un nombre, una fecha de lanzamiento, y una lista de canciones.
   3. Una canción está compuesta de un nombre y una duración
2. **(15 puntos)** Implemente los siguientes endpoints (o rutas) en node.js utilizando express:
   1. Un endpoint que permita crear una canción
   2. Un endpoint que permita eliminar una canción
   3. Un endpoint que permita consultar todas las canciones a partir de un álbum.

var express = require('express');  
var router = express.Router();

...

...

module.exports = router;