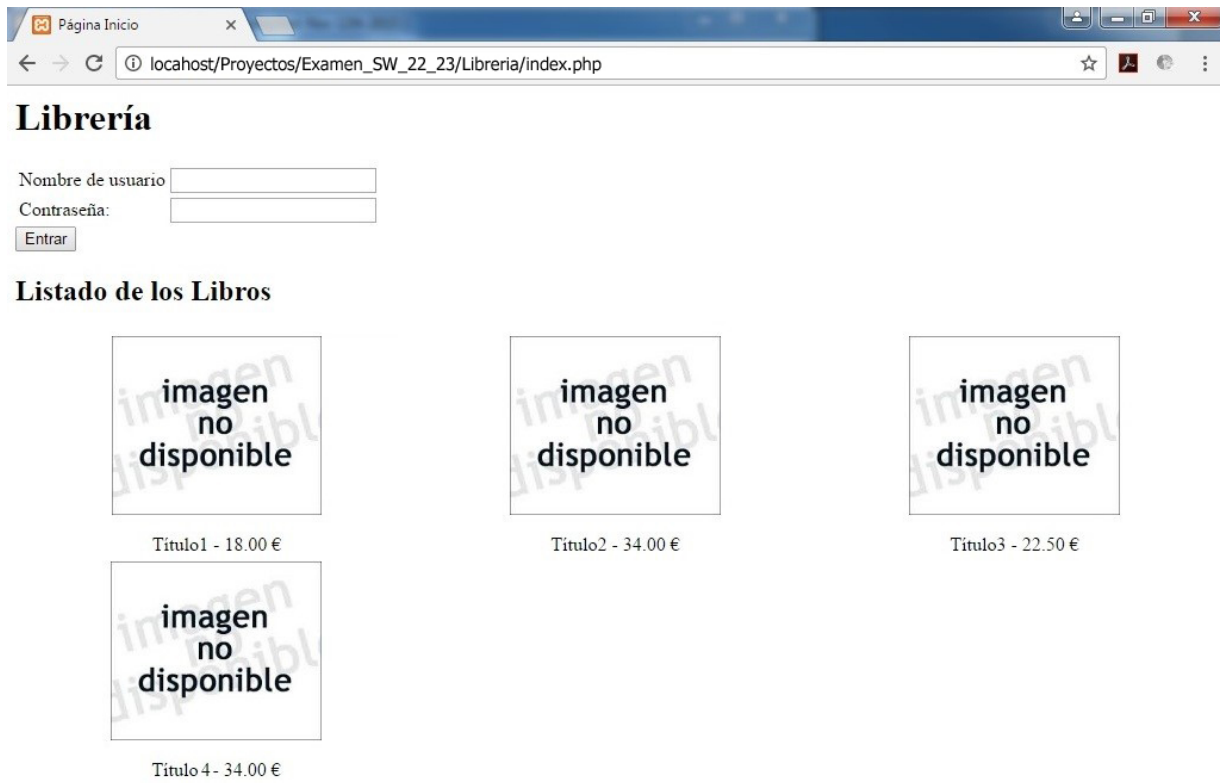


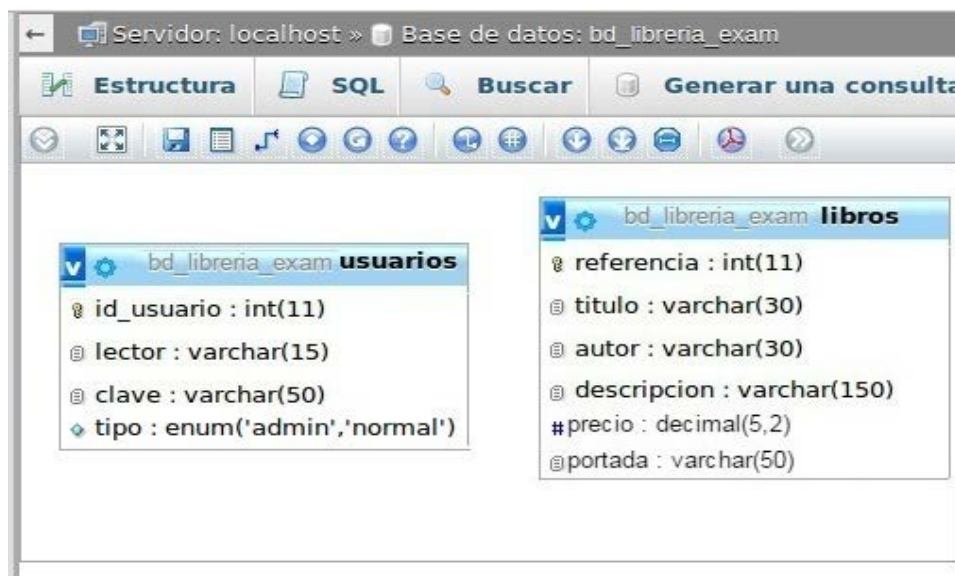
EXAMEN 2º DAW. Servicios Web

Realizar una aplicación web para llevar el control online de una librería teniendo en cuenta los siguientes aspectos:

1) En la página inicial del lugar (index.php), el lector va a poder sólo loguearse y ver los libros en filas de tres en tres:



2) La base de datos, así como las tablas y los atributos empleados deben ser EXACTAMENTE:

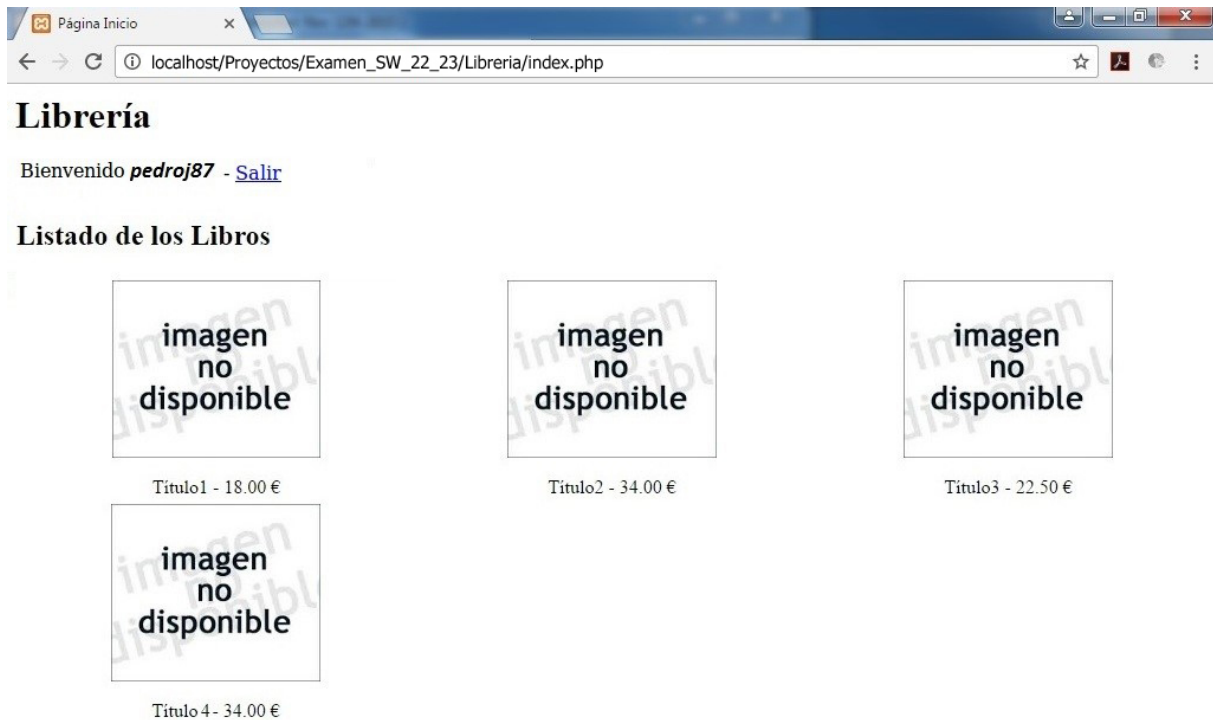


NOTA: La clave de la tabla **usuarios** es de tipo Auto-Incremento y la de **libros** no. Además, la columna “tipo” de la tabla usuarios y la columna “portada” de la tabla libros, tendrán los valores predeterminados a “normal” y “no_foto.jpg” respectivamente.

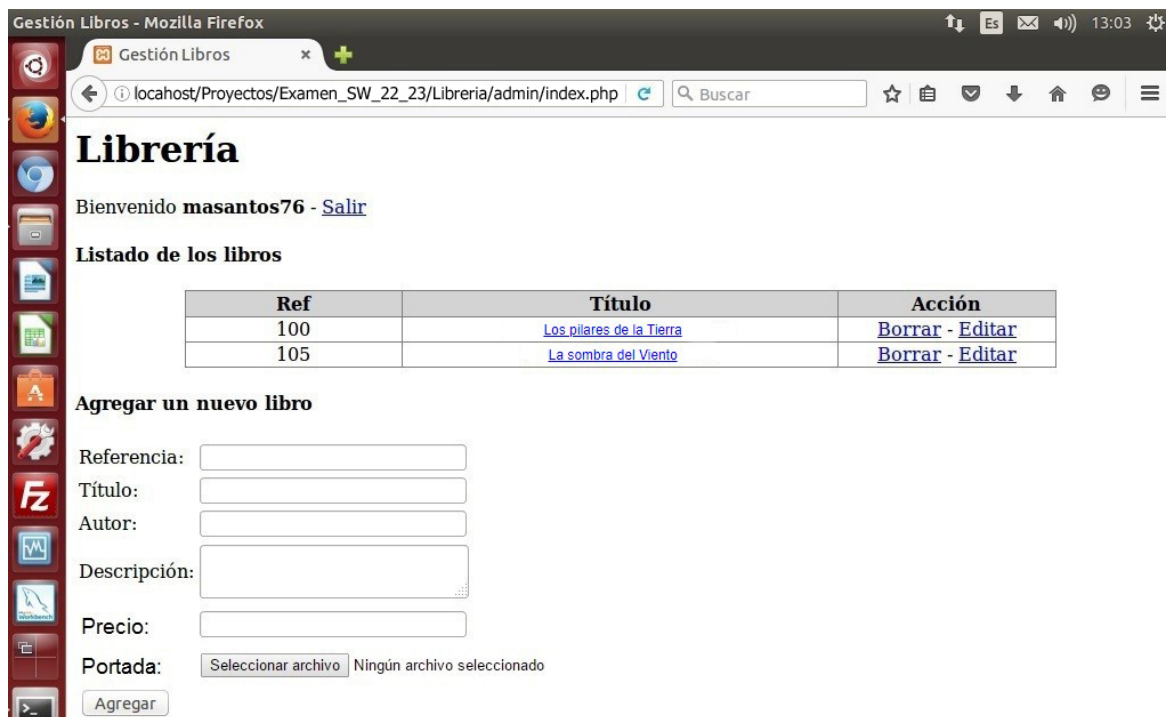
3) Introduzca al menos un usuario normal, otro admin, y cuatro libros.

EXAMEN 2º DAW. Servicios Web

4) Si un lector “normal” se loguea correctamente, en la página index.php se le dará la bienvenida y la opción de salirse, con un periodo de inactividad de 2 minutos.



5) Y cuándo un lector “admin” se loguee correctamente va a saltar a una nueva página (**admin/gest_libros.php**), donde va a poder gestionar los libros de la librería, siempre que no transcurra un periodo de inactividad superior a los 2 minutos, en cuyo caso, el usuario deberá volver a loguearse. Las acciones de los Borrar y Editar van a quedar limitadas a un simple mensaje de acción (“ El libro con referencia X ha sido editado/borrado con éxito”).



6) Los errores a controlar (con código php), al agregar un libro van a ser:

- Que no se quede ningún campo vacío (a excepción de la portada).
- Que no repita la Referencia y tanto ella como el Precio sean un número.
- Si se sube una portada (**img_referencia.extensión**), que el tamaño del archivo imagen no sea mayor de 500 kB.

EXAMEN 2º DAW. Servicios Web

7) El acceso a la BD por parte de nuestra aplicación Web sólo podrá hacerse mediante servicios Web. Para ello crea una API REST protegida con EXACTAMENTE los siguientes servicios:

- a) Solicitar autenticación de un usuario mediante una petición POST en la que aportaremos los datos necesarios para la autenticación (usuario y clave encriptada en MD5) mediante un array asociativo con los índices: "lector" y "clave". En caso de error por la BD el JSON devuelto será: { "error" : "Error...."}, en caso de que el usuario no se encuentre registrado el JSON será: { "mensaje" : "Usuario no se encuentra regis. en la BD"} y en el caso de que se encuentre registrado se devolverá los datos del usuario y la clave de la sesión de la API, el JSON será : { "usuario" : [...], "api_session": key_session}

URL de la petición: http://localhost/Proyectos/Examen_SW_22_23/servicios_rest/login

- b) Obtener los datos de un usuario logueado en la API, mediante una petición GET en la que aportaremos la clave de sesión mediante un array asociativo con índice "api_session". En caso de error por la BD, el JSON devuelto será: { "error" : "Error...."}, en caso de que el usuario no se encuentre registrado el JSON será: { "mensaje" : "Usuario no se encuentra regis. en la BD"} y en el caso de que se encuentre registrado se devolverá los datos del usuario logueado, el JSON será : { "usuario" : [...] }

URL de la petición: http://localhost/Proyectos/Examen_SW_22_23/servicios_rest/logueado

- c) Desloguearse de la API mediante una petición POST en la que aportaremos la clave de sesión mediante un array asociativo con índice "api_session" y nos devolverá un JSON: { "log_out": "Cerrada sesión en la API"}

URL de la petición: http://localhost/Proyectos/Examen_SW_22_23/servicios_rest/salir

- d) Mediante una petición GET, obtener todos los libros de la BD. En caso de error por la BD el JSON devuelto será: { "error" : "Error...."}, en otro caso el JSON será: { "libros" : [[...], [...],..., [...]] }.

URL de la petición: http://localhost/Proyectos/Examen_SW_22_23/servicios_rest/obtenerLibros

- e) Crear un nuevo libro mediante una petición POST en la que aportaremos los datos del libro y de la clave de sesión de la API mediante un array asociativo con los siguientes índices: "referencia", "titulo", "autor", "descripción", "precio", "email" y "api_session"). En caso de error por la BD el JSON devuelto será: { "error" : "Error...."}, en otro caso el JSON será : { "mensaje" : "Libro insertado correctamente en la BD"}

URL de la petición: http://localhost/Proyectos/Examen_SW_22_23/servicios_rest/crearLibro

- f) Actualizar la portada de un libro mediante una petición PUT en la que aportaremos mediante un array asociativo el nombre de la nueva portada y la clave de la API con los siguientes índices: "portada y "api_session"). En caso de error por la BD el JSON devuelto será: { "error" : "Error...."}, en otro caso el JSON será: { "mensaje" : "Portada cambiada correctamente en la BD"}

URL de la petición:

http://localhost/Proyectos/Examen_SW_22_23/servicios_rest/actualizarPortada/{referencia}, donde **referencia** es un atributo pasado por la URL.

- g) Mediante una petición GET, comprobar si en una tabla de la BD de datos, para una columna determinada ya existe un valor determinado. Para ello aportaremos la clave de sesión mediante un array asociativo con índice "api_session", En caso de error por la BD el JSON devuelto será: { "error" : "Error...."}, en otro caso el JSON será: { "repetido" :true ó false}.

URL de la petición:

http://localhost/Proyectos/Examen_SW_22_23/servicios_rest/repetido/{tabla}/{columna}/{valor}, donde **tabla**, **columna** y **valor** son atributos pasados por la URL.

NOTA: Si se llama un servicio protegido sin la clave de sesión oportuna, éste devolverá el siguiente JSON: { "no_auth": "No tienes permisos para usar este servicio"}

EXAMEN 2º DAW. Servicios Web

PUNTUACIÓN APROXIMADA

NOMBRE:

(0'5 puntos) Página inicial (libros de tres en tres)

(1'25 puntos) **Login.**

Mantiene valores: 0'1

Errores: 0'1 + 0'3 (Vacíos + Usuario no registrado)

Login : 0'75

(1'5 puntos) **Seguridad, vistas y saltos entre páginas según Enunciado**

Control baño y mensaje: 0'5 (0'25 + 0'25)

Control tiempo y mensaje: 0'5 (0'25 + 0'25)

¿Es seguro?: 0'5

(0'5 puntos) Bienvenida y Salir Sesión (0'25 en cada vista)

(1'5 puntos) Lista bien tabla y formulario agregar (1+ 0'5)

(3'5 puntos) Agregar

(1'6 puntos) Errores

(0'1 puntos) Vacío

(0'2 puntos) No números

(0'7 puntos) Repetido

(0'6 puntos) De la imagen

(0'15 puntos) Mantiene valores

(0'75 puntos) Inserta libro

(1 punto) Sube imagen

(0'75 puntos) Detalles

(0'5 puntos) Muestra info

(0'25 puntos) Muestra foto

(0'25 puntos) Borrar

(0'25 puntos) Editar