



# UNIVERSIDAD DE GRANADA

## Programación Web

### Práctica evaluable II

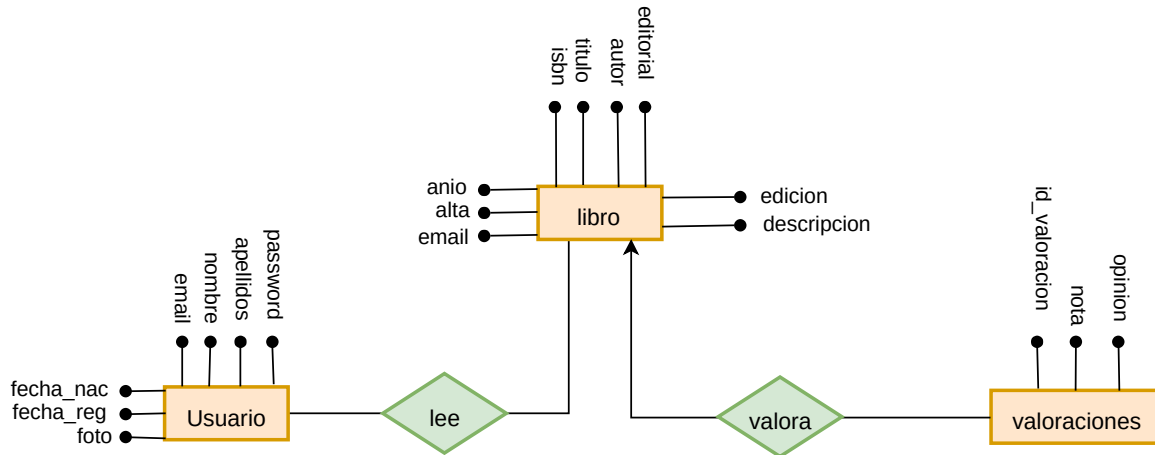
## Adaptación de la web del recomendador de libros

David Carrasco Chicharro

# MySQL

## Diseño conceptual

El diagrama E/R para el diseño de la base de datos del sistema es el siguiente:



## Diseño físico

```
CREATE TABLE usuario(
    email VARCHAR(60) NOT NULL PRIMARY KEY,
    nombre VARCHAR(25) NOT NULL,
    apellidos VARCHAR(60) NOT NULL,
    password CHAR(64) NOT NULL,
    fecha_nac DATE NOT NULL,
    fecha_reg DATE NOT NULL DEFAULT CURDATE(),
    foto VARCHAR(70)
);

CREATE TABLE libro(
    isbn CHAR(13) NOT NULL PRIMARY KEY,
    titulo VARCHAR(80) NOT NULL,
    autor VARCHAR(80) NOT NULL,
    editorial VARCHAR(30) NOT NULL,
    anio INTEGER(4) NOT NULL,
    edicion INTEGER(3) NOT NULL,
    descripcion VARCHAR(800) NOT NULL,
    alta DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP(),
    email VARCHAR(60) REFERENCES usuario(email),
);

CREATE TABLE valoraciones(
    id_valoracion INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nota INTEGER(1) NOT NULL,
    opinion VARCHAR(500) NOT NULL,
    email VARCHAR(60) REFERENCES usuario(email),
    CHECK (nota >= 1 AND nota <= 5)
);
```

```
CREATE TABLE lee(  
    email VARCHAR(60) REFERENCES usuario(email),  
    isbn CHAR(13) REFERENCES libro(isbn),  
    PRIMARY KEY(email,isbn)  
);  
  
CREATE TABLE valora(  
    isbn CHAR(13) REFERENCES libro(isbn),  
    id_valoracion INT REFERENCES valoraciones(id_valoracion),  
    PRIMARY KEY (id_valoracion)  
);
```

## Ficheros de configuración y clases

En el directorio bookrecsysII/classes se encuentran el archivo de configuración y las clases necesarias para realizar todas las operaciones que implique interactuar con la base de datos desde PHP. De esta manera se simplifica mucho el acceso, inserción, modificación y borrado en la base de datos.

`configuracion.inc` → archivo de configuración con la declaración de las constantes necesarias para conectar a la base de datos con la que vamos a trabajar y acceder a las tablas correspondientes.

`datosObject.class.inc` → clase abstracta que ofrece las operaciones más habituales para gestión de la base de datos.

El resto de ficheros (`*.class.inc`) corresponden con las operaciones de base de datos de cada una de las distintas tablas.

# PHP

## header\_nav.php

En el directorio bookrecsysII/template se encuentra el archivo `header_nav.php` que incluye el código común de cabecera de todos los ficheros con código PHP, donde aparece la foto del usuario, su nombre y un enlace para desconectarse (**punto 3**). Además, realiza una comprobación acerca de si existe una sesión activa; en caso de no existir impide ver páginas sin haber iniciado sesión, redirigiendo al `index.html`.

```
session_start();
if ( !isset($_SESSION["id"]) ){
    header("Location: index.html");
    exit;
}
```

## altausuario.php

Comprueba si el email proporcionado en el formulario existe ya en el sistema. En caso de existir, muestra de nuevo la misma página añadiendo un botón para iniciar sesión (redirigiendo al `index.html`); por el contrario, si no existe el email, añade el usuario a la base de datos (**punto 1**).

Para subir la foto del usuario al sistema se comprueba si el servidor permite la subida, de modo que si no es posible se utiliza una imagen por defecto:

```
if (move_uploaded_file($_FILES['foto']['tmp_name'], $formulario->getFotoWithDir()))
    $directorioFoto = "/imagenes/".$formulario->getFoto();
else
    $directorioFoto = "icono-usuario.png";
```

Una vez insertado el usuario en la base de datos, se establece la sesión:

```
$_SESSION["id"] = $formulario->getEmail();
$_SESSION["nombre"] = $formulario->getNombre();
```

Por último se redirige a `index.php`.

## login.php

Permite el inicio de sesión del usuario (**punto 2**). También presenta un formulario para iniciar sesión si el usuario introduce un email y/o contraseña inválidos y establece la sesión. Si el email y/o contraseña se siguen introduciendo de manera incorrecta, se redirige de nuevo a la misma página.

## logout.php

Termina la sesión cuando el usuario clicca en el link de “Desconectar” (**punto 2**) situado en la cabecera.

```
session_start();
$_SESSION = array();
session_destroy();
header("Location: index.html");
exit;
```

## index.php

Muestra los libros mejor valorados están ordenados por nota.

Para crear el enlace a cada libro, mediante la función `seoUrl($string)` que aparece en el código, se transforma el título del libro a una cadena con minúsculas con cada palabra separada por guiones. Se añade dicha cadena a continuación de `libro.php`, de modo que queda del siguiente modo: `libro.php?b=titulo-del-libro`

## mislibros.php

Muestra a la izquierda los libros leídos por el usuario. El link a dichos libros se forma de la siguiente manera: `librolibroleido.php?rb=ISBN`, siendo ISBN el número correspondiente al ISBN del libro. Si no hay libros leídos, se especifica dicho hecho.

En la parte derecha se muestran los libros dados de alta en el sistema, ordenados desde el más reciente en primer lugar. Para ello, en la consulta SQL se especifica que las tablas se ordenen por el atributo `alta` de forma descendiente (DESC).

## altalibro.php

Para dar de alta un nuevo libro (**punto 4**) se comprueba con `if(isset($_POST['enviar']))` que haya una recepción de atributos para el formulario. El siguiente paso es comprobar que el libro (ISBN) no existe ya en el sistema. Una vez realizado esto, se inserta en la base de datos el libro, se relaciona al usuario con el libro por haberlo leído, se inserta la valoración y se relaciona con la tabla de valoraciones.

## libro.php

De la URL se selecciona el título del libro con `$_GET['b']` y se busca en la base de datos toda la información relativa a éste.

Si el link a la imagen del libro no existe por algún motivo se realiza la comprobación de que no existe y se muestra una imagen por defecto.

```
$dir = "/home/x15520228/public_html/bookrecsysII/imagenes/";  
$file = 'libro-'. $libroBD['isbn'].'.jpeg';  
if (!is_file($dir.$file))  
    $file = 'icono.png';
```

Para mostrar la nota media de cada libro se realiza el cálculo de la media accediendo a cada valoración del libro en cuestión y multiplicando el resultado por 2, puesto que los usuarios introducen una nota del 1 al 5 y en la web resulta más interesante mostrar un número que llegue hasta el 10 para que resulten más atractivos aquellos libros con una nota alta.

Por último, el enlace a la valoración del libro será distinto para cada usuario, según si ha leído el libro o no. Si lo ha leído, el link llevará a `libroleido.php?rb=ISBN` y si no, a `valoracion_libro.php?vb=ISBN`.

## libroleido.php

Muestra los datos del libro, en el textarea de la opinión aparece la opinión que ya se había dado del mismo (**punto 5**) y en los radiobuttons se encuentra seleccionado el correspondiente a la nota con que se valoró (marcando como checked aquel cuyo valor sea igual a la nota).

## valoracion\_libro.php

Permite valorar un libro dado de alta en el sistema que no ha sido leído (**punto 5**). Es similar al apartado anterior, pero sin ninguna funcionalidad extra, puesto que no requiere captar datos de la base de datos, más allá de los datos del propio libro a valorar.

## opinion\_libro.php

En este documento se recogen los datos enviados por los formularios de libroleido.php y valoracion\_libro.php.

En primera instancia se comprueba que la sesión esté activa y se diferencia qué formulario ha enviado la información: `if( isset($_POST["modificar"]) || isset($_POST["valorar"]) )`

Si se está modificando la opinión simplemente se inserta en la tabla de valoraciones. En caso de estar añadiendo una nueva se indica que el usuario ha leído dicho libro, se incrementa el identificador de las valoraciones (pues es un id autogenerado) y se inserta la información correspondiente en las tablas de valoraciones y valora.

Se redirecciona siempre a libroleido.php porque al modificar la valoración tiene que volver a la página de procedencia, y en caso de valorar un libro pasaría a ser un libro leído.

## datospersonales.php

Obtiene los datos del usuario y los muestra en el formulario, permitiendo cambiar dichos datos y/o la contraseña (**punto 6**), cuyo procesamiento se describe en el siguiente apartado.

## cambiar\_datos.php

En este documento se recogen los datos enviados por los formularios de datospersonales.php.

En primera instancia se comprueba que la sesión esté activa y se diferencia qué formulario ha enviado la información:

```
if( isset($_POST["modificarDatos"]) || isset($_POST["cambiarPass"]) )
```

Cuando se cambia el email (que es la clave primaria de cada usuario), dicha modificación se propaga a todas las tablas de la base de datos donde aparece el usuario con su correo electrónico para mantener la integridad del sistema. El resto de datos tan sólo supone una actualización de la tabla del usuario en la base de datos.

Para cambiar la contraseña se comprueba que la “Contraseña actual” sea igual a la almacenada en el sistema. Además la nueva contraseña debe ser distinta a la anterior.

# JavaScript

Además de las diferentes porciones de código que se encuentran en el interior del código de los distintos documentos en PHP, caben destacar los ficheros `encrypt.js` y `validacion.js` que se encuentran dentro del directorio `bookrecsysII/js`. El primero de ellos se utiliza para encriptar todas las contraseñas de la web antes de ser enviadas al servidor con el algoritmo SHA-256; el segundo contiene las funciones necesarias para validar todos los formularios (**punto 7**) que hay en la página web.

Todos los formularios tienen el botón de envío desactivado hasta que todas las restricciones impuestas para la validación se cumplan.

Los errores relativos a los formularios se crean utilizando el DOM de la siguiente forma:

```
let div = document.getElementById(el);
let p = document.createElement('p');
let node = document.createTextNode(msg);
p.appendChild(node);
p.style.color = "red";
p.style.margin = "0 0 0 3%";
p.setAttribute("id", "alert-"+el+"");
div.parentNode.insertBefore( p, div.nextSibling );
```

## **index.html y login.php**

Realizan la validación del email y la contraseña. Para el email se comprueba que el formato de éste sea válido siguiendo una expresión regular. La contraseña debe tener entre 8 y 20 caracteres y, al menos, una mayúscula, una minúscula, un número y un carácter especial.

## **altausuario.html y altausuario.php**

Realiza la validación de todos los campos en longitud, no vacíos y formato. El email y la contraseña se realizan del mismo modo que se ha detallado en el apartado anterior. La contraseña, además, tiene que repetirse, de modo que si no coinciden los campos se mostrarán con el borde en rojo y, en verde, cuando sean iguales. Los campos de nombre y apellidos especifican que no puedan estar vacíos y cumplan un mínimo y máximo de caracteres en base a la longitud permitida en la base de datos y, por último, la fecha de nacimiento se valida con una expresión regular que sólo permite un determinado rango fecha.

Si el usuario intenta darse de alta con un email que ya existe en el sistema se muestra una alerta indicando dicho suceso.

## **altalibro.php**

Se realizan las comprobaciones típicas de validación, introduciendo como novedad a la anteriores el ISBN, que debe ser una cadena numérica de 13 dígitos, la editorial, que debe tener un valor distinto a 'Editorial' y la nota, que ha de tener un valor del `radiobutton` seleccionado. La descripción y la opinión muestran un contador de los caracteres que tienen escritos en sus respectivos cuadros de texto gracias a la función:

```
function contador( idTextArea ){
    let written = document.getElementById(idTextArea).value.length;
    document.getElementById('current-'+idTextArea).innerHTML = written;
}
```

## **libroleido.php**

Para activar el botón de envío del formulario debe haber algún cambio en la opinión y en la nota.

## **valoracion\_libro.php**

El botón de envío del formulario se activa cuando se ha escrito algo en el cuadro de opinión y además se ha puesto alguna nota.

## **datospersonales.php**

Permite enviar el formulario de datos del usuario si se produce algún cambio y valida los campos del mismo modo que se hace en `altausuario.php`. Para la validación de contraseñas el botón para enviar el formulario está deshabilitado y se habilita cuando el campo “Contraseña actual” está relleno, sendos campos de nueva contraseña se encuentran rellenos y con el mismo valor de contraseña y, además, sus valores no coinciden con el primer campo (“Contraseña actual”).

Cuando se pasa el ratón por encima de la foto del usuario se muestran los libros que éste ha subido, en caso de existir (**punto 8**).

```
let libros = <?php echo ( ' " '.$userLibros.' " '); ?>
    libros = libros.replace(/,/g, "\n");
    alert("Libros subidos: \n"+libros);
}
```

donde `$userLibros` es una variable en PHP donde se almacenan los libros del usuario o el mensaje "No ha subido ningún libro.", tal y como se ha indicado.

## **cambiar\_datos.php**

Cuando los datos o la contraseña se han cambiado correctamente o cuando no coincide la actual contraseña con la que hay almacenada en la base de datos, se muestra una ventana emergente indicándolo y se redirige a `datospersonales.php` mediante la función de JavaScript `window.location.replace("\datospersonales.php")`;