

Relación 9: BD.

Continuamos en el proyecto creado en la práctica 8.

Se deben tener en cuenta todas las consideraciones establecidas en prácticas anteriores. Además, hay que verificar que las operaciones con la base de datos son correctas.

1.- Para dar soporte permanente a los datos usaremos una BD. De la siguiente forma:

- Crear en MySql la BD practica9 (usar como codificación utf8mb3_spanish2).
 - Para mejorar la seguridad crearemos un nuevo usuario (usu9 en localhost) con acceso a la base de datos practica9 (permiso para escribir y leer). Será el usuario con el que accederemos desde php. (no se accede con root ya que tiene todos los permisos sobre todo).
 - Crear en la aplicación el fichero /aplicación/config/acceso_bd.php que definirá las variables necesarias para establecer una conexión a BD (servidor, usuario, contraseña, y base de datos). Usaremos estas variables cuando nos conectemos a la BD.

2.- Crear la tabla usuarios en la BD, una tabla propia independiente del sistema ACL donde vamos a indicar campos propios: *cod_usuario* (entero, autonumérico, clave principal), *Nick* (*cadena 50*) - mismo Nick que tenemos en ACL- no se puede repetir y una vez establecido no se puede cambiar, *nombre* (*cadena 50*, defecto '') - igual que el de ACL-, *nif* (*cadena 10* defecto ''), *dirección* (*cadena 50*), *población* (*cadena 30* defecto ''), *provincia* (*cadena 30* defecto ''), *CP* (*cadena 5* defecto '00000'), *fecha_nacimiento* (*date*), *borrado* (*boolean* defecto false), *foto* (*cadena 50* defecto '') nombre de archivo de foto; dirección relativa a carpeta /imágenes/fotos. A esta tabla se accederá igual que a otras tablas que tengamos en nuestra BD como compras, productos, etc.

3.- Crear el CRUD (apartado 4.3 del tema 3) correspondiente a los usuarios en la carpeta /aplicación/usuarios. Se tendrá:

- **página index.php.** Muestra una tabla con todos los usuarios que tenemos (el *cod_usuario* no se muestra). Tendremos un botón/enlace para añadir un nuevo usuario y enlaces para ver-modificar-borrar en la fila de cada usuario mostrado. Se debe poder filtrar por *Nick*, *provincia* y *borrado* y establecer un criterio de ordenación. A esta ventana se podrá acceder si hay un usuario validado que tenga el permiso 2 (permiso acceso a tablas administración). Para la operación ver, solo necesitas el permiso 2. Para las operaciones nuevo, modificar o borrar necesitas además el permiso 3 (permiso edición tablas administración).

- **página nuevoUsuario.php.** Muestra un formulario para poder recoger los datos de un nuevo usuario (el *cod_usuario* no se pide, ya que se genera en la BD). Se validan los datos y si todo es correcto, se añade el usuario a la BD y se va a la ventana verUsuario.php para mostrar los datos del usuario insertado. Si hay algún error se vuelve a mostrar el formulario con los últimos datos para modificarlos. A esta opción solo podrán acceder usuarios validados con permisos 2 y 3. Se debe tener enlace a index.php por si no deseamos continuar.

La foto se puede o no indicar. Si no se indica, pondremos una foto que tenemos por defecto. Si indicamos foto, ésta se subirá a la carpeta /imágenes/fotos y se guardará en el campo foto de la tabla usuarios el nombre con el que lo guardamos en esta carpeta. Podemos usar como nombre, uno generado de forma aleatoria con una longitud de 20 caracteres.

- **página verUsuario.php.** Dado un cod_usuario (que se pasa como parámetro y debe comprobarse que existe) se muestran los datos del usuario correspondiente (el cod_usuario no se muestra). A esta opción sólo podrán acceder usuarios validados con permiso 2. Se deben tener enlaces a modificar y borrar (solo si tiene el permiso3) y a index.php para cancelar. Lo más fácil es mostrar un formulario semejante al de nuevo usuario con los inputs de solo lectura.
- **página modificarUsuario.php.** Dado un cod_usuario (que se pasa como parámetro y debe comprobarse que existe) se muestra un formulario con los datos actuales del usuario para que pueda modificarlos (el Nick no se puede cambiar). Igual que en nuevo, se deben validar los valores introducidos. En caso de fallo al validar algún campo, se vuelve a mostrar el formulario con todos los datos que haya enviado para que los indique correctamente. A esta opción sólo podrán acceder usuarios validados con permiso 2 y permiso 3. Se debe tener enlaces a index.php y verUsuario para cancelar. Una vez modificado el usuario se va a la página de verUsuario
- **página borrarUsuario.php.** Dado un Nick (que se pasa como parámetro y debe comprobarse que existe) se muestra los datos del usuario y se pide confirmación del borrado. Si se confirma se borra el usuario de forma lógica (campo borrado a true), nunca física. A esta opción sólo podrán acceder usuarios validados con permiso 2 y permiso 3. Se debe tener enlaces a index.php y verUsuario para cancelar.

4.- La ACL es el mecanismo por el que identificamos y validamos usuarios para nuestra aplicación como ya vimos en la práctica anterior. Este mecanismo lo podemos implementar nosotros, o podemos usar las ACL de Facebook o Google, ...

En este caso lo implementamos nosotros. Tenemos la ACL usando base de datos. Esto nos permite tener los acl_usuarios y acl_roles de forma permanente. Para trabajar con esta clase:

- Crear las tablas acl_usuarios y acl_roles con los campos necesarios para contener la información de los usuarios y roles definidos en una ACL. Para ello tenemos el script creación_acl.sql
- Incorporar el objeto ACLBD a la carpeta de clases. (os paso una copia incompleta para que implementéis algunos métodos)
- Crear los roles iniciales (llamar al método apropiado de la clase ACLBD)
- Usar ACLBD como el mecanismo para el control de las listas de acceso de mi aplicación.

NOTA: No se acceden a las tablas de la ACLBD usando mysqli. Sólo se usarán los métodos de ACLBD

Se pide mantener coherencia entre la tabla usuarios y la ACL de forma que estén “enlazadas”.:

- El Nick será el mismo en usuario que en acl_usuarios (tras insertar usuario en el CRUD, llamar al método de ACLBD para insertarlo). Para crear un usuario (un Nick nuevo) hay que comprobar que no existe en la ACL. Si existe no se puede crear. No se puede modificar el NICK.
- El nombre será el mismo en usuario que en acl_usuarios (tras insertar/modificarlo en el CRUD, llamar al método apropiado de ACLBD).
- La contraseña se fija en la ACL. Llamaremos al método apropiado en ACLBD. Añadir el campo contraseña/confirmar contraseña en nuevoUsuario. Si queremos modificar la contraseña, para facilitar, podemos crear un formulario específico en modificarUsuario que lo permita.
- Al borrar un usuario se borra lógicamente en la tabla usuarios. Además, también se hace el borrado mediante el método apropiado de ACLBD.
- Añadir a nuevoUsuario y modificarUsuario un apartado para indicar el role (un select donde aparezcan los roles posibles). Al guardar llamaremos al método apropiado de ACLBD para actualizarlo