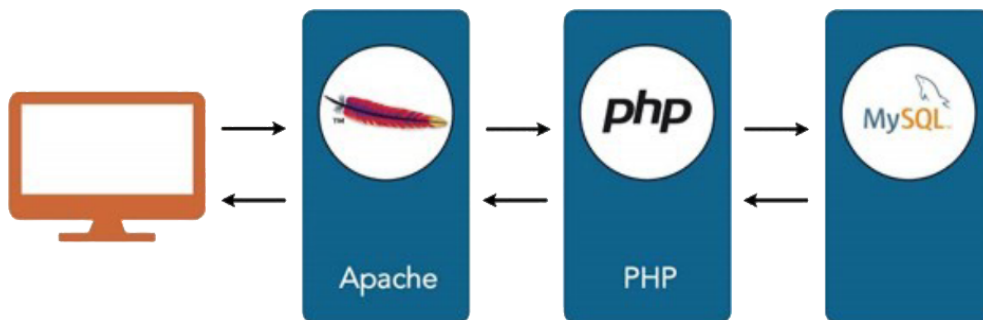


unidad didáctica 6

Actividades



1. [MySQLi](#)
2. [filtros y comodines](#)
3. [PDO](#)
4. [ficheros](#)
5. [proyecto CholloSevero](#)

1. MySQLi

Actividad 601

Crea una nueva base de datos con el nombre `lol` y cotejamiento de datos `utf8mb4_unicode_ci`.

Actividad 602

En nuestra base de datos `lol` que acabamos de crear, vamos a crear la tabla `campeon` con los siguientes campos (no olvides poner el tipo de datos de cada uno de los campos).

- `id` [*]
- `nombre`
- `rol`
- `dificultad`
- `descripcion`

[*] clave primaria.

Actividad 603

Rellena la tabla `campeon` con, al menos 5 registros, con los datos que tú quieras o si lo prefieres, puedes basarte en la [página oficial del juego](#) pero **¡¡ NO TE PONGAS A JUGAR !!**

Actividad 604

Crea `604campeones.php` y lista todos los campeones del LOL que has metido en tu base de datos (1º conexión a BD y 2º `foreach` para cada campeón que tengas en la tabla `campeon`).

Actividad 605

Modifica el archivo `604campeones.php` y guárdalo como `606campeones.php` pero pon al lado de cada uno de los campeones listados un botón para `editar` y otro para `borrar`. Cada uno de esos botones hará la correspondiente función dependiendo del id del campeón seleccionado.

- Al pinchar en editar, el usuario será redirigido al archivo `605editando.php` donde mostrará un formulario con los campos rellenos por los datos del campeón seleccionado. Al darle al botón de `guardar` los datos se guardarán en la base de datos y el usuario será redirigido a la lista de campones para poder ver los cambios.
- Al pinchar en borrar, el usuario será preguntado a través de un mensaje de JavaScript (prompt) si está seguro de que quiere borrar al campeón seleccionado. En el mensaje de confirmación debe aparecer el **nombre del campeón seleccionado**. Si el usuario pincha en Aceptar el campeón será eliminado de la base de datos y el usuario será redirigido nuevamente al listado de campeones para comprobar que, efectivamente dicho campeón se ha eliminado de la lista.

2. filtros y comodines

Actividad 606

Modifica el archivo `604campeones.php` y guárdalo como `606campeones.php` para que se muestre como una tabla con las columnas de la propia tabla de la base de datos, es decir; id, nombre, rol, dificultad, descripción. Al lado de cada nombre de cada columna, pon 2 iconos que sean \wedge \vee y que cada uno de ellos ordene el listado en función de cuál se haya pinchado.

- Si se ha pulsado en Nombre el icono de \wedge , el listado debe aparecer ordenado por nombre ascendente. Si por el contrario se ha pulsado \vee tendrá que ordenarse por nombre descendente.
- Ten en cuenta que cada icono debe llevar consigo un enlace al listado que contenga parámetros en la URL que satisfagan las opciones seleccionadas así que haced uso de `$_GET` para poder capturarlos y escribid las consultas SQL que sean necesarias para hacer cada uno de los filtros.
- Puedes usar [Font Awesome](#) para los iconos pero es algo opcional.

3. PDO

Actividad 607

Aprovecha lo que hiciste de los ejercicios 601 al 604 pero esta vez utilizando `PDO::FETCH_ASSOC`.

Actividad 608

Crea una tabla nueva dentro de la base de datos lol que ya tienes y crea un sistema de login con usuarios. Introduce en la base de datos al menos 3 usuarios diferentes con sus contraseñas distintas. Recuerda que:

- La tabla nueva ha de llamarse `usuario`.
- Los campos a crear en la nueva tabla deben ser
 - `id [*]`
 - `nombre`
 - `usuario`
 - `password`
 - `email`
- Las contraseñas deben ser cifradas antes de guardar el datos en la base de datos.
- Crea el formulario `608registro.php` donde el usuario introduzca los datos de registro y vincúlalo con `608nuevoUsuario.php` para que recoja los datos mediante POST y los inserte en la base de datos si todo ha ido bien.
- Queda **PROHIBIDÍSIMO** acceder a `608nuevoUsuario.php` sin el formulario rellenado.
- La sentencia de INSERT debe estar controlada para que no pueda introducirse ningún dato en blanco. Ten en cuenta que estás modificando la base de datos y no queremos campos mal rellenados.
- Si todo ha ido bien, muestra un mensaje por pantalla diciendo `El usuario XXX ha sido introducido en el sistema con la contraseña YYY`.

4. ficheros

Actividad 609

Entra en loremipsum.com y genera un texto de 3 párrafos. Copia el texto generado y guárdalo en un archivo nuevo llamado `609loremIpsum.txt`. Genera un archivo php llamado `609loremIpsum.php` y muestra por pantalla el texto del archivo txt que acabas de crear, su tamaño en **Kilobytes**, la fecha de su última modificación y el ID de usuario que creó el archivo.

Actividad 610

Vuelve a cargar el archivo `606campeones.php` y renómbralo a `610campeones.php` pero en vez de mostrar la tabla por pantalla, genera un archivo CSV `610campeones.csv` y otro `610campeonesCSV.php` donde saques por pantalla el contenido del archivo `610campeones.csv`.

5. proyecto CholloSevero

Actividad 615

Estructura el proyecto y piensa en las tablas y bases de datos que necesitéis para crear el proyecto. Crea los UML necesarios con nombres como `615UMLnombreTabla` metiendo todos los campos que se necesiten así como las relaciones que creas necesarias. Establece un sistema de archivos para el proyecto, teniendo en cuenta que van a haber imágenes, css, funciones php, constantes e incluso javascript (pero algo básico) para controlar los eventos del usuario a lo largo de la interfaz.

Actividad 616

Crea un sistema de login/password con los roles `administrador` y `usuario`. De momento que se validen los usuarios correctamente utilizando encriptación en la contraseña.

- `Administrador`: Puede ver todos los usuarios registrados así como los administradores y los chollos creados en la base de datos.
- `Usuario`: Puede ver sus propios chollos, editarlos y borrarlos, además de crear nuevos.

Actividad 617

Crea la vista para poner nuevos chollos y recuerda **sólo pueden entrar a esta vista usuarios registrados o administradores**.

Actividad 618

Crea la vista donde se muestren todos los chollos creados. Esta vista puede verla cualquier usuario, registrado o no en el sistema. Ten en cuenta que esta vista será la vista general de la web así que puedes llamarla `index.php` donde después aplicaremos filtros por `$_GET`.