

ACTIVIDADES UD5 - SERVIDOR

Mediante las tareas que se detallan en este documento vamos a conseguir la parte de los resultados de aprendizaje cubiertos por la UD5, atendiendo a sus correspondientes criterios de evaluación:

RESULTADOS DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN
RA6. Desarrolla aplicaciones de acceso a almacenes de datos, aplicando medidas para mantener la seguridad y la integridad de la información.	<p>a) Se han analizado las tecnologías que permiten el acceso mediante programación a la información disponible en almacenes de datos.</p> <p>b) Se han creado aplicaciones que establezcan conexiones con bases de datos.</p> <p>c) Se ha recuperado información almacenada en bases de datos.</p> <p>d) Se ha publicado en aplicaciones web la información recuperada.</p> <p>e) Se han utilizado conjuntos de datos para almacenar la información.</p> <p>f) Se han creado aplicaciones web que permitan la actualización y la eliminación de información disponible en una base de datos.</p> <p>g) Se han probado y documentado las aplicaciones.</p>

En este boletín de actividades vamos a revisar las tareas realizadas en la UD3 y la UD4, e introducir el acceso a BBDD en aquellos puntos en los que sea necesario.

Consideraciones adicionales:

- El criterio de evaluación a) se da por superado (calificación de 10) al hacer el ejercicio guiado de la parte de introducción-exposición.
- Se ha de continuar comentando el código conforme a la unidad anterior, mediante el código de tarea.
- No se va a valorar la estética de las funcionalidades elaboradas, solo que el resultado de cada actividad implemente las funcionalidades descritas.
- Cualquier copia (ya sea de fuentes externas, literal de los apuntes...) que no sea una respuesta original será calificada con 0. El código no estructurado, que presente dificultades para ser leído no podrá ser evaluado.
- Has de empezar las actividades basándote en tu última versión realizada hasta las

actividades de la UD4. Si has completado las actividades de la unidad anterior, toma la **versión mejorada**.

- En estas actividades vamos a ir insertando el acceso a BBDD en las tareas de las unidades anteriores que lo requieran. Por eso, se hace referencia a tareas de las unidades anteriores. Interceptaremos estos puntos y añadiremos el acceso a la BBDD.

Actividad 1: Comentarios en el código

Código: UD5.1**CE:** RA6.g**IE:** I3, I6**Puntos:** 10**Estimación:** 15 min

DESCRIPCIÓN

Para cada una de las actividades que siguen a ésta, inserta comentarios en el código para poder delimitar cada una de las soluciones que des. Para eso, inserta un comentario previo a cada fragmento de código que crees, especificando el código de cada tarea.

El código de cada tarea se formará con el código de la actividad, más la letra que identifica a la tarea en particular.

Ten en cuenta que pueden ser comentarios en PHP o en HTML.

Las respuestas que no sean identificables mediante comentario con el correspondiente código **podrán no ser evaluadas**.

ENLACES

- [Comentarios en PHP](#)
- [Comentarios en HTML](#)

Actividad 2: Conexión con BBDD

Código: UD5.2**CE:** RA6.b**IE:** I3, I6**Puntos:** 10**Estimación:** 15 min

DESCRIPCIÓN

En esta actividad vamos a estructurar el código que nos permite el acceso a la base de datos:

1. **(5 puntos)** Introduce la parte de acceso a la BBDD en el archivo db_access.php y crea dos funciones

- Función "open_connection" dentro de este fichero que cree una nueva conexión con la base de datos según el fichero de credenciales (db_credenciales.php), que tendrás que importar en db_access.php. La función ha de terminar la conexión si ha habido algún fallo, y en caso de éxito ha de devolver la conexión.
- Función "close_connection", a la que se le pasará el objeto de conexión por parámetro, y se le asignará el valor null, para cerrar la conexión.

Utiliza open_connection cada vez que necesites crear una conexión desde cualquier página PHP, y termina la conexión mediante close_connection, cuando acabes de realizar las operaciones en esa página PHP, de forma que establezcas una conexión por página, y la cierres.

2. **(5 puntos)** Estructura las operaciones sobre la base de datos en diferentes archivos dependiendo de cada tabla, y nombra las variables que almacenan las sentencias según se ha explicado en el documento de teoría (sufijos significativos como _list, _detail, _create, _update, _delete, etc). También las funciones que ejecutan las sentencias SQL y que reciben como parámetro la conexión a la BBDD.

Evita dispersar sentencias SQL a lo largo de los otros ficheros que manejan la lógica de negocio de la aplicación. Las tareas cuyas sentencias SQL no se encuentren debidamente organizadas no se considerarán válidas.

Actividad 3: Recuperación de información

Código: UD5.3**CE:** RA6.c**IE:** I3, I6**Puntos:** 10**Estimación:** 3 h

DESCRIPCIÓN

- a) (UD3.3.a y UD3.3.b) Inserta manualmente en la BBDD los datos que utilizaste en la tarea UD3.3.a y UD3.3.b. Además, rellena también la tabla de categorías y categorías por proyecto, con datos de tecnologías reales. **(1 punto)**
- b) (UD3.3.e) Implementa una función que te permita recuperar todas las categorías, llamada get_categorias_all. **(0.5 puntos)**
- c) (UD3.3.f) Implementa una función que te permita recuperar todos los proyectos que estén asociados a una determinada categoría. Llama a esta función get_proyectos_por_categoria **(1 punto)**
- d) (UD3.5.c) Implementa una función que te permita recuperar los proyectos ordenados de forma ascendente o descendente (ya sea por título o fecha). Esta función ha de aceptar por parámetro el tipo de ordenación (además del objeto de conexión). La ordenación no se realizará sobre el array recuperado, sino en la propia consulta SQL. Llama a esta función get_proyectos_order_by **(1.5 puntos)**
- e) (UD4.1.b) En el proceso de login, se leerá el usuario y la contraseña de la tabla de usuario, buscando por e-mail. Se introducirán varios usuarios en la BBDD, mediante phpMyAdmin. **NO** recuperes los datos de todos los usuarios y luego filtres el array resultante mediante código PHP, esto puede conllevar problemas de seguridad y es ineficiente. Recupera solo un registro de la BBDD (si existe) que corresponda al e-mail introducido por el usuario. Llama a esta función get_credenciales_usuario **(2 puntos)**
- f) (UD3.6) Implementar una función que nos permita recuperar los proyectos de forma paginada, llamada get_proyectos_paginados. Para ello, tendrás que: **(2 puntos)**
 - Determinar cuántos proyectos queremos mostrar por cada página. Lo puedes almacenar como parámetro de configuración ya sea en una tabla nueva, o en un fichero dedicado a ello.
 - Utilizar las directivas LIMIT y OFFSET (ver enlaces).
- g) Crea una nueva función llamada "get_user_logged_in" en utiles.php que realice lo siguiente: **(2 puntos)**
 - Comprobación de que la cookie de usuario existe y tiene un valor.
 - En caso de que la cookie tenga un e-mail, comprobará en BBDD que existe un registro en la tabla de sesiones, con referencia al usuario con el e-mail almacenado en la cookie.
 - La función devolverá:
 - a) **Falso:** si no existe la cookie o el registro de usuario en la tabla de sesiones
 - b) **Verdadero:** si existe la cookie con un e-mail, existe una sesión asociada a ese e-mail, y además el usuario tiene verdadero en la columna admin.
 - Utiliza esta función para comprobar que el usuario es administrador y además tiene una sesión abierta, en las siguientes tareas: UD5.4.e, UD5.5.b, UD5.6.b, UD5.6.c, UD5.6.d, UD5.7.

ENLACES

- [LIMIT y OFFSET](#)

Actividad 4: Visualización de información

Código: UD5.4	CE: RA6.d	IE: I3, I6	Puntos: 10	Estimación: 3h
----------------------	------------------	-------------------	-------------------	-----------------------

DESCRIPCIÓN

- (Relacionado con UD3.3.e, UD3.3.f y UD5.3.b) Construye el dropdown de categorías a partir de la consulta implementada en UD5.3.b., sustituyendo lo desarrollado en las tareas UD3.3.e y UD3.3.f (utiliza ahora el id de categoría como la clave de categoría) **(1 punto)**
- (Relacionado con UD3.3.f y UD5.3.c) Muestra los proyectos filtrados por categoría a partir de la consulta implementada en UD5.3.c. Respeta lo desarrollado en la tarea UD3.2.g. El filtrado se ha de producir al pulsar sobre: **(2 puntos)**
 - el desplegable de categorías
 - al pulsar sobre alguno de los enlaces con clase "badge" y el nombre de la categoría asociada, tanto en index.php como en proyecto.php.
- (Relacionado con UD3.5.c y UD5.3.d) En index.php, utiliza los botones de ordenación para ordenar los proyectos por título o fecha (de forma ascendente o descendente). La ordenación se ha de producir en la consulta a BBDD, no sobre el array recuperado. **(2 puntos)**
- (UD3.6) Implementar la paginación de los proyectos según se detalla en la actividad de ampliación de la UD3, y utilizando la función de UD5.3.f. **(2 puntos)**
- (Relacionado con UD5.3.b, UD4.2.f y con UD5.7) En el formulario de proyecto (en modo creación/actualización), vamos a implementar un [campo multiselección](#) basado en bootstrap. **(3 puntos)**
 - Para listar todas las categorías, tendrás que recuperar los registros de la tabla de categorías (función implementada en UD5.3.b) y formar el elemento select, teniendo un elemento option por cada categoría.
 - El atributo value de cada option será el id de la categoría.
 - Además, para marcar qué categorías están asociadas al proyecto, tendrás que imprimir el atributo "selected" en función de que esa determinada categoría esté asociada al proyecto (tabla de categorías por proyecto).
 - Debes haber creado ya algunos registros de prueba para poder probarlo.

ENLACES

- Para el apartado e) puedes consultar un ejemplo en este [enlace](#).

Actividad 5: Almacenar información

Código: UD5.5	CE: RA6.e	IE: I3, I6	Puntos: 10	Estimación: 1h
----------------------	------------------	-------------------	-------------------	-----------------------

DESCRIPCIÓN

- (UD4.1.c) En caso de que el login sea satisfactorio (comprobación de usuario y

contraseña contra los datos de la tabla de usuario), se insertará un nuevo registro en la tabla de sesiones, estableciendo una referencia al registro correspondiente de la tabla de usuario, mediante el ID de usuario. Se respetará el funcionamiento detallado en UD4.1.c, UD4.1.d y UD4.1.e. La función de inserción en la BBDD se llamará new_sesion, y no ha de incluir la creación de la cookie (**5 puntos**)

- b) (Relacionado con UD4.2.f y UD5.4.e) En la creación de un nuevo proyecto (solo disponible cuando la opción `get_user_logged_in` devuelve verdadero), se insertará un nuevo registro en la BBDD. Se recuperará el ID generado (consulta este [enlace](#)) y el formulario pasará a estar en modo actualización, mostrando los datos del proyecto creado. Respeta la página de confirmación de la tarea UD4.2.f. El ID generado se puede guardar en un input oculto, o mediante la URL. Por el momento, el campo multiselección de categorías no se guardará en la BBDD. La función de inserción se llamará new_proyecto (**5 puntos**)

Actividad 6: Actualizar y eliminar información

Código: UD5.6

CE: RA6.f

IE: I3, I6

Puntos: 10

Estimación: 1.5 h

DESCRIPCIÓN

- a) (UD4.1.f) Al realizar el LOG OUT (solo disponible cuando la opción `get_user_logged_in` devuelve verdadero), se eliminará tanto la cookie que almacena el e-mail del usuario, como el registro correspondiente de la tabla de sesiones. La función que elimina el registro de sesión se llamará delete_sesion, y no ha de incluir la eliminación de la cookie (**2 puntos**)
- b) (UD4.2) Crea un botón en el formulario del proyecto mediante el cual se pueda eliminar un proyecto (solo disponible cuando la opción `get_user_logged_in` devuelve verdadero). Dicho botón solo estará habilitado si el proyecto existe en la BBDD (tiene un id asociado). Una vez eliminado el proyecto, se redirigirá la navegación a `index.php`, donde se mostrará un mensaje indicando que se ha eliminado el proyecto. Elimina la funcionalidad implementada en UD3.3.h. La función se llamará delete_proyecto (**4 puntos**)
- c) (Relacionado con UD5.5.b y UD5.4.c) Tras la creación de un nuevo proyecto (solo disponible cuando la opción `get_user_logged_in` devuelve verdadero), se debe poder actualizar, dependiendo de si existe o no el id. La función para actualizar se llamará update_proyecto. Se respetará lo establecido en UD4.2.f. Por el momento, el campo multiselección de categorías no se guardará en la BBDD. (**2 puntos**)
- d) (UD4.3.b) El formulario de usuario actualizará ahora el registro correspondiente de la tabla de usuarios (solo disponible cuando la opción `get_user_logged_in` devuelve verdadero). El procedimiento será el mismo que para la tarea UD4.3.c: se recuperará el e-mail del usuario de la cookie, y se consultarán los datos del usuario en la BBDD a partir de ese e-mail. La función para actualizar se llamará update_usuario (**2 puntos**)

Actividad 7: Categorías por proyecto - AMPLIACIÓN

Código: UD5.7	CE: RA6.f	IE: I3, I6	Puntos: 10	Estimación: 1.5 h
DESCRIPCIÓN (Relacionado con 5.3.c) En el formulario de proyecto, vamos a utilizar dos tablas cuando salvemos el proyecto (tanto creación como modificación): <ul style="list-style-type: none">• La tabla de proyecto, donde se guardarán todos los campos propios del proyecto: título, fecha, descripción, imagen. (4 puntos)• La tabla de categorías de proyecto. Mediante el campo multiselección se recogerá un array en el que se guardarán los ids de las categorías seleccionadas. Al salvar el proyecto se puede realizar lo siguiente: borrar todas las categorías asociadas al proyecto, y volver a insertar tantos registros como ids de categorías se han recogido en el array. Consulta el ejemplo en este enlace. (6 puntos)				