



Importante: Compruebe las instrucciones que se dan para realizar la entrega, siga las mismas, márkelas cuando las haya comprobado y firme en la casilla correspondiente. Cualquier entrega que no cumpla las reglas de nombrado, el formato de compresión del archivo o el contenido de los archivos del mismo, será penalizada con 2 puntos sobre 10 por cada incumplimiento. Pasado el límite de entrega se aceptará el envío del trabajo, con una penalización de un 2 puntos sobre 10 de la calificación por cada minuto o fracción de retraso a partir del tercer minuto.

<ul style="list-style-type: none"><input type="checkbox"/> Restricciones: Las rutas de los ficheros empleados serán relativas, a fin de que su solución pueda ser examinada en cualquier equipo.<input type="checkbox"/> Material a entregar: El resultado de esta prueba consistirá en un proyecto de NetBeans que será comprimido en formato ZIP conteniendo la aplicación. Emplee las credenciales por defecto de XAMPP para la conexión con la base de datos.<input type="checkbox"/> Formato de entrega: Archivo comprimido ZIP.<input type="checkbox"/> Nombre del archivo de entrega: El nombre del fichero tendrá un formato específico dictado por el nombre de cada alumno. Por ejemplo, para un alumno llamado "José María Núñez Pérez" el fichero se nombrará como NunyezPerezJM.zip. Obsérvese que las tildes son ignoradas y las eñes sustituidas.<input type="checkbox"/> Vía entrega: La entrega del trabajo se hará a través de la tarea correspondiente en el Campus Virtual y se completará cuando se reciba por correo electrónico resguardo de la entrega. Conserve dicho correo como prueba del envío.<input type="checkbox"/> Comprobación de la entrega: Los alumnos deben comprobar que el archivo depositado en la plataforma es correcto descargándolo de la misma, descomprimiéndolo y comprobando su contenido.	Firma del alumno:
---	-------------------

- Demostrar los conocimientos adquiridos en el desarrollo de aplicaciones empleando PHP.

Llamamos tarea CRON a aquel *script* que se programa y ejecuta de forma periódica y que, por lo general, tiene una larga duración.

Se solicita la elaboración de una aplicación web (desarrollada en PHP) que permita a los usuarios gestionar tareas CRON. Aunque esta aplicación se implantará dentro de la intranet corporativa, la sensibilidad de la información es crítica, por lo que hay necesidad de establecer un mecanismo de autenticación. En la base de datos ya tiene cargado al usuario "root@entidad.com" con contraseña: "P4pCr0N 21".

Encontrará en el **material adicional** la **base de datos** exportada (con datos de ejemplo) y las **vistas que se han de generar**, desde las que podrá extraer el código HTML a emplear (use dicho código para no perder tiempo en esto).

Ej.1 [2 puntos] Cree la página “*login.php*” con la siguiente funcionalidad:

- **(0,50 puntos).** Mediante un formulario, se solicitará al usuario su email corporativo para acceder al sistema. Al pulsar el botón “Acceder” se *comprobará* que el email introducido es correcto (que tenga formato de email). Para ello, debe usar el paquete *filter*.
- **(0,50 puntos).** Si todo es correcto, se creará e iniciará la sesión y llevará al usuario a “*inicio.php*”, estableciendo una variable de sesión con el email usado, representando una sesión activa. Compruebe si el *hash* de la contraseña es el adecuado usando `password_hash($contrasenia, PASSWORD_BCRYPT)` y `password_verify($hash_contrasenia_introducida, $hash_contrasenia_en_bd)`. Podrá encontrar más información en el material adicional.



- **(0,50 puntos).** En caso de no haber introducido un email válido o si la combinación email/contraseña no se encuentra en la base de datos, se mostrará un mensaje describiendo el error y se permanecerá en la página de “login.php”.
- **(0,50 puntos).** Se asegurará que el sistema no puede verse comprometido con ataques de inyección SQL.

Ej.2 [1 punto] Cree la página “seguridad.php” con la siguiente funcionalidad:

- **(0 puntos)** Incluya este fichero en todas las páginas de PHP generadas para el proyecto.
- **(1 punto)** Realice un código que compruebe si el usuario tiene una sesión activa. En caso negativo, deberá redirigir al usuario a “login.php”.

Ej.3 [1 punto] Cree la página “logout.php” con la siguiente funcionalidad:

- **(1 punto)** Esta página cerrará la sesión activa y devolverá automáticamente (emplee cabeceras para ello) al usuario a la página de login.

Ej.4 [1 punto] Cree la página “inicio.php” con la siguiente funcionalidad:

- **(0,50 puntos)** Mostrará el listado de los registros de tareas CRON existentes según un filtro dado (o todas si no se ha seleccionado un filtro). Para seleccionar el filtro, use el desplegable que encontrará en la vista con los distintos estados de funciones CRON. Implemente una función en PHP “get_cron_by_state” que dado un estado obtenga todas las tareas CRON asociadas. Use dicha función para buscar las tareas CRON según el valor introducido en el filtro.
- **(0,50 puntos)** Por cada registro de tarea con estado “ESPERANDO EJECUCIÓN”, se dispondrá de un botón “Borrar” y al pulsar dicho botón se navegará a la página “borrarTarea.php” enviando por “GET” el identificador del registro a borrar.

Ej.5 [3 puntos] Cree la página “crearTareas.php” con la siguiente funcionalidad:

- **(0,50 puntos).** Cree un formulario para el alta de tareas CRON. Aquellos campos que no aparezcan en el formulario se deben cargar a NULL. Recuerde que la “ID” es autogenerada. Al pulsar el botón “Guardar” se creará un nuevo registro en la base de datos, dentro de la tabla “tarea”, y volverá a la página de inicio.
- **(0,50 puntos).** En caso de que algún campo no haya sido informado, se mostrará un mensaje indicando el error y los campos introducidos hasta el momento se auto-completarán para que pueda corregirse el problema y continuar la creación del registro.
- **(0,50 puntos).** Valide que el formato de la fecha es correcto respecto dd/mm/aaaa (por ejemplo 21/12/2020). Puede encontrar la expresión regular en el material adicional. Considere que, al guardar los resultados, los campos del tipo fecha están formateados como “aaaa-mm-dd” dentro de la base de datos.
- **(0,50 puntos).** Del mismo modo, verifique que se trata de un día válido (por ejemplo, el 30/02/2020 no sería un día válido). Haga uso de la función *checkdate* para ello. Podrá encontrar más información de esta función en el material adicional.
- **(0,50 puntos).** En caso de existir fecha de fin, calcule el número de días que la tarea CRON está activa.
 - Para calcular el número de días, reste la fecha de inicio de la fecha fin de la tarea CRON. Es posible restar dos fechas usando el siguiente código:

```
$date1 = date_create("dd-mm-aaaa");  
$date2 = date_create("dd-mm-aaaa");  
$diff = date_diff($date1,$date2);  
$num_days = date_interval_format($diff, "%a");
```
 - Indique un error en caso de que la resta calculada tenga como resultado un número inferior o igual a cero.
- **(0,50 puntos).** Se asegurará que el sistema no puede verse comprometido con ataques de inyección SQL.

Ej.6 [1 punto] Cree la página “borrarTarea.php” con la siguiente funcionalidad:

- **(0,50 puntos).** Al acceder a esta página se mostrarán los datos del registro a borrar, obteniendo todos los datos del mismo a partir del identificador recibido a través de ‘GET’, mostrando a continuación el enlace “Cancelar” y el botón “Confirmar borrado”.
- **(0,50 puntos).** En caso de hacer clic en “Confirmar borrado”, los datos de la tarea CRON a borrar serán eliminados de la base de datos, y se volverá a la página principal.

Ej.7 [1 punto] Implemente un botón dentro de “inicio.php” llamado “Actualizar”:

- Al pulsar sobre el botón se ejecutará la función “update_state”. Esta función realizará los siguientes puntos:
 - Almacenará los IDs de las tareas CRON con estado “ESPERANDO EJECUCIÓN” en un vector.



- Seleccionará una posición del vector de forma aleatoria. Puede usar la función *random_int* para generar números enteros aleatorios. Podrá encontrar más información de esta función en el material adicional.
- Una vez localizada de forma aleatoria la tarea CRON, cambiará ésta del estado “ESPERANDO EJECUCIÓN” a “CERRADA”.
 - Coloque la “fecha_fin” de la tarea CRON en el día de hoy usando la función *date("Y-m-d")*.
 - Para calcular el número de días, reste la fecha de inicio de la fecha fin de la tarea CRON. Es posible restar dos fechas usando el código visto en el ejercicio 5.

Material suministrado

Como material adicional dispondrá del fichero “Material Adicional.7zip” que contendrá:

- Carpeta **BBDD**: fichero *registro_tareas.sql*: Script de creación de la base de datos. Dispone de datos de ejemplo para que pruebe su solución.
- Carpeta **Vistas**: ficheros *login.html*, *inicio.html*, *crearTareas.html*, *borrarTarea.html*, con el código HTML de cada vista.
- Carpeta **Documentación**:
 - Fichero *Manual_checkdate.pdf*: con información sobre la función.
 - Fichero *Manual_password_hash.pdf*: con información sobre la función.
 - Fichero *Manual_password_verify.pdf*: con información sobre la función.
 - Fichero *Manual_random_int.pdf*: con información sobre la función.
 - Fichero *expresiones_regulares.txt* con las expresiones regulares para validación de:
 - Validación de fechas.

Descripción de las tablas de la base de datos

Tabla “*tarea*”

- ID
 - Tipo: INTEGER
 - Longitud: 10
 - Campo auto generado y auto incremental
- nombre
 - Tipo: VARCHAR2
 - Longitud: 255 caracteres
- fecha_ini
 - Tipo: DATE
- fecha_fin
 - Tipo: DATE
 - Puede ser nulo
- duracion_dias
 - Tipo: INTEGER
 - Longitud: 10
 - Puede ser nulo
- prioridad
 - Tipo: VARCHAR2
 - Longitud 255
 - Valores que puede recibir:

- ALTA
 - MEDIA
 - BAJA
- estado
 - Tipo: VARCHAR2
 - Longitud: 100
 - Valores que puede recibir:
 - CERRADA
 - ESPERANDO EJECUCIÓN
 - EN EJECUCIÓN

Tabla “*usuario*”

- email:
 - Tipo: VARCHAR2
 - Longitud: 255 caracteres
 - Este campo representa la clave primaria.
- contrasenia_hash
 - Tipo: VARCHAR2
 - Longitud: 255 caracteres