

# EJERCICIO: APLICACIÓN

Para las dos partes de esta aplicación, tendrás en cuenta lo siguiente:

- Utilizarás el patrón POST-redirect-GET con mensajes flash y persistencia de la información
- Evitarás la inyección de HTML y de SQL
- Utilizarás la clase PDO de PHP
- En todos los accesos a la base de datos, gestionarás los errores mediante excepciones.
- No permitirás entrar en los programas de gestión de la base de datos si no estás logado con el usuario correcto.

## PRIMERA PARTE: Gestión de usuarios

Vas a realizar una aplicación que gestione los usuarios y sus passwords. Para ello utilizarás la tabla de usuarios de tu base de datos.

Cuando termines, elimina todos los usuarios de la tabla de usuarios, y luego incluye algunos con esta aplicación, para asegurarte de que las passwords están correctamente encriptadas.

### loginadmin.php

Harás un primer programa login.php, que te pedirá usuario y password.

Tú serás el administrador del sistema, por lo que entrarás con el usuario: “admin” y el password: “admin”. No se admitirá ningún otro.

### logout.php

Este programa de logout, después de eliminar la sesión, redirigirá al programa de loginadmin.php

### usuarios.php

Este programa es similar al que hemos trabajado en las transparencias, a excepción del almacenamiento de la password, que irá encriptada:

- Muestra arriba los usuarios almacenados en la tabla
- Permite añadir un nuevo usuario
- Permite eliminar un usuario
- Tendrá un botón que permitirá hacer logout

← → ↻ ⓘ localhost/marta/usuarios.php

## Gestión de Usuarios

### Usuarios

Marta Olmedilla	marta@europa.es	\$2y\$10\$WMbr/RzR0LCRJnmO2vddp.Ou0uOnfcZeoeNs2VqDF3f9gPdg4d2de	Borrar
alumno	alumno@europa.es	\$2y\$10\$hiYSpH/XQcMHHzpQArYgLOJbQeK8xAZ4tHxc6wQhrkBccJou6iYkm	Borrar

#### Añade un nuevo usuario

Nombre :   
Email :   
Password :

### Encriptación de la password:

La password que teclea el usuario no es la que se guarda en la tabla. El usuario escribirá una password normal, pero el programa almacenará una hash salada de la misma. Para ello utilizará la función **password\_hash** de PHP:

```
$paraGuardarEnBD = password_hash($password, PASSWORD_DEFAULT);
```

El parámetro **PASSWORD\_DEFAULT** realizará un hash salado de la password. El algoritmo utilizado irá cambiando con las versiones de PHP, adaptándose a criterios de mayor seguridad. El manual de PHP recomienda utilizar un string de **longitud 255** para almacenar la password. Comprueba que tu tabla admite esa longitud.

## SEGUNDA PARTE:

Vas a hacer una aplicación para un usuario que gestiona los datos de los institutos.

Utilizarás la tabla de institutos de tu base de datos.

### loginapp.php

Este programa pedirá un email y una password.

Accederá a la tabla de usuarios para encontrar el registro correspondiente a ese email. Luego recuperará la password encriptada. Luego verificará que la password del usuario corresponde con la encriptada de la base de datos. Para ello, utilizarás la función **password\_verify** de PHP:

```
$trueOfalse = password_verify($passwordUsuario, $passwordBD);
```

### Institutos.php

Realizarás un formulario similar al de la primera parte, pero esta vez sobre la tabla institutos y sus datos.

## TERCERA PARTE:

Harás un menú que permita seleccionar entre la administración de usuarios y la aplicación de institutos. Desde el menú se accederá a la página de login correspondiente.



Añadirás un enlace con el texto “Volver” al resto de programas de tu aplicación:

