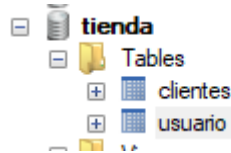


Taller 4

Taller de PHP

Insertando Registros

El objetivo de este taller es aprender a diseñar páginas de registro a una determinada aplicación. Suponemos que trabajamos con la base de datos “tienda” que ya tenemos creada y creamos una nueva tabla llamada “usuario”

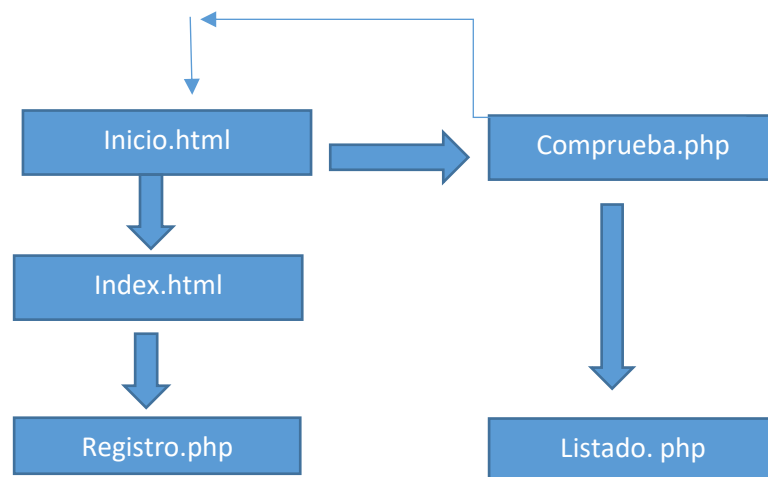


Cuya estructura es la siguiente

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?
<input type="checkbox"/>	usuario	varchar	12		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	nombre	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	clave	varchar	12		<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>

Como ves, el campo clave es **usuario** ya que no tendría sentido dos usuarios con el mismo nombre de usuario pues generaría un problema de seguridad.

En este ejemplo, tendremos varias páginas webs, como se muestra en la figura



la primera, inicio.html, es la página que arranca la aplicación y se encarga de mostrar el formulario para introducir los datos de autenticación, con un enlace a la página de registro (index.html) en caso de que el usuario no esté registrado

Credenciales

Usuario

Clave

Identificarme

[o Registrate](#)

Desde esta página hay, por tanto, dos vías: registrarse o loguearse. A la primera vía (registrarse) se accede pinchando en el enlace inferior y contiene dos páginas, index.html que presenta el formulario de registro y registro.php que coge los datos del formulario y los guarda en mysql

El aspecto de la página index.html es el siguiente

Registro

Usuario

Nombre Completo

Clave

Registrarse

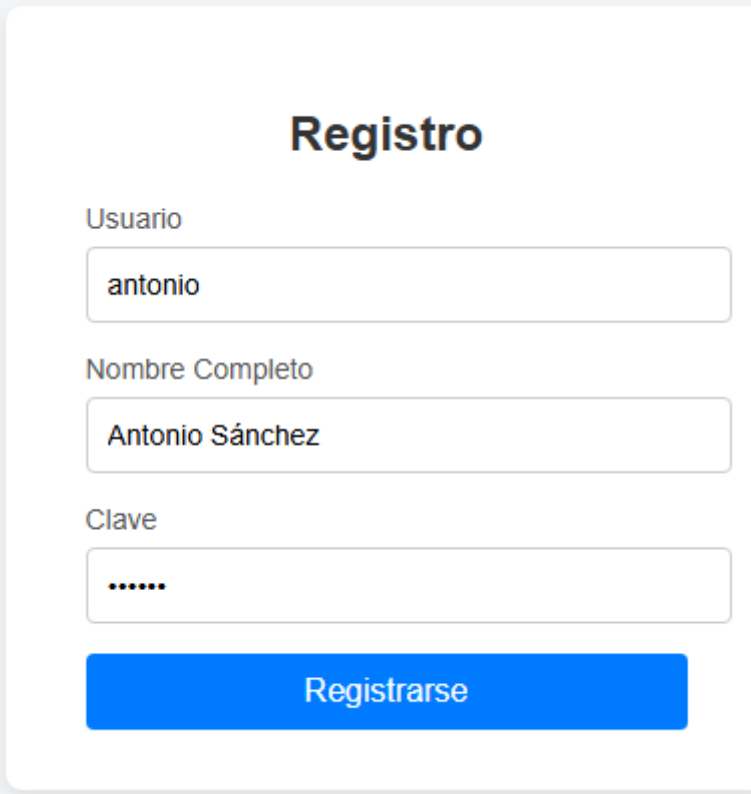
Se trata, por tanto, de un formulario en el que, **IMPORTANTE**, debe ser de tipo POST para asegurarnos de que nadie puede ver la clave que escribe el usuario a través de la URL.

```
<form action="registro.php" method="POST">
  <label for="usuario">Usuario</label>
  <input type="text" id="usuario" name="usuario" required>

  <label for="nombre">Nombre Completo</label>
  <input type="text" id="nombre" name="nombre" required>

  <label for="clave">Clave</label>
  <input type="password" id="clave" name="clave" required>

  <input type="submit" value="Registrarse">
</form>
```



The image shows a visual representation of the registration form. It features a light gray rounded rectangle with a title 'Registro' in bold black text. Below the title are three input fields: 'Usuario' with the value 'antonio', 'Nombre Completo' with the value 'Antonio Sánchez', and 'Clave' with masked characters '.....'. At the bottom is a blue button with the text 'Registrarse'.

Al pulsar el botón se envía dicho formulario a la página registro.php cuyo código incluye la conexión a mysql

```
$conexion = mysqli_connect($host, $usuario, $contrasena, $base_datos);
```

```
// Verificar la conexión
```

```
if (!$conexion) {  
    die("Error de conexión: " . $conexion->connect_error);  
}
```

Recoger los datos enviados desde el formulario. No olvides que el formulario es de tipo POST

```
$u=$_POST['usuario'];  
$n=$_POST['nombre'];  
$c=$_POST['clave'];
```

Definir la consulta y lanzarla

```
$sql = "insert into usuario values('$u','$n','$c')";  
$resultado = $conexion->query($sql);
```

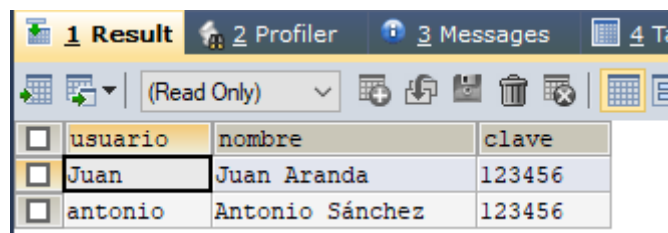
Finalmente, comprobamos que la operación ha tenido éxito. Podemos hacerlo con el método

```
$l=mysqli_affected_rows($conexion);
```

Si hemos conseguido insertarlo, el número de filas afectadas sería, evidentemente, uno. En caso de error obtendríamos 0 como respuesta. No es posible obtener un valor diferentes de 0 o 1. Por tanto,

```
if ($l==1) header("Location: index.html");  
else echo "<center><h1>No se ha podido insertar el registro</h1></center>";
```

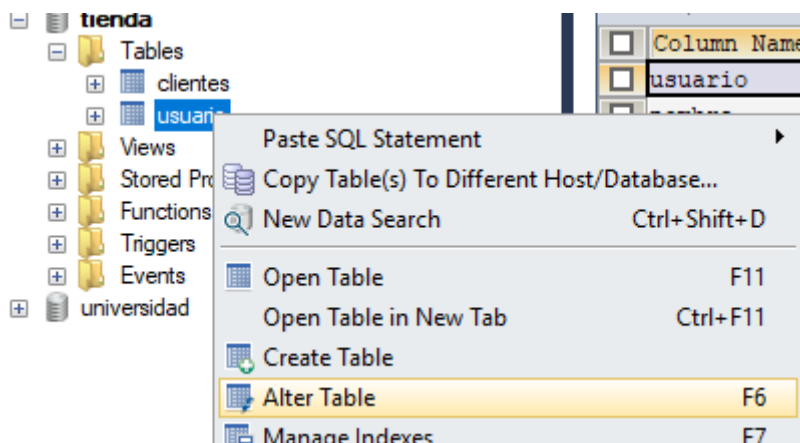
Podemos comprobar



usuario	nombre	clave
Juan	Juan Aranda	123456
antonio	Antonio Sánchez	123456

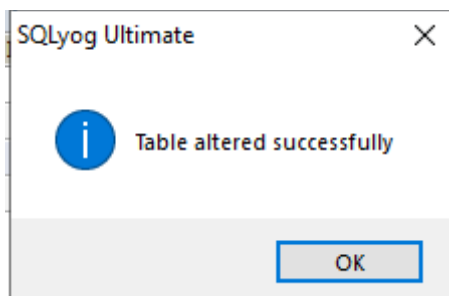
Que se ha registrado correctamente el nuevo usuario.

Posibles mejoras incluyen codificar la clave en MD5 o SHA1 para que no sea posible ver cuál es su valor. Para ello, tendremos que modificar la tabla ya que el hash de MD5 necesita más longitud.



Podemos ponerle, por ejemplo, 100 bytes de longitud

<input type="checkbox"/>	Column Name	Data Type	Length	Default
<input type="checkbox"/>	usuario	varchar	12	
<input type="checkbox"/>	nombre	varchar	30	
<input type="checkbox"/>	clave	varchar	100	
<input type="checkbox"/>				



Ahora tendremos que modificar el código php para codificarla

```
$codificada=md5($c);
```

e incluir este nuevo valor en la consulta SQ en lugar de la variable enviada por el formulario

```
$sql = "insert into usuario values('$u','$n','$codificada')";
```

Los registros ya insertados no se ven afectados y mantendrán el mismo valor, pero, a partir de ahora, el campo clave aparecerá como una cadena codificada que representa al valor introducido por el usuario que se registra.

Volvemos a lanzar la página principal con nuevos datos

Registro

Usuario

Nombre Completo

Clave

Y vemos que se guardan en Mysql, pero con el campo clave codificado.

<input type="checkbox"/>	profesor	Alejandro de la Torre	e10adc3949ba59abbe56e057f20f883e
*	(NULL)	(NULL)	(NULL)

Usar MD5 para contraseñas no es seguro. Es un algoritmo antiguo, rápido y vulnerable a ataques de diccionario y fuerza bruta. Hoy en día se recomienda usar `password_hash()`, `bcrypt`, `argon2`, etc.

En el tema de criptografía veremos en profundidad todos los algoritmos de HASH para poder mejorar la página.

La segunda vía está formada por la autenticación y se accede pulsando el botón "Identificarme". Esto hace que se invoque a la página `comprueba.php` que lanza una consulta para seleccionar el usuario que intenta loguearse.

Credenciales

Usuario

pepe

Clave

.....|

Identificarme

[o Regístrate](#)

La estructura de esta página es la siguiente:

1.- Recoge los datos enviados desde el formulario

```
$u=$_POST['usuario'];  
$c=$_POST['clave'];
```

y codifica la clave para poder compararla con lo que hay guardado en Mysql

```
$codificada=md5($c);
```

2.- Se conecta a Mysql

```
$host = "localhost";    // Servidor (por ejemplo, 127.0.0.1)  
$usuario = "root";      // Usuario de MySQL  
$contrasena = "";       // Contraseña del usuario  
$base_datos = "tienda"; // Nombre de la base de datos
```

```
// Crear la conexión
```

```
$conexion = mysqli_connect($host, $usuario, $contrasena, $base_datos);
```

```
// Verificar la conexión
```

```
if (!$conexion) {  
    die("Error de conexión: " . $conexion->connect_error);  
}
```


3.- Lanzamos un consulta para comprobar que el usuario existe

```
$sql = "select * from usuarios where usuario='$u'";  
$resultado = $conexion->query($sql);
```

Si existe, el número de filas devueltas debe ser 1 ya que el campo usuario es el campo clave y no puede repetirse.

```
$l=mysqli_affected_rows($conexion);
if ($l==1) {
    $fila= $resultado->fetch_array();
    if($fila[1]==$codificada) header("Location: listado.html");
}
else echo "<center><h1>No existe usuario o la clave es incorrecta</h1></center>";
```

En este caso, recogemos la fila devuelta y comprobamos si la clave guardada en mysql (\$fila[1]) coincide con la clave aportada por el usuario una vez codificada en MD5 (\$codificada). En caso de que sean iguales resulta que el usuario si está registrado y lo enviamos a la página listado.php de entrada a la aplicación (que ya está resuelta en un ejercicio anterior)



Código	Nombre	Localidad	
C0012	Dani Baena	Grand Blanc, MI 48439	<i>i</i>
C0013	Angelina de Arregui	Dalton, GA 30721	<i>i</i>
C0014	Samuel de Carranza	Marcus Hook, PA 19061	<i>i</i>
C0015	Jacinta Montenegro Garcés	Reynoldsburg, OH 43068	<i>i</i>

En caso de que la contraseña no coincida, mostramos un mensaje de error y un posible botón para volver a la página de registro

