



PROGRAMACIÓN WEB 2

Registro

PREPARED STATEMENTS

Sentencias preparadas son una forma segura de ejecutar consultas a bases de datos donde los datos del usuario se envían por **separado del código SQL**. Se usan mucho con MySQL y lenguajes como PHP para evitar el problema más común en sistemas de registro: la inyección SQL.

El SQL se prepara primero **con marcadores (?)** y después se envían los datos aparte.

EJEMPLO

```
1
2  <?php
3
4  $usuario = $_POST['usuario'];
5  $sql = "SELECT * FROM usuarios WHERE usuario = '$usuario'";
6
7  ?>
```

Si alguien escribe → ' OR '1'='1

La consulta se convierte seria → SELECT * FROM usuarios WHERE usuario = " OR '1'='1';

Devuelve todos los usuarios.

El SQL se prepara primero con marcadores (?) y después se envían los datos aparte.

EJEMPLO

```
1
2  <?php
3
4  $usuario = $_POST['usuario'];
5  $sql = "SELECT * FROM usuarios WHERE usuario = '$usuario'";
6
7  ?>
```

```
2  <?php
3
4  $stmt = $conexion->prepare("SELECT * FROM usuarios WHERE usuario = ?");
5  $stmt->bind_param("s", $usuario);
6  $stmt->execute();
7  ?>
```

El ? es un espacio reservado

El valor del usuario nunca se mezcla con el código SQL

La base de datos lo trata solo como dato, no como instrucción

EJEMPLO

```
2  <?php
3
4  $stmt = $conexion->prepare(
5      "INSERT INTO usuarios (usuario, password, correo) VALUES (?, ?, ?)"
6  );
7
8  $stmt->bind_param("sss", $usuario, $password, $correo);
9  $stmt->execute();
10
11  ?>
```

Los tipos de preparación de datos:

s = string

i = int

d = double

b = blob

\$_SESSION

Es una variable superglobal de PHP que permite guardar información del usuario mientras navega por el sistema.

Sirve para mantener al usuario recordado o guardado entre páginas después de iniciar sesión.

HTTP es sin estado: cada página que se carga olvida lo anterior.

\$_SESSION resuelve eso guardando datos en el servidor y asociándolos a un ID de sesión del usuario.

SINTAXIS

```
2  <?php
3
4  session_start();
5
6  $_SESSION['usuario'] = $usuario;
7  $_SESSION['rol'] = 'admin';
8
9  ?>
```

`session_start();` → Inicializa la variable superglobal
`$_SESSION['usuario'] = $usuario;` → guardamos el usuario
`$_SESSION['rol'] = 'admin';` → asignamos los permisos
Se utiliza cuando se inicia sesión por primera vez (login)

```
2  <?php
3
4  session_start();
5  echo $_SESSION['usuario'];
6
7  ?>
8
```

Verifica si la sesión esta iniciada y se utiliza en la paginas restringidas para acceder

SINTAXIS

```
2  <?php
3
4  session_start();
5
6  √ if (!isset($_SESSION['usuario'])) {
7      header("Location: login.php");
8      exit();
9  }
10
11  ?>
```

Si no ha iniciado sesión → lo regresa al login.

SINTAXIS

```
2  <?php
3
4  session_start();
5  session_destroy();
6  header("Location: login.php");
7
8  ?>
```

cerrar sesión → lo regresa al login.

Los datos no se guardan en el navegador.

Se guardan en el servidor.

PROYECTO DE LOGIN

sistema

conexion.php

registro.php

guardar_usuario.php

login.php

validar_login.php

panel.php

cerrar.php

CREAR LA BASE DE DATOS

- Usuario **admin** con privilegios de root
- Crear la base de datos **login**

Para ver los usuarios de la base de datos

1.- `mysql -u root -p`

2.- `SELECT user, host FROM mysql.user;`

CREAR LA BASE DE DATOS

```
CREATE DATABASE registro;
USE registro;

CREATE TABLE usuarios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100),
  correo VARCHAR(100) UNIQUE,
  password VARCHAR(255),
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE DATABASE registro;
USE registro;
```

```
CREATE TABLE usuarios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100),
  correo VARCHAR(100) UNIQUE,
  password VARCHAR(255),
  fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

CONEXIÓN DE BASE DE DATOS (CONEXION.PHP)

```
<?php
$servidor = "localhost";
$usuario = "admin";
$password = "123456";
$bd      = "registro";

$conexion = new mysqli($servidor, $usuario, $password, $bd);

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

FORMULARIO DE REGISTRO (REGISTRO.PHP)

```
1  <html>
2  <head>
3  |   <title>Registro de usuarios</title>
4  </head>
5  <body>
6
7  <h2>Registro de Usuario</h2>
8
9  <form action="guardar_usuario.php" method="POST">
10
11  Nombre:<br>
12  <input type="text" name="nombre" required><br><br>
13
14  Correo:<br>
15  <input type="email" name="correo" required><br><br>
16
17  Contraseña:<br>
18  <input type="password" name="password" required><br><br>
19
20  <input type="submit" value="Registrarse">
21
22  </form>
23
24  <br>
25  <a href="login.php">¿Ya tienes cuenta? Inicia sesión</a>
26
27  </body>
28  </html>
```

GUARDAR_USUARIO.PHP

```
1 <?php
2 require_once "conexion.php";
3
4 /* RECIBIR DATOS */
5 $nombre = $_POST['nombre'];
6 $correo = $_POST['correo'];
7 $password = $_POST['password'];
8
9 /* CIFRAR CONTRASEÑA */
10 $password_seguro = password_hash($password, PASSWORD_DEFAULT);
11
12 /* VERIFICAR SI EL CORREO YA EXISTE */
13 $stmt = $conexion->prepare("SELECT id FROM usuarios WHERE correo = ?");
14 $stmt->bind_param("s", $correo);
15 $stmt->execute();
16 $stmt->store_result();
17
18 if ($stmt->num_rows > 0) {
19     echo "El correo ya está registrado. <a href='registro.php'>Volver</a>";
20     exit();
21 }
22 $stmt->close();
23
24 /* INSERTAR USUARIO */
25 $stmt = $conexion->prepare("INSERT INTO usuarios (nombre, correo, password) VALUES (?, ?, ?)");
26 $stmt->bind_param("sss", $nombre, $correo, $password_seguro);
27
28 if ($stmt->execute()) {
29     echo "Usuario registrado correctamente.<br>";
30     echo "<a href='login.php'>Iniciar sesión</a>";
31 } else {
32     echo "Error: " . $stmt->error;
33 }
34
35 $stmt->close();
36 $conexion->close();
37 ?>
```