Guía Completa del Sistema de Autenticación

Este documento explica en detalle la implementación del sistema de autenticación en PHP usando PDO, incluyendo el login, registro de usuario, manejo de sesiones y la conexión a la base de datos.

1. Configuración de la Base de Datos

Para almacenar los usuarios y sus credenciales, se creó la tabla 'usuarios' con las siguientes columnas:

- `id`: Identificador único (auto_increment)
- `nombre`: Nombre del usuario
- `email`: Correo electrónico único
- `password_hash`: Contraseña encriptada
- `fecha_registro`: Timestamp de registro

```
CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. Configuración de la Conexión a la Base de Datos

Se creó el archivo `database.php` para manejar la conexión con MySQL usando PDO. Se recomienda siempre usar excepciones para detectar errores de conexión.

```
<?php
$host = "127.0.0.1";
$dbname = "mood-tracker";
$username = "root";
$password = "";

try {
        $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8mb4", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Error en la conexión: " . $e->getMessage());
}
```

```
?>
```

3. Registro de Usuarios

El formulario de registro (`registration.php`) envía los datos al `authController.php`, que almacena los usuarios en la base de datos tras encriptar la contraseña con `password_hash()`.

```
<form action="../controladores/authController.php" method="POST">
    <input type="hidden" name="registro" value="1">
    <label>Nombre de Usuario</label>
    <input type="text" name="usuario" required>
    <label>Email</label>
    <input type="email" name="email" required>
    <label>Contraseña</label>
    <input type="password" name="contrasena" required>
    <button type="submit">Registrarse</button>
</form>
```

4. Lógica del Controlador (`authController.php`)

Cuando el usuario envía el formulario de registro, el `authController.php` procesa los datos y almacena el usuario en la base de datos.

```
<?php
require_once '../config/database.php';

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['registro'])) {
    $usuario = $_POST["usuario"];
    $email = $_POST["email"];
    $contrasena = $_POST["contrasena"];

    $passwordhash = password_hash($contrasena, PASSWORD_DEFAULT);

    try {
          $sql = "INSERT INTO usuarios (nombre, email, password_hash) VALUES (:usuario,
:email, :password)";
        $stm = $pdo->prepare($sql);
        $stm->bindParam(":usuario", $usuario);
        $stm->bindParam(":email", $email);
        $stm->bindParam(":password", $passwordhash);
        $stm->execute();

        header("Location: ../vistas/login.php?success=1");
        exit();
    } catch (PDOException $e) {
        echo "Error al registrar: " . $e->getMessage();
```

```
    }
}
?>
```

5. Inicio de Sesión (`login.php`)

Si el usuario se registró correctamente, el `login.php` muestra un mensaje de éxito y permite que el usuario inicie sesión.

```
<?php if (isset($_GET['success']) && $_GET['success'] == 1): ?>
    <div class="alert alert-success text-center mt-3">
        ¡Usuario registrado con éxito! Ahora puedes iniciar sesión.
    </div>
<?php endif; ?>
```

6. Autenticación de Usuario

Cuando el usuario intenta iniciar sesión, se verifica su existencia en la base de datos y se compara la contraseña ingresada con la almacenada usando `password_verify()`.

```
if ($usuarioEncontrado && password_verify($contrasena, $usuarioEncontrado["password_hash"])) {
    $_SESSION["usuario_id"] = $usuarioEncontrado["id"];
    $_SESSION["usuario"] = $usuarioEncontrado["nombre"];
    header("Location: ../vistas/dashboard.php");
} else {
    header("Location: ../vistas/login.php?error=1");
}
```

Desarrollado por: Tu Nombre