DESAFÍO PRÁCTICO 3

UNIVERSIDAD DON BOSCO

Presentado por:

Wendy Marcela Aguilar Vázquez AV220801

Melissa Abigail Flores Alfaro FA220709

Escuela de Computación

Lenguajes Interpretados en el Servidor (LIS104)

Grupo de laboratorio 04L

Docente: Karen Medrano

Enlace a Repositorio GitHub:

https://github.com/MelissaFloresA/Desafio3

ESTRUCTURA DE DESAFÍO

Base de datos: nombrada como “datos”, es una base de datos sencilla que solo consta de una tabla usuarios con las siguientes columnas, donde la llave primaria es auto incrementable desde 1.

#	Nombre	Tipo
1	**idusuarios** 🔑	int(11)
2	**nombres**	varchar(200)
3	**apellidos**	varchar(200)
4	**correo**	varchar(200)
5	**contraseña**	varchar(50)
6	**fecha_nacimiento**	date

Estructura de proyecto en Visual Studio

Esta estructura hace un poco de referencia a MVC, siendo el modelo usuario_model.php, el controlador usuario_ajax.php y la vista registro.php.

Explicación de archivos:

Conexión.php

```php
<?php
$servidor = "localhost";
$usuario = "root";
$password = "";
$base_datos = "datos";

$conn = new mysqli(hostname: $servidor, username: $usuario, password: $password, database: $base_datos);

if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}

$conn->set_charset(charset: "utf8");
?>
```

Es la conexión a la base de datos, es modular ya que se manda a llamar en todas las operaciones que tenga que ver con la base de datos.

Registro.php

Es la vista principal del sistema, la interfaz de usuario. Las validaciones del lado del cliente se especifican en validaciones.js y las del servidor en usuario_ajax.

Se hace uso de invalid-feedback, es parte de Bootstrap y se utiliza para mostrar mensajes de error personalizados cuando un control de formulario no es válido.

Validaciones.js

En ese archivo es donde se realizan las validaciones del lado del cliente con JavaScript, Cabe destacar que se implemente en tiempo real y se activa cuando el usuario sale del campo por medio del evento blur (inicia cuando pierde foco)

Usuario Model.php

Se encarga de la interacción con la base de datos implementando un CRUD completo, es una conexión segura ya que las consultas son preparadas.

Obtener Usuario () es una función para la visualización dinámica de los registros

UsuarioAjax.php

Es el controlador que maneja operaciones Ajax, manda parámetros necesarios a las funciones en modelo. Manejo de acciones POST (CRUD) y Manejo de GET para listar usuarios, además están las validaciones del lado del servidor.

Registrousuarios.js

Tiene la función cargarusuario() que llena la tabla con una lista de usuarios en JSON

Vista