



MEMORIA PROYECTO FINAL DE CICLO

Alejandro Morán Lorenzo
2º DAW

Tutor colectivo: Luis Javier Menéndez Tascon

Tutor Individual: Gonzalo Fernández Naveira

Contenido

Introducción	3
Presentación y objetivos	3
Contexto.....	3
Planteamiento del problema.....	3
Funcionalidad y Usabilidad.....	4
Recursos	4
Especificaciones de los requisitos	5
Introducción.....	5
Descripción General	5
Requisitos Específicos	5
Requerimientos Funcionales.....	5
Función Registro:.....	5
Función Iniciar Sesión:.....	6
Función Cerrar Sesión:	6
Función Puntuar	6
Requerimientos de Interfaces Externas	7
Obligaciones del diseño.....	8
Análisis	9
Introducción.....	9
Diagramas de Casos de Uso.....	9
Diseño	10
Introducción.....	10
Capa de Presentación	10
Capa de Negocio	10
Capa de Datos	10
Implementación	11
Tecnologías utilizadas.....	11
• Front-end:.....	11
• Back-end:.....	11

Descripción del proyecto.....	11
Capa de Presentación	11
Capa de Negocio	14
Capa de Datos	16
PHPMYAdmin	17
Conclusión	19
Evaluación personal	19
Posibles Ampliaciones.....	19
Bibliografía	19

Introducción

Presentación y objetivos

El objetivo de esta aplicación es dotar de un criterio objetivo a aquellas canciones que nos han acompañado a lo largo de nuestra vida. Para ello he creado una aplicación en la que los usuarios puedan ver un listado de canciones y en función de su criterio darles una puntuación como si de un examen se tratase.

Los usuarios que solo hayan entrado en la web podrán ver su contenido, pero será necesario que inicien sesión para poder puntuar las canciones.

Contexto

Más de una vez nos ha pasado que estamos tranquilamente con nuestros amigos tomando algo tranquilamente o en una conversación con nuestros padres y surge el tema de hablar de la nueva canción que ha sacado el último artista de moda, en ese momento cada uno tiene su propia opinión acerca del tema y cada cuál juzga en función de sus gustos, sin embargo, siempre queda la incertidumbre de ver quién finalmente tiene razón y saber si realmente se considera una buena canción o se trata de, como los más melómanos suelen decir, “música comercial”. En respuesta a ello se ha creado esta aplicación en la que un usuario podrá dar su opinión real acerca de una canción y compararla con la de los demás.

Planteamiento del problema

La funcional de la aplicación es sencilla, el usuario debe iniciar sesión para poder puntuar las canciones, pero primero debe registrarse en la aplicación. Para ello se han planteado dos tipos de usuarios (un rol de usuario a ha sido descartado por problemas técnicos):

- Usuario invitado: Este usuario se trataría del usuario más común, aquel que accedería a la url de la página y le echaría un vistazo, pero no tendría acceso a ninguna funcionalidad.

- Usuario registrado: Este usuario sería el que se ha registrado correctamente en la web y después iniciado sesión, lo que le permitiría acceder a sus funcionalidades.
- Usuario administrador (Rol descartado): Este usuario se encargaría de la gestión de la base de datos, pero finalmente no se ha implementado por problemas técnicos.

Funcionalidad y Usabilidad

Esta aplicación está pensada para llegar a todo tipo de público que esté interesado en la música y pueda acceder desde cualquier dispositivo o sistema operativo.

Recursos

- PHP
- Javascript
- CSS3
- HTML5
- MYSQL
- PHPMyAdmin

Especificaciones de los requisitos

Introducción

En este apartado resolveremos todo aquello relacionado con los requisitos mínimos de la aplicación.

Descripción General

El usuario debe de ver el contenido de la página en todo momento para saber a qué está accediendo además de tener la posibilidad de crear su propia cuenta para después optar a las funcionalidades de la web, además podrá modificar en todo momento las puntuaciones que haya dado.

Requisitos Específicos

- **Inicio:** Se trata de la página principal donde se almacena todo el contenido.
- **Registrarse:** Se trata de la página que permite registrar al usuario.
- **Login:** Se trata de la página que permite al usuario ya registrado iniciar sesión.
- **Cerrar Sesión:** Se trata de la opción que permite al usuario cerrar su sesión.

Requerimientos Funcionales

Los requerimientos funcionales son distintos para cada tipo de usuario. Existen las funciones globales que son las de registrarse e iniciar sesión. Sin embargo, el usuario registrado contará con la funcionalidad de puntuar las canciones y cerrar sesión.

Función Registro:

1. El primer paso que debe realizar el usuario es acceder a la página de login/registro y ahí se le facilitaran ambas opciones.

2. Tras acceder a la página deberá escribir el nombre de usuario y contraseña que desee registrar y enviar los datos por el botón facilitado.
3. Tras enviar los datos, se comprueba la disponibilidad del usuario, o sea, se comprueba que el usuario introducido no exista ya en la base de datos, si el usuario ya existe se le volverá a pedir otro nombre de usuario.
4. Si el proceso ha sido exitoso y el nombre de usuario introducido no se repite en la base de datos, este se agregará a ella y se redirigirá a la página de login/registro para poder iniciar sesión.

Función Iniciar Sesión:

1. Si el usuario ya ha creado su perfil en la base de datos, este deberá iniciar sesión en la aplicación.
2. Para ello deberá rellenar el formulario de inicio de sesión, en él introducirá el nombre de usuario que haya elegido, así como la contraseña.
3. Después se comprobará que el usuario exista en la base de datos y de ser así que la contraseña sea correcta, si ambos campos son correctos tendrá acceso a la aplicación.

Función Cerrar Sesión:

1. Para ello el usuario debe haber iniciado sesión previamente.
2. Después de acceder a las funcionalidades de la web, podrá cerrar su sesión una vez finalizado su uso. Para ello se le facilitará la opción de cerrar sesión que hará un “log-out” en su cuenta y se redirigirá a la página de inicio para poder volver a iniciar sesión cuando desee.

Función Puntuar

1. Para acceder a esta función el usuario debe de haber iniciado sesión previamente.

2. Tras iniciarse sesión, podrá acceder a la función que permita dar puntuaciones a las canciones disponibles clicando en el botón correspondiente, después le aparecerá un desplegable donde deberá introducir la puntuación que desee.
3. Tras introducir la puntuación, comprobará que se encuentra un número entre valores 0 y 10, si no se cumple, se volverá a solicitar la puntuación y si sí se cumple, se mostrará en la página.

Requerimientos de Interfaces Externas

Interfaces de usuarios

Las interfaces de usuario deben ser claras y no muy complejas para que el usuario pueda familiarizarse rápidamente con la web.

Interfaces Hardware

La web debe de ser capaz de visualizarse en todos los dispositivos que tengan acceso a internet y navegador web.

Interfaces Software

La web debe poder visualizarse en cualquier sistema operativo.

Interfaces de Comunicaciones

Si la web fuese lanzada realmente, contaría con un protocolo HTTPS para garantizar su seguridad y con un servicio FTP para confirmar los registros de usuario y desarrollar un sistema de publicidad.

Requerimientos de Rendimiento

La aplicación deberá poder utilizarse al 100% de su rendimiento en cualquier dispositivo, sin importar su software o hardware, siendo los únicos posibles impedimentos la velocidad de conexión tanto del cliente como del servidor.

Obligaciones del diseño

Estándares Cumplidos

La aplicación cuenta con un cierto nivel de seguridad puesto que las funcionalidades de los usuarios registrados solo son accesibles mediante el sistema de inicio de sesión y no desde rutas externas.

Limitaciones de Hardware

Al tratarse de una aplicación de hardware no debería de haber limitaciones a nivel de hardware, a excepción de algún tipo de fallo en el servidor.

Portabilidad

La aplicación ha sido diseñada con tecnología libre, por lo que podrá ser soportada por cualquier plataforma, sistema operativo y navegador.

Otros Requerimientos

La web utiliza una base de datos de MySQL, donde se almacena toda la información respecto a la página.

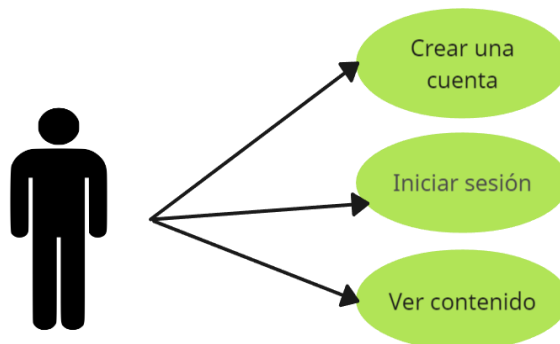
Análisis

Introducción

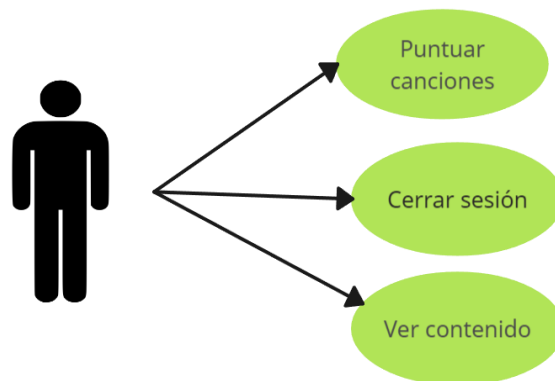
En este apartado vamos a analizar la web desde su utilidad para el usuario y sus posibles casos de uso.

Diagramas de Casos de Uso

El diagrama de casos de uso es útil para ver las posibilidades que tiene el usuario dentro de nuestra aplicación en función de su rol.



Usuario Invitado



Usuario Registrado

Diseño

Introducción

El Modelo Vista Controlador (MVC) es un patrón de diseño arquitectónico ampliamente utilizado en el desarrollo de software, especialmente en aplicaciones web y de escritorio. Este modelo organiza una aplicación en tres componentes principales, cada uno con responsabilidades bien definidas: Modelo, Vista y Controlador. Su objetivo principal es separar la lógica de negocio, la presentación y el flujo de control, promoviendo la modularidad y la reutilización del código. A continuación, se detalla el funcionamiento de cada una de estas capas.

Capa de Modelo

El Modelo representa la lógica de negocio y la estructura de los datos de la aplicación. Es responsable de gestionar cómo se almacenan, procesan y recuperan los datos. En otras palabras, es la parte que interactúa con la base de datos u otros servicios de persistencia, asegurando que las reglas de negocio se cumplan.

Capa de Vista

La Vista es responsable de la presentación de la información al usuario. Se enfoca en cómo los datos se muestran, pero no realiza modificaciones directas en ellos ni contiene lógica de negocio. Generalmente, las vistas consisten en interfaces gráficas de usuario (GUI) o páginas web diseñadas para ser atractivas e intuitivas.

Capa de Controlador

El Controlador actúa como intermediario entre la Vista y el Modelo. Su función principal es procesar las interacciones del usuario y decidir cómo deben responder las otras dos capas.

Implementación

Tecnologías utilizadas

Las principales tecnologías utilizadas en este proyecto han sido:

- **Front-end:**
 - HTML5
 - CSS3
 - JavaScript
- **Back-end:**
 - PHP
 - MySQL
 - PHPMyAdmin

También para poder realizar la aplicación en un entorno web se ha utilizado el servidor UniServerZ.

Descripción del proyecto

Capa de Vista

HTML5

HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el estándar fundamental para la creación y estructuración de páginas web. Utiliza etiquetas para definir y organizar el contenido, como textos, imágenes y enlaces. Este lenguaje posibilita la creación de sitios web interactivos al proporcionar la base estructural para la presentación y el funcionamiento de las páginas en la World Wide Web. HTML, esencial para el desarrollo web, define la estructura básica de una página, especificando encabezados, párrafos, enlaces y medios. Su versatilidad permite la integración de elementos multimedia y formularios interactivos. Las etiquetas clave, como <head>, <body> y <div>, facilitan la organización y presentación de contenido. Al ser interpretado por los navegadores, HTML forma la base de la experiencia de usuario en la mayoría de los sitios web. Su combinación con CSS y JavaScript amplía las posibilidades de diseño y funcionalidad. HTML5 representa una evolución significativa respecto a versiones anteriores de HTML. Algunas mejoras clave incluyen nuevos elementos semánticos, compatibilidad multimedia integrada, APIs de navegador avanzadas, compatibilidad con dispositivos móviles...

CSS3

CSS3, Cascading Style Sheets 3, complementa HTML al definir la presentación y el estilo de una página web. Con selectores y propiedades, CSS3 permite controlar el diseño, colores, fuentes y espaciado. Funcionalidades avanzadas, como animaciones y transformaciones, mejoran la experiencia del usuario. El modularidad de CSS3 facilita la creación de diseños responsivos para adaptarse a diferentes dispositivos. Integrado con HTML, optimiza la separación de contenido y diseño, simplificando el mantenimiento y la adaptabilidad de los sitios web modernos. CSS3 presenta algunas mejoras respecto a su base CSS como pueden ser el diseño responsivo, sombras y bordes avanzados, animaciones y transiciones, integración de flexbox y gridlayout.

JavaScript

JavaScript, un lenguaje de programación esencial para el desarrollo web, ha transformado la experiencia en línea al permitir la creación de sitios y aplicaciones interactivas y dinámicas. Desarrollado inicialmente como un complemento para mejorar la interactividad en el navegador, JavaScript se ha convertido en un componente fundamental de la programación web moderna.

En el corazón de su poder radica la capacidad de manipular el Document Object Model (DOM), permitiendo a los desarrolladores interactuar y modificar elementos HTML y CSS en tiempo real. Esto, combinado con su capacidad para realizar solicitudes asíncronas al servidor a través de AJAX y la Fetch API, ha llevado a la creación de interfaces de usuario fluidas y experiencias de usuario más ricas.

La evolución de JavaScript no se limita al cliente. Con la aparición de Node.js, JavaScript puede ejecutarse también en el servidor, lo que ha dado lugar a un enfoque más unificado en el desarrollo full-stack. Esta versatilidad ha llevado al surgimiento de numerosos frameworks y bibliotecas, como React, Angular y Vue.js, que simplifican el desarrollo de aplicaciones complejas.

JavaScript ha abrazado estándares modernos como ECMAScript 6 (ES6), introduciendo características como promesas, destructuring, y arrow functions, que mejoran la legibilidad y eficiencia del código. Además, la adopción de WebAssembly ha ampliado sus límites, permitiendo integrar código de alto rendimiento desarrollado en otros lenguajes.

```

public function cabecera()
{
    $resultado = "";
    $resultado .= "
    <!DOCTYPE html>
    <html lang='es'>

    <head>
        <meta charset='UTF-8'>
        <meta name='viewport' content='width=device-width, initial-scale=1.0'>

        <link rel='stylesheet' href='/css/general/general.css'>
        <link rel='stylesheet' href='/css/general/constantes.css'>
        <link rel='stylesheet' href='/css/general/sesion.css'>
        <link rel='stylesheet' href='/css/general/animaciones.css'>
        <link rel='stylesheet' href='/css/general/scrollbar.css'>
        <link rel='stylesheet' href='/css/general/nav.css'>
        <link rel='stylesheet' href='/css/general/vinilo.css'>
        <link rel='stylesheet' href='/css/general/fuentes.css'>
        <link rel='stylesheet' href='/css/general/footer.css'>
        <link rel='icon' type='image/x-icon' href='/img/logo.ico'>";

    //cargar archivos css añadidos con la funcion conRutasCSS
    foreach ($this->rutasCSS as $rutaCSS) {
        $resultado .= "\n<link rel='stylesheet' href='$rutaCSS'>";
    }
    $resultado .= "
    <title>$this->titulo</title>
    </head>

    <body>";
    return $resultado;
}

```

**Ejemplo de la Capa de Vista de la cabecera*

Capa de Controlador

PHP

PHP, acrónimo recursivo de "PHP: Hypertext Preprocessor", es un lenguaje de programación de servidor ampliamente utilizado para el desarrollo web dinámico. Su propósito principal es la generación de contenido web dinámico, permitiendo la creación de aplicaciones web interactivas y robustas. A lo largo de los años, PHP ha evolucionado y ha mantenido su relevancia en el mundo del desarrollo web. Veamos algunas de sus características y contribuciones más destacadas:

- **Interacción con Bases de Datos:** PHP simplifica la interacción con bases de datos, permitiendo la conexión y manipulación de datos de manera eficiente. Integrado con sistemas de gestión de bases de datos como MySQL, facilita la creación de aplicaciones que almacenan y recuperan información de manera efectiva.
- **Facilidad de Integración con HTML:** Una de las fortalezas de PHP es su integración fluida con HTML. Los bloques de código PHP pueden incrustarse directamente en documentos HTML, permitiendo la mezcla de lógica de programación con la presentación web.
- **Amplia Comunidad y Recursos:** PHP cuenta con una gran comunidad de desarrolladores y una amplia cantidad de recursos, bibliotecas y frameworks. PHP tiene frameworks como Laravel y Symfony, que simplifican el desarrollo de aplicaciones web robustas y escalables.
- **Manejo de Formularios y Sesiones:** PHP facilita la gestión de formularios web, permitiendo la recopilación de datos del usuario de manera sencilla. Además, ofrece un sistema de sesiones que permite mantener la información del usuario entre diferentes páginas, mejorando la experiencia de navegación.
- **Flexibilidad y Portabilidad:** PHP es compatible con varios sistemas operativos y servidores web, lo que proporciona flexibilidad en la elección de la plataforma de alojamiento. Su código fuente abierto y su disponibilidad gratuita han contribuido a su amplia adopción.

- Desarrollo de Aplicaciones Complejas: Aunque inicialmente se centró en la creación de páginas web dinámicas, PHP ha evolucionado para admitir el desarrollo de aplicaciones empresariales complejas. Su capacidad para integrarse con otros servicios y tecnologías hace que sea una elección sólida para proyectos diversos.

```
public static function registrarse()
{
    if (isset($_POST['usuarioRegistro']) && isset($_POST['correoRegistro']) && isset($_POST['contrasenaRegistro'])) {
        $usuario = trim(htmlspecialchars($_POST['usuarioRegistro']));
        $correo = trim(htmlspecialchars($_POST['correoRegistro']));
        $contrasena = trim(htmlspecialchars($_POST['contrasenaRegistro']));

        //comprobar si ya existe un usuario igual. En caso contrario continuar
        $result = ModeloUsuarios::comprobarPorUsuario($GLOBALS['conn'], $usuario);

        if ($result && mysqli_num_rows($result) > 0) {
            //si ya existe, error

            vista('registro', [
                'error' => '400 - Conflict: El usuario ya existe, inténtelo de nuevo con otro nombre de usuario'
            ])->conRutasCSS([
                '/css/login.css',
                '/css/general/general.css',
                '/css/general/constantes.css',
                '/css/responsive/responsive.css'
            ])->imprimirFooter();
        } else {
            //si aun no existe, crear
            $result = ModeloUsuarios::crear($GLOBALS['conn'], $usuario, $contrasena, $correo);

            //si no hay fallos, iniciar sesion con el nuevo usuario
            if ($result) {
                //registrado con éxito, iniciar sesion con ese usuario
                ControladorLogin::iniciarSesion(['usuarioLogin' => $usuario, 'contrasenaLogin' => $contrasena]);
            } else {
                //error al crear el usuario
                vista('registro', [
                    'error' => '400 - Internal Server Error: Hubo un error al crear el usuario en la base de datos.' . mysqli_error($GLOBALS['conn'])
                ])->conRutasCSS([
                    '/css/login.css',
                    '/css/general/general.css',
                    '/css/general/constantes.css',
                    '/css/responsive/responsive.css'
                ])->imprimirFooter();
            }
        }
    } else {
        vista('registro', [
            'error' => '401 - Bad Request: El formulario no contenia todos los datos necesarios'
        ])->conRutasCSS([
            '/css/login.css',
            '/css/general/general.css',
            '/css/general/constantes.css',
            '/css/responsive/responsive.css'
        ])->imprimirFooter();
    }
}
```

**Ejemplo de la Capa Controlador para la función de registro.*

Capa de Modelo

MySQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS, por sus siglas en inglés) que desempeña un papel crucial en el almacenamiento, organización y recuperación de datos en aplicaciones web y empresariales. Esta popular base de datos de código abierto ha evolucionado para convertirse en un componente esencial en el desarrollo de sistemas de información. Aquí se destacan algunas de las características y contribuciones clave de MySQL:

- **Eficiencia y Rendimiento:** MySQL está diseñado para ofrecer un rendimiento eficiente, lo que lo convierte en una opción preferida para aplicaciones con requisitos de acceso y procesamiento intensivo de datos. Su capacidad para manejar grandes conjuntos de datos y ejecutar consultas complejas lo hace adecuado para diversas aplicaciones.
- **Escalabilidad:** MySQL es altamente escalable, lo que significa que puede adaptarse y crecer para manejar un mayor volumen de datos y tráfico sin perder eficiencia. Esta característica es crucial para aplicaciones que experimentan un crecimiento significativo.
- **Compatibilidad con Estándares:** MySQL sigue los estándares de SQL (Structured Query Language), lo que facilita la migración y la interoperabilidad con otros sistemas de bases de datos relacionales. Esto también facilita la integración con diferentes lenguajes de programación y entornos de desarrollo.
- **Seguridad:** MySQL proporciona funciones sólidas de seguridad, como la gestión de usuarios y privilegios, la encriptación de datos y la capacidad de auditar actividades. Estas características garantizan la protección de la información almacenada y el cumplimiento de las normativas de seguridad.
- **Replicación y Alta Disponibilidad:** MySQL ofrece opciones avanzadas de replicación que permiten la creación de copias idénticas de una base de datos en múltiples servidores. Esto mejora la disponibilidad y la tolerancia a fallos, asegurando que los datos estén siempre accesibles incluso en caso de problemas en el servidor principal.

- Comunidad Activa y Soporte: Al ser una tecnología de código abierto, MySQL cuenta con una comunidad activa de desarrolladores y usuarios que contribuyen al mantenimiento y mejora continua de la base de datos. Además, hay servicios de soporte disponibles para empresas que buscan asistencia profesional.
- Almacenamiento de Datos Geoespaciales: MySQL incluye funciones para el manejo de datos geoespaciales, lo que lo hace adecuado para aplicaciones que requieren información relacionada con la ubicación, como mapas interactivos y análisis geográficos.

PHPMyAdmin

PhpMyAdmin es una herramienta de administración de bases de datos que facilita la gestión y manipulación de bases de datos MySQL a través de una interfaz web. Su utilidad radica en proporcionar a los desarrolladores y administradores de sistemas una forma intuitiva de interactuar con las bases de datos, sin tener que depender de comandos SQL directos. Aquí se destacan algunas características clave de phpMyAdmin:

- Interfaz Gráfica de Usuario (GUI): phpMyAdmin ofrece una interfaz basada en web que simplifica la administración de bases de datos. Los usuarios pueden realizar tareas como crear, modificar y eliminar bases de datos y tablas, así como ejecutar consultas SQL, todo ello a través de una interfaz gráfica fácil de usar.
- Gestión de Usuarios y Privilegios: Permite gestionar usuarios y asignar privilegios a nivel de base de datos o tabla. Esto es esencial para controlar quién tiene acceso a qué partes de la base de datos y con qué permisos.
- Importación y Exportación de Datos: Facilita la importación y exportación de datos a través de diversos formatos, como SQL, CSV y Excel. Esto es útil para realizar copias de seguridad, transferir datos entre bases de datos y realizar migraciones.

- **Diseño Visual de Tablas:** Proporciona una vista visual de las tablas de la base de datos, lo que facilita la comprensión de la estructura de la base de datos. Los usuarios pueden agregar, modificar y eliminar columnas de manera visual.
- **Ejecución de Consultas SQL:** Ofrece un área dedicada para ejecutar consultas SQL directas, lo que permite a los usuarios más avanzados realizar operaciones específicas sin salir de la interfaz de phpMyAdmin.
- **Historial de Consultas:** Registra un historial de las consultas SQL ejecutadas, lo que facilita el seguimiento de las acciones realizadas en la base de datos y proporciona una capa adicional de control y seguridad.
- **Soporte Multilingüe:** phpMyAdmin está disponible en varios idiomas, lo que facilita su uso por parte de una audiencia global y diversa.
- **Plataforma de Código Abierto:** Al ser un proyecto de código abierto, phpMyAdmin es gratuito y cuenta con una comunidad activa de desarrolladores que contribuyen a su mejora continua.

```

public static function comprobarPorUsuario($conn, $usuario)
{
    //consulta
    $sql = "SELECT * FROM usuarios WHERE nombreUsuario='$usuario'";
    $result = mysqli_query($conn, $sql);

    //guardar error en sesion
    if (!$result) {
        $_SESSION['errores'][] = "Hubo un error al seleccionar un usuario por nombre de usuario: " . $conn->error;
    }

    //devuelve estado. False implica error, !false implica que se realizo con exito
    return $result;
}

public static function seleccionarPorNombre($conn, $nombre)
{
    //consulta
    $sql = "SELECT * FROM usuarios WHERE nombreUsuario='$nombre'";
    $result = mysqli_query($conn, $sql);

    //guardar error en sesion
    if (!$result) {
        $_SESSION['errores'][] = "Hubo un error al seleccionar un usuario por nombre: " . $conn->error;
    }

    //devuelve estado. False implica error, !false implica que se realizo con exito
    return $result;
}

```

**Ejemplo de la capa de Modelo para la función comprobar usuario.*

Conclusión

Evaluación personal

A nivel personal este proyecto me ha ayudado a ver aquellas facetas de la programación que no llegaba a contemplar por mí mismo, ayudándome a aprender campos como la seguridad en línea, la optimización de documentos en línea, trabajar en un diseño multiplataforma, etc. También destacar que la diferencia de tiempo me ha ayudado a darle un contenido más dinámico y decorativo de lo que tenía planeado anteriormente. Por último añadir que no solo el proyecto me ha hecho mejorar sino que también me resulta una buena oportunidad de negocio bien sea a modo de exposición en plataformas como GitHub para grandes empresas.

Posibles Ampliaciones

- Mejora del diseño
- Mejorar la seguridad de los usuarios en la base de datos
- Reproductor de canciones en línea

Bibliografía

- <https://stackoverflow.com/>
- <https://www.php.net/manual/en/index.php>
- <https://www.w3schools.com/>
- *Apuntes del curso DAW (DESARROLLO APLICACIONES WEB) 22-23*