Manual tecnico y actividades 2 y 4

Omar Esteban Contreras Leal (ocontre8@estudiante.ibero.edu.co)

ID Estudiantil: 100144268

Ingeniería de software, Universidad Iberoamericana

Proyecto de software

Tatiana Cabrera

11 de Mayo 2025

Control de Acceso

```
include('./components/sessionControl.php');
if($_SESSION["rol"]!="administrador"){
    header("location: ./error.html");
}
include("./components/tokenControl.php");
```

Función: Restringe el acceso a esta página exclusivamente a usuarios con rol de administrador.

Propósito: Seguridad. Previene que usuarios sin privilegios accedan al panel.

Mostrar Información del Administrador

```
<img class="profile-pic-img-user" src="<?php echo $_SESSION['pic']; ?>">
<h2><?php echo $_SESSION["nombre"]?></h2>
<p>@<?php echo $_SESSION["username"]?></p>
<p><?php echo $_SESSION["rol"]?></p>
```

Función: Muestra la foto, nombre, nombre de usuario y rol del administrador.

Propósito: Personalización del panel con los datos del administrador logueado.

Lista de Usuarios

```
$sql = "SELECT * FROM userm WHERE chefid is null order by userid";
```

Función: Consulta y despliega en tabla todos los usuarios que no son chefs.

Propósito: Permitir al administrador ver, editar o eliminar usuarios comunes.

Listado de Chefs

$sql = "SELECT * FROM userm WHERE chefid is not null order by chefid";

Función: Consulta y despliega todos los usuarios con un chefid, es decir, los chefs registrados.

Propósito: Permitir ver, editar, eliminar o acceder a las recetas de los chefs.

Botones de Navegación Interna

<button class="seeDataBtn" id="seeUsers">Ver Usuarios</button>

<button class="seeDataBtn" id="seeChefs">Ver Chefs</button>

Función: Activan la visualización de las respectivas tablas.

Propósito: Navegación dinámica entre secciones del panel.

Acciones en la Tabla

En ambas tablas se incluye:

<a href="adminMakefs.php? delete=$userdata[userid]">

<a href="adminMakefs.php? editar=$userdata[userid]">

Función: Permite eliminar o editar usuarios mediante parámetros GET.

Propósito: Gestión directa de los datos de usuario desde la tabla.

Inclusión de Componentes

```
include('./components/header.php');
include('./components/menudesplegable.php');
include('./components/preloader.php');
include('./components/footer.php');
```

Función: Inserta partes reutilizables de la UI como encabezado, menú, preloader y pie de página.

Propósito: Modularidad y consistencia del diseño.

Archivos JS Vinculados

```
<script src="./js/adminMakefs.js"></script>
<script src="./js/axiosAdminMakefs.js"></script>
```

Función: Contienen la lógica JavaScript para manejar eventos del panel de administración (como alternar vistas, AJAX con Axios, etc.).

Propósito: Hacer el panel interactivo y funcional desde el frontend.

Claro, a continuación te presento el **manual técnico** de algunas de las funciones principales encontradas en el código PHP que compartiste, enfocándonos en las que realizan procesos clave como la recolección de datos, ordenamiento y cálculo de puntuaciones.

ordering($rate, $views)

Propósito:

Calcular una puntuación personalizada (fscore) para una receta, en base a su calificación promedio y número de visualizaciones.

Parámetros:

- float $rate: Calificación promedio de la receta.

- int $views: Número de visualizaciones de la receta.

Valor de retorno:

- int: Puntuación final resultante de combinar rate y views.

Funcionamiento:

1. Evalúa en qué rango cae el número de vistas usando el array $viewRanges.

2. Asigna un puntaje (vscore) basado en la posición del rango.

3. Calcula el puntaje de calificación (vrate) como un múltiplo de rate / 0.5 * 100.

4. Suma ambos puntajes para obtener el valor final (fscore).

2. usort($finalOrdering, function($item1, $item2) { ... })

Propósito:

Ordenar el arreglo $finalOrdering de recetas en orden descendente por su puntuación final fscore.

Parámetros del comparador:

- $item1, $item2: Elementos del array que contienen información de recetas.

Valor de retorno:

- Reorganiza $finalOrdering directamente.

Funcionamiento:

- Utiliza el operador <=> para comparar fscore de dos recetas y decidir su posición.

3. test_input

Propósito:

Sanitizar cadenas de texto para evitar inyecciones XSS o errores de formato al mostrarlas.

Parámetros:

- string $data: Cadena de entrada (como nombre de receta o chef).

Valor de retorno:

- string: Texto limpio y seguro para renderizar en HTML.

Query principal en $query

Propósito:

Extraer las 20 mejores recetas públicas que coincidan con los parámetros de región y tipo culinario.

Elementos clave:

- convert_from(decode(recipe.tags, 'base64'), 'UTF8') LIKE :sel: Filtro de etiquetas descodificadas.

- AVG(stars.star): Promedio de calificaciones.

- GROUP BY (...): Agrupación para combinar correctamente los JOIN.

- LIMIT 20: Limita los resultados a 20 recetas.

Parámetro dinámico:

array("sel"=>("%".$_GET["region"].ucfirst($_GET["type"])."%"))

- Une dinámicamente región y type para buscar coincidencias en las etiquetas.

Inicio de sesión (session_start): Verifica si hay errores previos de login o registro para eliminarlos.

Redirección (header("location: /")): Evita el acceso sin parámetros válidos region y type.

Componentes de UI: Se incluyen header, menú, preloader, footer, etc., desde archivos PHP separados.

Render dinámico en HTML: Las recetas se imprimen en tarjetas HTML dentro de un foreach.