

PEC 3: PHP



Desarrollo back-end con PHP

María Cañas Encinas

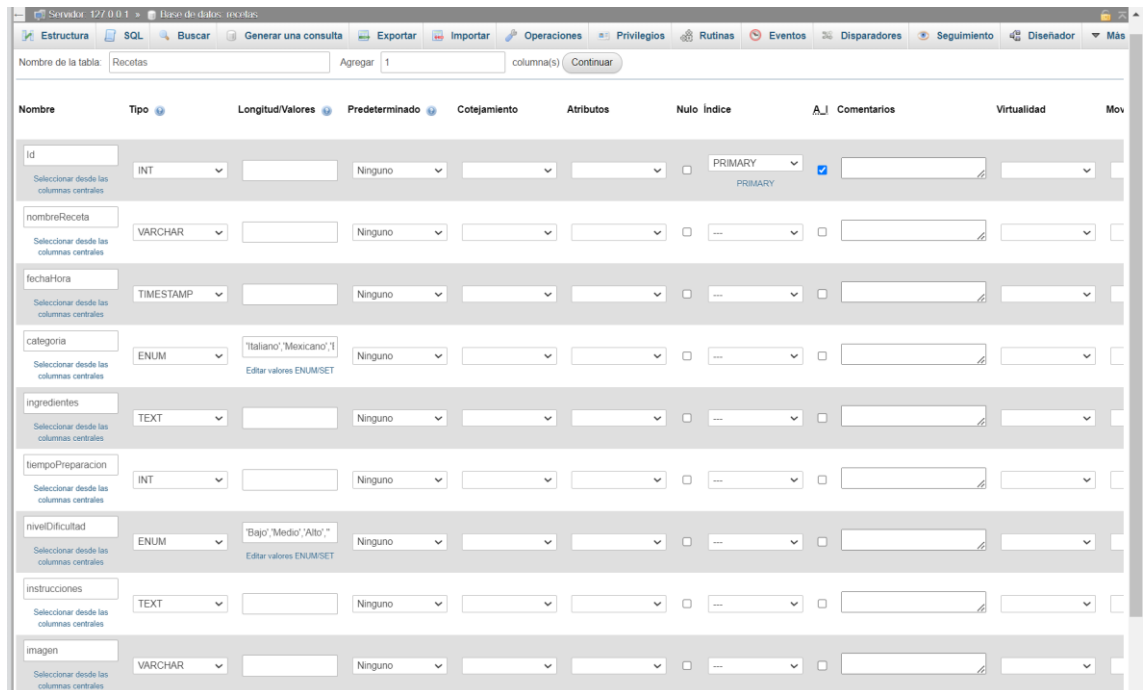
ÍNDICE

- 1. Creación de la Base de Datos**
- 2. Acceso a la Base de Datos**
- 3. Creación de la página de inicio**
- 4. Creación de las páginas para las recetas**
- 5. Ordenación y filtro de las recetas**
- 6. Menú**
- 7. API**
- 8. Login**
- 9. Alta y edición de usuarios**
- 10. Publicación en internet**

1. CREACIÓN DE LA BASE DE DATOS

El primer paso ha sido la creación de la base de datos en XAMPP. Para ello he creado una BD con el nombre de **Recetas** y dentro de esta base de datos he creado una tabla llamada Recetas la cual va a tener la información de las recetas según los requisitos que se nos han marcado en el enunciado de la actividad.

Para ello se crea una nueva Base de datos e introducimos los nombres y tipos de los campos de las columnas que formarán la base de datos. Como se puede ser a continuación:



The screenshot shows the phpMyAdmin interface for a database named 'Recetas'. The 'Estructura' (Structure) tab is active, displaying the table structure for the 'Recetas' table. The table has 10 columns: 'Id', 'nombreReceta', 'fechaHora', 'categoria', 'ingredientes', 'tiempoPreparacion', 'nivelDificultad', 'instrucciones', and 'imagen'. Each column has a dropdown menu to select its data type, a field for length/precision, a dropdown for 'Predeterminado' (Default), a dropdown for 'Cotejamiento' (Collation), a dropdown for 'Atributos' (Attributes), a checkbox for 'Nulo' (Null), a dropdown for 'Índice' (Index), a checkbox for 'Comentarios' (Comments), a dropdown for 'Virtualidad' (Virtuality), and a dropdown for 'Mov' (Move).

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice	Comentarios	Virtualidad	Mov
Id	INT		Ninguno			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>		
nombreReceta	VARCHAR		Ninguno			<input type="checkbox"/>	---			
fechaHora	TIMESTAMP		Ninguno			<input type="checkbox"/>	---			
categoria	ENUM	'Italiano','Mexicano','I'	Ninguno			<input type="checkbox"/>	---			
ingredientes	TEXT		Ninguno			<input type="checkbox"/>	---			
tiempoPreparacion	INT		Ninguno			<input type="checkbox"/>	---			
nivelDificultad	ENUM	'Bajo','Medio','Alto',''	Ninguno			<input type="checkbox"/>	---			
instrucciones	TEXT		Ninguno			<input type="checkbox"/>	---			
imagen	VARCHAR		Ninguno			<input type="checkbox"/>	---			

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	Id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	nombreReceta	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
3	fechaHora	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Cambiar Eliminar Más
4	categoria	enum('Italiano','Mexicano','Español','')	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
5	ingredientes	text	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
6	tiempoPreparacion	int(11)			No	Ninguna			Cambiar Eliminar Más
7	nivelDificultad	enum('Bajo','Medio','Alto','')	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
8	instrucciones	text	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
9	imagen	varchar(255)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Justificación de los tipos de datos elegidos

Id -> He utilizado el tipo de dato INT con la propiedad de auto_increment para que así cada receta tenga un Id único y este valor se incremente de forma automáticamente cada vez que se añade una nueva receta

nombreReceta -> He utilizado el tipo de dato varchar con una longitud máxima de 100 caracteres

fechaHora -> He utilizado el tipo de dato Timestamp como dice el enunciado de la actividad

categoria -> He utilizado el tipo de dato enum para limitar los valores posibles. Estos valores han sido: Italiano, Mexicano y Español como se puede ver en la siguiente imagen.

ingredientes -> He utilizado el tipo de dato Text para almacenar todos los ingredientes pertenecientes a esa receta

tiempoPreparacion -> He utilizado el tipo de dato INT donde se guardará el tiempo de preparación de la receta en minutos

Servidor: 127.0.0.1 » Base de datos: recetas

Estructura

SQL

Buscar

Generar una consulta

Nombre de la tabla:

Nombre	Tipo	Longitud/Valores
<div><input type="text" value="Id"/></div> <div>Seleccionar desde las columnas centrales</div>	<div>INT</div> <div></div>	<div></div>
<div><input type="text" value="nombreReceta"/></div> <div>Seleccionar desde las columnas centrales</div>	<div>VARCHAR</div> <div></div>	<div></div>
<div><input type="text" value="fechaHora"/></div> <div>Seleccionar desde las columnas centrales</div>	<div>TIMESTAMP</div> <div></div>	<div></div>
<div><div>categoria</div><div>Seleccionar desde las columnas centrales</div></div>	<div>ENUM</div> <div></div>	<div><div><div>Italiano, 'Mexicano', 'I'</div><div>Editar valores ENUM/SET</div></div><div><div>Ninguno</div><div></div><div></div><div></div></div></div>
<div><input type="text" value="ingredientes"/></div> <div>Seleccionar desde las columnas centrales</div>	<div>TEXT</div> <div></div>	<div><div>Ninguno</div><div></div><div></div><div></div></div>
<div><input type="text" value="tiempoPreparacion"/></div> <div>Seleccionar desde las columnas centrales</div>	<div>INT</div> <div></div>	<div><div>Ninguno</div><div></div><div></div><div></div></div>
<div><input type="text" value="nivelDificultad"/></div> <div>Seleccionar desde las columnas centrales</div>	<div>ENUM</div> <div></div>	<div><div>Ninguno</div><div></div><div></div><div></div></div>

Editor de ENUM/SET

Valores para la columna "nivelDificultad"

Introducir cada valor en un campo separado.

Bajo

Medio

Alto

Agregar 1 valor(es)

Continuar

Cerrar

imagen -> He utilizado el tipo de dato varchar con una longitud de 255 donde se almacenará la ruta de la imagen

He añadido 10 recetas en la tabla creada.

Mostrando filas 0 - 9 (total de 10, La consulta tardó 0.0003 segundos.)

SELECT * FROM "recetas"

☐ Perfilando [\[Editar en línea\]](#) [\[Editar\]](#) [\[Explicar SQL\]](#) [\[Crear código PHP\]](#) [\[Actualizar\]](#)

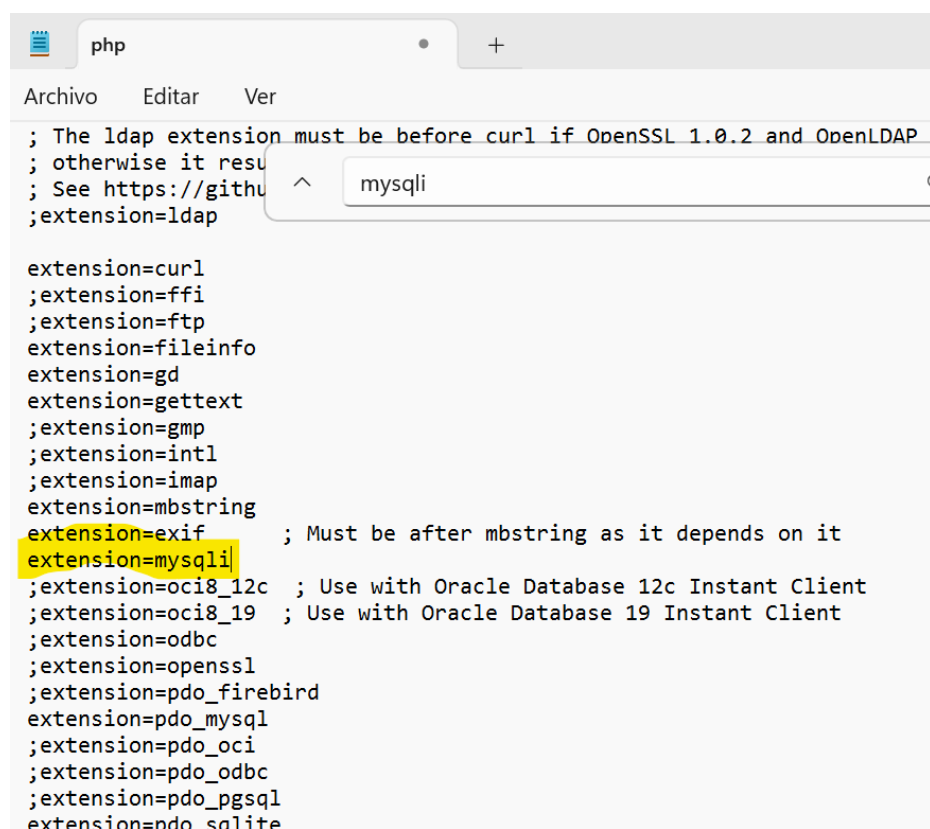
☐ Mostrar todo | Número de filas: 25 | Filtrar filas: | Ordenar según la clave: Ninguna

Opciones extra

				id	nombreReceta	fechaHora	categoria	ingredientes	tiempoPreparacion	nivelDificultad	instrucciones	imagen
<input type="checkbox"/>	Editar	Copiar	Borrar	1	Pollo al Horno con Patatas	2024-05-01 09:40:13	Español	Muslos de pollo, patatas, cebolla, aceite de oliva...	60	Bajo	Marinar el pollo con ajo, romero y aceite de oliva.	https://img.bonvivour.com/pollo-asado-al-horno-co
<input type="checkbox"/>	Editar	Copiar	Borrar	2	Tortilla de Patatas	2024-05-09 12:51:20	Español	Patatas, huevos, cebolla, aceite de oliva, sal	35	Medio	Freír las patatas y la cebolla, mezclar con los hu...	https://www.goya.com/media/3816/tortilla-espanola
<input type="checkbox"/>	Editar	Copiar	Borrar	3	Gazpacho Andalúz	2024-05-20 22:54:14	Español	Tomate, pepino, pimiento verde, cebolla, ajo, pan	10	Bajo	Triturar todos los ingredientes en una licuadora h...	https://cdn.recetasderechupete.com/wp-content/uplo...
<input type="checkbox"/>	Editar	Copiar	Borrar	4	Pasta con champiñones	2024-05-23 12:24:14	Italiano	Pasta, champiñones, crema de leche, mantequilla, a...	20	Alto	Cocinar la pasta al dente, saltear los champiñones.	https://recetasdecochina.elmundo.es/wp-content/uplo...
<input type="checkbox"/>	Editar	Copiar	Borrar	5	Ensalada de frutas	2024-05-11 13:05:54		Piña, fresas, kiwi, plátano, jugo de naranj...	5	Bajo	Cortar la fruta en trozos pequeños y mezclar con j...	https://www.bancodealimentoschicago.org/wp-conten...
<input type="checkbox"/>	Editar	Copiar	Borrar	6	Receta1	2024-05-11 13:12:19	Italiano	Ingredientes Receta1	10	Alto	Instrucciones Receta1	Imagen Receta1
<input type="checkbox"/>	Editar	Copiar	Borrar	7	Receta2	2024-05-29 13:09:35	Mexicano	Ingredientes Receta2	60	Alto	Instrucciones Receta2	Imagen Receta2
<input type="checkbox"/>	Editar	Copiar	Borrar	8	Receta3	2024-05-29 13:09:35	Mexicano	Ingredientes Receta3	50	Alto	Instrucciones Receta3	Imagen Receta3
<input type="checkbox"/>	Editar	Copiar	Borrar	9	Receta4	2024-05-11 13:12:48	Italiano	Ingredientes Receta4	25	Alto	Instrucciones Receta4	Imagen Receta4
<input type="checkbox"/>	Editar	Copiar	Borrar	10	Receta5	2024-05-11 13:13:05	Italiano	Ingredientes Receta5	35	Alto	Instrucciones Receta5	Imagen Receta5

2. ACCESO A LA BASE DE DATOS

Para crear el script en PHP, vamos a seguir unos pasos. Lo primero de todo es asegurarnos de que tenemos instalado la librería mysql, para ello vamos a acceder al fichero php.ini y vamos a buscar la línea extensión=mysql y nos aseguramos de que esté sin el punto y como al principio.








Una vez que nos aseguramos de que tenemos la extensión instalada, vamos a crear el script PHP. El script tendrá el nombre de activity_2.php y se ha creado con el editor de texto visual studio code.

Este será el script que se ha creado para obtener la información de una de las recetas. Y lo guardaremos en nuestro directorio raíz del servidor web (Yo lo he guardado dentro de la carpeta htdocs que tiene Xampp).

```
C: > xampp > htdocs > PHP activity_2.php > ...
1
2  <?php
3
4  //Conexion con la base de datos
5  $conexion = new mysqli("localhost","root","","Recetas");
6
7  //Consulta para obtener los datos de una Receta
8  | //Vamos a obtener los datos de la receta con id 1)
9  $datos = "SELECT * FROM recetas where id=1";
10 $resultado = $conexion->query($datos);
11
12 //Mostrar la información de la receta seleccionada
13 $receta = $resultado->fetch_assoc();
14
15 echo "<h1>" . $receta['nombreReceta'] . "</h1>";
16 echo "<p>" . $receta['fechaHora'] . "</p>";
17 echo "<p>" . $receta['categoria'] . "</p>";
18 echo "<p>" . $receta['ingredientes'] . "</p>";
19 echo "<p>" . $receta['tiempoPreparacion'] . "</p>";
20 echo "<p>" . $receta['nivelDificultad'] . "</p>";
21 echo "<p>" . $receta['instrucciones'] . "</p>";
22 echo "<img src='" . $receta['imagen'] . "' alt='Imagen de la Receta' width='400' height='300'>";
23
24
25 //Cerramos la conexión con la BD
26 $conexion->close();
27 ?>
```

Después accedemos a él mediante la url localhost/activity_2.php para así ejecutarlo y que nos muestre los datos de esa receta.

     localhost/activity_2.php

Pollo al Horno con Patatas

2024-05-01 09:40:13


Español

Muslos de pollo, patatas, cebolla, aceite de oliva.

60

Bajo

Marinar el pollo con ajo, romero y aceite de oliva, colocar en una bandeja con patatas y cebolla, hornear hasta que esté dorado y crujiente.



3. CREACIÓN DE LA PÁGINA DE INICIO

Para crear la página de inicio del sitio web, vamos a crear un php, el cual va a tener los siguientes pasos:

- Conexión con nuestra base de datos de las recetas
- Consulta a la base de datos para obtener las ultimas recetas que se han publicado
- Crear una lista y un bucle para mostrar la información de las recetas. Esta información va a ser: Nombre de la receta, fecha de publicación, categoría, tiempo de preparación, nivel de dificultad, las primeras 30 palabras de las instrucciones de la preparación y la imagen correspondiente de esa receta
- Cerrar la conexión con la base de datos

Este código va a ser el siguiente.

```
PHP index.php > ...
1
2
3 <?php
4
5 //Conexion con la base de datos
6 $conexion = new mysqli("localhost","root","","Recetas");
7
8 //Consulta a la base de datos para conseguir las últimas 5 recetas
9 $datos = "SELECT * from recetas ORDER BY fechaHora DESC LIMIT 5";
10 $resultado = $conexion->query($datos);
11
12 //Bucle para mostrar la información de las recetas en formato de lista
13 echo "<ul>";
14 while($receta = $resultado->fetch_assoc()){
15     echo "<li>";
16     echo "<h2>" . "Nombre " . $receta['nombreReceta'] . "</h2>";
17     echo "<p>" . "Fecha de publicación " . date('d/m/Y', strtotime($receta['fechaHora'])) . "</p>";
18     echo "<p>" . "Categoría " . $receta['categoria'] . "</p>";
19     echo "<p>" . "Tiempo de preparación " . $receta['tiempoPreparacion'] . " minutos" . "</p>";
20     echo "<p>" . "Nivel de dificultad " . $receta['nivelDificultad'] . "</p>";
21
22     //Limite de caracteres en las instrucciones
23     //Para ajustar el límite de las palabras, voy a dividir el texto de las instrucciones en palabras
24     //y seleccionar las 30 primeras
25     $palabras = explode(' ', $receta['instrucciones']);
26     $primeras30Palabras = implode(' ', array_slice($palabras, 0,30));
27
28     echo "<p>" . "Instrucciones " . $primeras30Palabras . "</p>";
29     echo "<img src='" . $receta['imagen'] . "' alt='Imagen de la Receta' width='400' height='300'>";
30     echo "</li>";
31 }
32 echo "</ul>";
33
34 //Cerramos la conexión con la BD
35 $conexion->close();
36 >
```

Y al acceder desde el navegador a ese archivo se nos va a mostrar la lista con las ultimas 5 recetas publicadas.

• Nombre Pasta con champiñones

Fecha de publicación 23/05/2024

Categoría Italiano

Tiempo de preparación 20 minutos

Nivel de dificultad Alto

Instrucciones Cocinar la pasta al dente, saltear los champiñones con ajo, agregar la crema de leche y la mantequilla, mezclar con la pasta y el queso parmesano.



• Nombre Gazpacho Andaluz

Fecha de publicación 20/05/2024

Categoría Español

Tiempo de preparación 10 minutos

Nivel de dificultad Bajo

Instrucciones Triturar todos los ingredientes en una licuadora hasta obtener una textura suave, enfriar en el refrigerador y servir bien frío



4. CREACIÓN DE LAS PÁGINAS PARA LAS RECETAS

En esta 4 actividad, se ha desarrollado la funcionalidad de mostrar todo el listado de las recetas por páginas y de forma individual. Esta actividad tiene el objetivo de permitir a los usuarios que puedan navegar por todas las recetas que están disponibles en nuestra página web y puedan ver la información completa de cada una. Para ello, se han creado 2 nuevas páginas que son **recipes.php** y **post.php**.

Recipes.php es una página donde se puesta un listado paginado de todas las recetas que tenemos disponibles en nuestro sitio web. La cual se ha desarrollado con una paginación que permite mostrar un máximo de 5 recetas por página con el objetivo de mejorar la experiencia del usuario. Cada receta va a contar con los mismos campos que ya teníamos en la página de inicio. Por lo que el código php de esta página quedaría:

```

2
3 <?php
4
5 //Conexion con la base de datos
6 $conexion = new mysqli("localhost","root","","Recetas");
7
8
9 //Consulta a la base de datos para conseguir la información de las recetas
10 $datos = "SELECT * from recetas";
11 $resultado = $conexion->query($datos);
12
13 $limiteRecetasPagina = 5;
14 $total = $resultado->num_rows;
15
16 $pages = ceil($total/$limiteRecetasPagina);
17
18 //Pagina actual
19 if (isset($_GET['pagina'])) {
20     $page = $_GET['pagina'];
21 } else {
22     $page = 1;
23 }
24
25 $start = ($page - 1) * $limiteRecetasPagina;
26
27 $consulta_recetas = "SELECT * FROM recetas LIMIT $start, $limiteRecetasPagina";
28 $resultado_recetas = $conexion->query($consulta_recetas);
29
30
31 echo "<ul>";
32 while($receta = $resultado_recetas->fetch_assoc()){
33     echo "<li>";
34     echo "<h2><a href='post.php?id=" . $receta['Id'] . "'> " . $receta['nombreReceta'] . "</a></h2>";
35     echo "<p> " . "Fecha de publicación " . date('d/m/Y', strtotime($receta['fechaHora'])) . "</p>";
36     echo "<p> " . "Categoría " . $receta['categoria'] . "</p>";
37     echo "<p> " . "Tiempo de preparación " . $receta['tiempoPreparacion'] . " minutos " . "</p>";
38     echo "<p> " . "Nivel de dificultad " . $receta['nivelDificultad'] . "</p>";
39
40     //Limite de caracteres en las instrucciones
41     //Para ajustar el limite de las palabras, voy a dividir el texto de las instrucciones en palabras
42     //y seleccionar las 30 primeras
43     $palabras = explode(' ', $receta['instrucciones']);
44     $primeras30Palabras = implode(' ', array_slice($palabras, 0,30));
45
46     echo "<p> " . "Instrucciones " . $primeras30Palabras . "</p>";
47     echo "<img src='" . $receta['imagen'] . "' alt='Imagen de la Receta' width='400' height='300'>";
48     echo "</li>";
49 }
50 echo "</ul>";
51
52 // Mostrar enlaces de las páginas
53 echo "<div>";
54 for ($i = 1; $i <= $pages; $i++) {
55     echo "<a href='?pagina=$i'>$i</a> ";
56 }
57 echo "</div>";
58
59 //Cerramos la conexión con la BD
60 $conexion->close();
61 ?>

```

Post.php es una página única de cada receta, en la cual se va a mostrar la información completa de una receta específica, ya que a esta página se la va a pasar el id correspondiente de la receta a través de la URL. Por lo que el código php de esta página quedaría:

```

2
3 <?php
4
5 //Conexion con la base de datos
6 $conexion = new mysqli("localhost","root","","Recetas");
7
8 //Variable donde se guarda el id de la receta que recibimos por la url
9 $idReceta = $_GET['id'];
10
11 //Consulta para obtener los datos de esa receta
12 $datos_receta = "SELECT * from Recetas where Id = $idReceta";
13 $resultado = $conexion->query($datos_receta);
14
15 //Comprobación de si tenemos la receta con ese id
16 if($resultado->num_rows>0){
17     $receta = $resultado->fetch_assoc();
18     echo "<h2>" . "Nombre " . $receta['nombreReceta'] . "</a></h2>";
19     echo "<p>" . "Fecha de publicación " . date('d/m/Y', strtotime($receta['fechaHora'])) . "</p>";
20     echo "<p>" . "Categoría " . $receta['categoria'] . "</p>";
21     echo "<p>" . "Tiempo de preparación " . $receta['tiempoPreparacion'] . " minutos" . "</p>";
22     echo "<p>" . "Nivel de dificultad " . $receta['nivelDificultad'] . "</p>";
23
24     echo "<p>" . "Instrucciones " . $receta['instrucciones'] . "</p>";
25     echo "<img src='\" . $receta['imagen'] . \"' alt='Imagen de la Receta' width='400' height='300'>";
26 }
27 else{
28     echo "No se ha encontrado esa receta";
29 }
30
31 //Cerramos la conexión con la BD
32 $conexion->close();
33 ?>

```

Además de crear estas dos nuevas páginas, se han añadido enlaces en el nombre de cada receta en la página de inicio y en la página de la lista de las recetas para que los usuarios puedan acceder de una forma más fácil a los detalles de las recetas.

Se ha pasado de:

```

echo "<h2>" . "Nombre " . $receta['nombreReceta'] . "</a></h2>";

```

A esto:

```

echo "<h2><a href='post.php?id= \" . $receta['Id'] . \"'>" . "Nombre " . $receta['nombreReceta'] . "</a></h2>";

```

Gracias a estas nuevas implementaciones y modificaciones de código los usuarios van a poder explorar y acceder a la información completa de cada receta.

5. ORDENACIÓN Y FILTRO DE LAS RECETAS

En esta actividad, se ha implementado la funcionalidad de permitir a los usuarios poder ordenar y filtrar las recetas mediante varias opciones que son la categoría, el nivel de dificultad y ordenar por el tiempo de preparación y por el título.

Se ha añadido al principio la parte de filtros en HTML para que el usuario los pueda seleccionar y en PHP mediante la URL se le van a pasar los valores de estos parámetros.

Por lo que el código final ha quedado así:

Parte del HTML donde el usuario selecciona los filtros:

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Recetas</title>
7  </head>
8  <body>
9
10 <h1>Listado de las Recetas</h1>
11
12 <!--Formulario con los filtros-->
13 <form action="recipes.php" method="get">
14     <!--Formulario de ordenación-->
15     <label for="ordenTitulo">Ordenar Título:</label>
16     <select name="ordenTitulo" id="ordenTitulo">
17         <option value="titulo_asc">Título (A-Z)</option>
18         <option value="titulo_desc">Título (Z-A)</option>
19     </select>
20
21     <label for="ordenTiempo">Ordenar Tiempo:</label>
22     <select name="ordenTiempo" id="ordenTiempo">
23         <option value="tiempo_asc">Ascendente</option>
24         <option value="tiempo_desc">Descendente</option>
25     </select>
26     <!--Formulario de filtro-->
27
28     <label for="categoria">Categoría:</label>
29     <select name="categoria" id="categoria">
30         <option value="">Todas</option>
31         <option value="Italiano">Italiano</option>
32         <option value="Mexicano">Mexicano</option>
33         <option value="Español">Español</option>
34     </select>
35
36     <label for="dificultad">Nivel de Dificultad:</label>
37     <select name="dificultad" id="dificultad">
38         <option value="">Todas</option>
39         <option value="Alto">Alto</option>
40         <option value="Medio">Medio</option>
41         <option value="Bajo">Bajo</option>
42     </select>
43
44     <input type="submit" value="Aplicar Filtros">
45 </form>
46
```

Código php donde se reciben los parámetros y se hace la consulta a la BD

```

49 // Conexión con la base de datos
50 $conexion = new mysqli("localhost", "root", "", "Recetas");
51
52 // Consulta base
53 $datos = "SELECT * FROM recetas";
54
55 // Variables para los filtros de búsqueda
56 $filtro = [];
57 if (isset($_GET['categoria']) && $_GET['categoria'] != "") {
58     $categoria = $_GET['categoria'];
59     $filtro[] = "categoria = '$categoria'";
60 }
61
62 if (isset($_GET['dificultad']) && $_GET['dificultad'] != "") {
63     $dificultad = $_GET['dificultad'];
64     $filtro[] = "nivelDificultad = '$dificultad'";
65 }
66
67 if (count($filtro) > 0) {
68     $datos .= " WHERE " . implode(" AND ", $filtro);
69 }
70
71 // Añadir ordenación a la consulta
72 $orden = [];
73 if (isset($_GET['ordenTitulo']) && $_GET['ordenTitulo'] != "") {
74     $ordenTitulo = $_GET['ordenTitulo'];
75     if ($ordenTitulo == 'titulo_asc') {
76         $orden[] = "nombreReceta DESC";
77     } elseif ($ordenTitulo == 'titulo_desc') {
78         $orden[] = "nombreReceta ASC";
79     }
80 }
81
82 if (isset($_GET['ordenTiempo']) && $_GET['ordenTiempo'] != "") {
83     $ordenTiempo = $_GET['ordenTiempo'];
84     if ($ordenTiempo == 'tiempo_asc') {
85         $orden[] = "tiempoPreparacion DESC";
86     } elseif ($ordenTiempo == 'tiempo_desc') {
87         $orden[] = "tiempoPreparacion ASC";
88     }
89 }
90
91 if (count($orden) > 0) {
92     $datos .= " ORDER BY " . implode(", ", $orden);
93 }
94
95 // Ejecutar consulta para obtener el total de resultados
96 $resultado = $conexion->query($datos);
97 $total = $resultado->num_rows;
98
99 // Paginación
100 $limiteRecetasPagina = 5;
101 $pages = ceil($total / $limiteRecetasPagina);
102
103 // Página actual
104 if (isset($_GET['pagina'])) {
105     $page = $_GET['pagina'];
106 } else {
107     $page = 1;
108 }
109

```

Código PHP para mostrar las recetas correspondientes:

```

10 $start = ($page - 1) * $limiteRecetasPagina;
11
12 // Añadir límites de paginación a la consulta
13 $datos .= " LIMIT $start, $limiteRecetasPagina";
14 $resultado_recetas = $conexion->query($datos);
15
16 echo "<ul>";
17 while ($receta = $resultado_recetas->fetch_assoc()) {
18     echo "<li>";
19     echo "<h2><a href='post.php?id=" . $receta['Id'] . "'> . $receta['nombreReceta'] . "</a></h2>";
20     echo "<p>Fecha de publicación: " . date('d/m/Y', strtotime($receta['fechaHora'])) . "</p>";
21     echo "<p>Categoría: " . $receta['categoria'] . "</p>";
22     echo "<p>Tiempo de preparación: " . $receta['tiempoPreparacion'] . " minutos</p>";
23     echo "<p>Nivel de dificultad: " . $receta['nivelDificultad'] . "</p>";
24
25     // Límite de palabras en las instrucciones
26     $palabras = explode(' ', $receta['instrucciones']);
27     $primeras30Palabras = implode(' ', array_slice($palabras, 0, 30));
28
29     echo "<p>Instrucciones: " . $primeras30Palabras . "</p>";
30     echo "<img src='" . $receta['imagen'] . "' alt='Imagen de la Receta' width='400' height='300'>";
31     echo "</li>";
32 }
33 echo "</ul>";
34
35 // Mostrar enlaces de las páginas
36 echo "<div>";
37 for ($i = 1; $i <= $pages; $i++) {
38     echo "<a href='?pagina=$i&ordenTitulo=" . (isset($_GET['ordenTitulo']) ? $_GET['ordenTitulo'] : '') .
39         "&ordenTiempo=" . (isset($_GET['ordenTiempo']) ? $_GET['ordenTiempo'] : '') .
40         "&categoria=" . (isset($_GET['categoria']) ? $_GET['categoria'] : '') .
41         "&dificultad=" . (isset($_GET['dificultad']) ? $_GET['dificultad'] : '') . "'>$i</a> ";
42 }
43 echo "</div>";
44
45 // Cerrar la conexión con la BD
46 $conexion->close();
47 >>
48
49 </body>
50 </html>
51

```

En resumen, en esta actividad se ha implementado un código que es una página web para mostrar las recetas con opciones de filtrado y de paginación. El formulario que se ha implementado en HTML va a permitir al usuario seleccionar varios filtros como el de categoría y el de nivel de dificultad de la receta y le va a permitir poder ordenar las recetas por el nombre o el tiempo de preparación. Estos datos se van a enviar por el método GET.

El código en PHP va a recibir estos filtros y va a realizar la consulta a la BD en base a los valores que ha recibido con el objetivo de obtener las recetas que coincidan con esos filtros.

Y así el usuario va a poder la información de cada receta, la cual va a consistir en el nombre, la fecha de publicación, la categoría, el tiempo de preparación, el nivel de dificultad, las instrucciones y la imagen. Por último, en el caso de que el resultado sea mayor a 5 recetas, el usuario va a poder navegar entre varias páginas, ya que en cada página el número máximo de recetas va a ser 5.

6. MENÚ

En esta actividad se ha implementado el menú de navegación, el cual va a ser visible en todas las páginas de nuestro sitio web y nos va a permitir acceder de

una forma más fácil a las diferentes secciones que se han descrito en el enunciado de la práctica.

El menú estará formado por enlaces a las siguientes páginas.

- **Inicio:** Muestra un listado de las últimas 5 recetas publicadas
- **Actividad 2:** Muestra el resultado de la 2 Actividad
- **Recetas:** Muestra un listado de todas las recetas, junto con los filtros para encontrar mejor las recetas
- **API_recetas:** Acceso a la API que muestra la primera página de recetas en formato JSON
- **API_receta:** Acceso a la API que muestra una receta en formato JSON
- **Login:** Muestra el formulario de inicio de sesión. Este enlace solo se mostrará si no hay un usuario logueado en la página
- **Sign Up:** Muestra el formulario para crear un nuevo usuario. Este enlace solo se mostrará si no hay un usuario logueado ya en la página
- **Perfil de usuario:** Va a dar la opción al usuario logueado de editar su perfil. Este enlace solo se va a mostrar si la página cuenta con un usuario ya logueado
- **Logout:** Va a cerrar la sesión del usuario y le va a redirigir a la página de inicio. Este enlace solo se va a mostrar si la página cuenta con un usuario ya logueado

El código de este menú quedaría así:

```
<!--Menu navegación -->

<div class="menu">
  <a href="index.php">Home</a>
  <a href="activity_2.php">Act_2</a>
  <a href="recipes.php">Recetas</a>
  <a href="api/recipes/1" target="_blank">API_recetas</a>
  <a href="api/recipe/1" target="_black">API_receta</a>

  <?php if(!isset($_SESSION['logged_in'])): ?>
  <a href="login.php">Login</a>
  <a href="signup.php">Sign up</a>
  <?php else: ?>
  <a href="edit.php">Perfil usuario</a>
  <a href="logout.php">Logout</a>
  <?php endif; ?>
</div>
```

Respecto a las líneas que están en código php. Sirven para realizar la comprobación de si existe un usuario que ya está logueado en el sitio web o no.

```
<?php if(!isset($_SESSION['logged_in'])): ?>
<a href="login.php">Login</a>
<a href="signup.php">Sign up</a>
<?php else: ?>
<a href="edit.php">Perfil usuario</a>
<a href="logout.php">Logout</a>
<?php endif; ?>
```

Después de crear menu.php, se ha añadido a las demás páginas de nuestro sitio web mediante el código de require “menu.php”, con el objetivo de mejorar la experiencia del usuario de navegar por el sitio web.

(Dentro de menu.php también se ha utilizado CSS para dar estilo al menú y que se aprecien mejor de que el usuario va a tener un menú de navegación)

7. API

En esta actividad se ha implementado una API que es de solo lectura para proporcionar acceso a los datos sobre las recetas desde el servidor. Se va a poder acceder a ella de dos formas diferentes (Dos puntos de entrada): “/api/recipes/<page>” y “/api/recipe/<id>”, la cual nos van a devolver datos en formato JSON de todas las recetas y de una específica. A continuación se muestran los pasos necesarios y modificaciones que se han realizado en el código que ya teníamos para configurar y utilizar esta API. También se muestran capturas de pantalla del código y del resultado obtenido.

El primero de los pasos ha sido crear los dos documentos **api/recipes.php** y **api/recipe.php**

Fichero api/recipes.php/<page>: Este fichero nos va a devolver un listado de recetas en formato JSON. El parámetro <page> nos permite seleccionar la página de los resultados. Se ha puesto de límite 10 recetas por página

Código del fichero recipes.php


```

5
6 // Conexión con la base de datos
7 $conexion = new mysqli("localhost", "root", "", "Recetas");
8
9
10 //Variable para obtener el numero de pag y paginación
11 $numPage = 1;
12 $limitRecetas = 10;
13 $start = ($numPage - 1) * $limitRecetas;
14
15 if (isset($_GET['page'])) {
16 |     $numPage = $_GET['page'];
17 }
18
19 //Consulta a la base de datos
20 $datos = "SELECT * from recetas LIMIT $limitRecetas OFFSET $start";
21 $resultado_datos = $conexion->query($datos);
22
23 $recetas = [];
24
25 //Se comprueba si ha encontrado resultados
26 if($resultado_datos->num_rows > 0){
27 |     //Mientras tenga recetas, se van a ir añadiendo al array
28 |     while($row = $resultado_datos->fetch_assoc()){
29 |         $recetas[] = $row;
30 |     }
31 |     //Imprime los resultados de las recetas en formato JSON
32 |     echo json_encode($recetas);
33 }
34 else{
35 |     echo json_encode(["Error" => "Error al encontrar las recetas"]);
36 }
37
38
39 //Cerrar la conexión con la BD
40 $conexion->close();
41
42 ?>

```

Fichero api/récipe/<id>: Nos va a devolver la información de una receta en específica en formato JSON, esta receta que se devuelve va a estar identificado por el ID

Código del fichero recipe.php

```

4
5 <?php
6
7 // Conexión con la base de datos
8 $conexion = new mysqli("localhost", "root", "", "Recetas");
9
10
11 //Variable para obtener el id
12 $id = 0;
13
14 if (isset($_GET['id'])) {
15     $id = $_GET['id'];
16 }
17
18 //Consulta a la base de datos
19 $datos = "SELECT * from recetas WHERE Id = $id";
20 $resultado_datos = $conexion->query($datos);
21
22 //Se comprueba si ha encontrado la receta de la que se pasa el id
23 if($resultado_datos->num_rows>0){
24     //En el caso de que encuentre la receta, se obtiene la información de ella para mostrarla por pantalla
25     $receta = $resultado_datos->fetch_assoc();
26     //Imprime la información de la receta en formato JSON
27     echo json_encode($receta);
28 }
29 else{
30     echo json_encode(["error" => "Receta no encontrada"]);
31 }
32
33 //Cerrar la conexión con la BD
34 $conexion->close();
35
36 ?>

```

Después de crear estos dos ficheros, el siguiente paso es modificar el menú de navegación añadiendo la dirección del enlace a estas páginas y pasándoles los valores correspondientes a cada una que son page e id (En cada una de ellas se las ha pasado el número 1). Por lo tanto el código de esta parte del menú quedaría:

```

<a href="activity_2.php" >Act_2</a>
<a href="recipes.php">Recetas</a>
<a href="api/recipes.php?page=1" target="_blank">API_recetas</a>
<a href="api/recipe.php?id=1" target="_black">API_receta</a>

```

Se ha utilizado la opción de **target="_blank"** para que al acceder a estas dos API, se nos muestre la información en una pestaña nueva.

Y como resultado de cada uno de los php, la información que se nos muestra por pantalla es la siguiente.

Resultado de la api/recipes.php?page=1

The screenshot shows a web browser window with the address bar displaying 'localhost/R3BACK/api/recipes.php?page=1'. The page content is a large JSON array of recipe objects. Each object contains fields such as 'Id', 'nombreReceta', 'fechaHora', 'categoria', 'ingredientes', 'tiempoPreparacion', 'nivelDificultad', 'instrucciones', and 'imagen'. The first few recipes listed are 'Pollo al Homo con Patatas', 'Españolito', and 'Tortilla-espanola-potato-omel'.

Resultado de la api/recipes.php?id=1



8. LOGIN

Crear la nueva tabla llamada “users_pec3”, la cual va a estar formada por los campos de username, nombre, apellidos, password y repetirContraseña (Este campo es opcional).

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice	Comentarios	Virtualidad	Mover
username	VARCHAR	255	Ninguno			<input checked="" type="checkbox"/>	PRIMARY			
nombre	VARCHAR	255	Ninguno			<input type="checkbox"/>				
apellidos	VARCHAR	255	Ninguno			<input type="checkbox"/>				
password	VARCHAR	255	Ninguno			<input type="checkbox"/>				
confirmpassword	VARCHAR	255	Ninguno			<input checked="" type="checkbox"/>				

Después de crear la tabla se ha añadido el usuario con los datos de mi usuario de la UOC.

username	nombre	apellidos	password	confirmpassword
mcanas21	Maria	Canas Encinas	\$2y\$10\$qw8tSDEt/aeWjPKQZeDVOyEfwALNuBY6iDj1/r1Vy...	NULL

El código de iniciar sesión va a estar formado por código HTML donde se ha implementado el formulario y código PHP que es el que procesa los datos enviados por el usuario en el formulario y va a comprobar si los datos coinciden con los que tenemos en la BD.

Código HTML del formulario, el cual está formado por los campos de nombre de usuario y contraseña y un botón para enviar esos datos.

```

<body>

    <h1>Iniciar Sesion</h1>

    <form action="login.php" method="POST">
        <label for="username"> Nombre de Usuario </label>
        <input type="text" id="username" name="username" required>
        <label for="password">Contraseña</label>
        <input type="password" id="password" name="password" required>
        <button type="submit">Iniciar Sesion</button>
    </form>

```

Código PHP que procesa los datos del formulario

```

<?php
//Comprueba si el formulario con los datos de iniciar sesión se han enviado
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $conexion = new mysqli("localhost","root","","Recetas");

    //Variable donde se almacena los datos introducidos por el usuario
    $username = $_POST['username'];
    $password = $_POST['password'];

    //Consulta a la tabla de la bd para comprobar si existen los datos introducidos
    $sql = "SELECT * FROM users_pec3 WHERE username = ?";
    //Para obtener los datos del usuario de la BD se usa prepare para evitar inyecciones SQL
    $resultado_datos = $conexion->prepare($sql);
    $resultado_datos->bind_param("s",$username);
    $resultado_datos->execute();
    $rst = $resultado_datos->get_result();

    //Si ya tengo a ese usuario creado en la base de datos
    if($rst->num_rows === 1){
        $datos_usuario = $rst->fetch_assoc();
        if(password_verify($password, $datos_usuario['password'])){
            $_SESSION['username'] = $datos_usuario['username'];
            //Mensaje de bienvenida a la página
            echo "<p>Bienvenida a la pagina " . $_SESSION['username'] . "</p>";
            echo "<p><a href='logout.php'> Salir sesion </a></p>";
        }
        //Si los datos de la contraseña no coinciden
        else{
            echo "<p>Datos NO correctos </p>";
        }
    }
    //Si no existe ese usuario
    else{
        echo "<p>Datos NO correctos </p>";
    }
    $resultado_datos->close();
    $conexion->close();
}
?>

```

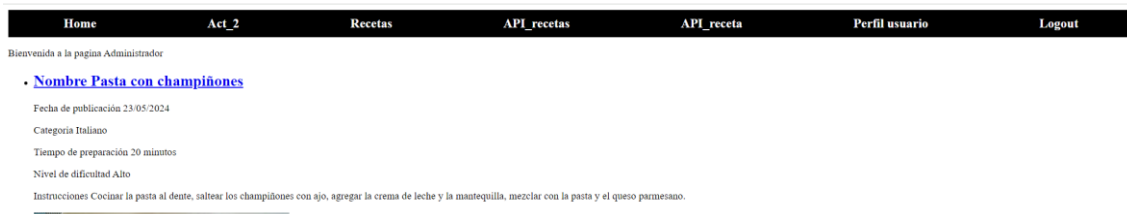
El código implementado para añadir el usuario con nuestros datos de la UOC es el siguiente:

```

4
5 require 'menu.php';
6 // Conexión a la base de datos
7 $conexion = new mysqli("localhost", "root", "", "Recetas");
8
9
10 // Datos de mi usuario de la UOC
11 $username = "mcanas21";
12 $password = password_hash("mcanas21", PASSWORD_BCRYPT);
13 $nombre = "Maria";
14 $apellidos = "Canas Encinas";
15
16 // Insertar usuario en la base de datos
17 $sql = "INSERT INTO users_pec3 (username, nombre, apellidos, password) VALUES (?, ?, ?, ?)";
18 $stmt = $conexion->prepare($sql);
19 $stmt->bind_param("ssss", $username, $nombre, $apellidos, $password);
20
21 if ($stmt->execute()) {
22     echo "Usuario insertado correctamente";
23     $_SESSION['username'] = $username;
24     //Mensaje de bienvenida a la página
25     echo "<p>Bienvenida a la pagina " . $_SESSION['username'] . "</p>";
26 } else {
27     echo "Error: " . $stmt->error;
28 }
29
30 // Cerrar conexión
31 $stmt->close();
32 $conexion->close();
33 ?>

```

Por lo tanto, si en la parte del menú accedemos a signup se va a añadir el usuario a la Base de Datos y se va a mostrar el mensaje de bienvenida a la página y las opciones del menú de login y singup pasaran a ser logout y editar perfil. Y se nos mostrará el mensaje de bienvenida a la página con el nombre del usuario.



En el caso de que el usuario introduzca a la hora de iniciar sesión unos datos que no son correctos, la página nos va a mostrar el mensaje con datos NO correctos.



9. ALTA Y EDICIÓN DE USUARIOS

Para el alta de usuario se ha creado el código de signup.php, el cual va a estar formado por un formulario HTML con los diferentes campos y un código PHP para procesar esos datos introducidos.

Código HTML con el formulario

```
<body>
  <h2>Crear usuario</h2>
  <form action="signup.php" method="post">
    <label for="username">Nombre usuario</label>
    <input type="text" id="username" name="username" required>
    <label for="nombre">Nombre</label>
    <input type="text" id="nombre" name="nombre" required>
    <label for="apellidos">Apellidos</label>
    <input type="text" id="apellidos" name="apellidos" required>
    <label for="password">Contraseña</label>
    <input type="password" id="password" name="password" required>
    <label for="password">Confirmar Contraseña</label>
    <input type="password" id="confirmpassword" name="confirmpassword">
    <button type="submit">Iniciar Sesion</button>
  </form>
</php>
```

Código PHP que procesa el formulario

Este código lo primero que hace es comprobar si han enviado los datos del formulario y los va a guardar en las variables correspondientes. Después va a establecer la conexión con la base de datos y va a comprobar si tenemos o no un usuario ya creado con ese nombre. En el caso de que ya exista, se va a mostrar un mensaje informando de ello, en caso contrario va a insertar este nuevo usuario y si se ha realizado de forma correcta va a mostrar el mensaje de ello y nos va a redirigir a la página principal.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $nombre = $_POST['nombre'];
    $apellidos = $_POST['apellidos'];
    $password = $_POST['password'];
    $confirmpassword = $_POST['confirmpassword'];

    // Conexión a la base de datos
    $conexion = new mysqli("localhost", "root", "", "Recetas");

    //Comprobar si el usuario que quiere crear ya existe
    $sqlUse = $conexion->prepare("SELECT * FROM users_pec3 WHERE username = ?");
    $sqlUse->bind_param("s",$username);
    $sqlUse->execute();
    $resUser = $sqlUse->get_result();

    //Si encuentra un usuario con ese username
    if($resUser->num_rows>0){
        echo "<p> Ya existe un usuario con ese nombre. Cambia el nombre </p>";
    }
    //Si no tengo ese usuario con ese username
    else{
        //comprobación de si las dos contraseñas son iguales
        if($confirmpassword == $password){
            // Cifrado de la contraseña
            $hashed_password = password_hash($password, PASSWORD_BCRYPT);

            // Inserción del usuario en la base de datos
            $datos = $conexion->prepare("INSERT INTO users_pec3 (username, nombre, apellidos, password) VALUES (?, ?, ?, ?)");
            $datos->bind_param("ssss", $username, $nombre, $apellidos, $hashed_password);
```

```

60
61         if ($datos->execute()) {
62             echo "Usuario registrado Correctamente";
63             header("Location: index.php");
64             exit();
65         } else {
66             echo "Error al crear el usuario ";
67         }
68     }
69     else{
70         echo "<p>Las contraseñas no coinciden </p>";
71     }
72     $datos->close();
73
74
75     $resUser->close();
76     $conexion->close();
77 }
78 ?>
79 </body>
80 </html>
81

```

En el caso de que se quiera crear un nuevo usuario y ya contemos con ese nombre de usuario en nuestra base de datos nos va a saltar la información de que ya existe un usuario con ese nombre y que lo deberá de modificar.

Home	Act_2	Recetas	API_recetas	API_receta	Login	Sign up
Crear usuario Nombre usuario <input type="text" value="Administrador"/> Nombre <input type="text"/> Apellidos <input type="text"/> Contraseña <input type="password" value="*****"/> Confirmar Contraseña <input type="text"/> <input type="button" value="Iniciar Sesión"/> Ya existe un usuario con ese nombre. Cambia el nombre						

Para la edición de los datos del usuario se ha creado el archivo edit.php, que estará formado por un formulario con los campos de nombre, apellidos y la nueva contraseña que quiera introducir. Para modificar los datos de nuestra cuenta, el usuario una vez iniciada sesión va a tener que acceder desde el menú superior a la parte de Perfil usuario y se nos mostrará el formulario y van a estar rellenados con nuestros datos el campo de nombre y el de apellidos.

Home	Act_2	Recetas	API_recetas	API_receta	Perfil usuario	Logout
Editar datos usuario Nombre <input type="text" value="Maria"/> Apellidos <input type="text" value="Ganas Encinas"/> Nueva contraseña <input type="text"/> <input type="button" value="Actualizar datos"/>						

Cuando el usuario modifique los datos de su cuenta y se haya realizado de forma correcta, aparecerá el mensaje de que los datos han sido modificados y se van a mostrar los nuevos datos cambiados.

Home	Act_2	Recetas	API_recetas	API_receta	Perfil usuario	Logout
Datos actualizados Editar datos usuario Nombre <input type="text" value="Maria"/> Apellidos <input type="text" value="Ganas Sanz"/> Nueva contraseña <input type="text"/> <input type="button" value="Actualizar datos"/>						

El código del archivo edit.php es el siguiente

Código HTML con el formulario de edición

```
<h2>Editar datos usuario</h2>
<form action="edit.php" method="post">
  <label for="nombre">Nombre</label>
  <input type="text" id="nombre" name="nombre" value="<?php echo htmlspecialchars($nombre); ?>" required>
  <label for="apellidos">Apellidos</label>
  <input type="text" id="apellidos" name="apellidos" value="<?php echo htmlspecialchars($apellidos); ?>" required>
  <label for="password">Nueva contraseña</label>
  <input type="password" id="password" name="password">
  <button type="submit">Actualizar datos</button>
</form>

</body>
```

Código PHP

Este código va a recibir los datos que ha añadido el usuario y los va a modificar en la Base de datos

```
<?php

$username = $_SESSION['username'];

// Conexión a la base de datos
$conexion = new mysqli("localhost", "root", "", "Recetas");

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $new_nombre = $_POST['nombre'];
    $new_apellidos = $_POST['apellidos'];
    $new_password = $_POST['password'];

    // Cifrado de la nueva contraseña si se proporciona
    if (!empty($new_password)) {
        $hashed_password = password_hash($new_password, PASSWORD_BCRYPT);
        $datos_update = $conexion->prepare("UPDATE users_pec3 SET nombre = ?, apellidos = ?, password = ? WHERE username = ?");
        $datos_update->bind_param("ssss", $new_nombre, $new_apellidos, $hashed_password, $_SESSION['username']);
    } else {
        $datos_update = $conexion->prepare("UPDATE users_pec3 SET nombre = ?, apellidos = ? WHERE username = ?");
        $datos_update->bind_param("sss", $new_nombre, $new_apellidos, $_SESSION['username']);
    }

    if ($datos_update->execute()) {
        echo "<p>Datos actualizados</p>";
    } else {
        echo "Error: " . $datos_update->error;
    }

    $datos_update->close();
}
```

```
// Obtener la información del usuario
$datos = $conexion->prepare("SELECT nombre, apellidos FROM users_pec3 WHERE username = ?");
$datos->bind_param("s", $username);
$datos->execute();
$datos->bind_result($nombre, $apellidos);
$datos->fetch();
$datos->close();

$conexion->close();

?>
```

Para cerrar la sesión del usuario se ha creado el archivo logout.php el cual se va a encargar de cerrar la sesión del usuario y lo va a redirigir a la página de inicio. El código implementado es el siguiente:

PHP logout.php

```
1
2 <?php require 'menu.php' ?>
3 <?php
4     //Cierre de sesión y redirige al usuario a index.php (Página principal)
5     session_start();
6     session_destroy();
7     header("Location: index.php");
8 ?>
9
```

10. PUBLICACIÓN

El código implementado junto con la base de datos se ha subido a un repositorio en Github, ya que como me pasó la otra vez, me está dando fallo el servidor y no lo he podido subir. Por lo que he creado un repositorio y he subido la documentación a él. La dirección a este repositorio es:

https://github.com/mariacanas/R3_BackEnd

Nombre de usuario: mcanas21 Contraseña: mcanas21