14-1-2025

Gestor de gastos

DAW - TFG

Diego Extremiana

ACCESO AL REPOSITORIO: GESTOR DE GASTOS

Tabla de contenido

Introducción

Este proyecto surge de una problemática común que varios amigos me comentaron: no sabían cómo gastaban tanto dinero ni a dónde se iba. A partir de estas conversaciones, se planteó la necesidad de una herramienta que permitiera a los usuarios tener un mayor control sobre sus finanzas personales.

La solución propuesta es una aplicación web enfocada en la gestión mensual de gastos, donde los usuarios pueden introducir sus gastos diarios y clasificarlos en categorías como "ocio" o "primera necesidad". Esto les permitirá visualizar de manera clara en qué áreas están destinando su dinero y facilitará la planificación y el ahorro.

La aplicación contará con un sistema de registro y login, y ofrecerá resúmenes mensuales de los gastos por categoría, permitiendo a los usuarios tener un mayor control sobre sus finanzas. Además, contribuirá a mejorar la organización financiera de manera sencilla y accesible.

A diferencia de las aplicaciones existentes, que a menudo son complejas o tienen barreras de entrada elevadas, esta herramienta buscará ser accesible y adaptarse a las necesidades básicas de cualquier usuario.

Objetivos del proyecto

Desarrollar una aplicación web que ofrezca a los usuarios una herramienta sencilla y eficaz para gestionar sus finanzas personales, haciendo que puedan registrar, clasificar y analizar sus gastos mensuales de manera intuitiva. El objetivo es facilitar el control de los hábitos de consumo, promoviendo una mejor comprensión de su situación financiera.

Objetivos personales

En este proyecto, mi principal objetivo es desarrollar aplicaciones web que realmente ayude a las personas a alcanzar sus metas. Quiero crear soluciones que no solo sean útiles, sino que también sean fáciles de usar y que aporten valor en la vida cotidiana de los usuarios.

Al mismo tiempo, busco ampliar mis conocimientos en el desarrollo de interfaces que sean tanto intuitivas como atractivas. Creo firmemente que una buena aplicación no solo debe cumplir su función, sino también ofrecer una experiencia agradable a quien la utiliza.

Otro aspecto que quiero fortalecer es mi capacidad técnica, especialmente en tecnologías como JavaScript y la gestión de bases de datos en el backend. A lo largo del proyecto, aplicaré lo que he aprendido hasta ahora y seguiré mejorando en estas áreas, buscando siempre la mejor forma de optimizar la aplicación para su uso real.

Fases del proyecto

Para llevar adelante el desarrollo de mi aplicación web, voy a dividir el proyecto en varias fases, cada una con objetivos y metas claras, aplicando una metodología ágil para avanzar de forma organizada y eficiente.

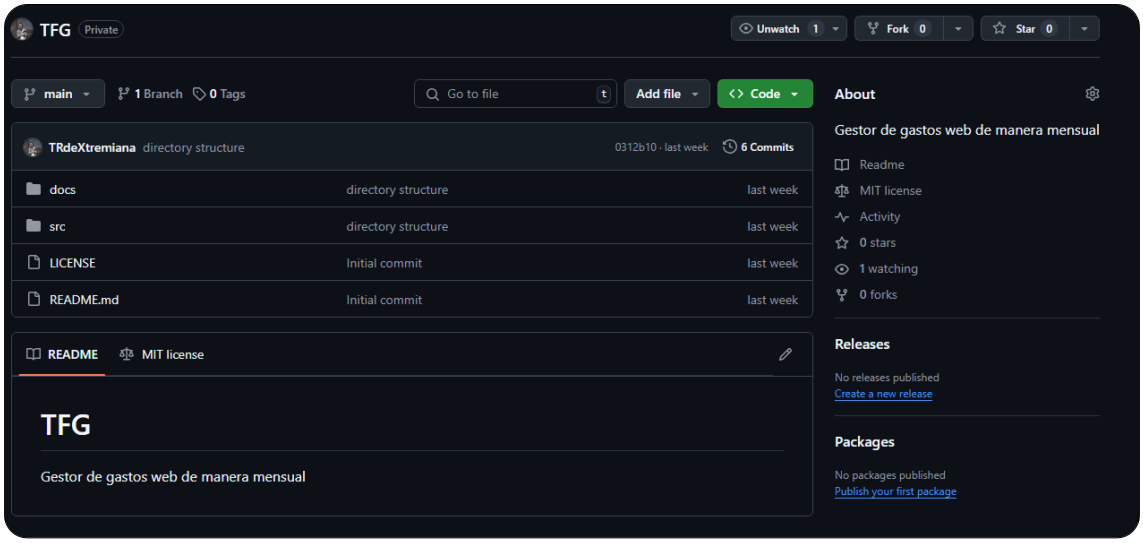
Fase 1: Planificación y Requisitos

Lo primero será tener claro todo lo que debe hacer la aplicación.

En esta fase, definiré los objetivos del proyecto y los requisitos que deberá cumplir, tanto para el usuario como en aspectos técnicos. Esto incluye cosas como: qué opciones de gestión de gastos habrá, cómo se clasifican los gastos por etiquetas, etc.

- Herramientas: Empezaré usando Microsoft Word para organizar todas estas ideas y requisitos.

- Control de versiones: Abriré un repositorio en Git (en GitHub -> https://github.com/TRdeXtremiana/TFG/) para ir guardando cada avance del proyecto.

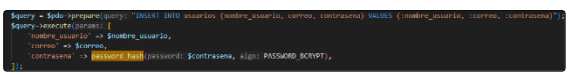


Fase 2: Diseño de la Base de Datos

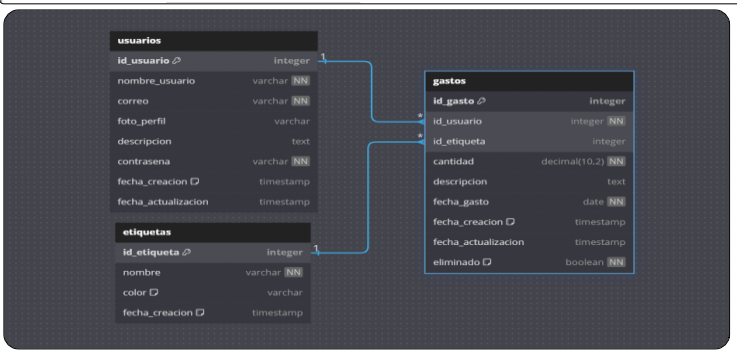
Después, me centraré en el diseño de la base de datos, que será fundamental para que todo funcione correctamente. Aquí definiré las tablas necesarias para almacenar los datos de usuarios, los gastos y las etiquetas, y cómo se relacionan entre ellas.

- Resultado esperado: Crearé un diagrama de la estructura de la base de datos.

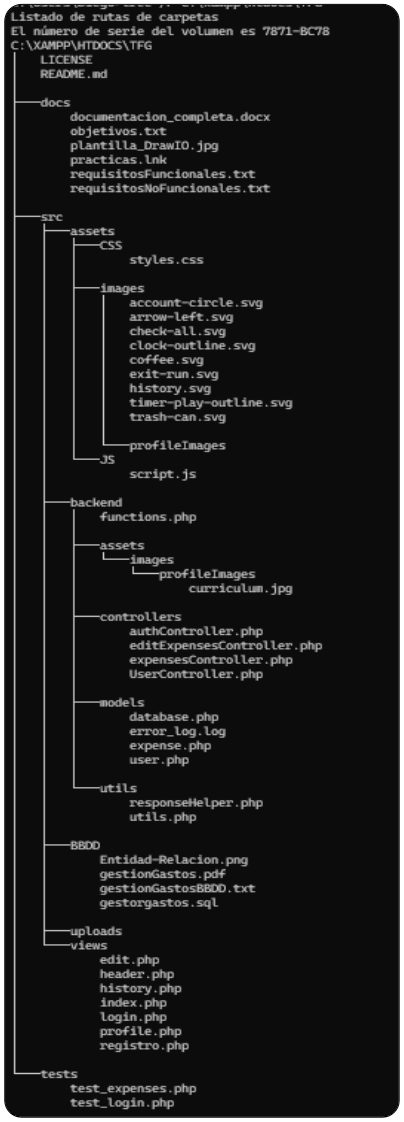
- Tecnologías: Usaré MySQL como gestor de base de datos, y una herramienta como https://dbdiagram.io/d para el diagrama de tablas y aplicaré medidas de seguridad como el cifrado de contraseñas:



En esta fase he tenido resultados no esperados y muchas modificaciones, a menudo he vuelto a editar y reestructurar la propia base de datos.



En este punto he aprovechado para hacer una estructura sólida de carpetas basándome en el MVC:



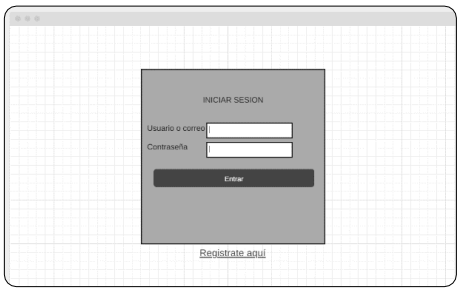
Fase 3: Diseño de la Interfaz de Usuario

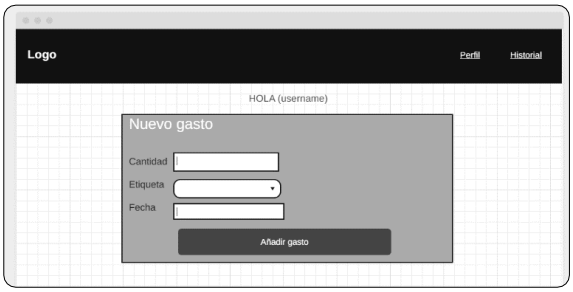
Una vez definida la base de datos, es momento de pensar en cómo interactuarán los usuarios con la aplicación. Aquí diseñaré la interfaz gráfica, pensando en una experiencia de usuario clara y amigable. Definiré pantallas para el login, la introducción de gastos, los resúmenes mensuales, etc.

- Herramientas: Para el diseño visual, me ayudaré de https://wireframe.cc/ para hacer los bocetos y prototipos: https://wireframe.cc/pro/pp/d2bc399a4855103#rfpsqif5.

- Tecnologías: HTML y CSS serán la base del diseño de la interfaz en el proyecto final.

- Control de versiones: Guardaré los diseños en mi repositorio y también una copia en local para tenerlos organizados.



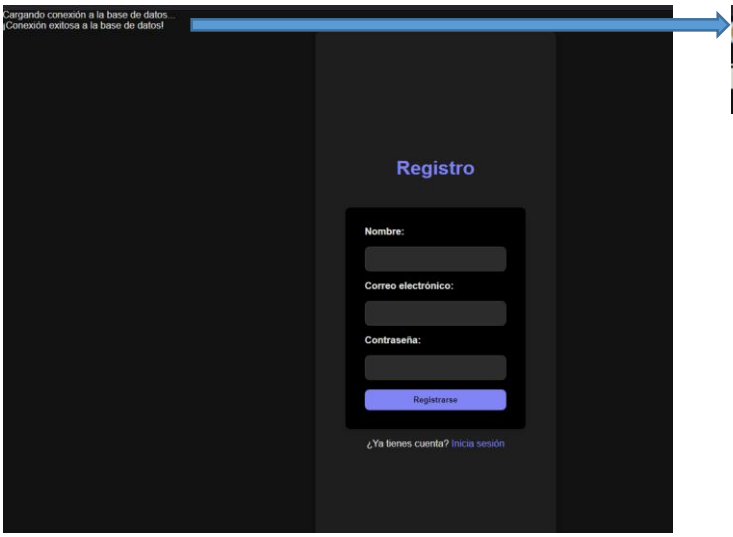


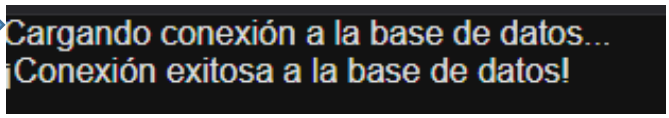
Fase 4: Desarrollo del Backend

Aquí empieza la programación más “pesada”. Crearé la lógica de servidor con PHP para manejar todo el sistema, como el registro y login de usuarios, la inserción y edición de gastos, y el guardado de los resúmenes mensuales. Esta parte es clave para que la aplicación gestione de manera segura los datos de cada usuario.

- Tecnologías: Usaré PHP y MySQL para la lógica de servidor y la conexión a la base de datos.

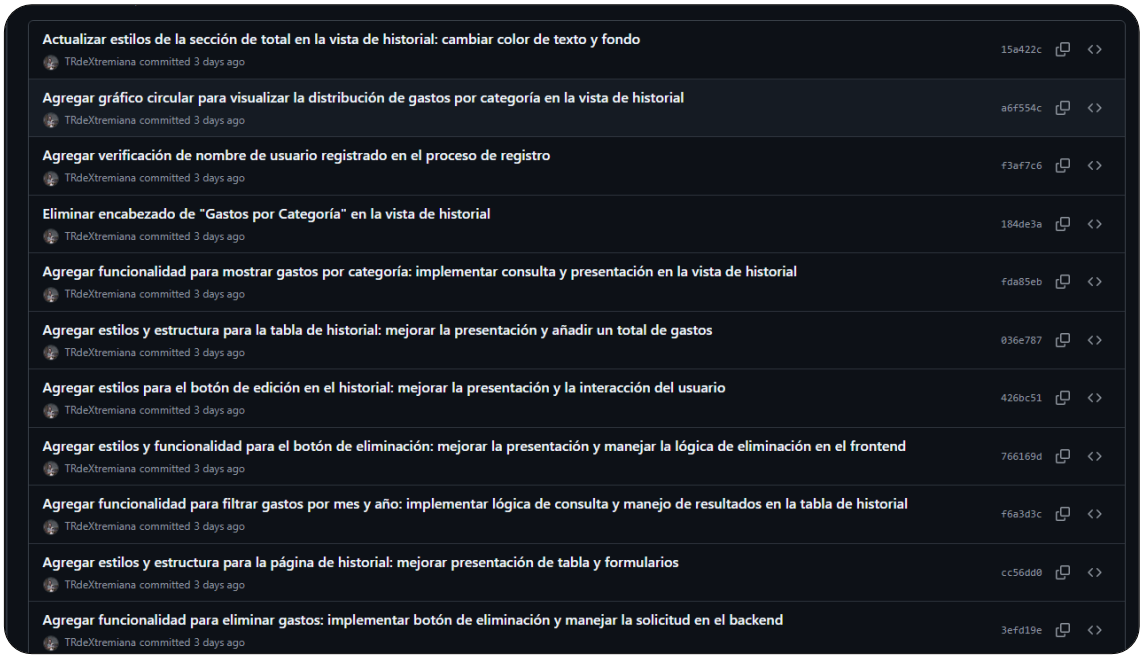
- Control de versiones: Cada avance en el desarrollo del backend estará versionado en el repositorio para llevar un control de los cambios.





Por supuesto, esto no aparecerá en la versión final de la aplicación

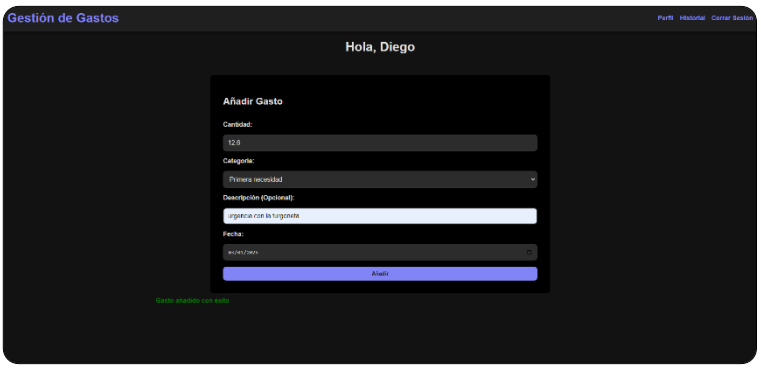
	id_usuario	nombre_usuario	correo	foto_perfil	descripcion	contrasena	fecha_creacion	fecha_actualizacion
Edit Copy Delete	1	Diego	dextremiana1998@gmail.com	assets/images/profileImages/curriculum.jpg	Soy solo un niño	$2y$10$0KehvBT0RE9dBFqG1pAhOeqmJNVjjl50kWK2O4Ni8SR...	2024-12-31 11:11:15	2025-01-03 14:11:11
Edit Copy Delete	2	pepe	pepe@pepenson.comm	NULL	Me llamo Pepe	$2y$10$14aU4aRDj/IKkLI2W0rpTOnN1PM2ZnDr98lkK91AG5o...	2024-12-31 11:11:40	2024-12-31 15:14:39
Edit Copy Delete	3	Marshall	marshall@hazmeuna.com	NULL	NULL	$2y$10$IDSANgyO21c8ndO2OhadYe9Xvblaud7nQU.coDy7tPz...	2025-01-03 14:08:17	2025-01-03 14:08:17
Edit Copy Delete	4	JUAN	juan@juan.com	NULL	NULL	$2y$10$PwJTtg/w1fyqEbC2uVxtzuXicEj3hxE7EfZyWz7gCAe...	2025-01-03 14:38:06	2025-01-03 14:38:06

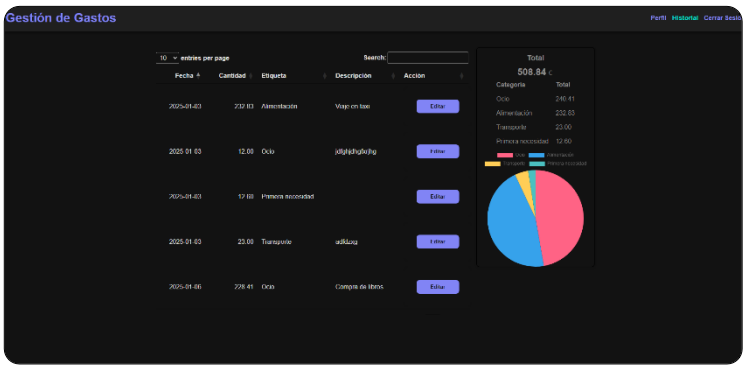


Fase 5: Desarrollo del Frontend

Aquí implementaré la interfaz de usuario que habré diseñado en la Fase 3. Es donde HTML, CSS y JavaScript harán que la aplicación se vea bien y sea fácil de usar. También, si es necesario, utilizaré AJAX para lograr que los datos se actualicen sin recargar la página.

- Tecnologías: HTML y CSS para la estructura y estilo de la página; JavaScript para hacerla interactiva.

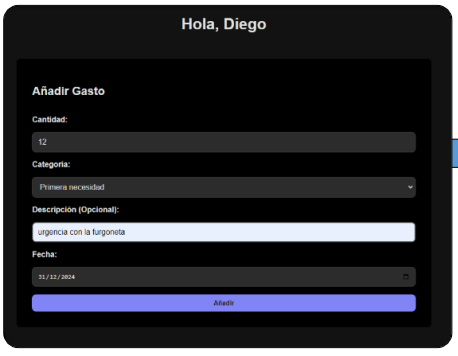




Fase 6: Pruebas de Funcionalidad

Con el backend y el frontend listos, toca asegurarse de que todo funciona correctamente. Probaré cada función, como el login, la introducción de gastos, y la generación de resúmenes mensuales, para asegurar que todo cumple con los requisitos y que la aplicación es estable y funcional.

- Método: Pruebas manuales y tests, con el objetivo de asegurar que los datos se guardan y se muestran correctamente.



Fase 7: Documentación y Entrega

Para finalizar, prepararé una documentación completa del proyecto, donde se explique cómo está construido, cómo se usa, y las decisiones de diseño que he tomado. Esto incluye desde una guía rápida de uso hasta un resumen técnico para futuras referencias.

- Herramientas: Para la documentación, Microsoft Word y formatearé el documento final en un documento PDF.

Detalles técnicos

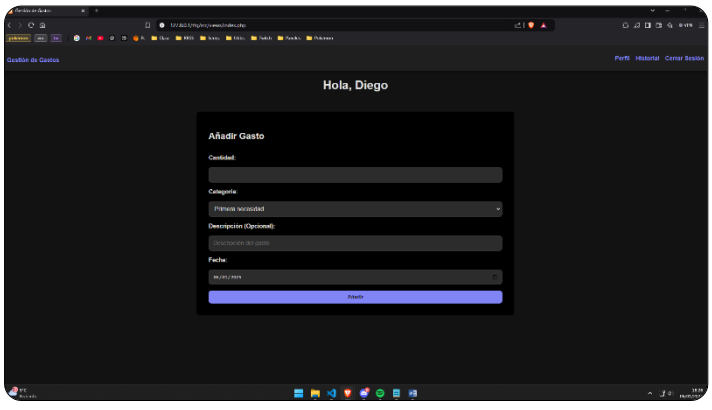
Tecnologías y herramientas

Como ya hemos comentado, las tecnologías del proyecto son HTML, CSS, JavaScript y PHP. A continuación, se detalla el uso específico de cada una:

- HTML: Se utiliza para estructurar la interfaz de usuario, creando la base de las páginas web y facilitando la visualización del contenido. Se puede usar desde un documento PHP, que es lo que hemos hecho en todos los documentos en los que lo usamos.

```php
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Gestión de Gastos</title>
    <link rel="stylesheet" href="../assets/css/styles.css">
    <script src="../assets/JS/script.js" defer></script>
</head>

<body>
    <?php include 'header.php'; ?>

    <h1 class="saludo">Hola, <?= htmlspecialchars(string: $user_name) ?></h1>

    <main>
        <section class="add-expense">
            <form id="expense-form">
                <h2>Añadir Gasto</h2>      You, hace 2 semanas • login and sign up

                <label for="amount">Cantidad:</label>
                <input type="number" id="amount" name="amount" step="any">

                <label for="category">Categoría:</label>
```

- CSS: Se usa para diseñar la apariencia visual de la aplicación, garantizando que sea atractiva y accesible. En particular, se aplican principios de diseño responsivo para asegurar que la aplicación sea funcional tanto en dispositivos móviles como en ordenadores de escritorio.




```css
/* Estilos Generales */
body {
    font-family: 'Arial', sans-serif;
    height: 100%;
    margin: 0;
    padding: 0;
    background-color: #121212;
    color: #e0e0e0;
}
```

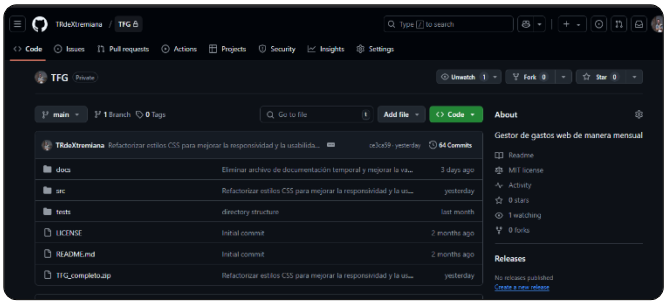
- JavaScript: Este lenguaje se emplea para agregar interactividad a la aplicación, como la validación de formularios, actualizaciones dinámicas de contenido sin recargar la página mediante AJAX, y manipulación del DOM para una experiencia de usuario fluida. He procurado modularlo y seccionarlo mucho para poder reciclar todo lo posible y evitar la repetición de código. Esto facilita la reutilización de funciones y bloques de código, mejorando la mantenibilidad y la escalabilidad del proyecto.

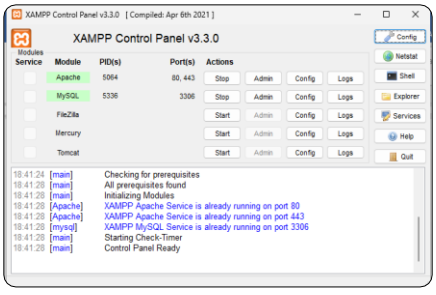
```
You, ayer | 2 authors (You and one other)
// Subfunción para manejar la configuración y ejecución de peticiones
async function postData(url, data) {
    return await fetch(url, {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify(data),
    });
}
```

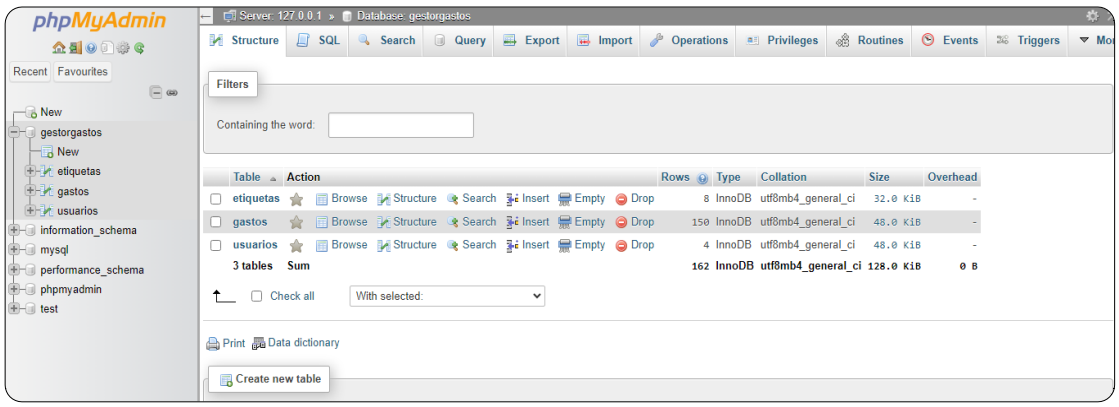
- PHP: Se utiliza para la lógica del backend, permitiendo gestionar la base de datos, manejar el registro e inicio de sesión de los usuarios, procesar los datos de los gastos y generar resúmenes mensuales.

```
<?php
session_start();

require_once __DIR__ . '/../backend/models/Database.php';

if (!isset($_SESSION['user_id'])) {
    header(header: 'Location: login.php');
    exit();
}

$user_name = $_SESSION['user_name'] ?? '';
?>
```

Adicionalmente, utilizo herramientas como GitHub para el control de versiones, XAMPP como servidor local, y MySQL para la gestión de la base de datos.



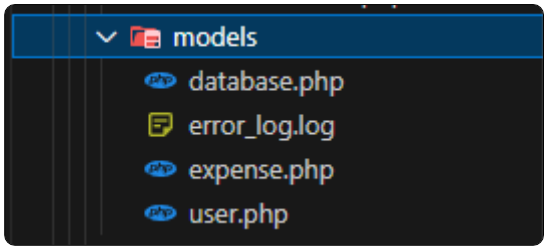




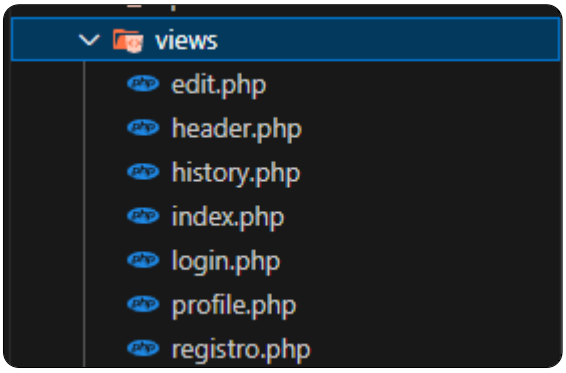
Desarrollo de la arquitectura

He elegido una arquitectura de Modelo-Vista-Controlador (MVC) porque permite una clara separación entre la lógica de negocio (modelo), la interfaz de usuario (vista) y la gestión de las interacciones del usuario (controlador). Esta separación mejora la mantenibilidad y escalabilidad del código, facilitando futuras modificaciones y ampliaciones del sistema.

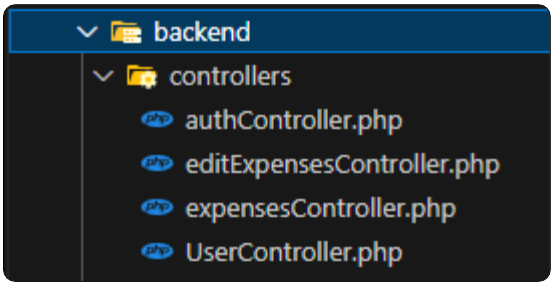
- Modelo: El modelo es responsable de la interacción con la base de datos, gestionando las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) de los usuarios y los gastos.



- Vista: La vista es la interfaz con la que el usuario interactúa. Está compuesta por las páginas web diseñadas con HTML, CSS y JavaScript, que permiten al usuario visualizar y manipular los datos.



- Controlador: El controlador actúa como intermediario entre el modelo y la vista. Recibe las solicitudes del usuario desde la vista, llama al modelo para procesar la lógica necesaria (como insertar o modificar gastos), y luego actualiza la vista con los resultados.

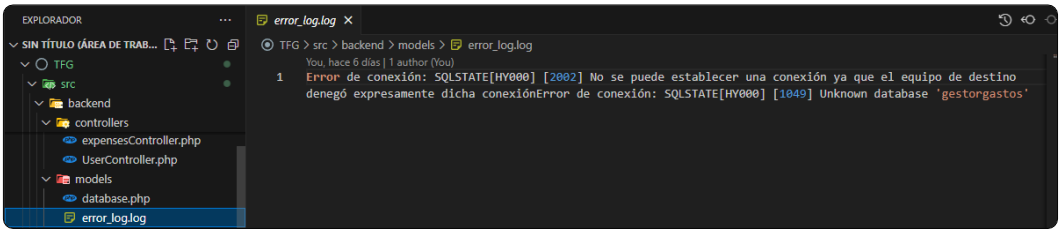


Este enfoque permite modificar y actualizar las diferentes partes de la aplicación sin afectar al resto del sistema, lo que facilita la colaboración en equipo y mejora la organización del código.

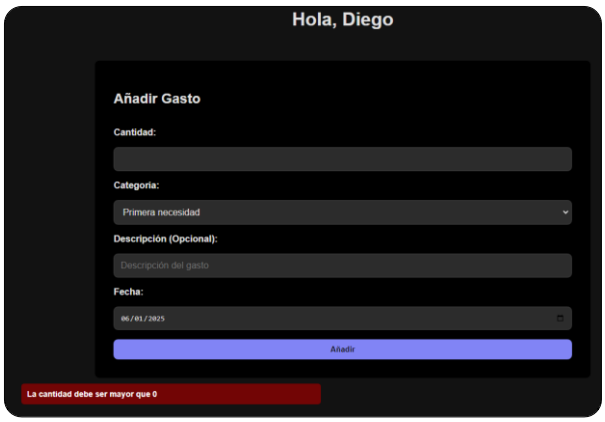
Manejo de errores

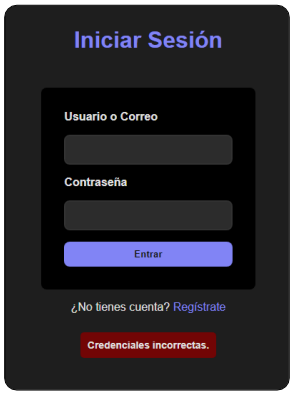
El manejo de errores se gestiona, en su mayoría, desde el servidor, es decir, en PHP. Para asegurar que la aplicación sea robusta y resistente a fallos, se implementan varios mecanismos de manejo de errores:

- Errores de base de datos: En caso de que se produzca un error durante la conexión o la consulta de la base de datos, se capturan las excepciones y se registran en un archivo de logs. Además, se muestra un mensaje genérico al usuario para evitar que se filtren detalles sensibles del sistema.



- Errores de validación: Para garantizar la integridad de los datos que se ingresan, se realizan validaciones tanto del lado del cliente (con JavaScript) como del lado del servidor (con PHP). Si el usuario intenta enviar datos incorrectos o incompletos, la aplicación mostrará un mensaje de error informativo.





- Manejo de excepciones: En todo el código PHP, se emplea la instrucción try-catch para gestionar excepciones. Esto permite capturar errores inesperados y realizar acciones correctivas, como redirigir al usuario a una página de error o realizar un reinicio de la sesión.

```php
try {
    // Insertar el nuevo usuario
    $query = $pdo->prepare(query: "INSERT INTO usuarios (nombre_usuario, correo, contrasena) VALUES
    (:nombre_usuario, :correo, :contrasena)");
    $query->execute(params: [
        'nombre_usuario' => $nombre_usuario,
        'correo' => $correo,
        'contrasena' => password_hash(password: $contrasena, algo: PASSWORD_BCRYPT),
    ]);
} catch (PDOException $e) {
    // Manejo de errores de base de datos
    if ($e->getCode() === '23000') {
        return "El correo electrónico o el nombre de usuario ya están registrados.";
    }
    return "Ocurrió un error al registrar el usuario: " . $e->getMessage();
}
```

Este enfoque integral garantiza que la aplicación sea segura, funcional y fácil de mantener a largo plazo.

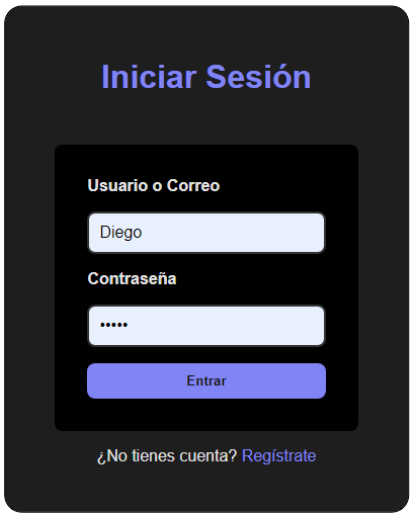
Desarrollo del frontend y el backend

Funcionalidades clave

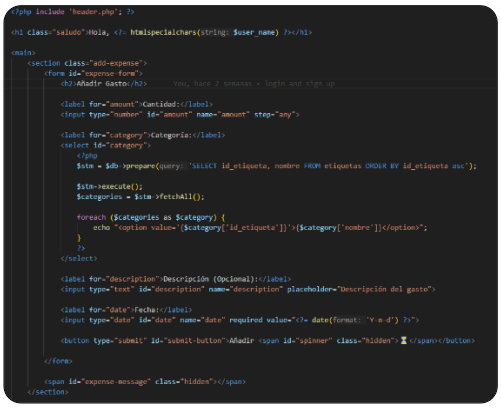
El desarrollo del frontend y el backend de la aplicación está orientado a la implementación de funcionalidades esenciales para una correcta gestión de los gastos. Estas son las principales funcionalidades clave que hemos desarrollado:

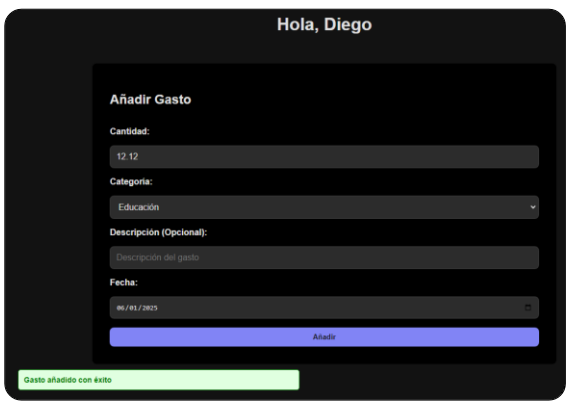
- Registro e inicio de sesión de usuarios: El frontend proporciona formularios interactivos para que los usuarios puedan registrarse e iniciar sesión, mientras que el backend valida los datos y gestiona las sesiones de los usuarios.

```php
<body>
    <div class="login-container">
        <h1 class="centrado">Iniciar Sesión</h1>

        <form action="login.php" method="POST">
            <label for="username">Usuario o Correo</label>
            <input type="text" id="username" name="username">

            <label for="password">Contraseña</label>
            <input type="password" id="password" name="password">

            <button type="submit">Entrar</button>
        </form>

        <p>¿No tienes cuenta? <a href="./registro.php">Regístrate</a></p>

        <?php if ($error): ?>
            <p class="error"><?= htmlspecialchars(string: $error) ?></p>
        <?php endif; ?>
    </div>
</body>
```



- Gestión de gastos: Los usuarios pueden añadir, editar y eliminar sus gastos a través de una interfaz sencilla. El backend gestiona el almacenamiento y recuperación de estos datos en la base de datos.



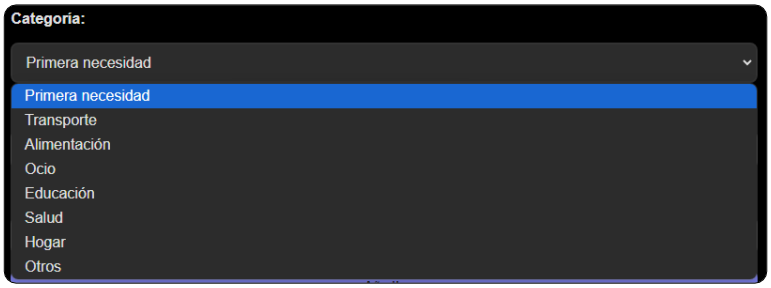


←T→	id_gasto	id_usuario	id_etiqueta	cantidad	descripcion	fecha_gasto	fecha_creacion 1	fecha_actualizacion	eliminado
Edit Copy Delete	166	1	5	12.12	NULL	2025-01-06	2025-01-06 19:34:00	2025-01-06 19:34:00	0
							2025-01-05		

- Categorías de gastos: Los usuarios pueden clasificar sus gastos en categorías como "ocio", "primera necesidad", etc. Esta categorización se gestiona desde el backend y se muestra de forma clara en la interfaz.

	id_etiqueta	nombre	color	fecha_creacion
Edit Copy Delete	1	Primera necesidad	#FF5733	2024-12-31 11:19:23
Edit Copy Delete	2	Transporte	#33B5E5	2024-12-31 11:19:23
Edit Copy Delete	3	Alimentación	#FF33A1	2024-12-31 11:19:23
Edit Copy Delete	4	Ocio	#33FF57	2024-12-31 11:19:23
Edit Copy Delete	5	Educación	#FFC300	2024-12-31 11:19:23
Edit Copy Delete	6	Salud	#FF3333	2024-12-31 11:19:23
Edit Copy Delete	7	Hogar	#3366FF	2024-12-31 11:19:23
Edit Copy Delete	8	Otros	#999999	2024-12-31 11:19:23

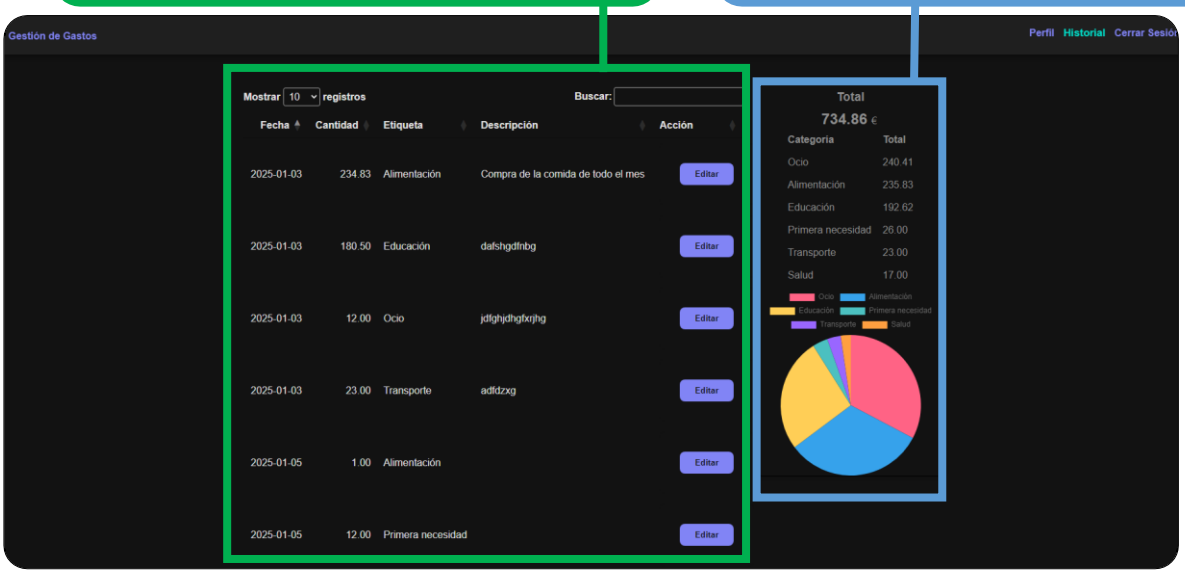
```php
<label for="category">Categoría:</label>
<select id="category">
    <?php
    $stm = $db->prepare(query: 'SELECT id_etiqueta, nombre FROM etiquetas ORDER BY id_etiqueta asc');

    $stm->execute();
    $categories = $stm->fetchAll();

    foreach ($categories as $category) {
        echo "<option value='{$category['id_etiqueta']}'>{$category['nombre']}</option>";
    }
    ?>
</select>
```



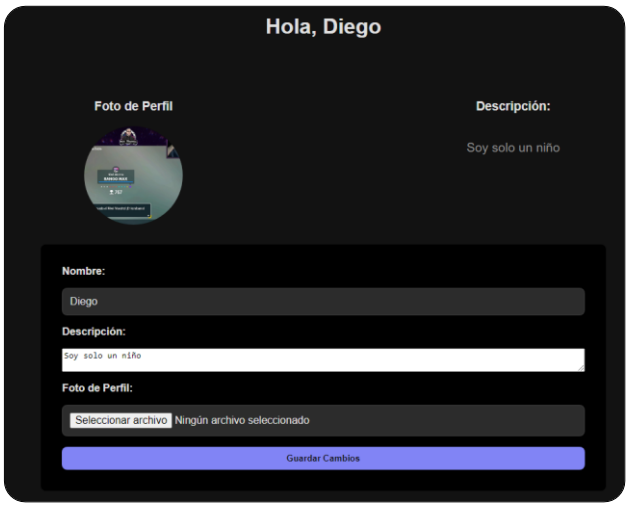
- Resúmenes mensuales: Los usuarios pueden ver un resumen de sus gastos por categorías, con gráficos y estadísticas. El backend se encarga de procesar los datos y generar los informes necesarios.

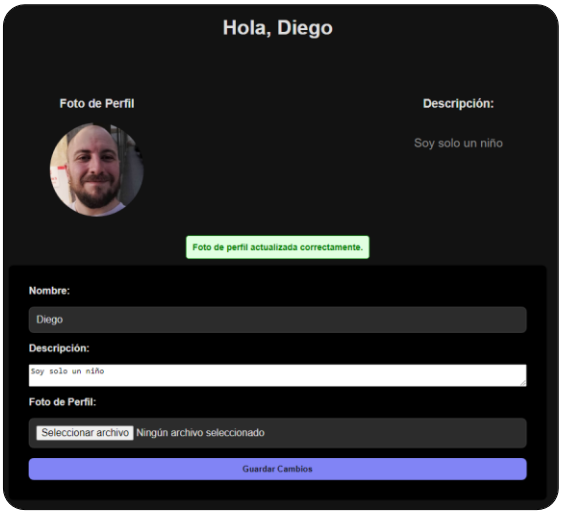
```php
<script>
let table = new DataTable('#tablaGastos', {
    responsive: true,
    pageLength: 10,
    language: {
        decimal: ',',
        thousands: '.',
        info: 'Mostrando _START_ a _END_ de _TOTAL_ registros',
        infoEmpty: 'Mostrando 0 a 0 de 0 registros',
        infoFiltered: '(filtrado de _MAX_ registros totales)',
        lengthMenu: 'Mostrar _MENU_ registros',
        loadingRecords: 'Cargando...',
        processing: 'Procesando...',
        search: 'Buscar:',
        zeroRecords: 'No se encontraron resultados',
        paginate: {
            first: 'Primero',
            last: 'Último',
            next: 'Siguiente',
            previous: 'Anterior',
        },
        aria: {
            sortAscending: ': activar para ordenar la columna ascendente',
            sortDescending: ': activar para ordenar la columna descendente',
        },
    },
});
</script>


</div>
<?php else : ?>
```

```php
<script>
let etiquetas = <?= json_encode(value: array_column(array: $totalesPorCategoria, column_key: 'categoria')) ?>;
let totales = <?= json_encode(value: array_column(array: $totalesPorCategoria, column_key: 'total_categoria')) ?>

document.addEventListener('DOMContentLoaded', function() {
    const ctx = document.getElementById('graficoCircular').getContext('2d');
    new Chart(ctx, {
        type: 'pie',
        data: {
            labels: etiquetas,
            datasets: [{
                label: 'Distribución de gastos por categoría',
                data: totales,
                backgroundColor: [
                    'rgb(255, 99, 133)',
                    'rgba(54, 162, 235)',
                    'rgba(255, 206, 86)',
                    'rgba(75, 192, 192)',
                    'rgba(153, 102, 255)',
                    'rgba(255, 159, 64)'
                ],
                borderWidth: 0
            }]
        },
        options: {
            responsive: true,
            plugins: {
                legend: {
                    position: 'top',
                },
                tooltip: {
                    callbacks: {
                        label: function(context) {
                            return `${context.label}: ${context.raw} €`;
                        }
                    }
                }
            }
        }
    });
});
</script>
```



- Interacción en tiempo real: A través de JavaScript y AJAX, el frontend permite actualizar la interfaz sin recargar la página, haciendo la experiencia más fluida y dinámica.





Integración entre el backend y el frontend

La integración entre el frontend y el backend se realiza a través de peticiones HTTP que el frontend envía al servidor para procesar los datos en el backend. El proceso sigue estos pasos:

- Envío de datos desde el frontend: Cuando el usuario realiza una acción (por ejemplo, añadir un gasto o hacer login), el frontend envía una solicitud HTTP al backend utilizando AJAX o un formulario estándar.

- Procesamiento de datos en el backend: El backend, escrito en PHP, recibe la solicitud, valida los datos recibidos y realiza la acción correspondiente (insertar en la base de datos, autenticar usuario, etc.).

- Respuesta del servidor: El servidor envía una respuesta al frontend, ya sea confirmando la acción (por ejemplo, confirmando que un gasto ha sido añadido) o proporcionando datos necesarios (como los gastos de un mes determinado).

- Actualización dinámica del frontend: Usando JavaScript y AJAX, el frontend actualiza la interfaz sin necesidad de recargar la página. Esto mejora la experiencia del usuario al hacer la aplicación más rápida e interactiva.

El uso de JSON como formato de intercambio de datos entre el frontend y el backend facilita esta integración, permitiendo una estructura flexible y ligera para enviar y recibir datos.

Optimización del rendimiento

Para garantizar un rendimiento adecuado, he decidido implementado varias técnicas tanto en el frontend como en el backend:

Backend:

- Consultas optimizadas: Las consultas SQL se optimizan para asegurar que el sistema puede manejar una gran cantidad de datos sin afectar la velocidad. Esto incluye el uso de índices y consultas eficientes para recuperar y modificar datos rápidamente.

```php
/**
 * Confirmar las credenciales del usuario.
 */
1 reference
function verifyCredentials($identifier, $contrasena): mixed
{
    $pdo = Database::connect();

    $stmt = $pdo->prepare(query: 'SELECT id_usuario, nombre_usuario, correo, contrasena
                                  FROM usuarios
                                  WHERE nombre_usuario = :username OR correo = :correo');
    $stmt->execute(params: [
        ':username' => $identifier,
        ':correo' => $identifier
    ]);

    $usuario = $stmt->fetch();

    // Confirmar si la contraseña coincide
    if ($usuario && password_verify(password: $contrasena, hash: $usuario['contrasena'])) {
        return $usuario;
    }
    return false;
}
```

- Caché: Se utiliza caché en el backend para almacenar temporalmente los datos más utilizados, como los resúmenes mensuales, evitando la necesidad de realizar las mismas consultas a la base de datos repetidamente.

- Compresión de datos: Los datos enviados entre el servidor y el cliente se comprimen para reducir el tiempo de carga y mejorar la velocidad.

Frontend:

- Carga asíncrona de recursos: Los recursos (como imágenes o scripts) se cargan de manera asíncrona para no bloquear la renderización de la página y mejorar la experiencia de usuario.

```javascript
// Función para añadir un gasto
async function addExpense(expenseData) {
    toggleButtonAndSpinner('submit-button', 'spinner', true);
    try {
        const response = await postData("../backend/controllers/expensesController.php", expenseData);
        await handleResponse(response, 'Gasto añadido con éxito');
    } catch (error) {
        showExpenseMessage('Error al conectar con el servidor', false);
    } finally {
        toggleButtonAndSpinner('submit-button', 'spinner', false);
    }
}
```

- Optimización de imágenes: Se utilizan imágenes optimizadas (reducción de tamaño sin pérdida significativa de calidad) para que las páginas se carguen más rápido.

Pruebas de responsividad

La aplicación está diseñada con un enfoque responsivo, optimizada para ofrecer una experiencia fluida en cualquier dispositivo, desde ordenadores de escritorio hasta tabletas y móviles. Este diseño permite a los usuarios gestionar sus finanzas personales desde cualquier lugar y en cualquier momento.

Objetivo del Diseño Responsivo

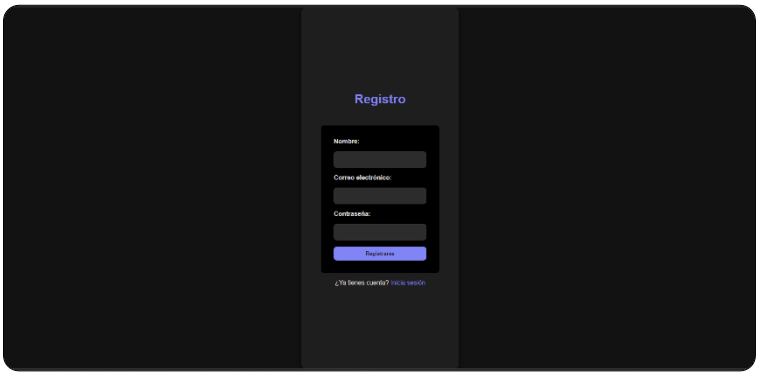
El diseño responsivo garantiza:

- Accesibilidad en cualquier dispositivo: La aplicación adapta su interfaz automáticamente a diferentes tamaños de pantalla.
- Uso práctico desde móviles: Los usuarios pueden introducir gastos y consultar resúmenes en tiempo real desde su móvil, facilitando la gestión financiera en su día a día.

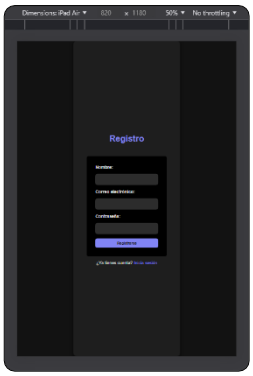
Capturas de Pantalla en Diferentes Dispositivos.

- Pantalla de registro

Ordenador



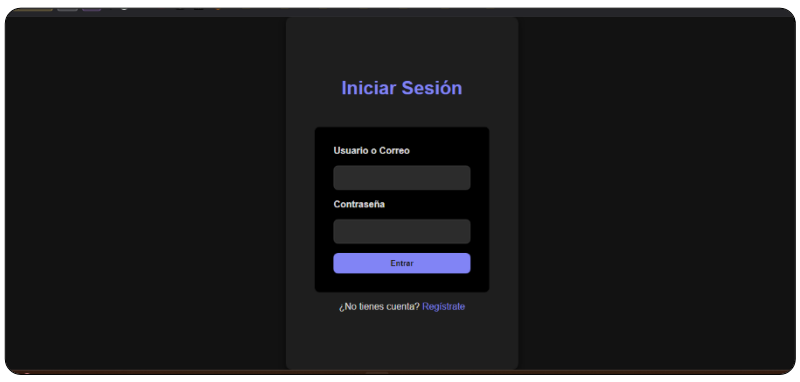
Tablet



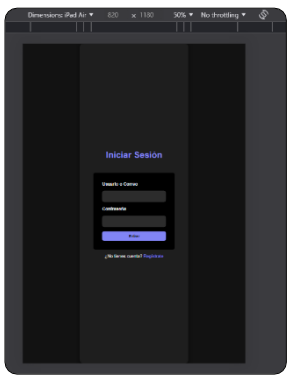
Móvil

- Pantalla de inicio de sesión.

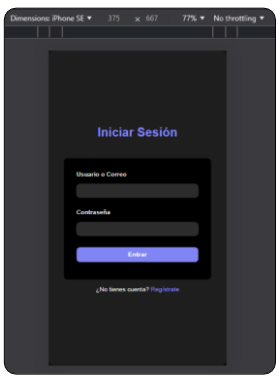
Ordenador



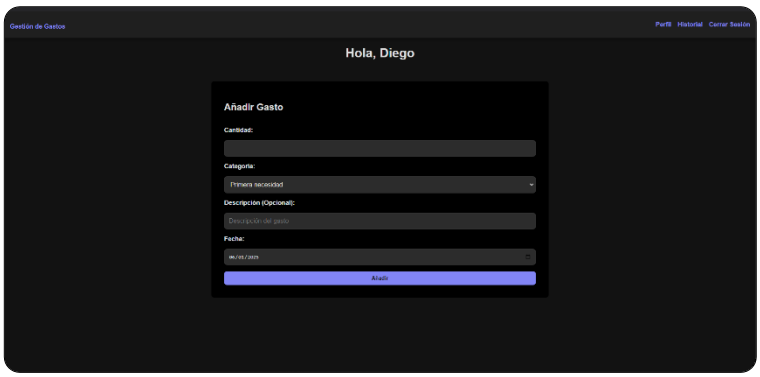
Tablet



Móvil

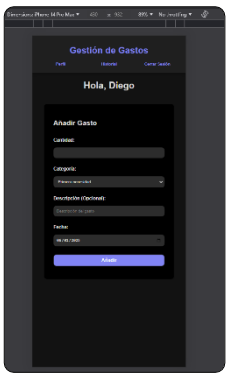


Ordenador



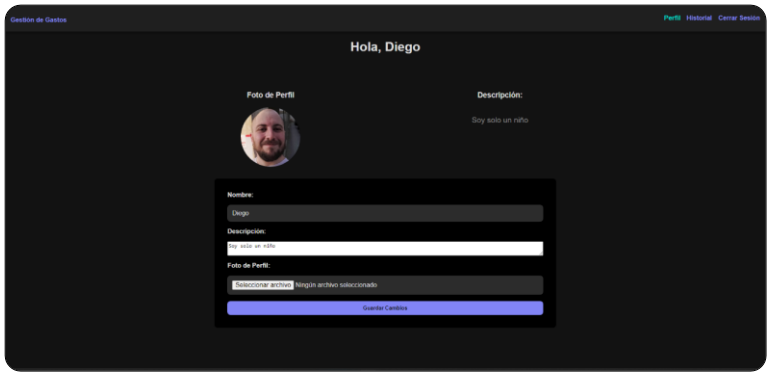
Tablet

Móvil

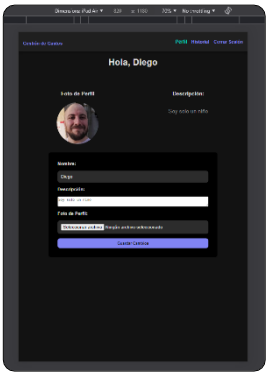


- Perfil.

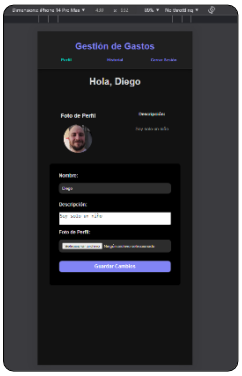
Ordenador



Tablet

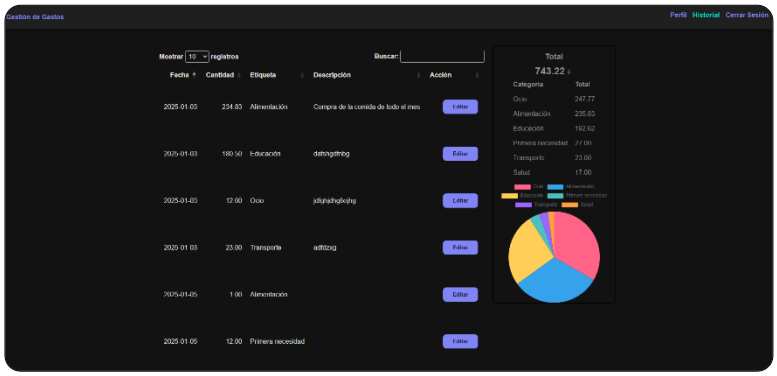


Móvil

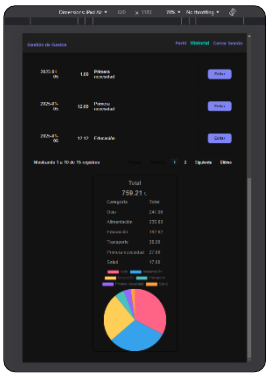


- Resumen mensual con gráficos.

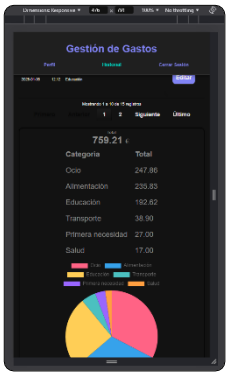
Ordenador



Tablet



Móvil



Características Clave del Diseño Responsivo

- Uso de media queries en CSS: Adaptación automática de elementos como menús, botones y gráficos según el tamaño de la pantalla.
- Diseño Mobile-First: La aplicación prioriza la experiencia móvil, permitiendo a los usuarios acceder fácilmente a sus datos desde cualquier lugar.
- Interactividad optimizada: Botones más grandes y gráficos adaptables para interacción táctil.

Beneficios del Diseño Responsivo

- Gestión financiera en cualquier lugar: Los usuarios no dependen de un ordenador, ya que pueden utilizar la aplicación desde su móvil.

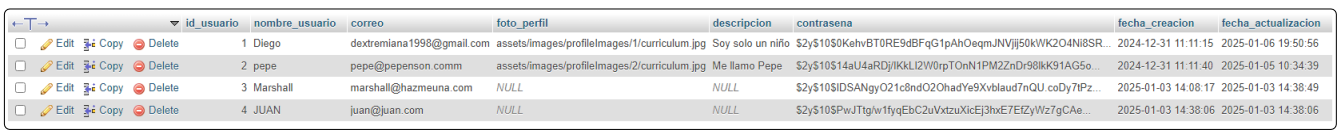
- Experiencia uniforme: Las funcionalidades están disponibles y son fáciles de usar sin importar el dispositivo.

Implementación y medidas de seguridad

La seguridad es una prioridad tanto en el frontend como en el backend para proteger los datos de los usuarios y evitar vulnerabilidades. Algunas de las medidas implementadas son:

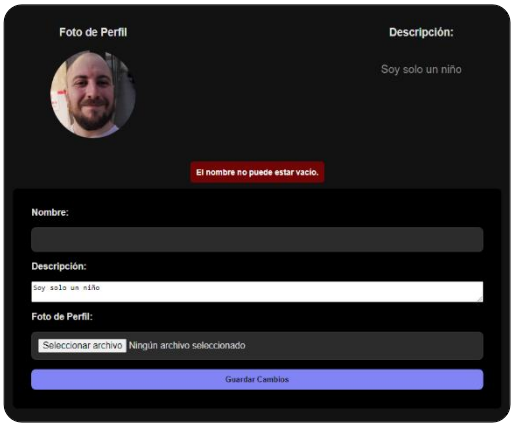
- Autenticación y autorización: El sistema de autenticación utiliza contraseñas cifradas para garantizar que los datos sensibles estén protegidos. Además, las sesiones de los usuarios se gestionan de forma segura, evitando el acceso no autorizado.

```php
try {
    // Insertar el nuevo usuario
    $query = $pdo->prepare(query: "INSERT INTO usuarios (nombre_usuario, correo, contrasena) VALUES (:nombre_usuario, :correo, :contrasena)");
    $query->execute(params: [
        'nombre_usuario' => $nombre_usuario,
        'correo' => $correo,
        'contrasena' => password_hash(password: $contrasena, algo: PASSWORD_BCRYPT),
    ]);
} catch (PDOException $e) {
    // Manejo de errores de base de datos
    if ($e->getCode() === '23000') {
        return "El correo electrónico o el nombre de usuario ya están registrados.";
    }
    return "Ocurrió un error al registrar el usuario: " . $e->getMessage();
}
```



	id_usuario	nombre_usuario	correo	foto_perfil	descripcion	contrasena	fecha_creacion	fecha_actualizacion
Edit Copy Delete	1	Diego	dextremiana1998@gmail.com	assets/images/profileImages/1/curriculum.jpg	Soy solo un niño	$2y$10$0KehvBT0RE9dBFqG1pAhOeqmJNVjlj50kWK2O4Ni8SR...	2024-12-31 11:11:15	2025-01-06 19:50:56
Edit Copy Delete	2	pepe	pepe@pepenson.comm	assets/images/profileImages/2/curriculum.jpg	Me llamo Pepe	$2y$10$14aU4aRDj/lKkLI2W0rpTOnN1PM2ZnDr98lkK91AG5o...	2024-12-31 11:11:40	2025-01-05 10:34:39
Edit Copy Delete	3	Marshall	marshall@hazmeuna.com	NULL	NULL	$2y$10$lDSANgyO21c8ndO2OhadYe9Xvblaud7nQU.coDy7tPz...	2025-01-03 14:08:17	2025-01-03 14:38:49
Edit Copy Delete	4	JUAN	juan@juan.com	NULL	NULL	$2y$10$PwJTtg/w1fyqEbC2uVxtzuXicEj3hxE7EfZyWz7gCAe...	2025-01-03 14:38:06	2025-01-03 14:38:06

- Validación de datos: Tanto en el frontend como en el backend, los datos proporcionados por el usuario se validan rigurosamente. Esto incluye la verificación de campos requeridos, la validación de formatos de correos electrónicos y contraseñas, y la comprobación de que los datos no contengan caracteres maliciosos (prevención de inyección de SQL y XSS).




```php
if (empty($newUserName)) {
    return ['message' => "El nombre no puede estar vacío.", 'class' => 'error'];
}
```

- Tokens personales: Se implementan tokens en los formularios para evitar ataques de falsificación de solicitudes entre sitios.

```php
// Redirigir al inicio si ya estoy logueado
if (isset($_SESSION['user_id'])) {
    header(header: 'Location: index.php');
    exit();
}
```

Con estas medidas, la aplicación asegura un nivel adecuado de protección de los datos y una experiencia de usuario confiable y segura.

Guía rápida: Instalación y Uso de la Aplicación Web

Requisitos previos

1. **Software necesario**:
   - Servidor local (XAMPP, WAMP o similar).
   - Navegador web actualizado.
   - Cliente Git (opcional, si vas a clonar el repositorio directamente).

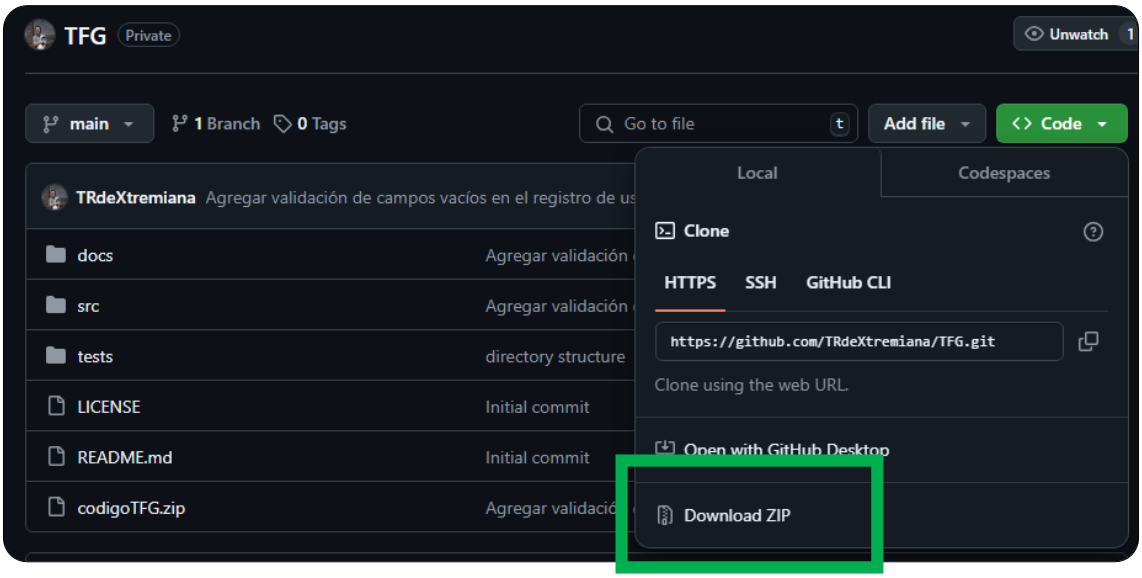
2. **Dependencias**:
   - PHP 7.4 o superior.
   - MySQL (cualquier versión compatible con PHP 7.4 o superior).
   - Conexión a Internet (para clonar el repositorio o descargar archivos).

Pasos de instalación

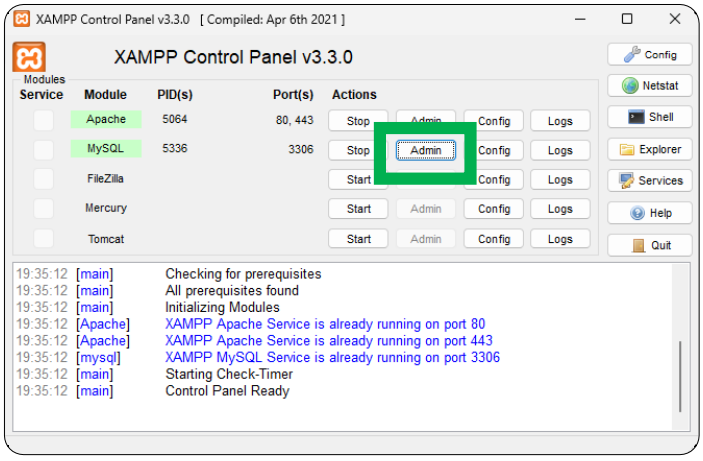
1. Descargar la aplicación:
   - Clonar el repositorio:

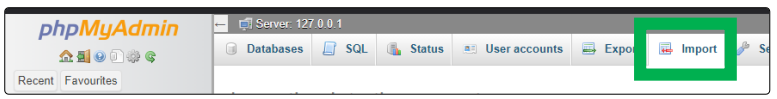
     git clone https://github.com/TRdeXtremiana/TFG.git

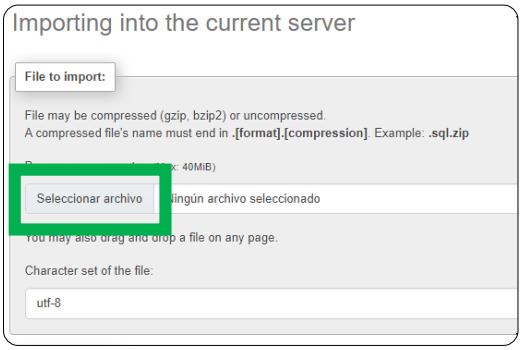
- O descargar el archivo ZIP desde GitHub y extraerlo en tu equipo.

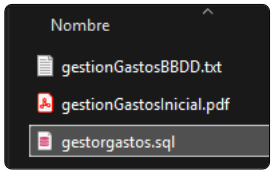


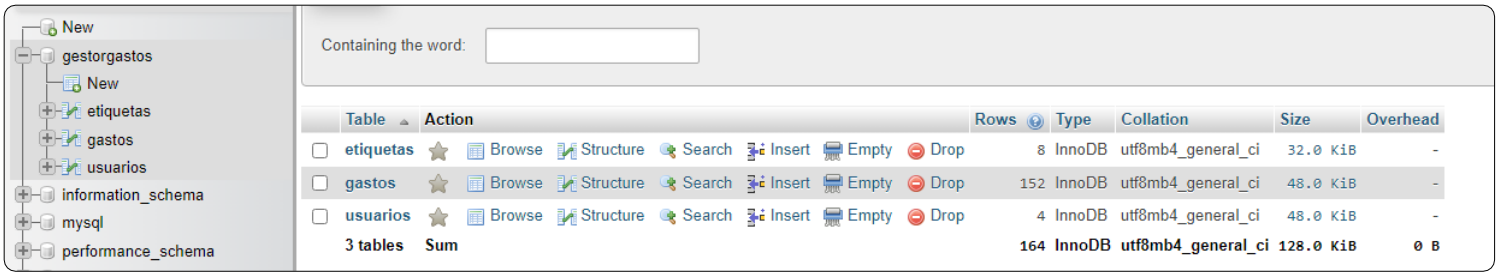
2. Configurar la base de datos:
   - Inicia tu servidor local.
   - Abre phpMyAdmin o cualquier gestor de bases de datos.
   - Crea una base de datos llamada gestorgastos.
   - Importa el archivo “gestorgastos.sql” incluido en el repositorio (en la ruta src/BBDD/) para crear las tablas necesarias.



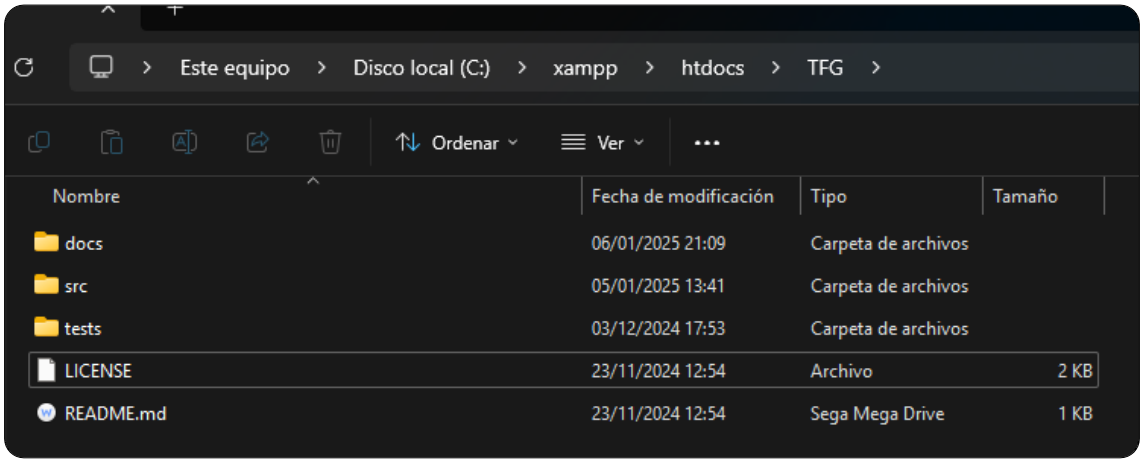








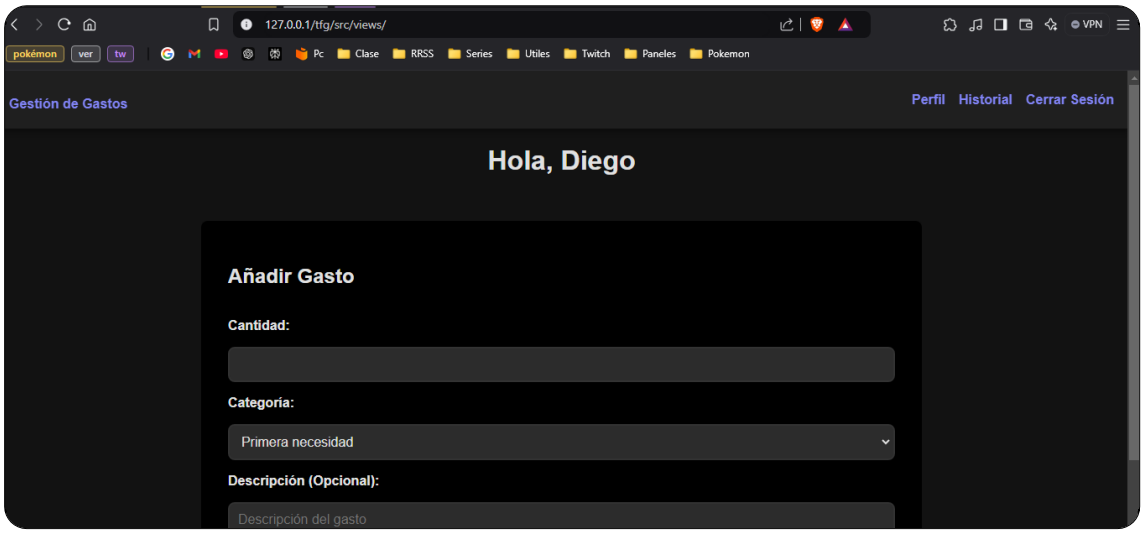
3. Subir los archivos al servidor:
   - Copia la carpeta del proyecto al directorio raíz del servidor local (por ejemplo, htdocs en XAMPP).



4. Acceder a la aplicación:
   - Abre tu navegador y ve a: http://127.0.0.1/src/views/

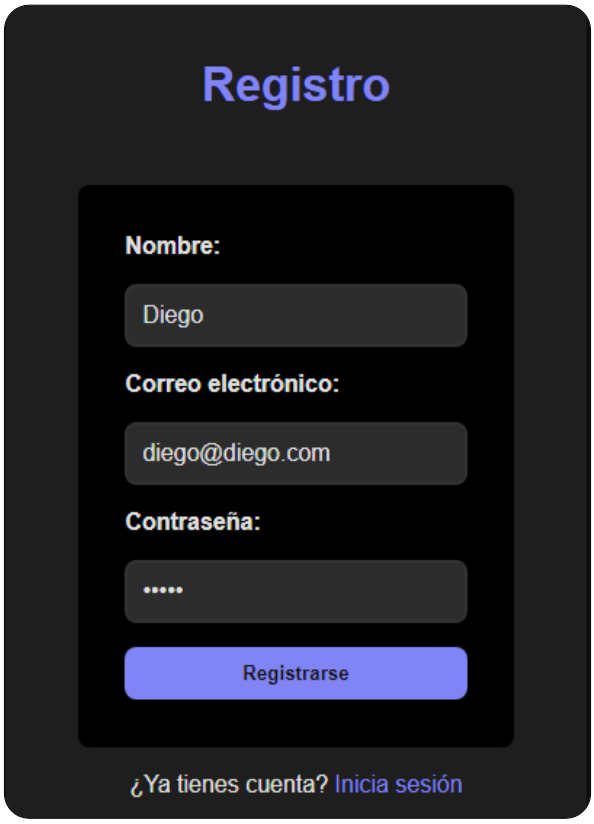
   - Si, por otro lado, lo has introducido en una carpeta dentro del directorio anterior, puedes acceder a : http://127.0.0.1/TuCarpeta/src/views/

En mi caso, la ruta es: http://127.0.0.1/tfg/src/views/ :



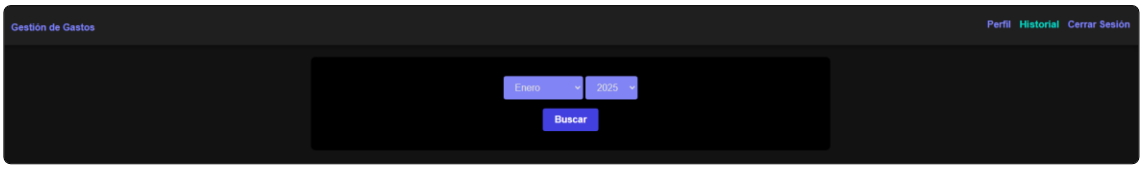
Guía de uso

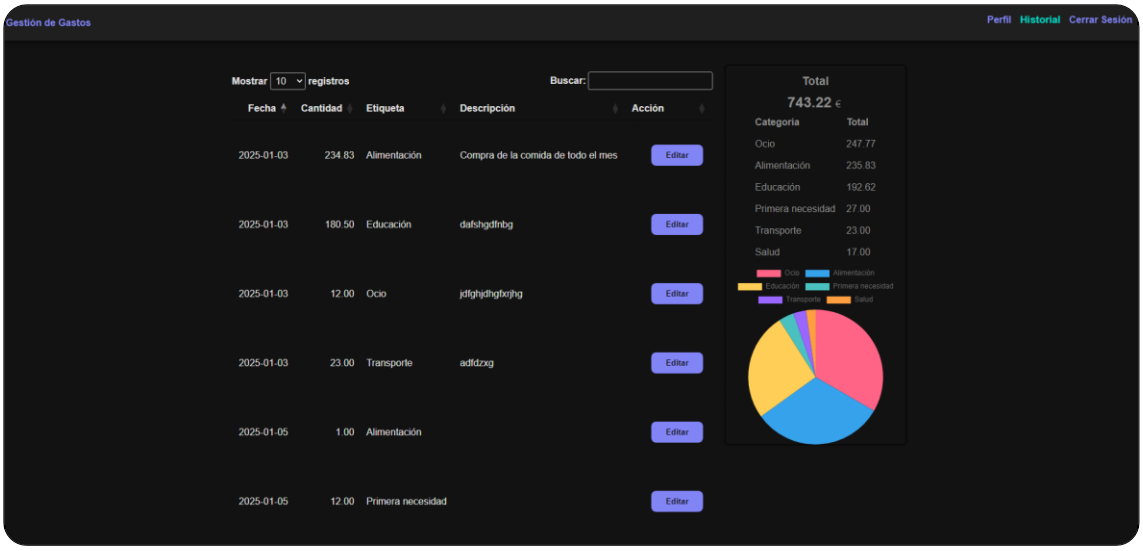
1. Registro e inicio de sesión:
   - Regístrate con tu email y contraseña.
   - Accede con las credenciales creadas.



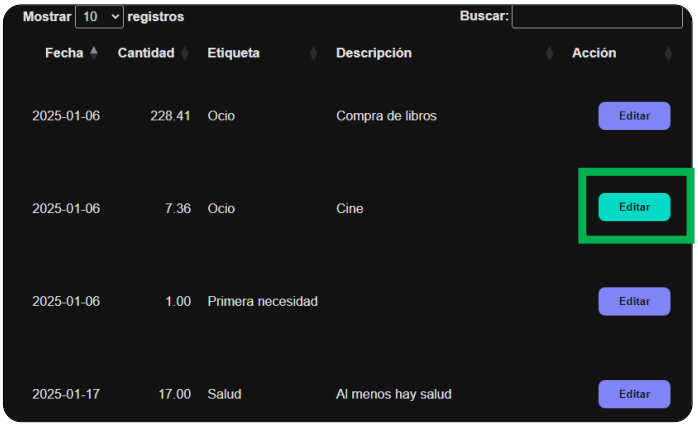
2. Gestión de gastos:
   - Ve a la sección "Añadir gasto" e introduce los detalles.
   - Clasifica los gastos por categoría (por ejemplo, "ocio", "primera necesidad").

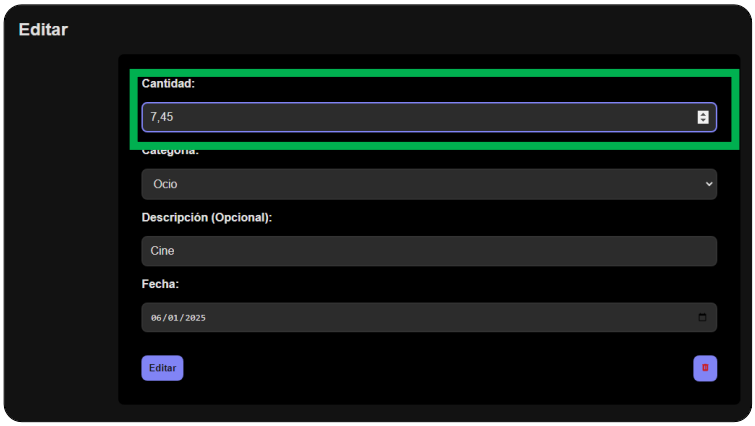
3. Resúmenes mensuales:
   - Accede a la sección de "Historial" y busca el mes en el que estés interesado para visualizar los gráficos y datos de los gastos.

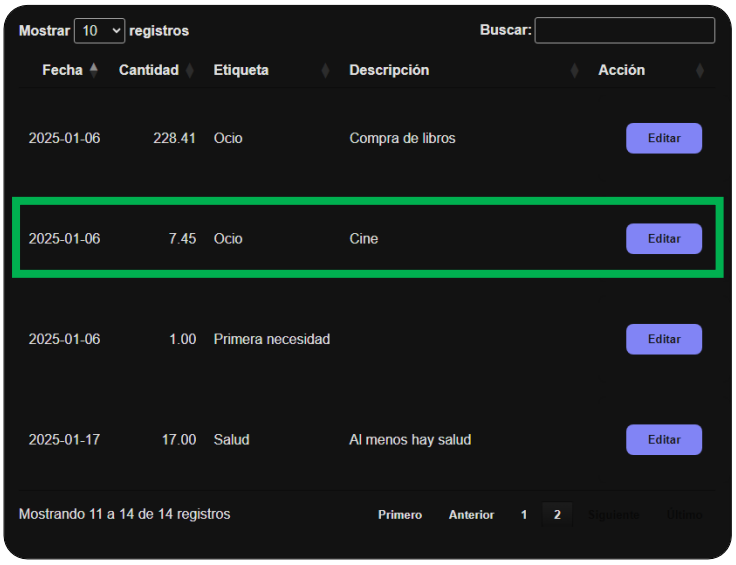




4. Editar o eliminar gastos:
   - Selecciona un gasto en la lista para modificarlo o eliminarlo.







Bibliografía

- W3Schools
  Página de referencia para aprender y consultar ejemplos de HTML, CSS y JavaScript durante el desarrollo del frontend.

- Mozilla Developer Network (MDN)
  Documentación técnica para comprender en profundidad conceptos de desarrollo web, incluyendo JavaScript, CSS y APIs.

- PHP.net
  Fuente principal de referencia para implementar la lógica del backend con PHP, como manejo de sesiones y consultas a la base de datos.

- MySQL Documentation
  Utilizada para diseñar y ejecutar consultas en la base de datos, así como para estructurar las tablas y relaciones de la aplicación.

- GitHub
  Plataforma usada para gestionar el control de versiones del proyecto, mantener un registro de cambios y colaborar con posibles colaboradores.

- dbdiagram.io
  Herramienta para crear y visualizar el diagrama de la base de datos, ayudando a estructurar las tablas y definir sus relaciones.

- Wireframe.cc
  Utilizada para crear prototipos de la interfaz de usuario, como las pantallas de login, introducción de gastos y resúmenes mensuales.

- Stack Overflow
  Comunidad consultada para resolver dudas técnicas, encontrar soluciones a errores comunes y optimizar código.

- ChatGPT
  Asistente para generar ideas, redactar secciones de la documentación y solucionar problemas de desarrollo.

- XAMPP
  Servidor local utilizado para alojar la aplicación durante el desarrollo y pruebas.

- Bootstrap
  Framework usado para agilizar el diseño de la interfaz de usuario con estilos predefinidos y diseño responsivo.

- Canva
  Herramienta empleada para crear diagramas o gráficos que pudieran ser incluidos en la documentación del proyecto.

Desarrollo personal y aprendizajes

Retos enfrentados durante el desarrollo

A lo largo del proyecto, me he enfrentado a varios retos tanto técnicos como personales que me han permitido crecer como desarrollador. Algunos de los principales retos han sido:

- Gestión de la base de datos: Inicialmente, me resultó complicado diseñar una base de datos eficiente, de hecho, este paso siempre dio resultados no esperados y hubo que volver varias veces a rehacer las tablas. Tuve que hacer múltiples iteraciones en el modelo de datos y optimizar consultas, lo que implicó aprender más sobre las relaciones entre tablas y cómo aprovechar las funcionalidades de MySQL.

- Integración del backend con el frontend: Asegurarme de que el backend y el frontend se comunicasen correctamente fue otro reto importante. Tuve que familiarizarme con tecnologías como AJAX y JSON para asegurarme de que las actualizaciones de la interfaz se hicieran de forma eficiente sin recargar la página.

- Seguridad: Implementar medidas de seguridad robustas, como el cifrado de contraseñas y la protección contra ataques CSRF, ha sido una parte fundamental del proyecto. Esto me ha permitido profundizar en el manejo de datos sensibles y aprender sobre las mejores prácticas de seguridad en aplicaciones web.

- Optimización del rendimiento: Mantener el rendimiento de la aplicación, especialmente con un número creciente de datos, fue otro desafío. Implementé soluciones como caché y carga asíncrona para mejorar los tiempos de respuesta y asegurar una experiencia de usuario fluida.

Estos retos me han ayudado a desarrollar habilidades técnicas avanzadas en desarrollo web, bases de datos, y seguridad, y me han permitido enfrentar problemas con un enfoque más estructurado y analítico.

Planes a futuro y proyectos relacionados

A pesar de que este proyecto me ha permitido avanzar significativamente en mi carrera de desarrollo web, tengo planes de especializarme en el futuro en el campo de los videojuegos, que es una de mis pasiones. Como usuario activo de videojuegos, tengo un canal de Twitch donde strimeo a diario, y soy top 1 en Pokémon a nivel competitivo en Pamplona. Además, el año pasado, en septiembre, participé en un torneo en Dortmund, donde quedé 390º de Europa.

Aunque mi objetivo a corto plazo es seguir desarrollando proyectos de aplicaciones web y ampliar mis conocimientos en esta área, tengo la intención de ir incorporando la creación de videojuegos en mis futuros proyectos personales y profesionales. Mi interés en los videojuegos no solo radica en la programación, sino también en la experiencia de usuario, narrativa interactiva, y el diseño de mecánicas de juego.

Planeo comenzar con proyectos más pequeños, como juegos independientes, y a medida que adquiera más experiencia, avanzar hacia proyectos más complejos. A largo plazo, mi objetivo es trabajar en una empresa de desarrollo de videojuegos o incluso crear mi propio estudio de desarrollo de juegos. Sin embargo, tengo claro que, debido a las responsabilidades laborales y profesionales, este cambio de especialización será gradual y lo iré implementando mientras continúo trabajando en el desarrollo de aplicaciones web.

Este enfoque de combinar mi carrera profesional con mis intereses en videojuegos me permitirá seguir aprendiendo y evolucionando como desarrollador, manteniendo una base sólida mientras persigo mi pasión por los videojuegos.