



Universidad Politécnica de Madrid
**Escuela Técnica Superior de Ingenieros
Informáticos**



Grado en Ingeniería Informática

Trabajo de Fin de Grado

**Diseño de la Modificación de
Comprobadores de Procesadores de
Lenguajes, Extracción de Listados
Estadísticos de Gamificación y
Mantenimiento**

Autor: José Manuel Jiménez Martínez

Tutor: José Luis Fuertes Castro

Madrid, Enero 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Diseño de la Modificación de Comprobadores de Procesadores de Lenguajes, Extracción de Listados Estadísticos de Gamificación y Mantenimiento

Enero 2025

Autor: José Manuel Jiménez Martínez

Tutor:

José Luis Fuertes Castro

Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software

ETSI Informáticos

Universidad Politécnica de Madrid

Resumen

Este Trabajo de Fin de Grado (TFG) se centra en la optimización y el mantenimiento de la plataforma web **DRACO**, una herramienta clave en la asignatura de **Procesadores de Lenguajes** de la Escuela Técnica Superior de Ingenieros Informáticos (ETSIINF) de la Universidad Politécnica de Madrid. DRACO permite a los estudiantes entregar prácticas, realizar ejercicios auto evaluables y participar en actividades gamificadas diseñadas para fomentar el aprendizaje y aumentar su motivación.

El desarrollo de este proyecto aborda problemas específicos que afectan el rendimiento y la funcionalidad de la plataforma, entre los que destacan la ausencia de opciones de importación múltiple de actividades, inconsistencias en la gestión de lenguajes y deficiencias en el sistema de corrección de prácticas. Para resolver estos problemas, se han implementado mejoras en los siguientes aspectos:

1. **Importación de actividades:** Se ha añadido la posibilidad de seleccionar múltiples actividades para importarlas de manera simultánea, mejorando la eficiencia del profesorado al gestionar tareas.
2. **Información de actividades:** Se ha ampliado la información visible en las páginas de importación y gestión de actividades, incorporando filtros avanzados y un sistema para mostrar los niveles mínimos requeridos para cada actividad.
3. **Gestión de lenguajes:** Se han corregido errores relacionados con la copia y el borrado de lenguajes, garantizando la persistencia de los datos y evitando configuraciones incompletas.
4. **Corrector de prácticas:** Se ha mejorado el módulo de corrección de la tabla de símbolos para validar con mayor precisión los entregables de los estudiantes, evitando errores frecuentes como la presencia de tablas adicionales o atributos con valores no nulos.
5. **Sistema de niveles:** Se ha establecido un nivel mínimo predeterminado para actividades nuevas, evitando inconsistencias en la asignación de niveles.

El proyecto hace uso de tecnologías web clave como **HTML5**, **CSS3** y **PHP**, junto con prácticas de desarrollo enfocadas en la eficiencia y la usabilidad. Las pruebas realizadas muestran una notable mejora en la experiencia del usuario y en la capacidad del sistema para gestionar grandes volúmenes de información de manera precisa.

La plataforma DRACO, mejorada con estas funcionalidades, no solo optimiza el proceso de enseñanza-aprendizaje, sino que también contribuye a una educación más accesible y dinámica, adaptada a las necesidades actuales de estudiantes y profesores. Con este TFG, se pone de manifiesto la importancia de integrar sistemas de gestión académica más intuitivos y robustos que favorezcan un entorno de aprendizaje enriquecedor.

Abstract

This Final Degree Project (TFG) focuses on the optimization and maintenance of the **DRACO** web platform, a key tool used in the **Language Processors** course at the School of Computer Engineering (ETSIINF) of the Polytechnic University of Madrid. DRACO allows students to submit assignments, complete self-assessment exercises, and participate in gamified activities designed to foster learning and increase their motivation.

The development of this project addresses specific issues that affect the platform's performance and functionality, including the absence of a multi-activity import option, inconsistencies in the management of languages, and deficiencies in the practice correction system. To resolve these issues, improvements have been implemented in the following areas:

1. **Activity import:** The option to select and import multiple activities simultaneously has been added, improving the efficiency of faculty when managing tasks.
2. **Activity information:** The information displayed on the activity import and management pages has been expanded, incorporating advanced filters and a system to show the minimum required levels for each activity.
3. **Language management:** Errors related to language copying and deletion have been corrected, ensuring data persistence and avoiding incomplete configurations.
4. **Practice correction module:** The symbol table correction module has been enhanced to more accurately validate student submissions, preventing common errors such as the presence of additional tables or attributes with non-null values.
5. **Level system:** A default minimum level has been set for new activities, avoiding inconsistencies in level assignments.

The project utilizes key web technologies such as **HTML5**, **CSS3**, and **PHP**, along with development practices focused on efficiency and usability. Tests show a significant improvement in user experience and the system's ability to manage large volumes of information accurately.

The enhanced DRACO platform not only optimizes the teaching-learning process but also contributes to a more accessible and dynamic education, tailored to the current needs of both students and faculty. This TFG highlights the importance of integrating more intuitive and robust academic management systems that foster an enriched learning environment.

Tabla de contenidos

1	Introducción	1
2	Estado de la cuestión	3
2.1	HTML	3
2.1.1	HTML5: Un Cambio Significativo en el Desarrollo Web	3
2.1.2	Soporte Nativo para Contenidos Multimedia	5
2.1.3	API de Interacción Avanzada en HTML5	6
2.1.4	Almacenamiento Local con Web Storage	9
2.1.5	API de Geolocalización	11
2.1.6	Impacto de HTML5 en el Desarrollo Web Actual	13
2.2	CSS	13
2.2.1	La Evolución de CSS	14
2.2.2	Control de Estilos y Consistencia Visual	16
2.2.3	Animaciones y Transformaciones en CSS3	16
2.2.4	Diseño Flexible con <i>Media Queries</i>	16
2.2.5	Diseño Basado en Rejillas (<i>Grid</i>) y <i>Flexbox</i>	17
2.3	PHP	20
2.3.1	Evolución de PHP: Desde PHP/FI hasta <i>PHP 8</i> y Más Allá	21
2.3.2	Generación de Contenido Dinámico y la Ventaja de PHP en Aplicaciones Interactivas	22
2.3.3	Compatibilidad con Bases de Datos y Gestión de Información+ ...	23
2.3.4	<i>Frameworks</i> de PHP: Escalabilidad y Funcionalidades Avanzadas	24
2.3.5	Flexibilidad y Versatilidad en Diferentes Entornos de Desarrollo	24
2.4	DRACO	24
3	Planteamiento del problema	27
3.1	Ausencia de una opción de importar múltiples actividades en la sección de importar actividades	27
3.2	Inclusión de nueva información dentro de la página de actividades e importar actividades	28
3.3	Problema con la definición de lenguajes	30
3.4	Problema con el borrado de lenguajes	31
3.5	Ampliación del funcionamiento del corrector de prácticas para el módulo Tabla de Símbolos	32
3.6	Corrección del nivel mínimo al crear una actividad	32
4	Solución	34
4.1	Solución a la ausencia de una opción de importar múltiples actividades en la sección de importar actividades	34
4.1.1	Diseño	34
4.1.2	Implementación	37

4.1.3	Pruebas	44
4.2	Nueva información dentro de la página de importar actividades y manejar actividades	51
4.2.1	Diseño	51
4.2.2	Implementación	54
4.2.3	Pruebas	61
4.3	Solución al problema con los lenguajes definidos	66
4.3.1	Diseño	66
4.3.2	Implementación	68
4.3.3	Pruebas	71
4.4	Problema con el borrado de lenguajes	73
4.4.1	Diseño	73
4.4.2	Implementación	75
4.4.3	Pruebas	76
4.5	Ampliación del funcionamiento del corrector de prácticas para el módulo Tabla de Símbolos	80
4.5.1	Diseño	80
4.5.2	Implementación	84
4.5.3	Pruebas	92
4.6	Corrección del nivel mínimo al crear una actividad	95
4.6.1	Diseño	95
4.6.2	Implementación	96
4.6.3	Pruebas	98
5	Conclusiones	101
6	Futuras líneas de trabajo	102
7	Impacto	103
8	Bibliografía	104

1 Introducción

El desarrollo y mantenimiento de aplicaciones web han experimentado una evolución significativa en los últimos años, impulsada por la consolidación y avance de tecnologías como HTML, CSS y PHP. Estas herramientas constituyen la base fundamental de numerosos sistemas de gestión de información en diversas áreas, especialmente en el ámbito académico. Su aplicación ha mejorado considerablemente la accesibilidad, interactividad y funcionalidad de las plataformas educativas, facilitando la difusión del conocimiento y optimizando los procesos de enseñanza y aprendizaje. Estas plataformas innovadoras ofrecen recursos tanto para estudiantes como para profesores, enriqueciendo la experiencia educativa.

En este contexto, el presente Trabajo de Fin de Grado (TFG) se centra en la optimización y mejora de la plataforma DRACO, una herramienta clave utilizada en la Escuela Técnica Superior de Ingenieros Informáticos (ETSIINF). DRACO desempeña un papel esencial en la asignatura de Procesadores de Lenguajes, actuando como un recurso vital que facilita la enseñanza y comprensión de conceptos complejos relacionados con el análisis y procesamiento de lenguajes de programación.

La plataforma DRACO ha sido diseñada e implementada con el objetivo de simplificar la autocorrección y gestión de tareas vinculadas al análisis y procesamiento de lenguajes de programación. Proporciona a estudiantes y profesores una herramienta integral que gestiona eficientemente el envío, recepción y evaluación de prácticas y ejercicios. Esto no solo optimiza el flujo de trabajo académico, sino que también mejora la experiencia educativa al ofrecer retroalimentación inmediata y personalizada.

Para lograr una optimización efectiva de DRACO, es fundamental analizar y comprender las tecnologías involucradas en su desarrollo y mantenimiento. A través de un estudio detallado de HTML, CSS y PHP, se pretende establecer la relevancia y el impacto de cada tecnología en el contexto del proyecto. Además, se busca identificar los beneficios y limitaciones de su aplicación en un entorno académico como DRACO, con el fin de proponer mejoras que potencien su funcionalidad y usabilidad.

Asimismo, se realizará un análisis exhaustivo de las funcionalidades actuales de DRACO y de su papel en la enseñanza de los procesadores de lenguajes. Este enfoque permitirá contextualizar la plataforma dentro de un marco técnico y educativo, reflejando tanto los desafíos inherentes a su mantenimiento como las oportunidades de mejora y actualización que se pueden implementar. En particular, se prestará especial atención al sistema de corrección de la tabla de símbolos, que representa un componente técnico esencial y cuyo correcto funcionamiento es crucial para la funcionalidad general de la plataforma.

DRACO es una plataforma web desarrollada por y para el Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software. Está diseñada específicamente para asistir en las asignaturas de Procesadores de Lenguajes y Traductores de Lenguajes, esta última siendo una continuación natural de la primera. La plataforma proporciona un entorno interactivo y práctico donde los estudiantes pueden aplicar los conceptos teóricos aprendidos en clase, facilitando así un aprendizaje más profundo y significativo.

En estas dos asignaturas, la función principal de DRACO es facilitar a los alumnos la corrección de prácticas y ejercicios. La plataforma permite a los profesores proponer y evaluar tareas de forma eficiente, mientras que los estudiantes pueden realizar autoevaluaciones para medir su propio progreso. Además, DRACO incorpora técnicas de gamificación para incrementar la motivación y el compromiso de los estudiantes. Estas técnicas transforman el proceso de aprendizaje en una experiencia más interactiva y entretenida, incentivando a los estudiantes a participar activamente y mejorar su rendimiento académico.

Las técnicas de gamificación utilizadas en DRACO incluyen la asignación de puntos por cada tarea completada, la posibilidad de alcanzar diferentes niveles, la consecución de logros, la creación de un *ranking* entre los estudiantes, etc. Estas características aportan un componente lúdico al proceso educativo, fomentando una competencia sana y la superación personal. Al transformar las actividades académicas en desafíos con recompensas, se promueve un mayor interés y participación por parte de los estudiantes. La gamificación no solo mejora la experiencia de aprendizaje, sino que también contribuye al desarrollo de habilidades como la resolución de problemas, la persistencia y el trabajo en equipo.

Este Trabajo de Fin de Grado, se enfoca en las funcionalidades específicas que DRACO realiza para la asignatura de Procesadores de Lenguajes. Se buscará optimizar y mejorar la plataforma, abordando aspectos técnicos como la implementación y corrección del sistema de la tabla de símbolos, y proponiendo mejoras que potencien su eficacia y usabilidad. El objetivo es desarrollar una herramienta más robusta y eficiente que continúe siendo un recurso valioso para estudiantes y profesores en el proceso de enseñanza y aprendizaje de los procesadores de lenguajes.

En conclusión, este proyecto pretende no solo optimizar una plataforma existente, sino también contribuir al avance de las metodologías educativas en el ámbito de la Informática. Al mejorar DRACO, se espera ofrecer una herramienta más efectiva que facilite el aprendizaje de conceptos complejos, utilizando tecnologías web modernas y técnicas innovadoras como la gamificación. Esto beneficiará a la comunidad académica, proporcionando una experiencia educativa más enriquecedora y alineada con las necesidades actuales de la enseñanza en Ingeniería Informática.

2 Estado de la cuestión

2.1 HTML

HTML, el Lenguaje de Marcado de Hipertexto, fue desarrollado por Tim Berners-Lee en 1989 para estructurar y enlazar documentos de manera accesible a través de la red. Su creación fue el pilar de la *World Wide Web*, permitiendo que documentos previamente independientes pudieran interconectarse y compartirse globalmente. Con el tiempo, HTML evolucionó para responder a las necesidades crecientes de interactividad y contenido multimedia en la web. Cada nueva versión ha ido añadiendo capacidades que han transformado la manera en que las aplicaciones web se desarrollan y presentan contenido a los usuarios, haciendo que la web sea cada vez más accesible y funcional (W3C, 2024).

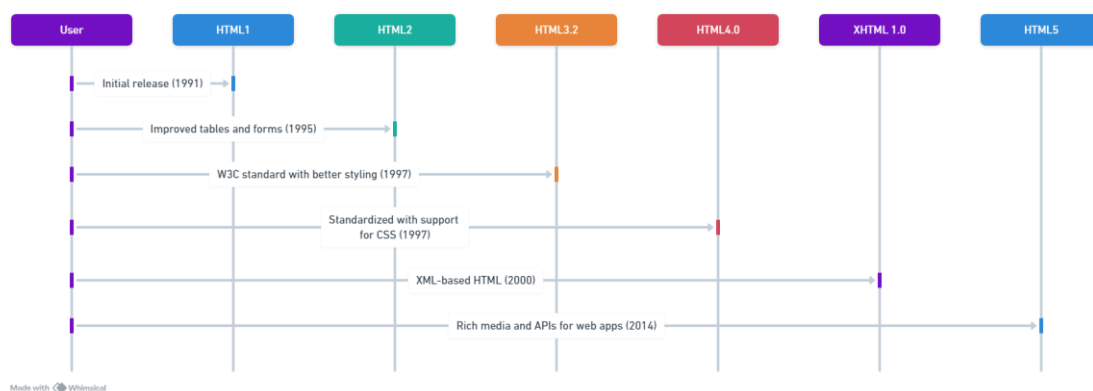


Ilustración 1 Línea Temporal de la evolución de HTML hasta HTML5

En la ilustración 1 puede verse cómo ha ido evolucionando HTML a lo largo de los años, junto con las tecnologías nuevas introducidas más importantes en cada actualización del lenguaje.

2.1.1 HTML5: Un Cambio Significativo en el Desarrollo Web

Uno de los cambios iniciales más destacados de HTML5 fue la introducción de una **sintaxis más flexible**. Esto permitió a los desarrolladores estructurar las páginas de manera semántica, es decir, de una forma en la que los elementos HTML indican claramente la intención de su contenido. Los elementos semánticos como `<header>`, `<footer>`, `<article>`, y `<section>` no solo mejoran la organización y el entendimiento del código por parte de otros desarrolladores, sino que también facilitan la interpretación del contenido por los motores de búsqueda, mejorando la accesibilidad y el *Search Engine Optimization* (SEO) de las páginas web (W3C, 2024).

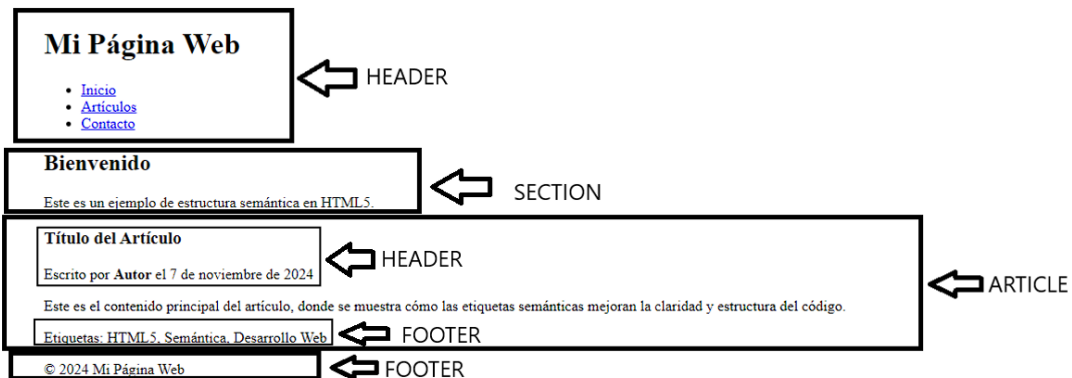


Ilustración 2 Ejemplo de los elementos semánticos mencionados

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Estructura Semántica en HTML5</title>
</head>
<body>
  <header>
    <h1>Mi Página Web</h1>
    <nav>
      <ul>
        <li><a href="#inicio">Inicio</a></li>
        <li><a href="#articulos">Artículos</a></li>
        <li><a href="#contacto">Contacto</a></li>
      </ul>
    </nav>
  </header>

  <section id="inicio">
    <h2>Bienvenido</h2>
    <p>Este es un ejemplo de estructura semántica en HTML5.</p>
  </section>

  <article id="articulos">
    <header>
      <h3>Título del Artículo</h3>
      <p>Escrito por <strong>Autor</strong> el 7 de noviembre de 2024</p>
    </header>
    <p>Este es el contenido principal del artículo, donde se muestra cómo las
    etiquetas semánticas mejoran la claridad y estructura del código.</p>
    <footer>
      <p>Etiquetas: HTML5, Semántica, Desarrollo Web</p>
    </footer>
  </article>

  <footer>
    <p>&copy; 2024 Mi Página Web</p>
  </footer>
</body>
</html>
```

Este código es el código que se ejecuta para crear la página mostrada en la ilustración 2. Contiene el header, section, article y footer que se han mencionado.

2.1.2 Soporte Nativo para Contenidos Multimedia

HTML5 introdujo soporte nativo para contenidos multimedia mediante las etiquetas `<audio>` y `<video>`. Antes de HTML5, para reproducir audio o video en una página web era necesario depender de complementos como Adobe Flash. Esto no solo representaba problemas de compatibilidad, sino que también afectaba la seguridad y el rendimiento de las páginas. Las etiquetas `<audio>` y `<video>` permiten que los navegadores reproduzcan archivos de audio y video directamente, con controles integrados para manipular la reproducción sin necesidad de *plugins* externos, lo que aumenta tanto la seguridad como la interoperabilidad en diversos dispositivos (W3C, 2024).

Ejemplo de Contenido Multimedia en HTML5

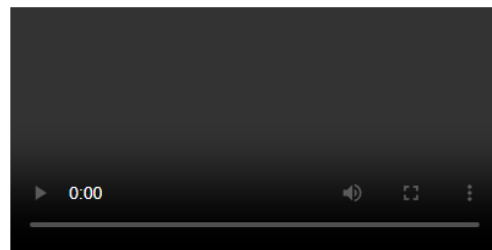
Audio

A continuación, un ejemplo de reproducción de audio:



Video

A continuación, un ejemplo de reproducción de video:



© 2024 Ejemplo de Multimedia en HTML5

Ilustración 3 Ejemplo de audio y video nativo

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Multimedia en HTML5</title>
</head>
<body>
  <header>
    <h1>Ejemplo de Contenido Multimedia en HTML5</h1>
  </header>

  <section>
    <h2>Audio</h2>
    <p>A continuación, un ejemplo de reproducción de audio:</p>
    <audio controls>
      <source src="audio-ejemplo.mp3" type="audio/mpeg">
      Tu navegador no soporta la reproducción de audio.
    </audio>
  </section>
</body>
</html>
```

```

</section>

<section>
  <h2>Video</h2>
  <p>A continuación, un ejemplo de reproducción de video:</p>
  <video controls width="400">
    <source src="video-ejemplo.mp4" type="video/mp4">
    Tu navegador no soporta la reproducción de video.
  </video>
</section>

<footer>
  <p>&copy; 2024 Ejemplo de Multimedia en HTML5</p>
</footer>
</body>
</html>

```

Este código es el utilizado para crear la página que se puede ver en la ilustración 3. La mayoría de los aspectos de este código se han podido ver anteriormente, como `section`, `header` o `footer`. Lo innovador es la sección de video, que incluye un origen del video, además del tipo de video que se verá. Además, hay un texto alternativo en caso de que el video no se pueda reproducir.

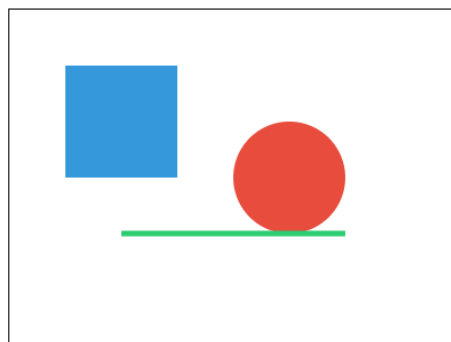
2.1.3 API de Interacción Avanzada en HTML5

HTML5 también trajo consigo una serie de API diseñadas para enriquecer las capacidades de interacción en la web. Entre ellas, la **API de Canvas** es especialmente destacable. Esta API permite el desarrollo de gráficos en 2D, facilitando la creación de visualizaciones dinámicas, animaciones, y juegos. El uso de *Canvas* ha revolucionado la forma en que se presentan gráficos y animaciones en la web, sin depender de bibliotecas externas, y se ha convertido en una herramienta fundamental en aplicaciones de diseño, visualización de datos y videojuegos.

Ejemplo de Canvas en HTML5

Gráfico Dibujado en Canvas

El siguiente ejemplo muestra cómo dibujar gráficos simples con la API de Canvas:



© 2024 Ejemplo de Canvas en HTML5

Ilustración 4 Ejemplo API de Canvas

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

    <title>Ejemplo de API de Canvas en HTML5</title>
</head>
<body>
  <header>
    <h1>Ejemplo de Canvas en HTML5</h1>
  </header>

  <section>
    <h2>Gráfico Dibujado en Canvas</h2>
    <p>El siguiente ejemplo muestra cómo dibujar gráficos simples con la API de
      Canvas:</p>
    <canvas id="miCanvas" width="400" height="300" style="border:1px solid #000;">
    </canvas>
  </section>

  <script>
    // Seleccionar el elemento canvas
    const canvas = document.getElementById("miCanvas");
    const ctx = canvas.getContext("2d");

    // Dibujar un rectángulo
    ctx.fillStyle = "#3498db";
    ctx.fillRect(50, 50, 100, 100);

    // Dibujar un círculo
    ctx.beginPath();
    ctx.arc(250, 150, 50, 0, 2 * Math.PI);
    ctx.fillStyle = "#e74c3c";
    ctx.fill();

    // Dibujar una línea
    ctx.beginPath();
    ctx.moveTo(100, 200);
    ctx.lineTo(300, 200);
    ctx.strokeStyle = "#2ecc71";
    ctx.lineWidth = 5;
    ctx.stroke();
  </script>

  <footer>
    <p>&copy; 2024 Ejemplo de Canvas en HTML5</p>
  </footer>
</body>
</html>

```

En este código se puede ver cómo se crean los dibujos con líneas como `ctx.moveTo(100,200)`, por ejemplo. Esto es una de las funciones con las que se crean las imágenes mostradas en la ilustración 4.

Otra API introducida con HTML5 es la **API de Drag and Drop**, que facilita la funcionalidad de arrastrar y soltar elementos en una página web. Esta característica es útil en aplicaciones interactivas y plataformas de aprendizaje, ya que permite a los usuarios manipular elementos de la interfaz de manera intuitiva. Este tipo de funcionalidades añade interactividad y hace que las aplicaciones sean más atractivas y usables.

Ejemplo de Drag and Drop en HTML5

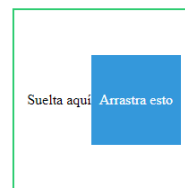
Arrastra el cuadro azul al área de destino:



© 2024 Ejemplo de Drag and Drop en HTML5

Ejemplo de Drag and Drop en HTML5

Arrastra el cuadro azul al área de destino:



© 2024 Ejemplo de Drag and Drop en HTML5

Ilustración 5 Ejemplo de un elemento Drag and drop

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Drag and Drop en HTML5</title>
  <style>
    #dragZone {
      width: 100px;
      height: 100px;
      background-color: #3498db;
      color: #fff;
      display: flex;
      justify-content: center;
      align-items: center;
      cursor: move;
    }
    #dropZone {
      width: 200px;
      height: 200px;
      border: 2px dashed #333;
      display: flex;
      justify-content: center;
      align-items: center;
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <header>
    <h1>Ejemplo de Drag and Drop en HTML5</h1>
  </header>

  <section>
    <p>Arrastra el cuadro azul al área de destino:</p>
    <div id="dragZone" draggable="true" ondragstart="drag(event)">
      Arrastra esto</div>
    <div id="dropZone" ondrop="drop(event)" ondragover="allowDrop(event)">
      Suelta aquí</div>
  </section>

  <script>
    // Permitir el arrastre sobre el área de destino
    function allowDrop(event) {
      event.preventDefault();
    }

    // Identificar el elemento arrastrado
```

```

function drag(event) {
    event.dataTransfer.setData("text", event.target.id);
}

// Manejar la acción de soltar el elemento
function drop(event) {
    event.preventDefault();
    const data = event.dataTransfer.getData("text");
    const draggedElement = document.getElementById(data);
    event.target.appendChild(draggedElement);
    event.target.style.border = "2px solid #2ecc71";//Cambia estilo al soltar
}
</script>

<footer>
    <p>&copy; 2024 Ejemplo de Drag and Drop en HTML5</p>
</footer>
</body>
</html>

```

Este código es el utilizado para la ilustración 5. En él se identifican aspectos estéticos para la parte protagonista, el *drag and drop*. También se pueden ver las funciones que hay para ello en el `<script>` del código, que son partes de la API.

2.1.4 Almacenamiento Local con Web Storage

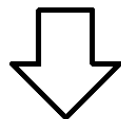
La capacidad de almacenamiento local, introducida a través de la API de Web Storage en HTML5, es una de las innovaciones más prácticas para los desarrolladores de aplicaciones web. Esta API ofrece dos métodos: `localStorage` y `sessionStorage`. El método `localStorage` permite almacenar datos de usuario en el navegador de manera persistente, incluso después de cerrar la ventana o el navegador, hasta que estos sean eliminados manualmente. Esto es especialmente útil para aplicaciones que necesitan retener información del usuario entre sesiones, sin requerir un almacenamiento en el servidor, lo cual puede optimizar el rendimiento y reducir el tiempo de carga.

Ejemplo de Web Storage en HTML5

Introduce tu nombre y elige el tipo de almacenamiento:

Nombre guardado en localStorage.

© 2024 Ejemplo de Web Storage en HTML5



Ejemplo de Web Storage en HTML5

Introduce tu nombre y elige el tipo de almacenamiento:

Nombre en localStorage: Jose

© 2024 Ejemplo de Web Storage en HTML5

Ilustración 6 Ejemplos Storage HTML

Por otro lado, `sessionStorage` guarda los datos solo durante la sesión de navegación, eliminándose automáticamente cuando el usuario cierra la pestaña o el navegador. Esta distinción permite a los desarrolladores elegir la opción más adecuada según el tipo de datos que manejan y el nivel de persistencia requerido (W3C, 2024).

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Web Storage en HTML5</title>
</head>
<body>
  <header>
    <h1>Ejemplo de Web Storage en HTML5</h1>
  </header>

  <section>
    <p>Introduce tu nombre y elige el tipo de almacenamiento:</p>
    <input type="text" id="nombre" placeholder="Tu nombre">
    <button onclick="guardarEnLocal()">Guardar en localStorage</button>
    <button onclick="guardarEnSesion()">Guardar en sessionStorage</button>
    <button onclick="mostrarNombre()">Mostrar Nombre Guardado</button>
    <button onclick="borrarDatos()">Borrar Datos</button>

    <p id="mensaje"></p>
  </section>

  <script>
    // Guardar el nombre en localStorage
    function guardarEnLocal() {
      const nombre = document.getElementById("nombre").value;
      localStorage.setItem("nombreUsuario", nombre);
      document.getElementById("mensaje").innerText =
        "Nombre guardado en localStorage.";
    }

    // Guardar el nombre en sessionStorage
    function guardarEnSesion() {
      const nombre = document.getElementById("nombre").value;
      sessionStorage.setItem("nombreUsuario", nombre);
      document.getElementById("mensaje").innerText =
        "Nombre guardado en sessionStorage.";
    }

    // Mostrar el nombre guardado en cualquiera de los almacenamientos
    function mostrarNombre() {
      const nombreLocal = localStorage.getItem("nombreUsuario");
      const nombreSesion = sessionStorage.getItem("nombreUsuario");
      let mensaje = "No hay datos guardados.";

      if (nombreLocal) {
        mensaje = "Nombre en localStorage: " + nombreLocal;
      } else if (nombreSesion) {
        mensaje = "Nombre en sessionStorage: " + nombreSesion;
      }

      document.getElementById("mensaje").innerText = mensaje;
    }

    // Borrar los datos de ambos almacenamientos
```

```

function borrarDatos() {
    localStorage.removeItem("nombreUsuario");
    sessionStorage.removeItem("nombreUsuario");
    document.getElementById("mensaje").innerText = "Datos borrados.";
}
</script>

<footer>
    <p>&copy; 2024 Ejemplo de Web Storage en HTML5</p>
</footer>
</body>
</html>

```

El código mostrado es tal que crea la página de la ilustración 6. Se debe poner especial atención a las funciones provenientes de la API mencionada, con funciones como `guardarEnSesion()`, entre otras muchas

2.1.5 API de Geolocalización

La **API de Geolocalización** es otro avance significativo en HTML5, permitiendo que las aplicaciones accedan a la ubicación del usuario, con su consentimiento, para personalizar la experiencia de navegación. Esta funcionalidad ha revolucionado aplicaciones comerciales, de navegación y de contenido, adaptando la información en función de la ubicación geográfica. Por ejemplo, servicios de entrega y aplicaciones de comercio electrónico pueden ofrecer productos específicos de una región, mientras que aplicaciones de mapas pueden calcular rutas precisas basadas en la ubicación actual del usuario.

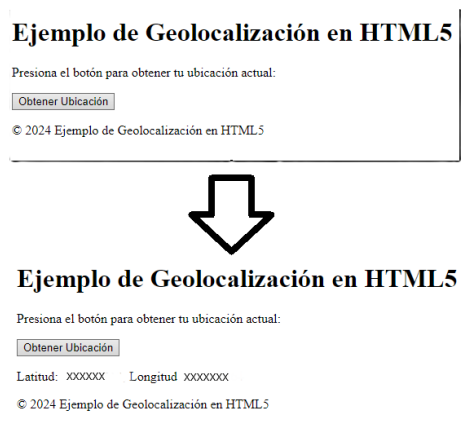


Ilustración 7 Ejemplo Geolocalización HTML

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ejemplo de Geolocalización en HTML5</title>
</head>
<body>
    <header>
        <h1>Ejemplo de Geolocalización en HTML5</h1>
    </header>

    <section>
        <p>Presiona el botón para obtener tu ubicación actual:</p>
        <button onclick="obtenerUbicacion()">Obtener Ubicación</button>

```

```

    <p id="ubicacion"></p>
</section>

<script>
    function obtenerUbicacion() {
        // Comprobar si la geolocalización es compatible con el navegador
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(mostrarUbicacion,
                mostrarError);
        } else {
            document.getElementById("ubicacion").innerText =
                "La geolocalización no es compatible con este navegador.";
        }
    }

    // Función para mostrar las coordenadas
    function mostrarUbicacion(position) {
        const latitud = position.coords.latitude;
        const longitud = position.coords.longitude;
        document.getElementById("ubicacion").innerText =
            `Latitud: ${latitud}, Longitud: ${longitud}`;
    }

    // Función para manejar errores
    function mostrarError(error) {
        switch(error.code) {
            case error.PERMISSION_DENIED:
                document.getElementById("ubicacion").innerText =
                    "Permiso denegado para acceder a la ubicación.";
                break;
            case error.POSITION_UNAVAILABLE:
                document.getElementById("ubicacion").innerText =
                    "Información de ubicación no disponible.";
                break;
            case error.TIMEOUT:
                document.getElementById("ubicacion").innerText =
                    "La solicitud para obtener la ubicación ha expirado.";
                break;
            case error.UNKNOWN_ERROR:
                document.getElementById("ubicacion").innerText =
                    "Error desconocido al obtener la ubicación.";
                break;
        }
    }
}
</script>

<footer>
    <p>&copy; 2024 Ejemplo de Geolocalización en HTML5</p>
</footer>
</body>
</html>

```

El código corresponde a la ilustración 7. Utiliza la API de Geolocalización de HTML5 para obtener la ubicación actual del usuario. Al presionar el botón "Obtener Ubicación", se llama a la función `obtenerUbicacion()`, que verifica si el navegador soporta geolocalización a través de la propiedad `navigator.geolocation`. Si es compatible, se utiliza el método `getCurrentPosition`, que toma dos funciones como argumentos: `mostrarUbicacion` para procesar y mostrar las coordenadas (latitud y longitud) del usuario y `mostrarError` para manejar posibles errores. Estos errores incluyen casos como permiso denegado, información no disponible,

tiempo de espera agotado o errores desconocidos, y cada caso se maneja mediante un mensaje informativo que se muestra en el elemento HTML con el ID ubicación.

2.1.6 Impacto de HTML5 en el Desarrollo Web Actual

HTML5 ha facilitado el desarrollo de aplicaciones web complejas y de alto rendimiento, eliminando dependencias y optimizando la compatibilidad entre navegadores y dispositivos. Gracias a sus nuevas funcionalidades y API, HTML5 ha permitido a los desarrolladores crear interfaces de usuario más ricas, seguras y accesibles, maximizando la flexibilidad de diseño y mejorando la experiencia del usuario en la web.

2.2 CSS

Cascading Style Sheets (CSS), u Hojas de Estilo en Cascada, es un lenguaje de diseño que se utiliza para definir la presentación de documentos HTML y XML. Desde su creación en 1996 por el W3C, CSS ha permitido separar la estructura de contenido del diseño visual, proporcionando a los desarrolladores un control más granular sobre el aspecto de los elementos de una página web sin afectar la estructura de los datos. Este enfoque permite que el HTML mantenga un propósito claro de estructuración de la información, mientras que CSS define cómo debe mostrarse, optimizando así tanto el desarrollo como el mantenimiento de aplicaciones web como DRACO (W3C, 2023).

Mi Página Web

Bienvenido

Este es un ejemplo de página web sin CSS.

Artículos

Título del Artículo

Contenido del artículo, explicando la importancia de CSS en el diseño web.

© 2024 Mi Página Web

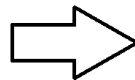


Ilustración 8 Comparación de página HTML sin CSS y con CSS

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo con CSS</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
      color: #333;
    }
    header {
      background-color: #3498db;
      color: #fff;
      padding: 20px;
      text-align: center;
    }
    section {
      padding: 20px;
      margin: 20px;
      background-color: #fff;
    }
  </style>
</head>
<body>
  <h2>Mi Página Web</h2>
  <h3>Bienvenido</h3>
  <p>Este es un ejemplo de página web sin CSS.</p>
  <h3>Artículos</h3>
  <h4>Título del Artículo</h4>
  <p>Contenido del artículo, explicando la importancia de CSS en el diseño web.</p>
  <small>© 2024 Mi Página Web</small>
</body>
</html>
```

```

        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }
    h2 {
        color: #3498db;
    }
    footer {
        background-color: #333;
        color: #fff;
        text-align: center;
        padding: 10px;
        position: fixed;
        width: 100%;
        bottom: 0;
    }
</style>
</head>
<body>
    <header>
        <h1>Mi Página Web</h1>
    </header>

    <section>
        <h2>Bienvenido</h2>
        <p>Este es un ejemplo de página web con CSS aplicado.</p>
    </section>

    <section>
        <h2>Artículos</h2>
        <article>
            <h3>Título del Artículo</h3>
            <p>Contenido del artículo, explicando la importancia de CSS en el
                diseño web.</p>
        </article>
    </section>

    <footer>
        <p>&copy; 2024 Mi Página Web</p>
    </footer>
</body>
</html>

```

La ilustración 8 muestra la comparativa del código expuesto, sin y con el CSS respectivamente. El código CSS embebido en el `<style>` del `<head>` estiliza la página web para mejorar su apariencia. Define una fuente general (Arial, sans-serif), elimina márgenes y rellenos predeterminados, y establece un fondo gris claro (#f4f4f4) con texto en color oscuro (#333) para el cuerpo. El `<header>` tiene un fondo azul (#3498db), texto blanco centrado y relleno para destacar el encabezado. Las secciones (`<section>`) tienen un fondo blanco, esquinas redondeadas (`border-radius`), sombra sutil (`box-shadow`), y márgenes internos y externos para mejorar la legibilidad. Los subtítulos (`<h2>`) heredan el color azul del encabezado. Finalmente, el `<footer>` tiene un fondo oscuro, texto blanco centrado, y está fijado en la parte inferior de la ventana con un ancho completo (`width: 100%`) y relleno para un diseño limpio.

2.2.1 La Evolución de CSS

Desde su versión inicial, CSS ha pasado por varias actualizaciones significativas. CSS2, lanzado en 1998, añadió soporte para posicionamiento avanzado de elementos y estilos para impresión, lo que mejoró la versatilidad del diseño web.

Sin embargo, CSS3, publicado en módulos desde 1999, representó un cambio fundamental al introducir la **modularización**, que permite a los desarrolladores implementar solo los módulos necesarios. Esto facilita la adopción de nuevas características sin comprometer el rendimiento.

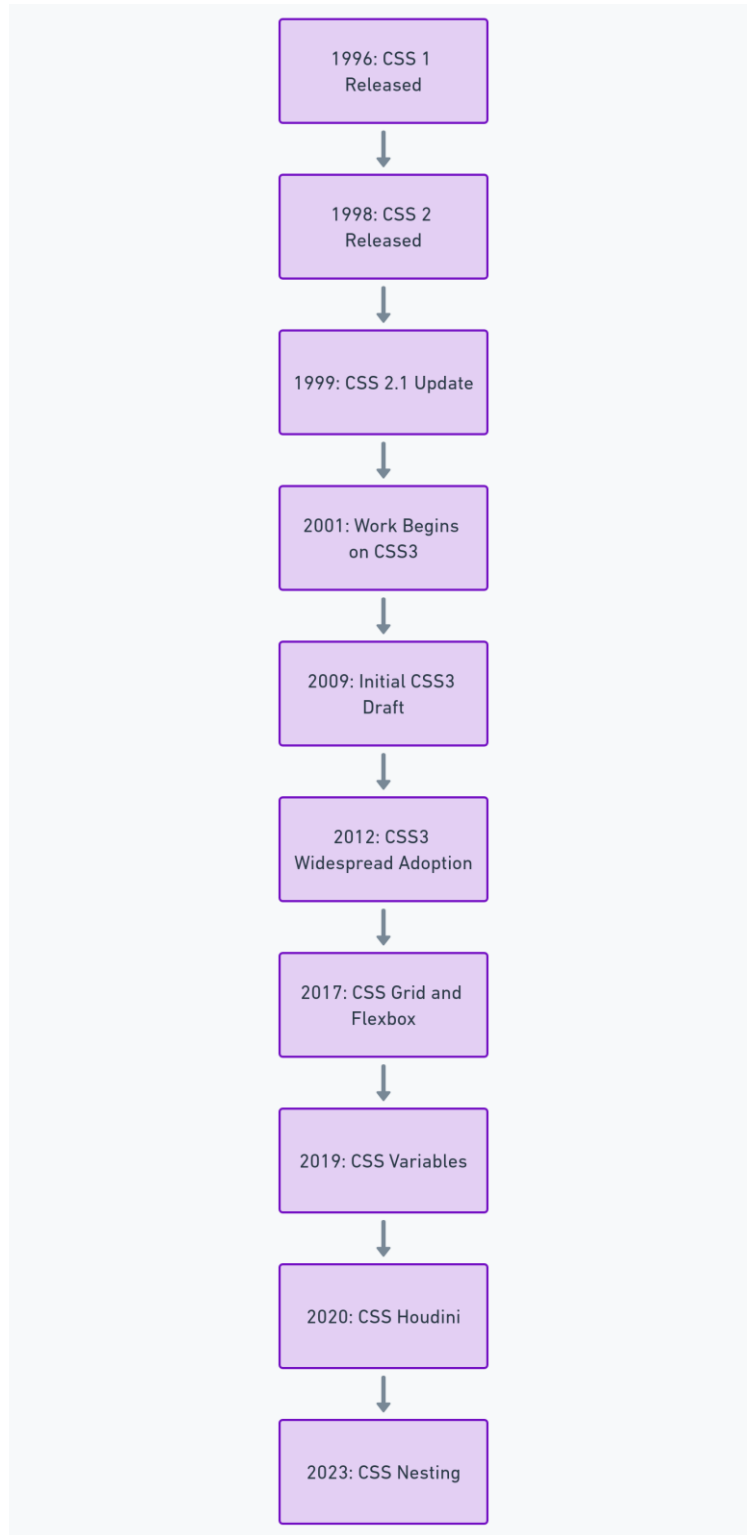


Ilustración 9 Historia de PHP

En la ilustración 9 se puede observar visualmente la evolución de CSS. CSS3 es especialmente relevante en el contexto de aplicaciones modernas, ya que añade funcionalidades avanzadas como **animaciones**, **efectos visuales** y **diseño adaptable** a diferentes dispositivos. Estas características permiten a los desarrolladores adaptar la presentación de la página a distintos entornos de uso, algo esencial para aplicaciones académicas como DRACO, que necesitan ser accesibles tanto en ordenadores como en dispositivos móviles (W3C, 2023).

2.2.2 Control de Estilos y Consistencia Visual

Entre las funcionalidades principales de CSS se encuentran la manipulación de **tipografía**, **colores** y **disposición** de elementos en la página, todo controlado a través de reglas de estilo definidas en hojas CSS externas (aunque, de manera menos recomendable, también se puede escribir el CSS dentro del mismo HTML de la página en vez de importarlo de una hoja externa). Esta metodología reduce la redundancia en el código HTML y permite aplicar estilos uniformes en toda la aplicación, manteniendo una identidad visual coherente y facilitando el mantenimiento del proyecto. En plataformas educativas como DRACO, esta consistencia es fundamental para asegurar una experiencia de usuario profesional y eficiente, donde los elementos de la interfaz mantienen un diseño cohesivo y fácil de navegar.

2.2.3 Animaciones y Transformaciones en CSS3

CSS3 introdujo capacidades avanzadas para **animaciones** y **transformaciones de elementos** en 2D y 3D. Propiedades como `transform`, `rotate`, y `scale` permiten añadir interactividad y dinamismo al diseño de la página, mejorando la experiencia del usuario sin necesidad de bibliotecas JavaScript adicionales. Las animaciones CSS se utilizan para hacer transiciones suaves, guiar al usuario a través de la interfaz y añadir interactividad visual que puede mejorar la comprensión y retención de la información.

Las animaciones son particularmente útiles en aplicaciones educativas, donde pueden emplearse para resaltar elementos importantes o dirigir la atención del usuario. Por ejemplo, en DRACO, animaciones sutiles podrían destacar el botón de envío de tareas o las notificaciones de éxito.

```
/* Animación en el cuadro */
.cuadro-animado {
  width: 100px;
  height: 100px;
  margin-top: 20px;
  background-color: #e74c3c;
  transition: transform 0.5s;
}

.cuadro-animado:hover {
  transform: rotate(45deg) scale(1.2);
}
```

2.2.4 Diseño Flexible con *Media Queries*

Otra característica destacada de CSS3 es el uso de **media queries**, que permite que el diseño de una página se adapte automáticamente a diferentes resoluciones de pantalla y dispositivos. Con *media queries*, los desarrolladores pueden establecer estilos específicos según el tamaño de la pantalla o el tipo de dispositivo, optimizando así la usabilidad de la aplicación. Esta capacidad es fundamental en entornos educativos, donde estudiantes y profesores pueden

acceder a la plataforma desde dispositivos móviles, tabletas o computadoras de escritorio.

```
/* Media Queries para Adaptabilidad */
@media (max-width: 768px) {
  nav ul {
    flex-direction: column;
    align-items: center;
  }

  main {
    flex-direction: column;
  }
}

@media (max-width: 480px) {
  header {
    font-size: 18px;
  }

  section {
    padding: 15px;
    margin: 10px 0;
  }

  nav ul li {
    margin: 5px 0;
  }
}
```

El código utiliza *media queries* en CSS para implementar un diseño responsivo que se adapta a diferentes tamaños de pantalla. Los *media queries* definen reglas específicas para dispositivos con anchos máximos de 768px y 480px, ajustando estilos para mejorar la visualización en pantallas más pequeñas. Cuando el ancho de la pantalla es de 768px o menos, se modifica el diseño del menú de navegación (`nav ul`) para que sus elementos se dispongan en una columna y se centren, y se organiza el contenido principal (`main`) también en una disposición de columna. Para pantallas aún más pequeñas, de 480px o menos, se ajusta el tamaño de la fuente del encabezado (`header`) a 18px, se reduce el espacio de relleno (`padding`) y margen (`margin`) de las secciones (`section`), y se disminuye el margen entre los elementos de la lista de navegación (`nav ul li`). Estas modificaciones garantizan que la página sea accesible y fácil de usar en dispositivos móviles y *tablets*.

2.2.5 Diseño Basado en Rejillas (*Grid*) y *Flexbox*

CSS3 ha facilitado la creación de interfaces web complejas mediante la introducción de sistemas de diseño basados en **rejillas (*grid*)** y ***flexbox***. Estos sistemas permiten una distribución más intuitiva y controlada de los elementos en la página, lo que es especialmente útil en aplicaciones que requieren interfaces dinámicas y bien organizadas, como DRACO.

- ***Grid***: Este sistema permite dividir el espacio disponible en una cuadrícula, facilitando el posicionamiento preciso de elementos en filas y columnas. Es particularmente útil en interfaces complejas que requieren una estructura visual coherente y adaptable.
- ***Flexbox***: *Flexbox* se diseñó para manejar el diseño de una manera más eficiente cuando se requiere una disposición flexible y alineada de los

elementos, permitiendo que estos se adapten al espacio disponible de forma automática. *Flexbox* es ideal para componentes de la interfaz como menús de navegación o tarjetas de información, que necesitan reordenarse según el tamaño de la pantalla.

De acuerdo con el W3C, el uso de *flexbox* y *grid* permite a los desarrolladores "organizar contenidos de manera más eficiente y flexible", lo cual optimiza la experiencia del usuario en aplicaciones académicas y de uso frecuente como DRACO (W3C, 2023).

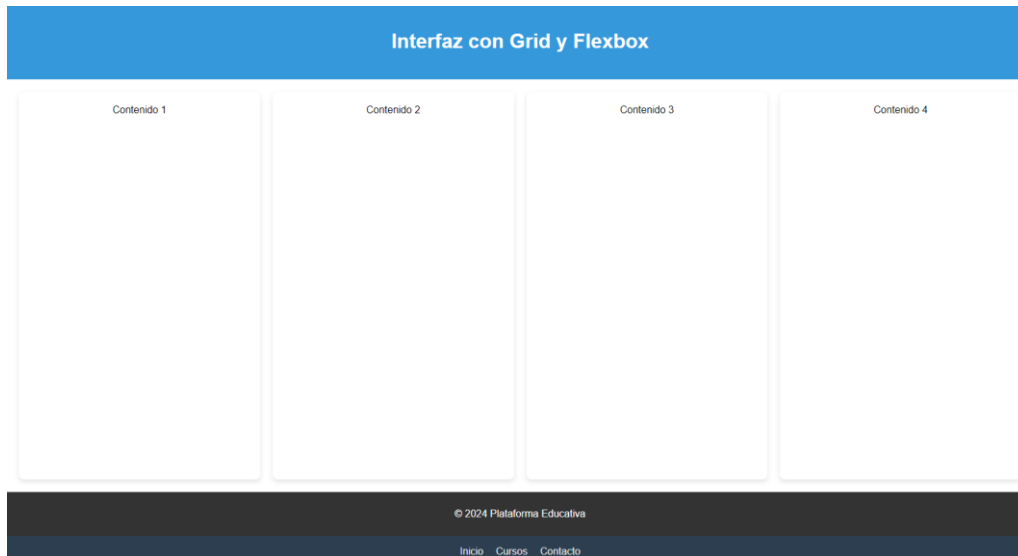


Ilustración 10 Ejemplo de uso de grid y flexbox

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  display: grid;
  grid-template-rows: auto 1fr auto;
  grid-template-areas:
    "header"
    "content"
    "footer";
  height: 100vh;
}

.header {
  grid-area: header;
  background-color: #3498db;
  color: #fff;
  text-align: center;
  padding: 20px;
}

.nav {
  background-color: #2c3e50;
  display: flex;
  justify-content: center;
}

.nav ul {
  list-style: none;
  display: flex;
}
```

```

        gap: 20px;
        padding: 0;
    }

    .nav ul li a {
        color: #ecf0f1;
        text-decoration: none;
    }

    .content {
        grid-area: content;
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
        gap: 20px;
        padding: 20px;
    }

    .box {
        background-color: #fff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        text-align: center;
    }

    .footer {
        grid-area: footer;
        background-color: #333;
        color: #fff;
        text-align: center;
        padding: 10px;
    }

    /* Media Queries para adaptar el diseño */
    @media (max-width: 768px) {
        .content {
            grid-template-columns: 1fr;
        }

        .nav ul {
            flex-direction: column;
            align-items: center;
        }
    }
}

```

Este código, que visualmente es representado en la ilustración 10, utiliza CSS *Grid* para estructurar la página y *Flexbox* para alinear elementos dentro de secciones específicas, combinando ambas técnicas para crear un diseño organizado y flexible. La página está dividida en tres áreas principales (header, content y footer) mediante *grid-template-areas*, con filas definidas en *grid-template-rows*, asegurando que ocupen toda la altura de la pantalla. Dentro del content, CSS Grid organiza los elementos en columnas que se ajustan automáticamente a través de *grid-template-columns: repeat(auto-fit, minmax(200px, 1fr))*, garantizando que cada elemento tenga un tamaño mínimo y se distribuya uniformemente. En la navegación (nav), se aplica Flexbox para centrar horizontalmente el menú (*justify-content: center*) y separar uniformemente los elementos de la lista con un espacio definido por *gap: 20px*. Esta combinación permite una estructura base consistente y un

posicionamiento detallado para los componentes individuales, destacando la versatilidad de Grid y la precisión de *Flexbox*.

2.3 PHP

PHP, acrónimo de *Hypertext Preprocessor*, es un lenguaje de programación de propósito general enfocado en el desarrollo de aplicaciones web dinámicas. Su creación en 1994 por Rasmus Lerdorf nació de la necesidad de gestionar contenido y procesar datos en la web de forma eficiente y accesible. Desde sus inicios, PHP ha sido un lenguaje de código abierto, lo cual ha favorecido una rápida adopción y una constante evolución gracias a la colaboración de la comunidad de desarrolladores. La popularidad de PHP, junto con su versatilidad en el desarrollo de aplicaciones web, lo ha convertido en una herramienta fundamental para proyectos académicos y de gestión de información, como es el caso de la plataforma DRACO (PHP Group, 2024). La ilustración 11 muestra la evolución de PHP de una manera más visual.

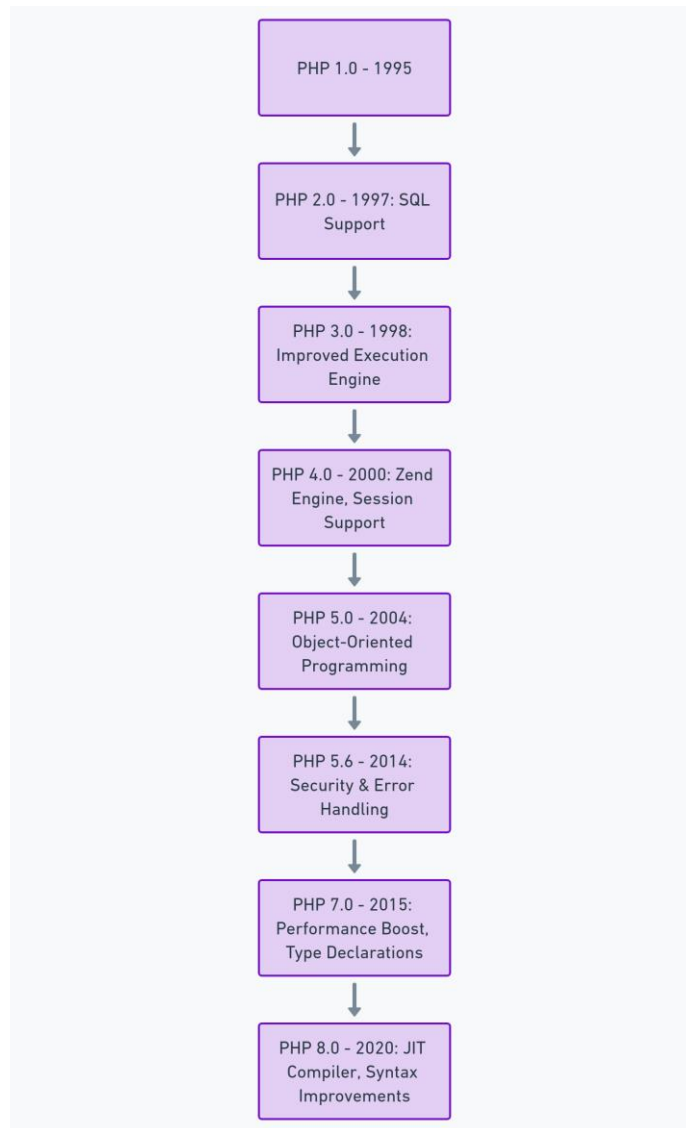


Ilustración 11 Evolución histórica de PHP

2.3.1 Evolución de PHP: Desde PHP/FI hasta PHP 8 y Más Allá

Las primeras versiones de PHP, como PHP/FI y PHP 3, se enfocaban principalmente en el procesamiento de formularios y la generación de contenido HTML dinámico. Sin embargo, a partir del lanzamiento de PHP 4 en el año 2000, el lenguaje experimentó un cambio fundamental. La introducción del motor Zend permitió una ejecución más eficiente de *scripts*, una mejor gestión de memoria y mayor estabilidad. PHP 4 marcó el inicio del uso de PHP para aplicaciones web complejas, gracias a su capacidad para interactuar con bases de datos y soportar estructuras de control avanzadas (PHP Group, 2024).

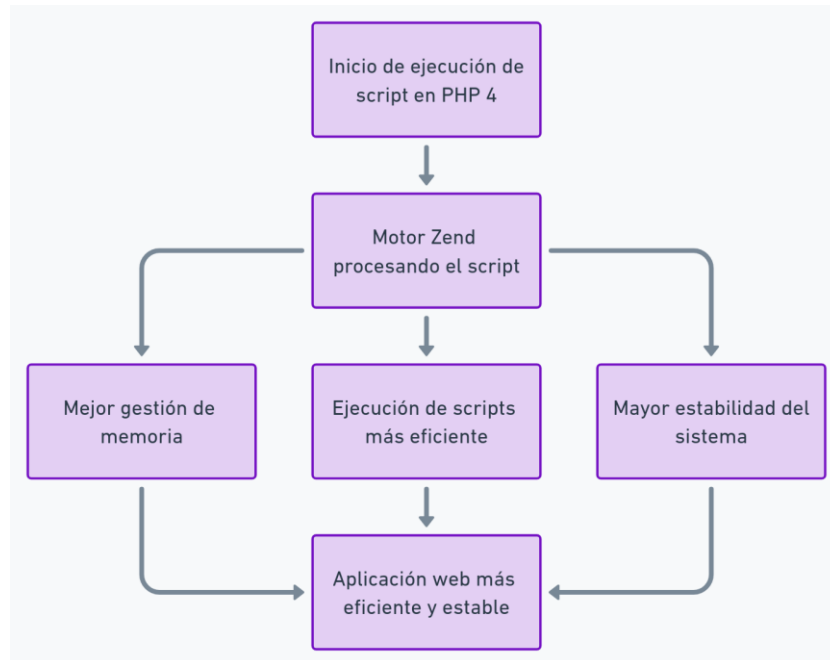


Ilustración 12 Impacto de motor Zend en PHP

El impacto del motor Zend se muestra visualmente en la ilustración 12. PHP 5, lanzado en 2004, trajo un enfoque más orientado a objetos, lo que facilitó la creación de sistemas modulares y escalables. Esto incluyó funcionalidades clave como excepciones, interfaces y destructores, que permitieron a los desarrolladores organizar y gestionar mejor el código. PHP 5 también mejoró la integración con bases de datos, permitiendo conexiones seguras y eficientes con MySQL, PostgreSQL y otros sistemas de bases de datos relacionales. Este avance hizo que PHP fuera aún más adecuado para proyectos académicos, donde el manejo de datos y la modularidad son esenciales (PHP Group, 2024).

PHP 7, lanzado en 2015, representó otro avance importante en la evolución del lenguaje. Con la introducción de un nuevo motor de ejecución, PHP 7 mejoró notablemente el rendimiento y la eficiencia en el consumo de recursos. Esta versión incluyó soporte para tipificación estricta de parámetros y el uso de operadores seguros para la navegación de datos, lo que proporciona un mayor control y seguridad en la gestión de variables. Esto resulta especialmente relevante en plataformas como DRACO, donde la seguridad en el manejo de datos académicos es crucial (PHP Group, 2024).

PHP 8, lanzado en noviembre de 2020, marcó otro hito significativo en la evolución del lenguaje, con la introducción de características innovadoras que

mejoraron tanto la experiencia del desarrollador como el rendimiento de las aplicaciones. Una de las principales novedades fue la inclusión de *Just-In-Time (JIT) Compilation*, que permite una ejecución más rápida al compilar fragmentos de código en tiempo de ejecución, beneficiándose las aplicaciones intensivas en cálculos. Además, PHP 8 introdujo *attributes* (anotaciones), que simplifican la metaprogramación al permitir a los desarrolladores agregar metadatos a clases, métodos y propiedades de manera nativa. Otras mejoras destacadas incluyen la validación de parámetros con la función de *union types*, las expresiones *match*, y los operadores de *null-safe*, que facilitan la escritura de código más limpio y seguro. Estas capacidades han consolidado a PHP como una herramienta poderosa para el desarrollo de aplicaciones modernas y escalables, incluyendo proyectos académicos y empresariales donde la eficiencia y la claridad del código son fundamentales. (PHP Group, 2024). Se pueden ver todos los avances de las distintas versiones de PHP visualmente en la ilustración 13.

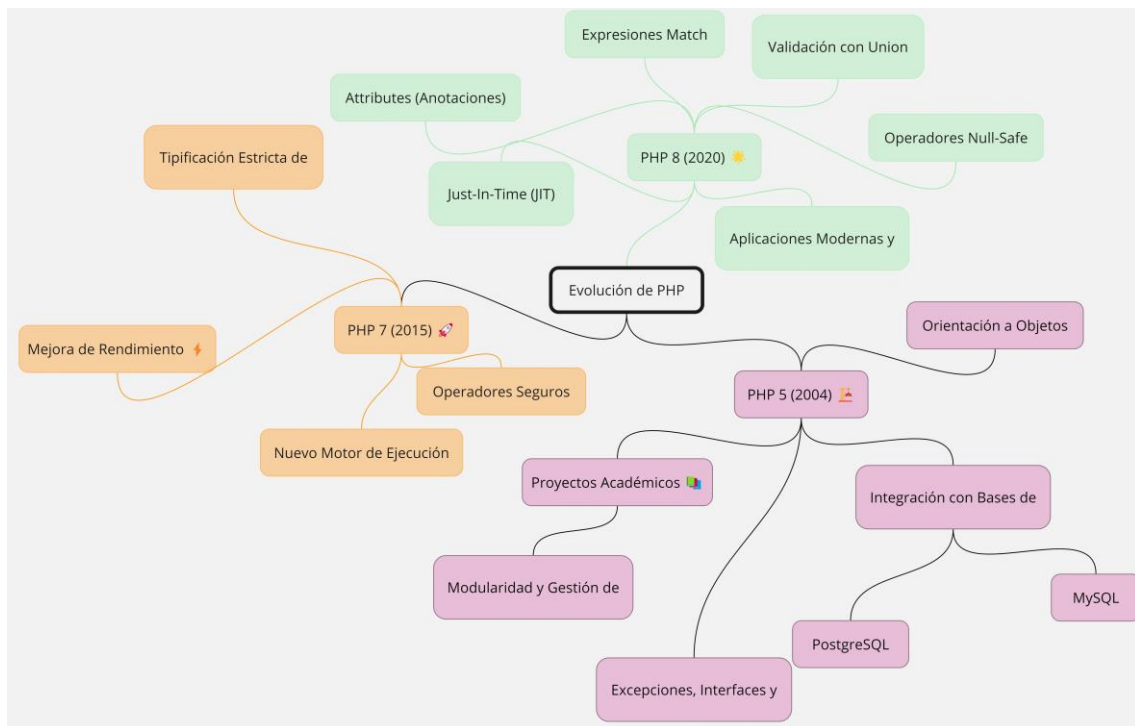


Ilustración 13 Cambios y mejoras clave introducidos en PHP 5, 7 y 8

2.3.2 Generación de Contenido Dinámico y la Ventaja de PHP en Aplicaciones Interactivas

PHP ha destacado por su capacidad para generar contenido dinámico en tiempo real, lo que lo convierte en una elección ideal para aplicaciones web que requieren una interacción constante con el usuario y dependen de bases de datos. Esto es posible gracias a que PHP puede integrarse directamente con HTML, permitiendo incluir lógica de programación dentro de la estructura de una página web y facilitando el desarrollo de interfaces interactivas y dinámicas.

En el caso de DRACO, PHP se utiliza para procesar envíos de tareas, gestionar la autenticación de usuarios y almacenar resultados en una base de datos. La capacidad de PHP para adaptarse a diversos entornos de bases de datos lo hace muy adecuado para proyectos académicos que requieren un manejo seguro y eficiente de datos de estudiantes (PHP Group, 2024).

2.3.3 Compatibilidad con Bases de Datos y Gestión de Información+

Uno de los aspectos más destacados de PHP es su compatibilidad con una amplia variedad de sistemas de gestión de bases de datos. MySQL, uno de los sistemas más comunes, permite que PHP gestione grandes volúmenes de datos de manera segura y eficiente. Esta compatibilidad es crucial para aplicaciones como DRACO, donde se maneja una gran cantidad de información académica, desde datos de estudiantes hasta resultados de ejercicios y calificaciones.

La capacidad de PHP para interactuar con bases de datos relacionales y no relacionales permite una gestión flexible de la información, lo que resulta especialmente útil en entornos académicos donde la seguridad y precisión en el manejo de datos son primordiales (PHP Group, 2024).

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Conexión a Base de Datos con PHP y MySQL</title>
</head>
<body>
  <header>
    <h1>Listado de Estudiantes</h1>
  </header>

  <section>
    <?php
      // Datos de conexión
      $servidor = "localhost";
      $usuario = "usuario";
      $contraseña = "contraseña";
      $baseDeDatos = "base_de_datos";

      // Crear una nueva conexión a MySQL
      $conexion = new mysqli($servidor, $usuario, $contraseña, $baseDeDatos);

      // Comprobar la conexión
      if ($conexion->connect_error) {
        die("Error de conexión: " . $conexion->connect_error);
      }

      // Consulta SQL para obtener datos de estudiantes
      $sql = "SELECT id, nombre, calificacion FROM estudiantes";
      $resultado = $conexion->query($sql);

      // Comprobar si hay resultados y mostrarlos
      if ($resultado->num_rows > 0) {
        echo "<table>
          <tr>
            <th>ID</th>
            <th>Nombre</th>
            <th>Calificación</th>
          </tr>";
        // Recorrer cada fila de resultados
        while($fila = $resultado->fetch_assoc()) {
          echo "<tr>
            <td>" . $fila["id"] . "</td>
            <td>" . $fila["nombre"] . "</td>
            <td>" . $fila["calificacion"] . "</td>
          </tr>";
        }
      }
    </?php>
  </section>
</body>
</html>
```

```

        echo "</table>";
    } else {
        echo "<p>No se encontraron resultados.</p>";
    }

    // Cerrar la conexión
    $conexion->close();
    ?>
</section>

<footer>
    <p>&copy; 2024 Plataforma Educativa DRACO</p>
</footer>
</body>
</html>

```

2.3.4 Frameworks de PHP: Escalabilidad y Funcionalidades Avanzadas

PHP también es conocido por su amplia gama de bibliotecas y *frameworks*, como Laravel y Symfony, que facilitan el desarrollo de aplicaciones modulares y escalables. Estos *frameworks* proporcionan herramientas avanzadas para la autenticación de usuarios, el control de acceso y la validación de datos, funcionalidades esenciales en entornos educativos donde se manejan datos confidenciales y sensibles.

Los *frameworks* modernos de PHP no solo permiten una estructura más organizada del código, sino que también optimizan el desarrollo de aplicaciones web. En el caso de DRACO, el uso de *frameworks* podría facilitar la implementación de funcionalidades avanzadas y garantizar una mejor gestión de la seguridad de los datos de los usuarios.

2.3.5 Flexibilidad y Versatilidad en Diferentes Entornos de Desarrollo

PHP es compatible con una amplia variedad de servidores web, lo que lo convierte en una opción versátil y adaptable a diferentes entornos de desarrollo y despliegue. Esta compatibilidad permite que PHP sea utilizado en sistemas operativos variados y en servidores de bajo costo, lo que facilita su adopción en proyectos académicos y comerciales sin la necesidad de configuraciones especializadas.

La versatilidad de PHP en términos de compatibilidad ha contribuido a su popularidad y permanencia como una de las principales opciones para el desarrollo web, especialmente en proyectos de acceso abierto y aplicaciones académicas como DRACO (PHP Group, 2024).

2.4 DRACO

DRACO es la plataforma web utilizada en la asignatura de Procesadores de Lenguajes, impartida en la Escuela Técnica Superior de Ingenieros Informáticos (ETSIINF) de la Universidad Politécnica de Madrid (UPM). Esta página funciona como un sistema complementario de evaluación para los estudiantes, permitiéndoles entregar prácticas, realizar ejercicios y completar cuestionarios activados y gestionados por el profesorado.

Una característica distintiva de DRACO frente a otras plataformas de asignaturas en la ETSIINF es su enfoque en la interacción con el usuario

mediante el uso de técnicas de gamificación. La gamificación, definida como el uso de elementos de diseño y mecánicas de juego en contextos no lúdicos para motivar la participación y el compromiso, tiene como objetivo hacer que la experiencia educativa sea más atractiva y estimulante para los estudiantes. DRACO implementa gamificación para incentivar la participación activa en los ejercicios y mejorar la experiencia de aprendizaje en la asignatura.

Entre las técnicas de gamificación empleadas en DRACO se incluyen la obtención de puntos por cada actividad y práctica completada, un sistema de *ranking* donde los estudiantes pueden comparar sus puntajes, la adquisición de “monedas” intercambiables por diversos beneficios en la aplicación, otorgadas en función de la actividad realizada por los alumnos, etc. Estos elementos no solo promueven el interés de los estudiantes por la asignatura, sino que también fomentan un ambiente de competencia sana y el logro de metas individuales.

A continuación, se van a exponer las distintas funcionalidades de DRACO, tanto para el estudiante como para el profesor.



Ilustración 14 Página Home del estudiante

Comenzando con el estudiante, puede apreciarse en la ilustración 14 la página que se muestra una vez se hace el *login*. Se van a listar y explicar cada una de las funcionalidades visibles:

- **Actividades:** Aquí se muestran las actividades que puede ejecutar en ese momento el estudiante. Estas actividades incluyen preguntas de respuesta larga, de respuesta corta, tipo test, etcétera. Cada una de estas preguntas darán puntos acumulables al estudiante, así como la posibilidad de subir de nivel, obtener insignias y acumular monedas.
- **Prácticas:** Se podrán entregar las prácticas que estén activas en ese momento para comprobar su funcionamiento. Para esta asignatura de Procesadores de Lenguajes, las prácticas serán Tabla de símbolos, Análisis Léxico, Análisis Sintáctico y Análisis Semántico.
- **Puntos:** Este apartado llevará a una lista con un registro de cuando y cómo ha ganado el estudiante puntos.
- **Insignias:** Muestra una serie de insignias que ha conseguido el estudiante. Las insignias se consiguen cuando el estudiante ha conseguido llegar a lograr distintas metas que otorgan esta recompensa.
- **Clasificación:** Muestra una lista de todos los estudiantes que están cursando esta asignatura, junto con sus puntos actuales, de mayor a

menor puntuación. El estudiante ve también cuál es su posición en el *ranking*.

- Noticias: Muestra acontecimientos de interés para el estudiante relacionados con DRACO o la asignatura. Esto incluye fechas de exámenes, pruebas, cambios de aulas, etc.
- Perfil: Muestra información sobre el perfil del estudiante y permite modificar información del mismo, así como añadir un avatar.
- DRACoins: Se permite cambiar puntos obtenidos en las pruebas y prácticas por monedas (DRACoins). Estas DRACoins se pueden canjear luego por recompensas, como disminuir tiempos entre pruebas de prácticas, más tiempo para contestar una pregunta, una pista al responder una pregunta, etc.



Ilustración 15 Página Home del profesorado/administrador

Ahora se puede ver el punto de vista del profesorado o administrador. Las opciones para actuar son las que se ven en la parte superior derecha de la ilustración 15.

Estas opciones incluyen:

- Configurar Actividades: Esto permite configurar las actividades como preguntas tipo test, respuestas cortas, etcétera. Dentro, se permite crear nuevas actividades, editarlas o importar actividades ya creadas de años anteriores, para su uso en futuros semestres.
- Configurar prácticas: Configura cuándo ocurrirán las prácticas de la asignatura, junto con opciones de información de los grupos que han realizado las prácticas, y otros datos de configuración de los comprobadores.
- Estadísticas: Muestran datos estadísticos sobre los estudiantes registrados. Incluye puntos, DRACoins, nivel, posición en el *ranking*, etc.
- Configurar Insignias: Permite crear, eliminar y modificar insignias de la asignatura y cómo conseguirlas.
- Configurar Sistema: Permite configurar aspectos de conexión del sistema y otros diversos parámetros, además de permitir exportar información del sistema.

3 Planteamiento del problema

A continuación, se plantean las principales problemáticas dentro de este TFG. La mejora y mantenimiento de DRACO es fuente de varios problemas que se han solucionado. Estos problemas incluyen los siguientes:

- Ausencia de una opción de importar múltiples actividades en la sección de importar actividades
- Inclusión de nueva información dentro de la página de actividades e importar actividades.
- Problema con la copia de lenguajes.
- Problema con el borrado de lenguajes
- Ampliación del funcionamiento del corrector de prácticas para el módulo Tabla de Símbolos.
- Corrección del nivel mínimo al crear una actividad.

A continuación, se tratarán cada uno de estos problemas más en detalle, además de los requisitos requeridos para su correcta solución.

3.1 Ausencia de una opción de importar múltiples actividades en la sección de importar actividades

Los profesores actualmente disponen de una página donde se listan todas las actividades publicadas en cursos anteriores para los alumnos. En esta página se permite realizar muchas acciones con estas actividades, incluyendo verlas, borrarlas e importarlas. Importar una actividad significa introducirla en la lista de actividades disponibles para este curso. Esto permite que esta actividad pueda ser activada y por tanto publicada para que la puedan realizar los alumnos.

Esta página tenía la apariencia que se muestra en las ilustraciones 15 y 16:

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Puntos Min	Puntos Max	Día Inicio	Día Fin	
<input type="text"/>	<input type="text" value="Seleccione"/>	<input type="text" value="Seleccione"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="2024-12-16"/>	<input type="text" value="2024-12-23"/>	<input type="button" value="Filtrar"/>

Seleccione la actividad a gestionar:

Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	
Página de la asignatura	Múltiples respuestas	Inactiva	5	25	2023-09-08 22:22:00	2023-09-27 13:57:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>
Fases de un compilador	Ordenar	Inactiva	10	50	2023-09-13 15:55:00	2023-09-28 14:32:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>
Gramática léxica	Múltiples respuestas	Inactiva	3	33	2023-09-13 19:29:00	2023-09-29 10:10:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>
Termina la frase	Respuesta corta	Inactiva	5	50	2023-09-14 13:13:00	2023-09-30 23:32:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>
Gramática del Léxico	Múltiples respuestas	Inactiva	3	33	2023-09-15 14:10:00	2023-09-30 13:10:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>
G para id	Múltiples respuestas	Inactiva	3	44	2023-09-15 18:18:00	2023-10-01 11:11:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>
Automata del Léxico	Múltiples respuestas	Inactiva	3	33	2023-09-17 11:11:00	2023-10-02 02:34:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>
AFD	Múltiples respuestas	Inactiva	3	44	2023-09-18 18:09:00	2023-10-02 12:22:00	<input type="button" value="Importar"/> <input type="button" value="Ver"/> <input type="button" value="Borrar"/>

Ilustración 16 Primera parte de página ImportActivities Original

					01:00:00	14:21:00		
Lenguaje sin declaración - 6	Múltiples respuestas	Inactiva	5	44	2024-01-11 10:00:00	2024-01-18 14:00:00	Borrar	Ver
First - 8	Múltiples respuestas	Inactiva	5	66	2024-01-12 00:00:00	2024-01-19 12:34:00	Importar	Ver
Gaudeamus	Ordenar	Inactiva	5	33	2024-01-13 12:00:00	2024-01-21 23:45:00	Borrar	Ver
Mejorar Draco	Múltiples respuestas	Inactiva	1	21	2024-01-14 13:18:00	2024-01-25 09:34:00	Importar	Ver
Continuación	Múltiples respuestas	Inactiva	3	20	2024-01-15 10:01:00	2024-01-23 23:11:00	Borrar	Ver
Síntesis	Ordenar	Inactiva	3	22	2024-01-16 00:30:00	2024-01-24 23:34:00	Importar	Ver
Continuación - web	Múltiples respuestas	Inactiva	2	23	2024-01-17 12:23:00	2024-01-25 10:10:00	Borrar	Ver
Ejemplo	Respuesta larga	Auto	0	10	2024-09-11 00:00:00	2025-09-11 00:00:00	Importar	Ver
Hola	Múltiples respuestas	Auto	0	10	2024-10-02 00:00:00	2024-11-02 00:00:00	Borrar	Ver
asdfadsfa	Encuesta de Respuesta larga	Activa	0	12	2024-10-08 00:00:00	2024-11-08 00:00:00	Importar	Ver

Importar todas

Volver

Ilustración 17 Segunda parte de página ImportActivities Original

Como se pueden ver en la ilustración 16 y 17, a pesar de que es posible importar cada actividad una a una, es muy ineficiente hacerlo en momentos en los que se quieran importar muchas actividades en un mismo periodo de tiempo, ya que cada vez que se importa una actividad, la página se refresca para mostrar solo las actividades no importadas. Y la opción de importar todas tampoco permite mucha flexibilidad. Esto hace ver que sería de gran ayuda algún método que permita seleccionar varias actividades a importar a la vez.

Los requisitos para este problema serían:

- **Requisito 1.1. Selección Múltiple de Actividades:** El sistema debe permitir a los profesores seleccionar múltiples actividades de la lista al mismo tiempo.
- **Requisito 2.1. Importación Masiva de Actividades:** El sistema debe permitir importar todas las actividades seleccionadas sin refrescar la página tras cada importación individual.
- **Requisito 3.1. Actualización de la Lista de Actividades:** Una vez completada la importación múltiple, la lista debe actualizarse dinámicamente para mostrar solo las actividades no importadas.
- **Requisito 4.1. Selección de Todas las Actividades:** Debe existir una opción para seleccionar todas las actividades visibles en la lista con un solo clic.
- **Requisito 5.1. Deshacer Selección:** El sistema debe permitir deseleccionar todas las actividades seleccionadas.
- **Requisito 6.1. Consistencia:** La apariencia y comportamiento de los botones y mensajes deben seguir los estilos utilizados en el resto del sistema DRACO.

3.2 Inclusión de nueva información dentro de la página de actividades e importar actividades.

Cuando un profesor maneja las actividades que va a permitir realizar a los alumnos, utiliza dos páginas principalmente. Una de ellas es la ya mencionada

página de importación de Actividades. Otra página es la página de gestión de actividades, que se muestra en la Ilustración 18.

Puede filtrar actividades que cumplan ciertas características

Título ▲ ▼ Tipo de Actividad ▲ ▼ Estado ▲ ▼ Medalla ▲ ▼ Puntos Min ▲ ▼ Puntos Max ▲ ▼ Día Inicio ▲ ▼ Día Fin ▲ ▼

Seleccione la actividad a gestionar:

Título	Tipo	Estado	Medalla	Puntuación	Fecha Inicio	Fecha Fin					
Segundo examen test	Respuesta múltiple	Auto	Sin definir	3 - 22	07-11-2021 23:59	14-11-2021 23:57	Revisar	Editar	Activar	Ver	
Examen parcial	Respuesta corta	Activa	Sin definir	3 - 20	16-11-2021 17:17	21-11-2021 22:22	Revisar	Editar	Desactivar	Auto	Ver
Encuesta - 500 puntos	Respuesta corta	Activa	Sin definir	1 - 8	06-01-2022 12:34	27-01-2022 12:34	Revisar	Editar	Desactivar	Auto	Ver
Música en Clase práctica TDS	Respuesta única	Activa	Sin definir	3 - 22	24-11-2022 17:00	30-11-2022 21:22	Revisar	Editar	Desactivar	Auto	Ver
Fiesta	Respuesta única	Activa	Sin definir	3 - 11	26-01-2023 17:59	27-01-2023 17:00	Revisar	Editar	Desactivar	Auto	Ver
Bienvenida	Respuesta única	Activa	Sin definir	0 - 50	06-09-2023 08:58	26-09-2023 23:32	Revisar	Editar	Desactivar	Auto	Ver
Mascarilla	Respuesta única	Activa	Sin definir	2 - 11	07-09-2023 18:00	26-09-2023 01:23	Revisar	Editar	Desactivar	Auto	Ver
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	Revisar	Editar	Activar	Borrar	Ver
Analizadores	Respuesta corta	Inactiva	Sin definir	5 - 40	12-09-2023 13:55	28-09-2023 09:55	Revisar	Editar	Activar	Borrar	Ver
Token	Respuesta única	Inactiva	Scanneator	10 - 75	14-09-2023 11:45	29-09-2023 23:11	Revisar	Editar	Activar	Borrar	Ver
Función del Analizador Léxico	Respuesta múltiple	Inactiva	Scanneator	10 - 75	16-09-2023 21:00	01-10-2023 13:13	Revisar	Editar	Activar	Borrar	Ver
Tokens del programa	Respuesta múltiple	Inactiva	Scanneator	5 - 44	22-09-2023 19:00	07-10-2023 11:22	Revisar	Editar	Activar	Borrar	Ver

Ilustración 18 Página manejar actividades

Esta página permite manejar las actividades importadas desde la página de importar actividades, así como todas las actividades nuevas creadas. Esto incluye activarlas y desactivarlas, haciendo que sean visibles como actividades a realizar para los alumnos o no, además de ver los resultados y respuestas de los alumnos, ejecutar las actividades desde el punto de vista del alumno, editar la actividad, etc.

El problema reside en dos partes principales:

- Primero, como puede verse en las ilustraciones 15 y 16, la página de importar actividades no tiene tanta información sobre las actividades a importar como la información que tiene la página de manejar actividades.
- El segundo problema es que ninguna de las dos secciones muestra el nivel necesario para la realización de una actividad. Cada alumno tiene un nivel que se incrementa cuando hacen actividades y prácticas. Las actividades tienen también un nivel mínimo del alumno para que pueda realizar ciertas actividades. No mostrar esta información puede causar la activación de una actividad que ningún alumno o muy pocos alumnos puedan realizar, sin darse el profesor cuenta.

Los requisitos para este problema serían:

- **Requisito 1.2. Ampliación de información en la página de Importar Actividades:** La página de Importar Actividades debe incluir la misma información que aparece en la página Manejar Actividades
- **Requisito 2.2. Ampliación de los filtros de la página de Importar Actividades:** La página de Importar Actividades debe incluir los mismos filtros que aparecen en la página Manejar Actividades.
- **Requisito 3.2. Visualización del Nivel Mínimo Requerido:** Ambas páginas (Importar Actividades y Manejar Actividades) deben mostrar el Nivel Mínimo Requerido para realizar cada actividad. Esta información

debe ser visible en una columna adicional claramente etiquetada como "Nivel".

- **Requisito 4.2. Añadir filtro de nivel en ambas páginas:** Ambas páginas han de dar la opción de filtrar por nivel.
- **Requisito 5.2. Consistencia:** La apariencia y comportamiento de los botones y mensajes deben seguir los estilos utilizados en el resto del sistema DRACO.

3.3 Problema con la definición de lenguajes.

Para cada curso académico, los profesores definen un lenguaje que se usará para la creación y corrección de prácticas. Normalmente, lo que se hace es copiar el lenguaje utilizado el año anterior y, en caso de que fuese necesario o conveniente, este se modificaría en los aspectos necesarios.

Hay muchísimos aspectos a tener en cuenta de cara a la copia del lenguaje, ya que hay muchas variables que influyen y hay que rellenar con la creación de un lenguaje para un nuevo curso académico; esta es una de las razones por las que la funcionalidad de copiar un lenguaje es tan útil.

El problema reside en que la copia, a pesar de que se realiza correctamente para gran parte del proceso de copia, no es así para todos los elementos. Específicamente, no se importan los apartados de Errores Sintácticos y Errores Semánticos.

ERRORES SINTÁCTICOS					
Los errores ya presentes en algún fragmento no se pueden modificar.					
Código	Nombre	Mensaje	Explicación	Parámetros	Eliminar
T1	Sentencia sin fin de sentencia	Falta el punto y coma del final de la sentencia	Las sentencias simples deben acabar siempre		<input type="checkbox"/>
T2	Falta cerrar paréntesis	En una expresión se ha abierto un paréntesis	Los paréntesis deben estar balanceados. La		<input type="checkbox"/>
T3	Sobra cerrar paréntesis	En una expresión se ha cerrado un paréntesis	Los paréntesis deben estar balanceados. La		<input type="checkbox"/>
T4	No pueden aparecer dos operandos consecutivos	Hay una expresión con dos operandos, pero	Cuando una expresión tiene dos operandos,	Operandouno, Operandos	<input type="checkbox"/>
T5	Falta un operando	El operador binario %1 requiere dos operandos	Los operadores binarios necesitan tener dos o	Operador	<input type="checkbox"/>
T6	Operador unario sin variable	El operador %1 debe ir siempre acompañado	Los operadores de auto-incremento o auto-dec	Operador	<input type="checkbox"/>
T7	Falta el tipo en la declaración de la variable	La variable %1 que se está declarando carece	Al declarar una variable (en este caso, %1) ha	Variable	<input type="checkbox"/>
T8	Falta la variable en la declaración	Se está intentando declarar una variable, pero	La instrucción 'let' requiere poner el nombre d		<input type="checkbox"/>
T9	No se puede declarar una variable de tipo void	Se está intentando declarar la variable %1 de	No se pueden declarar variables de tipo 'void'	Variable	<input type="checkbox"/>
T10	Falta la expresión en la instrucción put	La instrucción put debe ir seguida de una expres	La instrucción 'put' imprime el valor de una expres		<input type="checkbox"/>
T11	Falta el parámetro en la instrucción get	La instrucción get debe ir seguida de una variable	La instrucción 'get' lee un valor del teclado y s		<input type="checkbox"/>
T12	La instrucción get no puede llevar una expresión	La instrucción get debe ir seguida de una variable	La instrucción 'get' lee un valor del teclado y s		<input type="checkbox"/>
T13	Falta la expresión en una asignación	Falta la expresión en la asignación a la variable	La sentencia de asignación a %1 necesita d	Variable	<input type="checkbox"/>
T14	Parámetro vacío erróneo en la llamada a una función	En la llamada a la función %1 no se puede	Las llamadas a las funciones sin parámetros	Función	<input type="checkbox"/>
T15	Faltan los paréntesis en la condición de la sentencia	La condición de la sentencia %1 va siempre e	La sentencia de control de flujo %1 lleva una	Sentencia	<input type="checkbox"/>
T16	Falta el tipo de retorno de la función	En la declaración de la función %1 se ha o	Todas las funciones deben indicar el tipo de re	Función	<input type="checkbox"/>
T17	Faltan los parámetros en la declaración de la función	En la declaración de la función %1 se han	Todas las funciones deben indicar el tipo de lo	Función	<input type="checkbox"/>
T18	Un parámetro de una función no puede ser controlado	En la declaración de la función %1 se ha p	Los parámetros de las funciones no pueden s	Función, Parámetro	<input type="checkbox"/>
T19	No se permiten declarar funciones anidadas	La función %1 se está intentando declarar	Una función no se puede declarar dentro de c	Función interna, Función externa	<input type="checkbox"/>
T20	Falta la condición en una sentencia de control	La sentencia %1 debe llevar una condición	La sentencia de control de flujo %1 se debe es	Sentencia	<input type="checkbox"/>
T21	La instrucción case no puede contener una variable	La instrucción case contiene la variable %1	Las instrucciones 'case' solamente pueden llea	Variable	<input type="checkbox"/>
T22	Se esperaba una llave, pero se encontró un punto y coma	Se ha encontrado un punto y coma al analizar	El cuerpo de la sentencia de control de flujo %	Sentencia	<input type="checkbox"/>
T23					
T24					
T25					
T26					
T27					

Opciones para añadir al final:

Ilustración 19 Errores sintácticos de un Lenguaje en funcionamiento

ERRORES SEMÁNTICOS

Los errores ya presentes en algún fragmento no se pueden modificar.

Código	Nombre	Mensaje	Explicación	Parámetros	Eliminar
M1	Instrucción no permitida fuera de un <code>switch</code>	La palabra reservada <code>break</code> solo puede aparecer	La instrucción <code>break</code> solamente es válida de	Sentencia	<input type="checkbox"/>
M2	Tipos incompatibles en la asignación de variables	Se está intentando asignar la variable <code>%2</code> a	Los dos tipos de una asignación deben ser ig	Var1, Var2, Tipo1, Tipo2	<input type="checkbox"/>
M3	Instrucción no permitida fuera de las funciones	La palabra reservada <code>return</code> solo puede aparec	La instrucción <code>return</code> solamente es válida de	Sentencia	<input type="checkbox"/>
M4	Variable ya declarada	La variable <code>%1</code> que se intenta declarar ya es	Una variable no se puede declarar más de un	Variable	<input type="checkbox"/>
M5	Función no declarada	Se está intentando llamar a la función <code>%1</code>	Para poder llamar a la función <code>%1</code> es necesar	Función	<input type="checkbox"/>
M6	La expresión no es lógicamente	La condición de la sentencia <code>if</code> no e	La condición de toda sentencia <code>if</code> de	Sentencia	<input type="checkbox"/>
M7	El tipo de la expresión no se puede asignar a	Se está intentando asignar una expresión <code>%</code>	En una asignación, ambos tipos tienen que si	Variable, TipoExp, TipoVar	<input type="checkbox"/>
M8	Tipos incompatibles en el operador aritmético	El operador <code>%1</code> requiere que ambos operan	El operador aritmético <code>+</code> solamente	Operador	<input type="checkbox"/>
M9	Tipos incompatibles en el operador relacional	El operador <code>%1</code> requiere que ambos operan	El operador relacional <code><</code> solamente p	Operador	<input type="checkbox"/>
M10	Tipos incompatibles en el operador lógico	El operador <code>%1</code> requiere que sus operandos	El operador lógico <code>&</code> solamente pue	Operador	<input type="checkbox"/>
M11	Tipo no válido en operador de auto-incremento	El operador <code>%1</code> solo puede aplicarse a varia	El operador <code>%1</code> solamente puede aplicarse a v	Operador, Tipo	<input type="checkbox"/>
M12	Tipo no válido en instrucción de asignación	El operador de asignación con operación <code>%</code>	El operador de asignación con la operación <code>%</code>	OperadorAs, Tipo	<input type="checkbox"/>
M13	Redeclaración de una variable global	La variable <code>%1</code> ya se había declarado implícit	Si una variable (<code>%1</code>) estaba declarada implícit	Variable	<input type="checkbox"/>
M14	Tipo incompatible en instrucción de entrada/salida	El tipo indicado en la instrucción <code>%1</code> no es \	La instrucción <code>%1</code> solamente admite tipos ent	InstrucciónES	<input type="checkbox"/>
M15	Número incorrecto de parámetros en la llamada	La llamada a la función <code>%1</code> tiene <code>%2</code> pará	El número de parámetros actuales (<code>%2</code>) que	Función, NúmeroParAct, NúmeroParForm	<input type="checkbox"/>
M16	Llamada a función con un parámetro de tipo	Se está haciendo una llamada a la función <code>%</code>	La función <code>%1</code> está declarada con el <code>%2</code> par	Función, OrdenPar, TipoParAct, TipoParFormal	<input type="checkbox"/>
M17	El tipo del <code>return</code> no coincide con el declarado	La función <code>%1</code> tiene un <code>return</code> con u	La función <code>%1</code> se ha declarado que devuelve	Función, TipoRet, TipoFunc	<input type="checkbox"/>
M18	La expresión de la sentencia de selección múltiple	La expresión de la instrucción <code>switch</code>	Todas las expresiones de la instrucción <code>swi</code>	Tipo	<input type="checkbox"/>
M19	Un nombre de función no se puede usar como	El identificador <code>%1</code> está declarado como un	El nombre de una función solamente se pued	Función	<input type="checkbox"/>
M20					
M21					
M22					
M23					
M24					

Opciones para añadir al final:

Ilustración 20 Errores semánticos del mismo lenguaje que la Ilustración 18

Cuando se crea una copia del lenguaje, los errores léxicos sí se copian bien, pero los semánticos y sintácticos no, dejando solo unas tablas iguales que las mostradas en las ilustraciones 19 y 20, pero vacías.

Los requisitos para este problema serán:

- **Requisito 1.3. Copia Completa del Lenguaje:** El sistema debe asegurar que, al copiar un lenguaje de un curso académico anterior, se copien todas las configuraciones de errores.
- **Requisito 2.3. Persistencia de Datos:** El proceso de copia debe garantizar que ningún dato se pierda durante la operación. Esto incluye tanto los errores preexistentes como aquellos que han sido modificados.
- **Requisito 3.3. Consistencia:** La apariencia y comportamiento de los botones y mensajes deben seguir los estilos utilizados en el resto del sistema DRACO.

3.4 Problema con el borrado de lenguajes

Similarmente al problema anterior, se observa que, para los errores semánticos y sintácticos, el borrado no ocurre mientras para los errores léxicos sí ocurre el borrado de manera correcta. Cuando se habla de que el borrado no ocurre, se refiere a que siguen presentes en la tabla de la base de datos específica para los errores donde se deben borrar, aunque no tengan ningún lenguaje asociado. Esto se pudo comprobar después de solucionar el problema anterior.

Los requisitos para este problema serían:

- **Requisito 1.4. Borrado completo del lenguaje:** El sistema debe asegurar que, al borrar un lenguaje de un curso académico anterior, se borre toda la configuración de errores.
- **Requisito 2.4. Persistencia de Datos:** El proceso de borrado debe garantizar que ningún dato fuera de los que se quieren borrar se pierda durante la operación.

- **Requisito 3.4. Consistencia:** La apariencia y comportamiento de los botones y mensajes deben seguir los estilos utilizados en el resto del sistema DRACO

3.5 Ampliación del funcionamiento del corrector de prácticas para el módulo Tabla de Símbolos

Dentro de los módulos de prácticas a gestionar por el profesorado, se encuentra el módulo de Tabla de Símbolos, que se imparte al principio del temario. Este módulo comprueba que los alumnos consigan hacer una tabla de símbolos bastante rudimentaria, donde solo se exigen los lexemas relevantes dentro de un fragmento de código, y sin tener más de una tabla, ya que solo se corrige basándose en la tabla de símbolos principal y no a ninguna tabla de funciones. Esto es así, porque en ese momento, solamente se encuentra implementado el Analizador Léxico, y este módulo solamente puede incluir los lexemas de los identificadores en la Tabla de Símbolos. El resto de información se introducirá cuando los alumnos implementen el Analizador Semántico, al final del curso.

El problema reside en que el corrector actualmente da por válido algunas entregas que, en verdad, no son correctas. Especialmente, la versión actual permite que las tablas de símbolos tengan atributos con valores, lo cual es algo que no se debe hacer aun en las prácticas y, por tanto, debe de dar por malo. Además, tampoco se detecta si hay más de una tabla en el fichero aportado por el estudiante.

Este problema tendrá los siguientes requisitos para su solución:

- **Requisito 1.5. Validación de la Tabla de Símbolos Principal:** El corrector debe asegurar que solo se permita una tabla de símbolos principal. Se debe implementar una función que identifique si hay más de una tabla de símbolos dentro del fichero entregado por el estudiante. Si es así, el corrector debe marcar la práctica como incorrecta.
- **Requisito 2.5. Detección de Atributos con Valores:** El sistema debe validar que ningún lexema en la tabla de símbolos tenga atributos con valores asignados. No obstante, se permite que en el fichero aparezcan otros atributos, pero siempre con un valor nulo.
- **Requisito 3.5. Retroalimentación Específica al Estudiante:** Cuando se marque una práctica como incorrecta, el sistema debe generar una retroalimentación detallada que especifique:
 - Si hay más de una tabla de símbolos.
 - Si se han encontrado atributos con valores no nulos y cuáles son.
- **Requisito 4.5. Personalización de la puntuación:** El profesor debe poder configurar cuánto penaliza cada tipo de error encontrado.
- **Requisito 5.5. Compatibilidad:** Los cambios han de ser compatibles con el comprobador semántico, y no deben interferir con su funcionamiento.

3.6 Corrección del nivel mínimo al crear una actividad

Los requisitos para la solución a este problema serán:

- **Requisito 1.6. Nivel Mínimo Predeterminado para Nuevas Actividades:** El sistema debe establecer el nivel mínimo predeterminado en 1 al crear una nueva actividad, no permitiendo asignar el nivel 0 a ninguna actividad.

- **Requisito 2.6. Consistencia:** La apariencia y comportamiento de los botones y mensajes deben seguir los estilos utilizados en el resto del sistema DRACO.
- **Requisito 3.6. Compatibilidad:** Ha de ser compatible con los filtros mencionados en anteriores problemas.

4 Solución

4.1 Solución a la ausencia de una opción de importar múltiples actividades en la sección de importar actividades

4.1.1 Diseño

Se comienza el diseño hablando de la arquitectura inicial de este segmento de Draco, mostrada en la ilustración 21.

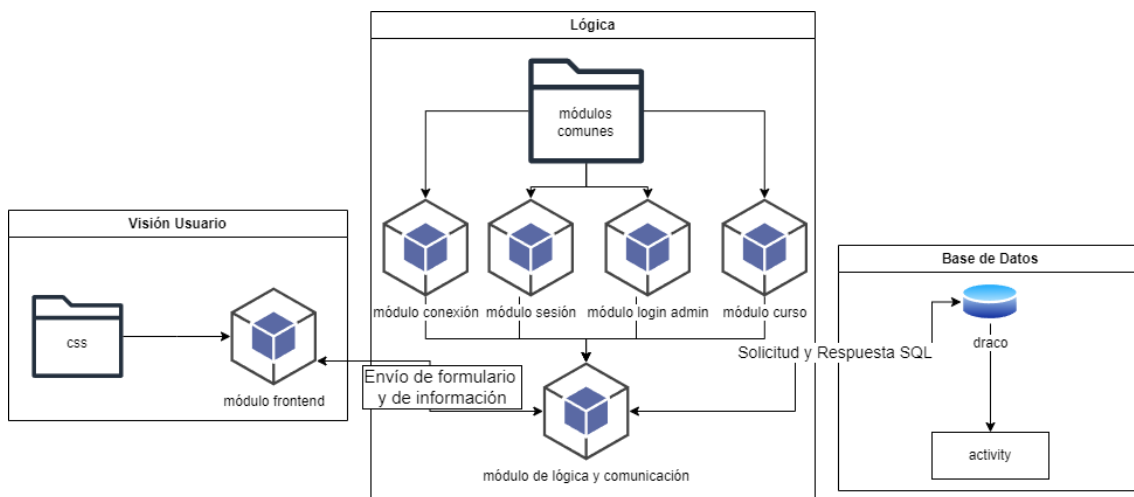


Ilustración 21 Arquitectura de los archivos relevantes a el problema actual

Se comienza hablando de la vista del usuario. Esta viene determinada por el módulo *frontend*, con ayuda de los distintos estilos definidos en código CSS.

Dentro del módulo *frontend*, hay un formulario que se envía al módulo de lógica y comunicación. Este módulo se encarga de toda la lógica de funcionamiento de esta página. Sus labores son dar la lista de actividades necesarias junto con toda su información al cargar la página, procesar los filtros y devolver la lista de actividades disponibles que cumplan el filtrado, y procesar las acciones de importar, ver y borrar actividades. Tiene algunas dependencias relevantes, pero que son muy comunes y genéricas a lo largo de las distintas vistas dentro de DRACO. Estas incluyen:

- El módulo de conexión permite la conexión a la base de datos.
- El módulo curso determina si se está en el módulo de PDL o de TDL.
- El módulo sesión determina cuando comienza una sesión de un usuario y determina cuándo se acaba.
- Finalmente, el módulo *login admin* redirige a la página de *login* del administrador en caso de que la sesión haya expirado.

Finalmente, se observa una comunicación con la base de datos draco. Específicamente, se comunica con la tabla *activity*, que contiene casi toda de la información relevante para las actividades, y toda la información relevante a este problema.

Para esta solución, no es necesario modificar la arquitectura, ya que no serán necesarias más dependencias con otros módulos externos, ni será necesario comunicarse con ninguna otra tabla dentro de la base de datos.

Esto lleva a considerar la modificación de los flujos de datos y lógicas dentro de los módulos principales: el módulo *frontend* y el módulo de lógica y comunicación.

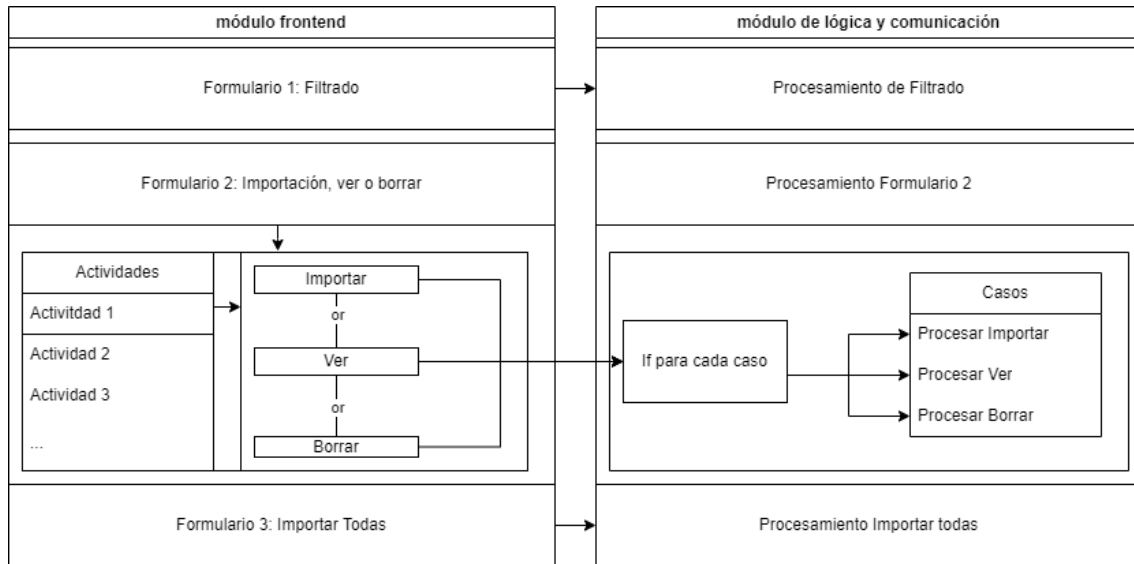


Ilustración 22 Lógica del módulo frontend y del módulo de lógica y comunicación

La lógica relevante a este problema de ambos módulos se muestra en la ilustración 22. Actualmente, toda la parte relevante a la importación se puede resumir en dos formularios: el 2 y el 3. Lo ideal sería tener un formulario cuyo funcionamiento sea un punto medio entre ambos, ya que el formulario 2 solo permite importar, ver o borrar cada actividad una a una y el formulario 3 permite importar todas las actividades a la vez. Actualmente, cada vez que se quiere importar, ver o borrar una actividad, se envía un formulario entero. Una solución sería implementar un mecanismo que permita la selección individual de cada actividad para su importación. Si esto se implementa con un mecanismo que permita seleccionar todas las actividades a la vez, se podría eliminar el formulario 3 y lograr el objetivo. Para ello habría que cambiar la lógica de ambos módulos.

En la ilustración 23 se mostrará una lógica nueva basada en la ilustración 22, pero incluyendo una solución que permitiría la selección de varias actividades, además de un funcionamiento general más eficiente y que consumirá menos recursos, ya que requerirá menos formularios a ser enviados.

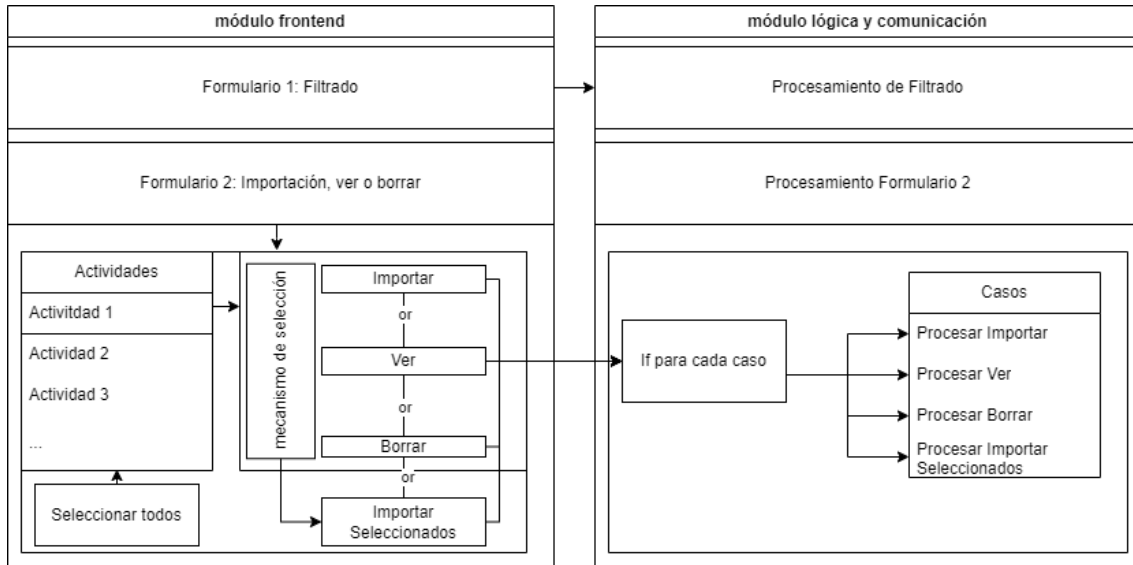


Ilustración 23 Diseño de lógica final

Esta lógica expuesta en la ilustración 23 permitiría una solución ideal para este problema. La solución mantiene el funcionamiento del formulario 2 expuesto en la ilustración 22, pero sustituye el formulario 3 por una extensión en el funcionamiento del formulario 2. Esta extensión permitiría una nueva señal en el envío del formulario, donde se envían a importar todas las actividades seleccionadas. Además, da la opción de seleccionar todas las actividades. También se modificaría el módulo de lógica y comunicación, permitiendo ahora procesar la nueva opción del formulario 2.

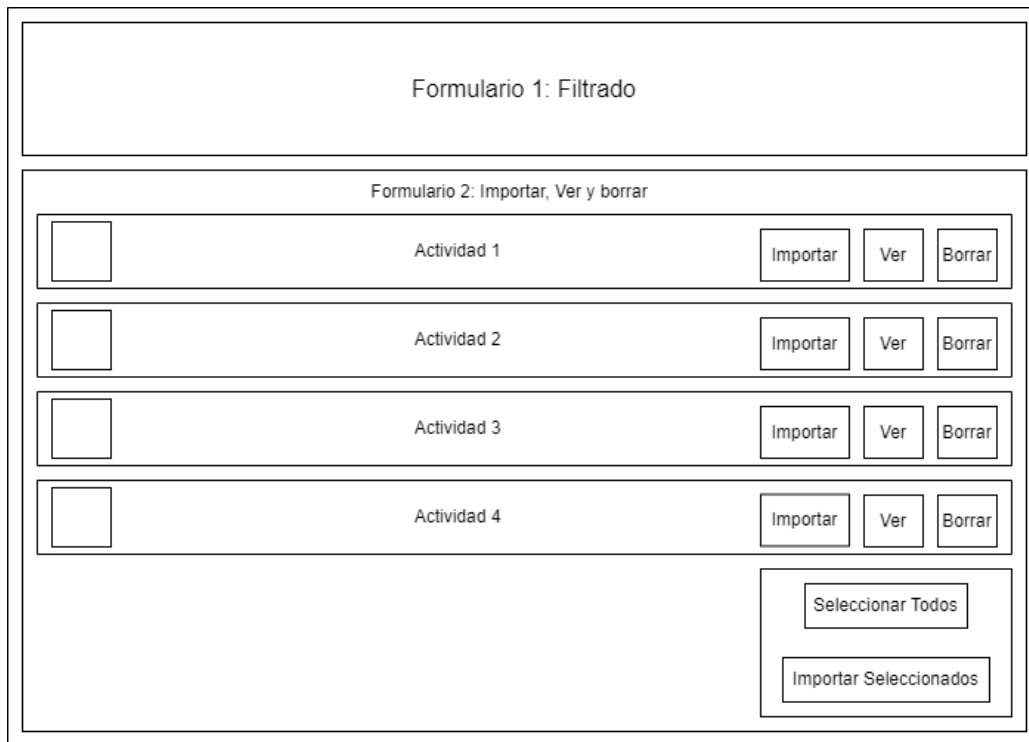


Ilustración 24 Propuesta de vista de la página con la solución

La ilustración 24 muestra un ejemplo de cómo se podría organizar la página de manera visual. Se mostraría cada actividad con una opción de selección, en este caso representada por un cuadro blanco, junto con opciones de importar, ver y borrar. Finalmente, hay un cuadro separado de las actividades que permite seleccionar todas las actividades e importar las actividades seleccionadas.

4.1.2 Implementación

A continuación, se empieza la sección de implementación para la página de importar actividades. Para ello se comenzará hablando de la solución, para lo cual parece imperativo hablar de los ficheros y documentos que son relevantes para esta sección.

En este caso, se tienen dos documentos principales: ImportActivity.php e ImportActivityAction.php.

Se discutirán ambos códigos tal y como están ahora, comenzando con importactivity.php:

```
<h3>Seleccione la actividad a gestionar:</h3>
<?php if($noquiz):?>
    <p>No hay actividades disponibles en este momento</p>
<?php else:?>
    <div class="table">
        <div class="thead">
            <div class="tr">
                <div class="td">Título</div>
                <div class="td">Tipo</div>
                <div class="td">Estado</div>
                <div class="td">Puntos Mínimos</div>
                <div class="td">Puntos Máximos</div>
                <div class="td">Fecha Inicio</div>
                <div class="td">Fecha Fin</div>
            </div>
        </div>
        <div class="tbody">
<?php foreach($available_act as $activity):?>
<?php
if ($activity["status"] == "Active"){
    $action = "<input name=\"submit\" type=\"submit\" value=\"Ver\">
        <input name= \"submit\" type=\"submit\" value=\"Borrar\">";
    $activity_status_display = "Activa";
}
else if ($activity["status"] == "Inactive") {
    $action = "<input name=\"submit\" type=\"submit\" value=\"Ver\">
        <input name= \"submit\" type=\"submit\" value=\"Borrar\">";
    $activity_status_display = "Inactiva";
}
else if ($activity["status"] == "Incomplete") {
    $action = "<input name= \"submit\" type=\"submit\" value=\"Borrar\">";
    $activity_status_display = "Incompleta";
}
else if ($activity["status"] == "Editing") {
    $action = "<input name= \"submit\" type=\"submit\" value=\"Borrar\">";
    $activity_status_display = "Editando";
}
else if ($activity["status"] == "Auto") {
    $action = "<input name=\"submit\" type=\"submit\" value=\"Ver\">
        <input name= \"submit\" type=\"submit\" value=\"Borrar\">";
    $activity_status_display = "Auto";
}
}
```

```

// Display the activity type name in a more understandable manner
switch ($activity["type"]) {
    case "Radio_Check_Type":
        $activity_type_display = "Múltiples respuestas";
        break;
    case "Short_Type":
        $activity_type_display = "Respuesta corta";
        break;
    case "Sequential_Type":
        $activity_type_display = "Secuencial";
        break;
    case "Long_Type":
        $activity_type_display = "Respuesta larga";
        break;
    case "Survey_Long":
        $activity_type_display = "Encuesta de Respuesta larga";
        break;
    case "Exam_Type":
        $activity_type_display = "Examen";
        break;
    case "review":
        $activity_type_display = "Revisión";
        break;
    case "Order_Type":
        $activity_type_display = "Ordenar";
        break;
    case "Pair_Type":
        $activity_type_display = "Emparejar";
        break;
    case "faulty":
        $activity_type_display = "Defectuoso";
        break;
    case "coop":
        $activity_type_display = "Cooperativa";
        break;
}
?>
<form class="tr" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
    method="post" >
    <input type="hidden" name="back" value="manageact">
    <input type="hidden" name="activity_id" value="<?php echo
        $activity["id"] ?>">
    <div class="td"><?php echo $activity["title"]?></div>
    <div class="td"><?php echo $activity_type_display?></div>
    <div class="td"><?php echo $activity_status_display?></div>
    <div class="td"><?php echo $activity["minpoints"]?></div>
    <div class="td"><?php echo $activity["maxpoints"]?></div>
    <div class="td"><?php echo $activity["idate"]?></div>
    <div class="td"><?php echo $activity["ddate"]?></div>
    <input name="submit" type="submit" value="Importar">
    <?php echo $action?>
</form>
<?php endforeach;?>
</div>
</div>
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
    method="post" id="filterform">
    <button class="boton" name="changeAll" value="import">
        Importar todas</button>
    <input type="hidden" name="back" value="filter">
</form>
<?php endif;?>
<br>

```

```

    <a href="../../manageActivities.php">Volver</a>
</section>

</body>
</html>

```

Este fragmento de `ImportActivity.php` comienza con un condicional. En caso de que no haya actividades a gestionar, se muestra un mensaje en pantalla que lo comunica. En caso de que sí haya actividades a gestionar, se comienza mostrando los atributos de las actividades, en este caso siendo: Título, Tipo, Estado, Puntos Mínimos, Puntos Máximos, Fecha Inicio y Fecha Fin.

Después se empieza un bucle que iterará sobre cada actividad. Dentro de este bucle, se comienza escribiendo el código que se añadirá a cada actividad dependiendo de su estado. También se listan los tipos de actividades existentes para asegurar que las actividades listadas estén entre ellas. Para finalizar el bucle, se crea un formulario con el ID de la actividad. Este formulario primero muestra toda la información requerida de cada actividad, añadiendo un botón de “Importar” que envía cada formulario, enviando una señal al `ImportActivityAction.php`.

Después de este formulario, el código termina con otro formulario que consiste en un solo botón para importar todas las actividades de una en una. Además, también hay un botón que permite volver a la página de `ManageActivities.php`, que, como se menciona anteriormente, es la página donde se muestran las actividades importadas.

Como la parte estética no es especialmente relevante de cara a `ImportActivityAction.php`, solo se mostrarán y explicarán las partes más relevantes del código de este archivo para explicar el actual funcionamiento de esta parte de la página.

Las peticiones a procesar por `ImportActivityAction.php` se pueden resumir en dos partes principales: El filtrado y Ver, Borrar o Importar Actividades.

- **El filtrado:** Este fragmento de código comienza determinando si se ha hecho un filtrado o no:

```
if (isset($_POST["back"]) && $_POST["back"] == "filter"){
```

Después, crea el *query* que dará la lista de actividades inicial, con una consulta tal que:

```
$query = "SELECT * FROM Activity WHERE `archived` = 1 AND `course` =
'".$course."'";
```

Después va determinando qué filtros han sido solicitados en el formulario, y modifica la consulta acordeamente. Aquí se muestra un ejemplo de si se hubiese filtrado por título:

```
if(!empty($_POST["title"])){
    $query .= " AND title LIKE \"%".$_POST["title"]."\"";
}
```

A partir de ahí, se devuelve el resultado de la consulta, el cual se mostrará en la pantalla, omitiendo todas las actividades que no cumplieran con los filtros.

- **Ver, Borrar o Importar Actividades:** Estas tres opciones se procesan a partir de una señal *manageact*, que después determina qué acción tomar a partir del valor de una señal *submit*. Aquí se muestra el fragmento de código:

```
elseif (isset($_POST["back"]) && $_POST["back"] == "manageact"){
    if ($_POST["submit"] == "Ver") {
        $location =
            "../do/doActivity.php?activity=".$_POST["activity_id"]."&t=0";
        header("Location: " . $location);
        die();
    }
    elseif ($_POST["submit"] == "Borrar") {
        $location =
            "../erase/confirmEraseActivity.php?activity_id=".$_POST["activity_id"];
        header("Location: " . $location);
        die();
    }
    else {
        $db_link = connect_db();
        sql_query = "UPDATE Activity SET archived = 0 WHERE
            `id`=".$_POST["activity_id"];
        mysqli_query($db_link, $sql_query);
        disconnect_db($db_link);
        $db_link = connect_db();
        $sql_query = "SELECT * FROM Activity WHERE `archived` = 1 AND `course` =
            '".$_course.'" ORDER BY idate ASC";
        $query_result = mysqli_query($db_link, $sql_query);
        $available_act = my_fetch_all($query_result, MYSQLI_ASSOC);
        count($available_act) > 0 ? $noquiz = 0 : $noquiz = 1;
        mysqli_free_result($query_result);
        disconnect_db($db_link);
    }
}
```

Esto muestra que ver y borrar se limitan a hacer su funcionamiento por medio de otros ficheros que no son relevantes para este problema. También se ve que el método de importar se limita a una modificación de una columna de la actividad en la base de datos, llamada *archived* que tiene valor 0 si está importada y 1 si está lista para importar.

Habiendo explicado bien el estado inicial del código se hablará de la solución en sí. Una manera de poder seleccionar solo cierto número de las actividades es hacer una opción de *checkboxes*, tal y como se mencionó en el diseño. La idea sería tener un *checkbox* al lado de cada actividad que puede ser seleccionado, seguido de un botón de importar seleccionados que permite importar solo la selección.

También puede añadirse un botón de “seleccionar todos” que permite seleccionar (y deseleccionar) todas las actividades, permitiendo borrar el botón de importar todos.

Los *checkboxes* son una función que existe dentro de HTML; y para enviar esta información de qué *checkboxes* han sido seleccionados, se necesita un formulario. El problema reside en que todas las actividades en sí mismas incluyen un formulario, que se envía dependiendo si se le da al botón ver,

importar o borrar, tal y como se vio antes. Por tanto, se necesitaría meter un formulario dentro de otro formulario.

Los formularios en HTML no pueden estar anidados porque el estándar HTML define que un elemento `<form>` debe representar un único formulario independiente, y permitir la anidación causaría ambigüedad en el envío de datos, ya que no estaría claro qué formulario debería procesar los elementos como campos de entrada. Además, los navegadores y agentes de usuario podrían interpretar de manera inconsistente la estructura y el comportamiento de los formularios anidados, lo que dificultaría garantizar una experiencia de usuario y funcionalidad uniforme. Por estas razones, se prohíbe la anidación de formularios, para mantener la claridad y la robustez del diseño web.

Aun así, se puede hacer de otra manera: Modificar el formulario original y el `ImportActivityAction.php` para poder procesar también las nuevas opciones.

Este algoritmo se podría realizar de la siguiente manera:

```
<div class="td">
  <input class="checkbox_import_act" type="checkbox" name="sel_<?php echo
    $index; ?>">
  <input type="hidden" name="activity_id_<?php echo $index; ?>" value="<?php
    echo $activity['id']; ?>">
</div>
```

Se comienza añadiendo al bucle la creación de un *checkbox* por cada una de las actividades.

```
<div class="td">
  <button type="submit" name="submit" value="Ver_<?php echo
    $index; ?>">Ver</button>
  <button type="submit" name="submit" value="Borrar_<?php echo
    $index; ?>">Borrar</button>
  <button type="submit" name="submit" value="Importar_<?php echo
    $index; ?>">Importar</button>
</div>
```

También se cambian los botones de importar, ver y borrar originales, para que se procesen correctamente en el `ImportActivityAction.php`.

Fuera del bucle, se crea un código HTML que siempre aparecerá en pantalla en una esquina, que permitirá seleccionar todas las actividades, deseleccionarlas e importar las seleccionadas:

```
<div class="fixed">
  <h3>Importar actividades seleccionadas</h3>
  <label for="sel_all">Seleccionar todo</label>
  <input type="checkbox" id="sel_all"><br><br>
  <button type="submit" name="submit" value="ImportarSeleccionadas">Importa
    seleccionadas</button>
</div>
```

Este fragmento va seguido del script que permite la selección de los *checkboxes*

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
  $(document).ready(function() {
    $("#sel_all").click(function() {
      $(".checkbox_import_act").prop("checked", this.checked);
    });
  });
</script>
```

```
</script>
```

Este *script* utiliza la biblioteca jQuery para implementar la funcionalidad de "seleccionar todo" en un grupo de *checkboxes*. Se incluye la referencia a la biblioteca jQuery mediante el enlace a la CDN de Google para jQuery versión 3.3.1, asegurando así que las funciones de jQuery estén disponibles en el proyecto.

Cuando el documento HTML se carga completamente (`$(document).ready()`), el *script* asocia un evento al *checkbox* con el ID `sel_all`. Este evento se activa cuando se hace clic en dicho *checkbox*. Dentro del evento, el estado de selección (*checked*) de todos los *checkboxes* que tienen la clase `checkbox_import_act` se sincroniza con el estado del *checkbox* `sel_all`. Esto se logra utilizando el método `prop()` de jQuery, que establece el valor de la propiedad *checked* de los *checkboxes* correspondientes.

Todos estos cambios también obligan a modificar `importActivityAction.php`. Las modificaciones que se necesitan serán para poder tratar las nuevas señales del formulario enviados para que pueda realizar las acciones anteriores (ver, borrar e importar individualmente), además de poder importar los seleccionados.

```
if (strpos($action, "Ver_") !== false) {
    $index = explode("_", $action)[1];
    $location = "../do/doActivity.php?activity=" .
        $_POST["activity_id_$index"] . "&t=0";
    header("Location: " . $location);
    die();
}

if (strpos($action, "Borrar_") !== false) {
    $index = explode("_", $action)[1];
    $location = "../erase/confirmEraseActivity.php?activity_id=" .
        $_POST["activity_id_$index"];
    header("Location: " . $location);
    die();
}

if (strpos($action, "Importar_") !== false) {
    // Nueva gestión del botón Importar individual
    $index = explode("_", $action)[1];
    $activity_id = intval($_POST["activity_id_$index"]);
    $db_link = connect_db();
    $sql_query = "UPDATE Activity SET archived = 0 WHERE id = $activity_id";
    mysqli_query($db_link, $sql_query);
    disconnect_db($db_link);
}
```

Se explica a continuación el funcionamiento de los botones originales. No hay mucha diferencia en su funcionamiento excepto en cómo procesan la petición con la función `explode`. La función `explode` en este código se utiliza para dividir una cadena en partes más pequeñas, basándose en un delimitador especificado. En este caso, el delimitador es el carácter de subrayado ("_"), y permite obtener el id de la actividad en cada caso.

```
if ($action === "ImportarSeleccionadas") {
    $db_link = connect_db();
    // Importar actividades seleccionadas
    foreach ($_POST as $key => $value) {
        if (strpos($key, "sel_") === 0) {
            $index = explode("_", $key)[1];
            $activity_id = intval($_POST["activity_id_$index"]);
        }
    }
}
```

```

        $sql_query = "UPDATE Activity SET archived = 0 WHERE id =
        $activity_id";
        mysqli_query($db_link, $sql_query);
    }
}
disconnect_db($db_link);
}

```

Finalmente, se muestra el Código de importar seleccionados, que también vale para la selección de todas las actividades. La forma de funcionar es idéntica a la de importar individual, pero lo realiza en bucle en todas las actividades seleccionadas, separando la cadena en partes con `explode()`, como se explicó anteriormente.

Una vez implementados estos cambios en el código, estos fueron los resultados:

GESTIÓN DE ACTIVIDADES								
Puede filtrar actividades que cumplan ciertas características								
Título	Tipo de Actividad	Estado	Puntos Min	Puntos Max	Día Inicio	Día Fin		
<input type="text"/>	<input type="text" value="Seleccione"/>	<input type="text" value="Seleccione"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="2024-12-28"/>	<input type="text" value="2025-01-04"/>	<input type="button" value="Filtrar"/>	
Seleccionar	Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Acciones
<input type="checkbox"/>	Página de la asignatura	Múltiples respuestas	Inactiva	5	25	2023-09-08 22:22:00	2023-09-27 13:57:00	Ver Borrar Importar
<input type="checkbox"/>	Termina la frase	Respuesta corta	Inactiva	5	50	2023-09-14 13:13:00	2023-09-30 23:32:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 1	Múltiples respuestas	Inactiva	5	44	2023-09-19 11:11:00	2023-10-03 23:34:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 2	Múltiples respuestas	Inactiva	5	44	2023-09-20 13:00:00	2023-10-04 22:00:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 3	Múltiples respuestas	Inactiva	5	44	2023-09-21 00:00:00	2023-10-05 22:01:00	Ver Borrar
<input type="checkbox"/>	Tokens empresa - 4	Múltiples respuestas	Inactiva	5	44	2023-09-22 15:55:00	2023-10-05 15:00:00	Ver Borrar
<input type="checkbox"/>	Tokens empresa - 5	Múltiples	Inactiva	5	44	2023-12-17	2023-12-17	Ver Borrar

Seleccionar todo

Ilustración 25 Resultado de implementaciones de checkboxes

La ilustración 25 muestra todo el código implementado de los *checkboxes*. Para que todo quedase correctamente, se tuvo que meter algunos cambios, como una columna con el título de seleccionar:

```

<div class="thead">
    <div class="tr">
        <div class="td">Seleccionar</div>
        <div class="td">Titulo</div>
        <div class="td">Tipo</div>
        <div class="td">Estado</div>
        <div class="td">Puntos Mínimos</div>
        <div class="td">Puntos Máximos</div>
        <div class="td">Fecha Inicio</div>
        <div class="td">Fecha Fin</div>
        <div class="td">Acciones</div>
    </div>
</div>

```

Siendo estas las columnas finales de información a ser mostradas en pantalla. Además, para que el cuadro de importar seleccionados no moleste en la selección de las acciones de la última actividad, se introdujeron unos saltos de línea al final del código, que en el código se escribirían como `
`.

4.1.3 Pruebas

Para probar el correcto funcionamiento hay que comprobar todas las cosas que se han modificado e implementado hasta el momento:

4.1.3.1 Botón Ver

Para esta prueba, se hará clic en el botón Ver en distintas actividades para comprobar si cada botón Ver lleva correctamente a la actividad correspondiente, permite realizar la actividad y permite volver a la lista de actividades a importar sin ninguna modificación:

Tokens empresa - 1

A la hora de diseñar el conjunto de *tokens* de un Analizador Léxico para un fichero de personal de una empresa que contiene números de teléfono, nombres y apellidos, departamento y sueldo (cuyo formato es una serie de dígitos seguidos del signo de euro, por ejemplo, 3000€), ¿es correcto el siguiente diseño de los *tokens*: <teléfono, lexema>, <palabra, lexema>, <Cod.Departamento, ->, <Sueldo, lexema>

Información: Solo una respuesta es correcta.

Tiempo límite: 2 minutos 22 segundos
Desde las 13:38:39
Hasta las 13:41:01

No es correcto

Es correcto

Finalizar

Ilustración 26 Pantalla mostrada al dar al botón ver. En este caso para la actividad Tokens empresa - 1

Tu respuesta:

Es correcto

Incorrecta.

Explicación:
No, hay que calcular el valor del sueldo y generar el token <Sueldo, valor>

Respuesta correcta:

No es correcto

Explicación:
Están mal pues hay que calcular el valor del sueldo y generar el token <Sueldo, valor>

Has obtenido el mínimo de puntos.

Puntos de la actividad: 5 puntos.
Bonus por tiempo: 6 puntos.
Has obtenido: 11 puntos.

Volver a importar actividades

Ilustración 27 Resultado de la actividad realizada al darle a ver

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Puntos Min	Puntos Max	Día Inicio	Día Fin		
<input type="text"/>	Seleccione ▾	Seleccione ▾	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	Filtrar	
Seleccionar	Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Acciones
<input type="checkbox"/>	Página de la asignatura	Múltiples respuestas	Inactiva	5	25	2023-09-08 22:22:00	2023-09-27 13:57:00	Ver Borrar Importar
<input type="checkbox"/>	Termina la frase	Respuesta corta	Inactiva	5	50	2023-09-14 13:13:00	2023-09-30 23:32:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 1	Múltiples respuestas	Inactiva	5	44	2023-09-19 11:11:00	2023-10-03 23:34:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 2	Múltiples respuestas	Inactiva	5	44	2023-09-20 13:00:00	2023-10-04 22:00:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 3	Múltiples respuestas	Inactiva	5	44	2023-09-21 00:00:00	2023-10-05 22:01:00	Ver Borrar
<input type="checkbox"/>	Tokens empresa - 4	Múltiples respuestas	Inactiva	5	44	2023-09-22 15:55:00	2023-15:00	Ver Borrar
<input type="checkbox"/>	Tokens empresa - 5	Múltiples	Inactiva	5	44	2023-12-17	2023-	Ver Borrar

Importar actividades seleccionadas
 Seleccionar todo
 Importar seleccionadas

Ilustración 28 Vuelta correcta a importar actividades

Esta prueba permitió detectar varios fallos. Una razón por la que se pudo ver que el código no funcionaría con formularios anidados fue esta prueba, ya que los formularios anidados no permitían que los botones de las acciones ni los *checkboxes* funcionaban, ya que había conflicto entre los distintos formularios. Eventualmente se llegó a la solución implementada, que permitió que las pruebas fuesen satisfactorias. Como se puede ver con el ejemplo dado en las ilustraciones 26, 27 y 28, además de haber realizado esa misma prueba con múltiples actividades, se da esta prueba por aprobada y el funcionamiento del botón como correcto.

4.1.3.2 Botón Borrar

Para esta prueba, se seleccionará el botón borrar en distintas actividades para ver si cada botón borrar borra correctamente la actividad correspondiente:

<input type="checkbox"/>	Tabla Descendente	Múltiples respuestas	Inactiva	3	44	2023-12-03 11:11:00	2023-12-13 10:10:00	Ver Borrar Importar
<input type="checkbox"/>	Análisis de conflictos	Múltiples respuestas	Inactiva	3	40	2023-12-02 11:11:00	2023-12-12 12:34:00	Ver Borrar Importar
<input type="checkbox"/>	Gaudeamus	Ordenar	Inactiva	5	33	2024-01-13 12:00:00	2024-01-21 23:45:00	Ver Borrar Importar
<input type="checkbox"/>	Encuestas	Múltiples respuestas	Inactiva	1	20	2023-12-14 14:14:00	2023-12-22 12:34:00	Ver Borrar Importar
<input type="checkbox"/>	Mejorar Draco	Múltiples respuestas	Inactiva	1	21	2024-01-14 13:18:00	2024-01-25 09:34:00	Ver Borrar Importar
<input type="checkbox"/>	Ejemplo	Respuesta larga	Auto	0	10	2024-09-11 00:00:00	2025-09-11 00:00:00	Ver Borrar Importar
<input type="checkbox"/>	Hola	Múltiples respuestas	Auto	0	10	2024-10-02 00:00:00	2024-11-02 00:00:00	Ver Borrar Importar
<input type="checkbox"/>	asdfsdfa	Desconocido	Activa	0	12	2024-10-08 00:00:00	2024-11-08 00:00:00	Ver Borrar Importar

Importar actividades seleccionadas
 Seleccionar todo
 Importar seleccionadas

Volver

Ilustración 29 Lista de actividades para comparar las actividades después del borrado



Ilustración 30 Comprobación de si se quiere borrar la actividad, en este caso la actividad "Hola"

		respuestas				11:11:00	12:34:00	Importar
<input type="checkbox"/>	DRACO	Desconocido	Inactiva	1	33	2023-12-07 11:00:00	2023-12-27 23:45:00	Ver Borrar Importar
<input type="checkbox"/>	Tabla Descendente	Múltiples respuestas	Inactiva	3	44	2023-12-03 11:11:00	2023-12-13 10:10:00	Ver Borrar Importar
<input type="checkbox"/>	Análisis de conflictos	Múltiples respuestas	Inactiva	3	40	2023-12-02 11:11:00	2023-12-12 12:34:00	Ver Borrar Importar
<input type="checkbox"/>	Gaudeamus	Ordenar	Inactiva	5	33	2024-01-13 12:00:00	2024-01-21 23:45:00	Ver Borrar Importar
<input type="checkbox"/>	Encuestas	Múltiples respuestas	Inactiva	1	20	2023-12-14 14:14:00	2023-12-22 12:34:00	Ver Borrar Importar
<input type="checkbox"/>	Mejorar Draco	Múltiples respuestas	Inactiva	1	21	2024-01-14 13:18:00	2024-01-25 09:34:00	Ver Borrar Importar
<input type="checkbox"/>	Ejemplo	Respuesta larga	Auto	0	10	2024-09-11 00:00:00	2025-09-11 00:00:00	Ver Borrar Importar
<input type="checkbox"/>	asdfsdfa	Desconocido	Activa	0	12	2024-10-08 00:00:00	2024-11-08 00:00:00	Ver Borrar Importar

Volver

Importar actividades seleccionadas
 Seleccionar todo
 Importar seleccionadas

Ilustración 31 Se comprueba si la actividad "Hola" sigue en la lista después de ser borrada

Como se ha podido comprobar por el ejemplo mostrado en las ilustraciones 29, 30 y 31, además de múltiples pruebas realizadas, el botón borrar funciona correctamente.

4.1.3.3 Botón Importar

Para esta prueba se seleccionará en el botón importar de varias actividades y se comprobará si se han importado correctamente visitando la página de manageActivities y comprobando que se muestran en pantalla.

La ilustración 32 mostrará la pantalla inicial, dando algo visual con lo que comparar una vez se ejecute la importación. La ilustración 33 muestra la misma página de la ilustración 32, pero después de haber realizado una importación. Finalmente, la ilustración 34 cambia la vista a todas las actividades que han sido importadas, también conocido como la página de manejar actividades. El ejemplo mostrado en las ilustraciones 32, 33 y 34 muestran el correcto funcionamiento del botón de importar.

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Puntos Min	Puntos Max	Día Inicio	Día Fin		
<input type="text"/>	Seleccione	Seleccione	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	Filtrar	
Seleccionar	Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Acciones
<input type="checkbox"/>	Tokens empresa - 3	Múltiples respuestas	Inactiva	5	44	2023-09-21 00:00:00	2023-10-05 22:01:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 4	Múltiples respuestas	Inactiva	5	44	2023-09-22 15:55:00	2023-10-06 15:00:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 5	Múltiples respuestas	Inactiva	5	44	2023-12-17 10:00:00	2023-12-27 10:00:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 7	Múltiples respuestas	Inactiva	5	44	2023-11-19 21:00:00	2023-11-29 13:55:00	Ver Borrar Importar
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	Ver Borrar
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	5	50	2023-10-14 19:55:00	2023-05-05	Ver Borrar Importar
<input type="checkbox"/>	Lenquaje sin declaración - 3	Múltiples	Inactiva	5	50	2023-10-19	2023-	Ver Borrar Importar

Importar actividades seleccionadas
 Seleccionar todo
 Importar seleccionadas

Ilustración 32 Lista inicial de actividades a importar

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Puntos Min	Puntos Max	Día Inicio	Día Fin		
<input type="text"/>	Seleccione	Seleccione	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	Filtrar	
Seleccionar	Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Acciones
<input type="checkbox"/>	Tokens empresa - 4	Múltiples respuestas	Inactiva	5	44	2023-09-22 15:55:00	2023-10-06 15:00:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 5	Múltiples respuestas	Inactiva	5	44	2023-12-17 10:00:00	2023-12-27 10:00:00	Ver Borrar Importar
<input type="checkbox"/>	Tokens empresa - 7	Múltiples respuestas	Inactiva	5	44	2023-11-19 21:00:00	2023-11-29 13:55:00	Ver Borrar Importar
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	Ver Borrar Importar
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	Ver Borrar
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	5	50	2023-10-19 02:00:00	2023-10-15	Ver Borrar Importar
<input type="checkbox"/>	Lenquaje sin declaración - 4	Múltiples	Inactiva	5	50	2023-12-16	2023-	Ver Borrar Importar

Importar actividades seleccionadas
 Seleccionar todo
 Importar seleccionadas

Ilustración 33 Pantalla después de importar "Token empresa - 3". Ya no está en esta página

Título	Tipo de Actividad	Estado	Puntos Min	Puntos Max	Día Inicio	Día Fin	Acciones
Música en Clase práctica TDS	Respuesta única	Activa	Sin definir	3 - 22	24-11-2022 17:00	30-11-2022 21:22	Revisar Editar Desactivar Auto Ver
Fiesta	Respuesta única	Activa	Sin definir	3 - 11	26-01-2023 17:59	27-01-2023 17:00	Revisar Editar Desactivar Auto Ver
Bienvenida	Respuesta única	Activa	Sin definir	0 - 50	06-09-2023 08:58	26-09-2023 23:32	Revisar Editar Desactivar Auto Ver
Mascarilla	Respuesta única	Activa	Sin definir	2 - 11	07-09-2023 18:00	26-09-2023 01:23	Revisar Editar Desactivar Auto Ver
Página de la asignatura	Respuesta única	Inactiva	Sin definir	5 - 25	08-09-2023 22:22	27-09-2023 13:57	Revisar Editar Activar Borrar Ver
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	Revisar Editar Activar Borrar Ver
Analizadores	Respuesta corta	Inactiva	Sin definir	5 - 40	12-09-2023 13:55	28-09-2023 09:55	Revisar Editar Activar Borrar Ver
Token	Respuesta única	Inactiva	Scanneator	10 - 75	14-09-2023 11:45	29-09-2023 23:11	Revisar Editar Activar Borrar Ver
Termina la frase	Respuesta corta	Inactiva	Scanneator	5 - 50	14-09-2023 13:13	30-09-2023 23:32	Revisar Editar Activar Borrar Ver
Función del Analizador Léxico	Respuesta múltiple	Inactiva	Scanneator	10 - 75	16-09-2023 21:00	01-10-2023 13:13	Revisar Editar Activar Borrar Ver
Tokens empresa - 1	Respuesta única	Inactiva	Scanneator	5 - 44	19-09-2023 11:11	03-10-2023 23:34	Revisar Editar Activar Borrar Ver
Tokens empresa - 2	Respuesta única	Inactiva	Scanneator	5 - 44	20-09-2023 13:00	04-10-2023 22:00	Revisar Editar Activar Borrar Ver
Tokens empresa - 3	Respuesta única	Inactiva	Scanneator	5 - 44	21-09-2023 00:00	05-10-2023 22:01	Revisar Editar Activar Borrar Ver
Tokens del programa	Respuesta múltiple	Inactiva	Scanneator	5 - 44	22-09-2023 19:00	07-10-2023 11:22	Revisar Editar Activar Borrar Ver
Tokens empresa - 6	Respuesta única	Inactiva	Scanneator	5 - 44	09-10-2023 15:00	19-10-2023 10:00	Revisar Editar Activar Borrar Ver
Tokens varios - 2	Respuesta múltiple	Inactiva	Scanneator	10 - 75	09-12-2023 12:05	19-12-2023 22:00	Revisar Editar Activar Borrar Ver
Tokens varios	Respuesta múltiple	Inactiva	Scanneator	10 - 70	19-12-2023 23:05	29-12-2023 05:05	Revisar Editar Activar Borrar Ver
Acciones semánticas del Léxico	Respuesta múltiple	Inactiva	Scanneator	5 - 60	28-12-2023 16:00	08-01-2024 08:00	Revisar Editar Activar Borrar Ver
Comida	Encuesta larga	Activa	Sin definir	0 - 66	09-12-2024 00:00	09-12-2025 00:00	Revisar Editar Desactivar Auto Ver

Ilustración 34 Se comprueba que sí está en manageActivities.php

4.1.3.4 Comprobar que el botón de Seleccionar Todo funciona correctamente

Se seleccionará el botón de seleccionar todo y se comprobará que todas las actividades han sido seleccionadas.

GESTIÓN DE ACTIVIDADES									
Puede filtrar actividades que cumplan ciertas características									
Título	Tipo de Actividad	Estado	Puntos Min	Puntos Max	Día Inicio	Día Fin			
<input type="text"/>	<input type="text" value="Seleccione"/>	<input type="text" value="Seleccione"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="2024-12-28"/>	<input type="text" value="2025-01-04"/>	<input type="button" value="Filtrar"/>		
Seleccionar	Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Acciones	
<input type="checkbox"/>	Tokens empresa - 4	Múltiples respuestas	Inactiva	5	44	2023-09-22 15:55:00	2023-10-06 15:00:00	<input type="button" value="Ver"/>	<input type="button" value="Borrar"/>
<input type="checkbox"/>	Tokens empresa - 5	Múltiples respuestas	Inactiva	5	44	2023-12-17 10:00:00	2023-12-27 10:00:00	<input type="button" value="Ver"/>	<input type="button" value="Borrar"/>
<input type="checkbox"/>	Tokens empresa - 7	Múltiples respuestas	Inactiva	5	44	2023-11-19 21:00:00	2023-11-29 13:55:00	<input type="button" value="Ver"/>	<input type="button" value="Borrar"/>
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	<input type="button" value="Ver"/>	<input type="button" value="Borrar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	<input type="button" value="Ver"/>	<input type="button" value="Borrar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	5	50	2023-10-19 02:00:00	2023-10-15	<input type="button" value="Importar actividades seleccionadas"/>	
<input type="checkbox"/>	Lenguaje sin declaración - 4	Múltiples	Inactiva	5	50	2023-12-16	2023-	<input type="button" value="Importar seleccionadas"/>	

Ilustración 35 Estado inicial de la página

<input checked="" type="checkbox"/>	¡Feliz Año Nuevo!	Respuesta corta	Inactiva	2	22	2024-01-01 00:00:00	2024-01-01 11:11:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Acción Semántica CONCATENAR	Múltiples respuestas	Inactiva	5	50	2023-10-28 07:33:00	2023-11-08 02:59:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Análisis Semántico de expresiones	Múltiples respuestas	Inactiva	5	70	2023-11-25 19:00:00	2023-12-05 23:00:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Análisis Semántico: expresiones y operadores	Múltiples respuestas	Inactiva	5	85	2023-11-30 00:02:00	2023-12-10 12:00:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Monedas para la práctica	Múltiples respuestas	Inactiva	1	5	2023-11-04 00:00:00	2023-11-15 00:00:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Semántico y Errores	Múltiples respuestas	Inactiva	5	42	2023-12-31 22:10:00	2024-01-10 12:34:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Análisis Semántico de Sentencias	Múltiples respuestas	Inactiva	10	90	2023-11-26 17:22:00	2023-12-06 21:23:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Continuación	Múltiples respuestas	Inactiva	3	20	2024-01-15 10:01:00	2024-01-23 23:11:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Continuación - web	Múltiples respuestas	Inactiva	2	23	2024-01-17 12:23:00	2024-01-25 10:10:00	Ver Borrar Importar
<input checked="" type="checkbox"/>	Síntesis	Ordenar	Inactiva	3	22	2024-01-16 00:30:00	2024-01-23 23:34	Importar actividades seleccionadas Seleccionar todo <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Orden de tokens	Ordenar	Inactiva	5	33	2023-12-21 20:20:00	2023-12-31 23:32	Importar seleccionadas

Ilustración 36 Seleccionar todo seleccionado

<input type="checkbox"/>	Continuación	Múltiples respuestas	Inactiva	3	20	2024-01-15 10:01:00	2024-01-23 23:11:00	Ver Borrar Importar
<input type="checkbox"/>	Continuación - web	Múltiples respuestas	Inactiva	2	23	2024-01-17 12:23:00	2024-01-25 10:10:00	Ver Borrar Importar
<input type="checkbox"/>	Síntesis	Ordenar	Inactiva	3	22	2024-01-16 00:30:00	2024-01-24 23:34:00	Ver Borrar Importar
<input type="checkbox"/>	Orden de tokens	Ordenar	Inactiva	5	33	2023-12-21 20:20:00	2023-12-31 23:32:00	Ver Borrar Importar
<input type="checkbox"/>	Práctica AL+TS	Respuesta corta	Inactiva	5	25	2023-10-07 20:58:00	2023-10-21 22:00:00	Ver Borrar Importar
<input type="checkbox"/>	Práctica AST	Respuesta corta	Inactiva	5	25	2023-11-05 13:55:00	2023-11-19 20:02:00	Ver Borrar Importar
<input type="checkbox"/>	Análisis Semántico de expresiones con operandos especiales	Múltiples respuestas	Inactiva	5	70	2023-11-27 22:00:00	2023-12-07 22:00:00	Ver Borrar Importar
<input type="checkbox"/>	Análisis Semántico de expresiones y sentencias	Múltiples respuestas	Inactiva	5	75	2023-12-11 11:00:00	2023-12-21 22:22:00	Ver Borrar Importar
<input type="checkbox"/>	Análisis Semántico de expresiones con conversión de tipos	Múltiples respuestas	Inactiva	5	77	2023-12-30 22:11:00	2024-01-09 22:55:00	Ver Borrar Importar
<input type="checkbox"/>	Examen	Respuesta corta	Inactiva	5	20	2024-01-01 18:00:00	2024-01-11 11:01	Importar actividades seleccionadas Seleccionar todo <input type="checkbox"/>
<input type="checkbox"/>	Práctica Procesadores de Lenguajes	Respuesta corta	Inactiva	3	25	2023-12-23 03:55:00	2024-01-22 22:22	Importar seleccionadas

Ilustración 37 Seleccionar todo deseleccionado

Se puede comprobar con las ilustraciones 35, 36 y 37 que el botón de “Seleccionar todo” funciona correctamente y tanto selecciona como deselecciona todos los *checkboxes*.

4.1.3.5 Importación correcta de actividades seleccionadas

Se seleccionará un grupo de actividades con los *checkboxes* y se comprobará que se han importado todos a la vez y correctamente, comprobándolo también en la página de `manageActivities.php`:

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título: Tipo de Actividad: Estado: Puntos Min: Puntos Max: Día Inicio: Día Fin:

Seleccionar	Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Acciones
<input checked="" type="checkbox"/>	Tokens empresa - 4	Múltiples respuestas	Inactiva	5	44	2023-09-22 15:55:00	2023-10-06 15:00:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input checked="" type="checkbox"/>	Tokens empresa - 5	Múltiples respuestas	Inactiva	5	44	2023-12-17 10:00:00	2023-12-27 10:00:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input checked="" type="checkbox"/>	Tokens empresa - 7	Múltiples respuestas	Inactiva	5	44	2023-11-19 21:00:00	2023-11-29 13:55:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	5	50	2023-10-19 02:00:00	2023-10-29 10:15:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 4	Múltiples respuestas	Inactiva	5	50	2023-12-16 00:33:00	2023-11:15	<input type="button" value="Ver"/> <input type="button" value="Borrar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 5	Múltiples respuestas	Inactiva	5	50	2024-01-04 05:00:00	2024-15:15	<input type="button" value="Ver"/> <input type="button" value="Borrar"/>

Importar actividades seleccionadas
 Seleccionar todo

Ilustración 38 Selección de actividades a importar en checkboxes

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título: Tipo de Actividad: Estado: Puntos Min: Puntos Max: Día Inicio: Día Fin:

Seleccionar	Título	Tipo	Estado	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Acciones
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	5	50	2023-10-19 02:00:00	2023-10-29 10:15:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 4	Múltiples respuestas	Inactiva	5	50	2023-12-16 00:33:00	2023-12-26 11:15:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/> <input type="button" value="Importar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 5	Múltiples respuestas	Inactiva	5	50	2024-01-04 05:00:00	2024-01-10 15:15:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/>
<input type="checkbox"/>	Lenguaje sin declaración - 6	Múltiples respuestas	Inactiva	5	44	2024-01-11 10:00:00	2024-14:00	<input type="button" value="Ver"/> <input type="button" value="Borrar"/>
<input type="checkbox"/>	Diseño de TS	Múltiples	Inactiva	5	70	2024-01-07	2024-	<input type="button" value="Ver"/> <input type="button" value="Borrar"/>

Importar actividades seleccionadas
 Seleccionar todo

Ilustración 39 Estado de la página después de importar

Página de la asignatura	Respuesta única	Inactiva	Sin definir	5 - 25	08-09-2023 22:22	27-09-2023 13:57	Revisar	Editar	Activar	Borrar	Ver
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	Revisar	Editar	Activar	Borrar	Ver
Analizadores	Respuesta corta	Inactiva	Sin definir	5 - 40	12-09-2023 13:55	28-09-2023 09:55	Revisar	Editar	Activar	Borrar	Ver
Token	Respuesta única	Inactiva	Scanneator	10 - 75	14-09-2023 11:45	29-09-2023 23:11	Revisar	Editar	Activar	Borrar	Ver
Termina la frase	Respuesta corta	Inactiva	Scanneator	5 - 50	14-09-2023 13:13	30-09-2023 23:32	Revisar	Editar	Activar	Borrar	Ver
Función del Analizador Léxico	Respuesta múltiple	Inactiva	Scanneator	10 - 75	16-09-2023 21:00	01-10-2023 13:13	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 1	Respuesta única	Inactiva	Scanneator	5 - 44	19-09-2023 11:11	03-10-2023 23:34	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 2	Respuesta única	Inactiva	Scanneator	5 - 44	20-09-2023 13:00	04-10-2023 22:00	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 3	Respuesta única	Inactiva	Scanneator	5 - 44	21-09-2023 00:00	05-10-2023 22:01	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 4	Respuesta única	Inactiva	Scanneator	5 - 44	22-09-2023 15:55	06-10-2023 15:00	Revisar	Editar	Activar	Borrar	Ver
Tokens del programa	Respuesta múltiple	Inactiva	Scanneator	5 - 44	22-09-2023 19:00	07-10-2023 11:22	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 6	Respuesta única	Inactiva	Scanneator	5 - 44	09-10-2023 15:00	19-10-2023 10:00	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 7	Respuesta única	Inactiva	Scanneator	5 - 44	19-11-2023 21:00	29-11-2023 13:55	Revisar	Editar	Activar	Borrar	Ver
Tokens varios - 2	Respuesta múltiple	Inactiva	Scanneator	10 - 75	09-12-2023 12:05	19-12-2023 22:00	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 8	Respuesta única	Inactiva	Scanneator	5 - 44	17-12-2023 10:00	27-12-2023 10:00	Revisar	Editar	Activar	Borrar	Ver
Tokens varios	Respuesta múltiple	Inactiva	Scanneator	10 - 70	19-12-2023 23:05	29-12-2023 05:05	Revisar	Editar	Activar	Borrar	Ver
Acciones semánticas del Léxico	Respuesta múltiple	Inactiva	Scanneator	5 - 60	28-12-2023 16:00	08-01-2024 08:00	Revisar	Editar	Activar	Borrar	Ver
Comida	Encuesta larga	Activa	Sin definir	0 - 66	09-12-2024 00:00	09-12-2025 00:00	Revisar	Editar	Desactivar	Auto	Ver

Cambiar la categoría de todas las actividades:

Ilustración 40 Comprobación en `manageActivities.php`

Como se ha podido comprobar en las ilustraciones 38, 39 y 40, la importación se realiza de manera correcta y sin ningún problema.

4.2 Nueva información dentro de la página de importar actividades y manejar actividades

4.2.1 Diseño

En este apartado, hay dos vistas que analizar: la vista de importar y manejar actividades. La arquitectura de la página de importar actividades ya ha sido mencionada en el problema anterior, con lo que ahora se analizará la arquitectura de la página de manejar actividades.

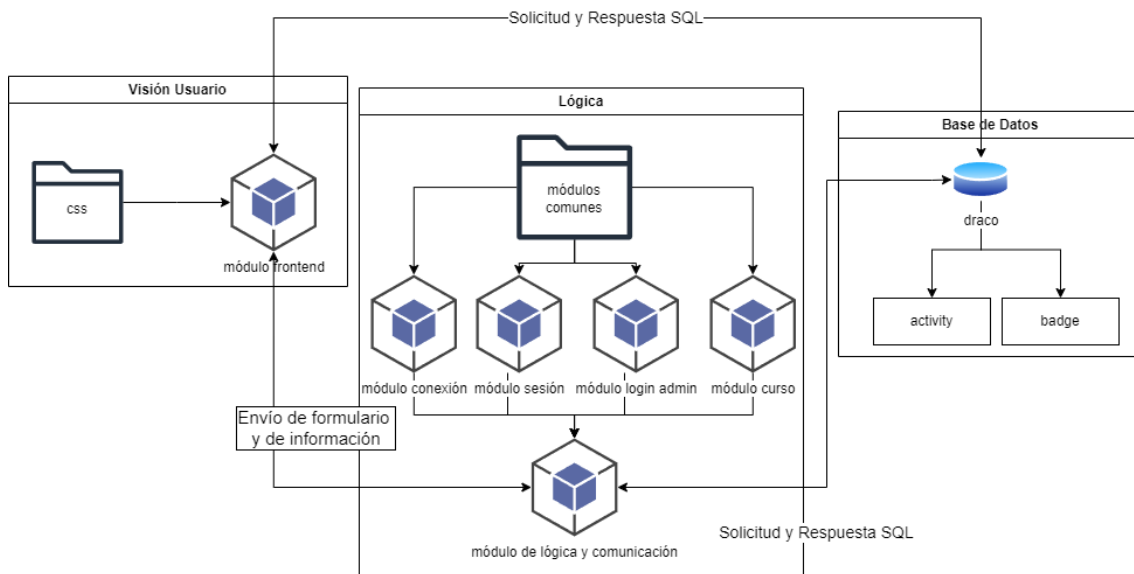


Ilustración 41 Arquitectura de vista de manejar lenguajes

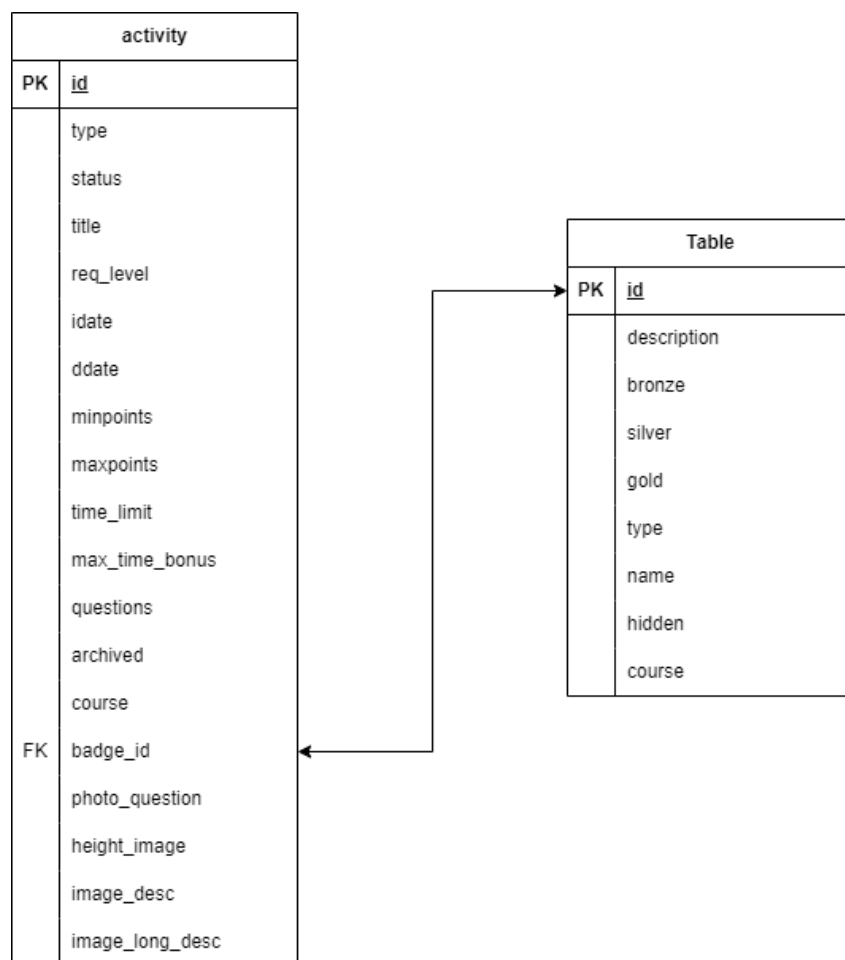


Ilustración 42 Relación de las tablas mencionadas en la ilustración 39

Se puede observar en la ilustración 41 que tiene una arquitectura muy similar a la arquitectura de la vista de importar actividades.

Un matiz importante es que, aunque los módulos en las imágenes tienen el mismo nombre, ya que para sus respectivas arquitecturas tienen funciones casi idénticas, no serán los mismos módulos. Los módulos de *frontend* y de lógica y comunicación no son los mismos.

Además, se ve que el módulo de *frontend* tiene contacto directo con la base de datos, aunque seguirá recibiendo algunos datos de la base de datos mediante el módulo de lógica y comunicación.

Finalmente, se observa que la comunicación con la base de datos ahora incluye otra tabla, la tabla *badge*. Esta tabla es clave para sacar el nombre correspondiente a una medalla, ya que en la tabla *activities* solo se le asigna a cada actividad el id de la tabla correspondiente, como se puede observar en la ilustración 42.

La arquitectura inicial es correcta, así que se observará cómo funciona la lógica de las páginas principales relevante para solucionar este problema, para determinar qué cambios se tendrán que hacer. Se observará específicamente cómo se muestra la información, cómo se obtiene la información a mostrar y el filtrado.

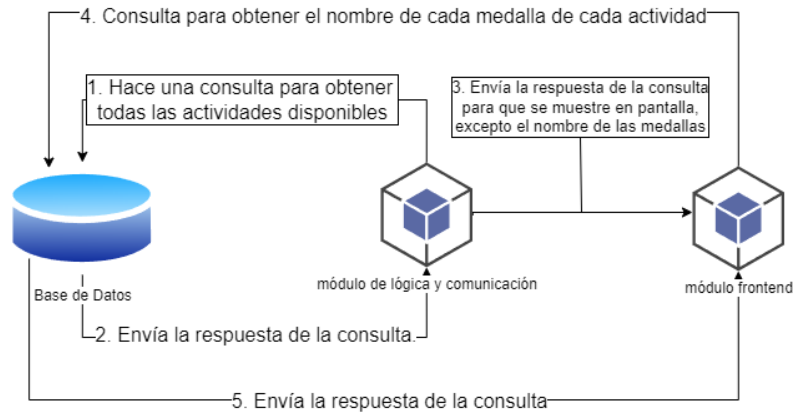


Ilustración 43 Explicación de cómo se obtiene la información al cargar la página de manejar actividades

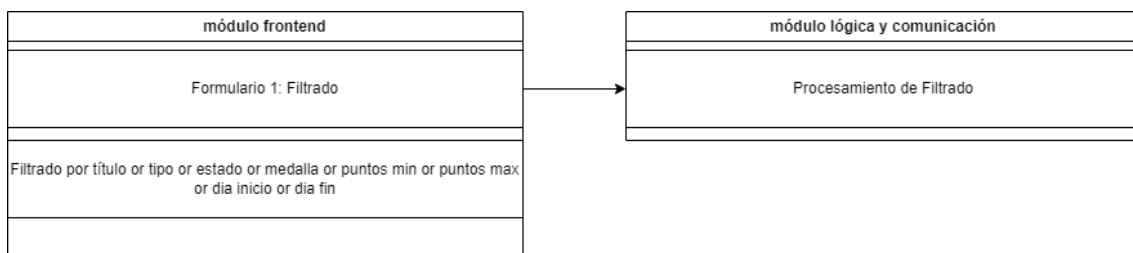


Ilustración 44 lógica de filtrado

La ilustración 43 explica cómo se carga la página para que muestre toda la información que piden el módulo de lógica y comunicación y el módulo de *frontend*. En los casos del filtrado, es muy parecido, excepto que habría un paso anterior. Este paso es el envío de un formulario con los datos específicos del filtrado, enviados por el módulo de *frontend* al módulo de lógica y comunicación. Esto se muestra en la ilustración 44. Esta ilustración también muestra los atributos que se pueden filtrar, es decir, todos los atributos que se muestran en pantalla. Como se pudo ver en la ilustración 42, la tabla *activities* contiene un valor de nivel requerido, con lo que lo único necesario sería modificar la lógica del procesamiento del filtrado, además del formulario del módulo *frontend*, además de cómo se muestra la información en pantalla.

Este mismo diseño se aplicaría también en el módulo *frontend* y el módulo de lógica y comunicación de la página de importaciones, ya que son estructuralmente iguales.

De cara a implementar la muestra y filtrado de medallas en los archivos relacionados a importar actividades, habría que hacer un ligero cambio en la arquitectura y lógica. Esto es debido a que, como se ha aprendido, se necesita consultar una tabla a distinta a la tabla *activity* que, como se vio en la arquitectura de la página de importaciones, actualmente es la única tabla que se consulta.

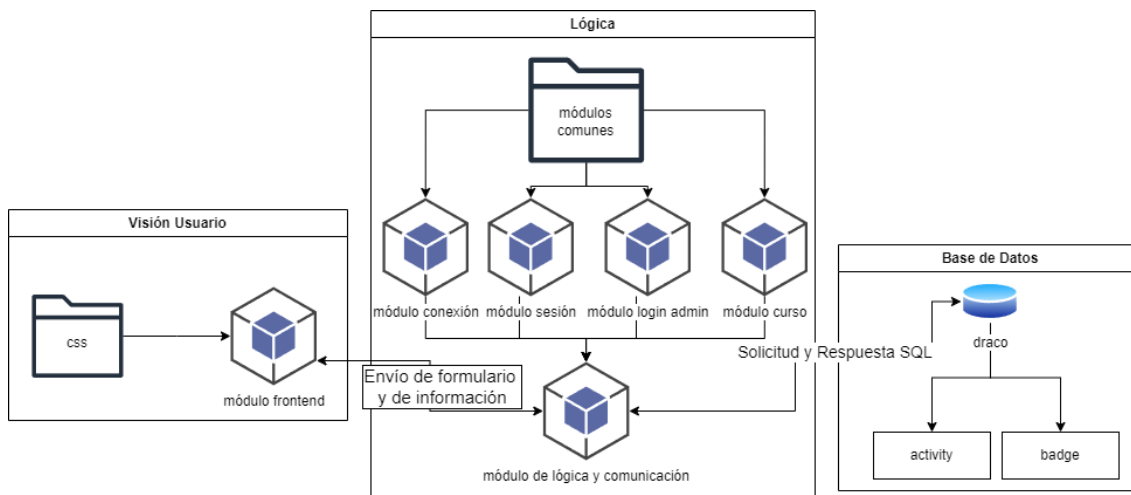


Ilustración 45 Nueva arquitectura de la página de importaciones

Este cambio mostrado en la ilustración 44 implica que han de hacerse modificaciones en las consultas de carga de página, que ya se mencionó es un proceso idéntico funcionalmente a lo mostrado en la ilustración 43, además de las modificaciones que se harán en el filtrado (modificaciones parecidas a las hechas para mostrar el nivel) y modificaciones para mostrar en pantalla la nueva información.

4.2.2 Implementación

Como se mencionó en el apartado de problemática, se puede observar que hay una discrepancia entre la información de las actividades mostrada en las páginas de importar y manejar actividades.

Ya se ha hablado en detalle en el anterior punto sobre cómo funciona la página de importActivity.php a la hora de sus funciones, pero no se ha especificado como obtiene la información a mostrar. Se hablará de ello para importActivity.php y manageActivities.php, que son las páginas más relevantes para la página de manejar actividades, comenzando con el anterior:

```
<div class="td"><?php echo $activity["title"]; ?></div>
<div class="td"><?php echo $activity_type_display; ?></div>
<div class="td"><?php echo $activity_status_display; ?></div>
<div class="td"><?php echo $activity["minpoints"]; ?></div>
<div class="td"><?php echo $activity["maxpoints"]; ?></div>
<div class="td"><?php echo $activity["idate"]; ?></div>
<div class="td"><?php echo $activity["ddate"]; ?></div>
```

Este código muestra cómo sale en pantalla actualmente la información de las actividades en importActivity.php. Se saca de una variable llamada \$activity para la mayoría de los casos, teniendo el tipo y el estatus. Ahora se explicará donde se determinan el tipo y el estatus. En estos dos casos, la obtención original de la información también se obtiene de la variable activity, pero en el código hay presente un segmento que se encarga de “traducir” el valor en ambos casos:

```
<?php
    // Conversión de tipos
    switch ($activity["type"]) {
        case "Radio_Check_Type":
            $activity_type_display = "Múltiples respuestas";
            break;
```

```

    case "Short_Type":
        $activity_type_display = "Respuesta corta";
        break;
    case "Sequential_Type":
        $activity_type_display = "Secuencial";
        break;
    case "Long_Type":
        $activity_type_display = "Respuesta larga";
        break;
    case "Review":
        $activity_type_display = "Revisión";
        break;
    case "Order_Type":
        $activity_type_display = "Ordenar";
        break;
    case "Pair_Type":
        $activity_type_display = "Emparejar";
        break;
    case "Faulty":
        $activity_type_display = "Defectuoso";
        break;
    case "Coop":
        $activity_type_display = "Colaborativa";
        break;
    default:
        $activity_type_display = "Desconocido";
}

```

```

// Conversión de estados
switch ($activity["status"]) {
    case "Active":
        $activity_status_display = "Activa";
        break;
    case "Inactive":
        $activity_status_display = "Inactiva";
        break;
    case "Incomplete":
        $activity_status_display = "Incompleta";
        break;
    case "Editing":
        $activity_status_display = "Editando";
        break;
    case "Auto":
        $activity_status_display = "Auto";
        break;
    default:
        $activity_status_display = "Desconocido";
}

```

?>

Este código muestra cómo, dependiendo del valor que haya en dentro de la variable *activity*, se asocia un valor distinto a las variables personales de estado y tipo. Ahora se resolverá la duda de dónde se determina el valor de la variable *activity*.

Se puede ver que esta variable se define en `importActivity.php`, al comienzo del bucle que itera por cada actividad para extraer la información. La línea donde se hace es:

```
<?php foreach ($available_act as $index => $activity): ?>
```

Aquí se ve que la información se saca de otra variable llamada `available_act`, que sale de `importActivityAction.php`. Esta variable se define de la siguiente manera:

```
$sql_query = "SELECT * FROM Activity WHERE `archived` = 1 AND `course` = '". $course.'"
ORDER BY idate ASC";
$query_result = mysqli_query($db_link, $sql_query);
$available_act = my_fetch_all($query_result, MYSQLI_ASSOC);
```

La variable `course` determina si la actividad es de Procesadores de Lenguajes (PDL) o de Traductores de Lenguajes (TDL), que son ambas asignaturas del departamento, pero en este trabajo se asumirá siempre que se está en PDL. A parte de eso, la consulta SQL recoge todas las actividades que tienen un `archived = 1`, que se recuerda significa que esta archivado y listo para importar. Después, los ordena de forma ascendente.

Ahora se observará cómo obtiene la información que quiere `manageActivities.php`. Observando, se ve una lista de actividades importadas en `manageActivitiesAction.php`:

```
$sql_query = "SELECT * FROM Activity WHERE `archived` = 0 AND `course` = '". $course.'"
ORDER BY idate ASC";
$query_result = mysqli_query($db_link, $sql_query);
$available_act = my_fetch_all($query_result, MYSQLI_ASSOC);
```

Aquí se puede ver que su funcionamiento es idéntico a `importActivity.php`, excepto que busca las filas donde `archived` tiene un valor 0. Después hace un bucle que itera sobre cada fila, cambiando el nombre de la variable a `activity`, exactamente igual que `importActivity.php`:

```
<?php foreach($available_act as $activity):?>
```

La principal diferencia es que, en vez de obtener los datos que quiere directamente desde la variable `activity`, la usa para hacer una consulta SQL, por ejemplo:

```
// Display the badge name
if($activity["badge_id"]!="") {
    $db_link = connect_db();
    $sql_query = "SELECT * FROM `badge` WHERE `id` = " . $activity["badge_id"];
    $query_result = mysqli_query($db_link, $sql_query);
    $activity_badge = mysqli_fetch_assoc($query_result)["name"];
    disconnect_db($db_link);
} else
    $activity_badge = "<em>Sin definir</em>";
```

Ahora que se tiene este contexto, hay que determinar cuáles son los cambios que se quieren hacer en ambos códigos. Es deseable que en `importactivity.php` se muestre la medalla de la actividad que hasta ahora no se ve. Además, se quiere que en `importActivity.php` y en `manageActivities.php` se muestre el nivel de la actividad.

Se comienza haciendo visibles las medallas en `importActivity.php`. Para que las medallas sean visibles, se tienen que obtener de la variable `activity`, para lo que es necesario, tal y como se mencionó en el diseño, la información de la tabla `badge`. Para ello, se crea la siguiente nueva consulta:

```
$sql_query = "SELECT
Activity.*,
Badge.id AS badge_id,
```

```

Badge.name AS badge_name
FROM Activity
LEFT JOIN Badge ON Activity.badge_id = Badge.id
WHERE Activity.archived = 1 AND Activity.course = '$course';

```

Esta consulta permite mantener el funcionamiento correcto anterior de la página, y permite ahora añadir el nombre de la medalla bajo el nombre de badge_name.

Ahora se añadirá el nivel. Es la consulta original, la información del nivel ya está en la tabla a la que se hace la consulta, con lo que solo hay que extraer la información de la misma manera que se han extraído el resto de columnas en la base de datos. Si se observa en la lista dada anteriormente de las columnas de la tabla *activity*, se puede ver una columna llamada req_level, que tiene la información necesitada.

De cara a que se muestren de manera correcta, sería tan sencillo como añadir la línea que corresponde en la lista de columnas, que sería tal que:

```

<div class="thead">
  <div class="tr">
    <div class="td">Seleccionar</div>
    <div class="td">Título</div>
    <div class="td">Tipo</div>
    <div class="td">Estado</div>
    <div class="td">Medalla</div>
    <div class="td">Puntos Mínimos</div>
    <div class="td">Puntos Máximos</div>
    <div class="td">Fecha Inicio</div>
    <div class="td">Fecha Fin</div>
    <div class="td">Nivel</div>
    <div class="td">Acciones</div>
  </div>
</div>

```

Esto va seguido de mostrar la medalla y el nivel:

```

<div class="td"><?php echo $activity["title"]; ?></div>
<div class="td"><?php echo $activity_type_display; ?></div>
<div class="td"><?php echo $activity_status_display; ?></div>
<div class="td"><?php echo $activity["badge_name"] ?? "Sin medalla"; ?></div>
<div class="td"><?php echo $activity["minpoints"]; ?></div>
<div class="td"><?php echo $activity["maxpoints"]; ?></div>
<div class="td"><?php echo $activity["idate"]; ?></div>
<div class="td"><?php echo $activity["ddate"]; ?></div>
<div class="td"><?php echo $activity["req_level"]; ?></div>

```

Se observa también, que se ha puesto una condición para el caso de que sea null, es decir, no haya una medalla asociada a la actividad, en cuyo caso se mostrará “Sin medalla”.

A continuación, se añaden ambas variables dentro del filtrado. Para ello, primero hay que añadirlos a las etiquetas de los filtros:

```

<th><label for="title">Título</label></th>
<th><label for="activity_type">Tipo de Actividad</label></th>
<th><label for="status">Estado</label></th>
<th><label for="badge">Medalla</label></th>
<th><label for="minpoints">Puntos Min</label></th>
<th><label for="maxpoints">Puntos Max</label></th>
<th><label for="idate">Día Inicio</label></th>
<th><label for="ddate">Día Fin</label></th>

```

```
<th><label for="level">Nivel</label></th>
```

Esto va seguido de cómo se va a filtrar cada uno de los casos. En el caso de la medalla, se debe dar a elegir solo medallas existentes y, específicamente, medallas que puedan estar asociadas a una actividad. Por suerte, esta lista ya está disponible en el código de `manageActivities.php`, resultando en lo siguiente:

```
<td><select name="badge" id="badge" style="width:11em;">
  <option value="blank">Seleccione</option>
  <?php
    foreach ($medallas_array as $medalla){
      echo"<option
        value='". $medalla[6]. "'>". $medalla[6]. "</option>";
    }
  ?>
  <option value="none">Ninguna</option>
</select></td>
```

Este código itera sobre una variable llamada `medallas_array`. Esta variable se crea en `importActivityAction.php`. Se basa en `manageActivities.php`, donde se hace de la misma manera para el filtrado por medallas. La creación de la variable se hace con una consulta:

```
$sql_query = "SELECT * FROM `Badge` WHERE `id` > 16 AND (`course` IS NULL OR `course` = '". $course. "') ORDER BY `id` ASC";
$result = mysqli_query($db_link, $sql_query);
$medallas_array = mysqli_fetch_all($result);
```

Esto lista todas las medallas necesarias solo para las actividades, debido a que se les dio `id` mayor que 16.

En el caso del nivel, solo se hace una entrada para un número, tal que:

```
<td><input type="number" name="level" id="level" min="0"></td>
```

Como bien se ha mencionado antes, el filtrado se hace esencialmente modificando la consulta SQL que muestra el listado inicial, haciendo la consulta más restrictiva acorde a los filtros solicitados. Para ello se muestran los filtros y sus consultas, comenzando con la medalla:

```
if (isset($_POST["badge"]) && $_POST["badge"] != "blank") {
  if ($_POST["badge"] != "none") {
    $badge_query = "SELECT id FROM `Badge` WHERE `name` = '" .
      mysqli_real_escape_string($db_link, $_POST["badge"]) . "'";
    $badge_result = mysqli_query($db_link, $badge_query);
    if ($badge_row = mysqli_fetch_assoc($badge_result)) {
      $badge_id = intval($badge_row["id"]);
      $query .= " AND Activity.badge_id = $badge_id";
    } else {
      // Si no se encuentra la medalla, forzar cero resultados
      $query .= " AND 0 = 1";
    }
    mysqli_free_result($badge_result);
  } else {
    $query .= " AND Activity.badge_id IS NULL";
  }
}
```

Este fragmento de código comprueba si se ha enviado un valor válido en el campo `badge` de un formulario. Si el valor no es vacío, el código realiza una consulta para filtrar actividades en función de una medalla asociada (`badge_id`).

Primero, si el valor de *badge* es distinto de “Sin medalla”, se construye una consulta para buscar el id correspondiente al nombre de la medalla en la tabla *badge*, utilizando `mysqli_real_escape_string` para evitar inyecciones SQL. Si se encuentra una coincidencia, se añade una condición a la consulta principal para filtrar las actividades que tengan asociado el `badge_id` encontrado. Si no se encuentra la medalla, se fuerza la consulta a devolver cero resultados añadiendo la condición `AND 0 = 1`.

Si el valor de *badge* es *none*, se añade una condición a la consulta principal para filtrar únicamente las actividades sin ninguna medalla asociada (`badge_id IS NULL`). Finalmente, se libera la memoria asociada al resultado de la consulta mediante `mysqli_free_result` para optimizar el uso de recursos.

En el caso del nivel, como se ha podido ir observando, es más sencillo, tal que:

```
if (isset($_POST["level"]) && $_POST["level"] != "") {
    $query .= " AND req_level = " . intval($_POST["level"]);
}
}
```

Este código verifica si el campo *level* ha sido enviado a través de un formulario y no está vacío. Si la condición se cumple, se añade una cláusula a la consulta principal para filtrar los resultados según el nivel especificado. El valor del nivel enviado en `$_POST["level"]` se convierte a un entero utilizando `intval` para garantizar que el dato sea seguro y válido antes de incluirlo en la consulta SQL, reduciendo así el riesgo de inyecciones SQL y errores de tipo.

Ahora se prosigue a meter el nivel en `manageActivities.php`. Primero se mete en la lista de columnas a mostrar:

```
<div class="tr">
    <div class="td">Título</div>
    <div class="td">Tipo</div>
    <div class="td">Estado</div>
    <div class="td">Medalla</div>
    <div class="td">Puntuación</div>
    <div class="td">Fecha Inicio</div>
    <div class="td">Fecha Fin</div>
    <div class="td">Nivel</div>
</div>
```

Esto va seguido de mostrar el dato de cada actividad en pantalla:

```
<div class="td"><?php echo $activity["title"]?></div>
<div class="td"><?php echo $activity_type_display?></div>
<div class="td"><?php echo $activity_status_display?></div>
<div class="td"><?php echo $activity_badge?></div>
<div class="td"><?php echo $activity["minpoints"]." -
".$activity["maxpoints"]?></div>
<div class="td"><?php echo date_format(date_create($activity["idate"]), 'd-m-Y
H:i')?></div>
<div class="td"><?php echo date_format(date_create($activity["ddate"]), 'd-m-Y
H:i')?></div>
<div class="td"><?php echo $activity["req_level"]?></div>
```

También se trabaja en el filtrado:

```
<th><label for="level">Nivel<button name="order" value="levelAZ"></button><button
type="submit" name="order" value="levelZA"></button></label></th>
```

A simple vista se puede ver que esto no es un simple título para el filtrado. Esto es el caso para todos los filtros de esta página. Este fragmento de código HTML genera un encabezado de tabla (<th>) que incluye una etiqueta (<label>) asociada al campo *level* junto con dos botones para ordenar datos. Los botones permiten ordenar los niveles en orden ascendente (*levelAZ*) o descendente (*levelZA*).

Cada botón tiene un atributo *name* configurado como "*order*" y un valor (*value*) que representa el criterio de ordenación. Además, los botones incluyen imágenes (flechas hacia arriba y hacia abajo) para indicar visualmente la dirección del orden. Estas imágenes se presentan con un estilo ajustado a un tamaño pequeño (1em por 1em) para mantener una apariencia compacta y coherente dentro de la tabla.

También se añade la entrada del valor para el filtro:

```
<td><input type="number" name="level" id="level" min="0" style="width:8.5em;"></td>
```

Y finalmente se modifica el filtrado en el *action* tal que:

```
// filter by level
if(isset($_POST["level"])){
    $level = $_POST["level"];
    if($level != "" && is_numeric($level)){
        $query .= " AND req_level = " . intval($level);
    }
}
```

Una vez implementado todo el código mencionado, se obtiene el siguiente resultado:

The screenshot shows a web interface for managing activities. At the top, there are navigation buttons: 'Añadir nueva actividad', 'Copiar actividades', and 'Importar actividades'. Below this is a filter section with the text 'Puede filtrar actividades que cumplan ciertas características'. The filter includes dropdown menus for 'Titulo', 'Tipo de Actividad', 'Estado', 'Medalla', and 'Nivel', along with input fields for 'Puntos Min' and 'Puntos Max', and date pickers for 'Dia Inicio' and 'Dia Fin'. A 'Filtrar' button is on the right. Below the filter is a section titled 'Seleccione la actividad a gestionar:' followed by a table of activities.

Título	Tipo	Estado	Medalla	Puntuación	Fecha Inicio	Fecha Fin	Nivel
Segundo examen test	Respuesta múltiple	Auto	Sin definir	3 - 22	07-11-2021 23:59	14-11-2021 23:57	1
Examen parcial	Respuesta corta	Activa	Sin definir	3 - 20	16-11-2021 17:17	21-11-2021 22:22	0
Encuesta - 500 puntos	Respuesta corta	Activa	Sin definir	1 - 8	06-01-2022 12:34	27-01-2022 12:34	0
Música en Clase práctica TDS	Respuesta única	Activa	Sin definir	3 - 22	24-11-2022 17:00	30-11-2022 21:22	4
Fiesta	Respuesta única	Activa	Sin definir	3 - 11	26-01-2023 17:59	27-01-2023 17:00	1
Bienvenida	Respuesta única	Activa	Sin definir	0 - 50	06-09-2023 08:58	26-09-2023 23:32	0
Mascarilla	Respuesta única	Activa	Sin definir	2 - 11	07-09-2023 18:00	26-09-2023 01:23	0
Página de la asignatura	Respuesta única	Inactiva	Sin definir	5 - 25	08-09-2023 22:22	27-09-2023 13:57	0
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	1
Analizadores	Respuesta corta	Inactiva	Sin definir	5 - 40	12-09-2023 13:55	28-09-2023 09:55	0
Token	Respuesta única	Inactiva	Scanneator	10 - 75	14-09-2023 11:45	29-09-2023 23:11	0
Termina la frase	Respuesta corta	Inactiva	Scanneator	5 - 50	14-09-2023 13:13	30-09-2023 23:32	0

Ilustración 46 ManageActivities.php con las correcciones e implementaciones

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Medalla	Puntos Min	Puntos Max	Día Inicio	Día Fin	Nivel		
<input type="text"/>	Selecione	Selecione	Selecione	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	<input type="text"/>	Filtrar	
Seleccionar	Título	Tipo	Estado	Medalla	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Nivel	Acciones
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	Scanneator	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	0	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	1	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-19 02:00:00	2023-10-29 10:15:00	2	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 4	Múltiples respuestas	Inactiva	Tableator	5	50	2023-12-16 00:33:00	2023-12-26 11:15:00	3	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 5	Múltiples respuestas	Inactiva	Tableator	5	50	2024-01-04 05:00:00	2024-01-10 15:15:00	4	Ver, Borrar
<input type="checkbox"/>	Lenguaje sin declaración - 6	Múltiples respuestas	Inactiva	Tableator	5	44	2024-01-11 10:00:00	2024-01-14 14:00:00		Ver, Borrar, Importar
<input type="checkbox"/>	Diseño de TS	Múltiples	Inactiva	Tableator	5	70	2024-01-07	2024-01-07		Ver, Borrar, Importar

Importar actividades seleccionadas
 Seleccionar todo
 Importar seleccionadas

Ilustración 47 ImportActivity.php con las correcciones e implementaciones

Como se puede ver en las ilustraciones 46 y 47, todas nuestras implementaciones, al menos visualmente, han sido exitosas. Queda hacer pruebas con los filtros para ver si funcionan de manera deseable.

4.2.3 Pruebas

4.2.3.1 Probar filtrado de medallas y nivel en ImportActivity.php

Para esta prueba, se realizan distintos filtrados. Primero se hará un filtrado solo por medalla, después solo por nivel, y finalmente se hará una combinación de ambos filtrados, observando que omita las actividades que se quieren que omita.

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Medalla	Puntos Min	Puntos Max	Día Inicio	Día Fin	Nivel		
<input type="text"/>	Selecione	Selecione	Scanneator	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	<input type="text"/>	Filtrar	
Seleccionar	Título	Tipo	Estado	Medalla	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Nivel	Acciones
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	Scanneator	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	0	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	1	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-19 02:00:00	2023-10-29 10:15:00	2	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 4	Múltiples respuestas	Inactiva	Tableator	5	50	2023-12-16 00:33:00	2023-12-26 11:15:00	3	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 5	Múltiples respuestas	Inactiva	Tableator	5	50	2024-01-04 05:00:00	2024-01-10 15:15:00	4	Ver, Borrar
<input type="checkbox"/>	Lenguaje sin declaración - 6	Múltiples respuestas	Inactiva	Tableator	5	44	2024-01-11 10:00:00	2024-01-14 14:00:00		Ver, Borrar, Importar
<input type="checkbox"/>	Diseño de TS	Múltiples	Inactiva	Tableator	5	70	2024-01-07	2024-01-07		Ver, Borrar, Importar

Importar actividades seleccionadas
 Seleccionar todo
 Importar seleccionadas

Ilustración 48 Filtrado solo de Medalla. En este caso para la medalla "Scanneator"

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Medalla	Puntos Min	Puntos Max	Día Inicio	Día Fin	Nivel		
<input type="text"/>	Seleccione	Seleccione	Seleccione	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	<input type="text"/>	Filtrar	
Seleccionar	Título	Tipo	Estado	Medalla	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Nivel	Acciones
<input type="checkbox"/>	Gramática léxica	Múltiples respuestas	Inactiva	Scanneator	3	33	2023-09-13 19:29:00	2023-09-29 10:10:00	0	Ver, Borrar, Importar
<input type="checkbox"/>	Gramática del Léxico	Múltiples respuestas	Inactiva	Scanneator	3	33	2023-09-15 14:10:00	2023-09-30 13:10:00	1	Ver, Borrar, Importar
<input type="checkbox"/>	G para id	Múltiples respuestas	Inactiva	Scanneator	3	44	2023-09-15 18:18:00	2023-10-01 11:11:00	0	Ver, Borrar, Importar
<input type="checkbox"/>	Autómata del Léxico	Múltiples respuestas	Inactiva	Scanneator	3	33	2023-09-17 11:11:00	2023-10-02 02:34:00	1	Ver, Borrar, Importar
<input type="checkbox"/>	AFD	Múltiples respuestas	Inactiva	Scanneator	3	44	2023-09-18 18:09:00	2023-10-02 12:22:00	0	Ver, Borrar
<input type="checkbox"/>	Gramática	Múltiples respuestas	Inactiva	Scanneator	5	75	2023-09-18 19:00:00	2023-10-0 10:00:00		Importar actividades seleccionadas <input type="checkbox"/> Seleccionar todo <input type="button" value="Importar seleccionadas"/>
<input type="checkbox"/>	Transiciones	Múltiples	Inactiva	Scanneator	3	33	2023-09-19	2023-10-0		

Ilustración 49 Resultado del filtrado en la ilustración 48

El filtrado por medalla, tal y como se puede ver en las ilustraciones 48 y 49 es correcto.

Como se puede observar en las ilustraciones 50 y 51, el filtrado por nivel se realiza correctamente.

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Medalla	Puntos Min	Puntos Max	Día Inicio	Día Fin	Nivel		
<input type="text"/>	Seleccione	Seleccione	Seleccione	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	6	Filtrar	
Seleccionar	Título	Tipo	Estado	Medalla	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Nivel	Acciones
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	Scanneator	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	0	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	1	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-19 02:00:00	2023-10-29 10:15:00	2	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 4	Múltiples respuestas	Inactiva	Tableator	5	50	2023-12-16 00:33:00	2023-12-26 11:15:00	3	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 5	Múltiples respuestas	Inactiva	Tableator	5	50	2024-01-04 05:00:00	2024-01-10 15:15:00	4	Ver, Borrar
<input type="checkbox"/>	Lenguaje sin declaración - 6	Múltiples respuestas	Inactiva	Tableator	5	44	2024-01-11 10:00:00	2024-0 14:00:00		Importar actividades seleccionadas <input type="checkbox"/> Seleccionar todo <input type="button" value="Importar seleccionadas"/>
<input type="checkbox"/>	Diseño de TS	Múltiples	Inactiva	Tableator	5	70	2024-01-07	2024-0		

Ilustración 50 Filtrado por nivel

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Medalla	Puntos Min	Puntos Max	Día Inicio	Día Fin	Nivel		
<input type="text"/>	Seleccione ▾	Seleccione ▾	Seleccione ▾	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	<input type="text"/>	Filtrar	
Seleccionar	Título	Tipo	Estado	Medalla	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Nivel	Acciones
<input type="checkbox"/>	First - 6	Múltiples respuestas	Inactiva	Maestro parse	5	66	2023-11-20 13:23:00	2023-11-30 20:20:00	6	Ver, Borrar, Importar
<input type="checkbox"/>	Follow Lenguaje Programación	Múltiples respuestas	Inactiva	Maestro parse	4	44	2023-11-23 19:00:00	2023-12-03 12:34:00	6	Ver, Borrar, Importar
<input type="checkbox"/>	Fiesta	Múltiples respuestas	Inactiva	Sin medalla	5	11	2023-12-08 00:00:00	2023-12-08 23:58:00	6	Ver, Borrar, Importar
<input type="checkbox"/>	Cultura	Múltiples respuestas	Inactiva	Sin medalla	3	22	2023-12-26 15:55:00	2024-01-02 21:00:00	6	Ver, Borrar, Importar
<input type="checkbox"/>	Continuación - web	Múltiples respuestas	Inactiva	Sin medalla	2	23	2024-01-17 12:23:00	2024-01-25 10:10:00	6	Ver, Borrar

Volver

Importar actividades seleccionadas

Seleccionar todo

Importar seleccionadas

Ilustración 51 Resultados del filtrado de la ilustración 50

Como se puede observar en las ilustraciones 52 y 53, el filtrado en importActivity.php es correcto.

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título	Tipo de Actividad	Estado	Medalla	Puntos Min	Puntos Max	Día Inicio	Día Fin	Nivel		
<input type="text"/>	Seleccione ▾	Seleccione ▾	Ninguna ▾	<input type="text"/>	<input type="text"/>	2024-12-28	2025-01-04	6	Filtrar	
Seleccionar	Título	Tipo	Estado	Medalla	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Nivel	Acciones
<input type="checkbox"/>	Diseño de tokens	Múltiples respuestas	Inactiva	Scanneator	10	50	2023-09-29 17:00:00	2023-10-12 17:00:00	0	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 2	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-14 19:55:00	2023-10-24 05:05:00	1	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 3	Múltiples respuestas	Inactiva	Tableator	5	50	2023-10-19 02:00:00	2023-10-29 10:15:00	2	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 4	Múltiples respuestas	Inactiva	Tableator	5	50	2023-12-16 00:33:00	2023-12-26 11:15:00	3	Ver, Borrar, Importar
<input type="checkbox"/>	Lenguaje sin declaración - 5	Múltiples respuestas	Inactiva	Tableator	5	50	2024-01-04 05:00:00	2024-01-10 15:15:00	4	Ver, Borrar
<input type="checkbox"/>	Lenguaje sin declaración - 6	Múltiples respuestas	Inactiva	Tableator	5	44	2024-01-11 10:00:00	2024-01-14 14:00:00		Ver, Borrar, Importar
<input type="checkbox"/>	Diseño de TS	Múltiples	Inactiva	Tableator	5	70	2024-01-07	2024-01-07		Ver, Borrar, Importar

Importar actividades seleccionadas

Seleccionar todo

Importar seleccionadas

Ilustración 52 Filtración por medalla y por nivel

GESTIÓN DE ACTIVIDADES

Puede filtrar actividades que cumplan ciertas características

Título: Tipo de Actividad: Estado: Medalla: Puntos Min: Puntos Max: Día Inicio: Día Fin: Nivel:

Seleccionar	Título	Tipo	Estado	Medalla	Puntos Mínimos	Puntos Máximos	Fecha Inicio	Fecha Fin	Nivel	Acciones
<input type="checkbox"/>	Fiesta	Múltiples respuestas	Inactiva	Sin medalla	5	11	2023-12-08 00:00:00	2023-12-08 23:58:00	6	Ver, Borrar, Importar
<input type="checkbox"/>	Cultura	Múltiples respuestas	Inactiva	Sin medalla	3	22	2023-12-26 15:55:00	2024-01-02 21:00:00	6	Ver, Borrar, Importar
<input type="checkbox"/>	Continuación - web	Múltiples respuestas	Inactiva	Sin medalla	2	23	2024-01-17 12:23:00	2024-01-25 10:10:00	6	Ver, Borrar, Importar

Seleccionar todo

Ilustración 53 Resultado de la filtración de la ilustración 52

4.2.3.2 Prueba de filtro de nivel en manageActivities.php

Se probará si funcionan bien las flechas de nivel, además del filtro en manageActivities.

Añadir nueva actividad

Puede filtrar actividades que cumplan ciertas características

Título: Tipo de Actividad: Estado: Medalla: Puntos Min: Puntos Max: Día Inicio: Día Fin: Nivel:

Seleccione la actividad a gestionar:

Título	Tipo	Estado	Medalla	Puntuación	Fecha Inicio	Fecha Fin	Nivel	Acciones
Bienvenida	Respuesta única	Activa	Sin definir	0 - 50	06-09-2023 08:58	26-09-2023 23:32	0	Revisar, Editar, Desactivar, Auto, Ver
Función del Analizador Léxico	Respuesta múltiple	Inactiva	Scanneator	10 - 75	16-09-2023 21:00	01-10-2023 13:13	0	Revisar, Editar, Activar, Borrar, Ver
Token	Respuesta única	Inactiva	Scanneator	10 - 75	14-09-2023 11:45	29-09-2023 23:11	0	Revisar, Editar, Activar, Borrar, Ver
Analizadores	Respuesta corta	Inactiva	Sin definir	5 - 40	12-09-2023 13:55	28-09-2023 09:55	0	Revisar, Editar, Activar, Borrar, Ver
Página de la asignatura	Respuesta única	Inactiva	Sin definir	5 - 25	08-09-2023 22:22	27-09-2023 13:57	0	Revisar, Editar, Activar, Borrar, Ver
Termina la frase	Respuesta corta	Inactiva	Scanneator	5 - 50	14-09-2023 13:13	30-09-2023 23:32	0	Revisar, Editar, Activar, Borrar, Ver
Tokens empresa - 1	Respuesta única	Inactiva	Scanneator	5 - 44	19-09-2023 11:11	03-10-2023 23:34	0	Revisar, Editar, Activar, Borrar, Ver
Examen parcial	Respuesta corta	Activa	Sin definir	3 - 20	16-11-2021 17:17	21-11-2021 22:22	0	Revisar, Editar, Desactivar, Auto, Ver
Encuesta - 500 puntos	Respuesta corta	Activa	Sin definir	1 - 8	06-01-2022 12:34	27-01-2022 12:34	0	Revisar, Editar, Desactivar, Auto, Ver
Mascarilla	Respuesta única	Activa	Sin definir	2 - 11	07-09-2023 18:00	26-09-2023 01:23	0	Revisar, Editar, Desactivar, Auto, Ver
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	1	Revisar, Editar, Activar, Borrar, Ver
Tokens empresa - 2	Respuesta única	Inactiva	Scanneator	5 - 44	20-09-2023 13:00	04-10-2023 22:00	1	Revisar, Editar, Activar, Borrar, Ver

Ilustración 54 Prueba de flecha menor a mayor en nivel

Añadir nueva actividad Copiar actividades Importar actividades

Puede filtrar actividades que cumplan ciertas características

Título Tipo de Actividad Estado Medalla Puntos Min Puntos Max Día Inicio Día Fin Nivel

Seleccione Seleccione Seleccione Seleccione 1970-01-01 00:00:00 1970-01-01 00:00:00 5 Filtrar

Seleccione la actividad a gestionar:

Título	Tipo	Estado	Medalla	Puntuación	Fecha Inicio	Fecha Fin	Nivel					
Tokens empresa - 7	Respuesta única	Inactiva	Scanneator	5 - 44	19-11-2023 21:00	29-11-2023 13:55	5	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 6	Respuesta única	Inactiva	Scanneator	5 - 44	09-10-2023 15:00	19-10-2023 10:00	4	Revisar	Editar	Activar	Borrar	Ver
Música en Clase práctica TDS	Respuesta única	Activa	Sin definir	3 - 22	24-11-2022 17:00	30-11-2022 21:22	4	Revisar	Editar	Desactivar	Auto	Ver
Tokens empresa - 4	Respuesta única	Inactiva	Scanneator	5 - 44	22-09-2023 15:55	06-10-2023 15:00	3	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 5	Respuesta única	Inactiva	Scanneator	5 - 44	17-12-2023 10:00	27-12-2023 10:00	3	Revisar	Editar	Activar	Borrar	Ver
Tokens varios	Respuesta múltiple	Inactiva	Scanneator	10 - 70	19-12-2023 23:05	29-12-2023 05:05	2	Revisar	Editar	Activar	Borrar	Ver
Tokens varios - 2	Respuesta múltiple	Inactiva	Scanneator	10 - 75	09-12-2023 12:05	19-12-2023 22:00	2	Revisar	Editar	Activar	Borrar	Ver
Acciones semánticas del Léxico	Respuesta múltiple	Inactiva	Scanneator	5 - 60	28-12-2023 16:00	08-01-2024 08:00	2	Revisar	Editar	Activar	Borrar	Ver
Tokens del programa	Respuesta múltiple	Inactiva	Scanneator	5 - 44	22-09-2023 19:00	07-10-2023 11:22	2	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 3	Respuesta única	Inactiva	Scanneator	5 - 44	21-09-2023 00:00	05-10-2023 22:01	2	Revisar	Editar	Activar	Borrar	Ver
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	1	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 2	Respuesta única	Inactiva	Scanneator	5 - 44	20-09-2023 13:00	04-10-2023 22:00	1	Revisar	Editar	Activar	Borrar	Ver

Ilustración 55 Prueba de flecha mayor a menor en nivel

Como se puede ver en las ilustraciones 54 y 55, las flechas funcionan como se desean.

A continuación, se probará el funcionamiento del filtro en sí.

Añadir nueva actividad Copiar actividades Importar actividades

Puede filtrar actividades que cumplan ciertas características

Título Tipo de Actividad Estado Medalla Puntos Min Puntos Max Día Inicio Día Fin Nivel

Seleccione Seleccione Seleccione Seleccione 1970-01-01 00:00:00 1970-01-01 00:00:00 8 Filtrar

Seleccione la actividad a gestionar:

Título	Tipo	Estado	Medalla	Puntuación	Fecha Inicio	Fecha Fin	Nivel					
Tokens empresa - 7	Respuesta única	Inactiva	Scanneator	5 - 44	19-11-2023 21:00	29-11-2023 13:55	5	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 6	Respuesta única	Inactiva	Scanneator	5 - 44	09-10-2023 15:00	19-10-2023 10:00	4	Revisar	Editar	Activar	Borrar	Ver
Música en Clase práctica TDS	Respuesta única	Activa	Sin definir	3 - 22	24-11-2022 17:00	30-11-2022 21:22	4	Revisar	Editar	Desactivar	Auto	Ver
Tokens empresa - 4	Respuesta única	Inactiva	Scanneator	5 - 44	22-09-2023 15:55	06-10-2023 15:00	3	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 5	Respuesta única	Inactiva	Scanneator	5 - 44	17-12-2023 10:00	27-12-2023 10:00	3	Revisar	Editar	Activar	Borrar	Ver
Tokens varios	Respuesta múltiple	Inactiva	Scanneator	10 - 70	19-12-2023 23:05	29-12-2023 05:05	2	Revisar	Editar	Activar	Borrar	Ver
Tokens varios - 2	Respuesta múltiple	Inactiva	Scanneator	10 - 75	09-12-2023 12:05	19-12-2023 22:00	2	Revisar	Editar	Activar	Borrar	Ver
Acciones semánticas del Léxico	Respuesta múltiple	Inactiva	Scanneator	5 - 60	28-12-2023 16:00	08-01-2024 08:00	2	Revisar	Editar	Activar	Borrar	Ver
Tokens del programa	Respuesta múltiple	Inactiva	Scanneator	5 - 44	22-09-2023 19:00	07-10-2023 11:22	2	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 3	Respuesta única	Inactiva	Scanneator	5 - 44	21-09-2023 00:00	05-10-2023 22:01	2	Revisar	Editar	Activar	Borrar	Ver
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	1	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 2	Respuesta única	Inactiva	Scanneator	5 - 44	20-09-2023 13:00	04-10-2023 22:00	1	Revisar	Editar	Activar	Borrar	Ver

Ilustración 56 Filtrado por nivel en manageActivities.php

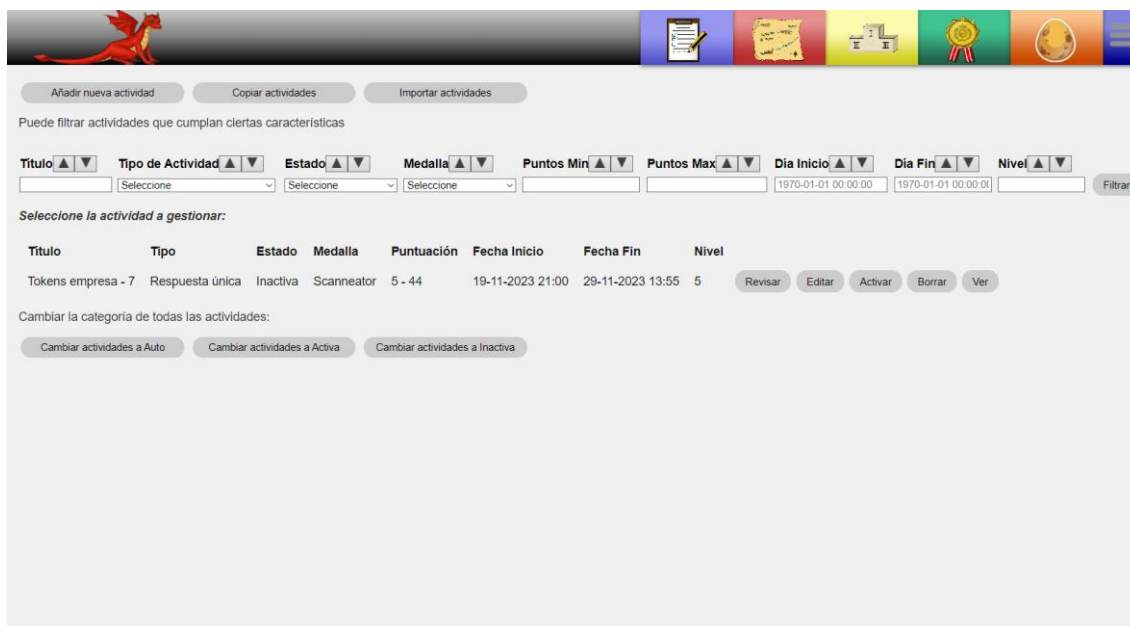


Ilustración 57 Resultado del filtrado en la ilustración 54

Como se puede ver en las ilustraciones 56 y 57, el filtrado funciona como debe.

4.3 Solución al problema con los lenguajes definidos

4.3.1 Diseño

Tal y como se ha hecho en apartados anteriores, se comienza analizando la arquitectura de la página en su actualidad.

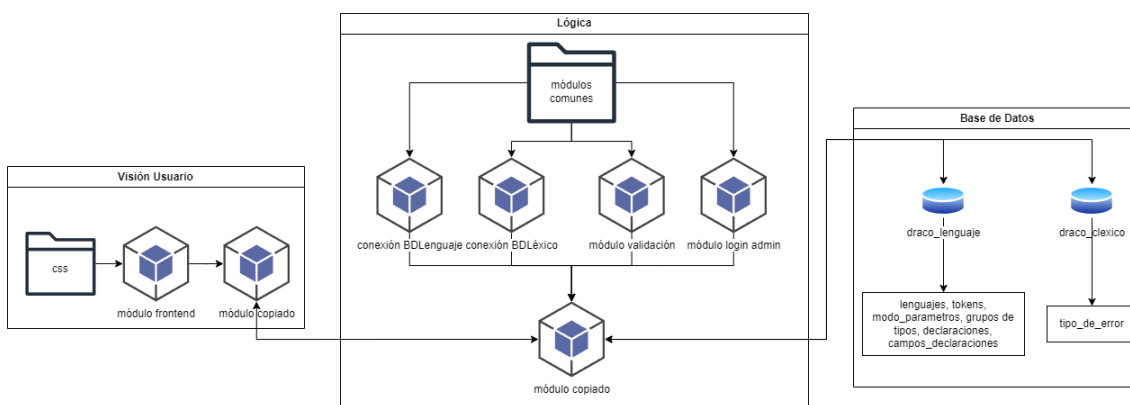


Ilustración 58 arquitectura de la página de copiar lenguaje

Cuando se observa esta página, se pueden ver dos módulos de copiado. Esto en verdad representa solo un módulo de copiado en el cual hay una parte *frontend* y una parte de lógica. También se observa que dentro de los módulos comunes que ya han sido explicados en anteriores soluciones, se observa un nuevo módulo llamado módulo de validación. Además del módulo de validación, también hay dos módulos de conexión para cada base de datos a la que se conecta, mostradas también en la ilustración 58. Cuando se quiere realizar la copia de un lenguaje, la vista donde se muestran los lenguajes, llamado en la ilustración 58 como módulo *frontend*, se redirige a la parte visual del módulo de copiado. Dentro del módulo de copiado hay un pequeño formulario donde se

mete información como año, semestre y nombre que permite asegurarse que no habrá dos lenguajes activados simultáneamente. Este formulado es enviado y recibido por el módulo de copiado, que mostrará en su parte *frontend* errores en caso de que haya. En caso de que no haya errores, introducirá el nuevo lenguaje copiado en la base de datos con la información introducida en el formulario.

Observando la arquitectura se puede ver el origen del problema: Al crear el sistema de copia, no se tuvo en cuenta los errores sintácticos y semánticos (dado que estos módulos fueron añadidos posteriormente). La prueba de ello está en que sólo se comunica con la base de datos de léxico, pero en ningún momento se comunica con la base de datos de ningún otro tipo de error.

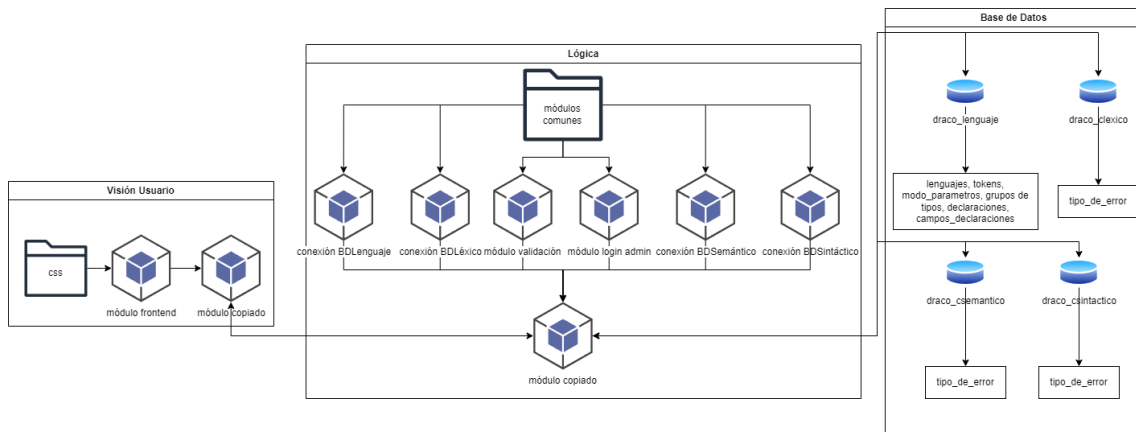


Ilustración 59 arquitectura nueva propuesta

Esta nueva arquitectura mostrada en la ilustración 59 solucionaría el problema, introduciendo dos nuevos módulos de conexión a las bases de datos donde están los errores semánticos y sintácticos, permitiendo la nueva comunicación con las bases de datos correspondientes por parte del módulo de copiado.

Esto también vendría acompañado de una expansión de la lógica del módulo de copiado, ya que actualmente no contiene ningún tipo de lógica que permita procesar estas solicitudes.

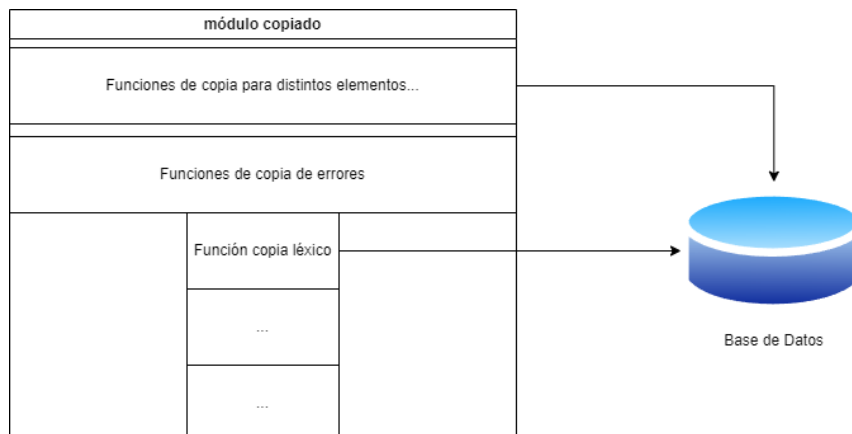


Ilustración 60 funcionamiento del módulo de copiado para los errores

Tal y como está la lógica actualmente, como se muestra en la ilustración 60, no existen funciones que se encarguen de copiar los errores semánticos y sintácticos. Para ello se procedería a su creación.

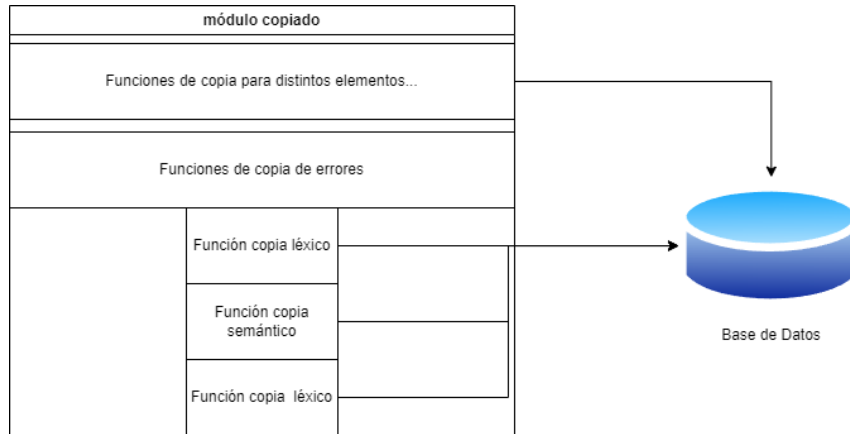


Ilustración 61 nuevo funcionamiento del módulo de copiado para los errores

Su nuevo funcionamiento se describe en la ilustración 61, donde también se incluyen las nuevas funciones de copia que estaban ausentes.

4.3.2 Implementación

Dentro del apartado de prácticas en la vista del profesor, hay un botón de “administrar lenguajes” donde se determina el lenguaje que se utilizará en las prácticas del curso de PDL.



Ilustración 62 Página de lenguajes definidos

En esta página, mostrada en la ilustración 62, se pueden ver lenguajes definidos en años anteriores y, en caso de que se hayan creado ya, lenguajes presentes y futuros.

Se observa el código fuente de la página de la ilustración 62 para poder encontrar la solución correctamente:

```
//Botón para copiar
```

```
print"<label for='boton$tok3' title='copiar' class='sr-only'>Copiar lenguaje</label>
```

```
<a href='copiarLenguaje.php?id_lenguaje=$id_lenguaje'><input id='boton$tok3'
value='Copiar lenguaje' type='button' /></a>";
```

Aquí se ve el momento que se crea el botón “Copiar Lenguaje” en el fichero IndexAdmin.php, que corresponde a la ilustración 62. Se puede observar que su funcionamiento está definido en otro fichero llamado copiarLenguaje.php.

Cuando se lee este fichero detenidamente, buscando específicamente en qué parte se copian las definiciones de los errores, se puede observar que existe un fragmento de código que copia la definición de errores léxicos, pero no existe ningún código que haga lo mismo para los errores sintácticos y semánticos, tal y como se definió en el apartado de diseño.

```
//Copiamos los tipos de error léxicos
$result = mysqli_query($conexionLexico,"SELECT * FROM tipo_de_error WHERE
id_lenguaje=$id_lenguaje;");
$errores = mysqli_fetch_all($result,MYSQLI_ASSOC);
$id = null;
$mensaje = null;
$explicacion = null;
$nombre = null;
$nombreprams = null;
if (!empty($errores)) {
    $stmt = mysqli_prepare($conexionLexico,"INSERT INTO tipo_de_error (id, error,
explicacion, nombre, id_lenguaje, nombreprams) VALUES
(?,?,?,?,$nuevo_id,?);");
$stmt->bind_param("issss",$id,$mensaje,$explicacion,$nombre,$nombreprams);
foreach ($errores as $error) {
    $id = $error["id"];
    $mensaje = $error["error"];
    $explicacion = $error["explicacion"];
    $nombre = $error["nombre"];
    $nombreprams = $error["nombreprams"];
    $stmt->execute();
}
}
```

El fragmento de código se encarga de preparar y enlazar una sentencia SQL para insertar datos en la tabla tipo_de_error. La función mysqli_prepare prepara una consulta SQL para su ejecución, definiendo un comando INSERT que agrega un nuevo registro con los campos id, error, explicacion, nombre, id_lenguaje (fijado como \$nuevo_id), y nombreprams. Los signos de interrogación (`?`) actúan como marcadores de posición para los valores que se insertarán dinámicamente. Luego, la función bind_param enlaza las variables correspondientes (\$id, \$mensaje, \$explicacion, \$nombre, \$nombreprams) a estos marcadores de posición, especificando sus tipos de datos: issss indica que el primer parámetro es un entero (i), y los restantes son cadenas (s). Esto asegura que los valores sean pasados de manera segura y en el formato correcto antes de ejecutar la sentencia preparada. Este enfoque mejora la eficiencia y protege contra inyecciones SQL.

Basándose en este código, se puede hacer dos códigos casi idénticos que hagan esta misma labor y función para los errores sintácticos y semánticos.

La principal diferencia entre los nuevos códigos y el código original será la conexión de la base de datos. La conexión que se hace en la copia de errores léxicos se hace a través de una variable llamada \$conexionLexico. Cuando se observa el código detenidamente, se ve que esta variable no está en copiarLenguaje.php, sino en un archivo requerido al principio del código:

```
require_once '../common/ConexionBDLexico.php';
```

Cuando se observa ConexionBDLexico.php, se puede confirmar que es el origen de esta variable:

```
$conexionLexico = connect_to_local_db("draco_cllexico", false);
```

Esto significa que se conecta específicamente a la base de datos que hay dentro de DRACO dedicada al léxico. Existen también bases de datos llamadas draco_csintactico y draco_csemantico, a las que será necesario conectarse para hacer estos nuevos fragmentos de código y para poder hacer una solución correcta. Por ello se crearán dos ficheros nuevos que harán la conexión y serán referenciados en copiarLenguaje.php.

Estos ficheros serán ConexionBDSintactico.php:

```
<?php
include_once $path_to_root.'common/db.php';

$conexionSintactico = connect_to_local_db("draco_csintactico", false);

?>
```

Y ConexionBDSemantico.php:

```
<?php
include_once $path_to_root.'common/db.php';
$conexionSemantico = connect_to_local_db("draco_csemantico", false);
?>
```

Y se referenciará a ambos en copiarLenguaje.php.

```
require_once '../common/ConexionBDSemantico.php';
require_once '../common/ConexionBDSintactico.php';
```

Finalmente dando acceso a las nuevas variables de conexión necesarias, que permiten crear el código para poder copiar ambas definiciones de errores, se obtiene el código necesario.

```
// Copiar errores sintácticos
$resultSintactico = mysqli_query($conexionSintactico, "SELECT * FROM tipo_de_error
WHERE id_lenguaje=$id_lenguaje;");
$erroresSintactico = mysqli_fetch_all($resultSintactico, MYSQLI_ASSOC);
if (!empty($erroresSintactico)) {
    $stmtSintactico = mysqli_prepare($conexionSintactico, "INSERT INTO
tipo_de_error (id, error, explicacion, nombre, id_lenguaje, nombreprams) VALUES
(?,?,?,?,?,?);");
    $stmtSintactico->bind_param("issss", $id, $mensaje, $explicacion, $nombre,
$nuevo_id, $nombreprams);
    foreach ($erroresSintactico as $error) {
        $id = $error["id"];
        $mensaje = $error["error"];
        $explicacion = $error["explicacion"];
        $nombre = $error["nombre"];
        $nombreprams = $error["nombreprams"];
        $stmtSintactico->execute();
    }
}
// Copiar errores semánticos
$resultSemantico = mysqli_query($conexionSemantico, "SELECT * FROM tipo_de_error WHERE
id_lenguaje=$id_lenguaje;");
$erroresSemantico = mysqli_fetch_all($resultSemantico, MYSQLI_ASSOC);
if (!empty($erroresSemantico)) {
```

```

$stmtSemantico = mysqli_prepare($conexionSemantico, "INSERT INTO tipo_de_error
(id, error, explicacion, nombre, id_lenguaje, nombreparams) VALUES
(?,?,?,?,?,?);");
$stmtSemantico->bind_param("isssss", $id, $mensaje, $explicacion, $nombre,
$nuevo_id, $nombreprams);
foreach ($erroresSemantico as $error) {
    $id = $error["id"];
    $mensaje = $error["error"];
    $explicacion = $error["explicacion"];
    $nombre = $error["nombre"];
    $nombreprams = $error["nombreprams"];
    $stmtSemantico->execute();
}
}

```

Cuando se observa la implementación del diseño mencionado, es importante especificar el por qué el código nuevo es distinto al anterior.

Se puede observar que hay una diferencia sutil con el anterior código: El nuevo id se inserta con el resto de los parámetros en vez de meterse directamente en el *insert into* definido en el `mysqli_prepare`.

Es mejor estructurar la consulta de esta manera porque se mantiene consistente al usar marcadores de posición (?) para todos los valores, incluidas las variables dinámicas como `$nuevo_id`. Esto permite que todos los parámetros sean manejados por la función `bind_param`, lo que garantiza una validación y formateo adecuados de los datos antes de ejecutar la consulta. Además, evita mezclar valores directamente en la consulta SQL, reduciendo significativamente el riesgo de inyecciones SQL y mejorando la claridad del código al centralizar toda la lógica de asignación de valores en `bind_param`. Esta forma también es más flexible y escalable, ya que permite cambios dinámicos en los valores sin alterar la estructura de la consulta SQL.

Esta mejora también se implementará en el código ya existente.

4.3.3 Pruebas

Para ver si la solución funciona correctamente, se realizará una copia de un lenguaje y se verá si se han copiado bien todos los errores sintácticos y semánticos.

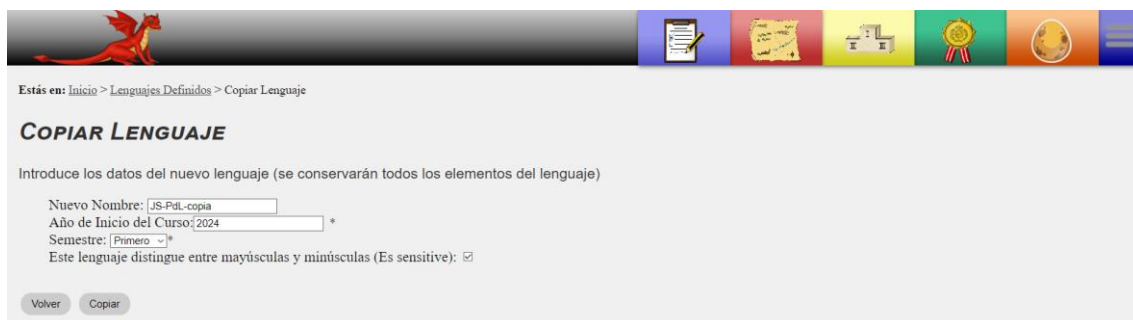


Ilustración 63 Pantalla mostrada al copiar el lenguaje de PDL mas reciente

La ilustración 63 muestra cómo se crea el lenguaje copiado. Es importante cambiar el año de inicio del curso, ya que no puede haber dos lenguajes simultáneos en el mismo semestre.

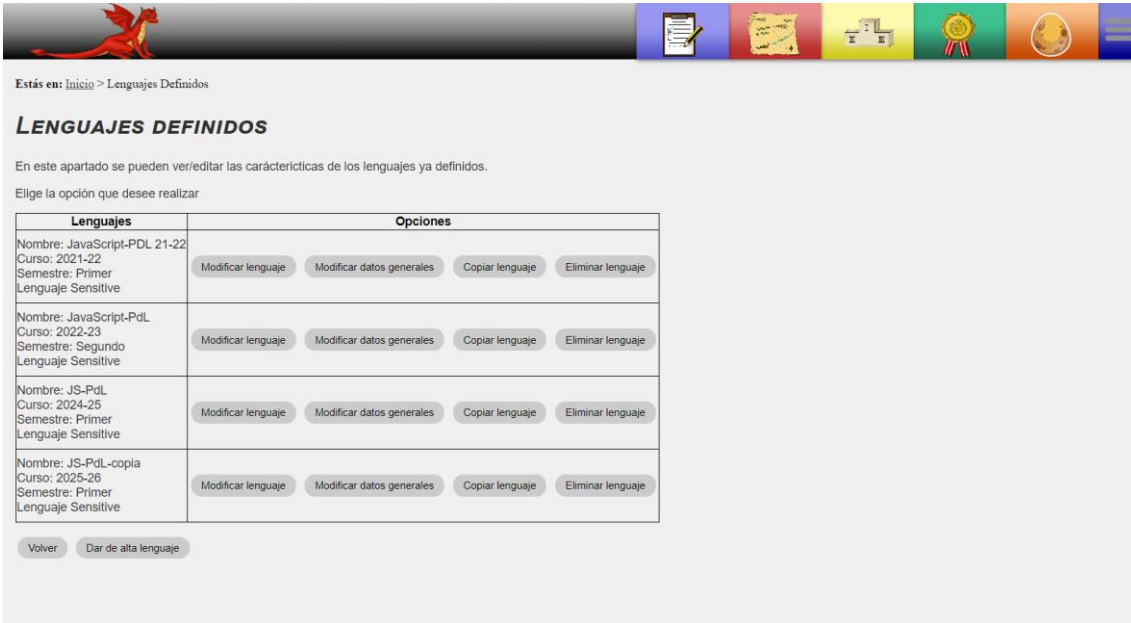


Ilustración 64 Pantalla que incluye el lenguaje copiado

Una vez el lenguaje ha sido copiado tal y como se muestra en la ilustración 64, hay que ver si están bien metidos los errores. Para ello se selecciona el botón modificar lenguaje, que permite ver y modificar todas las variables del lenguaje.

El operador lógico or se tiene que escribir duplicando el carácter '|', es decir '||'.

(Es necesario tener ejemplos de cuatro tipos de error para que el sistema genere pruebas con errores)

ERRORES SINTÁCTICOS

Código	Nombre	Mensaje	Explicación	Parámetros	Eliminar
T1	Sentencia sin fin de sentencia	Falta el punto y coma del final de la sentencia	Las sentencias simples deben acabar siempre		<input type="checkbox"/>
T2	Falta cerrar paréntesis	En una expresión se ha abierto un paréntesis	Los paréntesis deben estar balanceados. La		<input type="checkbox"/>
T3	Sobra cerrar paréntesis	En una expresión se ha cerrado un paréntesis	Los paréntesis deben estar balanceados. La		<input type="checkbox"/>
T4	No pueden aparecer dos operandos consecutivos	Hay una expresión con dos operandos, pero	Cuando una expresión tiene dos operandos, el	Operandouno, Operandodos	<input type="checkbox"/>
T5	Falta un operando	El operador binario '%1' requiere dos operandos	Los operadores binarios necesitan tener dos o	Operador	<input type="checkbox"/>
T6	Operador unario sin variable	El operador '%1' debe ir siempre acompañado	Los operadores de auto-incremento o auto-decr	Operador	<input type="checkbox"/>
T7	Falta el tipo en la declaración de la variable	La variable '%1' que se está declarando carece	Al declarar una variable (en este caso, '%1') ha	Variable	<input type="checkbox"/>
T8	Falta la variable en la declaración	Se está intentando declarar una variable, pero	La instrucción 'let' requiere poner el nombre de		<input type="checkbox"/>
T9	No se puede declarar una variable de tipo void	Se está intentando declarar la variable '%1' d	No se pueden declarar variables de tipo 'void'.	Variable	<input type="checkbox"/>
T10	Falta la expresión en la instrucción put	La instrucción put debe ir seguida de una expres	La instrucción 'put' imprime el valor de una expres		<input type="checkbox"/>
T11	Falta el parámetro en la instrucción get	La instrucción get debe ir seguida de una variable	La instrucción 'get' lee un valor del teclado y s		<input type="checkbox"/>
T12	La instrucción get no puede llevar una expresión	La instrucción get debe ir seguida de una variable	La instrucción 'get' lee un valor del teclado y s		<input type="checkbox"/>
T13	Falta la expresión en una asignación	Falta la expresión en la asignación a la variable	La sentencia de asignación a '%1' necesita de	Variable	<input type="checkbox"/>
T14	Parámetro vacío erróneo en la llamada a una función	En la llamada a la función '%1' no se puede	Las llamadas a las funciones sin parámetros s	Función	<input type="checkbox"/>
T15	Faltan los paréntesis en la condición de la sentencia	La condición de la sentencia '%1' va siempre e	La sentencia de control de flujo '%1' lleva una	Sentencia	<input type="checkbox"/>
T16	Falta el tipo de retorno de la función	En la declaración de la función '%1' se ha o	Todas las funciones deben indicar el tipo de re	Función	<input type="checkbox"/>
T17	Faltan los parámetros en la declaración de la función	En la declaración de la función '%1' se han	Todas las funciones deben indicar el tipo de lo	Función	<input type="checkbox"/>
T18	Un parámetro de una función no puede ser de tipo void	En la declaración de la función '%1' se ha p	Los parámetros de las funciones no pueden s	Función, Parámetro	<input type="checkbox"/>
T19	No se permiten declarar funciones anidadas	La función '%1' se está intentando declarar	Una función no se puede declarar dentro de c	Función interna, Función externa	<input type="checkbox"/>
T20	Falta la condición en una sentencia de control de flujo	La sentencia '%1' debe llevar una condición	La sentencia de control de flujo '%1' se debe es	Sentencia	<input type="checkbox"/>
T21	La instrucción case no puede contener una variable	La instrucción case contiene la variable '%1'	Las instrucciones 'case' solamente pueden llev	Variable	<input type="checkbox"/>
T22	Se esperaba una llave, pero se encontró un punto y coma	Se ha encontrado un punto y coma al analizar	El cuerpo de la sentencia de control de flujo '%	Sentencia	<input type="checkbox"/>
T23					
T24					
T25					
T26					
T27					

Opciones para añadir al final: 5

Ilustración 65 Listado de errores sintácticos del nuevo lenguaje

(Es necesario tener ejemplos de cuatro tipos de error para que el sistema genere pruebas con errores)

ERRORES SEMÁNTICOS

Código	Nombre	Mensaje	Explicación	Parámetros	Eliminar
M1	Instrucción no permitida fuera de un 'switch'	La palabra reservada '%1' solo puede aparecer	La instrucción 'break' solamente es válida de	Sentencia	<input type="checkbox"/>
M2	Tipos incompatibles en la asignación de variable	Se está intentando asignar la variable '%2' a	Los dos tipos de una asignación deben ser ig	Var1, Var2, Tipo1, Tipo2	<input type="checkbox"/>
M3	Instrucción no permitida fuera de las funciones	La palabra reservada '%1' solo puede aparecer	La instrucción 'return' solamente es válida de	Sentencia	<input type="checkbox"/>
M4	Variable ya declarada	La variable '%1' que se intenta declarar ya es	Una variable no se puede declarar más de un	Variable	<input type="checkbox"/>
M5	Función no declarada	Se está intentando llamar a la función '%1'	Para poder llamar a la función '%1' es necesar	Función	<input type="checkbox"/>
M6	La expresión no es lógicamente	La condición de la sentencia 'if' no es	La condición de toda sentencia 'if' debe de	Sentencia	<input type="checkbox"/>
M7	El tipo de la expresión no se puede asignar a	Se está intentando asignar una expresión '%	En una asignación, ambos tipos tienen que se	Variable, TipoExp, TipoVar	<input type="checkbox"/>
M8	Tipos incompatibles en el operador aritmético	El operador '%1' requiere que ambos operan	El operador aritmético 'op' solamente	Operador	<input type="checkbox"/>
M9	Tipos incompatibles en el operador relacional	El operador '%1' requiere que ambos operan	El operador relacional 'op' solamente p	Operador	<input type="checkbox"/>
M10	Tipos incompatibles en el operador lógico	El operador '%1' requiere que sus operandos	El operador lógico 'op' solamente pue	Operador	<input type="checkbox"/>
M11	Tipo no válido en operador de auto-incremento	El operador '%1' solo puede aplicarse a varia	El operador '%1' solamente puede aplicarse a v	Operador, Tipo	<input type="checkbox"/>
M12	Tipo no válido en instrucción de asignación	El operador de asignación con operación '%	El operador de asignación con la operación	OperadorAs, Tipo	<input type="checkbox"/>
M13	Redeclaración de una variable global declarada	La variable '%1' ya se había declarado implíc	Si una variable '%1' estaba declarada implíc	Variable	<input type="checkbox"/>
M14	Tipo incompatible en instrucción de entrada/salida	El tipo indicado en la instrucción '%1' no es v	La instrucción '%1' solamente admite tipos ent	InstrucciónES	<input type="checkbox"/>
M15	Número incorrecto de parámetros en la llamada	La llamada a la función '%1' tiene '%2' pará	El número de parámetros actuales '%2' que	Función, NAmParAct, NAmParForm	<input type="checkbox"/>
M16	Llamada a función con un parámetro de tipo	Se está haciendo una llamada a la función '%	La función '%1' está declarada con el '%2' par	Función, OrdenPar, TipoPAct, TipoPFormal	<input type="checkbox"/>
M17	El tipo del 'return' no coincide con el declarado	La función '%1' tiene un 'return' con u	La función '%1' se ha declarado que devuelve	Función, TipoRet, TipoFunc	<input type="checkbox"/>
M18	La expresión de la sentencia de selección múltiple	La expresión de la instrucción 'switch' no	Todas las expresiones de la instrucción 'sw	Tipo	<input type="checkbox"/>
M19	Un nombre de función no se puede usar como	El identificador '%1' está declarado como un	El nombre de una función solamente se pued	Función	<input type="checkbox"/>
M20					
M21					
M22					
M23					
M24					

Opciones para añadir al final: Aplicar

Volver Anterior Guardar Siguiente

Ilustración 66 Listado de errores semánticos del nuevo lenguaje

Una vez se entra y se navega en la página, se puede observar en la ilustración 65 y 66, que los errores se han copiado satisfactoriamente y por tanto se ha resuelto el problema.

4.4 Problema con el borrado de lenguajes

4.4.1 Diseño

Se comienza analizando la arquitectura de la página del problema, mostrada en la ilustración 67.

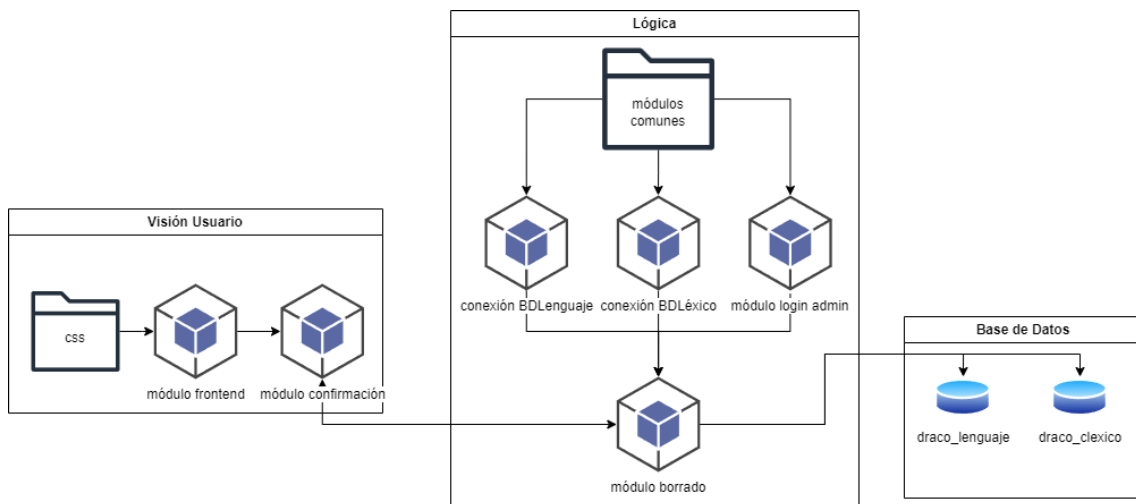


Ilustración 67 Arquitectura de borrado de lenguajes

Esta arquitectura es muy similar a la del apartado anterior ya que ambas provienen del mismo módulo *frontend*. La principal diferencia es que el módulo principal de esta arquitectura, el módulo de borrado no tiene una función *frontend* y una función lógica, sino que hay un módulo *frontend* intermedio entre la lógica y el *frontend* original que permite asegurar que el usuario tiene la

intención de borrar un lenguaje. Además, se vuelve a observar una ausencia en la arquitectura de la infraestructura necesaria para poder eliminar los errores semánticos y sintácticos.

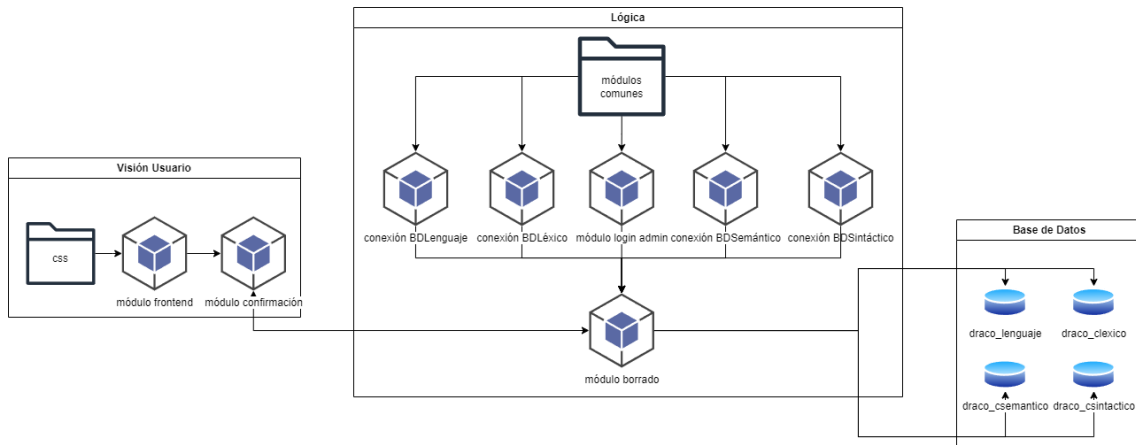


Ilustración 68 Nueva arquitectura de borrado de lenguajes

Esta nueva arquitectura expuesta en la ilustración 68 tiene la capacidad de modificar todas las bases de datos necesarias para el correcto borrado de un lenguaje, gracias a la inclusión de las nuevas dependencias de conexión que permiten conectar a la base de datos de manera correcta.

Finalmente, habría que modificar el módulo de borrado para que ahora también realizase el borrado de los errores semánticos y sintácticos.

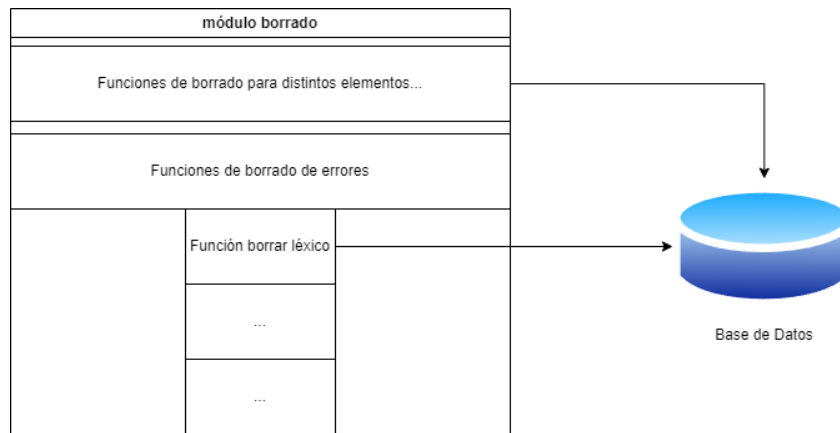


Ilustración 69 Lógica del módulo de borrado

Esta lógica actual, mostrada en la ilustración 69, no tiene ninguna función que permita borrar los errores semánticos y sintácticos.

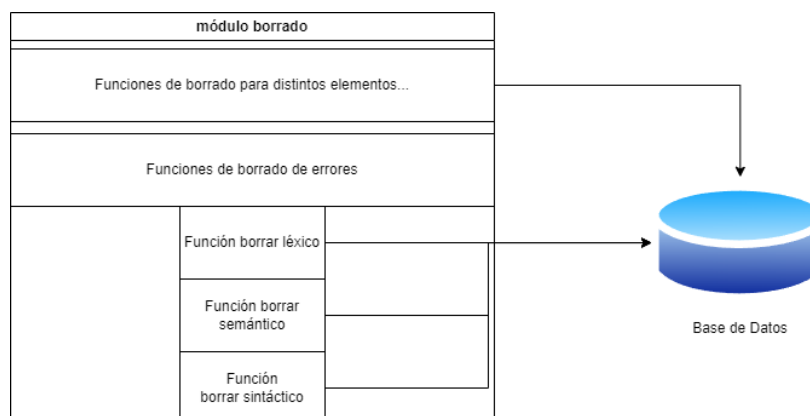


Ilustración 70 Nueva lógica del módulo de borrado

Se propone, tal y como muestra la ilustración 70, crear funciones que hagan esta labor para poder complementarse con la nueva arquitectura y solucionar el problema.

4.4.2 Implementación

En la misma interfaz inicial, mostrada en la ilustración 61, también se puede ver un botón de borrado. Antes de solucionar el problema que se solucionó en el apartado anterior, no parecía haber ningún problema con el funcionamiento del botón de borrado. El problema es que, una vez solucionado este problema, al utilizar el botón de borrado, se observó que el borrado se hacía de todo excepto de los errores semánticos y sintácticos. Se observa el funcionamiento del botón:

```
print"<label for='boton$tok4' title='eliminar' class='sr-only'>Eliminar
lenguaje</label>
<a href='ConfirmarEliminarLenguaje.php?id_lenguaje=$id_lenguaje' ><input
id='boton$tok4' value='Eliminar lenguaje' type='button' /></a>";
```

Se puede ver que el funcionamiento depende inicialmente del archivo ConfirmarEliminarLenguaje.php que pide confirmar la eliminación del lenguaje. Si se observa ese archivo, se ve el momento en el que se confirmaría la eliminación en este fragmento:

```
print"¿Está seguro de querer eliminar el lenguaje?<br>";
print"<label for='volver' >
<a href='IndexAdmin.php'><input id='volver' value='No' type='button' /></a>
</label>";
print"<label for='continuar' >
<a href='actions/eliminarLenguaje.php?id_lenguaje=$_id_lenguaje' ><input
id='continuar' value='Si' type='button' /></a>
</label>";
```

En este fragmento se ve que, al confirmar la eliminación del lenguaje, redirige a un archivo llamado eliminarLenguaje.php. Cuando se observa ese archivo, se obtiene este fragmento de código:

```
$sql_delete_errores_1 = "DELETE FROM tipo_de_error WHERE id_lenguaje=$_id_lenguaje";
mysqli_query($conexionLexico,$sql_delete_errores_1);
```

Y de nuevo se puede observar que este es el único tipo de error que se elimina, ignorando los semánticos y sintácticos. Además, utiliza la variable que también se mencionó anteriormente: \$conexionLexico, que proviene del archivo ConexionBDLexico.php.

Por suerte, la mayoría del trabajo está ya hecho, ya que se crearon los archivos para las conexiones de los errores semánticos y sintácticos. Basta con añadir:

```
require_once '../././common/ConexionBDSemantico.php';  
require_once '../././common/ConexionBDSintactico.php';
```

Además de también ser necesario crear dos fragmentos de código para la eliminación de los errores semánticos y sintácticos, basados en el código de borrado del léxico. La única diferencia será la variable de conexión. El fragmento entero sería:

```
$sql_delete_errores_2 = "DELETE FROM tipo_de_error WHERE id_lenguaje=$id_lenguaje";  
mysqli_query($conexionSemantico,$sql_delete_errores_2);  
$sql_delete_errores_3 = "DELETE FROM tipo_de_error WHERE id_lenguaje=$id_lenguaje";  
mysqli_query($conexionSintactico,$sql_delete_errores_3);
```

La implementación de estos cambios sugeridos en el diseño es sencilla, ya que solo implica la modificación de código que solo depende de algunos archivos creados con anterioridad, y no requiere ningún tipo de modificación o cambio dentro de la base de datos.

Una vez se ha implementado la solución, y al refrescar la página, se puede ver que, al menos visualmente no se aprecia ningún tipo de errores. No debería de haber ningún error visual hasta que se hagan las pruebas concretas, con lo que, por el momento, parece que no hay ninguna interferencia con archivos que tenían un correcto funcionamiento.

4.4.3 Pruebas

La prueba que se realizará en esta solución implicará la eliminación de un lenguaje, en este caso el lenguaje copia que se creó en el apartado de pruebas anterior. Una vez se haya seleccionado el botón de borrar lenguaje, se comprobará manualmente en la base de datos si se ha eliminado correctamente. Para ello, primero se buscará el id del lenguaje dentro de la base de datos para poder comprobar que los errores están actualmente en la base de datos, para poder hacer la comparativa después.

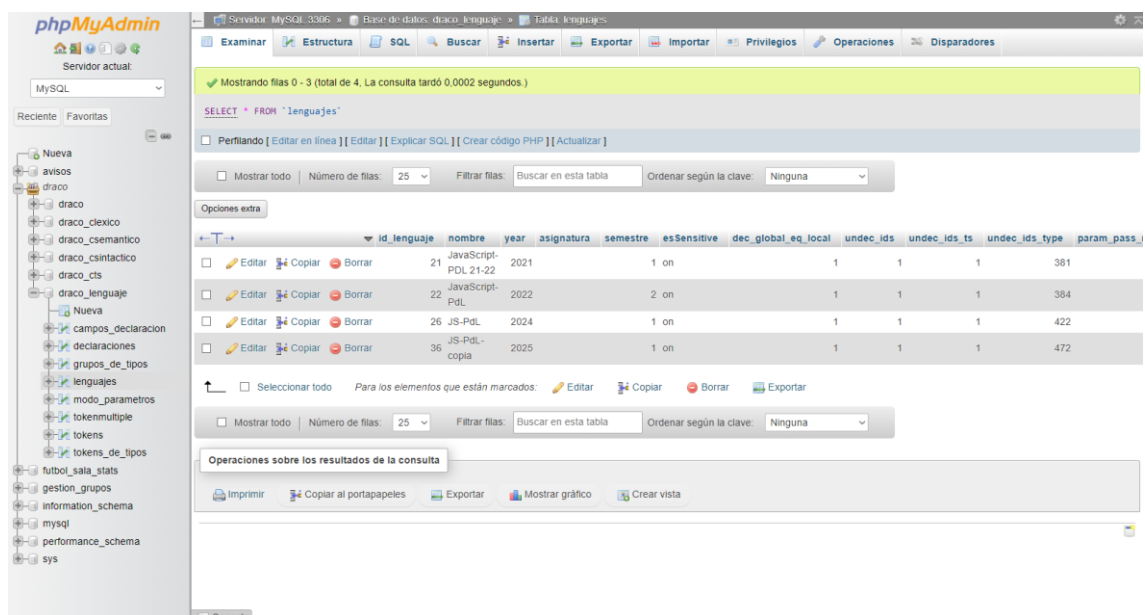


Ilustración 71 Listado de lenguajes presentes en la base de datos

id	error	explicacion	nombre	id_lenguaje	nombrepars
1	El número entero %1 se sale del rango permitido	Los números se deben representar con 16 bits. El ...	Número entero fuera de rango	22	Número
1	El número entero %1 se sale del rango permitido	Los números se deben representar con 16 bits (es ...	Número entero fuera de rango	26	Número
1	El número entero %1 se sale del rango permitido	Los números se deben representar con 16 bits (es ...	Número entero fuera de rango	36	Número
2	El carácter %1 no pertenece al lenguaje	El carácter %1 en "%0" no pertenece al lenguaje...	Carácter desconocido	22	Carácter
2	El carácter %1 no pertenece al lenguaje	El carácter %1 no pertenece al lenguaje.	Carácter desconocido	26	Carácter
2	El carácter %1 no pertenece al lenguaje	El carácter %1 no pertenece al lenguaje.	Carácter desconocido	36	Carácter
3	La cadena %1 es demasiado larga	Las cadenas no pueden tener más de 64 caracteres	Cadena demasiado larga	22	Cadena
3	La cadena %1 [sin las comillas] es demasiado larg	Las cadenas no pueden tener más de 64 caracteres	Cadena demasiado larga	26	Cadena
3	La cadena %1 [sin las comillas] es demasiado larg	Las cadenas no pueden tener más de 64 caracteres	Cadena demasiado larga	36	Cadena
4	El operador %1 no pertenece al lenguaje	El operador lógico %2 se tiene que escribir dupl...	Operador lógico no válido	22	Carácter, Operador
4	El operador %1 no pertenece al lenguaje	El operador lógico %2 se tiene que escribir dupl...	Operador lógico no válido	26	Operador no válido, Nombre del operador correcto
4	El operador %1 no pertenece al lenguaje	El operador lógico %2 se tiene que escribir dupl...	Operador lógico no válido	36	Operador no válido, Nombre del operador correcto
5	Las cadenas deben cerrarse	La cadena (%1) debe cerrarse en la misma línea en...	Cadena sin cerrar	26	Cadena
5	Las cadenas deben cerrarse	La cadena (%1) debe cerrarse en la misma línea en...	Cadena sin cerrar	36	Cadena
6	Los identificadores no se escriben as	Este error no se produce en este lenguaje	Nombre de variable mal escrito	26	
6	Los identificadores no se escriben as	Este error no se produce en este lenguaje	Nombre de variable mal escrito	36	

Ilustración 72 errores léxicos presentes en la base de datos

id	error	explicacion	nombre	id_lenguaje	nombrepars
1	La variable %1 ya estaba declarada	Solo se puede declarar cada variable una vez en el...	Variable declarada dos veces	22	Variable
1	La palabra reservada %1 solo puede aparecer dentro...	La instrucción 'break' solamente es válida dentro...	Instrucción no permitida fuera de un 'switch'	26	Sentencia
1	La palabra reservada %1 solo puede aparecer dentro...	La instrucción 'break' solamente es válida dentro...	Instrucción no permitida fuera de un 'switch'	36	Sentencia
2	La variable %1 no está declarada	Toda variable se debe declarar	Variable sin declarar	22	Variable
2	Se está intentando asignar la variable %2 a la v...	Los dos tipos de una asignación deben ser iguales...	Tipos incompatibles en la asignación de variables	26	Var1, Var2, Tipo1, Tipo2
2	Se está intentando asignar la variable %2 a la v...	Los dos tipos de una asignación deben ser iguales...	Tipos incompatibles en la asignación de variables	36	Var1, Var2, Tipo1, Tipo2
3	La expresión %1 no es lógica	El if requiere una expresión lógica	La expresión del if no es lógica	22	Expresión
3	La palabra reservada %1 solo puede aparecer dentro...	La instrucción 'return' solamente es válida dentro...	Instrucción no permitida fuera de las funciones	26	Sentencia
3	La palabra reservada %1 solo puede aparecer dentro...	La instrucción 'return' solamente es válida dentro...	Instrucción no permitida fuera de las funciones	36	Sentencia
4	La cadena %1 no se puede sumar con la cadena %2	La operación suma es solo entre enteros	No se pueden sumar cadenas	22	Cad1, Cad2
4	La variable %1 que se intenta declarar ya estaba	Una variable no se puede declarar más de una vez...	Variable ya declarada	26	Variable
4	La variable %1 que se intenta declarar ya estaba	Una variable no se puede declarar más de una vez...	Variable ya declarada	36	Variable
5	Se está intentando llamar a la función %1 pero...	Para poder llamar a la función %1 es necesario de...	Función no declarada	26	Función
5	Se está intentando llamar a la función %1 pero...	Para poder llamar a la función %1 es necesario de...	Función no declarada	36	Función
6	La condición de la sentencia 'if' no es de l...	La condición de toda sentencia 'if' debe ser ...	La expresión no es lógica	26	Sentencia
6	La condición de la sentencia 'if' no es de l...	La condición de toda sentencia 'if' debe ser ...	La expresión no es lógica	32	Sentencia
6	La condición de la sentencia 'if' no es de l...	La condición de toda sentencia 'if' debe ser ...	La expresión no es lógica	36	Sentencia

Ilustración 73 Errores semánticos presentes en la base de datos

id	error	explicacion	nombre	id_lenguaje	nombreparsams
1	No puede haber una llave en este lugar	No puede haber una llave en este lugar	La llave no se puede poner aquí	22	
1	Falta el punto y coma del final de la sentencia	Las sentencias simples deben acabar siempre en pun...	Sentencia sin fin de sentencia	26	
1	No puede haber una llave en este lugar	No puede haber una llave en este lugar	La llave no se puede poner aquí	30	0
1	Falta el punto y coma del final de la sentencia	Las sentencias simples deben acabar siempre en pun...	Sentencia sin fin de sentencia	32	
1	Falta el punto y coma del final de la sentencia	Las sentencias simples deben acabar siempre en pun...	Sentencia sin fin de sentencia	36	
2	Hay que poner la llave de cierre	Los bloques deben abrirse y cerrarse	El bloque no está cerrado	22	
2	En una expresión se ha abierto un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Falta cerrar paréntesis	26	
2	En una expresión se ha abierto un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Falta cerrar paréntesis	29	0
2	Hay que poner la llave de cierre	Los bloques deben abrirse y cerrarse	El bloque no está cerrado	30	0
2	En una expresión se ha abierto un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Falta cerrar paréntesis	32	
2	En una expresión se ha abierto un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Falta cerrar paréntesis	36	
3	Hay un ; sin sentencia	Las sentencias deben acabar en ; y solo las senten...	Fin de sentencia suelto	22	
3	En una expresión se ha cerrado un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Sobra cerrar paréntesis	26	
3	En una expresión se ha cerrado un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Sobra cerrar paréntesis	29	0
3	Hay un ; sin sentencia	Las sentencias deben acabar en ; y solo las senten...	Fin de sentencia suelto	30	0
3	En una expresión se ha cerrado un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Sobra cerrar paréntesis	32	
3	En una expresión se ha cerrado un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Sobra cerrar paréntesis	36	
4	No hay condición en el if	La sentencia if debe ir con una condici...	Falta la condición del if	22	

Ilustración 74 Errores sintácticos presentes en la base de datos

Como se puede observar en la ilustración 71, el id correspondiente a el lenguaje que se va a borrar es el número 36. En las ilustraciones 72, 73 y 74 se ven múltiples errores correspondientes al id_lenguaje 36.

Ahora, una vez borrado el lenguaje, se observa la base de datos en el siguiente estado para las tablas que se han mencionado en este apartado.

id_lenguaje	nombre	year	asignatura	semestre	esSensitive	dec_global	eq_local	undec_ids	undec_ids_ts	undec_ids_type	param_pass
21	JavaScript-PdL 21-22	2021		1	on			1	1	1	381
22	JavaScript-PdL	2022		2	on			1	1	1	384
26	JS-PdL	2024		1	on			1	1	1	422

Ilustración 75 Lista de lenguajes después del borrado

id	error	explicacion	nombre	id_lenguaje	nombrepars
1	El n�mero entero %1 se sale del rango permitido	Los n�meros se deben representar con 15 bits. El ...	N�mero entero fuera de rango	22	N�mero
1	El n�mero entero %1 se sale del rango permitido	Los n�meros se deben representar con 15 bits (es ...	N�mero entero fuera de rango	26	N�mero
2	El car�cter %1 no pertenece al lenguaje	El car�cter %1 en "%0" no pertenece al lenguaje...	Car�cter desconocido	22	Car�cter
2	El car�cter %1 no pertenece al lenguaje	El car�cter %1 no pertenece al lenguaje.	Car�cter desconocido	26	Car�cter
3	La cadena %1 es demasiado larga	Las cadenas no pueden tener m�s de 64 caracteres	Cadena demasiado larga	22	Cadena
3	La cadena %1 [sin las comilas] es demasiado larg...	Las cadenas no pueden tener m�s de 64 caracteres	Cadena demasiado larga	26	Cadena
4	El operador %1 no pertenece al lenguaje	El operador l�gico %2 se tiene que escribir dupl...	Operador l�gico no v�lido	22	Car�cter, Operador
4	El operador %1 no pertenece al lenguaje	El operador l�gico %2 se tiene que escribir dupl...	Operador l�gico no v�lido	26	Operador no v�lido. Nombre del operador correcto
5	Las cadenas deben cerrarse	La cadena (%1) debe cerrarse en la misma l�nea en.	Cadena sin cerrar	26	Cadena
6	Los identificadores no se escriben as�	Este error no se produce en este lenguaje	Nombre de variable mal escrito	26	
7	La variable est� repetida	Esto no es un error l�xico	Nombre de variable repetido	26	
8	El lexema no se corresponde con una palabra reserv.	Este error no se puede producir, porque el lexema	Palabra reservada no v�lida	26	
9	El comentario est� abierto y no est� cerrado	El comentario %0 se tiene que abrir con "/*" y c...	Comentario sin cerrar	26	

Ilustraci3n 76 lista de errores l3xicos despu3 del borrado

id	error	explicacion	nombre	id_lenguaje	nombrepars
1	La variable %1 ya estaba declarada	Solo se puede declarar cada variable una vez en el...	Variable declarada dos veces	22	Variable
1	La palabra reservada %1 solo puede aparecer dentr.	La instrucci3n 'break' solamente es v�lida dentr.	Instrucci3n no permitida fuera de un 'switch'	26	Sentencia
2	La variable %1 no est� declarada	Toda variable se debe declarar	Variable sin declarar	22	Variable
2	Se est� intentando asignar la variable %2 a la v.	Los dos tipos de una asignaci3n deben ser iguales...	Tipos incompatibles en la asignaci3n de variables	22	Var1, Var2, Tipo1, Tipo2
3	La expresi3n %1 no es l�gica	El if requiere una expresi3n l�gica	La expresi3n del if no es l�gica	22	Expresi3n
3	La palabra reservada %1 solo puede aparecer dentr.	La instrucci3n 'return' solamente es v�lida dent.	Instrucci3n no permitida fuera de las funciones	26	Sentencia
4	La cadena %1 no se puede sumar con la cadena %2	La operaci3n suma es solo entre enteros	No se pueden sumar cadenas	22	Cad1, Cad2
4	La variable %1 que se intenta declarar ya estaba.	Una variable no se puede declarar m�s de una vez	Variable ya declarada	26	Variable
5	Se est� intentando llamar a la funci3n %1, pero...	Para poder llamar a la funci3n %1 es necesario de.	Funci3n no declarada	26	Funci3n
6	La condici3n de la sentencia 'e' no es de l.	La condici3n de toda sentencia 'e' debe ser ...	La expresi3n no es l�gica	26	Sentencia
6	La condici3n de la sentencia 'e' no es de l.	La condici3n de toda sentencia 'e' debe ser ...	La expresi3n no es l�gica	32	Sentencia
7	Se est� intentando asignar una expresi3n %2 a la.	En una asignaci3n, ambos tipos tienen que ser igu...	El tipo de la expresi3n no se puede asignar a la ...	26	Variable, TipoExp, TipoVar
7	Se est� intentando asignar una expresi3n %2 a la.	En una asignaci3n, ambos tipos tienen que ser igu...	El tipo de la expresi3n no se puede asignar a la ...	29	0
7	Se est� intentando asignar una expresi3n %2 a la.	En una asignaci3n, ambos tipos tienen que ser igu...	El tipo de la expresi3n no se puede asignar a la ...	32	Variable, TipoExp, TipoVar
8	El operador %1 requiere que ambos operandos sean ...	El operador aritm3tico 'e' solamente puede o...	Tipos incompatibles en el operador aritm3tico	26	Operador
8	El operador %1 requiere que ambos operandos sean ...	El operador aritm3tico 'e' solamente puede o...	Tipos incompatibles en el operador aritm3tico	29	0
8	El operador %1 requiere que ambos operandos sean ...	El operador aritm3tico 'e' solamente puede o...	Tipos incompatibles en el operador aritm3tico	32	Operador

Ilustraci3n 77 lista de errores sem nticos despu3 del borrado

id	error	explicacion	nombre	id_lenguaje	nombreparams
1	No puede haber una llave en este lugar	No puede haber una llave en este lugar	La llave no se puede poner aquí	22	
1	Falta el punto y coma del final de la sentencia	Las sentencias simples deben acabar siempre en pun...	Sentencia sin fin de sentencia	26	
1	No puede haber una llave en este lugar	No puede haber una llave en este lugar	La llave no se puede poner aquí	30	0
1	Falta el punto y coma del final de la sentencia	Las sentencias simples deben acabar siempre en pun...	Sentencia sin fin de sentencia	32	
2	Hay que poner la llave de cierre	Los bloques deben abrirse y cerrarse	El bloque no está cerrado	22	
2	En una expresión se ha abierto un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Falta cerrar paréntesis	26	
2	En una expresión se ha abierto un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Falta cerrar paréntesis	29	0
2	Hay que poner la llave de cierre	Los bloques deben abrirse y cerrarse	El bloque no está cerrado	30	0
2	En una expresión se ha abierto un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Falta cerrar paréntesis	32	
3	Hay un ; sin sentencia	Las sentencias deben acabar en ; y solo las senten...	Fin de sentencia suelto	22	
3	En una expresión se ha cerrado un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Sobra cerrar paréntesis	26	
3	En una expresión se ha cerrado un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Sobra cerrar paréntesis	29	0
3	Hay un ; sin sentencia	Las sentencias deben acabar en ; y solo las senten...	Fin de sentencia suelto	30	0
3	En una expresión se ha cerrado un paréntesis, pe...	Los paréntesis deben estar balanceados. La expres...	Sobra cerrar paréntesis	32	
4	No hay condici�n en el if	La sentencia if debe ir con una condici�n	Falta la condici�n del if	22	
4	Hay una expresi�n con dos operandos, pero sin ope...	Cuando una expresi�n tiene dos operandos, es nece...	No pueden aparecer dos operandos consecutivos	26	Operando, Operandos
4	Hay una expresi�n con dos operandos, pero sin ope...	Cuando una expresi�n tiene dos operandos, es nece...	No pueden aparecer dos operandos consecutivos	29	0
4	No hay condici�n en el if	La sentencia if debe ir con una condici�n	Falta la condici�n del if	30	0

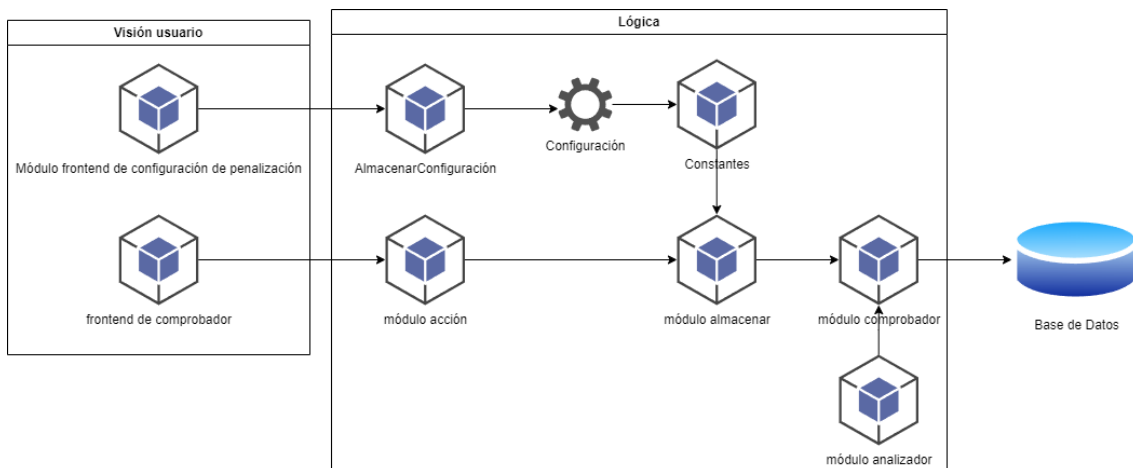
Ilustraci n 78 lista de errores sint cticos despu s del borrado

Como se puede ver en las ilustraciones 75, 76, 77 y 78, el borrado ha sido un  xito, ya que no hay rastro ni del lenguaje, ni de ning n error asociado a la id_lenguaje con el n mero 36.

4.5 Ampliaci n del funcionamiento del corrector de pr cticas para el m dulo Tabla de S mbolos

4.5.1 Dise o

Se comienza analizando la arquitectura del corrector, la cual se expone de manera visual en la ilustraci n 79.



Ilustraci n 79 Arquitectura del corrector de pr cticas de TS

Se observa que esta arquitectura es bastante distinta a lo que se ha podido ver hasta el momento. Esto es porque, entre otras cosas, este problema contiene dos p ginas relevantes. El m dulo de *frontend* de configuraci n de penalizaci n es el lugar donde el profesor puede modificar cu ntos puntos quitan los distintos errores que cometa el alumno en la tabla de s mbolos, adem s de

muchos otros valores que no son relevantes a este problema. Ese módulo envía un formulario al módulo de almacenar Configuración que se encarga de modificar, según el formulario recibido, el archivo de configuración. El módulo constantes crea variables constantes para el uso de otros módulos, extrayéndolos del archivo de configuración. Finalmente, estos se envían al módulo de almacenar, que sirve como un punto de encuentro entre ambas páginas para enviar toda la información al módulo comprobador.

La otra página consiste en la vista del estudiante. Se comienza con el *frontend* de entrega de prácticas. A continuación, al estudiante querer entregar la práctica, envía un archivo y se pasa al módulo acción. Este módulo actúa como un punto intermedio entre distintos módulos dependiendo de varios aspectos, entre ellos qué práctica quiere entregar el alumno, cuanto tiempo ha pasado desde la última entrega, etc. Todo esto se pone en común con lo obtenido del módulo constantes en el módulo almacenar.

Finalmente, el módulo almacenar envía todos los datos al módulo comprobador, que es donde se corregirá finalmente la práctica del alumno. Este módulo depende del módulo analizador, que se encarga de crear una tabla de símbolos correcta basándose en el código que se le dio al alumno, para poder comparar con el resultado entregado por el estudiante.

Se termina almacenando el resultado en la base de datos.

No es necesario hacer ninguna modificación en la arquitectura ya que no se requiere ninguna comunicación más profunda con la base de datos, y tampoco será necesaria la creación de ninguna dependencia en ninguno de los archivos. Lo principal a modificar es la lógica de los archivos.

Como se habló en el planteamiento del problema, se quiere modificar el corrector para que también detecte los atributos no vacíos de las tablas, además de detectar si hay más de una tabla. Esto significa que habría que añadir dos valores en los ficheros de configuración de penalización.

módulo frontend de configuración de penalización
Formulario con configuración de penalización
Penalización por identificador incorrecto
...
...

Ilustración 80 funcionamiento inicial del módulo frontend de configuración de penalización

Como se puede observar en la ilustración 80, actualmente no hay ninguna entrada en el formulario correcto que envíe esta información. Se tendrán que modificar los módulos de *frontend* de configuración de penalización, de almacenar configuración y de constantes para que traten con las nuevas

variables. Para ello, una opción es imitar el comportamiento que tienen estos módulos cuando se trata de penalizar los identificadores incorrectos.

Una vez estas variables ya estén disponibles, se tiene que modificar el funcionamiento del corrector para que detecte estos fallos nuevos y los tenga en cuenta a la hora de calcular la puntuación final que obtendrá el alumno.

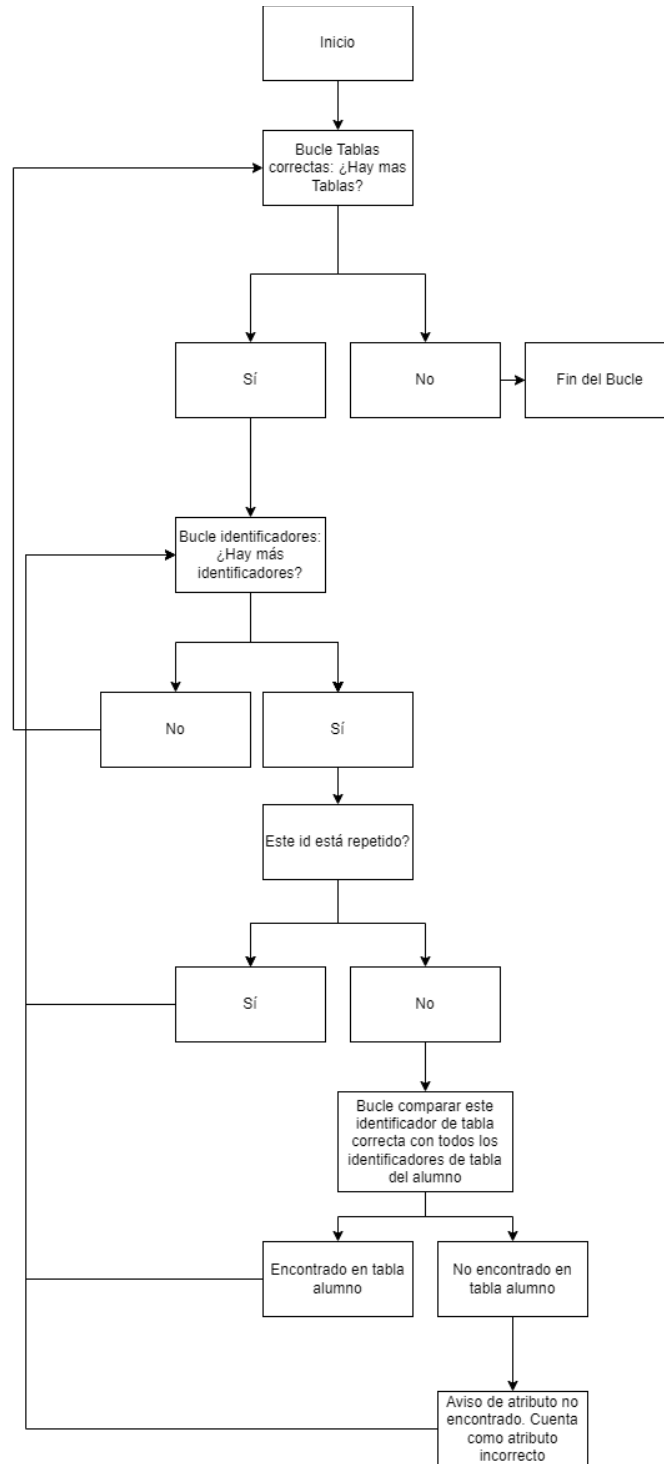


Ilustración 81 Diagrama de flujo de la lógica del módulo comparador

La ilustración 81 muestra el diagrama de flujo actual de la lógica del módulo comparador. Falta tener en cuenta cuando hay más de una tabla, además de los atributos. El diseño que se propone es el siguiente mostrado en la ilustración 82.

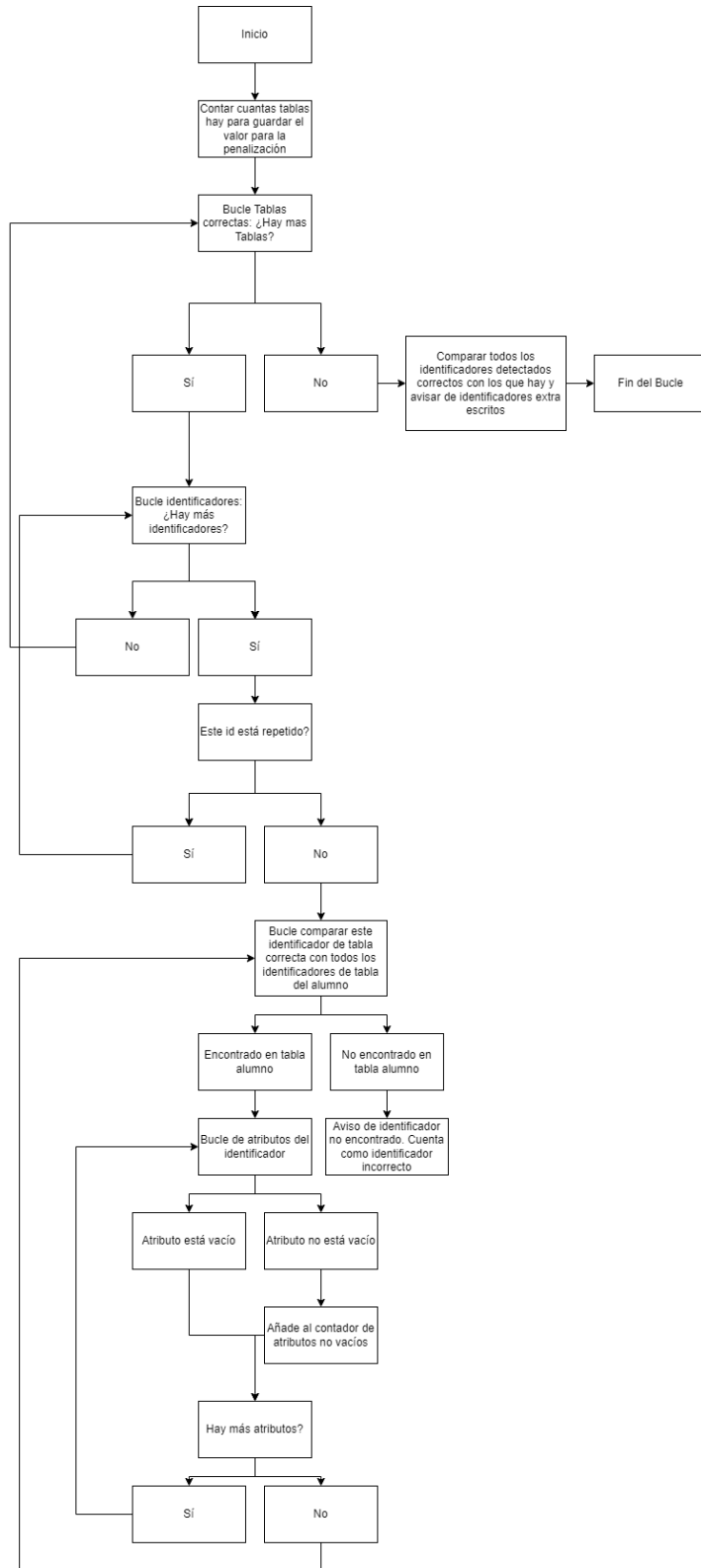


Ilustración 82 propuesta de diseño del funcionamiento de comprobador

Este diseño expuesto en la ilustración 81 añade algunas funciones importantes. Primero cuenta cuántas tablas ha metido el alumno en su solución, guardando el número para utilizarlo en el cálculo de la puntuación. A continuación, cada vez que detecta un identificador correcto, observa sus atributos y cuenta cuántos son no vacíos, añadiendo este número también a un contador para el cálculo de puntuación futura. Finalmente, se comparan todos los identificadores encontrados en la tabla del alumno con todos los identificadores correctos en la tabla de corrección, extrayendo así todos los identificadores que no tendrían que estar presentes en la tabla del alumno. Estos también se cuentan y se añaden al contador de identificadores incorrectos.

4.5.2 Implementación

Esta es, sin duda, la parte más complicada de este trabajo. Para corregir y ampliar el funcionamiento del corrector de tabla de símbolos, se tiene que analizar su comportamiento inicial. Para ello, teniendo ya una práctica de Tabla de Símbolos creada, se indaga en la vista del alumno para averiguar qué fichero se ha de analizar en un principio. En esa vista se aprecia que el fichero `Index.php`, dentro de la carpeta de prácticas específica para la tabla de símbolos, contiene el siguiente código:

```
<?php
    $modulo_activo = 'TS';
    require "../common/ControladorPrueba.php";
?>
```

Si se analiza ese código, se ve que lo único que contiene es la determinación de un atributo `$modulo_activo` a un valor de 'TS', seguido de una llamada a un fichero llamado `ControladorPrueba.php`.

Este fichero tiene muchas acciones diferentes que toma en función del valor del atributo `$modulo_activo`, que en este caso es TS como se acaba de mencionar.

En estos casos, el código solo da una instrucción que es relevante para este trabajo:

```
case "TS":
    require "AlmacenarTSyComprobar.php";
    break;
```

Con lo que se debe observar ahora el fichero `AlmacenarTSyComprobar.php`. Una vez se ha hecho eso, se puede ver que es un fichero que requiere varios archivos. Se destacan dos de los ficheros mencionados en este archivo: `Constantes.php` y `Comprobador.php`. `Constantes.php` es el fichero donde se definen todas las constantes relevantes para esta sección de las prácticas, mientras que `Comprobador.php` es el archivo donde se compara el archivo subido por el alumno a corregir con el resultado correcto, también dando la puntuación de la práctica. Este último archivo será el epicentro de este problema y de su solución.

Al observar este archivo, se comprueba que hay una función encargada de hacer la comparación:

```
public function comparar(){
    $ids_detectados = array();
    foreach ($this->tabla_correcta as $nombre_tabla => $contenido_tabla){
        foreach ($contenido_tabla as $lexema => $atributos){
            if (in_array($lexema, $ids_detectados))
```



```

Puntuacion='$puntuacion', score='$this->score' WHERE IdGrupo='$idGrupo' AND
IdPrueba='$fPrueba'");
$this->puntuacionPrueba=$puntuacion;
echo "<p class=\"pAlum\">Comprobación de la tabla de símbolos correcta,
¡Enhorabuena!</p><br>";
$texto=$puntuacion==1 ? "punto" : "puntos";
return true;
}
else{
//Faltan o sobran identificadores
$penalizacionIds=$penalizacionIds/100;
$puntos_id=$puntosACorrecto/$this->num_ids;
$puntuacion=($puntosACorrecto-($puntos_id*$this->ids_incorrectos)-
(($puntos_id*$penalizacionIds)*$this->ids_incorrectos))*$multdificultad;
$puntuacionAux=$puntosParticipar*$multdificultad;
$puntuacion=($puntuacion>=$puntuacionAux) ? round($puntuacion) :
round($puntuacionAux);
$this->score=min($puntuacion/$puntuacionMax,1);
$puntuacion=round($puntuacion*$_SESSION['per_cts']);
//Se actualiza con la puntuación final
$this->mysqli->query("UPDATE grupo_hizo_prueba SET FormaFichero='1',
Puntuacion='$puntuacion',
score='$this->score' WHERE IdGrupo='$idGrupo' AND IdPrueba='$fPrueba'");
}
}

```

En este fragmento de código, se pueden observar dos situaciones. El primer caso es que el valor de `ids_incorectos` sea igual a 0, en cuyo caso se da la puntuación máxima. En caso de `ids_incorrectos` tenga un valor distinto a 0, se hace un cálculo de puntuación. Se observa la línea exacta en la que se calculan los puntos:

```

$puntuacion=($puntosACorrecto-($puntos_id*$this->ids_incorrectos)-
(($puntos_id*$penalizacionIds)*$this->ids_incorrectos))*$multdificultad;

```

En esto, destaca que hay una variable que es global, siendo esta `penalizacionIds`. Esta variable global es una variable cuyo valor lo determina el profesor, en la página que permite manejar prácticas. Ahí se puede determinar qué porcentaje de puntuación sobre la puntuación de cada identificador quita un id incorrecto. Por ejemplo, si un id correcto da 10 puntos, y el valor de `penalizacionIds` es 10%, se quitará 1 punto por id incorrecto.

Una vez explicada esta infraestructura, se comienza a pensar en cómo se va a realizar esta solución. Como se puede ver, nuestra intención es añadir dos puntos relevantes más de comprobación: los atributos no vacíos y las tablas de símbolo extras. Para ello, será necesario tener un equivalente de `ids_incorrectos` y de `penalizacionIds` para ambos casos. En el caso de `ids_incorectos` es muy sencillo ya que es simplemente una variable determinada en local. Para ello se crearán dos variables: `ats_incorrectos` y `numeroTablas`.

En el caso de los equivalentes a `penalizacionIds` es un poco más complejo, ya que provienen de otro fichero que ya se ha mencionado `Constantes.php`.

Si se abre este fichero, se observa lo siguiente cuando se trata de `penalizacionIds`:

```

if ( strpos($bufferAux, "PenalizacionIds=")!=false){
    $penalizacionIds = dameNumTS($bufferAux, 16);
}

```

Esta función `dameNumTS`, también está definida en este fichero. Hace lo siguiente:

```
//Devuelve el numero que figure en el string cadena a partir de la posicion pos
function dameNumTS($cadena,$pos){
    $num="";
    $tam = strlen($cadena);
    while ($pos <$tam ){
        $num=$num.$cadena[$pos];
        $pos++;
    }
    return $num;
}
```

Esta función está muy bien descrita en el código: devuelve el número que figure en el *string* cadena a partir de la posición `pos`.

Esencialmente, en este código, se lee un archivo llamado `Configuración.txt`, donde todos los valores de las variables globales están guardados. Estos valores los determina el profesor en la página creada por el fichero `ConfigurarDatosSistema.php`. Este fichero llama a otro fichero, dando antes un valor 'TS' a la variable `modulo_activo`. Este fichero al que llama es `ConfigurarDatosSistemaAction.php`.

En él, se ven dos cosas relevantes: Un formulario que llama a otro fichero llamado `AlmacenarConfiguracion.php` y un fragmento de código dentro de este formulario:

```
<div class="Formulario">
    <label for="TPenIds" title="Porcentaje de los puntos correspondientes (Puntos
    por análisis correcto / número de identificadores) a restar por cada
    identificador no detectado o por cada identificador de más. Se recomienda no
    usar un valor superior al 50%"> Penalización por identificador
    incorrecto</label>
    <input type="number" id="TPenIds" name="PenIds" min="0" step="1" max="60"
    value="<?php echo (isset($newPenalizacionIds)?
    $newPenalizacionIds:$penalizacionIds);?>" />
</div>
```

Este fragmento de código PHP genera un campo de entrada de tipo numérico dentro de un formulario HTML, específicamente para configurar una penalización asociada a identificadores incorrectos. Esto se envía a `AlmacenarConfiguracion.php`.

En ese fichero, se observa un código casi idéntico al que se encuentra en `ConfigurarDatosSistema.php`, donde se llama al fichero de código `AlmacenarConfiguracionAction.php`. En este código se ve lo siguiente:

```
$newPenalizacionIds=filter_input(INPUT_POST, 'PenIds',FILTER_SANITIZE_SPECIAL_CHARS);
```

Esto recibe la información del formulario y da el valor numérico correspondiente a esta nueva variable que se usará para editar el archivo `Configuración.txt`.

```
if (!(is_numeric($newPenalizacionIds) || ($newPenalizacionIds<0))){
    $errorMessage .= "<p class='error'>La penalización por un id incorrecto debe
    ser un número mayor o igual a 0.<br></p>";
}
```

También existen condicionales que impiden que los valores de estas entradas sean menores a 0 o vacías.

Eventualmente, esta información se mete dentro del archivo de configuración en la instrucción:

```
$buffer=$buffer."PenalizacionIds=$newPenalizacionIds".PHP_EOL;
```

Con lo cual se tendrá que hacer todo esto para el análisis de atributos y de cantidad de tablas.

Se comienza creando ambas variables globales:

```
if ( strpos($bufferAux, "PenalizacionAts=")!==false){
    $penalizacionAts = dameNumTS($bufferAux, 16);
}
if ( strpos($bufferAux, "PenalizacionTabla=")!==false){
    $penalizacionTabla = dameNumTS($bufferAux, 18);
}
```

Este fragmento va seguido del código necesario para crear las entradas para el formulario en ConfigurarDatosSistemaAction.php:

```
<?php if($modulo_activo === 'TS'){ ?>
<div class="Formulario">
    <label for="TPenAts" title="Porcentaje de los puntos correspondientes (Puntos
    por análisis correcto / número de identificadores) a restar por cada atributo
    con un valor distinto a 0. Se recomienda no usar un valor superior al 50%">
    Penalización por atributo no vacío</label>
    <input type="number" id="TPenAts" name="PenAts" min="0" step="1" max="60"
    value="<?php echo (isset($newPenalizacionAts)?
    $newPenalizacionAts:$penalizacionAts);?>" />%
</div>
<div class="Formulario">
    <label for="TPenTablas" title="Porcentaje de los puntos correspondientes (Puntos
    por análisis correcto / número de identificadores) a restar por cada Tabla de
    símbolos de más que ponga el alumno. Se recomienda no usar un valor superior al
    50%">
    Penalización por Tabla de más</label>
    <input type="number" id="TPenTabla" name="PenTabla" min="0" step="1" max="60"
    value="<?php echo (isset($newPenalizacionTabla)?
    $newPenalizacionTabla:$penalizacionTabla);?>" />%
</div>
<?php } ?>
```

Se destaca que se ha metido este fragmento de código con un condicional para su ejecución. Esto es porque no es deseable que interfiera el funcionamiento con el corrector del semántico, ya que tanto este archivo como el de AlmacenarConfiguracionAction.php son comunes con el Semántico.

En este último archivo, se añade lo siguiente:

```
if ($modulo_activo == 'TS'){
    if ((!is_numeric($newPenalizacionAts)) || ($newPenalizacionAts<0)){
        $errorMessage .= "<p class='error'>La penalización por un atributo
        incorrecto debe ser un número mayor o igual a 0.<br></p>";
    }
    if ((!is_numeric($newPenalizacionTabla)) || ($newPenalizacionTabla<0)){
        $errorMessage .= "<p class='error'>La penalización por una tabla de más
        debe ser un número mayor o igual a 0.<br></p>";
    }
}
```

Se comienza metiendo la restricción que no permite valores no válidos, seguido de su incorporación al archivo de configuración:

```
$buffer=$buffer."PenalizacionAts=$newPenalizacionAts".PHP_EOL;
```

```
$buffer=$buffer."PenalizacionTabla=$newPenalizacionTabla".PHP_EOL;
```

Con esto, ya se pueden añadir las variables `$penalizacionAts` y `$penalizacionTabla`, además de que aparecerá para editar su valor en la vista del profesor.

Esto también obliga a meter las siguientes líneas de código en el cálculo de puntuación:

```
if (!$this->ids_incorrectos && !$this->ats_incorrectos && $this->numeroTablas==0)
```

Primero, se añaden condiciones para llegar al fragmento de código donde la práctica ha estado perfecta, ya que se tiene también que asegurar que no ha habido ningún atributo incorrecto o tabla extra.

Para el fragmento donde sí ha habido un error habrá que añadir lo siguiente:

```
$penalizacionAts=$penalizacionAts/100;  
$penalizacionTabla=$penalizacionTabla/100;
```

Esto, al igual que con `penalizacionIds`, se hace porque el valor lo mete el profesor como porcentaje, es decir, si el profesor quiere que quiten 10%, el valor será 10, con lo que se debe tratar este valor.

La puntuación se calculará de la siguiente manera:

```
$puntuacion=($puntosACorrecto-($puntos_id*$this->ids_incorrectos)-  
((($puntos_id*$penalizacionIds)*$this->ids_incorrectos) -  
((($puntos_id*$penalizacionAts)*$this->ats_incorrectos) -  
((($puntos_id*$penalizacionTabla)*$this->numeroTablas))*$multidificultad;
```

Esencialmente, quita los puntos de la misma manera que antes, pero con más variables que pueden quitar puntuación.

Ahora solo falta modificar la función de comparar, donde se han de añadir los casos de corrección que se han creado. Se comenzará con cómo tratar las tablas extras. Eso se haría de la siguiente manera:

```
if (count($this->tabla_comparar) > 1) {  
    echo "<p class='pAlum'>Error: Se han detectado múltiples tablas. Solo se  
    permite una tabla.</p>";  
    $this->numeroTablas = count($this->tabla_comparar);  
}
```

Esto permite contar la cantidad de tablas que hay y avisar al alumno cuando se muestre la corrección.

De cara a los atributos, se hará lo siguiente:

```
$ats_no_vacios = [];  
$novacio = false;  
// Validar los atributos del lexema  
foreach ($atributos_alum as $atributo => $valor) {  
    if (in_array($valor, ["0", "-", ""])) {  
        $this->ats_incorrectos++;  
        $novacio = true;  
        $ats_no_vacios[] = $atributo;  
    }  
}  
if($novacio){  
    $atributos_no_vacios = implode(", ", $ats_no_vacios);  
    if($this->ats_incorrectos == 1){  
        echo "<p class='pAlum'>El identificador <strong>$lexema</strong> tiene  
        el siguiente atributo no vacío: $atributos_no_vacios.</p>";  
    }  
}
```

```

    }else{
        echo "<p class=\"pAlum\">El identificador <strong>$lexema</strong> tiene
        los siguientes atributos no vacíos: $atributos_no_vacios.</p>";
    }
}

```

Esto se realizaría en cada iteración donde se encuentre un lexema correcto, ya que no se comprobará si un lexema incorrecto tiene atributos vacíos o no.

Primero se miran todos los atributos de un lexema y se determina si todos son vacíos. Los no vacíos incrementan el contador de ats_incorrectos y se añaden a un array, que después se recorre para informar al alumno de todos los atributos no vacíos que hay dentro de su tabla.

También se añadirá el siguiente fragmento de código al final de la comprobación:

```

if($this->ats_incorrectos){
    if($this->ats_incorrectos == 1){
        echo "<p class=\"pAlum\">Se han encontrado un atributo no vacío definido
        en la TS.</p>";
    }else{
        echo "<p class=\"pAlum\">Se han encontrado " . $this->ats_incorrectos .
        " atributos no vacíos definidos en la TS.</p>";
    }
}

```

Este código permite informar al alumno sobre el número total de atributos no vacíos que hay en su tabla entregada.

Una vez hecha la implementación, la función final de comparación será la siguiente:

```

public function comparar() {
    $ids_detectados = array();
    // Verificar si hay más de una tabla en tabla_comparar
    if (count($this->tabla_comparar) > 1) {
        echo "<p class=\"pAlum\">Error: Se han detectado múltiples tablas. Solo se permite
        una tabla.</p>";
        $this->numeroTablas = count($this->tabla_comparar);
    }
    foreach ($this->tabla_correcta as $nombre_tabla => $contenido_tabla) {
        foreach ($contenido_tabla as $lexema => $atributos) {
            if (in_array($lexema, $ids_detectados))
                continue;
            $encontrado = false;
            $this->num_ids++;
            foreach ($this->tabla_comparar as $nombre_tabla_alum => $contenido_tabla_alum)
            {
                foreach ($contenido_tabla_alum as $lexema_alum => $atributos_alum) {
                    if (strcasecmp($lexema, $lexema_alum) === 0) {
                        $encontrado = true;
                        $ids_detectados[] = $lexema;
                        $ats_no_vacios = [];
                        $novacio = false;
                        // Validar los atributos del lexema
                        foreach ($atributos_alum as $atributo => $valor) {
                            if (!in_array($valor, ["0", "-", ""])) {
                                $this->ats_incorrectos++;
                                $novacio = true;
                                $ats_no_vacios[] = $atributo;
                            }
                        }
                    }
                }
            }
            if($novacio){

```

```

$atributos_no_vacios = implode(", ", $ats_no_vacios);
if($this->ats_incorrectos == 1){
    echo "<p class=\"pAlum\">El identificador
    <strong>$lexema</strong> tiene el siguiente atributo no vacío:
    $atributos_no_vacios.</p>";
}else{
    echo "<p class=\"pAlum\">El identificador
    <strong>$lexema</strong> tiene los siguientes atributos no
    vacíos: $atributos_no_vacios.</p>";
}
}
break 2;
}
}
}
if (!$encontrado) {
    echo "<p class=\"pAlum\">No se ha encontrado el identificador
    <strong>$lexema</strong> en ninguna de las tablas aportadas.</p>";
    $this->ids_incorrectos++;
}
}
}
foreach ($this->tabla_comparar as $nombre_tabla_alum => $contenido_tabla_alum){
    foreach ($contenido_tabla_alum as $lexema_alum => $atributos_alum) {
        if (in_array($lexema_alum, $ids_detectados))
            continue;
        else {
            echo "<p class=\"pAlum\">Se ha detectado el identificador
            <strong>$lexema_alum</strong> pero no corresponde a uno de los
            identificadores de la prueba.</p>";
            $this->ids_incorrectos++;
        }
    }
}
if($this->ats_incorrectos){
    if($this->ats_incorrectos == 1){
        echo "<p class=\"pAlum\">Se han encontrado un atributo no vacío definido en
        la TS.</p>";
    }else{
        echo "<p class=\"pAlum\">Se han encontrado " . $this->ats_incorrectos . "
        atributos no vacíos definidos en la TS.</p>";
    }
}
}
}
}

```

También se ve que, una vez implementado todo el código para la implementación de las variables globales en la configuración, la vista del profesor para dar valor a esas variables es la mostrada en la ilustración 83.

Configuración de los tiempos correspondientes a pruebas.

Tiempo para plantilla fácil | 66 |
 Tiempo para plantilla intermedia | 77 |
 Tiempo para plantilla difícil | 99 |
 Tiempo límite de cálculo | 3333 |
 Tiempo entre pruebas | 0 |

Puntuaciones y penalizaciones otorgadas a los alumnos por conseguir determinadas metas en cada prueba.

Puntos por análisis correcto | 30 |
 Puntos por participar | 2 |
 Penalización por identificador incorrecto | 1 | %
 Penalización por atributo no vacío | 10 | %
 Penalización por Tabla de más | 10 | %

Multiplicadores según la dificultad de las pruebas realizadas por los alumnos y número de intentos que tendrán cada grupo.

Multiplicador dificultad fácil | 1 |
 Multiplicador dificultad intermedia | 1,5 |
 Multiplicador dificultad difícil | 2 |
 Intentos totales | 7 |

Probabilidades para alumnos con nivel de dificultad difícil de que se le asigne una prueba de una determinada dificultad.

Probabilidad difícil | 65 | %
 Probabilidad intermedia | 25 | %
 Probabilidad fácil | 10 | %

Ilustración 83 Página para editar los datos de la práctica

En esta ilustración 83, se pueden ver las entradas para las puntuaciones a restar cuando se realice la corrección para los distintos errores del alumno.

4.5.3 Pruebas

Para comprobar el correcto funcionamiento de estas funcionalidades, se van a tener que hacer diversas pruebas.

Las pruebas que se realizarán serán:

- Comprobar que se detectan bien los atributos no vacíos y vacíos
- Comprobar que se detecta si hay tablas de más
- Comprobar que también se corrija bien en el caso de que se envíe una tabla correcta.

4.5.3.1 Comprobar que se detectan bien los atributos no vacíos y vacíos

Se hará la prueba haciendo la tabla de símbolos para el siguiente fragmento de código dado por el generador de prácticas de tabla de símbolos:

```
let x int = "";let num int ;
continue
;put{num};alert(num,
n1)let uu int;
call hola();
```

Para hacer la prueba, se introduce como respuesta la siguiente tabla de símbolos:

```
CONTENIDOS DE LA TABLA #1:
* LEXEMA : 'x'
ATRIBUTOS :
+ atributo : ''
-----
* LEXEMA : 'num'
ATRIBUTOS :
+ atributo : '0'
-----
```

```

* LEXEMA : 'continue'
ATRIBUTOS :
  + atributo : '-'
-----
* LEXEMA : 'uu'
ATRIBUTOS :
  + atributo : 'int'
-----
* LEXEMA : 'alert'
-----
* LEXEMA : 'n1'
-----
* LEXEMA : 'call'
-----
* LEXEMA : 'hola'

```

Esta tabla contiene 4 atributos. Tres de ellos son considerados vacíos y correctos, mientras que hay un atributo que contiene el valor int, el cual no es correcto. Esta prueba ayudará a detectar si ignora los atributos vacíos y si detecta atributos no vacíos.

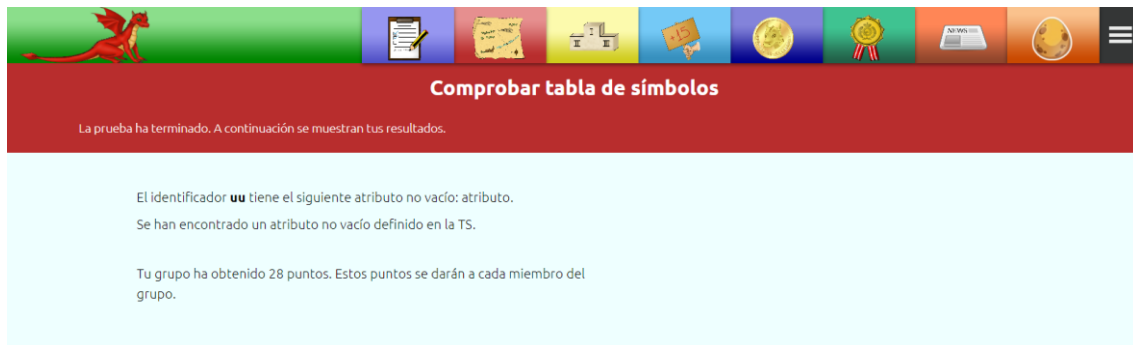


Ilustración 84 Resultados de la prueba

Tal y como se puede ver en la ilustración 84, se ignoran los atributos vacíos y se detecta perfectamente cuando un atributo es no vacío.

4.5.3.2 Comprobar que se detecta si hay tablas de más

En este caso, el generador ha dado el siguiente código:

```

let texto string;
function pideTexto void (void)
{
  put "Introduce una palabra";
  get texto;
}
function print void (string msg)
{
  put msg;
}
pideTexto();
let textoAux string;textoAux = texto;
print
(
  textoAux
)
;

```

Y se ha generado la siguiente tabla de símbolos:

CONTENIDOS DE LA TABLA #1:

* LEXEMA : 'texto'

* LEXEMA : 'pideTexto'

* LEXEMA : 'print'

* LEXEMA : 'msg'

* LEXEMA : 'textoAux'

CONTENIDOS DE LA TABLA #2:

Que se puede observar tiene el inicio de dos tablas distintas.



Ilustración 85 Resultado de la prueba

Como se puede ver en la Ilustración 85, ha detectado que hay más de una tabla y ha dado el aviso.

4.5.3.3 Comprobar que también se corrija bien en el caso de que se envíe una tabla correcta.

El corrector ha generado el siguiente código para crear la tabla de símbolos:

```
let texto string;
function pideTexto void (void)
{
    put "Introduce una palabra";
    get texto;
}
function print void (string msg)
{
    put msg;
}
pideTexto();
let textoAux string;textoAux = texto;
print
(
    textoAux
)
;
```

Se ha creado la siguiente tabla de símbolos como respuesta:

CONTENIDOS DE LA TABLA #1:

* LEXEMA : 'texto'

* LEXEMA : 'pideTexto'

* LEXEMA : 'print'

* LEXEMA : 'msg'

* LEXEMA : 'textoAux'

El resultado se expone a continuación en ilustración 86.

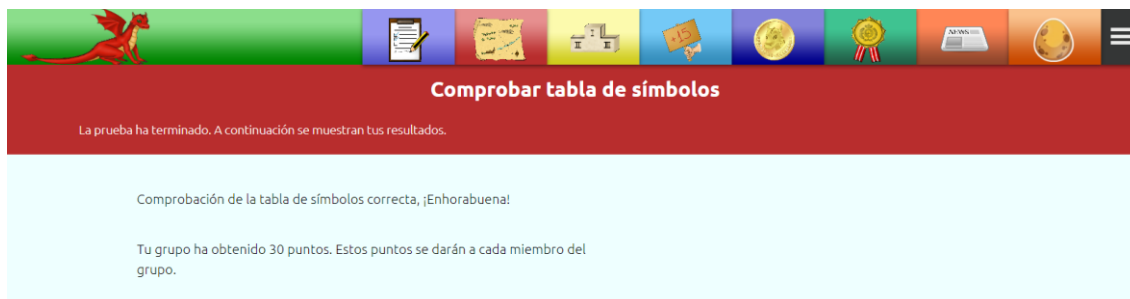


Ilustración 86 Resultado de la prueba

Tal y como se ve en la ilustración 86, la corrección ha sido exitosa.

4.6 Corrección del nivel mínimo al crear una actividad

4.6.1 Diseño

Se comienza analizando la arquitectura de esta página. Se expone de manera visual en la ilustración 87.

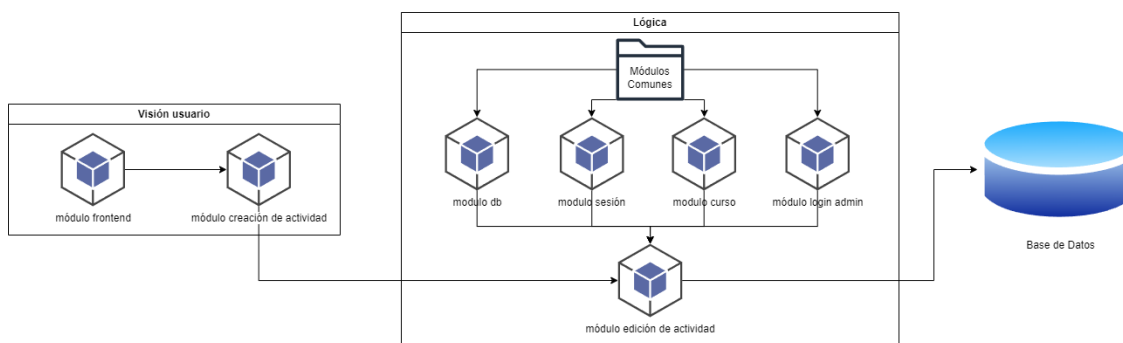


Ilustración 87 Arquitectura de la página de crear actividad

Esta arquitectura comienza con una vista que permite acceso a la página de creación de actividades. Esta permite elegir qué tipo de actividad se quiere crear y la crea con unos valores vacíos. Procede a redirigir directamente a una página de editar actividades que permite meter los valores que se desean a la actividad ya creada.

Esta arquitectura es correcta y cumple su función, el problema ha de residir en la lógica del programa al crear la actividad, específicamente en el módulo de edición de la actividad.

Actualmente el módulo permite la creación de una actividad de nivel mayor o igual a 0, cuando los alumnos siempre tienen un nivel mínimo de 1, tal y como se explicó anteriormente.

Se ha de modificar este valor mínimo para que sea 1 en vez de 0 y así sean consistentes todas las futuras actividades.

4.6.2 Implementación

Añadir nueva actividad Copiar actividades Importar actividades

Puede filtrar actividades que cumplan ciertas características

Título Tipo de Actividad Estado Medalla Puntos Min Puntos Max Dia Inicio Dia Fin Nivel

Seleccione la actividad a gestionar:

Título	Tipo	Estado	Medalla	Puntuación	Fecha Inicio	Fecha Fin	Nivel	
Segundo examen test	Respuesta única	Editando	Sin definir	0 - 0	30-12-2024 17:50	30-12-2024 17:50	0	Revisar Editar Borrar
Examen parcial	Respuesta múltiple	Auto	Sin definir	3 - 22	07-11-2021 23:59	14-11-2021 23:57	1	Revisar Editar Activar Ver
Examen parcial	Respuesta corta	Activa	Sin definir	3 - 20	16-11-2021 17:17	21-11-2021 22:22	0	Revisar Editar Desactivar Auto Ver
Encuesta - 500 puntos	Respuesta corta	Activa	Sin definir	1 - 8	06-01-2022 12:34	27-01-2022 12:34	0	Revisar Editar Desactivar Auto Ver
Música en Clase práctica TDS	Respuesta única	Activa	Sin definir	3 - 22	24-11-2022 17:00	30-11-2022 21:22	4	Revisar Editar Desactivar Auto Ver
Fiesta	Respuesta única	Activa	Sin definir	3 - 11	26-01-2023 17:59	27-01-2023 17:00	1	Revisar Editar Desactivar Auto Ver
Bienvenida	Respuesta única	Activa	Sin definir	0 - 50	06-09-2023 08:58	26-09-2023 23:32	0	Revisar Editar Desactivar Auto Ver
Mascarilla	Respuesta única	Activa	Sin definir	2 - 11	07-09-2023 18:00	26-09-2023 01:23	0	Revisar Editar Desactivar Auto Ver
Página de la asignatura	Respuesta única	Inactiva	Sin definir	5 - 25	08-09-2023 22:22	27-09-2023 13:57	0	Revisar Editar Activar Borrar Ver
Profesores	Respuesta múltiple	Inactiva	Sin definir	5 - 25	11-09-2023 11:55	27-09-2023 19:57	1	Revisar Editar Activar Borrar Ver
Analizadores	Respuesta corta	Inactiva	Sin definir	5 - 40	12-09-2023 13:55	28-09-2023 09:55	0	Revisar Editar Activar Borrar Ver
Token	Respuesta única	Inactiva	Scanneator	10 - 75	14-09-2023 11:45	29-09-2023 23:11	0	Revisar Editar Activar Borrar Ver
Termina la frase	Respuesta corta	Inactiva	Scanneator	5 - 50	14-09-2023 13:13	30-09-2023 23:32	0	Revisar Editar Activar Borrar Ver
Función del Analizador Léxico	Respuesta múltiple	Inactiva	Scanneator	10 - 75	16-09-2023 21:00	01-10-2023 13:13	0	Revisar Editar Activar Borrar Ver

Ilustración 88 Página de añadir actividad

Cuando un profesor quiere crear una actividad, actualmente iría a la página de `manageActivities.php` y haría clic en el botón de añadir nueva actividad mostrada en la ilustración 88.

Añadir una actividad nueva

Desde aquí podrá añadir una actividad.

Deberá elegir de que tipo será.

El tipo 'Test con Selección Única' presenta una pregunta y varias respuestas, pudiendo elegir una única respuesta.

El tipo 'Test con Selección Múltiple' presenta una pregunta y varias respuestas, pudiendo elegir varias respuestas.

El tipo 'Ordenar' pide al alumno que ponga en posición correcta las respuestas según el orden que se pide.

El tipo 'Emparejar' pide al alumno que coloque por parejas las respuestas proporcionadas.

Las preguntas de 'Respuesta corta' requieren que el alumno acierte ciertas palabras claves en la respuesta.

El tipo 'Respuesta larga' permite una respuesta de texto libre que se deberá de corregir de forma manual.

El tipo 'Secuencia' permite al alumno realizar varias actividades de manera continua, ganando más puntos por cada actividad realizada, y ofrecen posibilidad de apostar sus propios puntos.

El tipo 'Examen' presenta al alumno varias actividades, teniendo que realizar todas sin posibilidad de salir hasta terminar. Los puntos se asignarán en función de la nota obtenida (De 0 al 10).

El tipo 'Encuesta' permite crear una actividad sin penalización de puntos del tipo seleccionado.

Las 'Revisiones' asignan la correcciones de actividades de respuesta larga a otros jugadores.

En las preguntas de tipo 'Defectuoso', se debe de encontrar el error en la pregunta y presentar una respuesta correcta a la pregunta.

Las actividades 'Colaborativas' permiten que los jugadores completen preguntas con la participación de otros.

Nueva actividad: Test con Selección Única

Ilustración 89 Página donde se elige la actividad

INFORMACIÓN BÁSICA

Edite la información básica de la pregunta.

Título

Pregunta

Borrar la imagen de la pregunta

Añadir imagen Ningún archivo seleccionado No hay ninguna imagen subida.

Altura de la imagen

Descripción alternativa de la imagen

Nivel

Puntos mínimos

Puntos máximos

Tiempo

Máximo bonus por tiempo

Insignia relacionada

Fecha de inicio Hora de inicio

Fecha de terminación Hora de terminación

Ilustración 90 Página de creación de actividad

Ese botón redirige a la página mostrada en la ilustración 89. Una vez seleccionado un tipo de actividad y pulsado el botón de crear, lleva a la página donde se crea la actividad, añadiendo la información necesaria. Esta página se muestra en la ilustración 90.

Como se mencionó, el problema es que se permite crear actividades de nivel 0, cuando no es necesario y no es consistente con los niveles válidos. El archivo correspondiente a la ilustración 90 es el archivo editActivity.php, ya que la actividad se crea después de darle al botón de crear en la página de la ilustración 8, pero no se guarda hasta que se dé al botón de guardar en la página de la ilustración 90.

```
<p>
  <label for="level">Nivel </label>
  <input type="number" min="0" step="1" name="level" value="<?php echo
  isset($_POST["level"]) && $_POST["level"] != '' ? $_POST["level"] :
  (!empty($activity_info["req_level"]) ? $activity_info["req_level"] : 0); ?>"
  id="level" <?php if(isset($form_errors["level"])) echo "style='border: 2px solid
  red;";?>>
</p>
```

Este código corresponde a la entrada de nivel que se observa en la ilustración 89, que es el principal origen del problema. Por suerte este problema tiene una solución muy sencilla: cambiar el número mínimo y el número por defecto que aparece dentro de este campo a 1:

```
<p>
  <label for="level">Nivel </label>
  <input type="number" min="1" step="1" name="level" value="<?php echo
  isset($_POST["level"]) && $_POST["level"] != '' ? $_POST["level"] :
  (!empty($activity_info["req_level"]) ? $activity_info["req_level"] : 1); ?>"
  id="level" <?php if(isset($form_errors["level"])) echo "style='border: 2px solid
  red;";?>>
</p>
```

Una vez implementado el código, se puede notar visualmente si tiene algún efecto. Una simple recarga de la página muestra lo siguiente en la ilustración 91.

INFORMACIÓN BÁSICA

Edite la información básica de la pregunta.

Título

Pregunta

Borrar la imagen de la pregunta

Añadir imagen Ningún archivo seleccionado No hay ninguna imagen subida.

Altura de la imagen

Descripción alternativa de la imagen

Nivel

Puntos mínimos

Puntos máximos

Tiempo

Máximo bonus por tiempo

Insignia relacionada

Fecha de inicio Hora de inicio

Fecha de terminación Hora de terminación

Ilustración 91 página actual de editActivity.php

En esta ilustración 91, se puede ver que el nivel por defecto sube a 1.

4.6.3 Pruebas

Para las pruebas de esta solución, primero se probará si permite crear una actividad con nivel menor que 1. Después se hará una segunda prueba donde se creará una prueba de nivel 1 y se comprobará que se ha creado correctamente.

4.6.3.1 Creación de actividad de nivel 0

INFORMACIÓN BÁSICA

Edite la información básica de la pregunta.

Título

Pregunta

Borrar la imagen de la pregunta

Añadir imagen Ningún archivo seleccionado No hay ninguna imagen subida.

Altura de la imagen

Descripción alternativa de la imagen

Nivel

! El valor debe ser superior o igual a 1

Puntos máximos

Tiempo

Máximo bonus por tiempo

Insignia relacionada

Fecha de inicio Hora de inicio

Fecha de terminación Hora de terminación

Ilustración 92 Intento de creación de actividad con nivel 0

Como se puede ver en la ilustración 92, salta un aviso cuando se intenta crear una actividad de nivel 0.

4.6.3.2 Creación de actividad correcta

INFORMACIÓN BÁSICA

Edite la información básica de la pregunta.

Título

Pregunta

Borrar la imagen de la pregunta

Añadir imagen Ningún archivo seleccionado No hay ninguna imagen subida.

Altura de la imagen

Descripción alternativa de la imagen

Nivel

Puntos mínimos

Puntos máximos

Tiempo

Máximo bonus por tiempo

Insignia relacionada

Fecha de inicio Hora de inicio

Fecha de terminación Hora de terminación

Ilustración 93 Datos iniciales de actividad a crear

Actividad	Respuesta	Estado	Definición	Puntos	Inicio	Fin	Revisar	Editar	Activar	Borrar	Ver	
Analizadores	Respuesta corta	Inactiva	Sin definir	5 - 40	12-09-2023 13:55	28-09-2023 09:55	0	Revisar	Editar	Activar	Borrar	Ver
Token	Respuesta única	Inactiva	Scanneator	10 - 75	14-09-2023 11:45	29-09-2023 23:11	0	Revisar	Editar	Activar	Borrar	Ver
Termina la frase	Respuesta corta	Inactiva	Scanneator	5 - 50	14-09-2023 13:13	30-09-2023 23:32	0	Revisar	Editar	Activar	Borrar	Ver
Función del Analizador Léxico	Respuesta múltiple	Inactiva	Scanneator	10 - 75	16-09-2023 21:00	01-10-2023 13:13	0	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 1	Respuesta única	Inactiva	Scanneator	5 - 44	19-09-2023 11:11	03-10-2023 23:34	0	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 2	Respuesta única	Inactiva	Scanneator	5 - 44	20-09-2023 13:00	04-10-2023 22:00	1	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 3	Respuesta única	Inactiva	Scanneator	5 - 44	21-09-2023 00:00	05-10-2023 22:01	2	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 4	Respuesta única	Inactiva	Scanneator	5 - 44	22-09-2023 15:55	06-10-2023 15:00	3	Revisar	Editar	Activar	Borrar	Ver
Tokens del programa	Respuesta múltiple	Inactiva	Scanneator	5 - 44	22-09-2023 19:00	07-10-2023 11:22	2	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 6	Respuesta única	Inactiva	Scanneator	5 - 44	09-10-2023 15:00	19-10-2023 10:00	4	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 7	Respuesta única	Inactiva	Scanneator	5 - 44	19-11-2023 21:00	29-11-2023 13:55	5	Revisar	Editar	Activar	Borrar	Ver
Tokens varios - 2	Respuesta múltiple	Inactiva	Scanneator	10 - 75	09-12-2023 12:05	19-12-2023 22:00	2	Revisar	Editar	Activar	Borrar	Ver
Tokens empresa - 5	Respuesta única	Inactiva	Scanneator	5 - 44	17-12-2023 10:00	27-12-2023 10:00	3	Revisar	Editar	Activar	Borrar	Ver
Tokens varios	Respuesta múltiple	Inactiva	Scanneator	10 - 70	19-12-2023 23:05	29-12-2023 05:05	2	Revisar	Editar	Activar	Borrar	Ver
Acciones semánticas del Léxico	Respuesta múltiple	Inactiva	Scanneator	5 - 60	28-12-2023 16:00	08-01-2024 08:00	2	Revisar	Editar	Activar	Borrar	Ver
Comida	Encuesta larga	Activa	Sin definir	0 - 66	09-12-2024 00:00	09-12-2025 00:00	1	Revisar	Editar	Desactivar	Auto	Ver
TFG	Respuesta única	Auto	Sin definir	0 - 30	30-12-2024 00:00	30-12-2025 00:00	1	Revisar	Editar	Activar	Ver	

Cambiar la categoría de todas las actividades:

Ilustración 94 Listado con la nueva actividad

Como se puede ver en las ilustraciones 93 y 94, parece que la actividad se ha creado correctamente.

Para terminar de comprobarlo, se seleccionará el botón ver para asegurar que funciona correctamente.

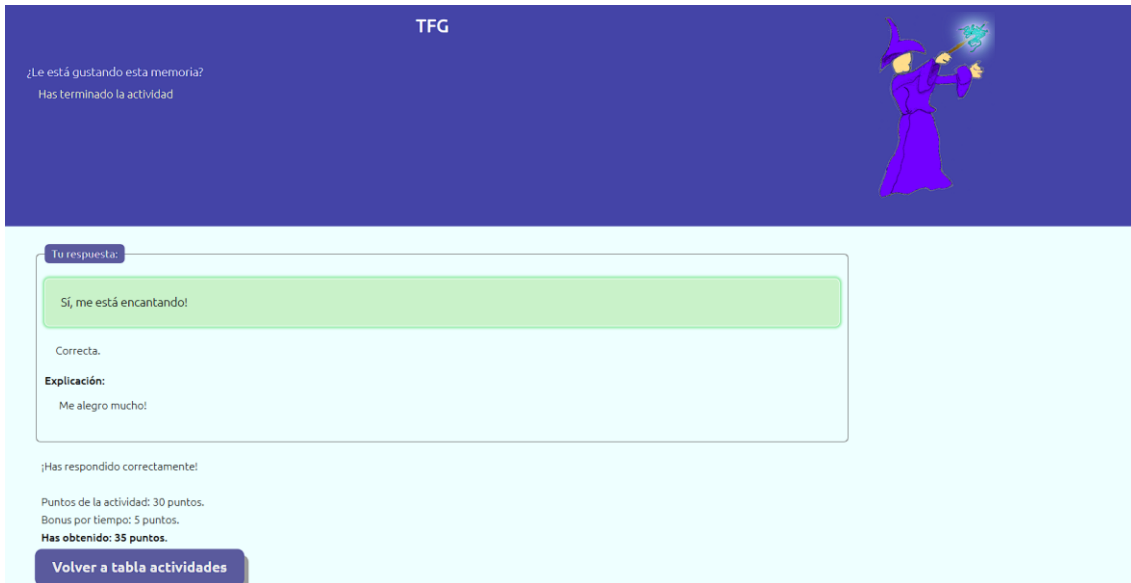


Ilustración 95 Actividad finalizada

Se puede ver en la ilustración 95 que la actividad se ha creado correctamente. Finalmente, se activará y se verá si sale correctamente en la vista del alumno.



Ilustración 96 Vista de actividades del alumno

Se puede ver que la actividad está disponible para el alumno en la ilustración 96.

5 Conclusiones

La realización de este Trabajo de Fin de Grado (TFG) ha supuesto un reto tanto técnico como académico que ha aportado conocimientos significativos en el desarrollo y mantenimiento de aplicaciones web complejas. A través de este proyecto, se ha logrado un profundo entendimiento de tecnologías web clave, como HTML5, CSS3, PHP, y la gestión de bases de datos, así como de técnicas de corrección automática y gamificación aplicadas al ámbito educativo.

Uno de los mayores logros ha sido la optimización de la plataforma DRACO, abordando problemas concretos que limitaban su funcionamiento y añadiendo funcionalidades clave que mejoran tanto la experiencia de los usuarios como la eficiencia en la gestión de actividades. Entre los avances más relevantes, destacan la implementación de la importación múltiple de actividades, la corrección de errores en la copia y el borrado de lenguajes, y la ampliación del sistema de corrección de la tabla de símbolos. Estas mejoras no solo refuerzan la utilidad de la plataforma, sino que también la hacen más robusta y fácil de usar.

Desde un punto de vista personal, este TFG ha supuesto una experiencia enriquecedora que ha reforzado las habilidades en el diseño de interfaces intuitivas y en la implementación de sistemas *backend* eficientes. También ha potenciado las competencias en resolución de problemas, documentación técnica y en la aplicación de metodologías de prueba para garantizar la calidad del software. Además, se ha comprendido la importancia de la interacción constante con los usuarios finales y la necesidad de adaptar las soluciones a sus necesidades reales.

Finalmente, este proyecto subraya la relevancia de las herramientas de autoevaluación y gamificación en la educación moderna, destacando su impacto positivo en el compromiso y motivación de los estudiantes. El trabajo realizado no solo optimiza una plataforma existente, sino que también contribuye al avance de las metodologías educativas en la formación de futuros ingenieros, consolidando el uso de la tecnología como un pilar esencial en el proceso de enseñanza-aprendizaje.

6 Futuras líneas de trabajo

Algunas futuras líneas de trabajo dentro del desarrollo de DRACO pueden incluir:

Premio por Buena Puntuación en las Prácticas:

Actualmente, los alumnos deben esperar un tiempo determinado entre cada entrega de prácticas para poder realizar una nueva corrección o entrega. Esta limitación pierde sentido cuando la solución entregada es casi perfecta y solo requiere correcciones menores. Se propone implementar una función que reduzca el tiempo de espera hasta un máximo del 50 % del tiempo general, dependiendo del porcentaje de la puntuación máxima obtenida por el alumno en la evaluación. Dentro de las distintas bases de datos, tal y como se vió anteriormente, hay algunas bases de datos que son para las prácticas. En cada una de ellas hay una tabla "grupos" que contiene una columna con tiempo *custom* de alumnos, que puede reducir el tiempo de espera de los alumnos. Un ejemplo de cuando se usa esto es cuando un alumno usa monedas de DRACO conseguidas durante la asignatura para una reducción del tiempo de espera a un 50%. La implementación de esta mejora propuesta tendría un funcionamiento parecido.

Opción de Borrado de Seleccionados:

Se plantea la incorporación de una función que permita eliminar todas las actividades seleccionadas en la página de "Importar Actividades". Actualmente, el sistema solo permite importar las actividades seleccionadas mediante *checkboxes*, pero no ofrece la opción de eliminarlas de manera masiva.

Corrección en la Creación de Actividades Vacías:

En la página de "Editar Actividades", existe un caso donde, al crear una nueva actividad sin introducir datos y salir de la página, se genera una actividad vacía que aparece en la lista de actividades. Se sugiere modificar este comportamiento añadiendo un aviso antes de cerrar la pestaña o impidiendo la creación de la actividad en caso de que no se hayan rellenado los campos obligatorios.

7 Impacto

Este Trabajo de Fin de Grado (TFG) tiene un impacto significativo en múltiples áreas relacionadas con los Objetivos de Desarrollo Sostenible (ODS) al contribuir al avance de la educación, la reducción de desigualdades, la innovación tecnológica y la sostenibilidad (Naciones Unidas, 2015).


1. **ODS 4: Educación de Calidad:** El TFG contribuye de manera directa a mejorar la calidad educativa al optimizar la plataforma DRACO, una herramienta que facilita el aprendizaje activo mediante autoevaluaciones, gamificación y la gestión eficiente de tareas. La implementación de nuevas funcionalidades, como el corrector mejorado de la tabla de símbolos y la importación múltiple de actividades, permite a estudiantes y profesores interactuar de manera más eficaz con el contenido del curso, promoviendo un aprendizaje inclusivo y participativo.
2. **ODS 9: Industria, Innovación e Infraestructura:** La mejora de DRACO se basa en la aplicación de tecnologías web modernas (HTML5, CSS3, PHP) y buenas prácticas de desarrollo. Esto fomenta la construcción de infraestructuras digitales sólidas y sostenibles en el ámbito educativo, proporcionando una base tecnológica adaptable y eficiente que puede servir como referencia para otros proyectos educativos.
3. **ODS 10: Reducción de las Desigualdades:** La plataforma permite una evaluación más justa y personalizada al proporcionar retroalimentación automatizada y accesible desde cualquier dispositivo con conexión a internet. Esto contribuye a reducir las desigualdades en el acceso a recursos educativos, especialmente para estudiantes que puedan enfrentarse a limitaciones geográficas o económicas.
4. **ODS 16: Paz, Justicia e Instituciones Sólidas:** La transparencia y la trazabilidad de los resultados académicos que permite DRACO fortalecen la equidad y la justicia en la evaluación de los estudiantes. Al evitar errores manuales y automatizar procesos, se refuerzan los principios de imparcialidad y se disminuye la posibilidad de sesgos en la calificación.
5. **ODS 17: Alianzas para Lograr los Objetivos:** Este proyecto se enmarca en un esfuerzo colaborativo entre estudiantes, profesores y la comunidad académica de la Universidad Politécnica de Madrid. El desarrollo de DRACO fomenta la cooperación entre diferentes actores académicos y puede promover alianzas con otras instituciones interesadas en adoptar soluciones similares, potenciando un ecosistema de aprendizaje compartido.

Este TFG demuestra cómo el uso eficiente de tecnologías de código abierto puede contribuir al cumplimiento de múltiples ODS, resaltando el papel fundamental de la innovación en la transformación de la educación superior y el acceso al conocimiento. Con las mejoras propuestas, DRACO se posiciona como una herramienta clave para modernizar la enseñanza y reforzar el aprendizaje autónomo y colaborativo.

8 Bibliografía

- Naciones Unidas. (2015). *Objetivos de Desarrollo Sostenible*. Obtenido de Página web de las Naciones Unidas:
<https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- PHP Group. (2004). *Historia de PHP*. Obtenido de PHP:
<https://www.php.net/manual/es/history.php.php>
- PHP Group. (2024). *Documentation*. Obtenido de PHP:
<https://www.php.net/docs.php>
- PHP Group. (2024). *PHP Releases*. Obtenido de PHP:
<https://www.php.net/releases/>
- W3C. (2023). *CSS Snapshot 2023*. Obtenido de World Wide Web Consortium (W3C): <https://www.w3.org/TR/CSS/>
- W3C. (2024). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. Obtenido de World Wide Web Consortium (W3C):
<https://www.w3.org/TR/html5/>
- W3C. (2024). *Making the Web Accessible*. Obtenido de World Wide Web Consortium (W3C): <https://www.w3.org/TR/html5-aapi/>

Este documento esta firmado por

	Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Mon Jan 13 20:30:00 CET 2025
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)