

Paradigmas y Lenguajes de Programación III

CARRERA: Ingeniería en Sistemas de Información

MATERIA: Paradigmas y Lenguajes de Programación III **COMISIÓN:**
“U” (única)

PROFESOR: Mgter. Ing. Encina Agustín.

ESTUDIANTE: Gamarra Kiara Barbarella

FECHA: 13-11-2025

Informe de implementación de SweetAlert2 en el proyecto Flashcards

Objetivo:

Mejorar la experiencia de usuario al confirmar acciones críticas, reemplazando los diálogos nativos alert() y confirm() del navegador por ventanas modales modernas y visualmente atractivas.

Link al repo con plugin: https://github.com/melodramaticbxtch/Proyecto_Flashcards

Paso a paso de la implementación

1. Identificación de puntos críticos en la interfaz

- Se detectaron dos acciones sensibles que requerían confirmación:
 - a) Eliminación de una tarjeta individual.
 - b) Eliminación completa de una colección junto con todas sus tarjetas.

2. Selección del plugin adecuado

- Se eligió **SweetAlert2**, un plugin de JavaScript moderno, liviano y ampliamente utilizado, que permite crear alertas y confirmaciones personalizadas con íconos, botones y estilos visuales consistentes.

3. Incorporación del plugin

Se incluyó la librería vía CDN al final del documento HTML antes de cerrar la etiqueta </body>:

```
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
```

-
- Esto garantiza que el script cargue después del DOM, evitando errores de referencia a elementos no existentes.

4. Modificación de la estructura HTML de los botones

Los botones que antes utilizaban onclick="return confirm(...)" se modificaron para utilizar atributos data- que almacenan información relevante (ID de tarjeta o colección):

```
<a class="delete-btn eliminar-tarjeta" data-id="..." data-coleccion="...">x</a>
```

```
<a class="btn-eliminar eliminar-coleccion" data-id="...">Eliminar colección completa</a>
```

-
- Esta estrategia permite que los scripts accedan a los datos necesarios para redirigir correctamente al eliminar.

5. Implementación del JavaScript de confirmación

- Se utilizó event delegation para asegurar que los eventos funcionen incluso en elementos generados dinámicamente.
- Para cada acción, se creó un listener que invoca SweetAlert2 mostrando:
 - **Título y mensaje claro** sobre la acción.
 - **Icono adecuado** (warning para tarjeta, error para colección completa).
 - **Botones de confirmación y cancelación.**

Si el usuario confirma, se realiza la redirección a la URL de eliminación correspondiente. Ejemplo para tarjeta:

```
Swal.fire({
  title: "¿Eliminar tarjeta?",
  text: "Esta acción no se puede deshacer",
  icon: "warning",
  showCancelButton: true,
  confirmButtonText: "Sí, eliminar",
  cancelButtonText: "Cancelar"
}).then((result) => {
  if (result.isConfirmed) {
    window.location.href = "eliminar_tarjeta.php?id=" + id_tarjeta + "&coleccion=" + id_coleccion;
  }
});
```

6. Pruebas y verificación

- Se verificó que los botones respondieron correctamente en diferentes navegadores.
- Se comprobó que los datos (id_tarjeta y id_coleccion) se transmitieron correctamente al script de eliminación en PHP.
- Se confirmó que la interfaz resultante era visualmente consistente, moderna y que mejoraba la experiencia de usuario respecto a los diálogos nativos.

Resultados obtenidos

- La interfaz ahora muestra ventanas modales atractivas y claras, mejorando la usabilidad y evitando errores accidentales.
- La implementación es liviana y compatible con el resto del sistema, sin modificar la lógica PHP existente.
- Se logró una mayor coherencia visual con la estética general del proyecto de flashcards.

Conclusión:

La implementación de SweetAlert2 en el proyecto de flashcards representó un avance significativo tanto en la experiencia de usuario como en la coherencia visual de la aplicación. Al reemplazar los diálogos nativos de confirmación (`alert()` y `confirm()`), se logró un sistema de interacción más moderno, intuitivo y seguro, que comunica de manera clara las consecuencias de acciones críticas, como la eliminación de tarjetas individuales o de colecciones completas.

Además, el uso de atributos `data-` para almacenar los identificadores de las tarjetas y colecciones permitió una comunicación precisa entre el front-end y el back-end, garantizando que la redirección hacia los scripts de eliminación en PHP se realice de manera correcta y sin errores. La adopción de `event delegation` en JavaScript asegura que los listeners se mantengan funcionales incluso en elementos generados dinámicamente, lo cual es fundamental en una aplicación donde las tarjetas se crean de manera dinámica y pueden variar en cantidad.

Desde un punto de vista estético y de usabilidad, SweetAlert2 aporta un diseño consistente con la identidad visual del proyecto, ofreciendo iconografía clara, botones con distinción entre acciones de confirmación y cancelación, y mensajes explicativos que ayudan al usuario a tomar decisiones informadas. Esto no solo reduce la probabilidad de errores accidentales, sino que también aumenta la satisfacción y confianza del usuario al interactuar con la plataforma.