

EXAMEN RECUPERACIÓN (UD6-UD7)

Los RA's que evaluaremos son:

RA6 - RA76

Parte III - Aventura de Super Mario

En esta parte del examen trabajaremos con **AJAX** y **jQuery** para gestionar el acceso a una base de datos donde se almacenarán personajes y aventuras. Se deberá interactuar con el DOM exclusivamente mediante **jQuery**, y se exigirá el uso de **Axios** y **jQuery AJAX** en determinadas peticiones al servidor.

Ejercicio 1: Gestión del DOM con jQuery (3,5 puntos)

El formulario ya está definido en el archivo index.html, y los alumnos deberán trabajar en index.js para gestionar la lógica con jQuery.

- El formulario incluye:
- Un campo de texto para el **nombre del mundo**.
- Un primer **select** para elegir entre **héroe o enemigo**.
- Un segundo **select** que muestra los personajes correspondientes según la selección anterior.
- Un botón para **añadir personajes** al equipo de la aventura.
- Un div donde se mostrarán las tarjetas con los **héroes y el enemigo seleccionados**.
- Un botón **"Iniciar Aventura"** para guardar la aventura en la base de datos.
- Una tabla donde se listarán las aventuras guardadas.

Los alumnos deberán:

- ♦ Usar **jQuery** para manejar la interacción con el DOM.
- ♦ Cuando se seleccione "Héroe" o "Enemigo", el segundo <select> debe actualizarse dinámicamente.
- ♦ Los personajes seleccionados se mostrarán con su imagen en el div de **personajes seleccionados**.

Ejercicio 2: Peticiones al Servidor y Validaciones (4,5 puntos)

Se trabajará con AJAX para gestionar los datos en el servidor.

Cada operación debe realizarse con **Axios** o **jQuery AJAX**, dependiendo de la acción.

Reglas para las peticiones al servidor:

Carga de personajes al seleccionar tipo:

- Se hará con **jQuery AJAX**.
- Cuando el usuario seleccione "Héroe" o "Enemigo", se debe hacer una petición GET a php/cargar_personajes.php y cargar la lista en el select de personajes.

Carga de aventuras guardadas:

- **Obligatoriamente con Axios**.
- Al cargar la página, se debe hacer una petición GET a php/cargar_aventuras.php con **Axios** para listar todas las aventuras en la tabla.

Guardado de la aventura en la base de datos:

- Cuando el usuario haga clic en **"Iniciar Aventura"**, se validarán los datos y se enviará una petición POST a `php/guardar_aventura.php`.

Eliminación de aventuras:

- Cada aventura tendrá un botón **✗** para eliminarla.
- Al hacer clic en **"Eliminar"**, se enviará una petición POST a `php/eliminar_aventura.php`, y la tabla se actualizará automáticamente.

Ejercicio 3: Validaciones y Mensajes de Estado (2 puntos)

Se validará que los datos sean correctos antes de enviarlos al servidor.

- ✓ **El nombre del mundo** debe tener al menos **5 caracteres**.
- ✓ **Deben seleccionarse 2 héroes y 1 enemigo**.
- ✓ Si los datos son incorrectos, se mostrará un SweetAlert con la información.
- ✓ Si los datos son correctos, se mostrará un SweetAlert con la información.
- ✓ Al eliminar una fila de la tabla se mostrará un SweetAlert con la información.

Entrega

Los alumnos deben entregar los siguientes archivos:

- ✓ `index.js` → Implementación de la lógica con AJAX y jQuery.

Archivos que se facilitan:

- ✓ `php/cargar_personajes.php` → Devuelve los personajes según el tipo seleccionado.
- ✓ `php/cargar_aventuras.php` → Devuelve todas las aventuras guardadas.
- ✓ `php/guardar_aventura.php` → Guarda una aventura en la base de datos.
- ✓ `php/eliminar_aventura.php` → Elimina una aventura de la base de datos.
- ✓ El archivo `index.html` y `styles.css`, directorio `img`.