



| | | |
|---|---|--|
|  | <p align="center">I.E.S. HERMANOS MACHADO 2º CFGS DAW Desarrollo Web en Entorno Cliente Unidad 7: Ejercicios</p> | <p align="center">DEPARTAMENTO DE INFORMÁTICA</p> |
|---|---|--|

En la carpeta compartida de ejemplos de esta unidad tenéis disponible los documentos HTML de todos los ejercicios del boletín.

1. A partir del documento HTML proporcionado para este ejercicio, implementar las funciones javascript necesarias para iniciar una promesa y atenderla si se pulsa el botón “Procesar Promesa” antes de 2 segundos. En el caso de no ser pulsado en ese plazo, la promesa no será atendida y se procesará un error. En el párrafo de salida se debe mostrar el mensaje que devuelva el objeto Promise, debiendo coger éste lo que el usuario haya indicado en el formulario de entrada para reflejar los mensajes de éxito y de error.
2. A partir del documento HTML proporcionado para este ejercicio, implementar las funciones javascript necesarias para iniciar una promesa y atenderla para que lea el texto incluido en el input y lo añada al párrafo de salida pasados dos segundos. Para este ejercicio no será necesario controlar el posible error del objeto Promise y se debe usar `async/await`.
3. A partir del documento HTML proporcionado para este ejercicio, implementar las funciones javascript necesarias para añadir a la capa de salida el contenido del fichero de texto especificado en el formulario. El contenido debe añadirse sin producir ninguna recarga de página, para ello use `fetch API`.
4. A partir del documento HTML proporcionado para este ejercicio, implementar las funciones javascript necesarias para atacar la API Rest facilitada en el input del formulario y mostrar los objetos leídos por la consola.
5. A partir del documento HTML proporcionado para este ejercicio, implementar las funciones javascript necesarias para atacar la API Rest siguiente: <https://picsum.photos/list> . Ésta devuelve un objeto JSON con una lista de datos de las fotos que almacena. Construir una lista donde cada ítem tendrá un enlace a la URL de cada foto junto al nombre del autor de la misma.
6. Arrancar un servidor Apache con MySQL, por ejemplo XAMPP. Copiar al servidor los ficheros facilitados en la carpeta del ejercicio y ejecutar el script de creación de la base de datos que se facilita. A continuación, programar una petición AJAX de escritura en el servidor para grabar los datos recogidos en el formulario en la base de datos, pero sin recargar la página. Para atacar la base de datos ya se facilita un script PHP llamado *grabar.php*. Si la escritura tiene éxito, el servidor debe devolver un JSON con los datos añadidos, que se mostrarán automáticamente en la capa “respuesta”. Si no tiene éxito deberá mostrar el mensaje de error.
7. Crear un proyecto en Firebase con la cuenta corporativa del centro que servirá para practicar los ejercicios siguientes. En el proyecto crear una Realtime Database e importar el fichero *alumnos.json* facilitado con el material de este ejercicio. Una vez generados los registros en la base de datos, modificar el proyecto facilitado para que al pulsar el botón *Recuperar Datos* se realice una llamada AJAX a la API Rest recién creada para descargar la lista de alumnos. Esa lista de alumnos deberá mostrarse en la capa de salida en formato JSON.

| | | |
|---|---|--|
|  | <p align="center">I.E.S. HERMANOS MACHADO 2º CFGS DAW Desarrollo Web en Entorno Cliente Unidad 7: Ejercicios</p> | <p align="center">DEPARTAMENTO DE INFORMÁTICA</p> |
|---|---|--|

8. A partir de la API Rest configurada en el ejercicio anterior y partiendo de la plantilla del proyecto facilitada con este ejercicio, implementar el código necesario para que se recuperen los datos de los alumnos almacenados en la base de datos mediante una llamada AJAX. Estos datos se deberán mostrar en una tabla cuyas columnas serán id, apellidos, nombre y edad. Además, se deberá configurar otra llamada AJAX para que se inserten nuevos alumnos a la base de datos usando el formulario facilitado.
9. A partir de la API Rest usada en los ejercicios anteriores y partiendo de la plantilla del proyecto facilitada con este ejercicio, implementar el código necesario para que se recuperen los datos de los alumnos almacenados en la base de datos mediante una llamada AJAX. Estos datos se deberán mostrar en una tabla cuyas columnas serán id, apellidos, nombre y edad. Además, se deberá configurar otra llamada AJAX para que se actualice el campo *apellidos* en un registro ya insertado en la base de datos. El valor nuevo del campo será el que se recoja del formulario facilitado. Para seleccionar el elemento que se actualiza usaremos el ID interno que Firebase añade a cada registro de alumno.
10. A partir de la API Rest usada en los ejercicios anteriores y partiendo de la plantilla del proyecto facilitada con este ejercicio, implementar el código necesario para que se recuperen los datos de los alumnos almacenados en la base de datos mediante una llamada AJAX. Estos datos se deberán mostrar en una tabla cuyas columnas serán id, apellidos, nombre y edad. Además, se deberá configurar otra llamada AJAX para que elimine un registro existente en la base de datos. Para seleccionar el elemento que se elimina usaremos el ID interno que Firebase añade a cada registro de alumno.
11. Arrancar un servidor Apache con MySQL, por ejemplo XAMPP. Copiar al servidor los ficheros facilitados en la carpeta del ejercicio y ejecutar el script de creación de la base de datos que se facilita. A continuación, programar una petición AJAX que permita hacer una lista de los registros de la tabla *alumno* cuya edad sea mayor o igual a la que se solicita por el formulario. Para atacar la base de datos ya se facilita un script PHP llamado *seleccionar.php*.