

Trabajo Práctico Especial 2021 - Parte 3

La tercer entrega de la materia consiste en:

1. El sitio web actualizado con las nuevas tecnologías y técnicas vistas.
2. Screenshot del sitio web en el raíz del curso (Fuera de la carpeta del grupo, ponerle número de grupo al archivo).

**LEER CUIDADOSAMENTE TODA LA CONSIGNA
REVISAR CADA PARTE ANTES DE ENTREGAR**

La presentación es de acuerdo a los grupos definidos en la planilla correspondiente. Se considera fundamental la aplicación de buenas prácticas, y la elección apropiada de cada tecnología para cada punto a resolver.

La exposición/defensa se realizará en una computadora dada por la cátedra, por lo que se recomienda probar la página web en diferentes monitores y sistemas operativos.

En el caso de trabajos grupales, la defensa es de cada miembro por separado. Todos los integrantes deben poder responder a cualquier pregunta de cualquier parte del trabajo.

Entrega

Se entrega por medio de la tarea de Moodle creada para tal fin. Cada grupo deberá subir un trabajo, no importa que los compañeros figuren con tarea sin entregar, eso se controlará internamente. Se debe subir a la tarea la captura de pantalla y un archivo comprimido con el sitio web.

La fecha de entrega es:

- **Tandil:**
Entrega: 01/07
Defensa: 02/07
- **Tres Arroyos:**
Entrega: 30/06
Defensa: 01/07
- **Olavarria:**
Entrega: 30/06
Defensa: 01/07 - 03/07

La defensa será en la clase siguiente. Por problemas laborales avisar y se coordinará para la siguiente clase que asistan. Si no avisan previamente se considera que no entregaron/defendieron.

Consigna

Para la tercera entrega, se debe continuar el trabajo de la segunda etapa. El objetivo es agregar nueva funcionalidad detallada a continuación.

Tabla dinámica por API REST

La tabla dinámica que existe en sus páginas tiene que consultar la información disponible mediante un servicio web (API Rest).

Agreguen en su página el código necesario para crear la tabla y llenarla con la información que obtienen del servicio. Cada “fila de la tabla” estará entonces asociado a un “item” o “elemento” del servicio web consultado. A su vez, dicha tabla debe mantenerse actualizada cuando el usuario agregue/edite/borre información.

El **servicio REST** deben generarlo utilizando <https://mockapi.io/>, o alguno similar, creando la API con la colección de datos que corresponden a cada trabajo. En la unidad de 3 de moodle, está disponible el video donde se explica como crear un servicio en **mockapi**.

Ejemplo: si la tabla almacena “cervezas” el endpoint será similar a
`https://5f8sd3342342ca.mockapi.io/api/v1/cervezas`

Requerimientos Funcionales

- La tabla se debe cargar automáticamente al mostrarse en la página: Al entrar por el nav a la página que tiene la tabla, automáticamente debe mostrarse la tabla cargada sin que el usuario tenga que hacer ningún click adicional, con los mismos datos que inicialmente existen en el servicio rest.
- El formulario que agrega información a la tabla debe mantener sincronizado el servicio. Estos datos se deben agregar usando la API Rest.
- **(OPCIONAL +1)** Debe haber un botón que permita crear varios ítems automáticamente (al menos 3 ítems), esos datos deberán agregarse en el servicio y verse en la tabla.
- Permitir eliminar filas de la tabla de a una. Cada fila tiene que tener una forma de indicarle “Borrar” que elimine la fila y el elemento correspondiente del servicio.
- Permitir editar filas de la tabla individualmente de alguna forma. Cada fila tiene que tener una forma de indicarle “Editar” que permita editar los valores de esa fila y los actualice en el servicio.
- Agregar filtros de búsqueda en la tabla desde JS (local). Al filtrar los datos, se mostrarán solo los datos que cumplan ese criterio, los demás datos no se ven, pero no se borran

realmente, se pueden ver al modificar el filtro de búsqueda. Sugerimos dos formas de hacer el filtro (otras opciones debatir con los docentes):

- filtros por combo (tag select) para las columnas con opciones fijas
- filtrar con un input por al menos una columna.
- Limitar tamaño de imágenes y peso del sitio. La carpeta completa del sitio no puede superar los 5 Mb.

OPCIONALES

- **SPA / Partial Render (+2)** Usar AJAX para la navegación de la página (técnica de partial render). La página no debe refrescarse completamente, cuando hago click en un link de la navigation bar se refresca sólo la porción de la información que cambia.
- **Paginación (+2)**: se debe poder traer la información del servicio rest de forma paginada desde el servidor. Leer documentacion <https://www.mockapi.io/docs>
Recomendación: solo botones “anterior” y “siguiente”.
- **(OPCIONAL +1)** Debe haber un botón que permita crear varios ítems automáticamente (al menos 3 ítems), esos datos deberán agregarse en el servicio y verse en la tabla.

Aclaración partial render de navegación (solo aplica si hace OPCIONAL SPA / PARTIAL RENDER):

- La página principal debe tener el contenido en blanco, pero al abrirla el usuario se debe mostrar el contenido completo (se descarga con AJAX automáticamente)

Aclaración Tabla REST:

- No se permite el uso de **JQuery** para REST

Los requerimientos deben ser los definidos en la planilla de trabajos y estar en el lugar indicado. La estructura HTML, CSS y JS si lo hubiera debe ser de acuerdo a las diferentes pautas explicadas. En caso de errores graves se puede desaprobado el trabajo (lo que implica la pérdida de cursada).

La página no requiere conservar el mismo diseño que su versión de la primer entrega.

Consideraciones

- El trabajo tiene que estar instalado y andando en un servidor web.

Recomendaciones

- **Aquellos trabajos que tienen correcciones a realizar o tienen que reentregar el TPE2, es conveniente que implementen las correcciones primero y guarden una copia de dicha versión. Defiendan esa versión en la comisión práctica. Tengan especial atención si están trabajando sobre el proyecto de la segunda entrega para no acarrear ni errores ni malas prácticas**

Adicional Final libre:

- A DEFINIR.

Criterios de evaluación:

Igual a la primera y segunda entrega. Los opcionales tienen peso diferente según su complejidad.