

TRABAJO PRÁCTICO INTEGRADOR

ETAPA #3 . PARTE 1 - JS

TUP - Laboratorio de Computación II

Etapas #3 - JavaScript

Continuando con el desarrollo de nuestra aplicación Web de Resultados Electorales. En esta etapa del TP Integrador, tenemos como objetivo realizar las implementaciones de lógica JavaScript para lograr que la aplicación sea funcional y los datos se consulten en tiempo real.

Vamos a hacer que nuestro sitio web nos permita consultar los diferentes resultados electorales de acuerdo con la información que nos regrese una API dispuesta para tal fin.

Utilizaremos funciones asincrónicas, almacenamiento de información en el browser del usuario, llamado y manipulación de API, manipulación del DOM, bucles, etc.

Para esta entrega, se presentan requerimientos EXIGIDOS los cuales se deberán implementar sí o sí en su programa para poder aprobar el TP. Al mismo tiempo, tendrán un listado de requerimientos EXTRAS los cuales son opcionales.

De dividirá esta última parte en dos, para facilitar el entendimiento y seguimiento de los pasos a realizar.

Consideraciones importantes:

1. Antes de avanzar con esta etapa, tener la entrega de la Etapa #1 y #2 corregida por su profesor y haber realizado los ajustes que se le han indicado vía email.
2. Realizar los cambios en el código HTML que requiera o considere necesarios.
3. Enlazar los archivos **.js** correspondientes a cada página. Por ejemplo, en el documento **favorites.html** agregar el **common.js** y el **favorites.js**. No te olvides de colocarle el “defer”-
4. Implementar lógica JS a cada página para lograr los requerimientos exigidos en las siguientes secciones. En caso de detectar que una funcionalidad se repite en diferentes páginas colocarla dentro del **common.js**, sino trabajar en el **js** correspondiente a cada página.
5. Finalizar todos los requerimientos EXIGIDOS y una vez cerrados avanzar, en caso de que cuente con el tiempo, con los requerimientos EXTRAS.
6. Los registros seleccionados por el usuario deben persistirse más allá de la sesión, seleccionar el tipo de storage a utilizar.
7. Mantener la estructura de carpetas y archivos solicitadas en la primer consigna.

Recomendación: Subir los cambios al repo a medida que se va avanzando con el desarrollo.

Conceptos JavaScript por poner en práctica:

- Manipulación del DOM
- Manejo de objetos y arreglos
- Local Storage
- Fetch API
- Función asincrónica

Requerimientos Exigidos

Se propone al alumno seguir los pasos de desarrollo tal y como se mencionan en el enunciado para que pueda ver de forma más tangible la evolución y funcionamiento del código.

PARTE 1

API

Para este ejercicio utilizaremos la API del Ministerio del Interior de la República Argentina, el cual es de uso libre y gratuito, y sólo de consulta, es por ello por lo que no requiere de generación de token key como en otros casos en consultas a API's.

Datos generales de la API a utilizar

url: <https://resultados.mininterior.gob.ar/desarrollo>

Página de Inicio

La página de inicio no requiere de la codificación de ninguna funcionalidad en JS.

Páginas de PASO / GENERALES

Las páginas de PASO y GENERALES tienen la misma estructura, lo que las diferencia a ambas es el tipo de elección que estamos consultando.

Existen dos tipos de elecciones que vamos a utilizar

1 – PASO

2 – GENERALES

Donde 1 o 2 es el ID que debemos utilizar a la hora de consultar

```
const tipoEleccion = 1;
```

o

```
const tipoEleccion = 2;
```

Va a depender de la página donde estemos haciendo la consulta. Sera un dato fijo.

Otro valor que usaremos por defecto es el del tipo de recuento, para este trabajo solamente usaremos aquellos de recuento definitivo

```
const tipoRecuento = 1;
```

Armado de combos

En este punto generaremos las opciones de los combos en forma automática, consultando las mismas desde la API.

Considere que al llenarse los tamaños de los combos van a variar, para ello tenga en cuenta limitar el ancho de estos para que no se desarme la barra de menú.

El llenado de los valores de los combos de selección se debe realizar desde el JS y debe seguir el orden descripto debajo(secuencia de armado):

Combo Año

Con la función fetch consultamos en la API todos los períodos disponibles

```
fetch("https://resultados.mininterior.gob.ar/api/menu/periodos")
```

La consulta nos devolverá el campo "periodo" con la información de los años para armar el combo con las opciones, donde el Texto y el Valor es el mismo.

Combo Cargo

Para completar el combo de Cargos debemos llamar a la función fetch con la url

```
fetch("https://resultados.mininterior.gob.ar/api/menu?año=" +  
periodosSelect.value)
```

Donde el período seleccionado es el valor seleccionado en el combo Año, es decir que debemos validar que se haya seleccionado un registro en el primer combo antes de hacer la consulta de este segundo combo.

Obtenida la respuesta de la consulta en la API se debe filtrar los datos por el tipo de selección que estamos consultando.

Por lo tanto, recorreremos la respuesta

```
datosFiltros.forEach((eleccion) => {
```

Buscando la coincidencia con el tipo de elección que estamos consultando

```
if (eleccion.IdEleccion == tipoEleccion) { //PASO o GERENALES
```

Y obtenemos los datos de los cargos para completar el combo

```
eleccion.Cargos.forEach((cargo) => {
```

Id para el Value y Descripción para el texto

```
cargo.Cargo  
cargo.IdCargo
```

Al cambiar el combo de año se debe limpiar las opciones de este combo antes de cargar las opciones nuevamente.

IMPORTANTE: esta respuesta que obtenemos de la consulta a la API contiene todos los datos que necesitamos para el resto de los combos, por lo tanto, la variable debe estar declarada en forma global ya que en los siguientes combos no volveremos a ejecutar dicha consulta, sino que utilizaremos y filtraremos directamente sobre este JSON.

CONSEJO: Impriman el json que van obteniendo en la consola para poder asociar la información que estan consultando y mostrando.

Combo Distrito

El combo de distrito se debe completar luego de que se hayan seleccionado el año y el cargo.

Como se menciona antes, los datos se encuentran almacenados en la respuesta de Cargos, por lo tanto, debemos recorrer el JSON para obtener estos datos.

- Recorremos la respuesta y Filtramos por Tipo de elección
- Del resultante recorremos cada cargo "Cargos" y lo filtramos por el cargo seleccionado en el combo
- Del resultante recorremos los "Distritos"
- Del resultante tomamos los datos para armar el combo, IdDistrito y Distrito

Al cambiar el combo de Cargo se debe limpiar las opciones de este combo antes de cargar las opciones nuevamente.

Combo Sección

Se debe agregar un nuevo combo para filtrar las secciones.

Al igual que con el combo de Distrito, vamos a tomar el resultado de la consulta de cargos y vamos a filtrar la información de acuerdo con los parámetros que ya fuimos seleccionado.

- Recorremos la respuesta y Filtramos por Tipo de elección
- Del resultante recorremos cada cargo "Cargos" y lo filtramos por el cargo seleccionado en el combo
- Del resultante recorremos los "Distritos" y filtramos por el Distrito seleccionado
- Por cada registro de Distrito recorremos las "SeccionesProvinciales" y obtenemos el valor del IDSeccionProvincial. En principio este valor lo vamos a guardar en un campo del tipo oculto, es decir que, si bien va a estar en el html, el usuario no lo va a ver.
Para ello en la página html, debajo del combo de Distrito, vamos a declarar este campo.

```
<input type="hidden" id="hdSeccionProvincial" name="hdSeccionProvincial">
```

- Del resultante recorremos las "SeccionesProvinciales"
- Del resultante recorremos las "Secciones"
- Del resultante tomamos los datos para armar el combo, IdSeccion y Seccion.

Al cambiar el combo de Distrito se debe limpiar las opciones de este combo antes de cargar las opciones nuevamente.

Botón FILTRAR

- Al filtrar la información se debe confirmar que todos los campos de selección estén completos, en caso de faltar alguno se debe indicar al usuario con un mensaje AMARILLO que campo falta seleccionar.
- En la función asíncrona en JS recupere los valores del filtro que necesitara para consultar en el servicio:
 - o anioEleccion
 - o tipoRecuento
 - o tipoElección
 - o categorialId: 2 (valor por defecto)
 - o distritold
 - o seccionProvincialId
 - o seccionId
 - o circuitold: "" (valor por defecto)
 - o mesald: "" (valor por defecto)
- Consulte el servicio por medio de un fetch considerando:
 - o url: <https://resultados.mininterior.gob.ar/api/resultados/getResultados>
 - o Parámetros:

?anioEleccion=XXXX&tipoRecuento=XXXX&tipoEleccion=XXXX&categoriald=XXXX&dist
ritId=XXXX&seccionProvincialId=XXXX&seccionId=XXXX&circuitId=XXXX&mesalId=XXXX

Donde cada XXXX debe ser reemplazada con el respectivo valor recuperado

- Si la respuesta fue correcta imprima en consola el JSON de la respuesta.
- Si hubo algún error en la ejecución muestre un mensaje ROJO con el detalle del error.

Consulta de resultados electorales

A continuación se describe como se completará la información de cada sección que compone la página.

Esta información se mostrará luego de realizar la consulta y la misma retorne información para mostrar. Al ingresar por primera vez a la página se verá solamente un mensaje AMARILLO indicando al usuario que “Debe seleccionar los valores a filtrar y hacer clic en el botón FILTRAR”.

En caso de que no se obtengan datos se mostraran el Título, y Subtítulo seguidos por un mensaje AMARILLO que indique que no se encontró información para la consulta realizada.

En caso de error, se mostrará el Título y Subtítulo y un mensaje ROJO con la información del error.

Título

El título está compuesto por:

“Elecciones ” + (Valor combo Año) + “ | ” + (Tipo de Elección)

Subtítulo

El subtítulo está compuesto por los parámetros seleccionados en el filtro:

(Valor combo Año) + “ > ” + (Tipo de Elección) + “ > ” + (Valor combo Cargo) + “ > ” (Valor
combo Distrito) + “ > ” + (Valor combo Sección)

Cuadros de Colores

Del JSON resultante de la consulta extraer los datos necesarios para los tres cuadros de colores

```
--  
--  
▼ estadoRecuento:  
  cantidadElectores: 131172  
  cantidadVotantes: 73173  
  mesasEsperadas: 0  
  mesasTotalizadas: 379  
  mesasTotalizadasPorcentaje: 0  
  participacionPorcentaje: 55.78
```

- **Mesas Escrutadas:** estadoRecuento.mesasTotalizadas
- **Electores:** estadoRecuento.cantidadElectores
- **Participación sobre escrutado:** estadoRecuento.participacionPorcentaje

Cuadro Agrupaciones Políticas

- Próximamente – Parte 2

Cuadro Mapa

- De acuerdo con el Distrito seleccionado en la consulta muestre el mapa correspondiente. Para mantener el orden del código genere un archivo js llamado mapas.js donde contenga en un array los pares provincia: svg. Incluya este archivo en todas las páginas donde utilizará los mapas

Resumen de Votos

- Próximamente – Parte 2

+ AGREGAR A INFORME

Al hacer clic el usuario en el botón se debe almacenar en un array los datos seleccionados y los mismos serán guardados para su utilización en la página de informes, para ello deberá optar por utilizar LocalStorage, SessionStorage o IndexDB.

El formato del storage será el siguiente:

Key: INFORMES

Values: [vAnio|vTipoRecuento|vTipoEleccion|vCategoriald|vDistrito|vSeccionProvincial|vSeccionID; ...; n]

Al agregar un nuevo registro de debe validar que el mismo ya no exista en el array almacenado. Para ello se comparará la cadena nueva con las cadenas existentes. En el caso de encontrar una coincidencia se debe mostrar un mensaje AMARILLO al usuario. Utilice la función .includes().

Una vez añadido el registro se debe indicar al usuario con un mensaje VERDE que la operación fue exitosa.

Mensajes a Usuario

Los mensajes de usuario se encontrarán ocultos, y se mostrarán cuando haya sucedido un acontecimiento de acuerdo con:

- AMARILLO, mensajes de advertencia, avisos.

! La operación no se pudo completar

- VERDE, mensajes de operaciones exitosas.

👍 Se agrego con éxito el resultado al informe

- ROJO, mensajes de error, fallas en las consultas.



Error: Se produjo un error al intentar agregar resultados al informe

Los mensajes deben aparecer en el espacio entre el menú y el título de la sección.

Página de INFORMES

○ Próximamente – Parte 2

Contemplar escenarios de error retornados por las API

En las solicitudes a las API puede que haya un error. Esta funcionalidad servirá para manejar esos errores de forma apropiada e informar al usuario de estos para que realice algún cambio sobre la información enviada.

En caso de obtener un error, notificar con un mensaje de error con los formatos indicados.

Se recomienda el uso de try catch para proteger diferentes secciones de la aplicación que sean susceptibles a errores e informar los mismos en caso de que ocurran.

Requerimientos Extras

Estos requerimientos a continuación son opcionales y servirán para sumar nota a la calificación final y por ende facilitará alcanzar la condición de aprobación directa.

Contemplar escenario de carga

Esta funcionalidad permitirá darle al usuario un mensaje para informarle cuando una operación está siendo realizada. Puede hacer uso de iconos o spinners en el HTML.

El usuario debería ver este aviso de cargando en la pantalla solo de modo informativo cuando realiza las siguientes operaciones:

- Cargar Resultados de las PASO / GENERALES
- Consultar página de INFORMES

Si considera que hay operaciones que se realizan de forma muy rápida, que no se alcanza a visualizar el icono de carga, puede utilizar `setTimeout` para dar una mejor experiencia al usuario.

Subir aplicación a Github Pages

Seguir instrucciones dadas en la página oficial de **Github Pages** (<https://pages.github.com/>) para la correcta subida y despliegue de la aplicación en la plataforma.

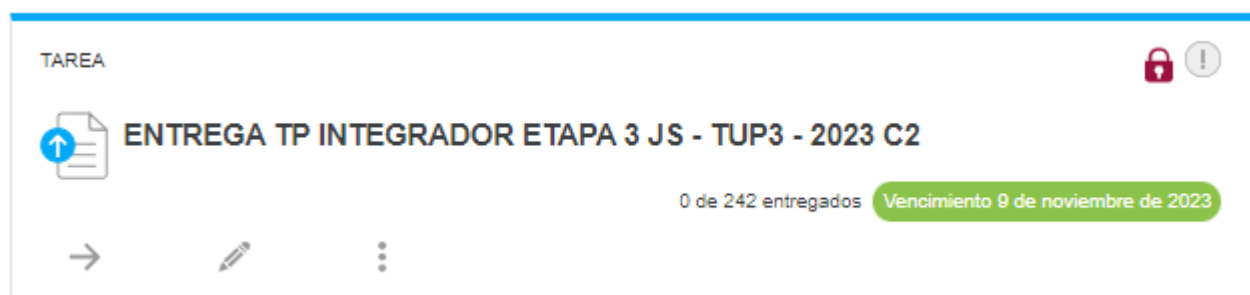
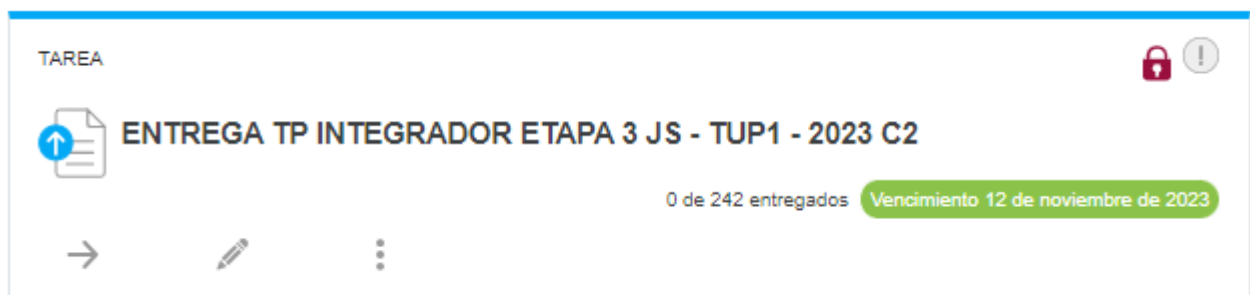
Condiciones de entrega

La entrega se realiza en la fecha de acuerdo con la Comisión a la que pertenece.



Por más que la presentación se pueda dar en días diferentes, por la división de grupos, toda la comisión entrega el mismo día.


Las entregas se encuentran habilitadas desde el 1ro de Noviembre hasta las 23:59hs de la fecha de vencimiento de cada comisión

En CVG, en el apartado de TP Integrador encontraran la sección habilitada para cada grupo de TUP.



TAREA








ENTREGA TP INTEGRADOR ETAPA 3 JS - TUP4 - 2023 C2

0 de 242 entregados

Vencimiento 12 de noviembre de 2023







En el cuerpo del mensaje, se debe indicar:

- URL del repositorio del código fuente.
- URL de Git Pages donde esta publicado el sitio.
- ZIP / RAR con archivos
- Consideraciones: cualquier consideración relacionada al desarrollo y/o resolución del problema. Si te pareció muy fácil o complicado. Que mejorarías para las futuras etapas.