

TRABAJO PRÁCTICO INTEGRADOR

ETAPA #3 . PARTE 2 - 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 2

Páginas de PASO / GENERALES

En esta segunda parte vamos a completar la información mostrada en los cuadros de Agrupaciones y de Votos.

De la respuesta que obtuvimos en la Parte 1 en este punto, vamos a tomar los datos que nos regresa en el array `valoresTotalizadosPositivos`:

```
▼ Promise {<pending>} 1
  ► [[Prototype]]: Promise
  ► [[PromiseState]]: "fulfilled"
  ► [[PromiseResult]]: Object
    ► estadoRecuento: {mesasEsperadas: 0, mesasTotalizadas: 517, mesasTotalizadasPorcentaje: 0, cantidadElectores: 178223, cantidadVotantes: 121815, ...}
    ► fechaTotalizacion: "2023-07-18T18:27:59.251Z"
    ► valoresTotalizadosOtros: {votosNulos: 942, votosNulosPorcentaje: 0.77, votosEnBlanco: 5192, votosEnBlancoPorcentaje: 4.26, votosRecurridosComandoImpugnados: 67, ...}
    ▼ valoresTotalizadosPositivos: Array(12)
      ► 0: {idAgrupacion: '5', nombreAgrupacion: 'DEMOCRATA CRISTIANO', votos: 1742, votosPorcentaje: 1.43, idAgrupacionTelegrama: '', ...}
      ► 1: {idAgrupacion: '86', nombreAgrupacion: 'FRENTE RENOVADOR AUTENTICO', votos: 138, votosPorcentaje: 0.11, idAgrupacionTelegrama: '', ...}
      ► 2: {idAgrupacion: '01-504', nombreAgrupacion: 'CONSENSO FEDERAL', votos: 7226, votosPorcentaje: 5.93, idAgrupacionTelegrama: '', ...}
      ► 3: {idAgrupacion: '01-262', nombreAgrupacion: 'EL MOVIMIENTO', votos: 396, votosPorcentaje: 0.33, idAgrupacionTelegrama: '', ...}
      ► 4: {idAgrupacion: '01-503', nombreAgrupacion: 'JUNTOS POR EL CAMBIO', votos: 54542, votosPorcentaje: 44.77, idAgrupacionTelegrama: '', ...}
      ► 5: {idAgrupacion: '13', nombreAgrupacion: 'MOVIMIENTO AL SOCIALISMO', votos: 1034, votosPorcentaje: 0.85, idAgrupacionTelegrama: '', ...}
      ► 6: {idAgrupacion: '01-501', nombreAgrupacion: 'FRENTE DE IZQUIERDA Y DE TRABAJADORES - UNIDAD', votos: 4604, votosPorcentaje: 3.78, idAgrupacionTelegrama: '', ...}
      ► 7: {idAgrupacion: '01-502', nombreAgrupacion: 'FRENTE DE TODOS', votos: 40482, votosPorcentaje: 33.23, idAgrupacionTelegrama: '', ...}
      ► 8: {idAgrupacion: '87', nombreAgrupacion: 'UNITE POR LA LIBERTAD Y LA DIGNIDAD', votos: 3678, votosPorcentaje: 3.02, idAgrupacionTelegrama: '', ...}
      ► 9: {idAgrupacion: '01-187', nombreAgrupacion: 'AUTODETERMINACION Y LIBERTAD', votos: 1331, votosPorcentaje: 1.09, idAgrupacionTelegrama: '', ...}
      ► 10: {idAgrupacion: '01-343', nombreAgrupacion: 'MOVIMIENTO DE JUBILADOS Y JUVENTUD', votos: 281, votosPorcentaje: 0.23, idAgrupacionTelegrama: '', ...}
      ► 11: {idAgrupacion: '88', nombreAgrupacion: 'PARTIDO DIGNIDAD POPULAR', votos: 160, votosPorcentaje: 0.13, idAgrupacionTelegrama: '', ...}
      ► length: 12
    ► [[Prototype]]: Array(0)
  ► [[Prototype]]: Object
```

Antes de continuar debemos asegurarnos de que todos los valores que conformen la url estén correctos, en el caso de tener algún valor undefined debemos cambiarlo por "".

Ejemplo:

```
https://resultados.mininterior.gob.ar/api/resultados/getResultados?anioEleccion=2019&tipoRecuento=1&tip
oEleccion=1&categoriaId=2&distritoId=1&seccionProvincialId=&seccionId=1&circuitId=&mesaId=
```

Para comenzar a trabajar con los cuadros de datos, debemos definir en forma global un objeto que contenga todas las agrupaciones políticas y el color que las representa.













(Se sugiere un objeto, si considera que es mejor otra forma de almacenar esta información impleméntela)

Por ejemplo

```
{5:{colorPleno, colorLiviano},}
```

Donde 5 representa el ID de la agrupación, y colorPleno es el color sin opacidad y colorLiviano es el color con opacidad.

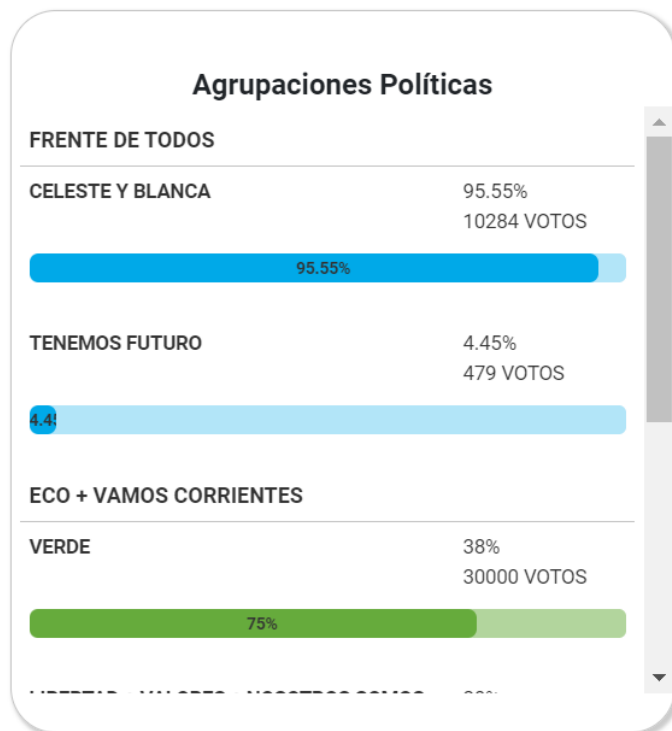
Actualmente tenemos definidos estos colores:

```
--grafica-amarillo:  rgb(252, 210, 0);
--grafica-amarillo-claro:  rgba(252, 210, 0, 0.3);
--grafica-celeste:  rgb(0, 169, 232);
--grafica-celeste-claro:  rgba(0, 169, 232, 0.3);
--grafica-bordo:  rgb(171, 40, 40);
--grafica-bordo-claro:  rgba(171, 40, 40, 0.3);
--grafica-lila:  rgb(112, 76, 159);
--grafica-lila-claro:  rgba(112, 76, 159, 0.5);
--grafica-lila2:  rgb(77, 46, 110);
--grafica-lila2-claro:  rgba(77, 46, 110, 0.5);
--grafica-gris:  rgb(128, 128, 128);
--grafica-gris-claro:  rgba(128, 128, 128, 0.5);
--grafica-verde:  rgb(102, 171, 60);
--grafica-verde-claro:  rgba(102, 171, 60, 0.5);
```

Pero dado que tenemos muchas más agrupaciones, deberán definir otras variantes de colores para todas las agrupaciones.

No tenemos un end point que nos permita consultar todas las agrupaciones políticas disponibles, por lo tanto, defina los colores para la mayor cantidad de agrupaciones que pueda y seleccione un color gris o neutro para el resto.

Cuadro Agrupaciones Políticas



Para armar este cuadro de Agrupaciones Políticas recorreremos el array y por cada agrupación armaremos el bloque de información donde se listará internamente cada una de las listas que componen el partido

Donde:

```
▼ 0:
  idAgrupacion: "17-501"
  idAgrupacionTelegrama: ""
  ▼ listas: Array(2)
    ► 0: {nombre: 'TENEMOS FUTURO', numero: 'K', votos: 479}
    ► 1: {nombre: 'CELESTE Y BLANCA', numero: 'A', votos: 10284}
    length: 2
    ► [[Prototype]]: Array(0)
  nombreAgrupacion: "FRENTE DE TODOS"
  urlLogo: ""
  votos: 10763
  votosPorcentaje: 43.73
  ► [[Prototype]]: Object
```

FRENTE DE TODOS ●

CELESTE Y BLANCA ●

95.55% ●

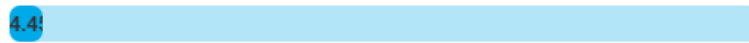
10284 VOTOS ●



TENEMOS FUTURO ●

4.45% ●

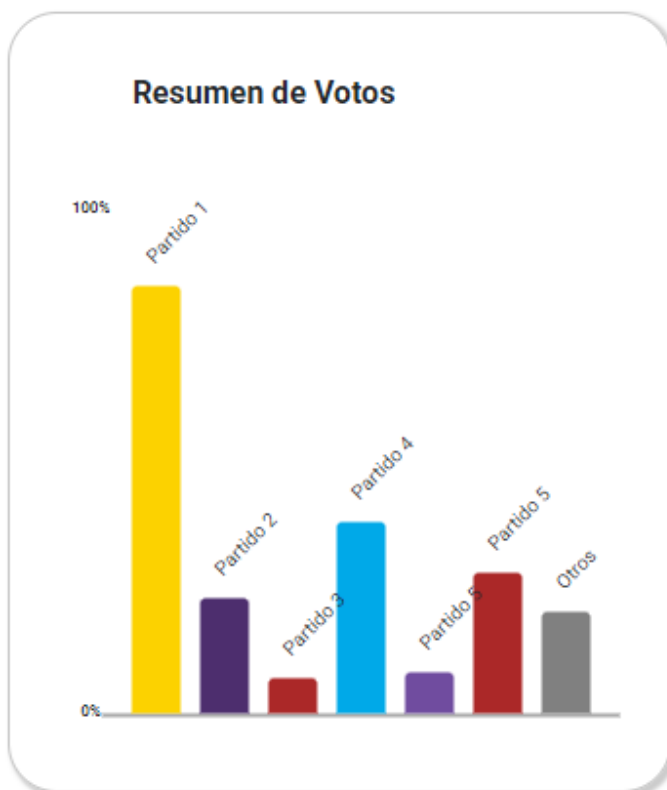
479 VOTOS ●



- **ROJO:** nombreAgrupación → Título de la agrupación
- Luego recorro el array **listas** y armo cada uno de los bloques debajo de la agrupación.
 - **VERDE:** nombre
 - **AZUL:** Valor calculado ($\text{listas.votos} * 100 / \text{valoresTotalizadosPositivos.votos}$)
 - La barra de avance debe tener este mismo porcentaje.
 - Utilizar los colores definidos para la agrupación (Pleno y Liviano).
 - **CELESTE:** votos

Es posible que muchas agrupaciones solo tengan una lista entre su detalle.

Resumen de Votos



Se igual manera que en el cuadro de agrupaciones políticas, vamos a recorrer solamente las agrupaciones y vamos a tomar los datos para armar la gráfica.

Importante: Graficar un máximo de 7 agrupaciones.

Donde:

- Título Partido: `nombreAgrupación`
- Porcentaje: `votosPorcentaje`
- Color: Tomar el color pleno definido para cada partido

Página de INFORMES

Al ingresar a la página de informes debemos contemplar:

- Validar que existan registros (valores) guardados en el localStorage
 - o Si no existen datos guardados mostrar mensaje AMARILLO indicando al usuario que “No hay informes guardados para mostrar”
- Si existen datos, tomar el Array, y recorrerlo.
 - o Por cada registro del array se debe separar los datos (split) para poder formar la url.
 - o Realizar la consulta a la api con los datos guardados
 - o Armar el registro de la tabla
 - **PROVINCIA:** Mapa Provincia (objeto de provincias)
 - **ELECCIÓN:** Tomo los datos guardados en el Storage para armar el registro (*)
 - **Título**
El titulo está compuesto por:
`“Elecciones ” + (Valor combo Año) + “ | ” + (Tipo de Elección)`
 - **Subtitulo**
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)`
 - **DATOS GENERALES**
 - ▼ `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
- **DATOS POR AGRUPACIÓN**
En esta ocasión solamente mostraremos los datos consolidados de la agrupación

```

▼ 0:
  idAgrupacion: "5"
  idAgrupacionTelegrama: ""
  ▼ listas: Array(1)
    ► 0: {nombre: 'CIUDAD CELESTE POR LA VIDA Y LA FAMILIA', numero: 'A', votos: 1742}
      length: 1
    ► [[Prototype]]: Array(0)
  nombreAgrupacion: "DEMOCRATA CRISTIANO"
  urlLogo: ""
  votos: 1742
  votosPorcentaje: 1.43

```

Recorremos todas las agrupaciones y por cada una ponemos

```

nombreAgrupacion
votosPorcentaje + "%"
votos + "Votos"

```

(*) Si es necesario, agregue los datos faltantes en el storage al momento de armar el Array, considere agregar estos datos al final de todos los parámetros necesarios para la consulta y recupere los mismos como ...REST

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.

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 - TUP1 - 2023 C2**

0 de 242 entregados **Vencimiento 12 de noviembre de 2023**






TAREA






 **ENTREGA TP INTEGRADOR ETAPA 3 JS - TUP3 - 2023 C2**

0 de 242 entregados **Vencimiento 9 de noviembre de 2023**






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.

