

TRABAJO PRÁCTICO INTEGRADOR

ETAPA #2 - JS

TUP - Laboratorio de Computación II

Introducción

El presente documento enuncia los requerimientos de desarrollo para el Trabajo Práctico Integrador para todos los alumnos de Laboratorio de Computación II.

El TPI tiene como objetivo principal la construcción de una Página Web aplicando los conceptos dados en la materia, durante el cursado, sobre **HTML, CSS y JavaScript**.

TPI 2024 – Cotizador de Monedas Web

Este trabajo práctico tiene como requerimiento que el alumno construya una aplicación web que sirva para visualizar las diferentes cotizaciones de monedas, guardar datos de interés, compartir los mismos, y generar gráficas evolutivas.

En las diferentes etapas se guiará al alumno en el desarrollo de la solución:

- **Etapas #1 - HTML:** Se trabajará principalmente con código HTML, agregando elementos y enlazando las diferentes páginas.
- **Etapas #2 - CSS:** En esta etapa se centrará en el agregado de estilo a las páginas para lograr una correcta visualización de lo que sería la maquetación de la App.
- **Etapas #3 - JS:** La Web se finalizará en esta etapa donde se realizará el desarrollo completo de la lógica JavaScript para darle vida y correcto funcionamiento según los requerimientos.

***IMPORTANTE:** cada una de las etapas tiene una entrega obligatoria que deberá ser aprobada para poder avanzar con la siguiente.*

Es requerido que cada alumno cree un repositorio en GitHub y vaya subiendo todo el código a medida que va avanzando con el desarrollo.

Para las presentaciones se va a solicitar que la web se encuentre publicada en GitHub Pages.

CONSEJO: Lee atentamente las consignas, avanza una a una, si tiene dudas consulte al docente, y prestar mucha atención a las imágenes de ejemplo.

Etapla #2 - 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, almacenado 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.

Consideraciones importantes:

1. Antes de avanzar con esta etapa, tener la entrega de la Etapa #1 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 `miarchivo.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 suficiente, 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 se forma más tangible la evolución y funcionamiento del código.

PARTE 2

API

En esta segunda parte vamos a utilizar la API pública DolarAPI.com desde donde obtendremos las diferentes cotizaciones del Dólar, Euro, Real y Pedro Chileno y Uruguayo.

Documentación API

<https://dolarapi.com/docs/argentina/>

Funcionalidades esperadas

Indice

~~Al ingresar a la pantalla el sistema debe mostrar TODAS las cotizaciones del día, de la misma manera que si seleccionase la opción TODAS del combo box y buscare.~~

Se debe validar que la consulta a la API no devuelva error. Si sucede mostrar un mensaje de advertencia al usuario en pantalla de acuerdo con lo especificado en Etapa #1

~~El usuario puede visualizar el listado de cotizaciones y seleccionar aquellas cotizaciones que desee agregar a su informe. El sistema indicara si la cotización se ha agregado correctamente o si la misma ya se encuentra almacenada en el caso de que esta coincida en La Moneda y Fecha / Hora.~~

~~El almacenamiento de las cotizaciones se deberá mantener, NO debe ser Temporal.~~

~~Por pantalla el usuario visualizará los cuadros con el mismo formato y los valores de COMPRA Y VENTA de la moneda.~~

~~Si lo desea el usuario puede buscar una moneda en particular.~~

~~La Fecha y Hora de actualización que se muestra por pantalla es la de consulta a la API.~~

~~Esta vista se deberá actualizar en forma periódica cada 5 minutos.~~

Mi Archivo

~~Al ingresar a la pantalla el sistema debe mostrar TODAS las cotizaciones que guardó el usuario para consultar luego.~~

~~Las mismas se muestran agrupadas por Fecha, Monedas (todas las monedas almacenadas, es posible que se repitan si la moneda fue guardada en diferentes momentos del día), y Precios de Compra y Venta.~~

Por último, en cada línea el usuario podrá quitar esa cotización de su informe.

Imprimir contenido del informe mediante una función JS que solo envíe a la impresora el recuadro de datos

Ejemplo

```
//JS

function imprSelec(nombre) {
    var contenido = document.getElementById(nombre).innerHTML;
    var contenidoOriginal = document.body.innerHTML;

    document.body.innerHTML = contenido;
    window.print();
    document.body.innerHTML = contenidoOriginal;
}

//HTML
<div id='imprimir'>
    <p>Nombre: <u>Francisco Perez Martinez</u></p>
    <table class="text-center table-bordered">
        <thead class="bg-secondary text-white">
            <th width="30"><font size="1">#</font></th>
            <th width="200"><font size="1">HERRAMIENTA</font></th>
            <th width="50"><font size="1">CANTIDAD</font></th>
        </thead>
    </table>
</div>
```

Considerar que pueden no existir datos almacenados, por lo tanto, NO se debe mostrar ninguna parte del cuerpo principal, solamente un mensaje que indique que no hay datos guardados hasta el momento.

Informes

La página de informes debe mostrar los datos almacenados en formato de informe.

Por medio de un combo box puede seleccionar una moneda o todas. Por defecto al ingresar se mostrarán TODAS las monedas almacenadas.

Se mostrará la gráfica de los datos que se exponen debajo.

En la grilla se muestran los datos agrupados por Moneda, luego de la misma la fecha y hora de la cotización, de la mas reciente a la mas antigua; valores de compra y venta y un icono de variación. La variación esta dada por la comparación del precio de venta, en el caso de tener un comparativo anterior, ejemplo, si la cotización del dólar ese día previamente o días anteriores era de \$10 y hoy es de \$15 significa que está en alza.

Se podrá compartir esta información por email, para ello se debe utilizar el componente de envío de emails mencionado más abajo. Se deben validar los datos de entrada y advertir al usuario si hay un error.

Hay que considerar que pueden no existir datos almacenados, por lo tanto, NO se debe mostrar ninguna parte del cuerpo principal, solamente un mensaje que indique que no hay datos guardados hasta el momento.

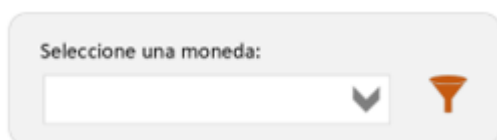
Contacto

En la pagina de contacto se mostrará un formulario que, por medio de la Api de envío de emails, enviará un correo de contacto.

Se deben validar los datos de entrada y advertir al usuario si hay un error.

Componentes en pantalla

Selector de Monedas


The image shows a UI component for selecting a currency. It consists of a light gray rounded rectangle. Inside, at the top left, is the text 'Seleccione una moneda:'. Below this text is a white rectangular input field. To the right of the input field is a small gray downward-pointing arrow icon. To the right of the arrow icon is a small orange funnel icon.

Las opciones del combo box deben ser:

- TODAS -> Se van a mostrar todas las cotizaciones disponibles

- Dólar Oficial
- Dólar Blue
- Dólar Bolsa (MEP)
- Dólar Contado con Liqui (CCL)
- Dólar Tarjeta
- Dólar Mayorista
- Dólar Cripto
- Euro
- Real Brasileño
- Peso Chileno
- Peso Uruguayo

Información Cotización de Moneda

OFICIAL	COMPRA \$847	VENTA \$887	
---------	-----------------	----------------	---

Datos endpoints a consultar:

- Dólar Oficial: <https://dolarapi.com/v1/dolares/oficial>
- Dólar Blue: <https://dolarapi.com/v1/dolares/blue>
- Dólar Bolsa (MEP): <https://dolarapi.com/v1/dolares/bolsa>
- Dólar Contado con Liqui (CCL): <https://dolarapi.com/v1/dolares/contadoconliqui>
- Dólar Tarjeta: <https://dolarapi.com/v1/dolares/tarjeta>
- Dólar Mayorista: <https://dolarapi.com/v1/dolares/mayorista>
- Dólar Cripto: <https://dolarapi.com/v1/dolares/cripto>
- Euro: <https://dolarapi.com/v1/cotizaciones/eur>
- Real Brasileño: <https://dolarapi.com/v1/cotizaciones/brl>
- Peso Chileno: <https://dolarapi.com/v1/cotizaciones/clp>
- Peso Uruguayo: <https://dolarapi.com/v1/cotizaciones/uyu>

Buscar la mejor manera de tener los datos disponibles y definidos una única vez y que se usen en todo el sitio (Variables, Array, Objeto, Array de Objetos, Objeto de Arrays, etc)

Ejemplo de implementación:

Llamado:

```
fetch("https://dolarapi.com/v1/cotizaciones/uyu")
  .then(response => response.json())
```

```
.then(data => console.log(data));
```

Resultado

```
{
  "moneda": "UYU",
  "casa": "oficial",
  "nombre": "Peso Uruguayo",
  "compra": 23.13,
  "venta": 23.17,
  "fechaActualizacion": "2024-06-09T11:55:00.000Z"
}
```

Rotador de opiniones

Implementar el código necesario para que el recuadro de opiniones vaya rotando y cambiando en forma automática en cada página.

Aquí algunos ejemplos de rotadores sencillos para utilizar.

Asegúrese que el mismo funcione en forma automática y no requiera intervención del usuario.

- <https://desarrolloweb.com/articulos/rotacion-banners-javascript.html>
- <https://www.uv.es/castelmo/javascript/banners.htm>
- <https://programadorwebvalencia.com/javascript-carousel-sencillo-con-controles-y-autoreproduccion/>

Envío de Emails

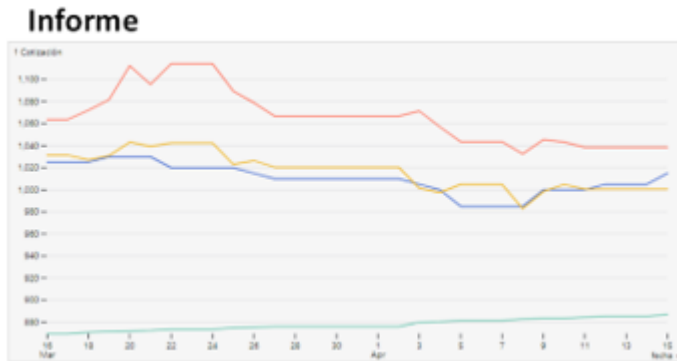
Para integrar la aplicación al servicio de mensajería por email se debe usar la librería [EmailJS](#). Esta librería JS permite el envío de emails únicamente utilizando su API JavaScript que realiza la operación internamente.

Esta implementación deberá realizarse sobre la página **Contacto y de Informe**. Pueden consultar [aquí](#) la documentación de la librería para obtener instrucciones precisas de cómo integrarla correctamente: <https://www.emailjs.com/docs/tutorial/overview/>.

En caso de fallo del envío de emails se debe informar al usuario del problema detectado y que vuelva a intentar más tarde.

Al hacer la entrega, deben cambiar la dirección de contacto de destino por hgerardo@gmail.com y sbruselario@gmail.com

Gráfica de Informes



Para utilizar el graficador incluir el script Chart.js

<https://cdn.jsdelivr.net/npm/chart.js@latest/dist/Chart.min.js>

Este script grafica a través del objeto de html canvas por lo tanto se debe incluir el código en el body:

```
<canvas id="miGrafica"></canvas>
```

Luego desde JS se deben configurar las propiedades para que se dibuje la gráfica.

Aquí compartimos ejemplos de cómo utilizar el graficador

```
/*Gráfica con una sola línea*/
const etiquetas = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "junio"];
const datos = [100, 150, 120, 200, 0, 20];

const ctx = document.getElementById("miGrafica").getContext("2d");
new Chart(ctx, {
  type: "line",
  data: {
    labels: etiquetas,
    datasets: [{
      label: "Ventas por mes",
      data: datos,
      borderColor: "blue",
      fill: false
    }]
  }
});
```



```

/*****
/*Gráfica con varias líneas*/
//Axis X
const etiquetas = ["Enero", "Febrero", "Marzo", "Abril", "Abril", "Mayo", "junio"];
//Datos
const datosLinea1 = [100, 150, 120, 200, 10, 20, 100];
const datosLinea2 = [80, 120, 140, 180, 0, 50, 56];
const datosLinea3 = [88, 100, 14, 200, 20, 0, 80];

const ctx = document.getElementById("miGrafica").getContext("2d");
new Chart(ctx, {
  type: "line",
  data: {
    labels: etiquetas2,
    datasets: [
      //Porción de código que se repite por cada ítem que se requiere dibujar
      { //Ejemplo de gráfica con relleno
        label: "Dolar Blue",
        data: datosLinea1,
        borderColor: "rgba(54, 162, 235, 1)",
        backgroundColor: 'rgba(54, 162, 235, 0.2)', // Color de fondo
        borderWidth: 1,
        fill: true
      },
      {
        label: "Dolar Oficial",
        data: datosLinea2,
        borderColor: "green",
        borderWidth: 1,
        fill: false
      },
      {
        label: "Euro",
        data: datosLinea3,
        borderColor: "red",
        fill: false
      }
    ]
  }
});

```

Cuando se realizó una búsqueda por una moneda en particular se deberá mostrar la variación de valores entre la venta y la compra.

Si la búsqueda se realizó por todas las monedas, entonces mostrar únicamente el precio de compra de esa moneda

Extra

- Como actividad extra, agregar una página que permita mostrar la evolución del dólar por:
 - Tipo de Dólar y Fecha
 - <https://argentinadatos.com/docs/operations/get-cotizaciones-dolares.html>
 - Tipo de Dólar
 - <https://argentinadatos.com/docs/operations/get-cotizaciones-dolares-casa.html>
 - Tipo de Dólar de acuerdo a una fecha en particular
 - <https://argentinadatos.com/docs/operations/get-cotizaciones-dolares-casa-fecha.html>

En los tres casos la pantalla debe tener los campos de filtrado, y los datos se mostrarán en una gráfica de líneas solamente.

El desarrollo puede estar realizado en una única página, con varias secciones o en varias páginas y debe estar linkeado desde todas las páginas.



Resultado final


Chequear que el resultado obtenido se asemeje lo más posible a las capturas de este documento. Si hay algún detalle que no has podido lograr, consultar en clase o vía email para resolverlo y en última instancia, realizar la consideración en la entrega vía email.

Entrega

La entrega para la parte de HTML y CSS es única, se realizan ambas entregas en la misma fecha y de acuerdo con la Comisión a la que pertenece.




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



TAREA





ENTREGA TP INTEGRADOR ETAPA 2 JS - TUP06 C1 2024

No publicado a los alumnos

0 de 251 entregados
Vencimiento 7 de julio de 2024










TAREA





ENTREGA TP INTEGRADOR ETAPA 2 JS - TUP07 C1 2024

No publicado a los alumnos

0 de 251 entregados
Vencimiento 7 de julio de 2024








TAREA




ENTREGA TP INTEGRADOR ETAPA 2 JS - TUP08 C1 2024

No publicado a los alumnos

0 de 251 entregados
Vencimiento 7 de julio de 2024

En el cuerpo del mensaje, se debe indicar:

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