	Facultad de Tecnología y Cs. Aplicadas – U.N.Ca.	CARRERA: Ingeniería en Informática
	CÁTEDRA: Programación III	PRACTICO III: Programación del lado del Cliente

TRABAJO PRACTICO N° III

PROGRAMACIÓN DEL LADO DEL CLIENTE

Objetivos: el objetivo del presente trabajo practico es que el alumno conozca las características básicas del lenguaje JavaScript, para el diseño de páginas Web con contenido dinámico.

Capacidades a desarrollar: Ser capaz de hacer una búsqueda bibliográfica por medios diversos (bibliotecas, librerías, Internet, centros de documentación, etc.), de seleccionar el material relevante (que sea a la vez válido y actualizado) y de hacer una lectura comprensiva y critica del mismo.

Bibliografía:

Sitios de Referencias:

- <https://www.w3schools.com/>

Carácter de elaboración: el presente trabajo práctico deberá ser desarrollado en forma individual o en grupo de hasta tres integrantes.

Forma y fecha de entrega: Subir al aula virtual una carpeta comprimida, incluyendo caratula del trabajo practico, integrantes del grupo, los archivos html correspondientes junto con el archivo “validarCampos.js”. Fecha de entrega 14/09/2018.

Actividades a desarrollar:

EJERCICIO N° 1

A partir del formulario generado en el Ejercicio N° 3 del Trabajo Practico N° 1 y utilizando el lenguaje del lado del cliente como lo es JavaScript, realizar las validaciones que se detallan en la Tabla 1:

Tabla 1. Lista de validaciones sobre controles de formularios


Campo	Validación
Fecha de Ticket	No puede ser anterior que la fecha actual
Horario	Una vez que el campo Fecha de Ticket pierda el foco, verificar si el día de la semana es viernes, actualizar la lista desplegable “horario” con las opciones: 14:00, 14:30, 21:00, 21:30, 22:00, 22:30. Para el resto de los días actualizar las opciones con 14:00, 14:30, 21:00, 21:30
2X1 3D Banco Santander Rio	Si se selecciona un número mayor a cero, se deberá mostrar el campo de texto obligatorio: PRIMEROS 6 DÍGITOS TARJETA SANTANDER RIO (ver Figura 1), caso contrario, no se debe mostrar ningún campo adicional.
2X1 3D Tarjeta Nevada	Si se selecciona un número mayor a cero, se deberá mostrar el campo de texto obligatorio: PRIMEROS 6 DÍGITOS TARJETA TARJETA NEVADA (ver Figura 1), caso contrario, no se debe mostrar ningún campo adicional.
2X1 3D Club Personal	Si se selecciona un número mayor a cero, se deberá mostrar el campo de texto obligatorio: CÓDIGO PERSONAL3D (ver Figura 1), caso contrario, no se debe mostrar ningún campo adicional.
SubTotal	El subtotal deberá ser calculado, multiplicando Cantidad * Costo
Total	El total deberá ser calculado sumando los subtotales

TIPO DE TICKETS	CANT:	COSTO:	SUBTOTAL:
3D LUN A MIERC DESCUENTO:	<input type="text" value="9"/>	\$90.00	\$810.00
2X1 3D BANCO SANTANDER RIO:	<input type="text" value="2"/>	\$75.00	\$150.00
2X1 3D TARJETA NEVADA:	<input type="text" value="2"/>	\$75.00	\$150.00
2X1 3D CLUB PERSONAL:	<input type="text" value="2"/>	\$75.00	\$150.00
<div> CÓDIGO PERSONAL3D: <input type="text"/> PRIMEROS 6 DÍGITOS TARJETA SANTANDER RIO: <input type="text"/> PRIMEROS 6 DÍGITOS TARJETA TARJETA NEVADA: <input type="text"/> </div>			
Incluye IVA		TOTAL: \$1260.00	

Figura 1. Campos actualizados dinámicamente

Nota:

- En caso que la validación de “Fecha” falle, emitir un mensaje informando el error ocurrido, por ej.: “Fecha de Ticket debe no debe ser Menor a la Fecha Actual”

	Facultad de Tecnología y Cs. Aplicadas – U.N.Ca.	CARRERA: Ingeniería en Informática
	CÁTEDRA: Programación III	PRACTICO III: Programación del lado del Cliente

- Utilizar funciones en un archivo separado con las funciones JavaScript para validar cada uno de los campos, el cual deberá llamarse “validarCampos.js”.

EJERCICIO Nº 2

En una aplicación web, se pretende crear una agenda, guardando el nombre del propietario de dicha agenda y agregarle a la misma una lista de tareas.

Para ello se define en JavaScript la siguiente clase

```
1. function Agenda (propietario) {
2. this.propietario = propietario;
3. this.tareas = [];
4. }
5.
6. Agenda.prototype.addTarea = function(t) {
7. this.tareas.push(t);
8. };
```

Las tareas tienen un nombre, una descripción, una prioridad (se elige de un radio button), un combo donde se elige la categoría (tramite, reunión, recordatorio, otro), un mail donde avisar, una duración en minutos, y una fecha de la tarea.

El diseño de la interface de usuario prevé una estructura de dos bloques:

- El bloque superior muestra quien es el dueño de la agenda y si la agenda no tiene dueño permite crear una agenda y asignar el dueño.
- El bloque inferior, permite cargar una a una las tareas y mostrarla en una lista

Cuando el usuario abra el archivo en el navegador deberá ver lo siguiente:

Agenda

Dueño de la agenda

PEPE

Crear Agenda

Una vez creada la agenda se visualiza quien es el dueño y se muestra el formulario para cargar tareas

Agenda

Dueño de la agenda

PEPE

Gestion de tareas

Si agrega una tarea la verá en un listado

Agenda

Dueño de la agenda

PEPE

Gestion de tareas

- sadfasdf - 00:59

Se le entregan 2 archivos

- pagina4.html : página HTML con la estructura básica del laboratorio
- agenda.js : código javascript para manipular el árbol DOM e interactuar con la aplicación

Tareas a desarrollar:

1. Agregue en el archivo agenda.js los atributos necesarios para describir una tarea.
2. Analice el código restante y modifique apropiadamente la línea que tiene como contenido `"btnAddTarea.addEventListener('click',function(){"` para que se tengan en cuenta los nuevos atributos de una tarea.
3. Resuelva los ítems comentados en el archivo pagina4.html
 - a. COMPLETAR#1: incluya el archivo javascript en el documento HTML.
 - b. COMPLETAR#2: indicar que cuando termina de cargar el documento se ejecuta el método de inicio de todas las funciones definido en el archivo javascript
 - c. COMPLETAR#3: crear un elemento button que permita crear la agenda, y asignarle un ID que será usado luego en el script para asociar el botón a un evento
 - d. COMPLETAR#4: crear los elementos necesarios para que se pueda dar de alta una tarea.
4. Agregar en la lista de tareas de la agenda, un botón o hipervínculo, asociado a cada tarea que permita eliminarla e implementar dicho método en la lógica JavaScript.