UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

EVALUACIÓN	**PARCIAL**	**GRUPO**	M1C	**FECHA**	29/11/2021
MATERIA	Programación 1				
CARRERA	AP/ATI				
CONDICIONES	**- Puntos: 50** **- Duración y mecanismo de entrega: 2 horas, realización y entrega presenciales** **- Sin material y realizado en forma individual** **- Consultas: Exclusivamente de interpretación y/o alcance de letra**				

Una empresa nos solicita el desarrollo de una aplicación para gestionar alquileres de herramientas. Se desea poder tener información de las herramientas que se tienen, los alquileres realizados, etc.

De cada herramienta se desea guardar la siguiente información: id (un autonumérico único), nombre, fecha de compra (compuesto por diaCompra, mesCompra, anioCompra), estado (que puede tomar los valores “Disponible”, “Alquilada” o “Rota”) y tipo (que puede tomar los valores “Jardín”, “Construcción”, “Electrónica” u “Otro”).

Las herramientas se encuentran guardadas en un array indexado como el que se presenta a continuación:
let herramientas = [… , {id: 1, nombre: “Tijera de podar”, diaCompra: 26, mesCompra: 4, anioCompra: 2020, estado: “Alquilada”, tipo: “Jardín”}, {id: 2, nombre: “Estañador eléctrico”, diaCompra: 23, mesCompra: 8, anioCompra: 2021, estado: “Rota”, tipo: “Electrónica”}, {id: 3, nombre: “Pala de obra”, diaCompra: 16, mesCompra: 12, anioCompra: 2000, estado: “Rota”, tipo: “Construcción”}, …];

De cada alquiler, se guarda: id (un autonumérico único), documento de identidad de quien alquila, fecha de inicio del alquiler (compuesto por diaAlquiler, mesAlquiler, anioAlquiler), cantidad de días que se alquila, la herramienta alquilada (seleccionable a partir de un combo), el precio por día y un atributo que indica si se realizó o no la devolución.

Se pide:

1) Crear las estructuras (clases) necesarias con sus tipos de datos para el correcto funcionamiento del sistema. **(3 puntos)**
    a) Respetar la estructura y los tipos de datos en las partes siguientes. **(2 puntos)**

2) Suponiendo que en el HTML se tiene una etiqueta *<select id=’comboHerramientas’></select>*.
   Crear la función *completarComboHerramientas()*, la cual deberá insertar en el select antes mencionado, las opciones correspondientes para poder seleccionar cualquier herramienta de las que se encuentran en estado “Disponible”.
   Las opciones deberán tener como *value* el *id* de la herramienta y como texto a mostrar, el nombre de la misma. **(10 puntos)**

3) Crear la interfaz HTML (sólo el contenido dentro del *body*) necesaria para poder dar de alta un alquiler. La herramienta se seleccionará de la etiqueta *select* antes mencionada. **(5 puntos)**

4) Crear una función que reciba el *Id* de una herramienta y retorne el objeto si lo encuentra o null en caso de que no lo encuentre. Se debe evitar seguir buscando una vez que se encuentra el objeto y no está permitido cortar la ejecución con *break* o *return* en ningún caso. **(10 puntos)**

5) Crear la funcionalidad js del botón “Registrar alquiler” que crea el mismo y lo guarda en la estructura de datos que corresponda.
   Para poder guardarlo se deberá validar que todos los datos hayan sido completados, y los campos correspondientes a la fecha, a la cantidad de días y el precio por día, sean números mayores a cero. Todo alquiler se dará de alta automáticamente con el atributo booleano que marca si está devuelvo en false.
   NO es necesario controlar que la combinación de los atributos de la fecha sean una fecha válida. **(10 puntos)**

6) Escriba el código de la función reporteDeAlquileres(), la cual deberá retornar un HTML que sea una tabla donde se listen aquellos alquileres pendientes de devolución. De cada alquiler, se debe mostrar el nombre de la herramienta, la fecha de inicio, la cantidad de días y el monto. **(10 puntos)**

Se valorará la utilización de nombres de variables y funciones mnemotécnicos, la optimización de la solución y que se respeten las estructuras que se solicitan utilizar.

En caso de no poder resolver alguna de las partes, se puede suponer como completada cuando se solicite en algún otro punto posterior.

Toda función auxiliar utilizada debe ser codificada y aparecer en la entrega realizada.

Se debe aclarar qué ejercicio se está resolviendo en cada parte.

```
// Ejercicio 1

let herramientas = [];

let proximoIdHerramienta = 1;

class Herramienta {

        constructor(pNombre, pDiaCompra, pMesCompra, pAnioCompra, pEstado, pTipo) {

                this.id = proximoIdHerramienta;

                this.nombre = pNombre;

                this.diaCompra = pDiaCompra;

                this.mesCompra = pMesCompra;

                this.anioCompra = pAnioCompra;

                this.estado = pEstado;

                this.tipo = pTipo;

                proximoIdHerramienta++;

        }

}


let alquileres = [];

let proximoIdAlquiler = 1;

class Alquiler {

        constructor(pDocumentoCliente, pDiaAlquiler, pMesAlquiler, pAnioAlquiler, pCantidadDias,
pIdHerramienta, pPrecioPorDia) {

                this.id = proximoIdAlquiler;

                this.documentoCliente = pDocumentoCliente;

                this.diaAlquiler = pDiaAlquiler;

                this.mesAlquiler = pMesAlquiler;
```

```
            this.anioAlquiler = pAnioAlquiler;
            this.cantidadDias = pCantidadDias;
            this.idHerramienta = pIdHerramienta;
            this.precioPorDia = pPrecioPorDia;
            this.devuelto = false;
            proximoIdAlquiler++;
        }
}

// Ejercicio 2
function completarComboHerramientas() {
        let opciones = `<option value="">Selecctione herramienta...</option>`;
        for (let i = 0; i < herramientas.length; i++) {
            if (herramientas[i].estado === 'Disponible') {
                opciones += `<option
value="${herramientas[i].id}">${herramientas[i].nombre}</option>`;
            }
        }
        document.querySelector("#comboHerramientas").innerHTML = opciones;
}

<!-- Ejercicio 3 -->
<div>
        <input id="txtDocumentoCliente" type="text" placeholder="Documento del cliente"><br>
        <input id="txtDiaAlquiler" type="text" placeholder="cliente"><br>
```

```
        <input id="txtMesAlquiler" type="text" placeholder="cliente"><br>
        <input id="txtAnioAlquiler" type="text" placeholder="cliente"><br>
        <input id="txtCantidadDias" type="text" placeholder="cliente"><br>
        <select id="comboHerramientas"></select><br>
        <input id="txtPrecioPorDia" type="text" placeholder="cliente"><br>
        <input id="btnRegistrarAlquiler" type="button" value="Registrar alquiler"><br>
        <div id="divMensajes"></div>
</div>

// Ejercicio 4
function obtenerHerramientaPorId(id) {
        let herramienta = null;
        let i = 0;
        while (!herramienta && i < herramientas.length) {
                if (herramientas[i].id === id) {
                        herramienta = herramientas[i];
                }
                i++;
        }
        return herramienta;
}

// Ejercicio 5
function validarNoVacio(valor) {
```

```
        return valor && valor.length > 1;
}

function validadNumeroPositivo(valor) {
        return !isNaN(valor) && valor > 0;
}

function btnRegistrarAlquilerHandler() {
        let documentoIngresado = document.querySelector("#txtDocumentoCliente").value;
        let diaIngresado = document.querySelector("#txtDiaAlquiler").value;
        let mesIngresado = document.querySelector("#txtMesAlquiler").value;
        let anioIngresado = document.querySelector("#txtAnioAlquiler").value;
        let cantidadDiasIngresada = document.querySelector("#txtCantidadDias").value;
        let idHerramientaSeleccionada = document.querySelector("#comboHerramientas").value;
        let precioPorDiaIngresado = document.querySelector("#txtPrecioPorDia").value;

        if (
                validarNoVacio(documentoIngresado) &&
                validarNoVacio(diaIngresado) &&
                validarNoVacio(mesIngresado) &&
                validarNoVacio(anioIngresado) &&
                validarNoVacio(cantidadDiasIngresada) &&
                validarNoVacio(idHerramientaSeleccionada) &&
                validarNoVacio(precioPorDiaIngresado)
```

```
        ) {
                let diaNumerico = parseInt(diaIngresado);
                let mesNumerico = parseInt(mesIngresado);
                let anioNumerico = parseInt(anioIngresado);
                let cantidadDiasNumerico = parseInt(cantidadDiasIngresada);
                let precioPorDiaNumerico = parseInt(precioPorDiaIngresado);
                if (
                        validadNumeroPositivo(diaNumerico) &&
                        validadNumeroPositivo(mesNumerico) &&
                        validadNumeroPositivo(anioNumerico) &&
                        validadNumeroPositivo(cantidadDiasNumerico) &&
                        validadNumeroPositivo(precioPorDiaNumerico)
                ) {
                        let alquiler = new Alquiler(documentoIngresado, diaNumerico, mesNumerico,
anioNumerico, cantidadDiasNumerico, parseInt(idHerramientaSeleccionada), precioPorDiaNumerico);
                        alquileres.push(alquiler);
                }
        }
}

// Ejercicio 6
function reporteDeAlquileres() {
        let tabla = `
                <table>
                        <thead>
```

```
                        <tr>
                            <th>Herramienta</th>
                            <th>Fecha de inicio</th>
                            <th>Cantidad de días</th>
                            <th>Monto</th>
                        </tr>
                    </thead>
                    <tbody>
        `;

        for (let i = 0; i < alquileres.length; i++) {
            let alquilerActual = alquileres[i];
            if (!alquilerActual.devuelto) {
                let herramientaActual = obtenerHerramientaPorId(alquilerActual.idHerramienta);
                tabla += `
                    <tr>
                        <td>${herramientaActual.nombre}</td>
<td>${alquilerActual.diaAlquiler}/${alquilerActual.mesAlquiler}/${alquilerActual.anioAlquiler}</td>
                        <td>${alquilerActual.cantidadDias}</td>
                        <td>${alquilerActual.cantidadDias *
alquilerActual.precioPorDia}</td>
                    </tr>
                `;
            }
```

```
        }


        tabla += `
                        </tbody>
                </table>
        `;


        return tabla;
}
```