

EVALUACIÓN	PARCIAL	GRUPO	N1E	FECHA	06/07/2022
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 de alquiler de juegos para fiestas nos solicita el desarrollo de un sistema para gestionar su operativa.

De cada juego se desea guardar la siguiente información: id (un autonumérico único), nombre, precio por hora, ancho, largo y tipo (que puede tomar los valores "Inflable", "Cama elástica", "Toro mecánico" u "Otro").

De los alquileres se conoce: id (un autonumérico único), documento del cliente, cantidad de horas, juego alquilado y precio total (calculado).

Se pide:

- 1) Crear las estructuras (clases) necesarias con sus tipos de datos para el correcto funcionamiento del sistema. Debe también crear las variables necesarias para guardar juegos y alquileres. **(3 puntos)**
 - a) Respetar la estructura y los tipos de datos en las partes siguientes. **(2 puntos)**
- 2) Crear una función que complete un combo con id="comboJuegos", el cual tenga tantas opciones como juegos existan guardados en el sistema. Las opciones deberán tener como value el id del juego y como texto a mostrar, el nombre del mismo. **(7 puntos)**
- 3) Crear la interfaz HTML (sólo el contenido dentro del *body*) necesaria para poder dar de alta un alquiler. El juego se seleccionará del combo mencionado ejercicio 2. Es necesario agregar la etiqueta `select` con el id correspondiente en este ejercicio. **(4 puntos)**
- 4) Crear una función que reciba el id de un juego 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, se debe utilizar la estructura de control repetitiva `while` y no está permitido cortar la ejecución con `break` o `return` en ningún caso. **(7 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, el documento tenga al menos 7 caracteres, la cantidad de horas sea un número mayor a cero y que el juego seleccionado exista. El precio total del alquiler se calcula multiplicando la cantidad de horas por el precio por hora del juego seleccionado. **(15 puntos)**

-
- 6) Escriba la función obtenerMetrosCuadrados(idJuego), que recibe el id de un juego y retorna los metros cuadrados del mismo (largo x ancho). Se puede asumir que el id recibido corresponde a un juego de los guardados en el sistema. **(5 puntos)**
- 7) Escriba una función que reciba un número y retorne un array indexado con los juegos cuyo metraje cuadrado sea mayor al número recibido por parámetro. **(7 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.

1)

```
let tipos = [];  
  
let proximoIdTipo = 1;  
class Tipo {  
  constructor (pNombre) {  
    this.id = proximoIdTipo;  
    this.nombre = pNombre;  
  
    proximoIdTipo++;  
  }  
}  
  
let juegos = [];  
  
let proximoIdJuego = 1;  
class Juego {  
  constructor (pNombre, pPrecioPorHora, pAncho, pLargo, pTipo) {  
    this.id = proximoIdJuego;  
    this.nombre = pNombre;  
    this.precioPorHora = pPrecioPorHora;  
    this.ancho = pAncho;  
    this.largo = pLargo;  
    this.tipo = pTipo;  
  
    proximoIdJuego++;  
  }  
}  
  
let alquileres = [];  
  
let proximoIdAlquiler = 1;  
class Alquiler {  
  constructor (pDocumentoCliente, pCantidadDeHoras, pJuego, pPrecioTotal) {  
    this.id = proximoIdAlquiler;  
    this.documentoCliente = pDocumentoCliente;  
    this.cantidadDeHoras = pCantidadDeHoras;  
    this.juego = pJuego;  
    this.precioTotal = pPrecioTotal;  
  
    proximoIdAlquiler++;  
  }  
}
```

2)

```
function completarComboJuegos() {  
  let optionsHTML = `<option value="">Selecione...</option>`;   
  
  for (let i = 0; i < juegos.length; i++) {  
    const juegoActual = juegos[i];  
  
    optionsHTML += `<option value="${juegoActual.id}">${juegoActual.nombre}</option>`;   
  }  
  
  document.querySelector("#comboJuegos").innerHTML = optionsHTML;  
}
```

3)

```
<div id="divAltaAlquiler">  
  <label for="txtDocumento">Documento del cliente:</label>  
  <input type="text" id="txtDocumento">  
  <br><br>  
  <label for="txtCantidadHoras">Cantidad de horas:</label>  
  <input type="text" id="txtCantidadHoras">  
  <br><br>  
  <label for="comboJuegos">Juego:</label>  
  <select id="comboJuegos"></select>  
  <br><br>  
  <input type="button" id="btnAltaAlquiler" value="Guardar">  
  <br>  
  <p id="pMensajesAltaAlquiler"></p>  
</div>
```

4)

```
function buscarJuegoPorId(idJuego) {  
    let juego = null;  
  
    let i = 0;  
    while (!juego && i < juegos.length) {  
        const juegoActual = juegos[i];  
  
        if (juegoActual.id === idJuego) {  
            juego = juegoActual;  
        }  
  
        i++;  
    }  
  
    return juego;  
}
```

5)

```
function btnAltaAlquilerHandler() {  
    let mensaje = "";  
  
    let documentoIngresado = document.querySelector("#txtDocumento").value;  
    let cantidadHorasIngresadas = document.querySelector("#txtCantidadHoras").value;  
    let juegoSeleccionado = document.querySelector("#comboJuegos").value;  
  
    if (documentoIngresado !== "" && cantidadHorasIngresadas !== "" && juegoSeleccionado !== "") {  
        if (documentoIngresado.length >= 7) {  
            let cantidadDeHorasNumerica = parseInt(cantidadDeHoras);  
            let juegoSeleccionadoNumerico = parseInt(juegoSeleccionado);  
            if (!isNaN(cantidadDeHorasNumerica) && cantidadDeHorasNumerica > 0) {  
                const juegoSeleccionadoObjeto = buscarJuegoPorId(juegoSeleccionadoNumerico);  
                if (juegoSeleccionadoObjeto !== null) {  
                    const precioTotal = cantidadDeHorasNumerica * juegoSeleccionadoObjeto.precioPorHora;  
                    const nuevoAlquiler = new Alquiler(documentoIngresado, cantidadDeHorasNumerica, juegoSeleccionadoNumerico, precioTotal);  
                    alquileres.push(nuevoAlquiler);  
                } else {  
                    mensaje = "Debe seleccionar un juego del sistema.";  
                }  
            } else {  
                mensaje = "La cantidad de horas, debe ser un número mayor a cero.";  
            }  
        } else {  
            mensaje = "El documento debe tener al menos 7 caracteres."  
        }  
    } else {  
        mensaje = "Todos los datos son obligatorios.";  
    }  
  
    document.querySelector("#pMensajesAltaAlquiler").innerHTML = mensaje;  
}
```

6)

```
function obtenerMetrosCuadrados(idJuego) {  
    const juego = buscarJuegoPorId(idJuego);  
    return juego.ancho * juego.largo;  
}
```

7)

```
function buscarJuegosPorMetrosCuadrados(metrosCuadrados) {  
    let juegosEncontrados = [];  
  
    for (let i = 0; i < juegos.length; i++) {  
        const juegoActual = juegos[i];  
        const metrosCuadradosJuego = obtenerMetrosCuadrados(juegoActual.id);  
        if (metrosCuadradosJuego > metrosCuadrados) {  
            juegosEncontrados.push(juegoActual);  
        }  
    }  
  
    return juegosEncontrados;  
}
```