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**	TODOS	**FECHA**	**JULIO 2020 V06**
MATERIA	Programación I				
CARRERA	Ing. en Sistemas. Eléctrica, Electrónica, Telecomunicaciones, Lic. en Sistemas				
CONDICIONES	- Puntos: Máximo: 35 Mínimo: 18 - Fecha máxima de entrega: 14 de julio - Defensa: Se realizará entre el 15 y 27 de julio, en horario de clase, en forma conjunta a la defensa del obligatorio. Ver en Aulas fecha y horario. La no asistencia a la defensa implica la pérdida de todos los puntos correspondientes. LA ENTREGA SE REALIZA EN FORMA ONLINE POR GESTION. IMPORTANTE: - Inscribirse. - El Parcial es **INDIVIDUAL** - Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO". - Verificar que el trabajo quedó correctamente subido.				

El parcial consiste de 3 ejercicios. Se debe a Gestión subir un único archivo .zip o .rar de hasta 40MB con los ejercicios:

- Ej1XXXXXX.js,
- Ej2XXXXXX.js,
- Ej3XXXXXX.js

siendo XXXXXX el número de estudiante.
NOTA: En caso de dudas sobre la letra exclusivamente, se puede consultar por chat de Teams a los docentes de Teórico o dejar anotado en el propio parcial cualquier suposición asumida.

Ejercicio 1: Nombre del archivo: Ej1XXXXXX.js (siendo XXXXXX el número de estudiante). Máximo 7 puntos

En un restorán, a los efectos de mejorar su gestión, registraron todas las llamadas realizadas al mozo desde las 30 mesas numeradas de 1 a 30.
Se tiene un array con los números de mesa que lo fueron llamando durante su horario (cada posición es un llamado):
Ejemplo de llamados: 1 4 3 1 1 4 5 2 5 23 1 1 4 30 26
El mozo, a su vez, fue anotando el número de mesa cada vez que atendía una. No hay errores en los datos (o sea, no hay registros de atenciones que no tuvieron llamado).
Ejemplo de atenciones: 4 1 5 23 1 4 2 1 3
Se desea saber los números de mesa que tuvieron faltantes de atención, ordenados crecientemente y sin repetidos.
Ejemplo: 1 4 5 26 30

La funcion recibe los datos de los llamados y los de las atenciones y retorna un array con los datos solicitados.

Encabezado: function gestionRestoran(llamados, atenciones)

Se pide: Subir un archivo de nombre **Ej1XXXXXX.js (NO PDF, XXXXXX es el número de estudiante)** con el código JavaScript de la función gestionRestoran. Incluir nombre de autor en el código.

Ejercicio 2: Nombre del archivo: Ej2XXXXXX.js (siendo XXXXXX el número de estudiante). Máximo 7 puntos

Un código secreto está formado por dígitos y "*".
Para verificar que es válido, se tiene que cumplir que existe al menos un par de dígitos en el código que cumplen todas estas condiciones:
a) entre ellos hay exactamente 3 "*"
b) no puede haber entre ellos otro dígito
c) los dos dígitos suman 11

Ejemplos: códigos válidos: ***4**2****9*6***5
28***34*5****
código inválido: 38***4*0*0**29

UNIVERSIDAD ORT Uruguay

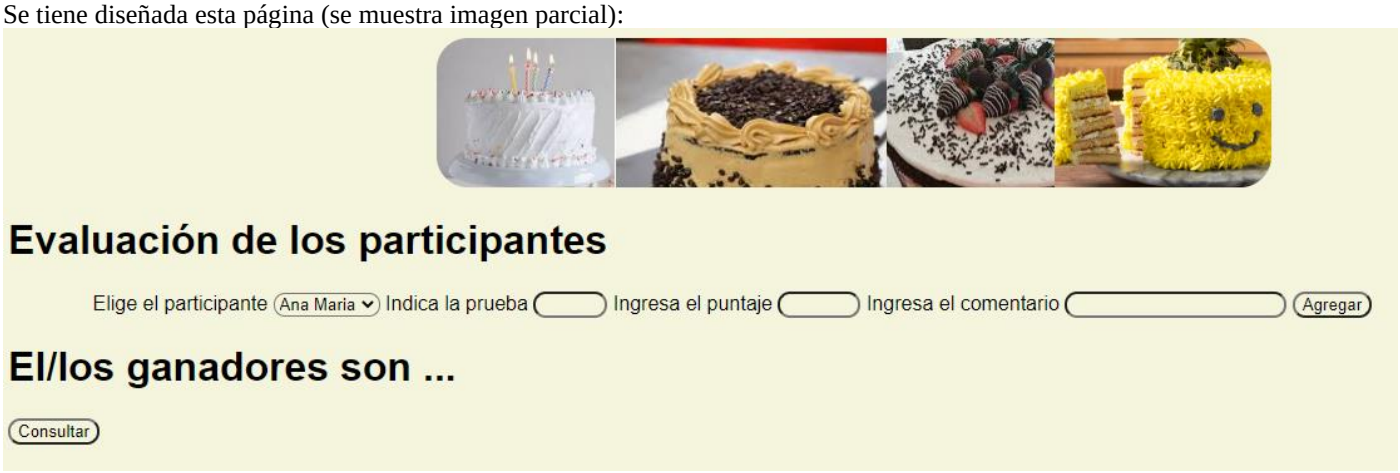
Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Implementar la función validoCodigo que recibe un string con el código secreto y retorna true si cumple las condiciones o false en otro caso.
Encabezado: **function validoCodigo (codigoSecreto)**

Se pide: Subir un archivo **Ej2XXXXXX.js (NO PDF, XXXXXX es el número de estudiante)** con el código JavaScript de la función **validoCodigo**. Incluir nombre de autor en el código.

Ejercicio 3: Nombre del archivo: Ej3XXXXXX.js (siendo XXXXXX el número de estudiante). Máximo 21 puntos (cada parte máximo 7 puntos)

En un concurso de pastelería hay varios participantes. Deben cumplir ciertas pruebas y diversos jueces les otorgan un puntaje y comentario por cada una de ellas. Para saber el/los ganadores hasta el momento, se presiona el botón de consulta.
Se tiene diseñada esta página (se muestra imagen parcial):



Parte del código HTML es:

```
<!doctype html>
<html lang="es">
    <head>
        <meta charset="utf-8">
        <title>Cocinando!</title>
        <link href="css/estilos.css" rel="stylesheet">
        <script src="js/clases.js"></script>
        <script src="js/fuentes.js"></script>

    </head>
```

Se tienen definidas estas clases en el archivo clases.js (el código se muestra parcial, se incluye solamente lo necesario para el ejercicio):

```
class Evaluacion{
 constructor(participante, puntos, comentario, prueba){
   this.participante = participante;
   this.puntos= puntos;
   this.comentario = comentario;
   this.prueba= prueba;
  }
}
```

```
class Participante{
  constructor (nombre, edad){
    this.nombre=nombre;
    this.edad= edad;
    this.evaluaciones = [];
  }
  agregarEvaluacion(evaluacion){
   // recibe un objeto de la clase Evaluacion
    this.evaluaciones.push(evaluacion);
  }
// pedido a
  total(){
  // codigo a completar
   }
 }
```

```
class Sistema{
  constructor(){
    this.listaParticipantes = [];
  }
 // pedido b
  ganadores(){
     // codigo a completar
  }
}
```

a) En la clase Participante, se pide agregar un método que retorne su total. Las pruebas son de la 1 a la 8. El total es la suma de los promedios de las evaluaciones de cada una de las 8 pruebas. Por ejemplo, si en la prueba 1 tuvo evaluaciones con puntaje 5, 2 y 2 (promedio 3), en la prueba 7 tuvo evaluaciones con puntaje 4 y 5 (promedio 4.5), y en las demás no tuvo evaluaciones, su total es 7.5
firma: total()

b) En la clase Sistema, se pide agregar un método que retorna la lista de ganadores. Para ser ganador debe tener máximo total. Si hay varios con el mismo total, se los considera ganadores a todos los que tengan ese total. Importante: asumir disponible la función del punto a). La lista debe estar ordenada por nombre.

firma: ganadores()

c) En la página se tiene (código parcial):

```
<h1>El/los  ganadores son ... </h1>
<input type="button" id ="idConsulta" value="Consultar">
<ul id="idLista">
</ul>
```

En el archivo fuentes.js se tiene (código parcial):

```
window.addEventListener("load",inicio);
let sistema = new Sistema();
function inicio(){
                document.getElementById("idConsulta").addEventListener("click", cargarLista);
}
// parte c)
function cargarLista(){
  // codigo a completar
}
```

Completar el código de la función cargarLista para que cargue en la página la lista de ganadores (asumiendo que la variable sistema tiene todos los datos cargados y que está disponible la funcion del pedido b) ganadores).

Se pide: Subir el archivo Ej3XXXXXX.js (NO PDF, siendo XXXXXX el número de estudiante), dentro de ese archivo poner solamente las 3 funciones pedidas: total, ganadores y cargarLista. Si se utilizan otros métodos, agregarlos e indicar en qué clase se ubican. Incluir nombre de autor en el código.

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

Los principales aspectos a destacar sobre la **entrega online** son:

1. La entrega se realizará desde **gestion.ort.edu.uy**
2. Cada estudiante deberá estar inscripto a la evaluación. **Sugerimos realizarlo con anticipación.**
3. Cada estudiante debe entregar **un único archivo en formato zip o rar**.
4. El archivo a subir debe tener **un tamaño máximo de 40mb**
5. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
6. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, etc)
7. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, contactar con la oficina del Coordinador o Coordinación adjunta **antes de las 20:00hs.** del día de la entrega.
   Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o Secretario Docente.