

## Escuela de Ingeniería

Segundo Parcial de: Programación I

Código de materia:

Fecha: Julio 2019

Grupo: NOCTURNO

Hoja 1 de 2

Duración:

3 horas

Con material: No

Puntaje máximo:

35 puntos

### Ejercicio 1) Carnaval (máximo 21 puntos)

La intendencia desea una única página web para comunicar noticias referidas a las actuaciones de los conjuntos del carnaval, así como sus horarios en los diferentes tablados (de viernes a domingo) y las evaluaciones de la participación.

#### Diseño HTML+CSS

**1a) Se pide (máximo 7 puntos):** Completar el código HTML de la página index.html y la hoja de estilo (estilos.css) para obtener un diseño similar al presentado correspondiente a la parte informativa de los horarios. El código HTML deberá ir dentro de la etiqueta <article>. El resto de la página NO se solicita y se asume disponible.

Archivo **index.html**

```
<article id="idInformacion">
```

**CODIGO A COMPLETAR**

```
</article>
```

Archivo **estilos.css**

**CODIGO A COMPLETAR**

Dentro del artículo, hay un título ("Programación de tablados"), un párrafo ("¡Aquí esta toda la información del Carnaval!"), una imagen (ubicada en img/portada.png), otro título de diferente nivel ("Seleccione el día de programación"), un combo para la selección de día (Viernes a Domingo) y un hipervínculo en la parte inferior de la página ("Ir al tope") que permite ir al tope de la página ("Programación de Tablados"). El título principal tiene como estilo la clase a definir denominada "azul", que corresponde a color "Blue" y tamaño de letra 25 px. El código del color de fondo de la página es #EFF2FB.



#### Código JS y HTML

En la página index.html se tiene un formulario para realizar consultas sobre las evaluaciones que les fueron otorgadas a los conjuntos participantes. Se tiene el siguiente código HTML y las definiciones en JS:

##### HTML parcial:

```
<article id="consulta">
  <h1>Evaluaciones</h1>
  <form id="formConsultas" >
    <label for="idNombre">Nombre de Conjunto: </label>
    <input id="idNombre" type="text" >
```

```
    <input type="button"
      value="Ver evaluaciones"
      id="idBoton">
  </form>
  <h4>Evaluaciones registradas</h4>
  <ul id="idReporte">
    </ul>
</article>
```

## Escuela de Ingeniería

Segundo Parcial de: Programación I

Código de materia:

Fecha: Julio 2019

Grupo: NOCTURNO

Hoja 2 de 2

### JavaScript parcial:

```
class Conjunto{
  constructor(elNombre,estaHabilitado){
    this.nombre = elNombre;
    this.habilitado = estaHabilitado;
  }
}
class Evaluacion{
  constructor(elNombreJurado,elConjunto, elPuntaje){
    this.nombreJurado = elNombreJurado;
    this.conjunto = elConjunto;
    this.puntaje = elPuntaje;
  }
  toString(){
    return "Jurado " + this.nombreJurado + " dio " + this.puntaje;
  }
}
```

```
class Organizacion{
  constructor(){
    this.conjuntos = [];
    this.evaluaciones = [];
  }
  agregarConjunto(unConjunto){
    this.conjuntos.push(unConjunto);
  }
  agregarEvaluacion(unaEvaluacion){
    this.evaluaciones.push(unaEvaluacion);
  }
  darEvaluacionesDeNombreDeConjunto(
    unNombreDeConjunto){
    // METODO A COMPLETAR
  }
}
```

**1b) Se pide: (máximo 7 puntos):** Completar en la clase Organizacion el método **darEvaluacionesDeNombreDeConjunto** que recibe el nombre del conjunto a consultar y retorna un array indexado con las evaluaciones que correspondan a dicho nombre de conjunto. En caso de que existan puntajes que estén fuera del rango correcto (1 a 10) no se tomarán en cuenta, omitiéndose para la consulta. Si el conjunto no existe o no está habilitado, se retornará un array vacío.

**1c) Se pide: (máximo 7 puntos):** Se tiene definido: **let org = new Organizacion();** y se asoció este evento: **document.getElementById("idBoton").addEventListener("click", cargaLista);**. Implementar la función **cargaLista** que toma el nombre del conjunto a consultar, invoca al método **darEvaluacionesDeNombreDelConjunto** y carga información del array resultado en la lista no ordenada en el html. Cada ítem de la lista corresponde a: Nombre de jurado - Puntaje.

### Ejercicio 2: Contaminación (máximo 14 puntos)

Se lleva un registro sobre eventos de contaminación ocurridos en los diferentes departamentos del país (1 a 19). Por cada evento, se conoce su número (1-1000), el número de departamento (1-19), el número de funcionario (1 a 100) que está realizando su seguimiento y un indicador de si fue resuelto o no (true = resuelto y false= no resuelto). Esta información está en un array indexado.

Ejemplo: [117, 1, 45, false, 514, 5, 22,true, ...] corresponde al evento de código 117, ocurrido en el dpto. 1 que está siendo seguido por el funcionario 45 y que aún no ha sido resuelto, el evento de código 514 ocurrido en el dpto. 5 que está siendo seguido por el funcionario 22 y que ya ha sido resuelto....

**2a) Se pide: (máximo 7 puntos):** Implementar en JS la función **conPocos** que recibe el array indexado y retorna un nuevo array indexado con los números de los departamentos que tuvieron menos de 10 eventos. El array debe estar ordenado por código de departamento descendente.

**Encabezado: function conPocos (eventos)**

**2b) Se pide: (máximo 7 puntos):** Implementar en JS la función **porcentajes** que recibe el array y un departamento y retorna un array indexado de 100 elementos, cada posición corresponde respectivamente a un funcionario. El array contiene el porcentaje de eventos que atendió cada funcionario en el departamento. Ej. si un cierto funcionario atendió 40 eventos de los 200 de ese departamento, su porcentaje es 20.

**Encabezado: function porcentajes(eventos, departamento)**